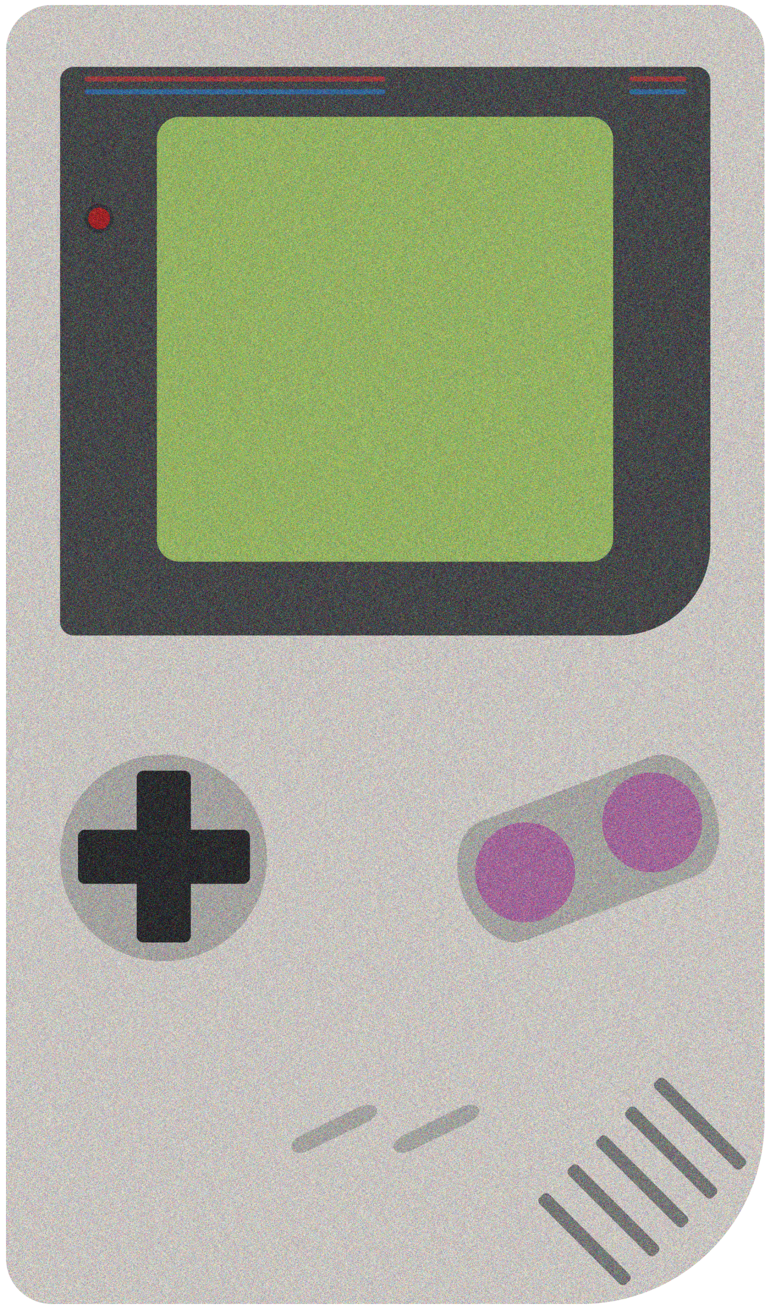
a history of video games



Tuakana Pokoina Hone Metuarau

Victoria University of Wellington 2017

A thesis submitted to the Victoria University of Wellington in fulfilment of the requirements for the degree of Masters of Design Innovation.

Acknowledgements

I first would like to thank my family for their continual love and support. I thank my parents for always pushing me to be the best that I can in all my endeavours, and thank my brother, sister and uncle for always being there.

I would like to thank all of my friends for their support, opposition and criticism over the years. Their input means a great deal more than they know and I am forever grateful for it.

I would like to thank David Hakaraia and Te Roopu Awhina for their tremendous support throughout my time at university.

I would also like to the thank the School of Design and all the VUW staff who have helped me along the way.

Lastly I would like to thank my supervisor Dr. Leon Gurevitch for all his work and pushing me through to the very end.

Abstract

This research begins with the premise that while video-games have become a pervasive cultural force over the last four decades, there is still a dearth of educational and historical material regarding the emergence of video game home consoles and their content. Games have an extensive history, dating back to early radar displays and oscilloscopes of the 1960s (Tennis for Two, 1958) and early home video game consoles of the 1970s (Magnavox Odyssey, 1972). From the JAMMA (Japanese Amusement Machine and Marketing Association) arcade standard of the 80s to the high powered processors of Sonys PS4, video games have come a long way and left a wealth of audio-visual material in their wake. Much of this material, however, is archived and engaged within a traditional manner: through text books or museum exhibitions (Games Master, ACMI 2015). Through interactive design however, this data can be made easily comprehensible and accessible as interactive data-visualisation content. This design research project explores processes of data visualization, interactive design and video game production to open up video game history and communicate its developmental stages in a universally accessible manner. Though there has been research conducted utilising game engines for visualizations in other fields (from landscape architecture to bio-medical science) it has rarely been used to visualize the history of gaming itself. This visualization (utilising the Unreal Engine and incorporating historical video content) creates an accessible preservation and catalogue of video game history, and an interactive graphical interface that allows users to easily learn and understand the history of console development and the processes that lead video games to their current state.

