A History of Educational Computing and Software Development for Australian Schools 1970 - 1990

by

Marcus Schmerl

Thesis

Submitted to Flinders University

for the degree of

Masters by Research

College of Humanities, Arts and Social Sciences

Date of completion: July 2020

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

This thesis explores two Australian educational software publishers and their titles during the emerging years of computing in schools through the 1980s – 1990s. Despite some existing literature investigating educational computing in Australia, there is a dearth of research into the environment leading to the creation and production of educational software and games throughout this emergent period of technological change in schools. Additionally, educational software titles have been paid little attention in educational computing historiography.

Existing at the intersection of education, technology, pedagogy, and policy, the research undertaken adopts a multifaceted methodological and analytical approach. Oral histories provided by recent interviews with programmers and educators informs hitherto unrecorded backgrounds, approaches taken, rationales, and intended uses of software created for Australian schools during the introduction of microcomputers during the 1980s. Key software titles (some of which were conceived of, programmed, or ported by the interviewees) and their paratexts were examined, providing examples of themes and trends explored in the interviews. Additionally, textual analysis of archival materials and documentation relating to educational publishers and their software was examined.

Two prominent and prolific Australian software publishers provide the cases for this study. Jacaranda Software, a privately-owned concern based in Queensland, and Satchel Software, an offshoot of the South Australian Education Department, both produced numerous titles for schools during the 1980s through the early 1990s.

These titles spanned the curriculum and adopted a number of instructional and learning models regarding intended classroom use.

The software programmers were, in the main, self-taught educators. Although their backgrounds and educational approaches differed, interviews revealed they all possessed passion for and saw the potential in, and limitations of, the use of computers and software in schools. As the evolution of microcomputers and use cases of software in schools developed, so too did the software in terms of complexity and their various applications in school classrooms. Despite the apparent relative success, high profiles, and prolific number of titles produced by both publishers, they both wound down their production of software for education during the early 1990s. The cessation of educational software publishing by these two companies is a failure, but rather a function of the ever-evolving nature of the pioneering years of educational computing.

DECLARATION

I certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signed: Marcus Schmerl

Date: 20 December 2019

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my initial supervisors on this thesis. My primary supervisor Professor Melanie Swalwell provided my introduction to the wider world of historical software and gaming research and scholarship, to which I am extremely grateful. Her support, inspiration, vast knowledge, motivation, and patience were very much appreciated throughout. I will carry my enthusiasm for this area of scholarship well-beyond the completion of this work, which I look forward to continuing in future. To my secondary supervisor Associate Professor Karen Vered, I also thank her for the encouragement and efforts toward developing my critical thinking and academic writing skills; these words ring true when questioning my analysis and conclusions for any and all of my writing.

To my supervisors during the completion of this thesis, Professor Susanne Schech as primary supervisor and Associate Professor Julia Erhart as secondary, I thank for their encouragement during these final months of writing and editing. The critical eyes, suggested readings, and new perspectives toward this work have been invaluable in shaping it into a final polished piece of research.

I would like to thank my partner Jen for her continued support, patience, and provision of coffee. Additionally, thanks are due to my extended family and workplace, Concordia College who've also supported me greatly throughout this academic learning journey.

1. INTRODUCTION

1.1 Introduction

This thesis records the development of educational computing for Australian schools from the late 1970s to the early 1990s, with a specific emphasis on the emergence of Australian educational software and games. The focus of the research is framed around this early Australian educational software, its development, and uses in the school classroom. My aim is to augment and enrich the developing historiography of educational computing in Australian schools with a study of the emergence and growth of two educational software and gaming publishing companies; namely Jacaranda Software based in the state of Queensland and Satchel Software in South Australia. Both companies established a foothold in the early Australian educational software market across a range of subject areas, producing numerous software and game titles for a variety of 8-bit and 16-bit computer platforms. The software created by both companies employed a variety of educational and pedagogical approaches, as well as teacher and student support materials, for classroom use. In this thesis I draw on interviews with developers and analyse the software and supporting documentation to build the story of this important but hitherto overlooked area of the emergence of computing in Australian schools.

The questions addressed by this thesis are "*How did early Australian educational software development organisations during the 1980s emerge to produce quality software and games for schools? What were the intended educational outcomes and methodological rationales intended for educational software produced during the genesis of educational microcomputing in Australia in the 1980s?"* In seeking to answer these questions, the thesis explores the approaches and ideologies of individual developers, educators, and other contributors. It provides a detailed exploration of key innovations in software design, support materials, and other related published works. In conducting my research, I propose that a historical study of Australian educational software is warranted and summarise the thesis as follows:

The development of early Australian educational software was primarily driven by a small number of enthusiastic educators and publishing companies. This approach resulted in incremental and uneven adoption and changes in educational practice across Australia through the 1970s to the 1990s. The software developed and used in schools was produced by a small group of educators and software programmers, with support from forward-thinking publishers and education departments. Policy directives had minimal impact on the direction of early computer use in education, although structural support provided the environment for growth. Pioneering efforts throughout the 1970s to the 1990s, the evolution of technology in a more homogenous computing market, and technical advances contributed to successful educational publishing.

1.2 Motivations Driving the Study

I have worked as a high school teacher for over twenty years, and prior to that was a school student during the emergence of 8-bit microcomputers in schools throughout the 1980s and 1990s. Looking beyond my professional and personal interests in historical computing, there is a lack of acknowledgement of educational software in the literature surrounding historical computing, software, and games. Video games created for early computing hardware are increasingly gaining recognition and acceptance in historical computing literature (Swalwell, 2007; Suominen, 2011b). Following from this, I believe that early software for schools is a meritorious area of study for numerous reasons. Inspiring this study was a desire to investigate the driving forces, educational culture, and individual motivating reasons behind the software created by people who were hitherto predominately classroom teachers. Beyond this, I wished to investigate further into why individuals and organisations during the 1970s and 1980s believed that investing themselves in classroom use of computers would reap educational benefits. Additionally, using contemporary textual and historical video game research and analysis methodologies in a different context provided an opportunity for further applications and adaptations of current approaches used in historical games studies. Although computer use in schools is now commonplace, modern educational software and games, when used, are generally World Wide Web or app based. Rather than using games for learning, computers are predominately used as a means to an end; for research, production, and creation of student work (word processing, art packages, programming, music). In contrast, the era of 8 and 16-bit computing era was a decade where locally produced born-digital educational software and games encouraging collaboration and higher order thinking flourished. Software from this period can very much be considered products of their time in a unique way, yet still have relevance for teaching and learning in school education in 2020 and beyond. These works are worthy of investigation, analysis, and historical consideration.

1.3 Rationale Behind the Study

As of 2019, computers are ubiquitous in Australian school environments and are utilised for a wide variety of educational purposes. It is no surprise that in order to reach such penetration in educational settings took several decades. Some schools and teachers were early adopters of computing technologies during the 1970s and 1980s (Tannock, 1983; Tatnall, 2014; Tatnall and Davey, 2014). A relatively small number of innovators embraced and experimented with the teaching and learning opportunities afforded by microcomputers. My experiences align with the literature suggesting that a minority of educators were considered early and eager adopters of new technologies prior to the wide distribution of computers and related technologies becoming commonplace in schools (Trinidad, 2002). This cohort tends to embrace new technologies, experimenting and refining how they are used in the classroom, and forging ahead to develop approaches and programmes of study to benefit students' learning. The inertia of bureaucracy, educational policy, and sometimes teacher reticence can reduce the impact and slow the uptake of technologies, specifically computational devices, equally across all schools and the curriculum at large (Hugo, 2000; Baek, 2008).

Computer use in Australian schools is currently (2019) governed by federal and state policies and curricula. Recent discourse around the landscape of Australian education has focused on numerous aspects of computing in education; the Digital Education Revolution (DER), one to one laptop and tablet programmes in Australian schools, the formal introduction to the national Australian Curriculum of Digital Technologies, drill and practice associated with standardised testing (NAPLAN), and pedagogical implications and consequences of the use of technologies in classrooms. Consideration of the use and impact of technologies in educational settings is, however, not just a contemporary issue. In practice, educators, students, and policy makers have grappled with the possibilities and challenges associated with educational computing since the development and availability of early microcomputers. Classroom practice involving computers and narratives of early computer use in Australian schools are currently reported to varying degrees in existing academic literature. Little work has, however focused on the emergence of educational software publishing companies, and the individuals who drove the development of the software and games used in Australian schools.

Contrasting the lack of studies into educational software, the work of Swalwell et al. with the Australian Research Council funded *Play It Again* project highlights expertise and knowledge about early Australian game history. There are, however, fewer structured historical records of born-digital educational materials, their creation, and the experiences of teachers and students using such software. Swalwell & De Vries (2013) indicate there has been little attempt to collate, organise, and document the development of educational software in Australia. Research publications document the introduction of computers in classrooms, although many of these works are derived from a Victorian perspective (Davey & Parker 2010; Jones et al. 2004; Tatnall 2014; Tatnall & Davey 2014). A recent study into New South Wales schools during the 1970s and 1980s (Sollorz, 2013) focused on computer use in Mathematics, but not a larger view of software use in the curriculum. Studies such as these can be described as micro views. State-based and national perspectives can be understood as macro views (Connell 1993; Tatnall & Davey 2008; Walker 1991; Wallace 1989). The literature describes a broad picture of the beginnings of Australian educational computing, though again there is little examination of the history and development of Australian educational software and publishing.

It can be surmised that the production, publication, and uses of educational software titles, particularly those developed in Australia, have mostly been overlooked. This is unsurprising for a number of reasons. Software may have been disregarded due the transitory nature of applications (and hardware) during this period. Additionally, recognition of software and games as historically significant is a relatively recent phenomenon. State and federal government funded reports into computer uses in schools during the 1980s focused on hardware and use cases rather than specific software titles. Subsequent historical studies of Australian educational computing were undertaken several years after early microcomputers had been widely used schools, with software and games not being the focus of this research.

1.4 Scope of the Study

The period of the 1970s to the early 1990s of Australian educational software development was selected for a number of interrelated reasons. Although computing was studied in schools conceptually prior to these years, computers were not deployed to schools in appreciable numbers prior to this time (Connell, 1993). Affordable, single user microcomputers as we recognise them today did not arrive in schools in appreciable numbers until the late 1970s, with educational software use flourishing through the 1980s. Software ranged from titles written by hobbyists or teachers and distributed within individual schools or districts, to titles afforded larger scale publication and distribution. It was during the early 1990s that a number of Australian software publishing endeavours ceased operation. During the 1990s consolidation and homogenisation of the computing industry occurred, with the plethora of hitherto emergent 8 and 16-bit hardware platforms making way for two dominant platforms, DOS / Windows IBM compatibles and the Apple Macintosh line of computers. In addition, the rise of communications technology and the creation of the World Wide Web largely redirected the focus of educational computing through the 1990s and early 2000s. The maturation of the computer industry with bigger players from around the world largely, although not entirely, reduced the creation and impact of Australian educational software after this time.

The formation and proliferation of educational software publishing companies in Australia has not been widely documented and was chosen to be addressed by this research project. The passage of time makes it critical to record the experiences of individuals if their memories and stories of the genesis of educational computing are to be collected and recorded before they are lost. There are two case studies of Australian software publishers that form the body of this thesis. The study initially focuses on the development of software and educational computing in South Australia at the Angle Park Computing Centre (APCC) as an example of a state government supported effort to provide educational computing to students across the state from the 1970s. A case study was undertaken of Satchel Software, which emerged as a publisher during the 1980s from the software developments occurring at the APCC. Jacaranda Software based in Queensland was chosen as a parallel case study due to the variety and quantity of educational software produced for schools across Australia during the equivalent time period. This provided comparison for both similarities and contrasts between the two publishing concerns. The achievements, driving forces (educational and political), and subsequent downsizing of both publishers were investigated. The software developed by both occurred similarly yet in complete isolation to each other; physically the companies were in different states of Australia, and there appears to be little to no crossover with coders or management of both. Additionally, the contexts in which they emerged were very different. Satchel Software was the publishing imprint of the South Australian Education Department, whereas Jacaranda Software emerged out of a privately held enterprise. The study was predominately limited to interviewing the individuals involved with these organisations, and examining software, paratexts, documentation, and publications related to their works. The scope of the interviews and software were restricted to people from and software created by Satchel and Jacaranda in order to gain a deeper understanding of the educational computing landscape as it directly related to these distinct software companies' emergence and success in delivering software to schools across Australia during a decade of flux and evolution in the computing market.

In order to address the research questions, this project examined the foundations of Australian educational software development and use in schools through an approach that deployed mixed methods. The approaches used were qualitative in nature; quantifying aspects of software use such as educational impact and success of software use are beyond the scope of the study. Although quantifying the efficacy or profitability of individual software titles was not incorporated into my research, key innovations in software design, classroom use, and learning approaches were considered in both historical and contemporary contexts. Firstly, I used oral history to explore the lived experiences of individuals who were active participants in the period as software developers, educators, and students. Semi-structured interviews with key figures involved in creating and using educational software provided snapshots of memories and experiences, as well enriching the social and cultural understanding of the realities surrounding this supposed technological shift in school education. Interviews enlivened the historical records provided by existing literature. The experiences of individuals also provide insights into the wider adoption and proliferation of educational computing throughout Australia. Secondly, I examined software, paratextual materials, and publications from the period, with methods of textual analysis and consideration of classroom practice. Analysis of supporting documentation, including software manuals, teacher-guides, and reviews of the works by third party publications was undertaken to supplement the oral histories of educational computer use. This provided a context to the reception of these published works beyond the interviews provided by the developers. Appraisal of the impact of successful (or otherwise) uses of these technologies in educational settings was considered where there was evidence available from interviews, although the scope of the project limited the ability to deeply investigate educational benefits of the titles. Similarly, consideration was given to any enduring contemporary legacies or continued software development of these pioneering efforts.

1.5 Structure of the Study

This thesis contains five discrete yet interrelated chapters. Chapter 2 is an overview of the approaches and research methods used to complete the study. This chapter provides an overview of how the study was conducted. The rationale for the mixed approaches toward analysis of oral histories, textual analysis, and other sources is discussed. Current literature in the field of historical computing research is explored, providing a rationale for the approaches behind the study.

Chapter 3 is a review of existing literature surrounding the history of the use of computers in Australian educational settings, with a focus on schools. In this chapter I survey the existing historical research around the early years of educational computing in Australian schools. Although there is a well-documented history of the introduction of educational computing in Australia, I highlight the lack of historical research into the accounts of development, production, and uses of Australian educational software throughout the 8 and 16-bit era of the 1970s to the 1990s. This provides key arguments for the following two chapters in addressing the gaps in existing literature relating to the emergence of early educational software produced locally for use in Australian schools.

Chapter 4 investigates the work undertaken by the APCC, the emergence of Satchel Software, the development of some of their key software titles, and a discussion of the approaches taken by software developers to augment the utility of the various titles they produced. Chapters 5 is a case study of Jacaranda Software, acting as a contrast to the work covered in Chapter 4; comparisons in software development strategies, approaches of the developers, and types of software were considered.

Chapter 6 provides a discussion of the two companies, their emergence, production, and operations, and a conclusion of the thesis. The findings of this project demonstrate that even with relatively few innovative and forward-thinking teachers with an interest in using computers in schools, their impact on the educational computing landscape in Australia was of historical and contemporary significance. Despite not being in a position to quantify the impact of these pioneers in the field of software development for schools, I argue that the approaches taken by the individuals and companies investigated in this study have relevance both historically as well as in the modern classroom.

2. RESEARCH APPROACH AND METHODS

2.1 Introduction

A number of complementary and interrelated sources of data and approaches to analysis were used throughout this project. The interplay of the actors involved in the study were considered in concert and presented in the two main case study chapters investigating the APCC / Satchel Software and Jacaranda Software. The primary sources used in this study fall into three main groups:

- Oral histories provided by interviews with educators and software creators to document authentic retrospective accounts of contributions to the nascent educational software industry in Australia during the 1980s. Qualitative interviews were conducted either in person in Adelaide and Melbourne, or online via Skype.
- Educational software and game titles were explored and, where possible, played. The titles were usually played via computer emulation due to the challenge of access to and variety of 1980s' hardware platforms that they were created for.
- Archival research and textual / document analysis of software instructions and user guides (hereafter described as paratexts), periodicals, publications, videos, and policy documents of the era was undertaken.

2.1.1 Primary Data Sources – Interviews / Oral Histories

Seven interviews were undertaken for this project with pioneers in the field of educational software development in Australia. User histories are a departure from most histories of computing, where the focus is on 'big' machines rather than the stories of users and early adopters (Swalwell, 2015). These interviews documented the lived experience of those working with computers in schools and creating software that was used by Australian school students in the 1980s. When reconstructing the stories surrounding computing, it "reminds us that the history of computing is not just about computers, but about the interaction of computers and people" (Lean, 2013). Oral histories were derived from interviews that were interpretive and not purely reconstructive; the motivations and aspirations of the software creators help drive the stories of the two companies and the software that was created. The "subjectivity of individual memory" was an asset whereby the "understanding of individual and collective memory can be augmented" (Green and Troup, 1999). Interviews were semi-structured directed conversations with a framework of questions that were tailored to suit the background and history of the interviewees. Rather than be overly long and prescriptive, there was a list of initial standard questions followed by discussions shaped by the individual's area of interest and early responses. Interviews helped to inform the identification of key educational software (and vice versa), how the titles were developed, intended and actual classroom use, with the underlying pedagogical rationale being discussed when appropriate. The interviews were conducted in person, via email, via Skype, or other similar methods, were recorded in both audio and video format where possible, and were transcribed according to the protocol outlined in Ethics Approval Number 5404 "A History of Creative Computing in Australasia".

The interviewees, who were informally or self-taught similar to most early programmers (Swalwell, 2008; Lean, 2012), were chosen in order to provide unique perspectives on educationally focused software development. The educational and teaching backgrounds of the interviewees were explored, where relevant, to develop the stories of how their interest in computers, associated technologies, and approaches to teaching shaped their forays as some of the pioneers of Australian educational software development. Furthermore, many of these individuals were involved in the formation of and activities undertaken by the first formal educational computing support groups, educational software publishers, and educational computing conferences. Documentation, where available, detailing the formation of such groups comprised of journals and newsletters was subsequently obtained and examined. These interviews provided insights into the genesis of educational software development in Australia and provided greater context surrounding the development of the software that was investigated.

2.1.2 List of interviewees

Numerous software developers and educators were identified and targeted for interviews. Sources for the preliminary list of potential interviewees included existing academic literature, developer listings from the *Australasian Digital Heritage* website (Swalwell and De Vries, no date; *Australasian Digital Heritage Database*, 2016), gaming history or archival websites that included screenshots of software titles often displaying the developers' names, and Australian library listings. Physical or digital copies of software titles and their documentation, once obtained, also revealed key individuals via credit lists revealing contributors to software creation. Many titles of the era included the developers' names on the title or credits screen, which assisted in both informing potential interviewees and confirming the software as produced in Australia. Interviews, once conducted, also assisted in identifying further potential interviewees for future projects and research in this area. Emails and phone calls were sent explaining the purpose of the project, with interviews organised in subsequent communications. Some potential interviewees were unavailable or were unable to be contacted, potentially affording opportunities for further studies in this field. The final cohort of interviewees, briefly detailed below, provided ample data for this project and were appropriate and relevant for the main case study chapters.

- Dean Hodgson: Hodgson's background was as teacher in South Australia whose interest in computing in education led him to work for the Angle Park Computing Centre and create software titles for Satchel Software. Hodgson currently works for the South Australian Education Department.
- Dean Rosenhain: Rosenhain was also a teacher in South Australian who worked for the Angle Park Computing Centre and created software titles for Satchel Software. Rosenhain currently runs his own (non eduational) software company.
- Philip O'Carroll: A teacher and coder based in Perth and then Melbourne, O'Carrol wrote numerous titles for Jacaranda Software. He currently runs a private school and educational software / book company.
- Rosanne Gare (nee Hood): Gare was involved with Jacaranda Software early in her teaching career. She was exposed to early 8-bit microcomputers during her pre-service studies, and she subsequently designed a number of open-ended adventure titles suited to primary classroom use.
- Gerald Wluka: A school student at the time of his involvement in writing titles for Jacaranda Software, he was employed by Jacaranda software to port a number of existing titles to the Apple II.

In addition to the interviews conducted for this study, I have utilised transcripts of interviews previously conducted by Melanie Swalwell and Helen Stuckey with former employees of Jacaranda Software, namely Bruce Mitchell, David L. Smith, and Steve Luckett.

2.1.3 Primary Data Sources – Australian Educational Software and Paratexts

Compared with video or computer game ‘entertainment software’, there are fewer fan websites dedicated to the preservation and celebration of notable early educational software and paratexts. Additionally, the academic literature surrounding Australian educational computing generally does not explore specific software in detail; rather it investigates introduction, implementation, and hardware. Furthermore, software titles produced in Australia have been absent from much of the literature discussing born-digital materials thus far. There were, however, a number of relevant and valuable entries on the Play it Again website and other fan-based software websites and repositories (Gamebase64.com, Lemon64.com, archive.org) provided a useful starting point to identify educational software created in Australia (*The Internet Archive Software Collection*, 2015; *Gamebase 64*, 2016; *Lemon 64*, 2016). Corroborating evidence in user manuals, teacher notes and guides, publications and newsletters, software reviews, and marketing materials, plus information provided by interviews, assisted with the selection of appropriate software to investigate. Original copies of Australian educational software titles were obtained from libraries, repositories, and personal collections (some from the original developer) from across the country. A number of programs across a range of genres were sourced. Additionally, some Australian educational software has been salvaged by fan efforts, with digital copies of some titles available for download. The software titles ranged across numerous genres and types; this was intentional to demonstrate the wide variety of, and rationale behind, educational software that was produced in Australia during the 1980s and early 1990s. The types of software investigated were extremely varied, and include those that were:

- Subject-specific, for example Mathematics and English focused programs.
- Drill and practice titles requiring the completion of repetitive tasks to learn knew knowledge or skills.
- Cross-curricula titles that were intended to develop a variety of thinking, logic, problem-solving and group-work skills.
- Titles that could be classified as games, varying widely from the those that were based on arcade or home video game titles to adventure and puzzle game.
- Application software (word processors, spreadsheets, database management systems) that had been adapted or programmed specifically for school use.
- Networking software, allowing students access to Bulletin Board Systems, services, and software in an era before ubiquitous access to the Internet and World Wide Web.

The chosen titles were published by Satchel Software in South Australia and Jacaranda Software based in Queensland. These two publishers provide the basis for the scope of the two main comparative case-study chapters that form the backbone of this study. There were a number of influential factors driving this decision during the early phases of the study despite their being numerous educational publishers in Australia during the 1980s. Both Satchel and Jacaranda were prolific publishers of software, with both publishing dozens of titles during their early years of educational computing. They published titles across a range of 8-bit and 16-bit computers in use in schools and the home, unlike some other companies that only published for a single platform (for example, Microbee only published titles for their own hardware).

Both companies were strongly educationally driven in both content and suggested classroom use of the software, with most programmers interviewed having personally been involved in school teaching as well as the introduction of computers. Although the success of both publishers cannot be quantified in terms of units of software sold, evidence suggests that they sold their titles throughout Australia and overseas. There were a number of distinguishing contrasts between the two companies that were revealed throughout the investigation that were explored. Satchel Software was a publishing arm of the South Australian Education Department, whereas Jacaranda Wiley was a privately held company. They were based in two separate states of Australia whose governing political ideologies at the time were almost diametrically opposed, with South Australia having a progressive government and Queensland being more conservative. The various commonalities, differences, and perspectives of the people producing the software, and the software itself, were explored throughout both case studies. Beyond the software produced and published by the two companies who are the focus of the study, some educational software from other publishers, or titles that were self-published and distributed by the programmer, were also explored where relevant.

Paratextual materials directly related to the software titles that were studied were obtained; these include included numerous booklets, teacher notes, instructions, and resource packages. Analysis of related paratexts is especially valuable for born-digital educational software (Desrochers and Apollon, 2014). Manuals, teacher notes, and student worksheets were just some of the items that provided valuable evidence showing how software was used in schools. Some paratexts were included with the software packages and were often substantial and detailed. These paratexts variously related to loading of software, teacher instructions, student instructions, suggested classroom use of the software, activities for students before, during, and after their time using the software at the computer, and other supporting activities to build upon the intentioned use of the software. Due to the new and exploratory nature of the introduction of microcomputers into classrooms, it is unsurprising that such documentation was included with software as many teachers and students in the 1980s were unfamiliar with general computer operations. Other related paratexts were produced in conjunction but sold or provided separate to software titles. These include materials for teaching of databases, programming in LOGO, and supplements to adventure games (Angle Park Computing Centre, 1985; J. S. C. C. Committee, 1986; Walsh, 1987).

2.1.4 Primary Data Sources – Archival Educational Documentation

There were numerous sources of additional archival educational computing documentation relating to Australian educational computing, and the software produced by Satchel and Jacaranda, that were obtained and analysed for this study. Copies of a number of Australian computing publications (both professional and hobbyist) were obtained. Resources produced by the Angle Park Computing Centre in South Australia, prior to the formal launch of Satchel Software as an 'offshoot', were obtained. Reports on the early use of computers in Australian schools during the 1980s, were also sourced. Where original materials could not be obtained, online repositories were used to source some documents. A number of relevant historical publications are preserved online, including the Australian Council for Computers in Education Journal (*Journal - Australian Council for Computers in Education*, 2016) and the archives of the Australian Society for Educational Technology (*Archives of the Australian Society for Educational Technology*, 2016). The magazine section of archive.org provided evidence of potential 'key' Australian software titles being noted or reviewed in mainstream print media about computing. An example is in The Commodore Annual (*Commodore Annual 1990 (No 3), Saturday Magazine Pty Ltd*, 1990a; *Commodore Annual 1990 (No 3), Saturday Magazine Pty Ltd*, 1990b), a general computing magazine published in Australia, which included a multi-page article highlighting numerous educational titles created by Satchel Software. Analysis of these publications helped foster understanding of not just what software was used, but how it was used, with reviews highlighting perceived benefits and weaknesses of the titles being used in classrooms. Finally, the influence of educational, curriculum, and political forces were investigated, although the impact of these on the programmers and the software titles studied was relatively minimal. Studies, policy documents, and government reports about the influence and impact of educational computing from the 1970s through the 1980s were sourced (Sandery, 1975; Education Department of South Australia, 1985; Hayton and Loveder, 1989).

2.2 Analysis

This thesis chronicles an account of the Australian educational software landscape throughout the 1980s framed by two successful and prolific publishing companies of the time. Expanding on the key questions informing the context and content of the study when conducting interviews, investigating software, and interpreting paratextual and other documentation can broadly be broken down as determining:

- What was the intended educational purposes of the software created?
- What methodological and pedagogical rationales were present behind the software design?
- How was the software intended to be used as part of educators' classroom practice?

Interrelated questions of interviewees connecting their educational and programming backgrounds, types of software, and their intended use, educational and methodological beliefs, views as to the extra context and experience it brought to students and the classroom, as well as the software itself, paratextual, and other published archival material all helped to develop the narrative and themes explored in this study.

The primarily emic and inductive nature of the analysis of the sources assisted the development of emerging themes as they related from 'small' stories to overarching statements about computers in Australian education through the 1980s. Identification of recurring themes, categorising, and describing phenomena was used to determine the patterns in the data, which provided an "interpretive portrayal of the studied world" (Charmaz, 2003; Charmaz and Belgrave, 2012; Sollorz, 2013). This helped in developing the accounts and narratives surrounding the Australian educational software creation and publishing landscape in the 1980s and early 1990s, specifically relating to the two companies that were investigated. The multi-faceted nature of data collection and approaches to analysis allowed for *triangulation*, “not the simple combination of different kinds of data but the attempt to relate them so as to counteract the threats to validity identified in each” (Wiltfang and Berg, 1990). More specifically, between-method triangulation, namely comparison, correlation, and contrast between oral histories, textual analysis, and archival research organised in the two case studies, was used to examine the development and use of early Australian educational software.

The study covers both personal and small local stories, in this case individual teachers and programmers and their experiences in creating educational software and attempts to build a larger picture of the publishing companies and the software produced. Although building a broad picture of the wide and uneven Australian educational computing landscape is beyond the scope of this study, historical narratives describing computer software by the creators themselves and the intended educational use cases were explored. Suominen’s Game Studies research considering historical and retrospective discourses (Suominen, 2011a) are considered throughout this thesis. Historical metanarratives describing a “grand chronology” are challenging to address in this study, however retrospection discourses of individual stories are “nostalgic, but also reflective and even critical”, helping to build bigger historical stories. Additionally, Swalwell’s work on the validation of microcomputing and oral histories “embracing the popular” (Swalwell, 2015), looking beyond nostalgia to critically assess and articulate the value of historical computing software and games (Swalwell, 2007), and amateur coders as makers

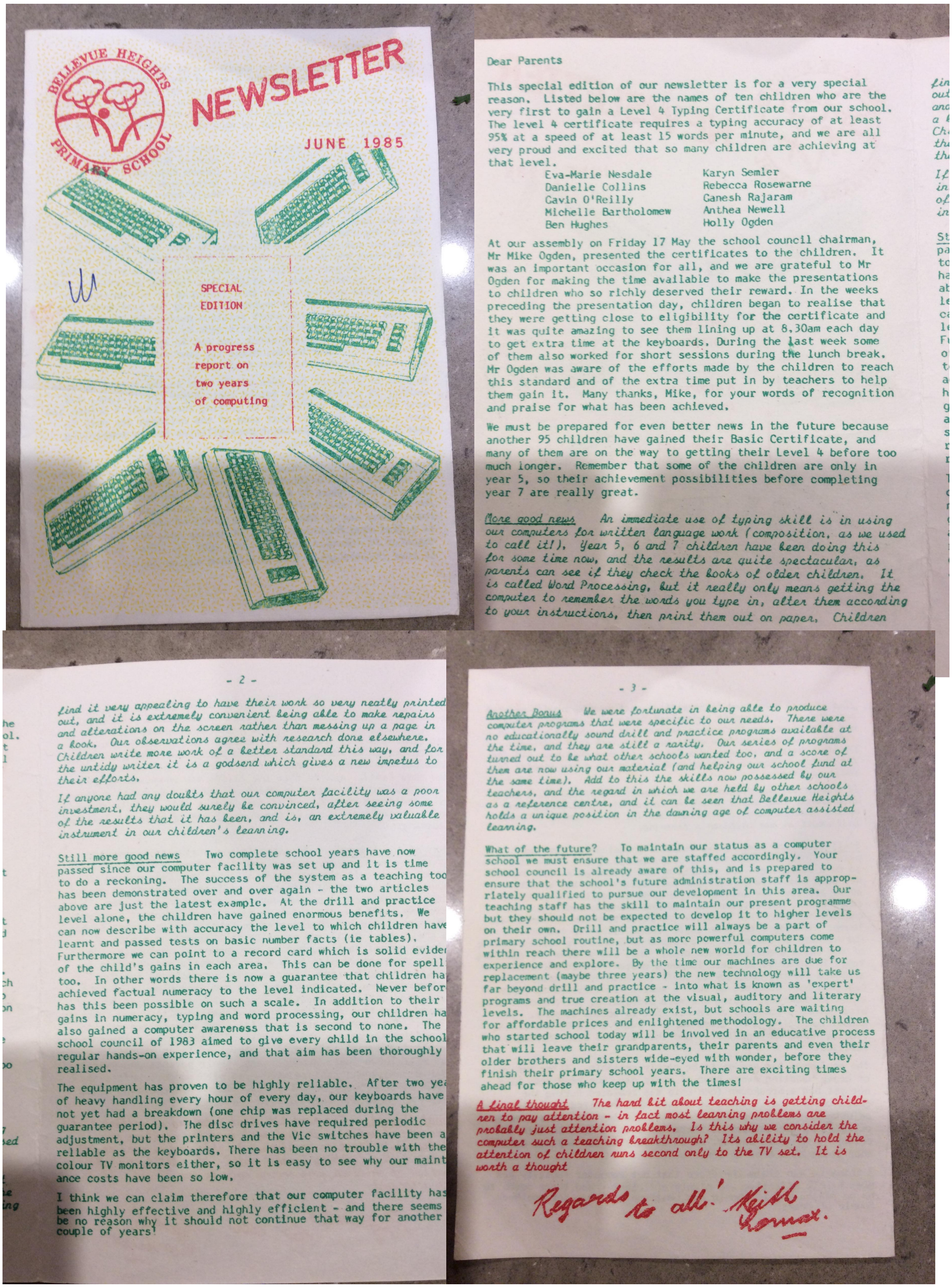
(Swalwell, 2008) strongly influenced my research. Relating to these influences of Swalwell, I investigated the processes of development of early educational software, improvements in subsequent software design, and highlighted the interrelationship of software with of classroom practice during. Determining and quantifying to what degree and how classroom practice change when computers were installed in classrooms of Australian schools was however limited to the evidence provided by the limited number of interviewees, the two companies explored, and the selected subset of software investigated. Opportunities for future studies in the area of the emergent Australian educational computing landscape of the 1970s through the 1990s are explored in the concluding chapter of this study.

3. THE DEVELOPMENT OF EDUCATIONAL COMPUTING IN AUSTRALIA

3.1 Introduction

In this chapter I review the existing literature surrounding the early years of Australian educational computing. I have investigated endeavours, accomplishments, and notable aspects of educational computing, while emphasising the lack of focus in existing studies on educational software as of historical importance. The review of literature presented is positioned at the intersection of education, computing technology, teaching practice, and educational policies (where they existed) relating to the early development of educational computing in Australian schools. The chapter identifies gaps in existing literature surrounding the examination of software titles, creators, and publishers. My narrative emphasises that although key events and influencers regarding computer use in Australian schools are documented, there is a lack of historiography surrounding educational software. This gap will be addressed in the remaining chapters of this thesis.

In my own twenty years of teaching in schools the advancement of computer uses and consequent adjustments in classroom practice have been steady and incremental rather than showing rapid change. Looking further back, this mirrors my own experience of computing in South Australian schools throughout the 1980s and 1990s; computer hardware and software was used in similar ways during my primary and secondary years for tasks predominately involving word processing, typing, programming, and design. Through the lens of a student during this time period, the major innovations were the relatively 'early' introduction of microcomputers themselves (1983) and the in-house production of software by staff at the school I attended (Figure 3.1 shows a newsletter summarising two years of computer use in my primary school). I personally had comparatively little exposure with educational software titles that were not produced within my primary school. Discovering that there was a large number of educational titles published by Australian companies was surprising, as was the lack of coverage of their development and use in historical



BELLEVUE HEIGHTS PRIMARY SCHOOL

NEWSLETTER

JUNE 1985

SPECIAL EDITION

A progress report on two years of computing

Dear Parents

This special edition of our newsletter is for a very special reason. Listed below are the names of ten children who are the very first to gain a Level 4 Typing Certificate from our school. The level 4 certificate requires a typing accuracy of at least 95% at a speed of at least 15 words per minute, and we are all very proud and excited that so many children are achieving at that level.

Eva-Marie Nesdale	Karyn Semler
Danielle Collins	Rebecca Rosewarne
Gavin O'Reilly	Ganesh Rajaram
Michelle Bartholomew	Anthea Newell
Ben Hughes	Holly Ogden

At our assembly on Friday 17 May the school council chairman, Mr Mike Ogden, presented the certificates to the children. It was an important occasion for all, and we are grateful to Mr Ogden for making the time available to make the presentations to children who so richly deserved their reward. In the weeks preceding the presentation day, children began to realise that they were getting close to eligibility for the certificate and it was quite amazing to see them lining up at 8.30am each day to get extra time at the keyboards. During the last week some of them also worked for short sessions during the lunch break. Mr Ogden was aware of the efforts made by the children to reach this standard and of the extra time put in by teachers to help them gain it. Many thanks, Mike, for your words of recognition and praise for what has been achieved.

We must be prepared for even better news in the future because another 95 children have gained their Basic Certificate, and many of them are on the way to getting their Level 4 before too much longer. Remember that some of the children are only in year 5, so their achievement possibilities before completing year 7 are really great.

More good news An immediate use of typing skill is in using our computers for written language work (composition, as we used to call it!). Year 5, 6 and 7 children have been doing this for some time now, and the results are quite spectacular, as parents can see if they check the books of older children. It is called Word Processing, but it really only means getting the computer to remember the words you type in, alter them according to your instructions, then print them out on paper. Children

- 2 -

find it very appealing to have their work so very neatly printed out, and it is extremely convenient being able to make repairs and alterations on the screen rather than messing up a page in a book. Our observations agree with research done elsewhere. Children write more work of a better standard this way, and for the untidy writer it is a godsend which gives a new impetus to their efforts.

If anyone had any doubts that our computer facility was a poor investment, they would surely be convinced, after seeing some of the results that it has been, and is, an extremely valuable instrument in our children's learning.

Still more good news Two complete school years have now passed since our computer facility was set up and it is time to do a reckoning. The success of the system as a teaching too has been demonstrated over and over again - the two articles above are just the latest example. At the drill and practice level alone, the children have gained enormous benefits. We can now describe with accuracy the level to which children have learnt and passed tests on basic number facts (ie tables). Furthermore we can point to a record card which is solid evide of the child's gains in each area. This can be done for spell too. In other words there is now a guarantee that children ha achieved factual numeracy to the level indicated. Never befor has this been possible on such a scale. In addition to their gains in numeracy, typing and word processing, our children ha also gained a computer awareness that is second to none. The school council of 1983 aimed to give every child in the school regular hands-on experience, and that aim has been thoroughly realised.

The equipment has proven to be highly reliable. After two yea of heavy handling every hour of every day, our keyboards have not yet had a breakdown (one chip was replaced during the guarantee period). The disc drives have required periodic adjustment, but the printers and the Vic switches have been a reliable as the keyboards. There has been no trouble with the colour TV monitors either, so it is easy to see why our maint ance costs have been so low.

I think we can claim therefore that our computer facility has been highly effective and highly efficient - and there seems be no reason why it should not continue that way for another couple of years!

- 3 -

Another Bonus We were fortunate in being able to produce computer programs that were specific to our needs. There were no educationally sound drill and practice programs available at the time, and they are still a rarity. Our series of programs turned out to be what other schools wanted too, and a score of them are now using our material (and helping our school fund at the same time). Add to this the skills now possessed by our teachers, and the regard in which we are held by other schools as a reference centre, and it can be seen that Bellevue Heights holds a unique position in the dawning age of computer assisted learning.

What of the future? To maintain our status as a computer school we must ensure that we are staffed accordingly. Your school council is already aware of this, and is prepared to ensure that the school's future administration staff is appropriately qualified to pursue our development in this area. Our teaching staff has the skill to maintain our present programme but they should not be expected to develop it to higher levels on their own. Drill and practice will always be a part of primary school routine, but as more powerful computers come within reach there will be a whole new world for children to experience and explore. By the time our machines are due for replacement (maybe three years) the new technology will take us far beyond drill and practice - into what is known as 'expert' programs and true creation at the visual, auditory and literary levels. The machines already exist, but schools are waiting for affordable prices and enlightened methodology. The children who started school today will be involved in an educative process that will leave their grandparents, their parents and even their older brothers and sisters wide-eyed with wonder, before they finish their primary school years. There are exciting times ahead for those who keep up with the times!

A final thought The hard bit about teaching is getting children to pay attention - in fact most learning problems are probably just attention problems. Is this why we consider the computer such a teaching breakthrough? Its ability to hold the attention of children runs second only to the TV set. It is worth a thought

Regards to all!

literature.

Figure 3. 1: School newsletter highlighting two years of computer use, 1985.

3.2 The early history of computing in Australian schools

"The key trend in this history of educational technology is that with each new technology, a revolution is expected. With each new technology social expectations have been high, but actual classroom use has been low and / or problematic." (Howard and Mozejko, 2015)

The introduction of computers into Australian schools was an uneven process throughout the 1970s and 1980s. Access to computers in schools was not uniform and inconsistent, both across the country and within states, cities, and school districts (Fitzgerald, Hattie and Hughes, 1986; Wallace, 1989). Disruption and advancement in approaches to student learning and classroom practice in Australian educational settings as a consequence of computing technologies has consequently taken many years to emerge. Rapid changes in education, and specifically schools, as a consequence of technological advancements are often overstated, with numerous influencing factors behind this. Inertia of teaching methodologies relating to pedagogical practices, unwillingness to be perceived as inadequate, and fear of the unknown have been found to reduce the integration of computers in teaching practice (Morton, 1996). Costs and logistics of introducing new technologies were also challenges for schools to address, along with the choice of a wide array of incompatible hardware platforms during the late 1970s through the 1980s (Anderson, 1984). Computers were relatively expensive, and often had little to no software to be used out of the box. Once microcomputers became more commonplace and affordable, questions of what to do with them and how to use them in schools came to the fore.

Despite these hurdles, computing was embraced in some Australian schools and educational settings during 1960s. Computation and programming courses were introduced in some Australian universities as early as the 1940s through 1950s (Pearcy, 1988). Studies in computing were mostly theoretical during this era, and by 1960 there were still only four computers used in tertiary educational settings throughout the whole of Australia (Connell, 1993). Computer technologies were used in schools as early as 1960, although initially programming and concepts were also taught theoretically due to the paucity of hardware (Connell, 1993). Subsequent processing improvements and decreasing physical size and cost caused a rapid increase in their use. Early educational computing user groups of hobbyist teachers were established during the 1960s and 70s, with the Education Department of South Australia considered to have implemented some of the first structured support for schools, teachers, and students in 1968 (Tatnall and Davey, 2008; Tatnall, 2013). Even though there was no obvious or direct link to existing curricula, enthusiastic and forward-thinking educators appreciated the value and utility of new technologies (Layton, 1972). Connell (1993) notes that by 1970 there were over one hundred computers used in educational settings in Australia. Following this period, early programming courses were implemented, and eventually the first Higher School Certificate (HSC) subjects in Computer Studies commenced, although this too did not occur uniformly across Australia.

Before microcomputers became commonplace and affordable, some of the first Australian schools to teach programming and computing topics in the 1970s used time-share computing systems. Minicomputers were located remotely at a universities or other institutions, and eventually locally at schools. These computers did not allow for one on one computing, and programs were 'written' or problems solved with holes placed on punch cards to be processed in batches. One early example of such a system was MONECS, developed by Monash University, where school students would fill in punch cards at school with simple program logic and then send these cards to the university to allow the system to determine program functionality and errors (Davey and Parker, 2010). This would involve a frustrating several day turn-around time. Similar approaches were also used by South Australian schools, details of which are explored in the next chapter. Educators in Tasmania utilized an innovative idea of physically moving a DEC minicomputer from school to school to allow periodic access to computers, although the logistical effort to provide this service was time-consuming (Jones, McDougall and Murnane, 2004). Eventually Tasmania moved to a statewide time-share network, as did Victorian schools, where using the existing phone network and teletype machines at schools to send programs provided much faster turn-around and feedback for students.

There was no national plan or program regarding computer use in Australian schools during the late 1970s and early 1980s, and variation among the states was considerable (Connell 1993; Robson et al. 1991; Tatnall 2013; Tatnall & Davey 2014; Wallace 1989). User groups and some government supported groups formed the first formal educational structures supporting computer use in Australian schools. An early documented use of microcomputers in Australian schools is in Victoria during 1976, where a Commonwealth grant led to hardware acquisition[1] and software creation. Since educational software did not exist, some teachers created their own programs (Walker, 1991). Walker describes how two teachers, Greg

---

[1] Specifics about hardware platforms procured in this grant are currently unknown.

Johnstone and Tim Mowchanuk, received the first Commonwealth Schools Commission Innovation Program grant in 1976 to further their interest in using microcomputers for educational purposes. This however came with the requirement of determining appropriate hardware, programming (almost) all of the software, and publishing a periodical to share their work with other interested parties. The resulting publication was called "COM-3", a journal that circulated across Australia after the official formation of the Computer Education Group of Victoria (CEGV) in 1978 (Jones, McDougall and Murnane, 2004). On the instigation of CEGV, state-based, and ultimately national computer in education conferences were held, resulting in the formation of the National Committee for Computers in Education (now the Australian Council for Computers in Education). These conferences provided "a forum for the sharing of innovation, ideas and practices in the use of learning technologies. They have proved to be a vital part of the professional development of educational computing practitioners at all levels" (McDougall and McCrae, 2000). Noted attendees include Seymour Papert, the developer of the LOGO educational programming language and the theorist behind the Constructionist approach of "learning as making", a development of Piaget's Constructivist approach of "learning as doing"[2] (Papert and Harel, 1991; Ackermann, 1996, 2001; Papert, 1996).

During the early forays into teaching with computers, a number of formal and informal teacher groups were established across Australia. This resulted in development of some preliminary hardware standards and collaborative development

---

[2] It is worth noting that Constructivist and Constructionist approaches to learning, classroom practice, and teaching programming in the 1970s still strongly informs my own approaches to teaching Digital Technolgies to high high school students as part of the Australian National Curriculum in 2019.

of educational software, curriculum, and teaching practices. These user groups formed within states as early as 1960 and across states by the early 1980s, consequently facilitating more rapid software development and support than official channels could (Tatnall and Davey, 2004). One example, the Computers in Education Group of South Australia (CEGSA), was formed in 1985 to provide support and training for educators in the use of Information and Communication Technologies (ICT) (EdTechSA, 2016). This group still provides professional in-service and training for educators, currently under the name of EdTechSA. In addition to local groups, intrastate and interstate collaboration amongst teachers allowed them to make considered decisions relating to computer platform and software choice. South Australia's Angle Park Computing Centre and Tasmania's Elizabeth Computing Centre, both state government supported initiatives, in conjunction with Western Australian teachers, set up the TASAWA consortium to share resources and expertise between the three states in the early 1980s (Newall, no date). Some standardisation of hardware platforms and resource sharing between the states of Tasmania, South Australia, and Western Australia was formalised in 1981 (Tatnall and Davey, 2008). The TASAWA consortium ultimately agreed on the BBC microcomputer, produced in England, as a hardware standard which helped streamline decision-making around software adoption. This minimised compatibility issues that plagued other states until their own user groups were set up and standards set. Participating in the TASAWA consortium also aided collective software and curriculum development and yielded improved prices on hardware and software purchases. These hardware standards were not necessarily adopted uniformly. One example is that by the mid-1980s South Australian schools were using a range of hardware platforms including the Commodore, Apple, and Amstrad CPC computers in addition to the BBC (Fitzgerald, Hattie and Hughes, 1986; Wheeler, 2008). Indeed the 8-bit computer standard in South Australia was later revised in the late 1980s to the Amstrad CPC line of microcomputers. Because of this this, individual schools with particular computer variants still required software and hardware support appropriate for their type of microcomputers. This fractured install base of a variety of different types of microcomputer continued across Australia through the 1980s and 1990s as the microcomputer market continued to evolve, which necessitated the need for educational software to be created for multiple platforms.

Tasmania developed and implemented numerous educational technologies during the formative years of educational computing. The Elizabeth Computer Centre (ECC) was introduced in 1975 (Connell, 1993; Tatnall, 2013). TASNET, a statewide timesharing network for educational purposes, was also established at this time. TASNET could operate in both online and batch modes, and was accessed by the schools, their libraries, and the State Library (Pullen, 1981) (Figure 3.2). By 1983 Tasmania was noted for having computers in half of the primary schools (Report, 1984), well ahead of most other states. Scott Brownell, a Tasmanian teacher, introduced the LOGO programming language into Australia in 1975 (McDougall, Murnane and Wills, 2014). The magnetic tape copy of LOGO was obtained from MIT, and ran on a PDP-11 minicomputer owned by the Tasmanian Education Department (Richardson, no date). Following from its early introduction into all school levels in Tasmania (Mckerrow, 1982), early robot turtles were introduced, educator Sandra Wills toured and introduced LOGO to many schools, the software was converted to have an on-screen virtual turtle, and ports to many variants of microcomputers were written and sold. After Papert visited in Australia in 1981, numerous locally written LOGO programming books were sold worldwide throughout the 1980s (McDougall and McCrae, 2000; Jones, McDougall and

Murnane, 2004). The bulk of the computers in Tasmanian schools during the 1980s were BBCs, with Apple also gaining a foothold (Fitzgerald, Hattie and Hughes, 1986).

Image removed due to copyright

Figure 3. 2 TASNET, adapted from Anderson (1984).

Despite some early innovations and drivers of educational computing, the state of Victoria could have been considered comparatively slow in formal adoption of educational computing (Tatnall and Davey, 2008). During the late 1970s organised support for the provision of computers in schools was limited to individuals, small teacher groups, and some state Education Department initiatives for educational computing advocacy. Some of these initiatives included a travelling computer ‘road show’ in 1979, a discretionary fund for limited purchases of computer equipment for some schools, the introduction of computer awareness courses in the early to mid 1980s, and the first HSC subjects in Computer Science around the same time (Walker, 1991; McDougall and McCrae, 2000; Tatnall and Davey, 2014). The Apple II (late 1970s) and Commodore (mid 1980s) were the dominant microcomputers during these periods in Victoria (Wallace, 1989). The Victorian State Computer Education Centre (SCEC) was set up in early 1984 as a “centralised support unit”, and provided services as diverse as curriculum development, resource sharing, evaluation and distribution of educational software, and recommendation of hardware platforms for schools (Tatnall and Davey, 2008).

Although the provision and installation of computers in Australian schools being unequally distributed and decentralised based on the above historical narratives I consider that the overall growth of educational computing progressed relatively quickly and independently across Australia. The formative years of educational computing was somewhat frontier like with little governmental or policy oversight (Tatnall, 2016). This afforded teachers and schools with autonomy when implementing new teaching practices using computers. The aforementioned educational user groups of teachers, many of whom were personally computer enthusiasts and hobbyists, furthered the introduction of computers in Australian schools throughout the 1970s and 1980s before relevant state and federal educational policies were written (Bigum, 2012). Many teachers invested time and money acquiring computers and creating educational software, as well as experimenting with how computers could support teaching and learning. Collaboration with local educational bodies facilitated rapid decision-making and software development

(Tatnall and Davey, 2004). Consequently, computer use in classrooms rose steeply during the early 1980s, often in unexpected areas. Hedberg and McNamara (2002) suggest Australia's use of computers in education in many ways mirrors other western countries, but the country's unique geography and educational factors have resulted in excellence of the use of educational technology. One Australian specific the School of the Air[3], operating across Australia for over seventy years allowing rural students to access formal education which would otherwise have been very difficult due to being isolated from traditional schools. Computer-centric examples include innovative applications of telecommuting including electronic mail and bulletin board services (BBS) (Hosie, 1985; Robson, Routcliffe and Fitzgerald, 1991) during the 1980s, long before this was considered commonplace, and the development of an aspirational but ultimately abandoned purpose built proprietary Australian Educational Computer (Tatnall and Leonard, 2010; Tatnall, 2013). This computer was conceived of in order to address the fractured hardware and software market in Australian education. This was similar in purpose to other education focused computers, including the Acorn BBC (UK), ICON (Canada), and Poly (New Zealand). That this series of computers was not produced was probably fortuitous as numerous 16-bit computers (Commodore Amiga, Atari ST, Apple Macintosh) and IBM compatibles were introduced during the mid 1980s. Had this project continued, the 8-bit AEC would have been more costly and less powerful than contemporary machines, and the breadth of educational software on these platforms would have dwarfed its library by the time it ultimately launched.

[3] The School of the Air refers to numerous correspondence schools operating across Australia since the 1950s where students on isolated properties corresponded with teachers and students via two way radio.

Notwithstanding what is known about the introduction and use of computers in Australian schools, and that teachers created software to be used to suit their needs, existing academic literature provides few details about what this early software was or how it was used. Due to the passage of time since the late 1970s, much of the earliest educator-created software would either be physically lost or non-functional, or there would be difficulty in reading or accessing it from tape or disk. I decided to investigate publishing companies that sold and distributed locally produced software once microcomputers were more firmly established as a presence.

3.3 Australian institutional, policy, and implementation contexts

There is evidence that computers were considered beneficial for students when used in schools, although analysis of early research in Australian schools raised many questions including appropriate pedagogies, access and equity, teacher training, and the overall role of the microcomputer in education (questions that are still asked to this day) (Maggs and Ray, 1985). Wallace (1989) presented summary findings from numerous ACEC conferences, as well as a 1986 Commonwealth Department of Education report, with the assertion that computers were generally used to support existing pedagogical practices rather than transforming education noticeably (Fitzgerald, Hattie and Hughes, 1986). There were a number of notable findings from these reports. Almost all secondary schools across Australia were using computers by the mid 1980s, with an average of fourteen machines per institution, yet just over half of primary schools had computers, averaging only three computers per institution. The patterns of use were somewhat different in primary schools, where simulations, games and Logo programming were three predominant uses, compared to secondary schools where more programming and database use took place. Word processing and general ‘computer awareness’, however, were common across all years.

Federal Government policy support and direction arrived as late as 1983 (Wallace, 1989; Connell, 1993; Beale, 2014). Despite the non-uniform yet reasonably quick uptake of technology in education and by educators, Australia was relatively slow to implement federal policy relevant to computers in schools. One of the earliest noted examples of federal policy was proposed during the 1983 election campaign (Beale, 2014). Following a Schools Commission report, the establishment of the Commonwealth Schools National Advisory Committee on Computers in Schools (NACCS) was recommended, and was set up in early 1983, providing funding and recommendations for the future provisions of computer use (Tannock, 1983; Tatnall, 2014). Relative to other countries such as the USA and UK, this was rather late, and it was suggested only a portion of a proposed $125 million funding over five years was distributed for direct use in schools over a truncated three year period (Wallace, 1989; Beale, 2014). At the time, Australian states autonomously allocated such Federal funding and developed their own school curricula relating to educational computing. Australian curriculum in this area was only standardised during the last few years of the 2010s when the Australian National Curriculum for Digital Technologies was finalised and implemented for all schools across the country.

Further investigation found gender issues, a scatter-gun approach to educational computing implementation, lack of professional development for teachers, and wide diversity in computer platforms were issues of concern during the 1980s and 1990s (Kinnear, 1995). Kinnear found boys responded more positively to computers than girls, with girls less convinced of the utility of computers in the classroom. Wallace (1989) also summarised that attitudes toward computers and gains in targeted areas, such as reading, were generally more favourable in male students. Management style and approach was also noted to be a factor influencing the success of implementation of computers in the curriculum (Schiller, 1991). Schiller found that principals who were initiators of change and possessed clear policies and vision for a school beyond implementation of innovation tended to achieve more with technologies rather than those who adopted the stance of managers of change or those who introduced computers as a response to the opportunity. McNamara (1985) suggested teacher-driven research as solution to the onslaught of educational technologies, but this was impractical to be adopted on a wide scale, let alone being able to replace or greatly influence state and federal jurisdictions. Early research into computer use in schools revealed that there were positive student outcomes including active learning, motivation, and deeper reflection by students who learn both "from" and "with" technologies (Reeves, 1998). A later report proposed dozens of recommendations, some of which were very similar to previous reports regarding research into efficacy and teacher training (Smith, 1994). One recommendation of note in this report was the suggestion of a creation of national computer software development centre(s), but there is no evidence of this coming to fruition, and indeed the software and development explored in this thesis relates to titles and events prior to 1994. Bigum (2012) describes a cyclical nature of technology acquisition by schools over the last thirty years. Bigum suggests that schools investing in computer technologies are caught in "a cycle of identifying, buying, and domesticating the "new best thing" driven largely by claims that the process is ultimately improving learning". This trend was borne in mind when conducting interviews and reviewing software. Beale's PhD and subsequent book (2014) takes a critical discourse analysis approach (mostly adapted from Fairclough and Wodak amongst others) where she investigates language, meaning, social location, power relations, differing world views and value systems to examine Australian educational computing policies as they developed over the years; some of these approaches were considered for this project when analysing historical documents, software, and paratexts.

3.4 Actors in early educational computing – computers, publishers, and software developers

As the complexity and speed of computers developed, their size and costs subsequently decreased, thus making them more suitable for use in schools. A single minicomputer could take up several cubic metres of space, and they were prohibitively costly for schools to purchase. Conversely, the development of 8-bit microcomputers created and marketed by Apple Computer, Commodore, Atari, Amstrad, and a myriad other companies during the 1970s and 1980s aided the proliferation and installation of computers in schools. Microcomputers were defined by their hitherto low prices, small physical footprints, built in keyboards, and the ability to use televisions or small monitors as output devices.

Commonwealth funded research into computer use in schools in the mid 1980s indicated there were still over ten computer platforms being used in schools across the country (Fitzgerald et al. 1986). By 1985, most states tended to have two dominant platforms, with one considered ‘expensive’ at the time and the other more affordable (see Figure 3.3) (Wallace, 1989). This is illustrated in the eastern seaboard states (Queensland, New South Wales, Victoria and the Australian Capital Territory), where the relatively expensive Apple II held over one third of the educational market. The bulk of the remaining computers in schools were divided amongst less expensive machines (Microbee, Commodore, Tandy, and Atari). Tasmania was a notable exception, with the two more expensive platforms dominating (Apple and BBC). The market, advertising by companies, incentives offered to schools, and the educational conditions leading to each state’s unique platform choice will be investigated. It is interesting to note, for example, that Apple were the most prolific supplier of computers to schools in all states except for members of the TASAWA Consortium. The Australian-developed Microbee computer was used in schools throughout Australia in the 1980s. It was particularly successful in New South Wales, and was also sold in Sweden, Denmark and Russia (Wordsworth, no date; Fitzgerald, Hattie and Hughes, 1986). Due to the efforts of the Microbee Software Preservation Project (*Microbee Software Preservation Project (MSPP)*, 2016) many of the educational titles created for the Microbee still exist as archived disk images. Cross-referencing this software with the entries catalogued in the Australasian Digital Heritage Database (*Australasian Digital Heritage Database*, 2016) yielded over one hundred results for Microbee software published by Australian companies including Goodison Software, Caresoft, Honeysoft, Flying Fox Software, and Jacaranda Wiley.

Computer	*NSW*	*Vic*	*Qld*	*Tas*	*SA*	*NT*	*WA*	*ACT*
Apple II	**37**	**36**	**52**	**25**	**18**	**38**	4	**56**
Apple Macintosh	2	0	1	0	3	0	1	0
Microbee	**32**	3	9	2	0	3	**19**	1
BBC	2	8	3	**54**	13	6	**34**	**2**
Atari	5	1	0	0	0	3	0	0
IBM	1	0	0	0	0	0	0	0
Tandy	3	4	**11**	0	3	3	5	0
Ohio	0	0	2	0	1	0	0	0
Commodore	8	**32**	6	3	**57**	**29**	12	1

Figure 3. 3: Percentage distribution of computers by state, highlighting the two most popular computers in Australian schools (mid 1980s). Adapted from Wallace (1989).

Table reused under Fair Dealing for Criticism or Review

The dearth of software suitable for each state and their disparate school curricula during these early years required that teachers write their own educational software. Evidence suggests much of the early computer use in Australian schools was in the areas of mathematics, science, and for programming by students (Jones, McDougall and Murnane, 2004; Tatnall and Davey, 2008). Throughout the 1980s, word processing, databases, graphics, simulations, and subject specific software use increased (Wallace, 1989). I searched and sorted Australian software collated and catalogued by the Australasian Digital Heritage Database for educational software across all computing platforms. This indicated hundreds of individual educational software applications written in Australia during the 1980s for a range of microcomputers. Although the variety of choice of new microcomputers available during the 1980s grew, there was often a degree of similarity in the underlying hardware. This allowed programmers to more easily create software targeted for multiple machines. Similar central processing units (CPUs) such as the 6502 used in the BBC microcomputers, Apple II, Commodore, and Atari, and the Z80 used in the Sega SC-3000, Amstrad CPC, and MicroBee, allowed software to be ported across multiple machines. Cross-referencing a number of the online software database repositories mentioned previously indicates there were individual programmers and specific software publishers who were prolific in creating educational software across a range of platforms.

One of the more prominent publishers was Satchel Software which was established by the South Australian Education Department. The department saw the potential of computing in schools, and set up one of the country's first teacher groups in the 1960s and the Angle Park Computing Centre (APCC) (Tatnall and Davey, 2008). In Jolly's anthology (2001), teachers from Angle Park Boys Technical High School and the Parks Community Centre recounted their personal experiences of educational computing through the 1960s and 70s (Phillips 2001; Fisher 2001; Veale 2001; Appleton 2001b). These recounts paint Ian Appleton as a key figure in the introduction of computing, acquiring an 'obsolete' IBM computer for the school. "Dimension: A Magazine about Computing in Secondary Schools" explained how APL, an early programming language, was taught to students, in addition to the school running in-service conferences for teachers from across the state (Appleton, 2001a; *Dimension: A magazine about computing in secondary schools*, 1972). The Angle Park Computing Centre initially used IBM 1130 minicomputers which were later succeeded by an IBM 370 mainframe (Newall n.d.; Tatnall & Davey 2008). The early uses of these machines were similar to the Melbourne / MONECS experience, allowing students to complete their 'programming' on punch cards and have printouts of results returned the during the following days. The work undertaken by the teachers and programmers at the APCC led to the formation of Satchel Software, and also the development of the pioneering NEXUS Online Curriculum Service online information system in the early 1990s. Satchel Software created dozens of educational software titles and supporting materials for 8 and 16-bit microcomputers during the 1980s and 1990s. The work pioneered by the APCC and Satchel Software supported the development of educational computing in South Australia, and their software was distributed and sold around the country and overseas. There is currently little in-depth exploration of these South Australian endeavors, and this is explored as the focus of Chapter 4.

Brisbane-based educational software publisher Jacaranda Software, the Australian arm of U.S. publishing company Wiley, similarly published numerous educational titles during the 1980s and early 1990s. One software example examined in literature of Australian games is *Raft-Away River*, a popular multi-platform educational title written by a team of developers employed by Jacaranda Software (Stuckey *et al.*, 2013a, 2013b). It was designed for language development and communication, yet encouraged "leadership, communication, cooperative behaviour and strategic planning", allowing for a wide range of classroom experiences (Stuckey *et al.*, 2013b). The development team behind software created numerous original programs, in this particular instance one designed by Roseanne Hood (nee Gare) and David L. Smith, and ported them to other 8-bit systems (for example Phillip O'Carroll for the Commodore 64, Tony Zadravec for the Microbee). This multi-author approach is associated with numerous other Jacaranda adventure, strategy, and simulation educational titles including *Gold-Dust Island*, *Quick-Cartage Company*, and *Cunning Running*. Greygum Software, established by a number of former employees of Jacaranda, still produce and distribute educational software to this day (Stuckey *et al.*, 2013a). This is another story of formalised Australian educational software development that has not been widely covered in existing literature and provides a counterpoint to the stories of Satchel Software. Chapter 6 will distill the are similarities, parallels, and contrasts of educational software creation by publishers who existed at the same time and produced similar born digital content, yet were conceived of and operated completely independently of each other.

3.5 Conclusion

Throughout this chapter I have identified a dearth of historical studies focusing on the development of Australian educational computing, namely software titles, their development, and publishers. Whilst substantial academic literature about historical use of computers in schools exists, there has been little focus on software creation and the emergence of educational computing publishers. Existing overviews of educational computing paint a broad picture of the educational computing landscape in Australia during the period of the introduction of 8-bit microcomputers during the late 1970s and 1980s. These are generally larger scale accounts of the emergence of (or summarising) educational computing as a whole. Additionally, there are numerous of studies looking at snapshots of individual personal accounts of the introduction of educational computing, generally from teachers and students of the time. These small stories can be designated as micro-histories. The identification of gaps in existing literature surrounding the examination of software titles, creators, and publishers has highlighted the area of educational software that I have addressed. There is little investigation into the individuals and the environments that cultivated innovation to fostered leading to successful publishing and proliferation of educationally focused software produced in Australia. There is also little historiography of the various software titles created during this period. The next two chapters investigate the people and the software, and the publishers during this brief but prolific period of the 1980s and early 1990s in the early history of Australian educational software production.

4. ANGLE PARK COMPUTING CENTRE AND SATCHEL SOFTWARE: THE ORIGINS OF EDUCATIONAL SOFTWARE IN SOUTH AUSTRALIA

4.1 Introduction

South Australia is recognised as one of the pioneers of educational computing in Australia (Tatnall and Davey, 2014).The focal hub of much of the state's introduction and advancement of computing was at the Angle Park Computing Centre (APCC) located at Angle Park High School. The APCC was set up by South Australian educators as one of Australia's first teacher support groups in the 1960s (Tatnall and Davey, 2008). Acting in a coordinating role as early as 1968, the APCC provided a "service to all schools, government and non- government, as well as a service for some years to the Northern Territory" (Anderson, 1984). This chapter argues that the APPC's functions of familiarising and training students and teachers with computers and their use throughout the late 1960s through to the mid-1980s in South Australia were fundamental to the development of educational computing in this state. State government backing provided the APPC with the autonomy and authority to develop educational computing programmes and policy. The APCC produced support materials for students and teachers, influenced curriculum, and created innovative learning courses as computer use in schools evolved. In contrast, schools in other states used a combination of titles from the "equipment supplier, software companies, or teacher-developed software" (Anderson, 1984). A clear distinction is the emergence of Satchel Software as a prominent publisher that developed out of the APCC. By the early 1980s microcomputers had become more commonplace both in homes and schools, and the focus of the APCC consequently evolved from offering a supporting and training role to educational software publishing.

During the early 1980s, the Satchel Software publishing arm grew from the APCC. Dozens of educational software titles were published by the APCC under the Satchel Software imprint across numerous 8-bit and 16-bit computing platforms used throughout the 1980s and early 1990s. These software packages included original titles conceived by individuals and teams. Additionally, the APCC ported a number of existing and successful educational titles from the United Kingdom and published them under the Satchel Software brand. These titles originated on the Acorn BBC Micro and were ported to a number of other 8-bit platforms, providing further evidence of the APCC's success and impact. The titles created under the Satchel Software brand ranged from educational adventure and arcade games, subject specific programs, and productivity packages (including word processing and database tools). The *NEXUS* online information system (1987), one of Satchel Software's final releases enabled South Australian schools to participate in a pioneering networked telecomputing environment.

By the mid-1980s there were numerous publishers in Australia specialising in educational software design, creation, and publishing. Another such publisher was Jacaranda Software, examined in Chapter 5, provides a counterpoint and contrast to the history of the APCC. Despite Satchel Software branded products being sold across Australia and some licensed for sale overseas, the development endeavours of the APCC were ultimately wound down. I make a case that the demise of Satchel was in response to changing market conditions, the maturation and consolidation of computer platforms, refocussing of government resources into other educational endeavours, and the introduction of new technologies (including the impact of the Internet).

The APCC and Satchel Software's position and growth in the educational software market of the 1980s bear hallmarks of recognisable aspects of the emerging

8-bit computer market. The developers and software designers had backgrounds in education and computing, and they were generally self-taught having learnt how to code out of interest and realising the potential of educational computing. There was a hobbyist and experimental approach toward software creation and use, with testing and refinement of titles often taking place with other teachers at the APCC or even students in classrooms. The nature of educational software production in the APCC and Satchel followed most of the characteristics of homebrew development (Swalwell, no date)[4], especially in the formative years before Satchel's educational software production became more formalised. As the software increased in complexity, accompanying activities, examples, and other supplementary materials became more elaborate. The packages were increasingly professionally produced, typeset, and branded with the Education Department of South Australia insignia, all of which contributed to legitimising microcomputing in Australian classrooms during these pioneering days of computers in schools. APCC programmers were afforded autonomy to create educational titles based on their own interests and perception of educational need. There was little influence from external educational policies or bureaucratic interference as the APCC teachers and programmers were tasked with shaping and developing the directions of educational computing in South Australia. Constraints on their output were minimal during the creative design and production process; the primary focus of the programmers was on the classroom use and the utility of their products. Much of the software created by Satchel allowed for cross-

[4] Swalwell defines homebrew development as having five characteristics – domestic location, amateur programmers, sole creators, local distribution, and experimental ethic. Obviously, the work was carried out with the support of an institution, but much of the output of the APCC and Satchel (especially early titles) strongly reflects the other four characteristics.

curricula and open-ended classroom use, in addition to a number of subject-specific titles produced.

The story of the evolution of the APCC as it developed from an officially registered school into a prominent educational software publishing arm presents several questions. How did the individuals, predominantly teachers, get involved in computing? Why was the Satchel Software imprint launched as a publishing arm of the APCC? How did educational concerns influence program design when creating software? What were the educational benefits of their software packages? In this, the first of two chapters examining case studies of Australia's educational software publishers, I discuss the pioneering efforts of the APCC. This chapter demonstrates that the APCC and the directions Satchel Software took were primarily driven by the educational utility of their products created throughout their operating life. The presence of the APCC as a centralised educational computing hub, with programmers working autonomously to create quality educational software contributed to the realisation of Satchel Software's brief but prolific period as an educational software publisher.

The chapter draws on interviews conducted with APCC programmers, examines software and paratexts, revealing the increasing variety of uses and approaches of computers as educational teaching and learning tools throughout the 1980s and the early 1990s. These documents also show how locally developed software packages were tailored (when appropriate) to suit the local Australian education market. I argue that these pioneering endeavours were impactful, valid, and educationally relevant. It took time before these educational influences spread while computers in schools became more pervasive. This provides the lingering question of whether ceasing production of local developed educational titles by Satchel Software was premature.

4.2 Data Sources

Interviews, software, and paratexts are the primary sources analysed in this chapter. As discussed in Chapters 2 and 3, there is little scholarly research specifically investigating Australian educational software from this period, and very few mentions of Satchel Software were found in existing literature. There is no official complete back catalogue of Satchel's work, nor is there any accessible repository of their titles and supporting paratextual materials. Exploring the holdings of a number of 8-bit software repositories and databases[5] yielded several dozen results for educational software titles published by the APCC and Satchel Software in South Australia.

Oral histories with two former teacher-developers (Dean Hodgson and Dean Rosenhain) enrich what little published historical information exists about the APCC and Satchel. Semi-structured interviews were conducted during 2017. Hodgson is a former teacher and prolific programmer, both prior and subsequent to his time creating software under the Satchel brand (1984-1991). Rosenhain worked with APPC (1982-1988) during the conception of the Satchel Software brand and wrote a

[5] Online repositories include the Australasian Digital Heritage Database, Gamebase 64 (2016), and CPCRulez (2015).

number of pieces of educational software, including the *NEXUS* online information system.

Numerous APCC and Satchel Software titles were obtained from private collections and from software held in libraries across Australia. Accessing these titles, and procuring physical copies them where possible, allowed me to determine a more complete and representative back catalogue and collection of software for analysis. In addition, I was able to obtain a number of titles and teacher / student support materials from the private collections of Hodgson and Rosenhain. Although the entirety of the Satchel Software oeuvre was not available, a representative selection of their software (from numerous 8-bit and 16-bit platforms), and instructional and support materials was reviewed. The software examined in this chapter ranges from the simple to more complex and advanced works, typifying the rapid advancement in software complexity and utility during the 1980s. The arcade style *Math Booster*, open-ended problem-solving adventure games *Granny's Garden* and *Jara-Tava – The Isle of Fire*, *Pathweaver – An Adventure Game Generator*, and the office suite *Forté – The Integrated Package* are the software packages explored in this chapter. My exploration of these titles and their supporting documentation exposed the comparatively swift evolution in complexity of the educational software produced by the APCC under the Satchel Software imprint. Finally, the *NEXUS* online information system[6] and the challenging development task to design and implement the software is also examined. Software was run and inspected on original hardware, where possible, or using emulators. In the case of *NEXUS,* which could no

[6] *NEXUS* was an online service for South Australian schools. It was designed to allow communication and provide access to educational materials for both teacher and students.

longer be run, documentation and a video of use cases were analysed. The software documentation for all titles was valuable in situating the educational and methodological rationale driving classroom uses, for open-ended, creativity, logic, and higher order thinking.

4.3 An Exploration of the Angle Park Computing Centre and Satchel Software

4.3.1 The Angle Park Computing Centre – 1960s to 1980

Angle Park is a suburb in north western Adelaide. Angle Park Boys' Technical High School first opened in 1961 (Veale, 2001), later merged with Angle Park Girls Technical High School in 1977, and in 1979 became known as the Parks High School as part of the Parks Community Centre (*The Parks Community Centre - Background*, 2013). In Erica Jolly's anthology (2001) exploring the history of technical and vocational educational in South Australia, several teachers from Angle Park recounted their personal experiences of teaching at the Angle Park schools through the 1960s and 1970s (Phillips 2001; Fisher 2001; Veale 2001; Appleton 2001b). Anecdotes from Gordon Phillips and R. W. A. (Wal) Fisher, both former principals at Angle Park, paint Ian Appleton as a key figure in the introduction of computing in South Australian schools. Appleton was initially employed as a temporary assistant at Angle Park, but his efforts resulted in the school acquiring an 'obsolete' IBM computer, which served as a catalyst for the establishment of Angle Park Computing Centre. Phillips (2001) noted that during the late 1960s:

> [Appleton] added computing and the boys were encouraged to write programmes which Ian, studying computing at The University of Adelaide, arranged for a lecturer to feed students' programmes through the university's computer...he had put bits and pieces into a

> spare room for boys to play around with including an adding machine.…Ian persuaded IBM to give the school one of its obsolete computers, IBM was not supposed to do this…Angle Park was the first secondary school in Australia to have its own computer and later the computing centre at The Parks supported computer teaching throughout the state... I always believed that Ian Appleton didn't get the acknowledgment he deserved for all his work setting up this programme. Someone in the office took the credit.

This quote is indicative of the innovation of singular individuals creating an impact within an educational environment that fostered and explored the initiation of computer use in South Australian schools in the 1960s.

The Angle Park Computing Centre was set up to "co-ordinate the development of computing in [schools in] the state" (Wallace, 1989). The APCC initially used an IBM 1130 minicomputer (Angle Park Computing Centre, 1975) acquired by Appleton in 1968. This computer was later succeeded by an IBM 370 mainframe (Newall n.d.; Tatnall & Davey 2008). In June 1973, the second stage of the APCC was launched. In addition to the local computers at the APCC, an official arrangement with IBM afforded the use of a computer in Canberra connected to terminals in some South Australian schools[7]. Then South Australian premier Donald Dunstan delivered a brief speech to officially open the APCC (Dunstan, 1973)

---

[7] The number of schools which had these terminals, as well as how long they were used for is currently undetermined. It is reasonable to assume that they would have remained in use throughout the 1970s, but such terminal use is likely to have declined by the mid-1980s as microcomputers rendered school use of remote access to a mainframe unnecessary.

(Figure 4.1). Dunstan's speech was decisive and pointed toward a range of benefits that the technology could provide to students. He suggested that the APCC was taking the lead in educational computing in Australia, providing many benefits to South Australian schools. Computers had the capacity to assist students across a range of subjects, with the advantage of helping them learn about new technologies. Dunstan foreshadowed the centrality of computing to education, "student-centred learning", and how "open space units" would be complemented and supported by computer use. The tone of the speech is aspirational and forward-thinking (Dunstan's legacy is widely recognised to be socially progressive), punctuated with several references to IBM, conveying legitimacy by association with a 'big iron'[8] company. To this day, several features of the discourse surrounding introduction of new technologies in education often maintain these characteristics of aspiration.

[8] Big iron is a somewhat obsolete term referring to mainframe or supercomputers.

SPEECH NOTES FOR THE PREMIER, MR. DUNSTAN, OPENING ANGLE PARK COMPUTER CENTRE PROJECT. 18.6.73

MR. JONES, MR. HOLMES A'COURT, DR. FORD, MR. APPLETON:

TODAY'S CEREMONY MARKS THE SECOND STAGE IN AN EXCITING AND PIONEERING DEVELOPMENT IN COMPUTER EDUCATION IN SOUTH AUSTRALIA.

THE SOUTH AUSTRALIAN EDUCATION DEPARTMENT HAS BEEN A LEADER IN THIS FIELD FOR THE PAST SIX YEARS. INDEED, NO OTHER STATE EDUCATION DEPARTMENT HAS YET REACHED OUR STAGE, AND THIS LATEST DEVELOPMENT ENSURES THAT WE WILL MAINTAIN OUR LEAD.

THE EDUCATION DEPARTMENT AND I.B.M. AUSTRALIA LTD. HAVE COMBINED RESOURCES TO INSTALL REMOTE TERMINALS IN A NUMBER OF ADELAIDE SECONDARY SCHOOLS, AND THESE ARE LINKED TO AN I.B.M. COMPUTER IN CANBERRA.

2.

THIS NETWORK ALLOWS STUDENTS TO USE THE COMPUTER AS AN AID TO THEIR STUDIES IN SCIENCE, ECONOMICS AND MATHEMATICS, AS WELL AS GIVING THEM AN APPRECIATION OF COMPUTER TECHNOLOGY ITSELF.

IT IS A PROJECT THAT I AM SURE WILL BE STUDIED BY EDUCATIONISTS THROUGHOUT AUSTRALIA BECAUSE OF ITS FAR REACHING IMPLICATIONS IN LEARNING SKILLS. I THINK IT'S VERY POSSIBLE THAT, IN THE NOT TOO DISTANT FUTURE, ALL STUDENTS WILL USE SIMILAR EQUIPMENT, AND THAT COMPUTER TAPES WILL BE AS IMPORTANT IN SCHOOLING AS CHALK AND BLACKBOARD.

IT IS PART OF THE NEW ERA OF STUDENT-CENTRED LEARNING WHICH, WITH DEVELOPMENTS SUCH AS OUR NEW LIBRARY COMPLEXES AND OPEN SPACE UNITS, MAKE SCHOOLING EXCITING AS WELL AS REWARDING, AND I HAVE MUCH PLEASURE IN DECLARING THE PROJECT OPEN.

THANK YOU.

Figure 4. 4: Premier Donald Dunstan's speech to open the Angle Park Computing Centre, June 1973.

Table reused under Fair Dealing for Criticism or Review

For schools without access to networked terminals, computers located at the APCC allowed students from around the state to complete their 'programming' by using punch cards (requiring an expensive hole punch) or less expensive optical mark-sense cards filled out with pencil (Figure 4.2). Upon receipt of a classroom's batch of cards, usually delivered via courier, the APCC staff would process them through their computer. Printouts of results would be returned to schools during the following days, including any errors in the output. In comparison, the similar system in Victorian schools, the Monash (University) Educational Computer System (MONECS) was not introduced until later than at the APCC (1974) (McDougall and McCrae, 2000; Tatnall and Davey, 2004; Davey and Parker, 2010). Whether the work of the APCC influenced development of MONECS is unknown. By 1983 the APCC was in possession of an IBM 4331 used for card processing, with twenty BBC and ten Apple II microcomputers (Wallace, 1989) used by visiting students. David Newall, who as a student experienced programs at the APCC, recalls:

> Ian [Appleton] ran an installation which, in 1976 comprised two IBM Model 1130 computers, and was later upgraded to an IBM Model 370. Every secondary school, and thus every secondary school student, in South Australia had access to those computers, with most people sending [optical mark-sense] cards (similar to punch cards, but you marked them with a dark black pencil) by courier and receiving their printouts about a week later. Slow, but very effective, and the delay meant that you thought about what you were doing! (Newall, no date)

The particular examples of APCC programming cards shown below originate from a high school student from the mid-1980s. The top card includes pre-cut holes to initialise the processing on the APCC's minicomputers, with the second card an example of the mark-sense cards that students filled in to create printed banners, calendars, or charts (Seifort, no date). Even with the increased prevalence of microcomputers in schools by the mid-1980s, logic and processing taught using a supposedly outmoded method of programming continued to be supported by the APCC during this time. It appears that APCC maintained a range of technologies and resources to allow for different approaches to teaching the fundamentals of computing and logic. Although the question of why APCC continued to run multiple systems and approaches wasn't definitively answered during my research, it is reasonable to suggest that multiple levels of support and technology were maintained until access to computers in South Australian schools was more equitable and ubiquitous.

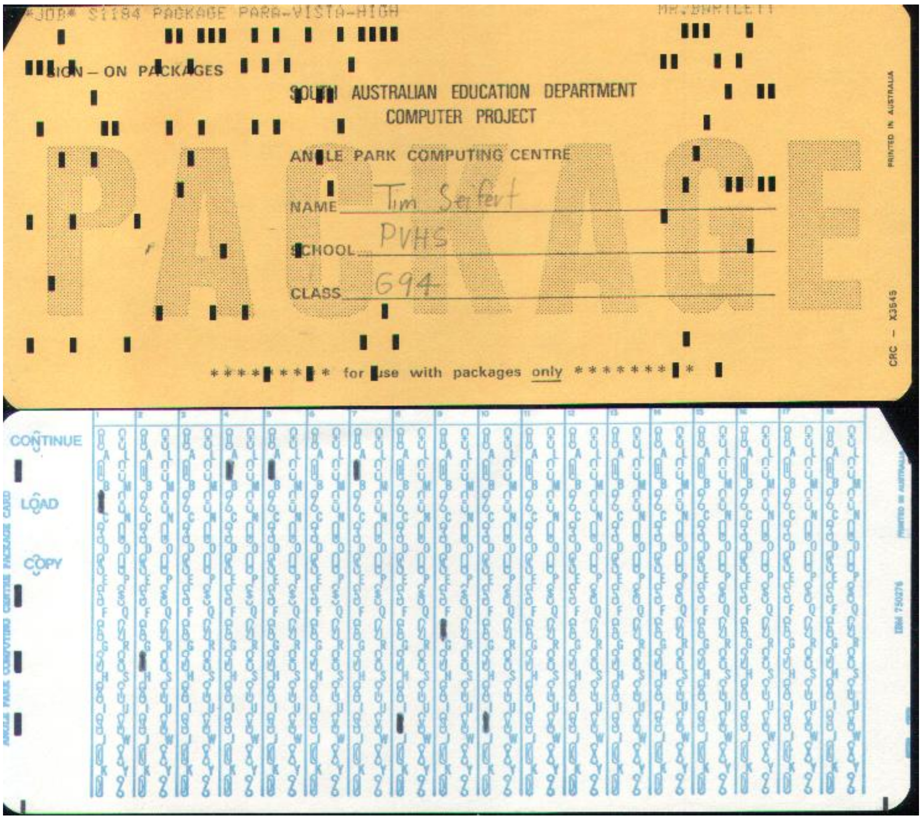


Figure 4. 5: Optical mark-cards used by the APCC to process student work. (Seifort, no date)

The microcomputers installed at the APCC were designed to provide students and teachers from other schools with hands-on experiences of computers during organised visits to the APCC (Education Department of South Australia, 1973).

"Dimension: A Magazine about Computing in Secondary Schools" explained how APL, an early programming language, was taught to students. The centre also ran in-service conferences and summer school programs for teachers and students from across the state (*Dimension: A magazine about computing in secondary schools, 1969-1972.*, 1972; Appleton, 2001a). The single copy of Dimension I obtained ('Dimension Volume 9', 1971) was a short twenty-page newsletter about the activities and accomplishments occurring at the APCC. A number of features were noted about the educational computing landscape at the APPC and what they were working on at the time. Regarding installation of the IBM 1130 minicomputer, Appleton noted in this issue of Dimension that there were problems including "headaches, concerned with both the hardware and software". This is not surprising given the complexity and relative immaturity of the technology at the time. Justifying the installation of this computer, the same piece in Dimension indicated there was a desire and hope that "the ever-increasing variety and complexity of available software will not cause too much confusion for the casual user of the 1130 system." It is unsurprising that availability of software was sparse, but there was an expectation that the number of useful titles would increase over time. The newsletter also includes information about the programming cards for schools, including how students should fill them in by making a "thick, black mark with either a B or 2B pencil." Beyond further discussions of procurement of resources, half-day visiting sessions open to teachers and students from across the state to introduce and familiarise them with computers, and information about inservice conferences during 1972, the Dimension newsletter is very technical and aimed at those familiar with hardware and programming. This is not unexpected as minicomputers during the 1970s required more technical operations and understanding of programming than the microcomputers that followed in the late 1970s and beyond.

Follow up publications from APCC in the throughout the 1970s and 1980s demonstrate an evolution of focus from hardware and how to use to, through coding and programming, to using computers in and educational context. Whilst there was still technical information included, they were aimed at teachers and students with a range of computer experience and classroom use, in addition to professional development activities. The Handbook of Computing Resources (Figure 4.3) (Angle Park Computing Centre, 1975) is an updated booklet of an early APPC publication. It is a well-structured and clearly laid out teaching and learning resource of over fifty pages that introduces the objectives of educational computing and how computers and algorithms can be used to solve problems. Additionally, it outlines the functions and services of the APCC including their summer school, how programming can be used in mathematics, using the various programming packages (small programs to teach the basics of computing) that the APCC provided, and a variety of programming commands and techniques which are the bulk of the booklet's content. This booklet feels relatively mature and comprehensive for the time of publication. Although the packages that students could use (via students completing programming cards off-site) perform relatively simple operations that could be completed quickly on a contemporary calculator, microcomputer, or touch-screen device, they offered a variety of uses which benefitted from the processing power of computers. A few of the examples of the packages offered include programs for high school physics (projectile motion, particle interaction, diffraction) and mathematics (dice rolling, quadratic functions, plotting of graphs). The programming content includes instructions that clearly and logically demonstrate the concepts of inputting data, processing according to specific instructions, and outputting data in text or visual form. Whilst the information contained within the booklet is technical and detailed, there was an obvious shift, compared to the hardware and related technical content in Dimension, toward using computers (even if off-site) to solve problems to real-world or academic problems. The aforementioned packages were useful in solving subject-specific problems but were not fully-fledged educational programs such as those that will be explored later in this chapter and in Chapter 5.

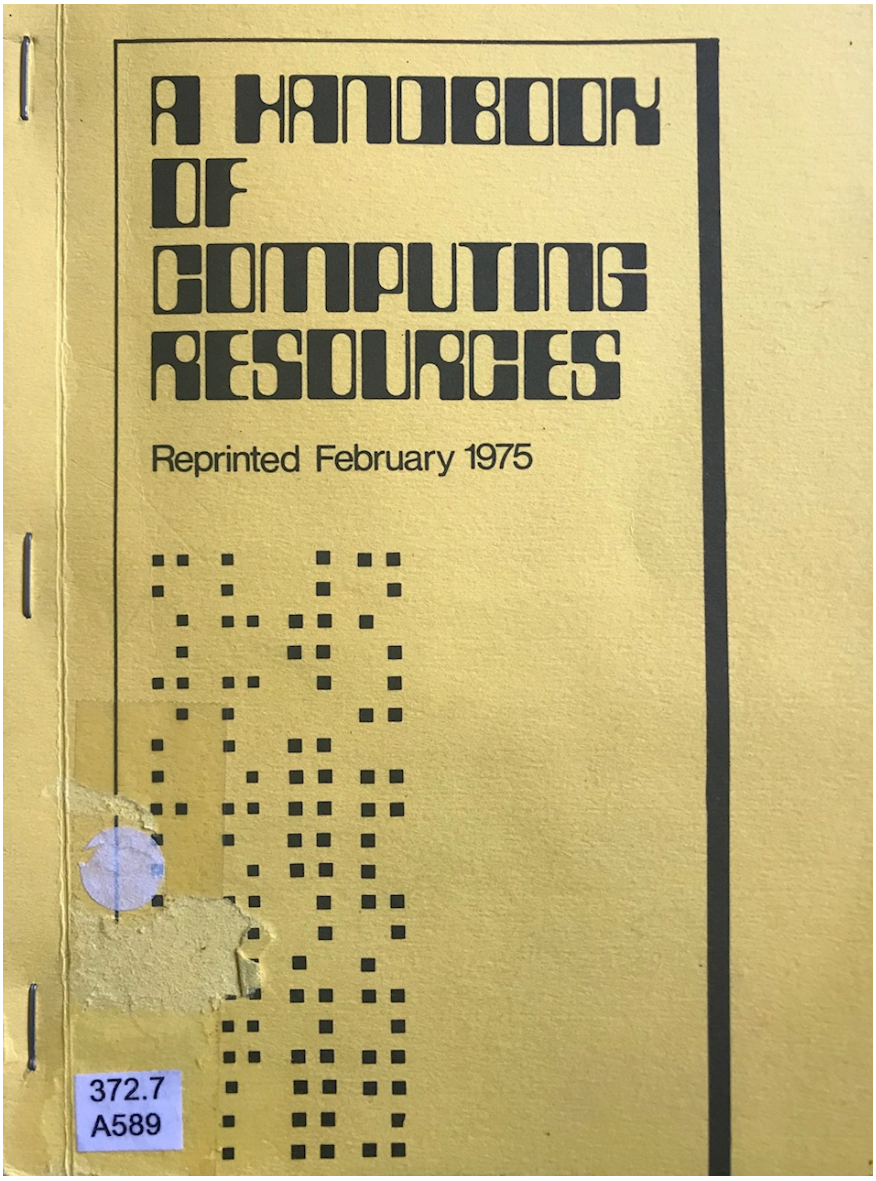


Figure 4. 3: A Handbook of Computing Resources, 1975.

As previously mentioned, the APCC ran annual summer schools for students, teachers, and other parties interested in computing. I obtained the proceedings for the

Eighth Summer School in Computing: Learning and Enjoying ('Eighth Summer School in Computing: Learning and Enjoying', 1981) for consideration of the types of activities the APCC undertook at these conferences. Newall (2007) recalls:

> Once a year, during the summer school holidays, a conference cum workshop (the "summer school in computing") was held at Angle Park. Students (and teachers) would come from all over the world, and many eminent computing practitioners would come and lecture, present seminars, and teach! There was nothing unusual about mixing with students from far away places like Canada or Sweden, nor about talking shop to senior researchers from IBM's Thomas Watson Research Labs. There was a reason why the program at Angle Park was world-leading!

Framing this particular conference in 1981 was a period when the educational computing landscape in South Australia was maturing from off or-site student learning using the APCC's resources toward local school-based microcomputing and the APCC producing software for school use. The content of the conference was broad and focused on real-world applications and uses for computers. The conference's opening address framed how computers could potentially be used to solve real-world problems including those relating to population, food, resources, and pollution. There was content varyingly relating to programming pension payments, computing at the Institute of Medical and Veterinary Science in South Australia, overpayment of employees due to programming and computational errors and how this was perceived by the news media, computing in society and as a career in the 1980s, and some programming and computer technology content. The conference content (contained within a proceedings booklet, Figure 4.4) appears to have been curated to be inspirational and aspirational, with a focus on real-world applications, benefits, and potential issues related to computing beyond schools being considered. These conferences and other activities introducing computing technology are known to have inspired a number of South Australian computing pioneers including Simon Hackett (founder of the Internet Service Provider Internode) and Marty Gauvin (Tier 5, Hostworks) (England, 2015). This conference is an demonstration of the APCC's direction progressing from using computers for hardware and programming focused activities to those which included harnessing computers for potential real-world problems and inspiration to the next generation of computer users. The next chapter section explores how various software titles subsequently created by APCC staff was used in schools throughout the 1980s and 1990s to develop learning skills.

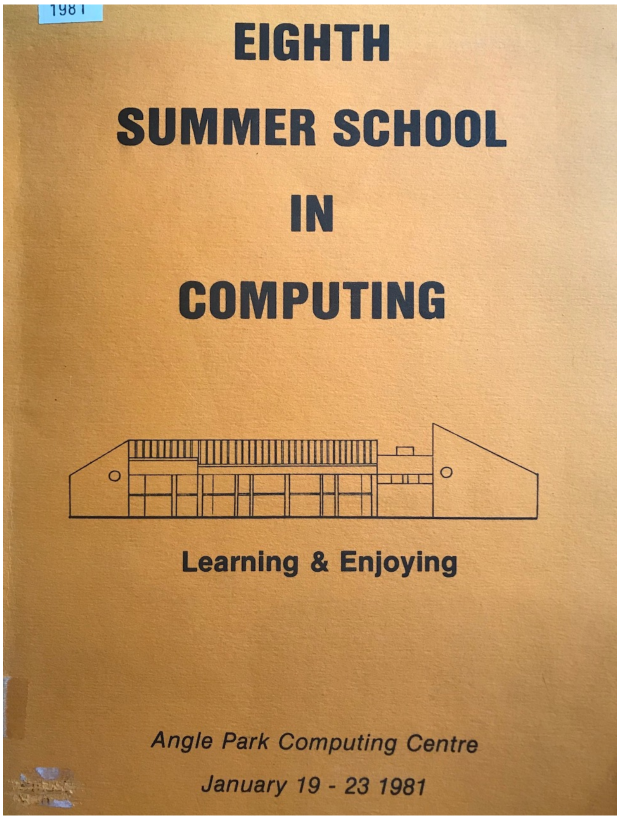


Figure 4. 4: Eighth Summer School in Computing, Angle Park Computing Centre 1981.

4.3.2 Angle Park during the 1980s: accounts of two teacher programmers

The relevance and impact of the APCC developed throughout the 1970s as computer use and awareness in South Australian schools increased. During the 1980s the APCC continued to provide training and support for schools, students, and teachers, but it also evolved into becoming a fully-fledged publisher of software under the Satchel Software label. This section focuses on two educators and programmers who worked at the APCC during the 1980s, Dean Rosenhain and Dean Hodgson. They joined the APCC having previously taught in South Australian public schools. Like many of their APCC colleagues, both were early adopters of microcomputers and incorporated computers into their classroom practice early, programming several educational game and productivity titles.

Prior to working at APCC, Dean Rosenhain taught at Brighton Secondary School in South Australia. His informal forays into computing during the late 1970s and early 1980s led to his transfer to Angle Park. With a background in science and some limited exposure to programmable calculators, his interest in personal computing was piqued when he learnt about and purchased an Apple II in 1978. As there was no formal computing course at the school, he took the computer to school for several years and ran an informal computing group where “At lunchtime I’d show it to kids and do simple writing of programs.” (Schmerl, 2017b).Rosenhain taught interested students how to program with mark-sense cards provided by the APCC to teachers and students:

> When I was teaching at Brighton in the 70s, school students would send off mark cards to a place called Angle Park Computing Centre. They were optical mark cards…like programming cards by putting

> little marks on with pencil, with detailed instructions… They're like parameters for programs to run, so there'd be a card that you'd fill in to run [for example] a calendar. And you would say what year you wanted the calendar for and what picture you'd want on your calendar… [The cards would] be couriered over to Angle Park Computing Centre, stacked into their IBM mainframe [and] run through that. A stack of paper would come back at the end of the day or the next day, and every student would end up with a calendar that got produced. We thought we were pretty clever, [but] it wasn't very interactive. Nine times out of ten the kids didn't mark the cards properly…they'd get a big error sign. (Rosenhain, June 2017)

The process described by Rosenhain underscores the challenges schools faced prior to the wider introduction of microcomputers. Delays in getting feedback on students' work, the lack of real-time interaction, and difficulty in completing conceptually challenging work accurately were common at the time. Rosenhain highlights the APCC's principle that "…all of the schools [in South Australia] had access to send these cards". Although access was open to schools state-wide, how many schools took up this opportunity remains unclear.

Records show that by 1982 approximately sixteen thousand students and six hundred classroom teachers had used the support materials produced by the APCC (Anderson, 1984). These numbers indicate the APCC had a strong impact on the teaching of computing within the state even before many schools had their own computers. The teacher support materials produced by the APCC were broad and covered five main areas: (1) The components of a computing system, (2) the history of computers, (3) using a computer, (4) everyday use of computers in society, and (5) the social implications of the widespread use of computers. As seen from these topics, the APCC's support materials were predominantly concerned with teaching about the computer as 'an object' rather than using it as a tool to supplement existing classroom practice, pedagogy, or to drive change in teaching and learning. The focus was on learning about 'the computer' during the 1970s and early 1980s, rather than how computers could be used. This exemplifies how the early work of APCC sought to introduce and familiarise teachers with the potential of computers and computing. Anderson notes that some other materials were already becoming dated and were being revised to accommodate for the rapid changes in computing during the early 1980s. By the mid 1980s the APCC's focus shifted to using the computer as an educational tool. Ultimately software development for classroom use became a key aspect of the APCC, as discussed throughout this chapter in section 4.3 and beyond.

Sharing ideas and experience between computer users via user groups was a key method of communicating experiences with new technologies., and Rosenhain's involvement led him to start working for the APCC in 1982. He was a foundation member of the first Apple user group in South Australia, which held its meetings in Random Access, an Adelaide computing shop. Rosenhain is typical of many of the attendees at these early meetings who were "hobbyists or teachers", which is unsurprising given the first affordable microcomputers had only been on the market for five years or less in 1981. Through this user group, Rosenhain met Ralph Leonard who has a long history of involvement with the APCC and educational computing in South Australia (EdTechSA, 2013). Rosenhain remembers how his position at the APCC arose:

> Ralph Leonard was working at APCC at the time, so he was one of the people that helped students about how to fill in these [mark sense cards]…he realised I was working at Brighton High, taking my computer to school and writing software for kids in the school… One day I got a message from a bloke called Peter Sandery, who was the head of Angle Park Computing Centre…Peter said, "Put in an application for transfer to APCC", which was classified as a school…I didn't know there was a vacancy there. He says, "There will be once you put in an application."

Rosenhain says one reason his position was created was to expose students to new computing technologies and assist with learning basic operational skills. And increased number of microcomputers at the APCC allowed hands on use by students. Additionally, proximity to the APCC mainframe minicomputer allowed students to complete programming tasks and receive results more efficiently than the protracted process of completing mark-sense cards offsite and submitting them for processing. Rosenhain explains:

> … we didn't have our own students, they were trucked in… to show them what a [microcomputer] looked like…at that stage most schools wouldn't have access to a computer… They'd have a room full of students… kids would use the optical mark cards…they'd go straight into the computer and they'd get their results back within twenty minutes, so it was more interactive…at the front of the room we would have a couple of Apple IIs…where we'd run these demos…we'd do sound, we'd do graphics, all very basic sort of things [compared to now].

Rosenhain highlights the benefits of the immediacy that the APCC provided to visiting students. Several classes of visiting students per day would attend the centre. The allowd provision of intensive courses conducted by confident and experienced teachers to a high volume of students from around the state. Rosenhain's job was balanced between teaching and programming, with the coding and software design aspect of his role being explored futher in section 4.3.

Typifying the self-taught and hobbyist nature of computer users and programmers at the time, Dean Hodgson, another teacher hired by APCC (in 1984) began his programming life at school when he was introduced to computers by his mathematics teacher (Schmerl, 2017a):

> I go back a long way, 1969…I was in high school in the United States and…it was a math teacher who had learned how to write programs…And he thought he'd try to teach some of the students. So, we had this little class of about a half a dozen kids that I managed to get into and I was no math whiz at all, but that's what started it off. And I learned how to do some simple programs and I ended up doing some not so simple ones. When I hit University a couple of years later I managed to get my own time using their big IBM system and their programming language…So I taught myself that. (Hodgson, 2017)

Upon arrival in Australia and teaching in Port Pirie, located in rural South Australia, in 1978, Hodgson became aware of a Hewlett Packard desktop computer (model unknown) that was shared amongst three schools in the town. He implies the computer served limited utility for students, but in his own time out of school hours he wrote planetary simulation software for the computer. Procuring his own personal Tandy Model 1 computer, he would take it to school to use with the students as Rosenhain had also done. Eventually a Tandy was purchased for the school, but the dearth of available software necessitated teacher-coded software if the computers were to be of any functional use in the classroom.

> I got my hands on the Tandy Model 1, and then just started doing stuff initially for myself, but being a teacher, I brought it into school and started doing things with the kids, and "Oh we need programs," because there wasn't any. I started writing my own programs…Eventually I managed to get a Tandy colour computer into the school and again no software, so I wrote tons of stuff for that school that the children ended up using.

Hodgson focused on using computers and programming as an educational and classroom tool. Rather than make the focus of teaching on the computer itself, he developed software to engage primary students in subjects such as Mathematics. Additionally, Hodgson designed games in collaboration with students as part of his classroom practice. Some of these programs were submitted as code listings to be published in computer magazines. Hodgson explains how he ensured that the work he produced was fun and engaging as well as educational:

> It was all trial and error…[asking myself] can I produce something that is going to have an effect on this child, a positive effect on their attitude, as well as [asking] can they learn something out of it?...part of it was [to] design the software to present itself in a way that wasn't dry, that didn't put the kids off, so it was more encouraging and they felt they were getting somewhere...When I was teaching a class I had

> the class design a computer game, the whole thing. We planned out what it was, the students drew up the graphics [for] it, I wrote the program, we published it in a computer magazine. It was Space Meanies. You know just shoot the thing coming down kind of stuff.

In 1982 Hodgson was convinced by magazine editors to submit some of his programs to the Tandy computer company with the hope of convincing them to publish his work. This led to three titles being published by Tandy, followed by six titles for the Sega SC-3000 in 1984 (published by John Sands). Because Hodgson had created these various educational titles and was using computers in the classroom, his work gained the attention of the Education Department of South Australia. In addition to programming on the various computers he had already worked with, Hodgson was also looking at using the Commodore 64 computer for future development work. The comparatively wide use of the Commodore 64 in South Australian Schools compared to most other states, was the catalyst for him joining the APCC in 1984. The demand for teacher training in educational computing and the need for appropriate software on these microcomputers were cited by Hodgson as key reasons for his appointment:

> …the Department discovered that I could do this [program educational games]…I got called up one day by somebody at the Angle Park Computing Centre to say, "Hey we need some programs on a Commodore 64, can you work on that machine?" [I replied] "Yeah I'm having a look at it now, I can do that." And so I first took some things they had done on Apple II and then versioned them onto the Commodore…that just started like a snowball. It just grew from there and about September of '84 I was transferred…eventually to Angle Park. My main job was to work with teachers out in schools

> teaching them how to use these computers with kids. But the second thing was that I was also writing and creating and producing all this raft of classroom programs and that ended up eventually being a team effort… There were very few programs [in] the day so there was [demand for] a teacher like myself who could write software [and] who worked at Angle Park.

Rosenhain suggests that Peter Sandery, the Angle Park principal in 1982, was already thinking that as the proliferation of computers in schools increased, the functions of the APCC would evolve. The second reason for Rosenhain's employment "was for software development… They had this idea that they wanted to write software that they could give to schools or sell to schools." Software written by staff at the APCC was created with the centre's microcomputers. Rosenhain recalls there was little bureaucratic or government interference in terms of the software being produced:

> We'd do what we wanted to a certain extent…I'd come up with an idea for something that was cool…and I'd write it and then everyone would have a look and say "Yes, we'll give it to schools." It might be done in a week…An idea, it gets turned into a program, and before you know it we're advertising that we've got it and selling it to schools for a couple of dollars.

Hodgson's work with computers prior to his time at the APCC similarly demonstrated a do-it-yourself ethic with a focus on creating effective educational software. His independent approach aligned with the manner of working at the

APCC. Echoing Rosenhain's thoughts on autonomy, Hodgson emphasised the freedom they had when designing and creating software:

> I was asked a few times, "Oh can you do something like this?" But no most of it's my own initiative or somebody else's idea. That was the thing about the way Angle Park was run, that we were pretty much, we had a free reign to pretty much do what we thought was good for [students and teachers] and we did.

During a time when there was a scarcity of appropriate software for schools, and computers were difficult to use or even intimidating, the APCC programmers were entrusted to use both their educational backgrounds and continued programming experience to create software that they thought would be beneficial in the classroom.

Information about the APCC's software and operations spread in numerous ways, including via user groups and by word of mouth amongst teachers. Additionally, the APPC produced a newsletter called *The Computer in the General Curriculum* which was distributed throughout the 1980s to South Australian schools. The title of the newsletter is telling; including the phrase 'general curriculum' is indicative of where Sandery's vision aligned with the potential of computers as an educational tool, and how the software produced by APCC staff could support this vision. The copy of this periodical I obtained is the seventy sixth issue, has a publish date of September / October 1985, and is nearly fifty pages long. Although it is not known how long this newletter was published, there was obviously a demand for a publication aimed for teachers to support the use of computers in education in South Australia. The publication included information about the APCC's software as well reviews, opinion pieces, and other articles about computer use in schools. Of note are pieces discussing the merits of the status of year 11 and 12 computing subjects, and how to balance the inclusion of programming, spreadsheets, and databases into the curriculum, as well as hardware and software options to support student learning. Although the specific details relating to hardware and software are dated, over thirty years later these are still considerations when discussing the development of computing in schools, although much of this happens online rather than via physical newsletters or formalised user groups. Other articles of prominence, some written by Rosenhain and Hodgson in addition to other APCC staff, include considerations of computers and gender, computer implementation reports, and a summary of the state of computing in South Australian schools (surveyed in mid-1985) where it is noted that there was just over two and a half thousand computers installed in South Australian schools. (Figure 4.5). Notably, over eighty percent of these were networked, showing that schools valued the utility of shared resources such as disk drives and printers. The various software reviews, including titles by Fitzroy Community School written by Philip O'Carroll (interviewed in Chapter 5), detail use cases and offer recommendations. Again, much of these details as they related to today's educational computing landscape are often sourced online or via other literature, but the breadth and detail included in this single newsletter is indicative of the need for a broad educational computing resource of locally related information to be disseminated during the 1980s. Key interrelating points are the variety of computing platforms discussed within, the specificity of the South Australian context, and the varied use cases and types of software and applications of computing across the curriculum. A regularly published periodical dense with a variety of information would have been invaluable to many teachers during the emergence of computing in schools

THE STATE OF COMPUTING IN SCHOOLS

TOTAL NETWORKS 10/07/85

MACHINE	TOTAL NETWORKS	TOTAL MACHINES	HIGH SCHOOLS	PRIMARY SCHOOLS	TOTAL NO. OF SCHOOLS
BBC	75	882	73 876 MACHINES 74 N/WORKS	1 6 Machines	74
APPLE	5	64	5 64 MACHINES 5 N/WORKS	0	5
COMMODORE	135	994	48 493 MACHINES 58 N/WORKS	72 501 MACHINES 77 N/WORKS	120
VIC-20	12	99	6 72 MACHINES 8 N/WORKS	3 27 MACHINES IN 4 N/WORKS	9
OTHERS	6	82	5 66 MACHINES 5 N/WORKS	1 16 MACHINES	6
TOTALS	233	2121	137	77	214
			1571 MACHINES 150 N/WORKS	550 MACHINES 83 N/WORKS	

TOTAL STAND-ALONE SYSTEMS 10/07/85

MACHINE	TOTAL MACHINES	HIGH SCHOOLS	PRIMARY SCHOOLS	TOTAL SCHOOLS
BBC	58	23 44 MACHINES	10 14 MACHINES	33
APPLE	337	135 258 MACHINES	61 79 MACHINES	196
COMMODORE	275	30 40 MACHINES	158 235 MACHINES	188
VIC-20	33	6 10 MACHINES	18 23 MACHINES	24
OTHERS	95	33 72 MACHINES	16 23 MACHINES	49
TOTALS	798	227	263	490
		424 MACHINES	374 MACHINES	

30

Figure 4. 5: The state of computing in South Australian schools in 1985.

Despite Hodgson joining the APCC several years after Rosenhain, there was still a focus on the training of teachers to best use computers in the classroom. The outcome and curriculum-driven nature of their work, rather than an emphasis on teaching 'the computer' as an object to learn about, as noted by Hodgson:

> …most of the people at the centre had been teachers, we [came] out of schools…just about all of us were school people and so we had that headset. And our, the whole, the computing centre was changed from the big IBM to saying, "Okay well, we have these other computers, we have Apple, we have the BBC and now we're supposedly getting Commodores, how do we help teachers work with these?" And that's what happened…I was part of an effort to help teachers. So our focus was not students it was teachers, to help teachers teach with computers.

The continued importance of the APCC acting in a support role was also reinforced by Rosenhain. He recounts when schools purchased or received hardware to outfit computer rooms, he would travel to train the teachers and speak to parents and the school community at large to discuss the benefits of the technology in the classroom. The focus of the APCC had already moved beyond teaching about computers themselves:

> Peter Sandery's vision of computers was not so much learning about computers, but using computers as a tool to do other learning…It was a visionary sort of thing at that time. People…thought we were going to learn about computers. And he would say "No you're doing History, but we want you to use a computer to do word processing

> and write up your notes in History efficiently."…That leads onto Satchel Software.

The realisation of using computers as a classroom learning tool was a gradual process as envisaged in premier Dunstan's APCC launch in 1973. Although ubiquity of microcomputers in most South Australian schools would not be realised until the 1990s, the groundwork done by APPC staff with South Australian teachers and students facilitated confidence and familiarity with computers. The shift from teaching about the computer toward classroom use in conjunction with APPC staff creating software for schools led to the formation of a formally branded software publishing arm, namely Satchel Software.

4.3 Satchel Software

The educational publishing concept that would ultimately be branded as Satchel Software was conceived during Rosenhain's first few years at the APCC. He personally chose the Satchel Software brand name based simply on a name that had not previously been trademarked, but had obvious educational connotations. Rosenhain recalls:

> Our first versions of software we were distributing were for Apple II and [instructions and resources] would be printed on a photocopier… someone typed up and put in a ziplock bag with a floppy disk. We'd make those and sell them for a couple of dollars to a school…It was pretty basic…at that stage we then decided we needed some sort of branding for it, some sort of identity. This idea evolved during 1982 and 1983.

Hodgson notes that there was a marked improvement in the quality of software produced by the APCC teachers during his time there. Initially each software title was entirely coded and produced by individuals, but collaboration amongst employees allowed them to share ideas and utilise their various skills to create more professional quality educational software. Steve Walsh, a primary teacher and graphics artist who joined the APCC during the mid-1980s, introduced the concept of releasing software as thematic packages focusing on concepts related to curricula such as children's literature. Walsh also used contemporary technologies to assist in the production of graphics; the then novel KoalaPad[9] was used to draw graphics.

[9] The KoalaPad was a tablet device released in 1983 by Koala Technologies, a United States based company. The KoalaPad allowed input via a stylus or finger. A range of software drawing packages allowed users to create bitmap graphics.

This allowed other programmers to focus on coding and design. As the software improved in quality, so did the associated the manuals, teacher guides and resources provided to support classroom use (Figure 4.6). As the uptake of microcomputers in schools was increasing during the early 1980s, initially software was given to schools by computers or sold for a token price of "$10 or you know something really cheap" according to Rosenhain. As their production of software and resources continued to increase and evolve, the APPC transitioned into a more structured and self-supporting entity. With an emphasis on classroom use and consideration for student activities that took considerable time, resources, and money to produce, Hodgson recalls:

> …by that time [the mid 1980s] we'd produced quite a few titles that were being widely used in this state, and one of the other things that began to happen was in order to increase the budget Angle Park made a decision to become commercial and sell its software. We weren't giving it away, we're going to be selling it…the packaging got better, and the manuals got better…and so it became eventually quite a commercial outfit.

Figure 4. 6: Examples of Satchel’s branding and design.

Although it is unclear exactly when the Satchel branding was launched, some of the earliest examples of works under the Satchel name appeared in 1985. Approximately 95% of secondary schools and 20% of primary schools in South Australia had at least one microcomputer by 1983 (Fleer and Stout, 1991), with the number of computers in schools rapidly rising in the ensuing years. Consequently, the APCC's role evolved from a centralised computer training facility to one of teacher support and software production. Satchel's publications were not limited to software and related documentation. They produced supporting materials for teachers in schools. Two of the earliest Satchel publications, *Logo in the early years: 3 case studies* (Satchel Software, 1985) and *Databases in the Classroom* (Satchel Software, 1986) (Figure 4.6), were booklets used to assist, develop, and inspire teachers when using computers in the classroom. Some Satchel-branded training packages included video-based materials for software were also produced by the APCC (Spreadsheet Training Package, n.d). The majority of materials produced under the Satchel branding from 1985 onwards were, however, software and related teacher and student documentation, which will be explored in the next section of this chapter.

There is notable evolution in the many software titles produced and published under the Satchel Software brand throughout the 1980s and 1990s. Early software often drew upon obvious influences from existing popular arcade, console, and computer games. The graphics and sounds were simple, options were limited, and educational benefits and relatedness to existing curriculum could be considered tenuous. This early software was generally conceptualised and created by individuals. Over time there was increasing collaboration between team members, target platforms had incremental improvements in computational power, and as the skillsets of creators developed, the software consequently improved in quality and diversity. Similarly, the quality and utility of the documentation and support materials accompanying the software also improved. The various facets of software evolution are noted in the selection of titles presented in this section.

4.3.1 Mathbooster (1985)

> It was like "Space Invaders" but there were mathematical problems that came up…It was like a game then maths…of course there's zillions of things that do that now. (Rosenhain, 2017)

Mathbooster (not to be confused with the more well-known *Math Blaster!* series of educational titles[10]) is an arcade style mathematics game, published in 1985 for the Commodore 64, followed by ports for Amstrad, PC, and Amiga. The game is an iterative take on earlier educational titles programmed by Hodgson on the Tandy computer. More rudimentary versions were designed by Hodgson during the late 1970s, with this incarnation including some options and to increase classroom utility, in addition to attractive colours and sounds. *Mathbooster* was designed to be engaging for students due to its arcade style heritage, but it could be considered to have limited scope as it was only related to a single school subject. As a single player game released in an era when computer access and time was likely limited, the utility of the game in a school setting may appear questionable. Regarding *Mathbooster* and other similarly 'simplistic' titles, Hodgson explains "there were programs that…[in]

[10] The *Math Blaster!* series was produced for a range of 8 and 16-bit computrers from 1983 until 1990. It was published by Davidson & Associates in the United States. Although sharing some similar arcade mechanics to *Mathbooster*, they were designed and developed independently.

their own right they just did what they did" and suggests it was widely used. This title provides a good launching point for the examination of Satchel Software; despite the game's apparent simplicity there are options available that allow customisation and tailoring to class and student needs.

Mathbooster is indicative of exploratory forays into game design, and indeed the creation of educational software. Over several years Hodgson made minor iterative improvements to his 1970s original. In addition, another APCC programmer (unknown) created a similar game to his Tandy original on the Apple II, although whether it is a direct clone or a coincidental case of convergent evolution in game design is unclear:

> That's an interesting history because I'd done this one called *Math Invaders* on the Tandy and six months later *Mathvader* turns up on an Apple II [published by Angle Park Computing Centre][11], which is almost a copy of what I'd done, and then they [Angle Park] said, "Could you do the Commodore version?" So I ended up doing that and we ended up re-writing it and calling it *Mathbooster*…

There are obvious similarities to existing shoot-em-up games such as *Space Invaders*, in both the shoot-em-up mechanic and also arcade-like challenge where the game ultimately overwhelms the player. The focus on mathematical problem solving tied in with arcade game qualities proved engaging for the students. By the time the Commodore 64 version was released, the game included options for loading more

---

[11] *Mathvader*, published by Angle Park Computing Centre., Adelaide: Education Department of South Australia in 1983.

mathematical problems as well as allowing the teacher (or students) to manually customise the problems (Figure 3.7).

> …the idea was that there were these satellites falling. And they start out kind of slow and the student could move this rocket back and forth and then put the answer into a math problem because you see these little math problems underneath. The teacher could set it up with different kinds of math problems and different levels involved. And so the student would launch a rocket to push the satellite back up and then the next wave would be a little faster, a little faster until finally they…couldn't cope with the speed.

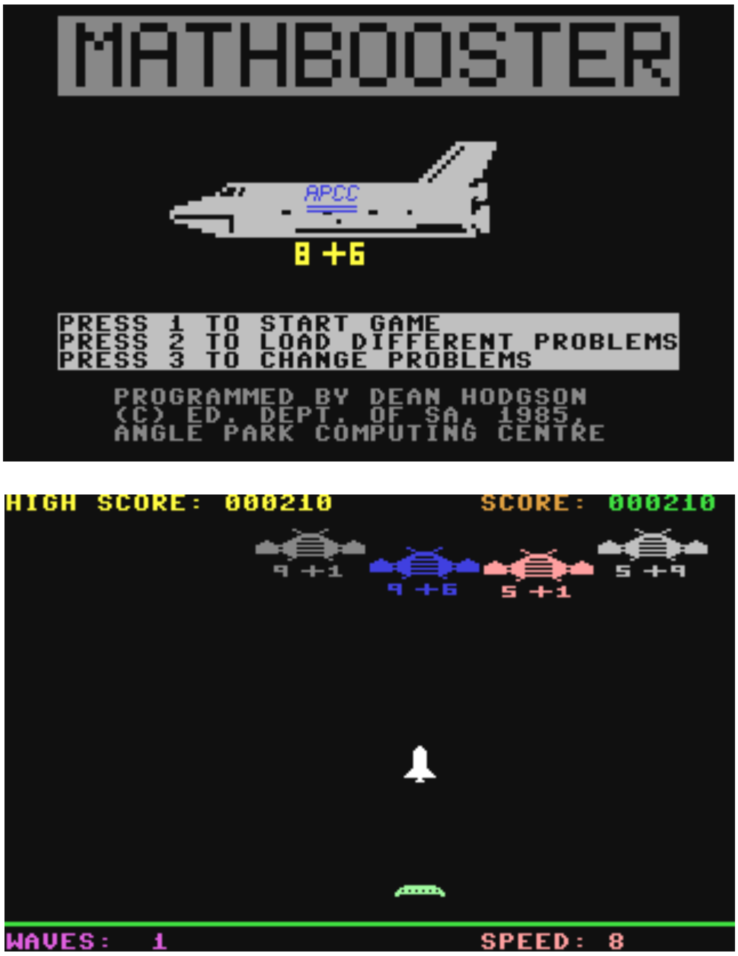


Figure 4. 7: *Mathbooster* options and game screens.

There is evidence suggesting that reception of the title was positive. Australian newsletter *The Amstrad User* (n.d.) praised the game's ability to be "entirely customised to the needs and weaknesses of [the] child" using the four mathematical operators, as well as making otherwise rote learning addictive and providing a sense of achievement. *Amazing Computer Magazine* (1992) noted that *Free Spirit Software* licensed the game (and other Satchel Software titles) for distribution in the United States. This is notable in that the game that was over five years old by this date but was still considered appropriately engaging for school use. *Compute! Magazine 154* (1993) reviewed the game suggesting that they were "impressed by the solid programming that provides the actual computer game yet allows you to create and modify within the program to make it fit your needs. It's powerful, flexible, and easy to operate." Hodgson notes that when students played the title, the gameplay was considered exciting and engaging. Based on his observations of students' reactions when he was testing the game "I could tell by the kid's reactions that it would work for them."

The relatively long commercial life-span of *Mathbooster* with ports across numerous 8 and 16-bit computer platforms provided Satchel with opportunities to experiment with techniques to improve the ease of porting titles. Hodgson suggests the work initiated with *Mathbooster* "started it all", in regard to the growth of Satchel as a publisher. Additionally, *Mathbooster* was the catalyst for Satchel's innovative software and hardware experiments that streamlined the porting process from platform to platform. These pioneering works contributed to Satchel's prolific output of titles. Rather than create completely independent versions of titles coded from scratch, which would have increased time and resources required for the process, Hodgson expands upon the techniques used:

> I worked out ways of taking source code out of a Commodore and literally running it through a serial port cable into the Amstrad, and the graphic as well and I wrote programs to convert all that stuff across and then we would rewrite it. So we would develop a package on both platforms at the same time.

Similarly, when they later recreated the game for PC (DOS) based on the Commodore Amiga version, a combined hardware and software solution streamlined the process they undertook:

> PC version [and] the Amiga version looked exactly the same because we actually took all the graphics out of the Amiga and cabled them across into a PC and then I wrote the program that would drive all this sort of thing.

Although the graphics and sounds of the later ports were not drastically improved over the earlier versions, the core gameplay, classroom utility, and educational values carried through. Hodgson also notes that “the documentation was pretty universal to [multiple versions] of them”. These processes go some way toward explaining why neither Hodgson or Rosenhain expressed particular concerns about working with numerous target hardware platforms throughout Satchel’s commercial lifespan.

Mathbooster’s combination of arcade mechanics and drill-and-practice questions was a compelling mix. Beyond the initial novelty of having computers within the school environment, the use of games as teaching tools was clearly recognised by teachers and programmers, especially given the options for customisable experiences that *Mathbooster* provided. When queried on the appropriateness or if there was any negativity associated with arcade-like games within the school space, Hodgson indicates that student engagement was prioritised:

> I think teachers would, most of our programs were pitched at primary schools, and high schools they were more like simulations and databases… I don't remember…anyone saying, "Oh no we shouldn't be using these games." It was all a case of, "Oh no the kids are turned on by these games, we'll take advantage of it."

Mathbooster presents an interesting case of how a seemingly superficial modification of a game in the *Space Invaders* genre was successfully sold and used in schools across Australia and the United States. The game spanned several hardware generations and, despite its comparative simplicity to other titles released in the late 1980s and early 1990s, it both succeeded in being useful as a mathematics learning tool and as a testing ground for future Satchel Software work.

4.3.2 Granny's Garden (1987)

Granny's Garden is an open-ended graphical adventure game that is both relatively well-known and fondly remembered by computer users across the globe, and mentioned variously in a numerous online forums and blogs (*Fond memories*, 2011; Oddbloke, 2012; VGJunk, 2018). This title was not conceived of by Satchel Software. It was originally developed for the BBC Microcomputer in 1983 by Mike Matson and published by his company 4Mation Software (*4mation Educational Resources Ltd*, 2017). Matson developed the game while he was a Council Advisory teacher in the county of Devonshire, United Kingdom, where his role was to support and develop computer use in the district schools (Green, 2010). He found existing software for teachers and students did not motivate or excite the target audiences. Satchel Software obtained the license to port *Granny's Garden* and other 4Mation games to the Commodore 64, Amstrad CPC, and DOS platforms (Figure 4.8 show's Satchel's attributions on the title's instructions). Hodgson recalls:

> There was a program called *Granny's Garden* that had been developed by 4Mation in England on a BBC computer. We got the licence to reproduce that on the Commodore, the Amstrad, the PC. I did all of those. The graphics were done by others, but I did the coding…and then we put our little theme pack together to go with it…to our schools… [The] PC one we did around 1991, and see it's got [a] book with the activity pack and…one about the program.

Hodgson also discussed the parallels between 4Mation's origins and his own introduction to educational software:

> Matson came out several times, he was quite a character. He'd been a teacher himself and that's, you know it's almost the same, similar parallel story to my background…they got these BBC computers in the schools over there but there wasn't any software. Taught himself how to do it, it sort of got widespread, and he formed a company and went from there.

According to Matson, over twenty thousand copies of *Granny's Garden* were sold with it being "a huge success in Australia and New Zealand." (Green, 2010). There are a number of Australian connections to *Granny's Garden* and 4Mation, with Jacaranda Software acting as distributors for 4Mation's original software and the

Australian company Dataworks porting the game to Apple II and Macintosh. 4Mation software is still actively selling and supporting modern versions of the games on Windows, Macintosh, and iOS.

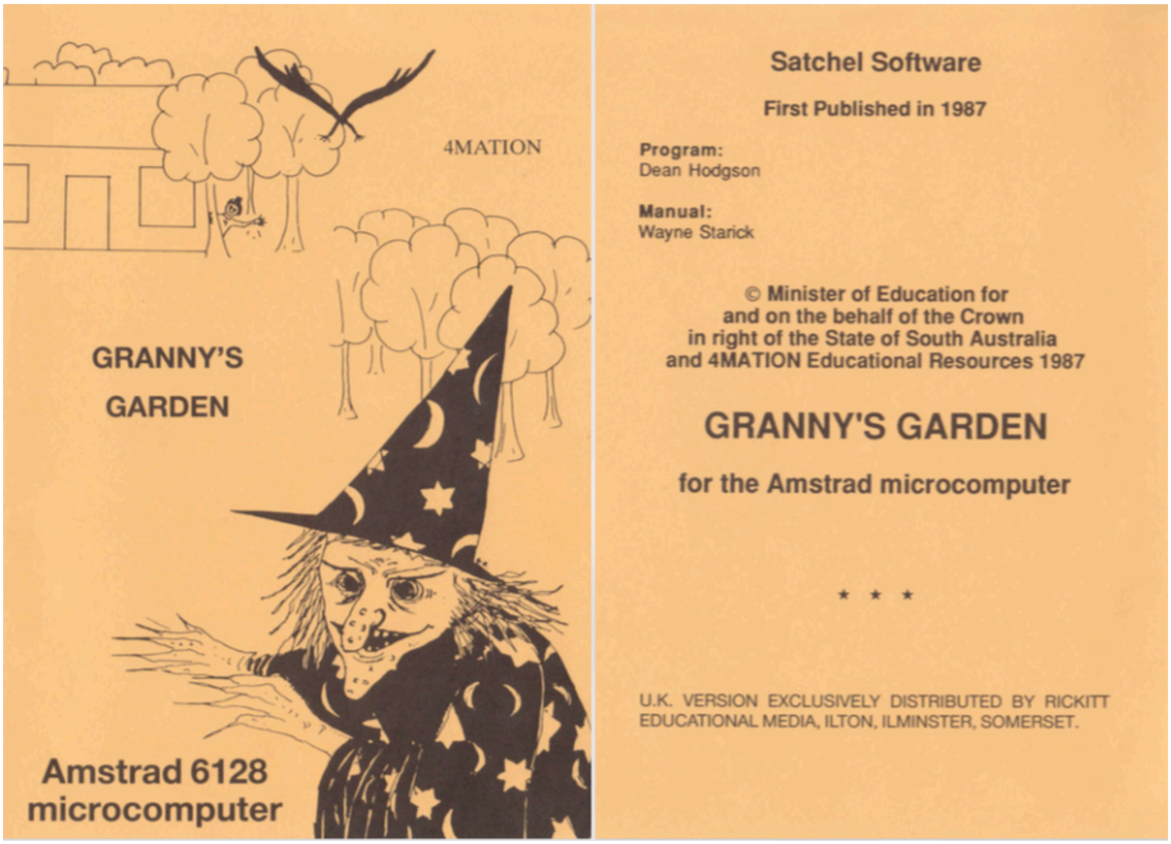


Figure 4. 8: Granny's Garden instructions - Amstrad CPC128, programmed by Dean Hodgson, published by Satchel Software.

The background story of how Satchel Software was tasked with porting *Granny's Garden* (1987) and other similar 4Mation titles (such as *Dragon World* (1989)) despite being located in another country is a little unclear. Neither Hodgson nor Rosenhain recalled the exact details, although Hodgson suggests Wayne Starick, a manager from Satchel Software was likely involved in the process of courting Matson:

> [Starick] had certain fascinations and I got a suspicion he was the one that got a hold of Mike [Matson]. Certainly, Mike was with him when he came out here a few times and visited. I think all we did is we were just given the source code to the program and, well there's the original (and I know Wayne did the original graphics that we first started with) and I just worked from that. There wasn't any talking to the company [4Mation] or anything… And we had a graphics artist by then, named John Gordon, who did the graphics on these programs... They were just saying, "Well here's the original, look at it, [you] work it out."

Although the story behind Satchel being chosen as the coders of these titles was not fully revealed in my interviews, it brings to light some technical aspects of porting existing software. These statements reveal that the simplicity of logic and graphics in 8-bit games afforded a relatively smooth porting process between platforms The software was programmed based on viewing and playing earlier versions, adapting source code (usually BASIC), and the software and hardware solutions discussed previously in the *Mathbooster* section. Because software was ported across multiple platforms and written in BASIC, the titles did not take particularly sophisticated advantages of the various graphical or sound capabilities of each computer. This sped up the process of porting, and also consequently allowed for consistent documentation due to similarity in game logic and functionality. *Granny's Garden* in particular looked and operated similarly across the Commodore 64 and Amstrad CPC ports (Figure 4.9). Hodgson explained that the lack of advanced graphical and audio aspects of the software across Satchel's library was not a negative in terms of educational programs as long as they were seen as beneficial to students. The philosophy of educational utility and ease of use remained the primary consideration to program design throughout their time as a publisher. Satchel eventually created more graphically and sonically advanced educational game titles such as *Jara-Tava – The Isle of Fire*, but software functionality and benefits remained their focus. As Hodgson explains, Satchel's core principles of engaging educational software were always paramount:

> …we had the game background, but we also had this teaching background and we learned very quickly what you do and don't do in writing programs – the ones that work, the ones that don't work. You don't waste somebody's time with flash graphics stuff…because eventually you get tired of that introductory stuff… [There] was a lot of things that we learned that went into the design of the programs. Right down to the colours. You know what colours you pick, why do you pick those colours? It affects the reading ability.

Commodore 64 Version

Amstrad CPC Version

Commodore 64 Version

Amstrad CPC Version

Figure 4. 9: Granny's Garden – Commodore 64 and Amstrad CPC versions.

Hodgson's comments about game design, educational benefit, and utility – even when considering legibility of screen text and graphics – highlight the range of considerations when creating early computer software. Although tasks would be divided amongst staff, when working in small teams, programmers of 8-bit titles had to factor in the integration of all facets of the software they created. A combination of game logic, user interface, graphics, sounds and, in the case of educational software, the benefits, usability and educational value were amongst the broad range of considerations an educational software developer needed to contribute. During the

8-bit era this was feasible, but it became more challenging as the complexity of hardware and the expectations of software increased.

Granny's Garden is an adventure game that was conceived of at a time when there were only limited numbers of computers in schools. When starting a game, the player is "transported" to the Kingdom of the Mountains where - via a series of tasks - six children of the king and queen must be rescued. The gameplay involved students solving puzzles using logic puzzles, some of which involve randomness, trial and error, fetch quests. Matson designed it for use in groups and also where it could be played by students in short bursts. The supporting documentation is clear from the outset that *Granny's Garden* is not a subject specific title, but one designed to stimulate learning, imagination, and subsequently lead into other classroom activities for primary students:

> The package is not designed to teach children anything but to provide a gateway to another world, a world in which children would be stimulated and motivated to ask questions, find answers, discuss issues, keep records and use their imaginations to make that world their own world. The software was designed as a starting point for many other activities in a variety of curriculum areas, indeed hopefully the best things that are done will be done not at the keyboard, but away from the computer altogether.

To support the choice-based and varied learning opportunities provided by the game, it was packaged with an array of suggested classroom activities for students to undertake when not sitting at a computer. Although these activities are neither comprehensive or detailed, being a short paragraph description of ideas at most, the ideas cross over the curriculum areas of arts, craft, science, health, mathematics, and language (Figure 4.10). The potential for cross-curricula use with activities beyond the game itself, as well as assisting the development of problem-solving and other skills, was noted by Hodgson:

> [Matson's] headset was a little different, but quite fits in with modern program[ming for students]… He liked to produce little games for kids where the kids had to figure out what the game was all about, not just…follow something, but [rather have students work out] what do you actually do on this screen… there were quite a few of these [titles that we ported that] are like that. But it's the same way that we looked at things that the program is part of a bigger thing that you're doing with your children. And [teachers are] trying to teach them other things including [developing] some attitudes and values and stuff.

Classroom Activities

This is not intended to be an exhaustive list of absolutely everything that can or should be done with this program in the classroom. It is a collection of ideas and resources, some of which you may reject, some you may alter, some may suggest entirely new activities and some you may even use!

Some come from the original BBC version of Granny's Garden and others were provided by Bev Olive from Gawler East Primary School.

Art/Craft

1. Silhouettes.
2. Painting a picture of 'My favourite adventure in Granny's Garden'.
3. Torn paper mosaics of the witch and other characters from the adventure.
4. Make snails from curled paper.
5. Use crumpled paper to make worm mobiles.
6. Make pompoms from wool on a cardboard former.
7. Make a large woodcutters house and put 'magic' words on the walls. The house could be used as a quiet corner for reading, activities and talking.

11

8. Design signs that could be erected to warn other adventures of the dangers that they might encounter.
9. Draw large maps of the Kingdom of the Mountains, showing the location of all of the scenes from the adventure.
10. Children could cut silhouettes of themselves and stick them on the window. They could cut leaf shapes from coloured paper, decorate them with veins and attach them to string or wool and hang them on the window to 'hide' the silhouettes.
11. Children could trace around their hands, cut out the shape, put their name on it and hang it on a large cut out of a Pompom tree.

Science/Health

1. Set up a pot plant garden - discuss things needed to make the plants grow.
2. Discuss the 'healthiness' of the Baby Dragon foods.
3. List and make mobiles of healthy foods.
4. Use Ah-choo to talk about allergies.
5. The Giant's Garden module refers to a number of animals, this could be used as a starting point to investigate the differences between the animals mentioned.

12

6. The forest location in The Land of Mystery mentions forest fires and litter. This may provide a jumping off point for environmental issues.

Mathematics

1. Coordinates.
2. Following instructions to make patterns using colours.
3. Using grids, experiment with square patterns.
4. Order of numbers.
5. Blue raven square picture.
6. Plotting the adventure map using coordinates.

Language

1. Read the book "Granny's Garden" by S. Harrison and H. Wilks and compare the two gardens.
2. Read and write stories and poems about witches, dragons, giants and/or grannies.
3. The witch in the adventure is a 'bad' witch, what would a 'good' witch do to help the children (cf. The Wizard of Oz).
4. Use storyline sheets to discuss the passwords.

13

5. Write stories describing life in the Kingdom of the Mountains before and after the witch came.
6. Make a large book about witches, dragons, giants and/or grannies.
7. Look at families of words - "itch" words, "wh" words and build word mobiles.
8. Tongue twisters - "How much wood would a Woodcutter cut...".
9. Children could write letters to the rescued children (and send them to someone using (electronic?!) mail).
10. Make up games involving finding hidden children.
11. Write a travellers guide to The Kingdom of the Mountains.
12. The music for the Kingdom of the Mountains is in fact the opening bars of a piece of well known classical music - 'In the Hall of the Mountain King' from Grieg's 'Peer Gynt". Enterprising teachers may be able to use this as a starting point to explore some other classics - 'The Sorcerers Apprentice' and 'Peter and the Wolf', for example.

14

Fig 4. 10: Granny's Garden selection of classroom activities.

In Green's interview with Matson (2010), he echoes these sentiments:

> I thought: I need to show them something more exciting. An actual adventure. 250 hours later, I had *Granny's Garden*. I wanted to create something that was a starting point for children, something that would get them enthusiastic and using their imagination. Because back then you had just one computer *for a whole school*. So a game had to be designed for use in groups, 10-15 minutes each.

Their comments above indicate that both Matson and the titles Satchel were creating had similar goals. Satchel and Matson were aiming to avoid drill and practice activities, but wanting students to learn more intangible concepts like decision making, problem solving, positive behaviours such as cooperation, and skills that transcend the microcomputer and discrete school subjects. Despite great geographic distances, the educational goals of Matson and the programmers at Satchel Software were shared and common.

Matson's intentions and Hodgson's observations regarding the variety of uses for *Granny's Garden* in the classroom aligns with Sandery's vision of using computers as a learning tool rather than as an *object* to be taught about. In Swalwell's upcoming manuscript (n.d.), she reflects on software developed by early hobbyists and argues that personal and professional considerations and approaches were carried over into their software design:

> …programs were created by people who brought their interests and "avocations" to the task "wondering what to do" with something leads to experimentation and to new uses being found…[enabling] them to

> see new opportunities for developing new programs that would be useful in a field of their own interest or expertise.

In the cases discussed in this thesis, aspects of classroom practice, learning theories, and pedagogical beliefs by programmers carried across to the titles they created. An emerging theme shown by a number of the titles covered in this study are that the programmers created works that encouraged higher order thinking skills and other concepts and behaviours that go beyond subject-based learning.

Granny's Garden is a title fondly remembered beyond its country and computer platform of origin. Contemporary reflections, however, note the limitations of *Granny's Garden*. Online user 'Oddbloke' (2012) muses:

> For many people my age, Granny's Garden is the very embodiment of nostalgia. It is always discussed in very endearing terms. But please don't hate me when I point this out: it really isn't as amazing as you remember!... The puzzles were solved mostly by luck...rather than any real judgement.

Similarly the Raspberry Pi forum user 'liz' wonders:

> There were other bits of educational software kicking about in those days whose usefulness I still simply can't get my head around. What was Granny's Garden actually for?

Despite the game having less freedom when weighed against modern titles, Satchel Software's ports to platforms beyond the BBC widened the reach of the game to schools in Australia and around the world where the BBC was less commonly used.

At a time when much of the educational software in schools was subject and school district specific, *Granny's Garden* proved an early example of an engaging graphical adventure style of gaming for primary students. This type of game – one that embraced logical puzzles and engaging story lines – and its associated gameplay paved the way for many titles on 8 and 16-bit platforms in schools.

4.4.3 Jara-Tava – The Isle of Fire (1998-1992) and Path Weaver (1985)

Jara-Tava – The Isle of Fire (hereafter referred to as *Jara-Tava*) is a graphical adventure, treasure hunting, and problem-solving game. The game has origins in a text adventure game written by Hodgson in 1980. Earlier versions under the working title *Treasure Island* were released into the public domain and created with *Path Weaver*, Satchel Software's adventure game creator for teachers. *Jara-Tava* demonstrates iterative improvements in design, graphics, audio, complexity, and classroom utility. It was released on multiple 8 and 16-bit platforms over several years including Commodore 64 and Amstrad (1988), Apple II (1989), and PC DOS and Commodore Amiga (1992).

The aim of *Path Weaver* was to allow teachers to craft adventure games tailored to their students by giving "control back to the teacher" as "adventure games can be used across the curriculum in almost all subject areas and at all year levels" (Angle Park Computing Centre, 1985). The origins of *Path Weaver* as a tool for teachers to design their own text adventure games, and ultimately the release of *Jara-Tava* as a more advanced graphical adventure game were inspired by Hodgson's own interest in *Dungeons and Dragons*. As Hodgson explains:

> I was very influenced by *Dungeons and Dragons*…so I started writing old style text computer adventure games very early on, way back [1979 and] 1980… I got to the point where I'd made a basic program that you could use to write your own games, and that was one of the ones that Angle Park asked me to [complete and release]. It ended up being a thing called *Path Weaver*. And the idea was teachers could use that to make up little adventure games which then uses part of a unit of what [students] were studying.

Path Weaver Version 1.0 was released in 1985, and was written by Hodgson and Paul Starick, another programmer at Angle Park. The manual provided a guide for teachers explaining how the menu driven system within the software could be used to create adventure games. The first ten pages of the manual set the scene for teachers who may not have been familiar with adventure games, and it is explanatory and persuasive regarding the benefits that bespoke adventure games could provide. This introduction sidesteps the technical aspects of *Path Weaver* and focuses on curriculum relevance, examples of classroom use, development of logic and reasoning skills, and the ultimate goal of Angle Park providing a public domain library of adventure games to share with South Australian schools. The *Path Weaver* software was relatively simple to use, with explanations, diagrams, tutorials and examples spanning less than fifty A5 pages in the manual. Hodgson explains the software's simplicity and utility further:

> …you could make up your adventure game with it. You didn't have to write a computer program to do it. So it was kind of like a database in a way but it had a component that you could put in your different

> locations and then you could put your movements in, and objects and even vocabulary and put a whole game together…The idea was teachers could use it to do games, but children could do it as well, primary, high school level kids, they could use this because it wasn't that complicated to do a game at all.

Hodgson elaborates on what he saw as the utility of computers as educational tools and how *Path Weaver* empowered teachers, although it probably wasn't used by as many schools as Satchel had envisaged or hoped for:

> …a student primary teacher [was teaching the book] *We're Going on a Bear Hunt*, and she wrote a little adventure game using *Path Weaver* on an Amstrad machine that was part of [the teaching unit] that she did. And that's exactly how we saw computers being used at that time. They were part of what you did to teach something…teachers would take children's books and they'd write a little game based on it. Not a lot of that but we had a few things like that.

Jara-Tava should be considered the zenith of Satchel Software's adventure game production. With the collaboration of numerous programmers, designers, musicians, and graphic artists, the comparatively simple *Treasure Island* text adventure concept was expanded upon. Hodgson discusses this further:

> I'd written this little *Treasure Island* game, a text game. [Hodgson and Walsh] took that, embellished it into one we eventually called *Jara-Tava – The Isle of Fire*, and we dropped all these [items] on

> these two islands which were references to a history. You could literally make up a timeline of what happened on that island…And that game came with a big book of classroom activities based on all the different things that were scattered around with[in] the game.

The hard copy resources contained within the package total nearly seventy pages. They consisted of activity-based materials to supplement the game, as well as a map sharing its more than twenty locations. These materials suggest entire units of work could have been built around the game, as suggested by Walsh's ideas of thematic packs supporting the software created by Satchel. Disk-based classroom resources were also contained within the package. These resources include a word-processor, database, and *Tree of Knowledge* (a question and answer game based on the *Jara-Tava* game).

In addition to the educational resources, disk-copying software was included in the package. Satchel encouraged duplication within schools of not just hard-copy teaching and learning resources, but also the software itself. These copied disks were to be used as working copies of software in the classroom (as noted in some of Satchel's software manuals). There was no copy protection on most of Satchel's titles, but there were provisions for site licenses for schools where cheaper volume prices were given. This apparent paradox suggests a degree of trust was placed on schools that they would backup and use software appropriately and not breach copyright. A floppy disk drive may have been network shared between multiple computers via a switching device, with several drives per computer room. It is plausible that multiple disk copies from one purchased copy could have serviced a single classroom. It is uncertain how and if this was monitored, although the presumption is that schools would generally have followed the licensing agreements included at the beginning of most written instruction manuals.

Jara-Tava demonstrates a number of improvements over previous Satchel titles that go beyond its graphical and audio qualities, which even in the 8-bit incarnations were noticeably improved over earlier programs (Figure 4.11 comparing). The programmers of the various ports took advantage of the target platform's capabilities, such as mapping some commands to the Amiga mouse. Customisation by teachers, as in *Mathbooster*, became possible affording granular control over the game's difficulty. The game itself allowed for a range of logical solutions depending on the approach taken by the player, with four ways of travelling between islands, different choices of items to use depending on player tactics, and three ways of finding the treasure to complete the game. This allowed students to apply different problem-solving strategies and higher-order thinking skills, as well as improving replayability. Hodgson explains that the hitherto standard of one or two typed words used to navigate text adventures[12] was also overcome. He states "I also cooked up a really nice intelligent smart parsing [system]… if [students] typed in a sentence it could pretty well work out what they were trying to do."

[12] Commonly used examples of instructions in text adventure games at the time include GO NORTH, LOOK UP, CLIMB LADDER, READ SIGN.

Commodore 64 Version

Amstrad Version

Amiga Version

Figure 4. 11: *Jara-Tava – The Isle of Fire* – Commodore 64, Amstrad CPC, and Amiga versions.

Jara-Tava was received well by reviewers at the time, with reflections including “All in all, *Jara-Tava* is an outstanding package that does more than merely entertain players. It stimulates and educates them.” (*Compute Magazine Issue 154*, 1993). The Amstrad User (*Jara-Tava: The Isle of Fire*, no date) elaborates:

> [Jara-Tava] requires the skilled hand of an enlightened and caring teacher to capitalise on the possibilities provided…Jara-Tava has been well pitched to the middle-to-upper primary age group. It's simple enough that it is immediately accessible, even to the child who's never seen a computer keyboard before, yet challenges the child's powers of reasoning and deduction Best of all, this game is fun, and making education fun while not compromising standards is a challenge Angle Park Computing Centre are well on the way to licking.

It was one of the more ambitious educational gaming packages offered by Satchel and was sold across Australia and overseas for more than five years (similar to the shelf-life of *Mathbooster*). Where it differs from simpler educational game titles is that a larger number of people contributed and collaborated on different aspects of the game design, code, graphics and sound.

My research indicates that *Jara-Tava* was one of the last full-featured gaming titles developed from the ground-up at Angle Park. By the mid-1980s, Satchel Software’s output and focus was veering toward the production of application based educational software and teacher support packages. There are likely a number of reasons for this. Educational gaming titles tended to be sold for less than serious productivity software, so economically it made sense to create pricier packages (Satchel did offer generous site-licenses for schools using their software). Although the various APCC programmers had devised innovative solutions for creating software for multiple platforms, the lack of standardisation of computer platforms in schools increased their workload when creating the various audio and visual aspects of game software. Finally, the programmers at Angle Park had started to use their own software to modularise the creation of educational packages for schools. By 1985 Satchel had already released a number of application programs including *Path Weaver*, *Softword* (a simple word processor), and *RAMfiler and COMMBASE* (easy to use database programs for schools). An example of how Satchel published a package repurposing their existing titles is *Classic Fantasies* (1987). This thematic package is based on novels by H. G. Wells, Arthur Conan Doyle, and Jules Verne, and utilises resources created with *Path Weaver*, *RAMfiler*, and *Tree of Knowledge*. A package such as this with numerous facets and options that could be used across many areas of the curriculum, some of which are detailed in the comprehensive fifty-page handbook, could be used for weeks or months on end by classes of students with sporadic access to computers. There is greater efficiency of production, and utility and flexibility for students with such packages over 'just' games, no matter how well structured or received these educational games were.

4.3.4 Forté – The Integrated Package (1998)

By the mid-1980s, the Angle Park programmers had produced a number of productivity application software packages. Word processors, database management programs, graphics utilities, and even a programming language (*ADL* for the Apple II, designed created by Rosenhain) were published by Satchel before 1985. Seizing a gap in the market, the concept of an all-in-one suite for use in schools was conceived, as explained by Rosenhain:

> Following Peter Sandery's vision we ended up doing things like writing…*Forté*… That was written for the IBM PC…it's an integrated package like a word processor, a database, and a spreadsheet [plus a report generator, and communications software]… The reason we did that was because schools couldn't afford Office. Microsoft Office was well over one thousand dollars per license…so we ended up investing in this… There weren't many competitors out there… Lotus 123, Harvard Graphics, Wordperfect, and they didn't work together. The concept of having them all work together is the attractive bit.

A number of the APCC programmers who created the aforementioned application packages worked on *Forté*, in addition to repurposing existing Angle Park software. Released in 1988 for IBM (DOS), *Forté* is an integrated package included a WYSIWYG word processor, spreadsheet, relational database, report writer, and communications modules. This package (including a several hundred-page user guide) was created in collaboration with a number of programmers at Angle Park. Although Hodgson wasn't directly involved in creating the package, he elaborates about its creation and use:

> [*Forté*] was an integrated package…that a team produced… This ended up being used in a lot of high schools here. The data between the programs was interchangeable. You could move it from a spreadsheet into the word processor. The word processor was essentially based on *Softword* but embellished and changed around [to

> match the other modules]. It was a DOS system not Windows, so it was all text based [with menus]…[*Forté* was] widely used in schools here back in the early 1990s.

Forté was an ambitious project designed with three main benefits for schools and students. It was considered an affordable alternative to other DOS packages of the time, provided interoperability between the different modules, and demonstrated consistency in operation and use, for example the saving and loading of files uses similar menus or keyboard combinations.

The creation of *Forté* heralded a number of forward-looking approaches toward software production for Angle Park. The aforementioned consistency in design, interoperability, and use of the different modules pre-dated modern computer standards. In order to ensure standardization between the modules, there was a 'Project Coordinator' rather than each of the four main module programmers working entirely in isolation. Other than some collaboration between audio and graphics creators, there was usually only one or at most two coders for previous projects published by Satchel. *Forté's* modules were still coded independently, but the production of this package was a stepping-stone toward multiple developers working on a single task. As Rosenhain reminds, prior to ubiquitous networks and shared storage:

> There were no source controls, there were no shared repositories, things were on floppy disks. We eventually had hard drives…a five megabyte hard drive was pretty cool… A lot of it wasn't done by massive collaboration when writing an application… It evolved a bit.

> It probably wasn't until NEXUS came along that we had true multi-developers working on [one] thing.

The quality and comprehensiveness of the *Forté* product and its documentation suggest that only targeting schools rather than homes or small businesses may have been a missed opportunity to broaden the reach and use of the software. Given the lack of integrated, consistent, and affordable business software at the time, this was somewhat surprising. When queried on this, Rosenhain suggests that there was no effort made to sell or market *Forté* to the general public:

> Our only interest was in schools, so I don't even know if we tried to sell it anywhere else apart from schools…Our target was South Australian schools, but we were selling software all around Australia by this stage, as the quality of the packaging [and software] went up.

Further opportunities to develop the software arose after the production and refinement of the original version of *Forté*. Computers running DOS becoming more common in primary schools during the early 1990s at the expense of the range of 8-bit machines that had shared the market in the preceding decade. *Forté Junior* (1991) was conceived as "a cut down version of the word processor that one of the other guys at Angle Park thought would be a good idea" according to Hodgson. Simplified commands, large screen fonts, and even predictive text brought the power of the main program to primary school students. Microsoft Windows, which was eventually released in November 1995 as a replacement for DOS as the user interface on compatible computers, was targeted as the platform for an improved version of *Forté*. Unfortunately for the Angle Park programmers, Microsoft's own integrated office package for Windows (*Microsoft Works*) led to the halting of development of future versions of *Forté*.

There was a lot of talk of developing a Windows version of this. They actually started looking at it seriously, doing some planning. Microsoft came out with *Microsoft Works*[13], which was essentially the same thing. That killed [*Forté*] off, but they did look at it for that. (Rosenhain, 2017)

4.3.5 NEXUS

It's hard to imagine a world without email or without the Internet, but that's exactly what we had in those days… Bulletin boards were something that [were] cropping up in the 70s and 80s…I said to Peter Sandery "Let's do our own bulletin board." (Rosenhain, 2017)

NEXUS was an online information system conceived by Rosenhain for use by South Australian school students and teachers. It was first established during 1984 and 1985, with three main versions deployed throughout the late 1980s into the early 1990s. Hodgson recalls that early versions of the software predominantly allowed email (or messages) between users, although this was soon expanded to included databases. Rosenhain remembers using terminal emulation software running on computers at Angle Park to connect remotely to CompuServe in the United States. As there was no available off the shelf software to run a bulletin board service from Angle Park, Rosenhain decided to write his own:

---

[13] Microsoft Works was an integrated productivity office package that was either very cheap or even bundled with computers running Windows during the period of 1997 - 2007.

> I decided that it would be a good idea to start our own…We went through three iterations [of *NEXUS*], so it must have been early '84, that sort of era [when the idea was conceived]. I got an Apple II… In Pascal [I] wrote a bulletin board service on it and connected one or two modems…and hosted this bulletin board…We told teachers about it and they therefore had to buy a modem…They'd dial up these two telephone numbers we [Angle Park] had…They'd leave messages and get some information they could search.

The software outwardly mimicked the CompuServe service, with numbered menu items giving the user choices to access options. Rosenhain shared the information about their rudimentary bulletin board service with teachers, and eventually found that the two dialup lines into Angle Park were often busy. The hardware was quickly upgraded to include from six to eight modems. The service was well-used from its inception.

As the service evolved, there were six main components providing users access to electronic mail, chatting, software reviews, bulletin boards, news, and database services. Due to the variety of microcomputers in use during the 1980s, the service was platform independent. The staff at Angle Park had visions of sharing Satchel Software's library through the service, predating the ability to play or stream games on demand. This was, however impractical as Rosenhain points out that the first modems they used ran at 300 baud, which would take two seconds to transmit a single line of text, which was too slow to transfer programs to a users' local computer. Had *NEXUS* continued, I surmise that this feature would have ultimately been implemented due to the incremental increases in modem speeds throughout the 1990s. Beyond the ‘standard’ features offered, the bulletin boards were tailored specifically to schools, teachers, and students. The news service *NEXUS* offered was innovative, and by 1987 it was able to pull in data from the existing Australian Associated Press (AAP) news database (Clyde, 1988). Rosenhain elaborates:

> One of the biggest features…we went to the Adelaide Advertiser and we got a feed…from this news service that sent stories across the wire [from overseas]…There would be a header and then a bit of text…there’d be one of these news stories, and they came through every few minutes…from somewhere in the world… We took that data and pushed it on the hard drives of the [server] so that schools could read these articles…it was an amazing service, an example of what you can do through computers.

By 1988 version 3.1 of *NEXUS* had launched, with over one thousand users across South Australia accessing the various services offered. Due to the rapid expansion and high use of the service, a Sun Microsystems server was installed with faster modems, with the *NEXUS* software being rewritten to suit the new hardware. The iterative nature of improving the service allowed Angle Park to develop the capabilities and speed of the system, with Rosenhain recalling:

> For me personally, and for Angle Park Computing Centre, it was a real playground for what could you do with dial-ins, and communications, and having people linked up…One of the things I wanted to do was fast data retrieval, so we started putting bigger databases onto the server. I wrote full-text indexing routines to find words…this was written in C.

These search, data retrieval, messaging, and email capabilities pre-dated the popularisation of widely available home Internet and web searches by several years. An instructional video primarily aimed at teachers, with some student related content, was produced and distributed to schools in 1991 (South Australian Education, 1991). Similar to early microcomputer software including detailed instructions about loading and operating titles, this video was an introductory instructional title assuming no previous knowledge of accessing information online. Additionally, there were numerous scenarios demonstrating the utility of accessing information, conducting database searches, and incorporating a variety of data gathered into useable and presentable content for school research assignments. The benefits presented clearly delineated how accessing current and up-to-date information online could potentially be more effective, timely, and relevant than data retrieval from hitherto traditional library books and resources. It is clear from this video that the APCC saw great potential in *NEXUS*, and more broadly online access of information, as a potential game-changer for education. The APCC had intentions for *NEXUS* to be developed into an "Australia wide communications user pay network" as part of "cost recovery programme within their software development section" (Fleer and Stout, 1991), but this did not eventuate. Although there is no budget or revenue figures available for the APCC, the requirement for cost recovery as part of the *NEXUS* project indicates a need for the APCC to generate greater revenue beyond the sales of Satchel Software titles. The fast-evolving Internet with its standardised protocols and methods of accessing information throughout the 1990s resulted in the demise of *NEXUS*. There was increasing educational utility provided by Internet and ultimately the World Wide Web, usurping the proprietary model driving *NEXUS*. Additionally, the focus of technologies within the South Australian Education Department shifted; it is not unsurprising that consequently the

NEXUS project, Satchel Software, and the APCC itself were ultimately wound down. Rosenhain reflects upon the success of *NEXUS* and how it overtook his work on individual Satchel Software titles:

> It's hard to explain how popular *NEXUS* became. It outgrew the Satchel side of things. I think it's because it was so interactive, it had people talking to each other. The popularity was huge…the Internet changed all that.

4.4 Winding down Angle Park and Satchel Software

Despite selling software across Australia and overseas, and the development and deployment of *NEXUS* being successful, the APCC closed in December 1995 and Satchel Software publishing ceased within the following two years. By 1991 the APCC had "ten professional staff and four clerical staff, who have been encouraged to take on an entrepreneurial role with software development and courseware production" (Fleer and Stout, 1991). The Technology School of the Future (TSOF) was established in South Australia in 1989 providing hands on teaching and learning within the wider domain of Information and Communication Technology (ICT) and media education for students and educators in South Australia. The TSOF filled a role similar to that of the APCC prior to its software publishing endeavours. Operating two similar but complementary computer and technology focused branches within the South Australian Education Department would have been challenging to justify financially for the State Government, especially in the years following the State Bank of South Australia collapse in 1991[14]. Despite the relative successes and prolific software output of the APCC and Satchel Software, numerous factors contributed to the temporal nature of their work. Influencers resulting in the winding down of Satchel Software included the maturation of the educational computing and technology market, the population becoming familiar with computer and online technologies minimising the APCC's influence and relevance, shifting focus of the Education Department in the sphere of educational computing and training, the rising popularity and accesibility of the Internet in students' homes, and the evolution of computing in general. Rosenhain suggests the maturation of the computing industry in general as a key factor contributing to the changes affecting the closure of the APCC despite continuing to develop *NEXUS* and produce other software:

> *NEXUS* was big and growing. Graphical and primary adventure [software] was strong…[but the Angle Park Computing Centre's] heyday of software development was winding down… [There were] lots of factors around that, but I think availability of better-quality software generally, the Internet, the whole industry was picking up… Schools became more aware and a bit smarter about their choices.

Rosenhain left the APPC and joined a company called CPM&S in 1990 with two other staff previously from Angle Park. For a time after he left, he was contracted back to the APCC to complete work on the final version of *NEXUS.* Although home, school, or business Internet access was not yet commonplace, he could already

---

[14] South Australia lost approximately $970 million dollars due to the State Bank collapse, which affected South Australian budgets and government spending for approximately twenty years.

foresee that the Internet was going to change the computer landscape, especially in education. He believes that there may have been an intention for the government to run Angle Park with *NEXUS* as an Internet Service Provider (ISP). Numerous ISP focussed businesses were founded in South Australia in the early 1990s, and Rosenhain believes that it did not make commercial sense for the Education Department to do so:

> I left Angle Park Computing Centre and the web [Internet] appeared…things happen like that and it changes the whole relevance of an organisation or a product…Bulletin boards that were big news in the 80s became or migrated into ISPs in the 90s and became the place to connect to the Internet. Teletyping and terminal stuff [evolved into] the web. There were a few [other factors] …Peter Sandery left…[and] you could see that the Education Department was less willing to spend money on [the APPC]. And the focus had moved to another organisation the Education Department had set up called the Innovation Centre [the TSOF]…[which] was started to do that next phase. It was looking at innovations rather than a computing centre. It was looking at wider things like CAD and robotics.

Hodgson reiterates Rosenhain's suggestion that the TSOF was a spiritual successor to the APCC. Regarding teacher training and how the TSOF inherited the roles of the APCC, he notes:

> The [Technology] School of the Future had been created at Mawson Lakes and they were basically training teachers, they were essentially doing the same things we were, or we had been doing but with a…lot more support from the government and a lot more funding to do it.

> They weren't in the business of producing software, they were doing teacher training and they did it with a very high public profile…we concentrated more on the software. So somebody made a decision that we don't need both organisations.

The shift from many hardware platform choices in the 1980s to a choice of either Windows or Macintosh in both primary and secondary schools reduced the need for locally produced ports of software to multiple platforms. By the 1990s, Satchel's titles such as *Forté* and *Forté Junior* were produced for PC only rather than a range of platforms. This may appear to have been beneficial by simplifying production of software to target only one or two platforms. Once the APCC had closed, the remaining staff producing software relocated to Ingle Farm High School. Some of the staff continued to develop *NEXUS*. Hodgson focused on developing a school library catalogue management software utility called *Bookmark*, title he has been working on in various iterations since 1982. He suggests that departmental reorganisation and the complexity of educational software titles required by schools in the 1990s introduced additional challenges that contributed to the ultimate demise of Satchel Software as a publishing imprint:

> Angle Park, it was closed down…the whole Angle Park operation was combined with the publishing group in the [Education] Department…and this was end of '95, '96. We were moved from the Parks Community Centre up to Ingle Farm [High School]… The packages we were producing then were much more complex, they were going onto CDs…the *NEXUS* people and my *Bookmark* operation were then transferred to what is now known as ICT services, [it] was the IT Department that was doing all the administration for schools. And we became part of that in about

> 1997…Satchel was pretty much closed down at that time…They stopped doing any new software. It had kind of wound down anyway partly because of the size of the packages and also just because of the reorganisation of the Department.

As of 2017, Hodgson continues to develop *Bookmark,* and he is still employed by the Education Department. From the software's humble beginnings on the Amstrad, it is used by many schools in Australia and even some overseas. Despite not being branded as a Satchel Software title, Hodgson explains it is the one remaining title that is still being developed from the APCC and Satchel period:

> I had developed a piece of software that managed the school libraries called Bookmark. It had become a big operation in its own right. It was earning a lot of money, it was being run as a commercial operation, it [earned] more than cost recovery by a couple of us that were working on it, and most of the schools in the state had adopted it… but it really started about '87 when I had, school principals were asking me, "Can I have something that is not going to cost an arm and a leg that will work on the computers we have in the school to do our library?" And that's where Bookmark actually came from. It was that, it was a request from schools for it and I went ahead and did it. So you know what I'm doing nowadays which is still Bookmark is the only survivor of that whole effort, that's it.

The TSOF, ultimately based in the Education Development Centre in the inner Adelaide suburb of Hindmarsh, was eventually wound down during 2006 (Kleinig, 2006). This decision by the Education Minister cited declining attendance at the school and was made in preference for remote support via video conferencing, online development of activities, and in a throwback to the APPC days, technology teachers visiting schools to offer support. The closure of the TSOF concluded over thirty-five years of centralised support for computing and related technologies in South Australia. The support for computers in education offered to students and teachers during this period undoubtedly had a positive impact on the development of computing, related technologies, and education in the state of South Australia.

4.5 Conclusion

This chapter documents the history of the Angle Park Computing Centre and Satchel Software. It is a story that is uniquely and inherently positioned within its period of operation throughout the 1980s and early 1990s. The establishment of the APCC and Satchel Software was a culmination of the interests of teachers and support by the state government in emerging computing technologies during the 1960s and 1970s. The South Australian state government and education department furthered these endeavours by establishing the APCC. The APCC played an advocacy role for launching the use of computers in schools, and it provided support for schools, teachers, and students throughout the state for over twenty-five years. Staff were given autonomy to create educational support materials and programmes for students and teachers. Leading into the 1980s, teachers who use the APCC's resources or showed interest and aptitude in programming were provided opportunities to work in this non-traditional teaching environment. The large volume of educational software and support materials produced by Satchel Software through the 1980s and 1990s delivered high quality resources with a range of uses in schools across Australia and, in limited volumes, overseas. Winding down of Satchel Software was perhaps inevitable given the increased homogenisation and corporatisation of the educational software market around the world. Although it may seem counterintuitive, it would have been logistically challenging for the APCC to continue to provide a steady stream of software, support, and (in the case of *NEXUS*) computer network infrastructure for schools. This initial exploration of the story of the APCC and Satchel Software reveals scope for future investigation into numerous aspects of the operations that were undertaken during the APPC’s active lifetime.

5. JACARANDA SOFTWARE: A QUEENSLAND BASED EDUCATIONAL SOFTWARE COMPANY

5.1 Introduction

This chapter explores the roles played by Queensland based Jacaranda Software and their production of educational titles for Australian schools. It provides a contrasting overview of a second Australian educational software company, revealing numerous similarities and clear differentiations during a similar timeframe to that of the APCC and Satchel Software. By exploring Jacaranda, a privately-owned company, the motivations, approaches toward software creation, educational rationale of the software that was produced, its successes and challenges can be compared to Satchel, as a state government run concern covered in the previous chapter.

Jacaranda Software presents an opportunity for a case study of a commercial, privately-owned educational software development company. The success and contributions made by Jacaranda can be attributed to the positioning of software as a classroom tool for learning higher-order problem solving skills, in addition to the relative dearth of localised educational software for Australian schools during the 1980s. I address how the educational and teaching background of (most) Jacaranda employees informed software design, the accompanying support materials, and even the software packaging. I argue that even though the operation was commercially and financially successful during the lifespan, this was not a goal in and of itself for the software creators. This non-corporate mentality and student-centred approach toward software design afforded Jacaranda viability throughout their lifespan as well as success in a low-key manner. In addition, it provided Jacaranda designers a high level of control over software design and implementation decisions, including choosing not to implement copy protection on any of their software. This small-scale decentralised ethos continues to be sustainable and endures with the educational software currently being produced by Greygum Software, some of which are descendants of Jacaranda's work in the 1980s.

Jacaranda Software operated out of Brisbane between 1984 and 1992, producing numerous titles at the behest of the Queensland Education Department. They published over twenty educational titles intended for classroom use during this time. Jacaranda was a subsidiary of Wiley, that has a long history as a traditional academic and instructional book publishing company. Their titles were released on most 8 and 16-bit computing platforms used in Australian schools during this period.[15] An experimental venture testing the nascent educational computer software market, Jacaranda was profitable from their inception until their dissolution in 1992. A number of former employees (Bruce Mitchell, David L. Smith, and Steve Luckett) purchased "the remaining stock, rights, and equipment from Jacaranda, setting themselves up as Greygum Software", which continues to produce and sell educational software through to 2019 (*Greygum Software*, 2007). The software produced by Jacaranda spanned a range of genres. Their early works included a number of simulations and adventure games, including titles such as *Gold-Dust Island* (1984), *Raft-Away River* (1984), *Goldfields* (1986), and *Kraken: A deep-sea quest* (1989). The software designers' educational philosophies are noted in the open-ended design applied to these titles. These releases were designed to develop students' higher order learning, spatial awareness, and problem-solving skills. Many of the titles were designed for students to work on in groups due to the limited

[15] The predominant hardware platforms that Jacaranda released their titles on were the Apple II, Commodore 64, Microbee and BBC Micro. There were a small number of Apple Macintosh and DOS releases.

number of computers available in schools at the time. In addition, a number of subject or skill specific and application titles were released. Examples of these titles include *Maths Bingo* (1991) (mathematics), *Wordswork* (1985) (cloze reading), and *Wordsmith* (1985) (a simple word processor).

Exploration and appraisal of the titles and the bundled paratexts, which usually included comprehensive teacher notes and student activities, reveals the high-level engagement that the titles were designed to elicit from students. Interviews with game designers reveal a collaborative approach toward software design that is mirrored in the requirements for teamwork and communication among players that were built into their software designs. Contrary to this, there was minimal interaction between the original designers and those who ported software to different platforms, with a number of them working interstate away from Queensland. The simplicity of the software afforded this flexibility and allowed the programmers to recreate software on different platforms and solve problems of their own accord. Contrasts are made between the centralised approach of a core team of programmers and designers at Satchel Software with the centralised design / decentralised (contracted) coding approach of Jacaranda. Comparison between the two companies' approaches to software design and support material provides evidence that even in the early years of educational computing in Australia, the people creating these works were driven by similar pedagogical approaches and passionate about their work. Concluding the chapter, I discuss the legacy of local educational software and provide argument for recognition of both the works and their educational underpinnings in the broader history of Australian software, games, and education.

5.2 Data Sources

Jacaranda Software, a number of their titles, and three programmers and have been covered in recent historical research into games developed in Australia with information incorporated into the Play It Again website (Stuckey *et al.*, 2013a). There is, however, no published research and analysis of the broader history of Jacaranda Software, the programmers and the processes they undertook, and the software created by the company. This chapter continues the approaches employed in the previous chapter addressing the history of Satchel Software. Examples of Jacaranda Software's titles were examined from gameplay, educational benefit, and classroom practice perspectives. Supplementing the software itself, support materials produced by Jacaranda were inspected in conjunction and parallel with the software.

Semi-structured interpretive interviews with former programmers and staff of Jacaranda Software were undertaken during 2017. Interviews were conducted in person in Melbourne and via Skype from Adelaide, with any requisite follow up questions via email. The programmers interviewed were Roseanne Hood (nee Gare) and Philip O'Carroll who were both teachers and programmers, and Gerald Wluka who was a school student at the time he ported software for Jacaranda. Interview sessions ranged from thirty minutes to two hours. Participants were chosen from authors listed in library repositories of Jacaranda's titles, and from those listed in software titles or paratexts, which (in the case of Jacaranda's titles) usually included software programmers or designers. Additionally, this chapter draws on an interview with Jacaranda alumni and current Greygum Software owners Bruce Mitchell and Steve Luckett conducted by Melanie Swalwell, and emailed responses to questions received from David Smith in 2012. Similar to the processes undertaken in Chapter 4, a representative sample of Jacaranda's software and paratexts were examined to support the evolution of the company and its position in the emerging Australian educational software market.

5.3 The Emergence of Jacaranda Software in Australia

Jacaranda Wiley Ltd was the Australian imprint of John Wiley & Sons[16] (hereafter referred to as 'Wiley') established in Australia in 1954 (*About Jacaranda*, no date). Jacaranda historically produced a "wide range of books and other products for the school, tertiary, professional, reference, and trade markets" (*Jacaranda Wiley*, no date). In the early 1980s John Collins was the head of Jacaranda, when he "decided that there might be a place for materials for the computers that were then beginning to appear in schools." (Stuckey *et al.*, 2013a). Bruce Mitchell, one of the early employees of Jacaranda and ultimately a founder of Greygum Software, recalls that their entry into the educational software publishing market as based on a hunch that it might be lucrative (Swalwell, 2012a):

> As I was shutting down Jacaranda Software [in 1992] I did find a paper that had been put together by somebody saying that there was possibly a market [in the 1980s] and we should look at it.

The first employee Collins hired for Jacaranda Software was Rosanne Hood. During the early 1980s prior to her time with Jacaranda, she worked for the Curriculum Branch of the Queensland Education department. Moving from New Zealand to Australia in 1975 to teach in schools, Hood undertook further teaching training in 1978 to obtain her Bachelor of Education Degree. She chose to specialise in

---

[16] John Wiley & Sons was established in 1807 in lower Manhattan in the United States. They continue to have global subsidiaries and specialise in academic publications.

‘Computing’ at university, so called “because there was no ‘Computers in Education’” (Schmerl, 2017e) stream at the time. During this time, she learnt binary and programming using punch cards on an HP 3000 minicomputer. As she was still teaching in primary schools whilst studying, she used this time as an opportunity to expose students to computing and to teach simple programming to Year 7 students. Hood recalls, “They [students] thought it was wonderful fun. Any excuse for an excursion to go out and talk to [work with] a computer. [The HP 3000 computer students used when on excursions] filled the whole room. It was definitely sensational.” Her statement highlights both the novelty of computers in schools during the late 1970s and her foresight regarding their potential. Additionally, Hood would take an Apple II microcomputer into classrooms she was teaching in because the school(s) were not yet equipped with computers. Her year-long programming project in 1978 resulted in the creation of an educational adventure game called *Gold Dust Island* (1984), which was ultimately published by Jacaranda and will be examined in greater depth later in this chapter.

Seconded to the Curriculum Branch of the Queensland Department of Education in 1981, Hood worked on computing curricula for Queensland schools (the Computer Education Curriculum Project)[17]. She later found out she was considered the best applicant but was hired along with two other male colleagues in case she “couldn’t cope”. Part of her role was to run computer-training courses for teachers around Queensland. Hood notes that upon arriving at a school with several computers, a staff member asked, “Where’s the guy?” These are just two examples

[17] Discussed in the interview with Gare, and also included in the packaging for Gold-Dust Island.

that highlight the prevailing gender bias and assumptions of the time that micro-computing was considered a male domain. Despite these pre-conceived notions, the courses were considered popular and successful. She describes her training as including group-work and rotating through activities, both of which aimed to model and mirror the types of tasks the teachers could subsequently use with students. In addition to teacher training courses, the Curriculum Branch team wrote software (not named) and documentation entitled *Educational Software Development* for Queensland teachers and schools. Most of the teachers who had access to training and the materials produced by the team "loved it, they were very excited…there was always teachers' notes and student handouts…all they had to do was fit it into their other activities". Even during these exploratory forays into educational computing, Hood considered the computer a classroom resource rather than an object to focus on. Although early in her teaching career at the time, Hood makes clear that she saw the computer as a tool to facilitate learning across the curriculum, echoing the commentary and approaches undertaken by the Satchel Software programmers.

Hood's work in educational software brought her to the attention of John Collins of Jacaranda, who sought her out to set up the Educational Software Publishing department of Jacaranda Wiley. Part of the role involved hiring other programmers. David L. Smith, who was recruited by Hood from Melbourne Grammar School, and in an email interview with Melanie Swalwell (2012b) he observed:

> Rosanne was an interesting and bold appointment. Like Bruce [Mitchell] and myself, her background was classroom teaching rather than publishing, and she was/is highly individualistic – someone who

> will follow what she works out is the right way to go, rather than follow what others are doing or the conventional wisdom.

Smith credits Jacaranda Wiley's educational focus for Hood's appointment to lead the software division by John Collins, noting that Hood was an "original thinker rather than a corporate yes-man". Smith's own appointment to Jacaranda followed six years at Melbourne Grammar School where he was director of computing and had installed an early educational computing network of Acorn BBC microcomputers. Although not formally trained in programming, his mathematics skills facilitated the development of his forays into coding:

> [I] very much learnt programming on the fly as a means to an end…indeed what I love about programming is the ability, as a person with low manual dexterity skills, to sit down and create something essentially just by thinking.

Much like Hood, Smith saw beyond students learning about computer hardware as an end in and of itself, and considers the early years of educational computing a "golden age when genuinely thought-provoking [educational] software was written and used." He suggests this was primarily due to the relative simplicity of writing software for 8-bit computers and the lower administrative demands placed on teachers at the time. Jacaranda's search for subsequent employees who would "ideally had [have] a strong background in education, knew about computers, but above all was an independent thinker who would get things done" confirms that the core direction of the company was to develop educationally beneficial software for classroom use. Indeed, Bruce Mitchell and Philip O'Carroll who were subsequently hired or contracted as programmers, were also teachers at the time.

Philip O'Carroll, previously a lecturer in logic at the University of Western Australia, recognised the potential for computers to engage learners when a very simple game he programmed on a Sinclair ZX Spectrum computer held peoples' attention. Following the foundation of a new school in Melbourne[18], he started programming more polished educational software to use with his classes on Commodore microcomputers, namely the Vic 20 and Commodore 64. O'Carroll began writing educational programs because, as he explains "…[I] couldn't find anything that I really liked from anywhere else" (Schmerl, 2017d) illustrating both the dearth of software and the lack of titles suited to Australian schools. Many of his programs were mathematically based, but he wrote titles that spanned the curriculum. Under his own publishing company, Fitzroy Programs[19], titles were sold independently across Australia via mail order from brochures mailed to schools. Over four years in the early 1980s, O'Carroll wrote and sold approximately fifty individual educational software titles. He also wrote a number of books for learning computer programming for Penguin Books and Pitman Publishing.

O'Carroll describes much of this early software as "normally classified as drill and practice…but the way I programmed it, it seems more exciting than that", implying early use of multimedia capabilities to encourage student participation and learning. His text and graphical adventure game *Excalibur* (released prior to being employed by Jacaranda) allowed students to select their own level of difficulty as well as progress at their own pace, both of which are representative of the benefits of

[18] O'Carroll and his wife founded the Fitzroy Community School in 1976, which still operates to this day.
[19] O'Carroll first produced books for schools in the 1980s, branched into software development, and continues to publish readers for schools under the 'Fitzroy Readers' brand in Australia and abroad.

student focused learning that he believes computers bring to the classroom. In addition to Jacaranda becoming aware of his presence in the educational software market, there was a personal connection with Smith of Jacaranda, as O'Carroll's sons were at Melbourne Grammar where Smith was head of computing. He was subsequently hired to port five Jacaranda titles from the BBC Microcomputer or Apple II to the Commodore 64 between 1984 and 1986. Beyond recreating the logic of the software, he also refashioned graphics, sometimes using multiple Commodore 64 sprites to make the larger images, and sounds from scratch. So I just looked at what happened on the screen.". The Commodore had superior sound capabilities than that of the Apple II, with the Sound Interface Device (SID) chip capable of generating three channels of sounds. The graphic representation was at a lower resolution than the Apple II, and had to be adapted to suit the target system. O'Carroll recalls, "They accused my sheep of looking like poodles" in the game *Sheep Dog Trials*. Despite the unrealistic graphic styles, he recollects that any graphics in educational software at the time were generally well-received due to the comparatively rudimentary hardware of the time.

Despite the personal gratification and challenge of porting purely based observation rather than using existing code and assets, O'Carroll explains that, ultimately, he preferred to create his own programs. The work for Jacaranda did inspire him to create more 'entertaining' adventure games after this. As the Commodore 64 educational market declined due to the rise of IBM compatible and other 16-bit microcomputers, by the early 1990s it was no longer viable to continue supporting the Commodore with new software. O'Carroll was wary of learning new computer languages that may not afford easy transfer of his Commodore BASIC skills and knowledge base. Under the Fitzroy Programs imprint, he chose to concentrate on the production of primary educational books, focusing on mathematics and reading literacy. Greygum Software, formed by Jacaranda Wiley alumni David L. Smith, Bruce Mitchell, and Steve Luckett, currently produce the supporting educational software for these readers, highlighting that the professional connections made during the 1980s still continue in a viable form today.

Gerald Wluka was another contracted programmer who ported software for Jacaranda, in this instance from the BBC Micro titles to the Apple II. Wluka owned an Apple II, and was a confident and competent programmer; David L. Smith subsequently chose him to program ports of *Raft-Away River*, *Goldfields*, and *Wordsmith*. Of note, at the time Wluka was still a high school student in Smith's mathematics class at Melbourne Grammar School. Wluka recalls (Schmerl, 2017c) his first interactions with computers:

> My dad bought a copy of APC, Australian Personal Computer, the inaugural edition. That's when I got into it...I started hanging out in computer stores, and one thing led to another and I ended up getting for my Bar Mitzvah an Apple II computer, and that's how I got into programming the Apple.

Despite the relatively obscure text-based commands required to operate 8-bit microcomputers like the Apple II, Wluka learned not just how to use his computer, but to problem solve and program with them. He developed these skills in an informal way driven by his interest in computers, similar to the interest-based learning demonstrated by Hood, Smith, and O'Carroll, and also echoing those of the Satchel Software programmers. The 'do it yourself' homebrew approach to creating software and hardware for microcomputers was not out of the ordinary at the time[20]. Wluka was, after learning how to program microcomputers the age of 14 or 15, being contracted by companies to create large spreadsheets in VisiCalc. Echoing Smith's sentiments, Wluka recounts, "…so on the Apple II, unlike a computer today, you could understand everything…The computers were simple then. You could understand the electronics, the software, and the hardware." Despite slow processors and comparatively slow BASIC program execution, in addition to relatively simple computer architectures, 8-bit computers provided a realistic environment for the first generation of microcomputer owners to master machine code programming. Although learning how to use and program 8-bit computers by teaching himself, O'Carroll noted the limitations of coding software in BASIC (the accessible built-in programming languages used by computers such as the Commodore 64 and Apple II), namely less memory available to run software and slower execution of code. Wluka recalls, as a young student, there were a number of contributing factors that led him to not just learn to write software, but also to program more efficiently using machine code:

> There wasn't much else you could do with it. You could use VisiCalc...[creating] spreadsheets for my dad and programming was it. I didn't buy any games or anything, I was too cheap. First I got it just with a tape deck and I pretty quickly got a floppy drive. I started with Apple Soft Basic, and then I got other programming languages. I

[20] "They represent a moment in history when it was normative for people to write their own games and when computer games could be identified with the pleasures of coding and hardware tinkering rather than solely focused on the pleasures of play." Stuckey, 2015.

think there was a Pascal, which I played with and never really used. And 6502 machine code.

Similar to O'Carroll's recollections of being contracted to address gaps in Jacaranda's software library as the target Australian educational market had a wide range of 8-bit computers in schools, Wluka was also approached to port software:

> …he [Smith] was doing titles for Jacaranda Wiley on the BBC Micro. And my guess is they said, "Hey, we need to develop these on the Apple II", because there was a lot more of them around in Australian schools than there are of this BBC Micro. And he knew I was good on computers because he'd seen me in the lab playing with the BBC Micro…he knew I had an Apple II, and he said "Hey, could you do this?" and that's how I got introduced to it. I'm sure I was very cheap for Jacaranda Wiley to have me do their (work). [21]

Again, in a similar fashion to O'Carroll, Wluka was provided access to source code from which he could use to port software, but he found the BBC Micro code was not transferrable to the Apple II. Wluka was given access to the original titles on the BBC and analysed the provided source code. He then deconstructed the game logic then reconstructed it for the Apple II. In reference to *Raft-Away River* and *Goldfields* Wluka states:

[21] Wluka suggests he was paid $1400 for the Raft-Away River Apple II port, a task which took him forty to fifty hours. As a teenager during the early 1980s, he considered this very good remuneration for his work.

> I could just understand how the game worked and what the constraints were or the rules, just by looking at his code, and then I'd go ahead and implement it on the Apple II. That was all [coded in] BASIC.

This indicates both Wluka's skill for logic and coding that belied his age, but also the relative simplicity of the software he was porting. For *Wordsmith*, a children's word processing program, Wluka used a more efficient approach, programming in machine code (see more about *Wordsmith* in the software section below). Wluka ultimately abandoned the contracted work with Jacaranda, instead concentrating on his school studies.

The genesis of Jacaranda Software was founded on a hunch by John Collins that educational software might be a commercially and educational worthwhile market to enter. The evidence provided by the early software designers and programmers indicate that its foundation was built on employing staff who were not just good at using computers but had a combination of qualities. Recognition of microcomputers as a useful object at teachers' disposal, an aptitude to apply logic and problem-solving techniques to create educationally sound software, and an ability to work autonomously are three assets demonstrated by the Jacaranda alumni who were interviewed. These attributes are further recognised and explored when investigating a selection of Jacaranda's back catalogue.

5.4 Software Case Studies

This section includes analysis of select titles from Jacaranda's Wiley software oeuvre, exploring a range of their software, insights into development, teacher and student support materials, and use cases. There is no existing definitive list or archive of the software Jacaranda produced during 1984 – 1992, although twenty-seven titles released during this period were identified. Over half of their titles were released in their first three years of operation. The focus of this analysis will be on a selection of these early titles. The titles were chosen due to their open-ended nature and innovative approaches toward developing higher order thinking and learning skills. A number of titles are located in various Australian libraries, although these deposits appear to have been made haphazardly with little input, and perhaps some reticence, from Jacaranda or Greygum. According to Smith:

> In the early days libraries were blissfully ignorant of how copyright applied to software, and would buy packages and loan them out like books…We did experiment with producing library copies of some software, with various mechanisms for trying to ensure that the software was lent out and returned [rather than be easily copied], but this never caught on to any commercially significant extent.

Smith, Mitchell, and Luckett variously mention that the majority of their software produced during the 1980s through to the present is not copy protected. With the increasing onus on schools to ensure all software is licensed for site use, they found that their experimentations with copy protection were not warranted in the long term. Smith follows "We have found, or think, or maybe just kid ourselves, that schools are generally pretty honest, unlike say the home market. Certainly there have been instances of misuse, but not many." Beyond the consideration of copyright, copy protection caused complications with loading protected Jacaranda titles and, as computer suites in schools developed, difficulties with network installation of software. These challenges, in addition to increased production time, subsequent increases in technical support being required, and educational software being less of a target for piracy than games, were cited as reasons to negate the need for onerous copy protection.

> Mitchell: Because it [copy protection] caused so many support problems. People couldn't make backup copies of their disks, which they're always entitled to.
>
> Luckett: We had our own experience with other people's software that was copy protected. It wouldn't boot. It wouldn't load it… schools are worse because [the staff are] generally not particularly computer literate. They weren't at the time, and if you made it any more difficult at all you'd be in trouble. They'd be on the phone and there'd be support calls; [it's] too hard.
>
> Mitchell: Definitely too hard and all the companies that put copy protection on went out of business and we didn't.

Despite the lack of copy protection and online fan efforts, Jacaranda's softography has not been archived comprehensively across all platforms. Websites including archive.org and fan sites including Gamebase64, whose mission statement is to catalogue and "document ALL Commodore 64 gameware" (*Gamebase 64*, 2016), do not contain disk or tape images of every Jacaranda title. Nevertheless, fan archives, blogs, and forums provided access to disk or images and scans of packaging and documentation of a range of Jacaranda's titles. A selection of these titles were examined by being run in 8-bit computer emulator software, either native to Windows or Macintosh computers, or using archive.org's built in web-based emulator software. Additionally, some interviewees subsequently provided their software titles and paratexts (or scans). Some software was run on original hardware where possible. A number of Jacaranda titles from the 1980s and 1990s have a continued legacy in modern software published by Greygum Software. These modern iterations of Jacaranda's classic software were not explored for this study, but are resources that could be investigated in future explorations of Jacaranda Software's ongoing legacy.

5.4.1 Early forays in software publishing – Moving Into Maps series (1984)

Jacaranda Wiley's first series of four software titles was based on their *Moving Into Maps* "pre-atlas" primary school activity book (Butler, 1983). The book was positioned as an introductory mapping skills resource that was story-based. *Moving Into Maps* utilised "games and colourful graphics…relevant to the children's ages and interests – pirates, fun parks, fantasy, toys, board games, codes, animals and BMX bikes." ('Scavenger Hunt software documentation', 1983). A hitherto traditional paper-based resource used for "learning to read a map, coordinates, [and] scale" (including black-line masters of work sheets for students) (Schmerl, 2017e), Hood's first job at Jacaranda was to develop software to complement it. Hood, based in Queensland, and Smith, in Melbourne, designed all four titles. Smith coded all BBC Micro versions, with ports to other 8-bit platforms handled by contractors. By creating four software titles with a direct link to a popular existing resource, Jacaranda situated their first titles within a specific curriculum framework. This provided educational justification for schools to purchase the software, as well as potential tie-in marketing opportunities for Jacaranda.

Scavenger Hunt (1984) was the first educational software title released in the *Moving Into Maps* series; description and analysis of this title is provided as an example of the set of titles. The outcomes of the software were framed around teaching students the eight cardinal map points and estimating distance. Rather than a simple directional memory exercise, students actively participated in retrieving select items from a top-down map screen by avoiding obstacles and trying to achieve the tasks in the shortest distance possible (Figure 5.1). This title allowed one to four players to work together. Multiplayer titles affording collaboration are a key feature of a numbers of Jacaranda titles at the time, with Hood noting that this was intentional, in part due to the small numbers of computers present in schools that even had them installed. To improve replayability, Hood makes it clear that with regard to the items and the order in which they must be collected “everything is randomised, you can’t predict anything.” The gamified aspects of computer-based learning drove student engagement, along with randomisation. These software design philosophies aimed to develop deeper understanding than rote learning, underpinning the core of Jacaranda’s releases throughout their early years, primarily under the driving influence of Hood.

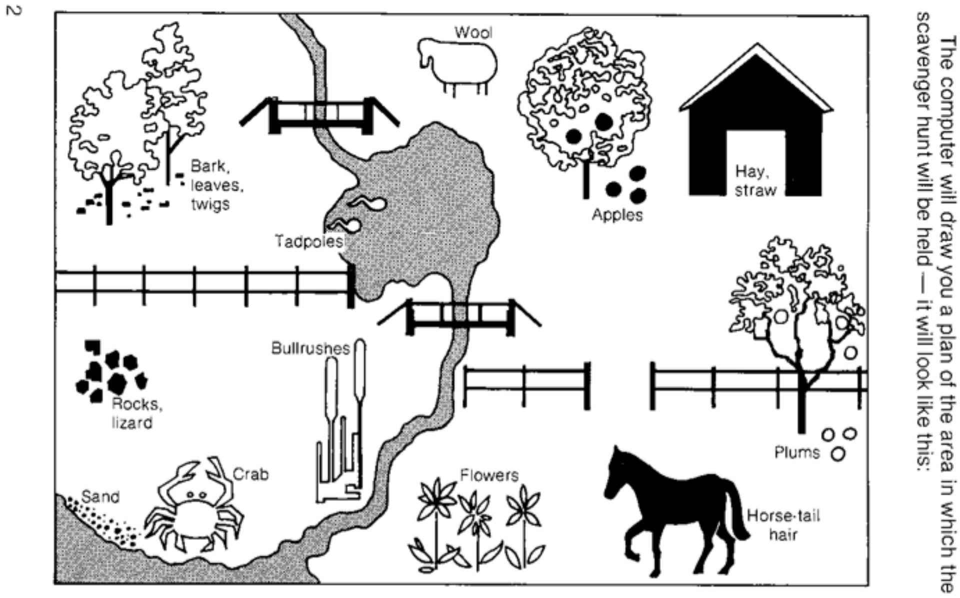


Figure 5. 12: *Scavenger Hunt map screen representation from Student's Guide*.

At this time of creating this game, Hood saw numerous potential challenges of both designing and programming quality education software, in addition to producing appropriate documentation for students and teachers, especially with a number of planned forthcoming *Moving Into Maps* tie-in releases:

> I said to my boss [Collins]…I can design this program, but to carry on designing programs [in addition to] writing them would take forever…If I could find a programmer who could do the job, then he could write the programs and I could design…That's when I found David Smith, who was a programmer…he was a teacher…He said leave it with me and I'll get back to you next week. He came back next week with a prototype.

A refined version of the prototype was the basis of the released software. Smith considers the small team of two was advantageous. Despite Hood providing documents planning the game, he considers that "we have always designed on the fly, helped by the fact that the projects have involved only a very small number of people." The themes of rapid software design and development, synergy between programmer and designers, and autonomy persist upon examination of the development of Jacaranda's other titles.

The design language of both packaging style and resource content for Jacaranda's releases were clearly established with these early titles. Bold colours, defined branding of the Jacaranda logo, rationale for classroom use and objectives formed key features of the snap-lock disk and tape cases. Instructional materials and resources were delineated into separate booklets aimed specifically at, in the case of all four *Moving Into Maps* releases, teachers and students. The Teacher's Guides for all four titles in this series contain similar information that can be referred to as 'standard', regarding software aims, operation, loading, prerequisite background knowledge and time required. Beyond this general information, three other inclusions in all four Teachers' Guides are noteworthy (Figure 5.2). Emphasis is provided discouraging teachers from direct supervision and allowing students to correct their own mistakes for "greater educational benefit". Most Jacaranda titles that included sound effects or music could disable these features with students pressing "Q" upon game launch. This inclusion is possibly indicative of the perception at the time of classrooms needing to be 'quiet spaces', despite these titles requiring student discussion. In hindsight, this was likely a fortuitous inclusion once computer numbers increased in classrooms. The 'Conclusion' in the Teacher Guides of all four titles highlighted Jacaranda's focused goals for their software; namely that student driven experiences with computers should be used as just another educational tool and an experience and not a replacement for any specific existing classroom activities. The software afforded opportunities for otherwise difficult to experience simulations and adventure games in a classroom environment, and that learning "should be fun, exciting, and effective." By virtue of educators driving the design of Jacaranda's software, these strong educational justifications for using their software provided compelling arguments for not only use these early Jacaranda titles, but also their subsequent offerings. The simulation and adventure style game genre that aimed to develop problem solving and group skills was developed and explored further in *Gold Dust Island*, the next title written by Hood for Jacaranda that will be explored.

Teacher involvement

It is usually unnecessary for the teacher to supervise the students to any great extent when they are playing *Scavenger Hunt*. The program is easy to use and children will gain greater educational benefit from correcting their own errors. However, it is important to ensure that they understand

- when to use the RETURN key,
- what the SPACE BAR is, and
- how to ask the computer for HELP.

The HELP routine

The program is able to offer help in a number of ways:

- It can remind students what is on their lists, which objects have been collected and give hints about where to find the remaining items.
- It also reminds students about the compass directions they are using and allows them to try out moves in these directions, thus helping them to judge distances.
- Finally, it will provide students with current information about their scores.

Noises?

You can cancel all the various tunes and noises that are produced in *Scavenger Hunt* by pressing "Q" instead of the SPACE BAR at the very beginning of the program. The operating instructions (page 4) give full details.

Conclusion

In *Scavenger Hunt* we have tried to embody the Jacaranda Software philosophy:

- Students drive the computer, not the other way round.
- The computer doesn't replace teachers, blackboards, playgrounds or books — it complements them.
- Activities that may be inaccessible, time-consuming or expensive can, through computer simulation, be made readily available in the classroom.
- Education should be fun, exciting and effective.

Figure 5. 13: *Scavenger Hunt Teacher's Guide excerpt*.

5.4.2 Gold-Dust Island (1984)

5.4.2.1 The Game

Gold-Dust Island originated as a paper-based tabletop game partly designed by an acquaintance (unknown) of Hood. During her year of study in Australia in 1978, she programmed the game for Apple II. After Jacaranda Wiley employed her, this title was amongst the first of seven titles released by Jacaranda in 1984 on up to four formats (Apple II, BBC Micro, Commodore 64, and Microbee). *Gold-Dust Island* was released on these four systems, with David L. Smith (BBC Micro), Philip O'Carroll (Commodore 64), and Gerald Preston (Microbee) programming the ports.

The game was originally programmed by Hood in Applesoft BASIC for the Apple II, with subsequent ports also being coded in other versions of BASIC native to their target platforms. According to Hood, Smith was responsible for some graphical reprogramming in order to make the game more "flashy". Typifying the autonomous working culture of Jacaranda employees and contractors, Smith recalls, "We always met regularly, but the design and development was always done wherever each of us happened to be at the time – just an electricity supply was needed." O'Carroll expands on this; "I did look at his [David L. Smith's][22] source code [for the BBC Micro version] but I couldn't do anything with it. So I just looked at what happened on the screen." These recollections exemplify both the 'remote office' ethos of Jacaranda long before the concept of working from home became popularised, in addition to the implied problem-solving ability of those who were employed to port the software.

---

[22] O'Carroll ported software from Smith's BBC Micro ports of software originally programmed by Gare for the Apple II.

Gold-Dust Island is described in the teacher documentation as a "social science simulation", revolving around a group of survivors of a shipwreck landing on an island and trying to survive. Two to five players are required, according to Hood not only to foster the desired outcomes of the simulation, but also to address the availability of computers in primary schools. As primary students used to rotate through different classroom activities, either individually or in groups, incorporation of this game into the classroom was seen as an extension of that type of task.

Although *Gold-Dust Island's* graphics and sounds are rudimentary by modern standards, the gameplay is relatively involved. Upon loading, students were asked to enter their names, which personalised the gameplay experience. A top down view of an island appears, with stick figures representing each player. Players were required to remember which avatar was their own, as the graphics couldn't display individual distinguishing features or names. The graphics are superficially similar on each platform (Figure 5.3), although not identical due to the different display technologies of each platform. The programmer of each port created their own graphics based on the version they viewed before writing the code for their target platform. This can be seen with O'Carroll's Commodore 64 version having a similar colour scheme to Smith's BBC micro port.

Upon beginning the game, a randomised assortment of items were given to the players, a feature Hood describes as increasing replayability and reducing rote gameplay mechanics. Players complete tasks around the island, namely boat building, fetching water, or digging for gold. There were only seven individual 'moves' a player could makeper turn, each triggered by a single character keypress

(‘S’ for status, ‘M’ for move, ‘D’ for dig for gold, ‘T’ for trade, ‘W’ for fetch water, ‘B’ for build boat, ‘F’ for fill well). These moves were conveniently noted on the screen in all versions of the game. By assuming specific roles, trading items, and cooperating, the goal of the game was for students to work together to escape the island before a cyclone ultimately rolls in to end the game.

Figure 5. 14: Gold Dust Island main interface. (1) Apple II, (2) BBC Micro, (3) Commodore 64

5.4.2.2 Software Packaging, Documentation, and Educational Rationale

The packaging and support materials provided with the game were comprehensive and clearly aimed at teachers and schools to justify the purchase. The layout and design style of the packaging and documentation (Figure 5.4) built on previous releases and was used as a template for subsequent titles. A prominent Jacaranda Software logo placed in the top-left (of the front and back) is the first thing the eye sees, demonstrating a desire to 'brand' their products from their initial release. The title, author and cartoon style artwork are reminiscent of modern school textbooks, rather than a game, clearly situating the software as an educational resource suitable for primary and middle schools. The 'About the Program' description on the back of the game box (Figure 5.5) explains, in three paragraphs, the purpose of the software as it relates to students' educational objectives rather than a more typical description of gameplay. In addition, The 'About the Author' section is aimed to promote the author's credentials as an educator, her educational experience, and a more progressive use of computer technology beyond typical drill and practice and mathematics / science applications seen in schools up to this point in time. Gare is noted as being "dismayed" by more typical educational computing practices and the resultant software package is described as a "meaningful" classroom tool. This description positions and markets this 'game' as a legitimate classroom tool.

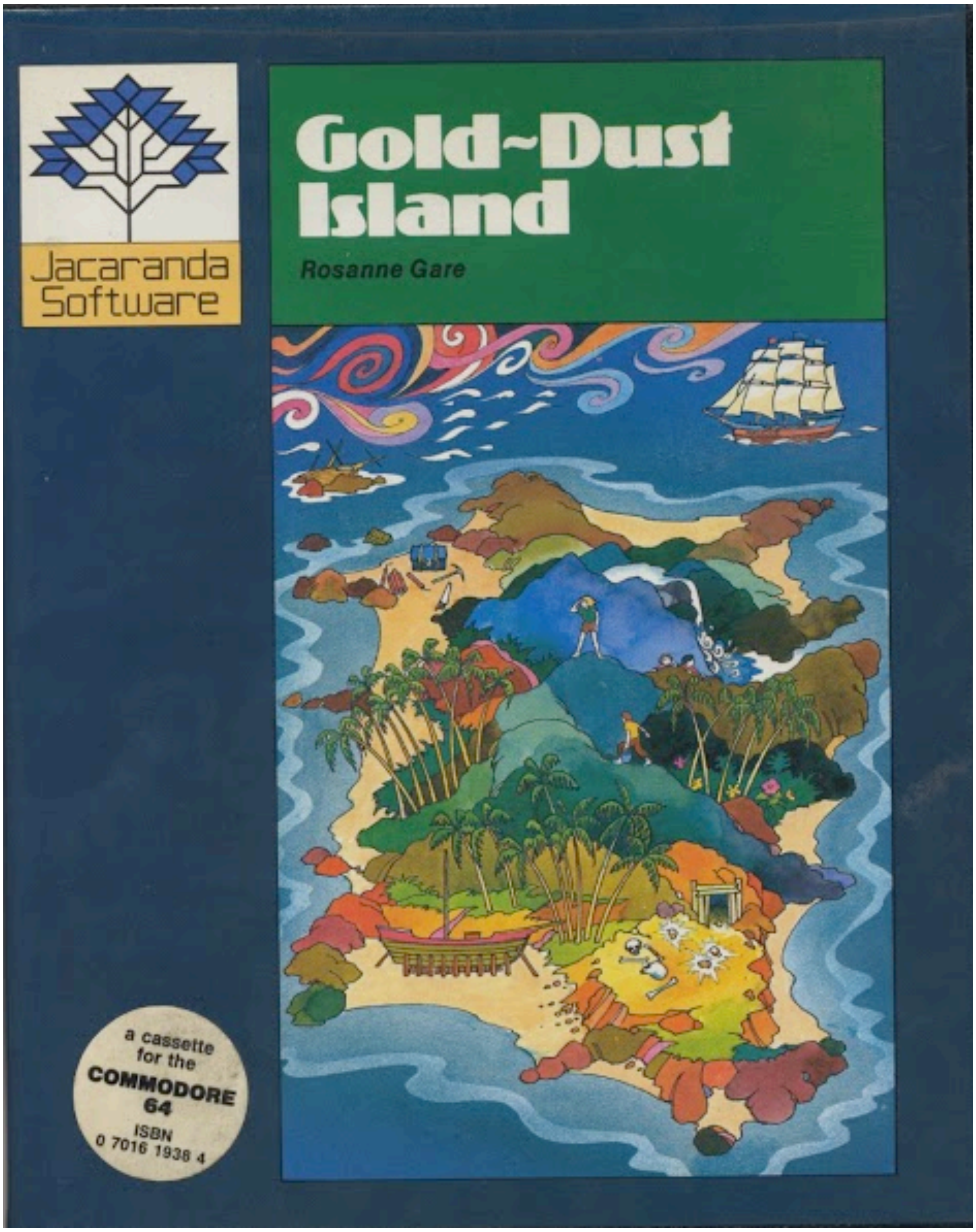
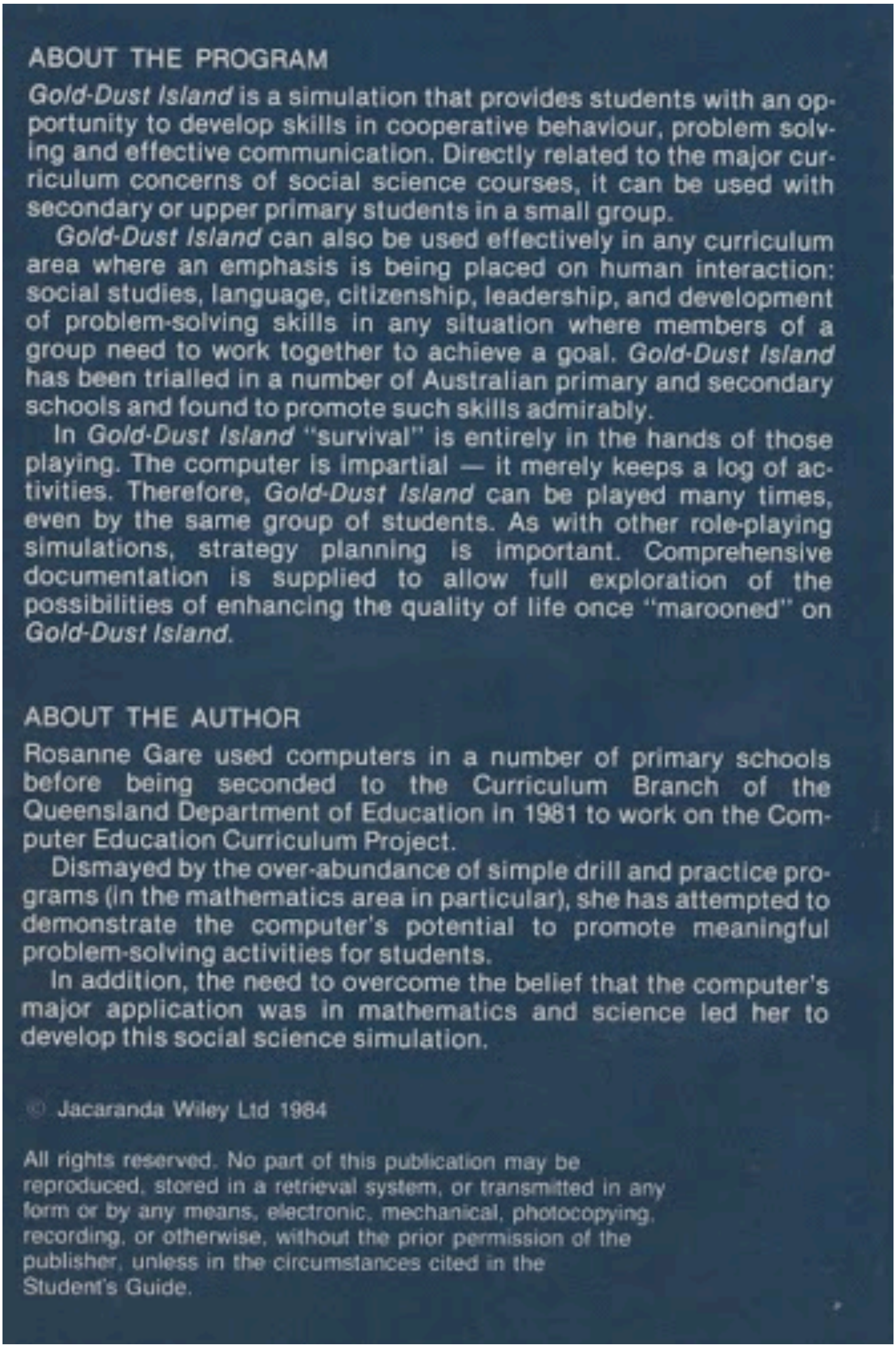


Figure 5. 15: Gold-Dust Island packaging front.



ABOUT THE PROGRAM

Gold-Dust Island is a simulation that provides students with an opportunity to develop skills in cooperative behaviour, problem solving and effective communication. Directly related to the major curriculum concerns of social science courses, it can be used with secondary or upper primary students in a small group.

Gold-Dust Island can also be used effectively in any curriculum area where an emphasis is being placed on human interaction: social studies, language, citizenship, leadership, and development of problem-solving skills in any situation where members of a group need to work together to achieve a goal. *Gold-Dust Island* has been trialled in a number of Australian primary and secondary schools and found to promote such skills admirably.

In *Gold-Dust Island* "survival" is entirely in the hands of those playing. The computer is impartial — it merely keeps a log of activities. Therefore, *Gold-Dust Island* can be played many times, even by the same group of students. As with other role-playing simulations, strategy planning is important. Comprehensive documentation is supplied to allow full exploration of the possibilities of enhancing the quality of life once "marooned" on *Gold-Dust Island*.

ABOUT THE AUTHOR

Rosanne Gare used computers in a number of primary schools before being seconded to the Curriculum Branch of the Queensland Department of Education in 1981 to work on the Computer Education Curriculum Project.

Dismayed by the over-abundance of simple drill and practice programs (in the mathematics area in particular), she has attempted to demonstrate the computer's potential to promote meaningful problem-solving activities for students.

In addition, the need to overcome the belief that the computer's major application was in mathematics and science led her to develop this social science simulation.

© Jacaranda Wiley Ltd 1984

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the publisher, unless in the circumstances cited in the Student's Guide.

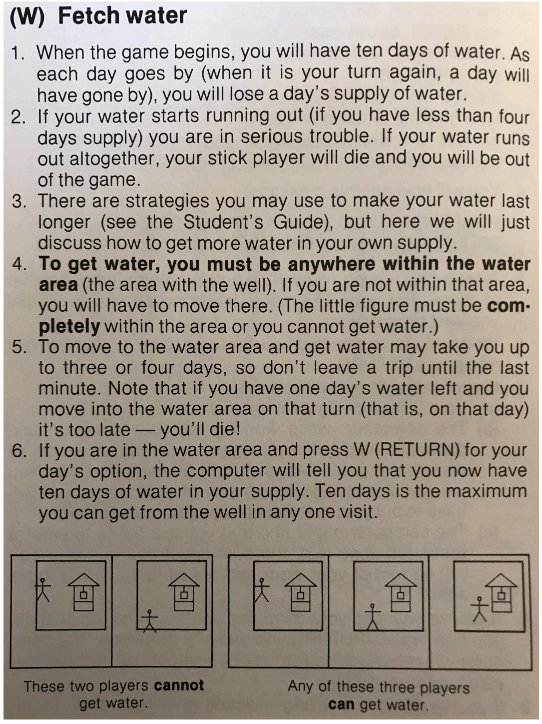
Figure 5. 16: Gold-Dust Island packaging back (cropped).

The package included three sets of A5 documentation booklets, each serving a different purpose. The layout of each booklet is easy to read, with numbered and labelled notes with obvious section headings to assist the reader, and features prominently branded Jacaranda logos (Figures 5.6 and 5.7). The Operating Manual includes detailed and numbered instructions about how to load the game (on all platforms), and step-by-step instructions explaining gameplay, across ten pages. These instructions provide details of almost every possible scenario or option a player may face during a game. The details present in these instructions belie the relative simplicity of the game. The rationale behind this contradiction is multi-faceted. Teachers at the time may not have been confident or experienced with computers, especially as each platform operated in its own way, hence the loading instructions are comprehensive, clear, and easy to follow. The gameplay instructions itemised almost all imaginable circumstances a player may have faced during a game, affording the prepared teacher or student who read them a deeper understanding of any gameplay situation that arose, including edge cases[23]. An example can be seen in the 'Fetch water' set of instructions. If a player's avatar was in contact with a boundary around the well, fetching water would not be possible. To the uninitiated player or teacher, this may have seemed like a bug, a problem with the computer, or incorrect input by the player. The instructions (Figure 5.7) clearly specify the conditions and rationale for this feature of the game, thus putting the player at ease and assisting the teacher in understanding what actions are or are not taking place.

---

[23] Edge cases occur within a program when extreme maximum or minimum values for a given variable are input or calculated.

Figure 5. 17: Gold-Dust Island Documentation covers.



(W) Fetch water

1. When the game begins, you will have ten days of water. As each day goes by (when it is your turn again, a day will have gone by), you will lose a day's supply of water.
2. If your water starts running out (if you have less than four days supply) you are in serious trouble. If your water runs out altogether, your stick player will die and you will be out of the game.
3. There are strategies you may use to make your water last longer (see the Student's Guide), but here we will just discuss how to get more water in your own supply.
4. **To get water, you must be anywhere within the water area** (the area with the well). If you are not within that area, you will have to move there. (The little figure must be **completely** within the area or you cannot get water.)
5. To move to the water area and get water may take you up to three or four days, so don't leave a trip until the last minute. Note that if you have one day's water left and you move into the water area on that turn (that is, on that day) it's too late — you'll die!
6. If you are in the water area and press W (RETURN) for your day's option, the computer will tell you that you now have ten days of water in your supply. Ten days is the maximum you can get from the well in any one visit.

These two players **cannot** get water.

Any of these three players **can** get water.

Figure 5. 18: Gold-Dust Island Operating Manual excerpt.

The Teacher's Guide is comparatively brief, yet still offers a clear rationale behind the *Gold-Dust Island's* use. The objectives of the game are listed as follows: "to promote cooperative behaviour, to encourage effective communication, to encourage group productivity, coordination of effort and division of labour; and to provide an enjoyable problem-solving activity." ('Gold-Dust Island Teachers Notes', 1984). Aims, objectives, curriculum relevance, and a description of how cooperative problem solving is promoted by the program comprise over half the manual. Suggestions for prior instruction, how to use the game, and follow up activities provided support for teachers throughout the process of using the game in the classroom context. This information provides, beyond basic game operations, how the game should be situated within an existing primary school social studies or social science course, although no particular topic context is suggested. This offered even more justification for purchasing the software as schools could use the title in a range of contexts and subject areas. It is clearly explained that the teacher can observe and give assistance to students, but the decision-making in terms of the game is best left to students, thus affording the development of higher-order problem solving and cooperation skills developed from learning by doing and assisting others. Finally, the guide includes brief descriptions of concepts that are implicitly explored by using the program; the group and its functions, locality and social relationships, roles, class and status, and power. These relatively advanced concepts for primary students could be learnt by playing the game and during post gameplay discussions and activities. It is clear that developing the skills discussed above was the primary concern of the developers. There is a statement on the packaging that the software directly "related to the major curriculum concerns of social science courses", but this could be interpreted as secondary to the outcomes desired by Hood and the interpersonal skills Jacaranda promoted in the software's documentation.

The final part of the package is a Student's Guide. This booklet provides a simple introduction to the game, walks them through the early decision-making processes, and includes an abridged version of the instructions provided in the Operating Manual. Rather than a step-by-step guide, the instructions are embedded in the story and description of the game. This implies that students using the software would learn the goals of the game by reading a story. A series of post-game questions relating to cooperation, resource management, planning and strategy, communication, and 'winning and losing' are also included. These could have been answered either in discussion with other members of the group, or as writing or creative activities post-game. This would have enabled students to reflect upon the communication, decision-making, and cooperative processes undertaken rather than view the game as a standalone activity. This is another clearly designed aspect of the software package designed to tap into students' higher learning capabilities.

5.4.2.3 Reflections on Gold-Dust Island

There is relatively little online discussion about *Gold-Dust Island* on fan sites, but some contemporary reflections have been uncovered. Cameron Davis of blowthecartridge.com, an Australian retro games blog, reflects on the game:

> In Primary School…we had Commodore 64 machines that did nothing but load educational software…and this game…it taught me a lot about life. There's not a lot to do in the game, but it's sort of compelling in its own way. Every player can choose on their turn to move around, "trade" items between nearby players, get water, dig for gold or build. The more relevant items they have, the more effective they'll be at the task at hand…you just give items to other players and get nothing in return. I guess the developer was hoping there'd be some kind of social contract in play in front of the computer which I can tell you was not honoured in the school room…The game ended up teaching me about how to get people to work together. The trick to this game is that working towards your own ends will quickly result in your demise…try it out if you get a chance. It's definitely not the flashiest game you can spend your time on but I think it still holds up today. (Davis, 2014)

Unpacking Davis' short post reveals a number of interesting aspects about his perception of *Gold-Dust Island*. He mentions "educational software…and this game", potentially suggesting that educational software and games that are considered enjoyable are still perceived as being mutually exclusive. On being asked about Jacaranda's back catalogue and the perception of gaming in schools, Smith suggests:

> I don't think we ever found a problem. We were always pretty clear that the software had to have a clear educational objective, but that it didn't matter much if this was achieved via a "gamelike" activity.

When discussing educational software, Luckett clearly supports Smith's position that Jacaranda intended to produce fun software that facilitated learning:

> It's always been the case that the games stuff, your Maths Circuses and Land of Ums and things, were in primary school and high school was much too important for that. [Schools] have Word and Excel [productivity software]… but even now some high schools are still buying Maths Circus because it works for the same reason it works in primary schools. The idea is to give kids an interest in maths in that case without them knowing that they're doing maths. It's an application.

Davis notes that the "game ended up teaching me about how to get people to work together", indicating an enduring legacy of learning that goes beyond subject or informational content, which supports the original multifaceted learning aims of the title. Davis' post notes that the game isn't the "flashiest" and suggests "It looks and behaves like it was written in BASIC, and I'd love to pore through the source code one day." Despite the relatively simplicity of the BASIC programming underpinning the various ports of the game, and thus not pushing any of the 8-bit systems graphically or sonically, the underlying game mechanics and multi-player cooperative gameplay still has relevance over thirty years later. This adds credence to Greygum Software continuing to release some of Jacaranda's titles for contemporary computing hardware. Conversely, a user on stardot.org.uk, an online forum for Acorn microcomputers (including the BBC), is of the opinion that *Gold-Dust Island* is "A good idea but somewhat lacking in programming skill." (*Stardot.Org.UK Forums [Educational Archive not in STH]*, 2013). This poster has made numerous posts on archival BBC Acorn software, games, and paratexts, so was likely comparing it to the best software on Acorn. The user suggests "Now if Sherston [Software] had produced this ...", indicating that a more prominent and experienced BBC Acorn developer may have improved the aesthetics to match the gameplay.

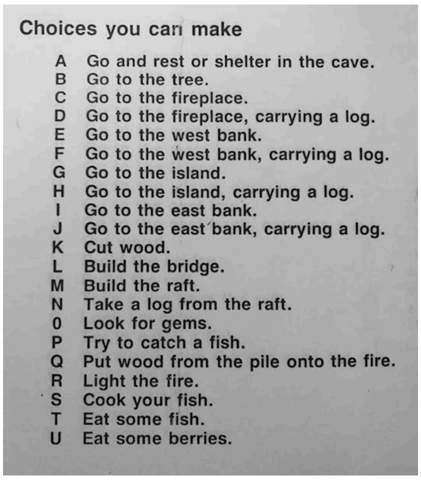
Davis' post supports the idea that despite the game's relative simplicity, the design and gameplay draws the player in. Because the in-game trading system relies on altruism rather than currency, it is unsurprising that Davis mentions that students abused this system. Despite this, settling into defined roles, teamwork, and learning life skills shone through and remain lingering memories of the game. It is worth noting that the one comment on the blowthecartridge post is where 'Tim' mentions that *Gold-Dust Island* "reminded me of this great old game", referring to *Raft-Away River*, the next Jacaranda software title I will explore.

5.4.3 Raft-Away River (1984)

5.4.3.1 The Game

Capitalising on the release of *Gold-Dust Island*, Jacaranda released *Raft-Away River.* Published on the four major 8-bit educational computing platforms of the time, the game was programmed by David L. Smith (BBC Micro, original version), Gerald Wluka (Apple II), Philip O'Carroll (Commodore 64), and Gerald Preston (Microbee). High demand by schools warranted reissues of the package during 1985 ('Raft-Away River documentation', 1985).

The simulation is based on a 'disaster' scenario similar to that of *Gold-Dust Island*. The game world of *Raft-Away River* was again a top down, turn-based survival game emphasising multiplayer collaboration. Two to six players could play the game together, with one of four difficulty levels (a new feature implemented in this package). The goal of players was to build a raft to collect and use resources to escape an island before a river flooded. An expanded roster of up to twenty-one 'real world' tasks could be undertaken, entered with the keys 'A' to 'U' (Figure 5.8), with a number of basic tools (rope, matches, axe) allocated randomly to each player. Trading of items was not a feature of this game, so each player's options were ultimately limited depending on their initial provision of tools. This framed each player with a set of skills and limitations, necessitating negotiation and teamwork based on interpretation of the game rules and situations. The graphics remained serviceable, representing the player characters and the game world with simple sprites (Figure 5.9). Feedback via sounds or on-screen prompts were similarly modest but functional. The gameplay requires reading of text and interpretation of the environment, with graphical cues indicating successful completion of tasks. Cutting wood from trees, lighting a fire, catching a fish, and building a raft or bridge are represented with simple but clear on-screen animations.


Choices you can make

A Go and rest or shelter in the cave.
B Go to the tree.
C Go to the fireplace.
D Go to the fireplace, carrying a log.
E Go to the west bank.
F Go to the west bank, carrying a log.
G Go to the island.
H Go to the island, carrying a log.
I Go to the east bank.
J Go to the east bank, carrying a log.
K Cut wood.
L Build the bridge.
M Build the raft.
N Take a log from the raft.
0 Look for gems.
P Try to catch a fish.
Q Put wood from the pile onto the fire.
R Light the fire.
S Cook your fish.
T Eat some fish.
U Eat some berries.

Figure 5. 19*: Player optional tasks in Raft-Away River.*

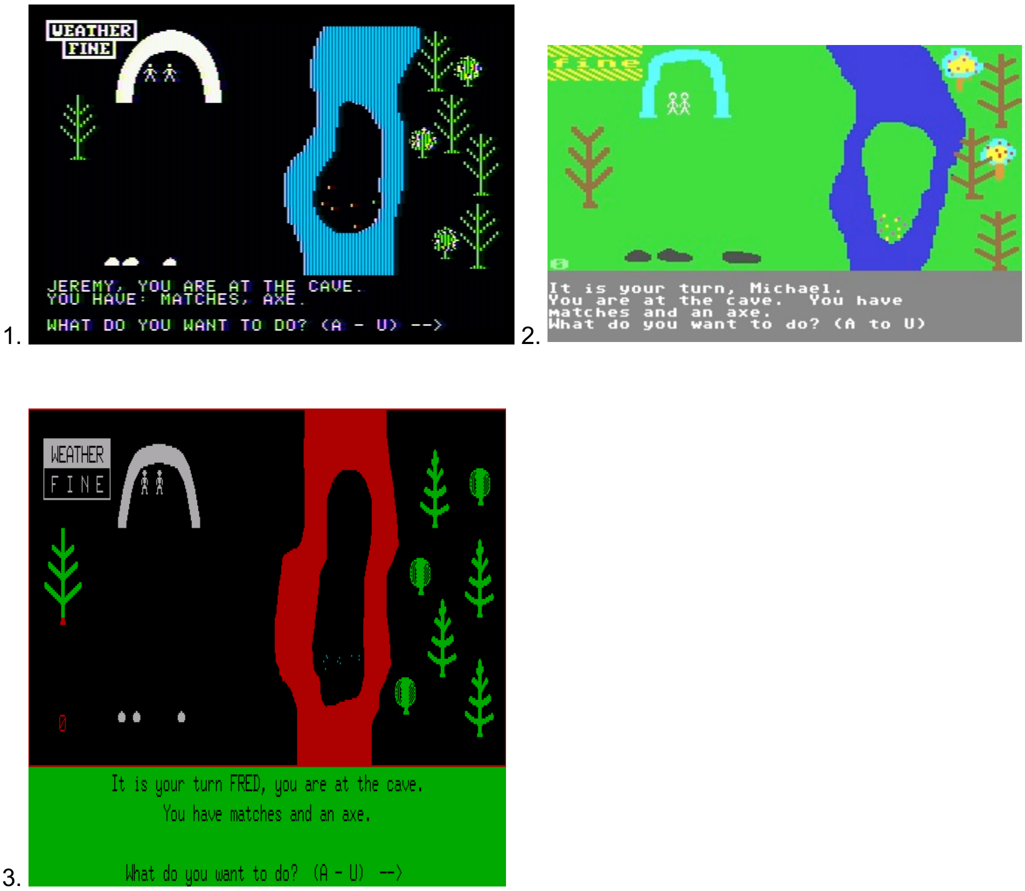


Figure 5. 20*: Raft-Away River – user interface (1)Apple II, (2)Commodore 64, (3)BBC Micro*.

5.4.3.2 Development and Rationale

Whereas Jacaranda initial software titles were related to their social studies and mapping books, they also published educational books addressing reading and language development. Collins subsequently offered Hood and Smith the opportunity to develop software related to developing language skills. Co-designed by Hood and Smith, Hood describes the planning of the game as follows:

> *Raft-Away River*...now David [Smith] had the hang of it [educational software design and programming], David and I spent a merry weekend in…Melbourne, wining and dining…at the insistence of John Collins that we spent a lot of money. And we worked [the game design] out…it's a concept based on *Gold-Dust Island*, but the kids [players] are on a camping trip…and they have difficulties and all sorts of problems.

Rather than create an application based around more typical spelling or cloze reading exercises, *Raft-Away River* is an adventure / survival simulation title that shares aesthetic and gameplay elements with *Gold-Dust Island.* Building on the strengths of previous releases, and an increasing familiarity of the established Jacaranda game style, *Raft-Away River* was designed to address students' reading comprehension, communication, cooperative behaviour, and problem-solving skills. The rationale behind this approach exemplifies and plays to the strengths of early Jacaranda titles; namely open-ended software that promotes semi-structured and student-centred collaboration and negotiation. The language comprehension development occurs transparently and informally to students. Hood reflects upon the implementation and aims of these early educational simulation games, as an extension beyond their book or board game origins:

> The aim was to make it a resource to make teaching better…to stimulate a different type of activity that you couldn’t do without it…you had a range of opportunities to give kids more experiences…All of them are simulations, none of them are rote…there’s no right or wrong way that is in the program, the kids come to the right or wrong way by managing it.

The software design philosophy and implementation during the Jacaranda Wiley era, and continuing through to the present with Greygum Software, did not resemble corporate systems analysis, design, and development processes. This can be attributed to several interrelated factors. The software development teams were small, with a single programmer working on one version of the software for each target computer platform. Programming cycles were short, with Hood suggesting most titles were completed between one to two months. The software itself, especially that which was created in BASIC, was relatively simple to create for programmers. When queried about the design process Smith reflects on how software development was informal and non-hierarchical between team members rather than following a specific product design lifecycle with rigidly defined roles:

> What's that?...We have some sort of idea of what a package is going to be about, but then it gets developed direct in the programming environment…We have never gone for the usual splitting of the task into some sort of system analysts or designers who dream up what will happen and then lowly programmers who cut the code as instructed, and are I think [the software is] much the better for it.

Mitchell highlights that team members contributed to software design and creation in a way that played on each employee's preferred computer platform and their perceived strengths in the area of software development:

> I normally coded for BBCs because I found that was the best thing to write on in terms of ease of use and just getting ideas on your screen quickly. And then David [Smith] would sort of look at it and say, "Leave it with me." And he would go off and completely rewrite it in a completely different direction or most of the time he'd actually have the idea. I'd say things like, "We've got these mapping programs. Can we incorporate mapping and language or maybe a bit of maths into the one thing, so we could make it the centre of an overall theme in the classroom?" And he would come back again with something that was mathematically correct and efficient and then it was up to me and other people to look at it and put some colour...David's a brilliant programmer and comes up with some amazing techniques and it just needed a little bit of softening around the edges occasionally. That's why we worked well together.

Wluka's approach was similarly practical and focused on his strengths when porting existing software, in this case basing his Apple II port on Smith's BBC Micro version. Like O'Carroll, Wluka was provided access to source code, but he found the BBC Micro BASIC was not directly transferrable to the Apple II. There was no handover of design documents or formal meetings to explain the software. Considered an experienced coder of the Apple II, even as a teenager, he was entrusted to develop the port as he saw fit. Based on the software running on the BBC and the provided source code, he deconstructed the game logic then wrote it for the Apple II. For *Raft-Away River* (and *Goldfields*) "I could just understand how the game worked and what the constraints were or the rules, just by looking at his code, and then I'd go ahead and implement it on the Apple II. That was all [coded in] BASIC." This indicative of both Wluka's skill for logic and coding that belied his age, but also the relative simplicity of the software he was porting. Minimal support was offered nor required, although graphics were supplied by Jacaranda. He was able to complete the work in forty to fifty hours based on observation and testing of the existing title.

> I could see it on the computers at school and I, actually…literally just saw how it worked and I wrote it from scratch on the Apple II… I could see his code but it didn't really help me, because the Apple II was that different. I could just understand how the game worked and what the constraints were or the rules, just by looking at his code, and then I'd go ahead and implement it on the Apple II.

The development of *Raft-Away River* across multiple platforms was indicative of Jacaranda's focus on producing genuinely useful educational software. There is a sense of efficiency and outcome driven expediency in their approaches. Smith suggests other companies "seemed to do little but publish glossy photos of themselves in trade journals, make business plans and projections, and look around to see what the others were doing, instead of actually getting on with writing software." Despite the corporate and profit driven nature of a Jacaranda Wiley as a company, they dispensed with supposed norms and development cycles of software companies while simultaneous creating software that addressed specific educational concerns in an innovative and expedient way.

5.4.3.3 Software Packaging and Documentation

Raft-Away River's accompanying package and documentation were similar in style but notably leaner than the *Gold-Dust Island* bundle. There was subtle refinement of the information presented to their target audience of students and teachers. I suggest that this is likely in response to users' familiarity and confidence with using computers in the classroom. Hood describes receiving fan-mail from software users, although there is no direct indication that Jacaranda solicited feedback on their previous works. Whether intentional or not, this subtle evolution of paratexts is noteworthy.

The four-paragraph spiel on the rear of the *Raft-Away River* packaging dispenses with the 'About The Author' notes of *Gold-Dust Island*, instead focussing on the intended aims and more description of the software itself (Figure 5.10). The text remains persuasive, outlining (in order) educational outcomes, gameplay, rationale, and benefits. The text directly emphasises the gameplay in paragraph two, which was minimised in *Gold-Dust Island*, or in paragraph three in the *Moving Into Maps*. This demonstrates confidence in the game itself, while also indicating acceptance of gameplay as a legitimate use of computers use in the school classroom. Rather than tying the simulation into a social studies or mapping unit like previous Jacaranda works, the flexibility of the software is highlighted in paragraph four. The multiple difficult levels make the software accessible to a wider age range. No specific subjects are suggested, instead it is highlighted that *Raft-Away River* can be used in the context of any teaching unit "wherever an emphasis is being placed on interpersonal communication, leadership, cooperative behaviour, strategy planning and small group productivity." Flexibility of use cases and higher order learning skills are emphasised. There is also admission that *Raft-Away River* is "fun", an admission that learning via computers can be enjoyable.

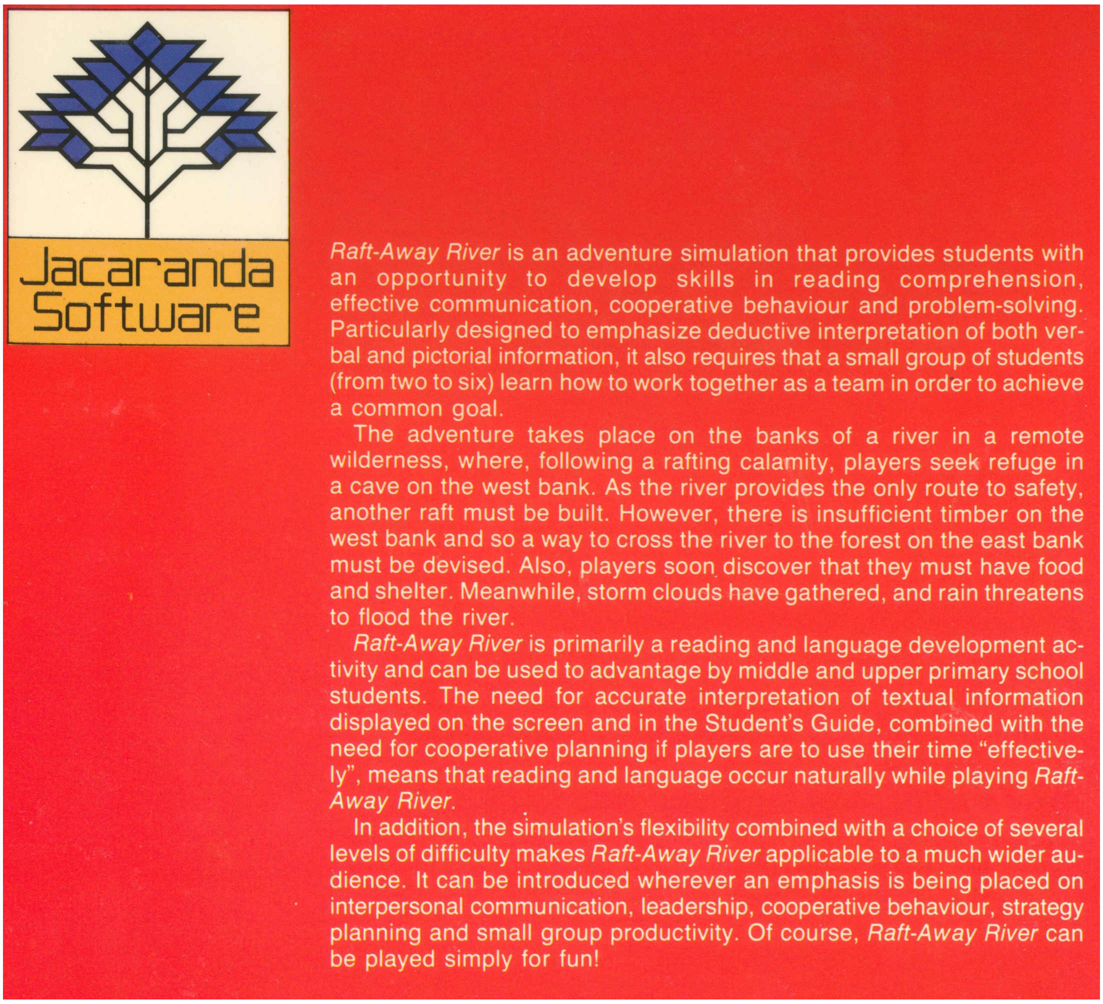


Figure 5. 21: Raft-Away River packaging back (cropped).

The documentation provided with the game is refined from previous titles. The Teacher's Guide emphasises that the "greatest benefit will come if the students discover the need to take all aspects of the situation into consideration, and then make sensible and realistic decisions."[24] Restrictions, limitations inherent to the

---

[24] Emphasis noted from the Teacher's Guide.

game, and the in-game world rules are designed for student discovery, with only brief gameplay suggestions made. Rationale behind the software's purpose is made, with emphasis placed on teamwork, students discussing impending decisions, and taking time to complete the game (presumably rather than prematurely ending it due to rigid lesson schedules). There is a clear message being conveyed to the teacher that beyond an introduction to the gameplay elements and how to use the software, the best uses of the game are in student-centric rather than teacher-driven environments.

The *Raft-Away River* Student's Guide is similarly brief, with less than eight pages devoted to story, gameplay options, and structured step-by-step gameplay suggestions (compared to *Gold-Dust Island's* fifteen pages). Diagrams are not provided, indicating the gameplay is more self-explanatory compared to other titles and, perhaps, familiarity with other similar titles. Multi-platform loading instructions are also present in this guide, suggesting students would be likely to load and the game rather than the teacher. Multiple copies of the list of commands were included for distribution to students, negating the requirement to remember over twenty commands. This refinement and reduction of documentation implies that students were capable of initialising, playing, and learning the intended skills and processes from the simulation with minimal instruction. As computers became more prevalent in schools and homes, with teachers and students becoming more familiar with loading and operating software, more verbose and prescriptive instructional steps were not needed.

5.4.3.3 Reflections on Raft-Away River

Reflections on *Raft-Away River* are more numerous online than for *Gold-Dust Island*, indicating a degree of fond memories for this title. The more noted memories are likely due to the later release of this title, resulting in wider distribution and uptake by more schools having computer hardware at this time. Discussion and musings on this, and indeed most educational titles (games, simulations, or otherwise), are generally less common compared with games of the era. For those who owned microcomputers, access and use would have been far greater than at schools and, consequently, more remembrance of gameplay experiences shared. The Play It Again website archive (Stuckey *et al.*, 2013b) includes a number of positive reflections. 'Steven' recognises the importance of the presence of an Australian produced game at school. Considering *Raft-Away River* was a relatively simple game coded in BASIC and produced on a relatively meagre budget, the implication is that at his school it was held in some regard:

> *Raft-Away River* and *Where in the World is Carmen Sandiego* were the only two games that were ever presented to us in school...In retrospect, it is quite an achievement for the small local publisher given that Carmen Sandiego was produced by a significantly larger company.

'Nicholas' notes that he lost many lunchtimes to playing the game despite never finishing it. This is evidentiary of the enjoyment students had despite (or perhaps unaware) of the learning aims of the simulation. He also acknowledges the rudimentary crafting aspect of game, long before Minecraft popularised the concept with school age children, as well as the co-operative survival feature many years before multiplayer online games became commonplace. Alan Laughton of the Microbee Software Preservation Project (*Microbee Software Preservation Project*

(MSPP), 2016) recalls the "Microbee version was very popular in schools, though their choice of colour for text [orange on a red background] made it difficult to read."

Posts on the Australian Whirlpool Forums (*80s Desert Island game... anyone?*, 2011) and Reddit (*80's computer game. Had to cross a river, build raft in a set amount of turns*, 2009) have been made asking for the title and platform of this particular game based on descriptions of the gameplay. This is demonstrative of enduring memory of classroom experiences with the *Raft-Away River*. Despite the title being forgotten by a number of people, the graphics and collaborative gameplay mechanics are remembered, notwithstanding the over thirty years after the game was released. Other forum and Reddit users answered these queries with the correct titles; although at least one suggestion was made that the game could be *The Island of Jara Tava* by Satchel Sofware. This indicates the similarity of the games as well as the popularity and prevalence of simulation styled open-ended Australian educational games in schools in the 1980s. On the Whirlpool forums, the questions elicited positive memories of people playing the game, although reminiscing sometimes brings tacit acceptance of the limitations of the software.

> Oh wow I remember that game now. I remember some things like if you didn't build your raft in time, you'd be swept away by a flood. Is that the same game? Sorry I can't remember the name though...
>
> Yes! I remember this! You had to build a boat to escape the island, and you could also mine for gold to bring back from the island if you survived. I have no idea what it was called though. Sadly, I'm sure that just like Magnum P.I., it is nowhere near as entertaining now as I remember it to be...

OMG I used to play that on the old Microbee, no idea on the name...

You had to collect logs, fish, water – it was very simple, but fun. I used to play it at school on an old Apple IIe or something....dammit, OK, it's ancient, but would love to play it again. Any ideas?

There were a number of less positive comments including “If [I] recall you died quite easily if [you] moved too far, eventually we got frustrated by it” on Whirlpool and “Wow I just played the game again, I didn't remember how shit it was what is wrong with my memory?” These reflections demonstrate that the initial positive impressions felt by some people for the game has given way to a realisation that the game mechanics, graphics, and sounds have long been surpassed. These examples of software indicate a historical legacy and snapshot of a particular period in the 1980s where small teams could create impactful educational titles, albeit tied to the technical limitations of the era.

A request in 2002 on the dosgames.com forum (*Raft Away River*, 2003) mistakenly requesting a Macintosh copy of the game[25] highlights the relative dearth of fan-based archives of educational software. A number of forum users queried where to obtain disk images of *Raft-Away River*. The only online copy retrieved circa 2002 was the Commodore 64 port freely available via gamebase64.com. Historical educational titles exist on the margins compared to mainstream retro game software. Because the majority of students predominately played educational games at school and generally did not own their own copy, the volume of surviving copies for fan-

[25] The request may have been referring to the Apple II version not the Apple Macintosh.

based archival is limited. I have received numerous original copies of Jacaranda's works, including *Raft-Away River*, from Jacaranda's programmers, which has greatly supplemented the fan-archived materials. Moving forward, opportunities exist for historical Australian educational software to be archived, displayed, and discussed online in future in order to develop the historiography of these titles. Luckett agrees with the value of preserving Jacaranda's legacy software:

> …that there's a huge culture out there of people that like to run these old titles on their machines and should be released out to public domain for them. So we have to make that decision. I reckon it would be great because they're just sitting in boxes rotting now.

The three Jacaranda Software titles discussed thus far are indicative of the somewhat experimental and 'playful' nature of educational software development during the early phase of Jacaranda's lifespan. In the next section I will explore a number of their later titles that contrast with the style of these open-ended game.

5.4.4 Wordsmith (1985)

Although there doesn't appear to be a conscious shift away from Jacaranda publishing simulation and collaborative educational titles with game-like mechanics, they branched into producing a number of utility and subject specific educational titles. After designing and contributing to the aforementioned titles, Hood moved on from Jacaranda Software in 1984 within twelve months of the foundation of their software division. During 1985, Smith wrote a school student centric word processor called *Wordsmith: a junior word-processor*. This title was originally programmed and produced for the BBC Microcomputer, with Wluka porting it to the Apple II, and a Commodore 64 version was also produced (programmed by Ian Treacy). Mitchell notes that he created a functional, albeit rudimentary, word processor in 1980 prior to the formation of Jacaranda Software with:

> …a genuine World War II teletype, which I installed in my classroom and the kids were actually writing stories and then we'd insert their names in various places in the stories and…this is a word processor running in 8 [kilobytes of] memory…I didn't think it was educationally very useful but it certainly kept some of [the students] off the streets at lunchtimes… Luckily I had a good ute. Taking it home was always a bit of a problem. What did it weigh, about 30 kilos? This is right at the computers in schools kind of moment when it all starts to happen.

Wordsmith was created by Smith several years later, by which time microcomputers were becoming more common in the classroom. It was designed as a word processor that would be user-friendly enough for students to use in an era where functions were limited and they were not intuitive. Mitchell recalls:

> …at a time when word processors were very user unfriendly …[*Wordsmith*] was written with incredible system limitations on it…the thing that sold it, I think, was that David had organised to print everything in letters…on a dot matrix printer. So he'd written a printer driver to take text that was on your screen and write it on pages that big, which was ideal for younger kids, in a font that was similar to the sort of things they'd find in their readers. And that worked well. I used that in the classroom when it was in its trial stages.

Mitchell's quote shows awareness of and responsiveness to the target audience by including features that were useful for school-age students. Although *Wordsmith* was functionally rudimentary compared to modern word processing titles, it included numerous features aimed specifically for use "in primary schools, at home with children learning to read or write, by others with reading or vision difficulties, or as an introduction to word-processing" ('Wordsmith manual', 1985). Of note, it could display upper- and lower-case letters, had automatic line wrapping of text, word searching, printing, saving, and colour coding to indicate text entry (green) or text editing modes (red). The entire manual for the program is less than twenty pages, which allowed students to become familiar with its functions and operation quickly.

In order for Wluka to port *Wordsmith* from the BBC to the Apple II, he was presented with a number of technical challenges that required innovative solutions. This is especially notable in that he had to problem solve on the fly to achieve the functionality on the Apple II that Smith had created for the BBC hardware. Unlike many of the previously discussed simulation and game-based programs, Wluka decided to create the software using machine code instead of basic. Although this produced software that ran natively, machine code was more difficult and involved to write software in:

> *Wordsmith* had machine code, and that was a technically more interesting, because the BBC Micro let you have big text and do graphics things a lot quicker than the Apple II. So, in order to do what I needed to do for *Wordsmith* I actually had to create those capabilities in the Apple II. And that was, if you think the Apple screen I think was… a 40 by 25 characters. And for *Wordsmith* they

> wanted to be double height characters so it'd only be, so for little kids they'd get big text on the screen, and as you typed, where your typing stayed in the middle of the screen so all that text moved around as you typed. And that was difficult to do on the Apple II. So that was interesting for me, technically. And I made tools.

Predominantly working independently on the *Wordsmith* port, Wluka created the title from scratch to mimic what he saw on the BBC Microcomputer. He believed this was more efficient and saved time during the creation and testing process:

> "There was a version on the BBC Micro, and I saw that and played with that and then said "Okay, you want to do that on the Apple? Okay, I can do that"… I didn't like take his code and "on the Apple II you say this and the BBC Micro say that"… the structure of the languages were different enough and the way the computers worked were different enough, that that would not have saved me very much time, if any. It probably would have cost me more time because I would have had to learn the details of the BBC Micro."

Wluka was supported by Jacaranda's senior staff when assistance was required, although he did not have any dealings other than with Smith. According to Wluka, he doesn't think he ever met anyone from Jacarada Software other than David Smith. Regarding the double height font which was intended to make text easier to read, Wluka created a tool for Jacaranda to supply the fonts required, as he describes:

> There [were] no fonts to do that, so I actually made a little program that I'd sent to Jacaranda Wiley and they had their graphics people design the fonts that they wanted in *Wordsmith*… There wasn't much

> dialogue. It's like "Yeah, I can do that" and "Yeah, you can pay me"…I would debug everything myself and test it. I don't remember any [bug] reports saying "Hey, this doesn't work" or so on, because the programs were relatively simple back then…[*Wordsmith* is] basically a kiddy typewriter, or a word processor. If you don't want a full function word processor and a kid to sit in front of the computer…I thought it was stupid. Stupid idea. Kids can learn to type on a computer. They can see they type, while the text comes up, backspace, they could use a real word processor."

Interestingly, although he was happy to be paid as a contractor to undertake the challenge of porting the software, as a school student himself at the time he did not necessarily agree with the simplified and somewhat cut-down word processing package he was creating.

Similar to Satchel Software, Jacaranda Software evolved steadily throughout the 1980s and early 1990s in order to address the shifting landscape of the educational software industry. Different target platforms, software diversification and complexity, and evolving expectations for documentation and examples of classroom use are notable features of Jacaranda's software oeuvre were noted in the selected cases of software explored in this chapter. Further analysis of the source materials from Satchel Software and Jacaranda Software is offered in Chapter 6

5.5 Jacaranda's legacy and the emergence of Greygum Software

Despite early commercial success, earning in excess of one million dollars in sales by 1984, Jacaranda Wiley's software division was wound down in 1992. Smith, Luckett, and Mitchell formed Greygum Software after purchasing all rights, equipment, and software from Jacaranda. This allowed sales, support, and development of new software to continue through to the present. Mitchell recalls:

> …in 1992 when we got the word that we were closing down…We bought all the rights to all the Jacaranda material. We bought all the stock and most of the equipment as well and just shifted it into my basement…

Just prior to Jacaranda Software shutting down, they were working on a title called *Maths Circus* (1992). Ultimately, it was not published by Jacaranda, with Greygum releasing it. This was fortuitous for Greygum, as it is still currently being sold.

> *Maths Circus* was the last thing we did at Jacaranda. It was just about to go into production at Jacaranda in 1992…And *Maths Circus* was probably the best thing that David [Smith] had come up with up to that time. It's a great program and it sold like it should have. It's still selling well in its sixth incarnation. But basically for what it cost us to buy the rights and the stock and the equipment from Jacaranda Wiley, that was all paid back in about six months on the strength of that program. So it was a bit of a loss for them.

What is noteworthy, according to Mitchell in the above interview excerpt, is that the legacy (and continuing sales) of Math Circus and its sequels endure for over twenty-five years. This suggests that whatever directorial, internal, or market forces resulted in the closure of Jacaranda's software, its software was of high quality and profitable. *Maths Circus* is based on decades old educational software yet continues to sell, as do other newer incarnations of Jacaranda titles including *Kraken*, *Crossing the Mountain*, and *Desert Quest*. Despite computers in modern educational settings being predominately used for production of written work or research, the market for educational learning and skill-development software still exists, especially in primary schools. The purchase price of the software may be considered high for an individual copy of a given title, with most over $100 per copy. Conversely, Greygum have continued to use the same value for money model for school site licenses that Jacaranda did in the 1980s, with an entire school being licensed when two copies of any given title are purchased. This provides value for money in the era of monthly software license fees. Beyond the Windows and Macintosh versions of their software, Greygum have ported a number of titles to tablet and smartphone. In addition, they are investigating cloud-based downloads of purchased software, which would remove the need for packaging and postage of the comprehensive packages that they continue to as a legacy from the Jacaranda era.

Luckett currently runs Greygum from his home where "we do all the production and sales and packaging and all that sort of thing, just to keep the thing rolling. It's not fulltime but it's close to it." He notes that the Jacaranda Software name was also purchased, for which they still pay a biennial fee, although the name is used sparingly on their website and promotional material. As of 2015, Smith still programs for Greygum, with Mitchell reflecting "…we try not to ask too much about what David gets up to. He just delivers." Of note is the *Fitzroy Readers* series of software that they write. O'Carroll's company produces a series of over forty reading books based on phonics, and Greygum produces the software to accompany the texts.

Smith also volunteers at the school that O'Carroll's spouse manages. This exemplifies the continued professional relationship between many of the parties who were originally involved with Jacaranda:

> …the reading software has been extremely successful and effective. Fitzroy Reading Scheme, which is what it was originally designed for, was and is a very successful reading program...that school had the highest [NAPLAN] skills in Australia…It's a very small school but they do things well there and David has done a lot of work with them, voluntary and so on.

Regarding the overarching purpose of the variety of software Greygum produce, Luckett continues: "We want to write game stuff that has value for … social value." Smith elaborates on this ethos:

> We have three guidelines or objectives…the first two of which are of equal importance, and the third is decidedly in the back seat. These are:
>
> - that what we produce must be socially valuable
> - that we must have fun producing it
> - and that by and large most projects need to run at a profit.

Greygum's concept of social value, in this instance, can be directly tied to the statement on *Gold-Dust Island's* packaging where Hood attempted to demonstrate the computer's potential to promote meaningful problem-solving activities for students. Perhaps not coincidentally, Greygum only produces a new title every two or three years, with the rest of them considered “rehashed” versions of old titles. This demonstrates the lasting educational benefits of titles created during the 1980s where despite, or perhaps due to, the limitations of microcomputers, the titles needed to be designed to be useful educational tools. The continued positioning of their updated yet decades old software holds credence to the assertion that Jacaranda Software essence and ethos still exists and continues to thrive, albeit with a new name.

5. DISCUSSION AND CONCLUSION

6.1 Introduction

The key findings of the case studies of Angle Park Computing Centre and Satchel Software in Chapter 4 and Jacaranda Software in Chapter 5 are discussed and concluded in this chapter. Additionally, I explore potential future areas of investigation uncovered by this exploration of educational software in the 1980s and early 1990s.

6.2 Discussion and Analysis

Throughout this study numerous themes, commonalities, parallels and distinctions were noted between the two publishers, their development of educational software, and the nature of the titles produced. This discussion is framed around the questions of the emergence of the two publishers, attitudes toward computer software and games in the classroom, patterns in the types of software created and classroom use, branding and visual recognition of each company's titles, software development strategies, and the local political environments behind the APPC, Satchel, and Jacaranda.

Analysing the comments by former programmers in the interviews conducted by myself and Melanie Swalwell, it is striking how consistently the interviewees countered suggestions that there may have been negative attitudes toward computers, educational software, and games during the 1980s and 1990s. Indeed, the opposite seems to be the case; the interviewees provide opinion and evidence that educational software and games were accepted and considered positive factors in the classrooms of the 1980s. The value of learning by play was considered by the programmers to be a valuable classroom experience. In contrast to contemporary concerns about negative aspects of gaming and social media on learning, the interviews present a refreshing view of the acceptance and benefits of educational computing at a time when questions around what could be achieved with computers in schools were just starting to be addressed. The programmers were mindful of the limitations of hardware and software they created and did not claim their titles were necessarily transformational. Outside of educational settings, the lack of discourse surrounding educational software is indicative of a wider degree of indifference to these titles. Selwyn (2011, p 715) argues for a degree of technological pessimism regarding classroom use of computing, stating that "pessimism is a rewarding and heartening position from which to approach education and technology." As he explains (p 716):

> [there is] usefulness of starting from a position that acknowledges the parameters and boundaries of any technological endeavour, and has realistic expectations of the political struggles and conflicts that surround any social change.

I argue that the programmers involved in the creation of software surveyed in this study approached the production of their titles from a positive yet cautious position; perhaps not pessimistic, but realistic. They created software by embracing new microcomputing technologies, but did so from a position of accepting the limitations of the educational benefits, predictability, and efficacy of these technologies.

I consider that software published, and approaches taken, by Satchel and Jacaranda to be aspirational toward furthering educational microcomputing in Australia. This occurred many years before national policies for computer literacies and use in Australian schools were developed. Only recently was there development, introduction, and rollout of an Australian Curriculum in Digital Technologies standards from reception through year 10 (*Australian Curriculum: Digital Technologies*, 2018). This is not only an acceptance that digital literacies are important in the modern world, but is also indicative of forward-thinking approaches by these early educational computing pioneers who could see the educational benefits of computers and, more importantly, the software that was used in classrooms. Any influence on current educational trends of the software created by Satchel and Jacaranda is harder to determine, although I can draw some direct links between the software created in the 1980s and current educational standards. The software surveyed in this study can variously be directly related to the currently prescribed learning areas of Critical and Creative Thinking, Literacy, and Personal and Social Capabilities. Beyond this, programming and database software titles developed and used in Australia that were not directly investigated in this study are intrinsically linked to the areas of Numeracy and Digital Technologies Processes and Production Skills. It is therefore not surprising that some of Greygum Software's titles based on the early work of Jacaranda are still sold and used in Australian schools today. There is potential for further research drawing links and relevance between the software used in Australian schools throughout the 1980s and 1990s and contemporary Digital Technologies standards.

The titles created published by both companies showed patterns in the types of software created for effective classroom use, namely a preference for, but not limited to, open-ended, collaborative and cross-curricula titles. Although there were subject specific titles explored, there was a noted preference by the programmers for creating titles that encouraged exploration and experiential learning. Titles like Satchel's *Mathbooster* were highly customisable and included arcade game like elements to ameliorate the perception of the drill-and-practice like nature of the repetitive maths problems being solved by students. *Granny's Garden*, *Jara-Tava The Isle of Fire*, *Scavenger Hunt*, *Gold-Dust Island*, and *Raft-Away River* were notably similar in a number of ways. These adventure games allowed students, often working in small groups, to work collaboratively, negotiate tasks, tackle interactive problems with feedback, and ensure that students were the focus of the learning activities. All of these features are one of a *constructivist classroom*; a learning environment where students develop knowledge based on experience rather than a more traditional teacher focused didactic classroom. There is evidence in the interview material that programmers at both Satchel and Jacaranda saw the benefits from the flexibility of computer programs to allow for choice and collaboration in problem solving. Looking beyond the software, many of these titles included classroom activity suggestions spanning the breadth of the curriculum. Despite some of the memories collated from online sources suggesting these titles were not quite as objectively good as some people remember them, it remains that these titles are still remembered somewhat fondly from their use in schools decades ago. There is scope for further structured investigation into the reception and use of these titles by surveying students of the day, and also how they were more widely implemented by classroom teachers and used by students. It is clear, however, that the computer was viewed by the programmers as a conduit for the process of learning and not the object of learning itself.

One finding from the study were the professional practices employed regarding documentation, branding, packaging by both publishers. The clarity and quality of the printed instructional materials and comprehensive nature of the documentation produced by both publishers were also a noted. The instructions and suggested classroom activities were comprehensive and useful; the teaching backgrounds of the creators shone through. This was unsurprising due to the professional educational backgrounds of Satchel and Jacaranda's employees. The quality of the visual aspects of the packaging and distinctive branding employed by both publishers show distinct visual languages and consistency. This resulted in the titles published by Satchel Software or Jacaranda Software being clearly distinguishable by their packaging, logos, artwork, and even the style of instructions and other documentation. The branding and package design was consistent across all titles from each publisher throughout their software library, with only subtle changes in graphic design and layout occurring. Even Satchel's educational computing activities booklets distributed to schools (sans software) used consistent designs. This echoes design language of books and book series targeted for schools during this time. I surmise that this consistency was targeted at schools and educators, and possibly even students, in order to develop brand recognition and legitimacy during the emergence of software as a teaching and learning tool. I perceive a conviction by the companies that they saw the software packages as a valuable and positive educational resource that deserved to be packaged and marketed professionally.

Software development cycles across both companies during this period were also similar; teams were small, the software design and development process was flexible, and titles were created in a relatively short time frame (generally in the space of weeks). The rudimentary nature of graphics, sound, and game logic employed by the software titles were dictated by the limited speed and RAM of microcomputers used in the 1980s. This contributed to the speed of development and the ability of staff at both companies to collaborate on titles and share roles in game creation based on each team member's strengths and familiarity with each particular brand of microcomputer. The abundance of titles produced by Satchel and Jacaranda in their early years is a testament to the relative ease and speed of software creation on a range of 8-bit computers. As the titles produced by both companies were ultimately targeted at 16-bit microcomputers that were more complicated to program, and the software titles increased in complexity and expanded in scope toward application and communications software (especially noted in the case of Satchel Software), the number of contributors for each title and the time taken to create them increased. This suggests that extended development periods and more personnel required to create each title, in addition to voluminous documentation, would have resulted in increased development costs. Although there were no sales figures or financial statements obtained for either publisher, a logical conclusion to draw is that both were wound down due to market realities of the early 1990s. Even through there were less target 16-bit microcomputer platforms by the early 1990s than during the 8-bit era, software creation, publishing, and marketing had taken the form of an increasingly expensive and corporatised space. My research did not reveal whether any other local or overseas publishers encroached into the spaces occupied by Satchel or Jacaranda, or whether the market shifted incrementally away from their software over a period of time. Further investigation into the evolving educational computing market in Australia through the 1990s could shed light on the shift from locally produced titles such as those by the two publishers focused on in this study toward the Internet connected era of the late 1990s through the present.

The relative longevity of Satchel Software and Jacaranda Software documented in this thesis is indicative of their ability to make an impact on the geographically dispersed, uneven, and small emerging Australian education software market spread across multiple microcomputer platforms. Although each publisher had a different path of formation, there was government and departmental backing for their endeavours, albeit to varying degrees. There were highly contrasting political environments during the emergence of the APCC in South Australia and Jacaranda Software in Queensland. South Australia was governed by a noted progressive party throughout the 1970s through the early 1990s (excepting a period from 1979-1982 when a conservative government was in power), with Queensland predominately having conservative governments throughout this period. Support of the establishment and continual operation of the APCC by the South Australian government demonstrates an appreciation of potential benefits of computing in education. This carried through with financial and governmental support of the APCC and Satchel Software until the early 1990s. Jacaranda Software was a private company, and although no physical contractual documentation was obtained, interviews suggest the Queensland Department of Education supported their work from the outset. Additionally, at least one title programmed by Jacaranda was published by the Queensland Department of Education who is the sole copyright holder of the game (*Pieces of eight*, 2014). In these contrasting environments, programmers were able to experiment with software development based on their own educational experience with little policy interference.

The 1980s was a decade where small publishers with a modest number of programmers could succeed and remain viable while producing a range of software titles for use in Australian schools. Although gauging their economic success or number of software sales are outside the scope of this study, the range of titles released across several years paints both Satchel Software and Jacaranda Software as fruitful and impactful publishers. The evolution of the computer hardware market in the early 1990s resulted in winding down these two companies, positioning them as features of their time. However, both publishers have legacies that continue to the present. Satchel's dissolution and absorption of staff into the South Australian Education Department, resulted in the continued development of Hodgson's library software *Bookmark*, and a number of staff capable of programming were deployed to a variety of departmental development roles. Jacaranda Software's winding down was in practice a name change; many of the philosophies and individuals who drove Jacaranda's success continue to do so, directly or indirectly, with Greygum Software.

6.3 Conclusion

This thesis aimed to address the key questions of "*How did early Australian educational software development organisations during the 1980s emerge to produce quality software and games for schools? What were the intended educational outcomes and methodological rationales intended for educational software produced during the genesis of educational microcomputing in Australia in the 1980s?*" This is an area of historical research that has little previous attention in the literature. Although the broad field of research under the banner of Game Studies continues to develop, including the historiography of retro video games worldwide and within Australia, this project addresses the dearth of published work about educational software publishers, their games, and the programmers behind these titles. Educational software is worthy of investigation due to the temporal nature of software storage media that degrades over time, the age of the individuals involved with implementation of educational microcomputing in the Australian schools, and its relationship with contemporary Digital Technologies education. Analysis of interviews with software creators, most of whom were also teachers, from two organisations, exploring their software and paratexts from a perspective of scope and intent, as well as touching on reception and success of these products were the primary approaches toward answering these questions.

The insights provided by each of these participants present an emerging picture of an educational software market that was largely experimental and forging new ground. Software design was informed by the educational and methodological views of the creators, but was also limited by hardware constraints. Some titles had designs based on the influence of existing software and games, but other titles established their own distinguishing features based on intended educational uses. Limitations of access to hardware by schools during the mid-1980s were also factored into the operations of the software published by the two companies, which resulted in many titles that afforded and encouraged collaboration between students, in addition to cross-curricular use cases of software. Despite the now archaic nature of the 8 and 16-bit platforms, much of the software surveyed appeared to popular and educationally beneficial in the eyes of the creators and from limited retrospective recollections by now adult players of these games whilst in schools. Although both Satchel Software and Jacaranda Software were reportedly successful, both in terms of longevity and pervasiveness into the commercial Australian educational computing software markets, both were wound down as software imprints in the early 1990s as the educational software (and software market in general) matured toward more commercially driven enterprises worldwide. That both companies were wound down should not be seen as a failure, but more of a sign of the evolving and emerging educational computing market, and more broadly the nature of software development maturing and taking on the form of increasingly corporate enterprises in the years prior to the widespread emergence of the Internet.

This study moves toward developing strategies toward structured approaches to analysing and discussing the development of educational software in Australian schools. My investigations were based on the triangulation of interviews, software analysis, and reflecting on related paratextual materials. This provided both personal reflections and historical memories relating to software development in addition to objective discussions about software and related documentation. This helped develop the stories of the two company's evolution in addition to the varying influences and interplay that the individual programmers had on the direction that the companies took. There is potential for continued refinement and structure to the methods in which the history of educational software can be recorded, studied, and evaluated. This thesis has used two case studies to present a history of two software companies, their software, and the production processes undertaken by the individuals or small groups collaborating on a number of titles throughout each company's history. I have demonstrated that documenting the unique stories of educational software production during the emergence of computers in education sheds light on an important aspect of the history of educational computing in Australia that has been neglected in the existing literature. Additionally, this research is potentially interdisciplinary as it intersects with numerous fields including educational policies relating to implementation of computing in schools, teaching practice, pedagogy, as well as Game Studies. The case studies presented contribute to knowledge in the area of historical educational computing by including new oral histories of early software and game programmers, analysis of educational software that has hitherto not been recorded, in addition to their related paratexts. This assisted in documenting and analysing the history of the emergence of educational software in Australia as a facet of the history of emergent educational technologies of the 1970s through 1990s.

More studies and historical analysis in this area are warranted to fill in this history, as there are many unexplored areas of educational computing and software production in Australia during this period of computational and technological growth.

6.4 Suggestions for Future Research

The research undertaken during the production of this thesis has identified numerous potential areas of related study for further investigation, analysis, and documentation. Developing a complete historiography and software catalogue of both Satchel and Jacaranda's software titles would be a worthy undertaking to comprehensively document both company's decade long endeavours. My research to initiate the respective historiographies of both companies was limited to a hand-picked selection of software and interviews with software designers who were predominately also teachers to initiate the respective historiographies of both companies. Whilst I have presented examples of how each company evolved and how their programmers developed, refined, and improved their titles over time, it is an initial snapshot of their work. Future research could measure the success of each company's titles (either by units sold, money earned, or educationally) which was a task beyond the scope of this study. Furthermore, interview-based studies could include other former staff, teachers, and developers to develop a richer understanding of the educational and computing landscape during the operational lifespans of both companies. This a timely and sensitive aspect of oral histories involving computing companies that ceased operation nearly thirty years ago and whose former staff are now approaching old age. Several developers in South Australia from the APCC and Satchel Software are still potentially accessible for interviews, as are a number of other Jacaranda staff throughout Australia. In parallel to this, it would be timely to identify and acquire any software that has not yet been archived and preserved from each company's oeuvre to avoid it being lost due to the passage of time and likely degradation of the various media they are stored on. Further research and analysis of the various newsletters and periodicals produced by the APCC is potentially valuable from a historical and educational use case perspective. Beyond the preservation of historical written materials, I suspect that relevance and connections to contemporary teaching practices and approaches to educational computing could be drawn.

Beyond these two companies, research is needed to construct a richer historiography of other software development and microcomputer use in schools during the 1970s-1990s in Australia. Investigation into the variety of Australian educational software development could include history of works by other commercial entities, government agencies, teacher developed titles, and other homebrew software intended for schools. This includes software and educational computing support materials that emerged from the TASAWA consortium were not revealed during this research. Further investigation into the development of the educational computing landscape other states of Australia and the software created and used across the various states are worth exploring. An interesting case study would be an investigation into the evolution of the *First Fleet* convict software developed in Tasmania in the 1980s, the online descendant of which is still available for student use today (Wills, 1999). Looking beyond Australia, the themes explored in this study could be applied and expanded to other educational jurisdictions and countries. Such exploration could enrich knowledge about the evolution of educational computing through drawing parallels and recurring themes, revealing distinctions, and identifying unique local flavours in different countries and regions.

Further questions remain as to the efficacy of early educational software design concepts and their continued use in schools. Investigations could be conducted into how older software titles such as those explored in this study could be used as contemporary learning tools to develop computing skills alongside creativity, thinking, and communication skills. Comparisons could be made with modern software, either locally installed or web-based, that are currently used by students. Perhaps there are justifications for some of the types of software used in the 1980s and 1990s to again come to the fore as teaching and learning tools.

APPENDICES

Appendix 1: Interview Information Sheet

Marcus Schmerl
Screen and Media Department
School of Humanities & Creative Arts
GPO Box 2100
Adelaide SA 5001
Tel: 0423 924 687
marcus.schmerl@flinders.edu.au
http://www.flinders.edu.au/people/marcus.schmerl
CRICOS Provider No. 00114A

Information Sheet
"A History of Creative Computing in Australasia"

Lead Researcher: Dr Melanie Swalwell, Screen & Media Department, School of Humanities & Creative Arts, Flinders University.

Student Researcher: Marcus Schmerl, PhD Candidate, Screen & Media Department, School of Humanities & Creative Arts, Flinders University.

Mr Schmerl is undertaking PhD studies in "Educational Computing and Software in Australia: A History of Australian Educational Software Development for Australian Schools, 1970 – 1990."

You have been approached to participate in this research on the grounds of your involvement with educational computing during this time. Your participation is voluntary, and you may withdraw your participation at any stage prior to data collection and analysis being completed.

If you agree to participate you will be asked some interview questions regarding your involvement in, and knowledge about, aspects of educational computing in Australia, and possibly answering some follow up questions. It is anticipated that this will not take in excess of 1 hour. With your permission, the interview will be audio or video recorded.

The results of the research will be used to inform a history of educational computing in Australia. This will be presented in a range of formats, including Mr Schmerl's PhD thesis, professional academic journals and conference presentations, and may also inform teaching on this subject.

The results of the study will be held in a locked and secure environment. As the project is expected to generate public interest, informants are requested to allow the use of their name, and attribution of information and opinions to them in the publishing of this research.

Informants are further requested to allow the lodging of research materials in a publicly accessible repository (e.g. a National Library) once the research is completed. This is to ensure that the precious oral history material this research will collect will be around for future generations of researchers. You will be asked to indicate whether you agree to this on a consent form and whether you agree to the lodging of recordings or would prefer that I just lodged transcripts of our conversation.

This research has been approved by the Flinders University Social and Behavioural Research Ethics Committee.

If you have any questions please contact Marcus Schmerl (Ph. 0423 924 687 or marcus.schmerl@flinders.edu.au or Melanie Swalwell Ph. 0882012619 or melanie.swalwell@flinders.edu.au) at the School of Humanities & Creative Arts, Flinders University, G.P.O. Box 2100, Adelaide SA, 5001, Australia.

Thank you.

This research project has been approved by the Flinders University Social and Behavioural Research Ethics Committee (Project No. 5404). For more information regarding ethical approval of the project the Executive Officer of the Committee can be contacted by telephone on 8201 3116, by fax on 8201 2035 or by email human.researchethics@flinders.edu.au.

Appendix 2: Interview Consent Form

Flinders UNIVERSITY

Marcus Schmerl
Screen and Media Department
School of Humanities & Creative Arts
GPO Box 2100
Adelaide SA 5001
Tel: 0423 924 687
marcus.schmerl@flinders.edu.au
http://www.flinders.edu.au/people/marcus.schmerl
CRICOS Provider No. 00114A

CONSENT FORM FOR PARTICIPATION IN RESEARCH BY INTERVIEW

I ___________________ (participant's name) agree to participate in the research being conducted by Mr Marcus Schmerl (Screen and Media Department, marcus.schmerl@flinders.edu.au, 0423 924 687) as part of the the project "A History of Creative Computing in Australasia." Mr Schmerl is undertaking PhD studies on "Educational Computing and Software in Australia: A History of Australian Educational Software Development for Australian Schools, 1970 – 1990."

I understand that the purpose of this study is to collect information on a range of aspects of educational computing in the 1970s and 1980s, including what people did with early microcomputers in schools, how the computer was viewed in educational settings at that time, how educators and students used microcomputers, what institutional supports there were, and what survives of the educational computing work of this period. I understand that this research will be used in assembling a history of educational computing in Australia during this time frame. This will be presented in a range of formats, including Mr Schmerl's PhD thesis, professional academic journals and conference presentations, and may also inform teaching on this subject.

I understand that my participation in this research will involve an interview about my involvement in the early years of educational computing, and possibly answering some follow up questions. I give my permission for this interview to be audio or video recorded.

I understand that the interviews and transcripts generated by this study will be kept and held securely by the researchers, in locked premises.

I understand that the Flinders University Social and Behavioural Research Ethics Committee have approved this research. I have had an opportunity to ask questions and have them answered to my satisfaction. I am aware that I can contact Marcus Schmerl or Dr Melanie Swalwell if I have any concerns about the research. I also understand that I am free to withdraw my participation from this research project at any time I wish before data collection and analysis is complete without having to give reasons or without penalty of any sort.

I consent to information or opinions that I have given being attributed to me in a report on this research.

Please tick one of the following:

☐ I consent to the video recording of the interview being lodged in a public oral history repository (such as a National Library) at the conclusion of the research.

☐ I would prefer that only the transcript of the interview be lodged in a public oral history repository at the conclusion of the research.

Participant's signature...Date.......................

I certify that I have explained the study to the participant and consider that she/he understands what is involved and freely consents to participation.

Researcher's name...

Researcher's signature.......................................Date........................

This research project has been approved by the Flinders University Social and Behavioural Research Ethics Committee (Project No. 5404). For more information regarding ethical approval of the project the Executive Officer of the Committee can be contacted by telephone on +61 8 8201 3116, by fax on +61 8 8201 2035 or by email human.researchethics@flinders.edu.au.

Appendix 3: Phone and Email Interview Introduction Scripts

Verbal script (eg, telephone call, face-to-face recruitment) (if applicable)
Hello, my name is Marcus Schmerl. I am a PhD candidate at Flinders University, undertaking a project on the history of educational computing in Australasia. I understand that you were involved with early educational microcomputing during the 1970s and / or 1980s. I am wondering if you would be willing to speak with me about this?

If yes: Great. It is a requirement of the Social and Behavioural Research Ethics Committee that I supply you with an Information Sheet introducing the project and its purpose. Is there an email or postal address that you'd like me to send this information to?

I'll call back/you can contact me once you've had a chance to peruse this, and if you're willing to participate, we can make a time for an interview.

Email text (if applicable)
Dear (insert name),
Hello, my name is Marcus Schmerl. I am writing to introduce myself and to ask if you would consider participating in a research project I am undertaking, on the history of educational computing in 1970s and 1980s in Australia. I am a PhD candidate at Flinders University. I understand that you were involved with educational computing in the 1970s and / or 1980s. Would you be prepared to talk with me about this some more, and possibly record an interview with me on the subject?

It is a requirement of the Social and Behavioural Research Ethics Committee that I supply you with an Information Sheet introducing the project and its purpose. I have attached this Information Sheet with some further information about the project.

I hope you will consider participating. I look forward to hearing back from you.

Yours sincerely,
Marcus Schmerl

BIBLIOGRAPHY

4mation Educational Resources Ltd (2017). Available at: http://www.4mation.co.uk/ (Accessed: 2 May 2017).

80's computer game. Had to cross a river, build raft in a set amount of turns (2009). Available at: https://www.reddit.com/r/tipofmytongue/comments/9enpg/tomt_80s_computer_game_had_to_cross_a_river_build/ (Accessed: 14 June 2017).

80s Desert Island game... anyone? (2011). Available at: https://forums.whirlpool.net.au/archive/1339021 (Accessed: 14 June 2017).

About EdTechSA (2016). Available at: http://edtechsa.sa.edu.au/edtechsa-who-we-are-and-what-we-do/ (Accessed: 10 April 2016).

About Jacaranda (no date). Available at: https://www.wiley.com//WileyCDA/Brand/id-16.html (Accessed: 1 May 2017).

Ackermann, E. (2001) 'Piaget's Constructivism , Papert's Constructionism : What's the difference ?', *Constructivism: Uses and Perspectives in Education*, pp. 1–11. doi: 10.1.1.132.4253.

Ackermann, E. K. (1996) 'Perspective-Taking and Object Construction: Two Keys to Learning', *Constuctionism in Practice: Designing, Thinking, and Learning in a Digital World*, pp. 25–37.

Anderson, J. (1984) *Computing in Schools: An Australian Perspective. Australian Education Review*. 21st edn. Australian Council for Educational Research. Available at: http://files.eric.ed.gov/fulltext/ED246860.pdf.

Angle Park Computing Centre (1975) ‘A handbook of computing resources’. Adelaide, South Australia: Education Department of South Australia, p. 55.

Angle Park Computing Centre (1985) ‘Path Weaver: an adventure game authoring system for the Commodore C-64 microcomputer’. Education Department of South Australia.

Appleton, I. (2001a) ‘Dimension: A Magazine about Computing in Secondary Schools’, in *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. Michael Deves Publishing. Available at: http://www.lythrumpress.com.au/vision/265.html.

Appleton, I. (2001b) ‘John Walker Trod on Toes but Chose Where He Trod’, in *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. Michael Deves Publishing. Available at: http://www.lythrumpress.com.au/vision/258.html.

Archives of the Australian Society for Educational Technology (2016). Available at: http://ascilite.org/archived-journals/aset/index.html (Accessed: 5 December 2016).

Australasian Digital Heritage Database (2016). Available at: http://www.ourdigitalheritage.org (Accessed: 1 February 2016).

Australian Curriculum: Digital Technologies (2018).

Baek, Y. K. (2008) ‘What Hinders Teachers in Using Computer and Video Games in the Classroom? Exploring Factors Inhibiting the Uptake of Computer and Video Games’, *CyberPsychology & Behavior*, 11(6), pp. 665–671. doi: 10.1089/cpb.2008.0127.

Beale, D. (2014) *How the computer went to school : Australian government policies for computers in schools, 1983-2013*. Clayton, Victoria: Monash University Publishing.

Bigum, C. (2012) ‘Schools and Computers: Tales of a Digital Romance’, in *Transformative Approaches to New Technologies and Student Diversity in Futures Oriented Classrooms*. Dordrecht: Springer Netherlands, pp. 15–28. doi: 10.1007/978-94-007-2642-0_2.

Butler, J. E. (1983) *Moving Into Maps - Volume 1 of Jacaranda atlas programme*. Jacaranda Wiley.

Charmaz, K. (2003) ‘Qualitative Interviewing and Grounded Theory Analysis’, in *Inside Interviewing: New Lenses, New Concerns*. SAGE Publications, Inc, pp. 311–330.

Charmaz, K. and Belgrave, L. L. (2012) ‘Qualitative Interviewing and Grounded Theory Analysis’, in *The SAGE Handbook of Interview Research: The Complexity of the Craft*. SAGE Publications, Inc, pp. 347–365.

Clyde, L. A. (1988) ‘Online information services in schools’, *The Electronic Library*, 6(2), p. 9.

Commodore Annual 1990 (No 3), Saturday Magazine Pty Ltd (1990a) ‘Commodore in Education’, pp. 38–40. Available at: https://archive.org/details/Commodore_Annual_1990_Saturday_Magazine_AU/.

Commodore Annual 1990 (No 3), Saturday Magazine Pty Ltd (1990b) ‘Education with Satchel Software’, pp. 40–46. Available at: https://archive.org/details/Commodore_Annual_1990_Saturday_Magazine_AU/.

Compute Magazine Issue 154 (1993) 'Jara-Tava Review', July, pp. 100–103. Available at: https://archive.org/stream/1993-07-compute-magazine/Compute_Issue_154_1993_Jul#page/n99/mode/2up/search/jara+tava+satchel+software.

'Computers in Education: An Outline of Country Experiences.' (1984) in *Report of the Asian Seminar on Educational Technology (3,31, Tokyo, Japan, September 26-October 2, 1984).* Unesco Regional Office for Education in Asia and the Pacific.

Connell, W. F. (1993) *Reshaping Australian Education 1960 – 1985*. Hawthorn, Victoria: The Australian Council for Educational Research Ltd.

Davey, B. and Parker, K. R. (2010) 'Turning Points in Computer Education', in *IFIP International Conference on the History of Computing*. Berlin, Heidelberg: Springer, pp. 159–168. doi: 10.1007/978-3-642-15199-6_16.

Davis, C. (2014) *Return To Gold-Dust Island*, *Blow the Cartridge*. Available at: https://blowthecartridge.com/2014/04/21/return-to-gold-dust-island/ (Accessed: 20 February 2017).

Desrochers, N. and Apollon, D. (eds) (2014) *Examining paratextual theory and its applications in digital culture.* IGI Global.

Dimension: A magazine about computing in secondary schools, 1969-1972. (1972). Adelaide: Angle Park Computing Centre, Education Department of South Australia. Available at: http://catalogue.nla.gov.au/Record/2608021.

'Dimension Volume 9' (1971). Angle Park Computing Centre, Education Department of South Australia, p. 20.

Dunstan, D. (1973) 'Speech notes opening Angle Park Computer Centre project'.

Adelaide, South Australia. Available at: http://hdl.handle.net/2328/27231.

Education Department of South Australia (1973) *Angle Park Community Centre : rationale, brief & sketches / a report prepared by the Education Department of South Australia with the financial assistance of the Australian Department of Tourism and Recreation*. Adelaide, South Australia: Education Department of South Australia.

Education Department of South Australia (1985) *Schools computing policy*.

'Eighth Summer School in Computing: Learning and Enjoying' (1981). Angle Park Computing Centre, Education Department of South Australia, p. 46.

England, C. (2015) 'Internode founder Simon Hackett's new Adelaide business incubator', *The Advertiser*, 3 March. Available at: https://www.adelaidenow.com.au/business/sa-business-journal/internode-founder-simon-hacketts-new-adelaide-business-incubator/news-story/d328794b60ddf16d05fa3ad77b458b09.

Fisher, R. W. A. (2001) 'Angle Park was Needed in This Very Poor District', in *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. Michael Deves Publishing. Available at: http://www.lythrumpress.com.au/vision/228.html.

Fitzgerald, D., Hattie, J. and Hughes, P. (1986) *Computer applications in Australian classrooms.* Canberra, ACT.: Australian Government Publishing Service. Available at: http://trove.nla.gov.au/work/11841279?selectedversion=NBD4835622.

Fleer, M. and Stout, A. (1991) *Support Services for Computer Education Teachers in Australia*. Bangkok, Thailand: UNESCO Principal Regional Office for Asia and the Pacific. Available at: http://unesdoc.unesco.org/images/0009/000907/090718eo.pdf.

Fond memories (2011). Available at: https://www.raspberrypi.org/forums/viewtopic.php?t=264 (Accessed: 19 May 2018).

Gamebase 64 (2016). Available at: http://www.gamebase64.com (Accessed: 1 February 2016).

'Gold-Dust Island Teachers Notes' (1984). Jacaranda Wiley.

Green, A. and Troup, K. (1999) *The Houses of History: A Critical Reader in Twentieth-century History and Theory*. Manchester University Press. Available at: https://books.google.com.au/books/about/The_Houses_of_History.html?id=Sea9JQ322k0C.

Green, M. (2010) *Granny's Garden creator interview*, *Pixelatron*. Available at: http://pixelatron.com/blog/grannys-garden-creator-interview/ (Accessed: 2 May 2017).

Greygum Software (2007). Available at: https://www.greygum.com.au/website/index.php (Accessed: 10 August 2016).

Hayton, G. and Loveder, P. (1989) *The application of computers in technical and vocational education: the experience of South Australia*. Technical and Further Education National Centre for Research and Development.

Hedberg, J. G. and McNamara, S. (2002) 'Innovation and Re-Invention: A Brief Review Of Educational Technology In Australia', *Educational Media International*, 39(2), pp. 111–121. doi: 10.1080/09523980210153471.

Hosie, P. (1985) 'Promises , promises - Viatel and education', *Australian Journal of Educational Technology*, 1(1), pp. 39–46.

Howard, S. K. and Mozejko, A. (2015) 'Considering the history of digital technologies in education', in *Teaching and Digital Technologies: Big Issues and Critical Questions*. Port Melbourne, Australia: Cambridge University Press., pp. 157–168.

Hugo, J. S. (2000) *A case study of teacher use of technology: The decision beyond the barriers of time, training, and adequate equipment*. Available at: https://search-proquest-com.ezproxy.flinders.edu.au/docview/304628916?accountid=10910.

Jacaranda Wiley (no date). Available at: http://www.wiley.com/legacy/products/worldwide/jacaranda/ (Accessed: 1 May 2017).

Jara-Tava: The Isle of Fire (no date). Available at: https://cpcrulez.fr/GamesTest/jara-tava-the_isle_of_fire.htm (Accessed: 20 May 2015).

Jolly, E. (2001) *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. Michael Deves Publishing.

Jones, A., McDougall, A. and Murnane, J. (2004) 'What Did We Think We Were Doing?', in *History of Computing in Education*. Boston: Kluwer Academic Publishers, pp. 63–72. doi: 10.1007/1-4020-8136-7_7.

Journal - Australian Council for Computers in Education (2016). Available at: https://acce.edu.au/journal/ (Accessed: 5 December 2016).

Kinnear, A. (1995) 'Introduction of microcomputers: A case study of patterns of use and children's perceptions', *Journal of Educational Computing Research*, 13(1), pp. 27–40. doi: 10.2190/NA0H-1RV6-LFLU-23H0.

Kleinig, X. (2006) 'Outrage as school plan is ditched', *The Advertiser*, 16 November, p. 1. Available at: https://www.adelaidenow.com.au/news/outrage-as-school-plan-is-ditched/news-story/85dba8ad0d51fedaf71c20b27e41b6d0.

Lean, T. (2012) 'Mediating the microcomputer: The educational character of the 1980s British popular computing boom', *Public Understanding of Science*, 22(5), pp. 546–558. doi: 10.1177/0963662512457904.

Lean, T. (2013) 'The Voice in the Machine: Oral History and Making the Computer Relevant', in *IFIP International Conference on the History of Computing*. Berlin, Heidelberg: Springer, pp. 163–172. doi: 10.1007/978-3-642-41650-7_16.

Lemon 64 (2016). Available at: http://lemon64.com (Accessed: 1 February 2016).

Maggs, A. and Ray, E. (1985) 'Microcomputers and education', *Australian Journal of Educational Technology*, 1(1), pp. 2–11.

McDougall, A. and McCrae, B. (2000) 'Australian computers in education conferences come of age', *Australian Educational Computing*, 15(1), pp. 3–6.

McDougall, A., Murnane, J. S. and Wills, S. (2014) 'The Educational Programming Language Logo: Its Nature and Its Use in Australia', in *Reflections on the History of Computers in Education*. Berlin, Heidelberg: Springer, pp. 393–407.

Mckerrow, P. J. (1982) *LOGO and turtle robots (University of Wollongong, Working Paper 82-9)*. Available at: http://ro.uow.edu.au/compsciwp/26.

McNamara, S. (1985) 'Teacher generated research in educational technology', *Australian Journal of Educational Technology*, 1(2), pp. 37–42.

Microbee Software Preservation Project (MSPP) (2016). Available at:

http://www.microbee-mspp.org.au (Accessed: 1 February 2016).

Morton, A. (1996) 'Factors Affecting the Integration of Computers in Western Sydney Secondary Schools', in *Learning Technologies: Prospects and Pathways. Selected papers from EdTech '96 Biennial Conference of the Australian Sciety for Educational Technology*, pp. 107–114. doi: 10.1017/CBO9781107415324.004.

Newall, D. (no date) *Angle Park Computing Centre*. Available at: http://davidnewall.com/angle-park.html (Accessed: 12 August 2015).

Oddbloke (2012) *Granny's Garden*, *The Further Adventures of Oddbloke*.

Papert, S. (1996) 'A Word For Learning', in *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Routledge, pp. 9–24. Available at: https://books.google.com.au/books?hl=en&lr=&id=3-Gdmm0-2wAC&oi=fnd&pg=PP2&dq=papert+constructionism&ots=TjPMbddjzR&sig=6J033RoWEYQXam-5Fe_Am1ROygg#v=onepage&q=papert constructionism&f=false.

Papert, S. and Harel, I. (1991) 'Situating Constructionism', in *Constructionism*. Ablex Publishing Corporation. Available at: http://www.papert.org/articles/SituatingConstructionism.html.

Pearcy, T. (1988) *A History of Australian Computing*. Melbourne, Victoria.: Chisholm Institute of Technology.

Phillips, G. (2001) 'The Angle Park Kids Were the Easiest I've Had to Teach', in *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. Michael Deves Publishing. Available at: http://www.lythrumpress.com.au/vision/070.html.

Pieces of eight (2014). Available at:

http://www.ourdigitalheritage.org/archive/playitagain/games/pieces-of-eight/ (Accessed: 11 October 2019).

Pullen, G. C. (1981) ‘School libraries in Tasmania’, in *Recent Advances in School Librarianship: Recent Advances in Library and Information Services (Vol. 1).* Pergamon Press, pp. 184–186.

‘Raft-Away River documentation’ (1985). Jacaranda Wiley.

Raft Away River (2003). Available at: http://www.dosgames.com/forum/post-22459.html (Accessed: 14 June 2017).

Reeves, T. C. (1998) *The impact of media and technology in schools*, *A Research Report prepared for the Bertelsmann Foundation. University of Georgia. Partnership for 21st Century Skills (2003). Learning for the 21st Century: A Report and a Mile Guide for 21st Century Skills. Washington*. Available at: http://it.coe.uga.edu/~treeves/edit6900/BertelsmannReeves98.pdf.

Richardson, J. (no date) *Logo in Australia: A Vision of Their Own*. Available at: http://www.microworlds.com/support/logo-philosophy-australia.html (Accessed: 10 November 2016).

Robson, J., Routcliffe, P. and Fitzgerald, R. (1991) *Remote Schooling & Information Technology (A Guide for Teachers)*. Canberra: Australian Catholic University.

Sandery, P. (1975) *Computing in South Australian Schools*. Sigma.

Satchel Software (1985) ‘Logo in the early years: 3 case studies’. Angle Park Computing Centre, Education Department of South Australia.

Satchel Software (1986) ‘Databases in the Classroom’. Angle Park Computing

Centre, Education Department of South Australia.

'Scavenger Hunt software documentation' (1983). Jacaranda Wiley.

Schiller, J. (1991) 'Implementing Computer Education: The Role of the Primary Principal', *Australian Journal of Educational Technology*, 7(1), pp. 48–69. Available at: http://ascilite.org.au/ajet/ajet7/schiller.html.

Schmerl, M. (2017a) 'Interview with Dean Hodgson by Marcus Schmerl'.

Schmerl, M. (2017b) 'Interview with Dean Rosenhain by Marcus Schmerl'.

Schmerl, M. (2017c) 'Interview with Gerald Wluka by Marcus Schmerl'.

Schmerl, M. (2017d) 'Interview with Philip O'Carroll by Marcus Schmerl'.

Schmerl, M. (2017e) 'Interview with Rosanne Hood by Marcus Schmerl'.

Seifort, T. (no date) *Computer cards*. Available at: http://www.cameratim.com/computing/punch-cards.

Selwyn, N. (2011) 'Editorial: In praise of pessimism-the need for negativity in educational technology', *British Journal of Educational Technology*, 42(5), pp. 713–718. doi: 10.1111/j.1467-8535.2011.01215.x.

Smith, R. (1994) *Technologies for enhanced learning : current and future use of technologies in school education. The Directorate of School Education, Victoria.* Melbourne.

Sollorz, P. L. (2013) *The History of Computer Use in Senior School Mathematics Teaching in the Government School System of New South Wales: 1971 – 1992.* Curtain University. Available at: http://espace.library.curtin.edu.au/R?func=dbin-jump-full&local_base=gen01-era02&object_id=194451.

South Australian Education Department, Angle Park Computing Centre and Darlington Materials Development Centre (1991) *Nexus : an introduction*. Education Department of South Australia.

Stardot.Org.UK Forums [Educational Archive not in STH] (2013). Available at: http://stardot.org.uk/forums/viewtopic.php?t=7190&start=30.

Stuckey, H. *et al.* (2013a) *Jacaranda Software*. Available at: http://playitagainproject.org/companies/jacaranda-software/ (Accessed: 15 August 2016).

Stuckey, H. *et al.* (2013b) *Raft Away River*. Available at: http://playitagainproject.org/games/raft-away-river/ (Accessed: 7 January 2016).

Suominen, J. (2011a) 'Retrogaming Community Memory and Discourses of Digital History Jaakko Suominen', *Navigating Landscapes of Mediated Memory*. Available at: http://www.inter-disciplinary.net/wp-content/uploads/2011/04/jsuominenpaper.pdf.

Suominen, J. (2011b) 'The Past as the Future? Nostalgia and Retrogaming in Digital Culture', *The Fibreculture Journal*, 75, pp. 1–13.

Swalwell, M. (2007) 'The Remembering and the Forgetting of Early Digital Games: From Novelty to Detritus and Back Again', *Journal of Visual Culture*, 6(2), pp. 255–273. doi: 10.1177/1470412914565965.

Swalwell, M. (2008) '1980s Home Coding : The Art of Amateur Programming', in Brennan, S. and Ballard, S. (eds) *The Aotearoa Digital Arts Reader*. Aotearoa Digital Arts and Clouds, pp. 193–201.

Swalwell, M. (2012a) 'Interview with Bruce Mitchell and Steve Luckett'.

Swalwell, M. (2012b) ‘Interview with David L. Smith’.

Swalwell, M. (2015) ‘History of Video Games as History of Technology?’, *Society for the History of Technology annual meeting*.

Swalwell, M. (no date) *Homebrew Gaming and the Beginnings of Vernacular Digitality*. Cambridge, MA: MIT Press.

Swalwell, M. and De Vries, D. (2013) ‘Collecting and Conserving Code: Challenges and Strategies’, *SCAN: JOURNAL OF MEDIA ARTS CULTURE, 10(2)*. doi: 10.1177/1354856509105107.

Swalwell, M. and De Vries, D. (no date) *Australasian Digital Heritage*.

Tannock, P. D. (1983) *Teaching, learning and computers : report of the National Advisory Committee on Computers in Schools.*

Tatnall, A. (2013) ‘The Australian educational computer that never was’, *IEEE Annals of the History of Computing*, 35(1), pp. 35–47. doi: 10.1109/MAHC.2012.16.

Tatnall, A. (2014) ‘Schools, Students, Computers and Curriculum in Victoria in the 1970s and 1980s’, in Tatnall, A. and Davey, B. (eds) *Reflections on the History of Computers in Education*. Berlin, Heidelberg: Springer, pp. 246–265. doi: 10.1007/978-3-642-55119-2_17.

Tatnall, A. (2016) ‘The Beginnings of Government Support for Computers in Schools – The State Computer Education Centre of Victoria in the 1980s’, in Kreps, M., Fletcher, G., and Griffiths, M. (eds) *Technology and Intimacy: Choice or Coercion. HCC 2016. IFIP Advances in Information and Communication Technology*. Springer, Cham, pp. 291–302. doi: 10.1007/978-3-319-44805-3_23.

Tatnall, A. and Davey, B. (2004) 'Streams in the History of Computer Education in Australia', in Impagliazzo, J. and Lee, J. A. N. (eds) *History of Computing in Education*. Boston: Kluwer Academic Publishers, pp. 83–90. doi: 10.1007/1-4020-8136-7_9.

Tatnall, A. and Davey, B. (2008) 'Computer Education Support Structures in Victorian Schools in the 1980s', in Impagliazzo, J. (ed.) *History of Computing and Education 3 (HCE3). IFIP Advances in Information and Communication Technology, vol 269.* Boston, MA: Springer US, pp. 1–22. doi: 10.1007/978-0-387-09657-5_1.

Tatnall, A. and Davey, B. (2014) *Reflections on the History of Computers in Education*. Edited by A. Tatnall and B. Davey. Berlin, Heidelberg: Springer. doi: 10.1007/978-3-642-55119-2.

Tatnall, A. and Leonard, R. (2010) 'Purpose-Built Educational Computers in the 1980s : The Australian Experience', in Tatnall, A. (ed.) *History of Computing. Learning from the Past. IFIP Advances in Information and Communication Technology, vol 325.* Berlin, Heidelberg: Springer, pp. 101–111. Available at: https://link.springer.com/chapter/10.1007/978-3-642-15199-6_11.

The Internet Archive Software Collection (2015). Available at: https://archive.org/details/software (Accessed: 20 July 2015).

The Parks Community Centre - Background (2013). Available at: https://www.infrastructure.sa.gov.au/the_parks_community_centre/about_the_project (Accessed: 10 August 2017).

Trinidad, S. (2002) 'A Journey of Networking Learners Down Under: A Reflection of the Evolution of ICT in Western Australian Primary Schools', in McDougall, A., Murnane, J., and Chambers, D. (eds) *WCCE2001 Australian Topics: Selected Papers from the Seventh World Conference on Computers in Education*, pp. 125–129. Available at: http://crpit.com/confpapers/CRPITV8Trinidad.pdf.

Veale, V. (2001a) 'Team teaching allowed exciting flexibility', in *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. 1st edn. Adelaide: Michael Deves Publishing. Available at: http://www.lythrumpress.com.au/vision/225.html.

Veale, V. (2001b) 'The Concept of The Parks Community Education Centre', in *A Broader Vision : Voices of Vocational Education in Twentieth-Century South Australia*. Adelaide Michael Deves Publishing. Available at: http://www.lythrumpress.com.au/vision/272.html.

VGJunk (2018) *Granny's Garden (BBC Micro)*. Available at: http://retrovania-vgjunk.blogspot.com/2018/04/grannys-garden-bbc-micro.html (Accessed: 25 June 2019).

Walker, R. (1991) 'The Development of Educational Computing Policy in the Victorian School System, 1976-1985', *Australian Journal of Education*, 35(3), pp. 292–313. doi: 10.1177/000494419103500306.

Wallace, A. R. (1989) 'The Use of Computers in Australian Schools : Six Years of Pragmatism', in *The Annual Meeting of the American Educational Research Association (San Francisco, CA, April 1989).*, p. 32. Available at: https://archive.org/details/ERIC_ED312989.

Walsh, S. (1987) *Classic Fantasies: A resource package for the Commodore 64*. Adelaide, South Australia: Satchel Software, South Australian Education Department.

Wheeler, A. (2008) *An Interview with Dean Hodgson*, *Sega SC-3000 Survivors*. Available at: http://www.sc-3000.com/images/stories/interviews/DeanHodgson/InterviewDeanHodgson.pdf (Accessed: 1 September 2015).

Wills, S. (1999) *First Fleet Database*. Available at: http://firstfleet.uow.edu.au/search.html (Accessed: 20 September 2016).

Wiltfang, G. L. and Berg, B. L. (1990) *Qualitative Research Methods for the Social Sciences*. Fourth, *Teaching Sociology*. Fourth. Allyn and Bacon. doi: 10.2307/1317652.

'Wordsmith manual' (1985). Jacaranda Wiley.

Wordsworth, E. J. (no date) *Microbee Technology - About Us / History*. Available at: http://www.microbeetechnology.com.au/aboutus.htm (Accessed: 8 October 2016).