Table of Contents

Preface I

Acknowledgements III

Abstract V

Chapter 1: Introduction 9

- Video Games, Interactive Design and Data Visualization 10

Chapter 2: Methodology 14

- Research and Development Methods 15

Chapter 3: Literature Review 19

- Data Visualization within Video Games 20
- Virtual Navigation and Video Game Evolution 23

Chapter 4: Site Analysis 31

- Thesis Outline and Considerations 32

Chapter 5: Project Review 34

- Game Engines for Visualization 35

Chapter 6: Preliminary Design 38

- Text Mining and Knowledge Distillation 39
- Operation Prototyping and UI Development 41
- Level Construction 44
- UI Revision and Data Integration 68

Chapter 7: Developed Design 71

- Output Reflection 72

Chapter 8: Conclusion 73

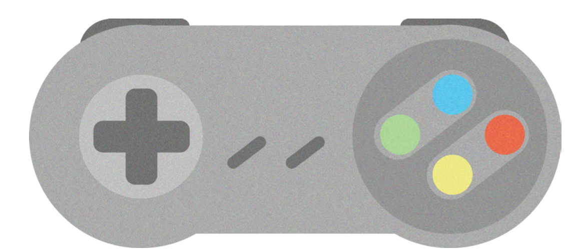
- Conclusions and Project Reflection 74

Bibliography 76

List of Figures 78

Chapter 1

INTRODUCTION



Video Games, Interactive Design and Data Visualization

Video games have an extensive history, a history that is not easily accessible or comprehensible to the general public. Having become a cultural force and influential medium one may ask how from a technical, visual and interactive standpoint they have developed to such an extent. With origins in military simulation and a process whereby games production explored “the *power* of the computer” (Wolf, 2008, p.31) throughout the 1950s and 1960s, video games have grown to become a multi-billion dollar industry over the past thirty to forty years (Wolf, 2015, p.469). This project intends to investigate how video game history, specifically that of video game home consoles, can be made accessible and comprehensible to the public through a combination of both interactive design and data visualization.

Investigating how interactive design can visualize video game history is a broad topic to say the least. With this in mind various factors had to be taken into consideration, the first and most obvious being time and resources. In the limited time allocated to this research project, the choice was made to strictly focus on the home console market and its development. However, even with this focus in mind, a complete preservation and catalogue of the whole of home console history could not be included. Therefore, the intention was to utilize interactive design to develop a visualization that would make this history as comprehensible and accessible to users as possible.

The intended design output of this thesis project was an interactive visualization that consists of two interactive components. The first, a library in which data would be stored and could be accessed and easily sorted, allowing the user to view statistical information on home consoles, ranging from sales figures to release dates, of the past 30 to 40 years. The second component, a working game comprised of multiple levels allowing users to experience the history in both an interactive and visual manner. Each level corresponds to a generation in home consoles and attempts to faithfully recreate the gameplay, interaction and aesthetics of each era. Console generations are a broad term used to describe a time period in which certain home consoles were released (Wolf, 2008). Each constructed level displays data to the user in the form of visual text integrated with gameplay, with this information highlighting technological innovations and notable moments in a home console or console generations development and design. These levels attempt to create a form of narrative, allowing the player to navigate each level of the game and essentially play through video game history. To achieve this knowledge and techniques from disciplines of interactive design, game design and data visualization have all been utilized.

Interactive design shares an obvious relationship with video games, with interaction being the key feature that games are primarily built upon, while data visualization shares a similar but not necessarily obvious connection. Previous research discussing both data visualization and video games often discusses how video games visualize player data. Veronica Zammitto in “Visualization Techniques in Video Games” highlights video games extensive reliance on displaying game information to the player, and how said information is crucial to player performance within the game world (2008). Her research focuses on video games use of visualization to display player statistics such as health, ammunition, maps and other user interface components. Prior research also discusses the use of video game engines for the development of visualizations in other fields. For instance, Friese, Herrlich and Wolters, in their paper “Using Game Engines for Visualization in Scientific Applications”, attempted to visualize geographical data utilising video game engines, in this case using the Unreal Engine and the Crysis Engine. A third example of research undertaken has been the extraction and visualization of game data. Thawonmas and Lizuka (2008), in their paper “Visualization of Online-Game Players Based On Their Action Behaviours”, applied visualization techniques to research in-game player behaviour based on their actions within the game world. From this visualization they hoped to better understand player behaviour and improve the service quality of online games. Prior research, however, has not used data visualization and video game software to visualize video games themselves. Video games and video game tools have either been used to visualize in the service of other fields of research, or been used as a source from which to extract data, with a separate data visualization software interpreting the gathered information.

The intentions of this thesis were to utilize video games and their respective development tools in service of video game research undertaking the following aims to do so. The first aim was to research, compile and catalogue video game home console data, with the objective of identifying key video game data. This data consisted of console and game release dates, sales figures and information regarding hardware and software innovations. The data is then sorted and categorised before the second aim of the project is undertaken. The second aim being to visualize the measured data through interactive design. To achieve this, the development of an interactive digital library, which stores all of the gathered data, was undertaken. Alongside this a game that interacts with the library of data and displays a refined visualization was developed. This game was split into multiple levels that each represent a different period in home console history, referred to by generations, and utilize the gathered data, allowing one to play through the evolution of home console history in an engaging interactive manner. These tasks were undertaken utilising the Unreal Engine alongside other software tools. In summary the design output consists of an interactive digital visualization that acts as both video game and visualization of video game history.

Figure 1.1 Game-Masters-Image-Gallery-01. From Games Masters, Inside the exhibition. Photography by Mark Gambino, Image Courtesy ACMI. Reprinted with permission.

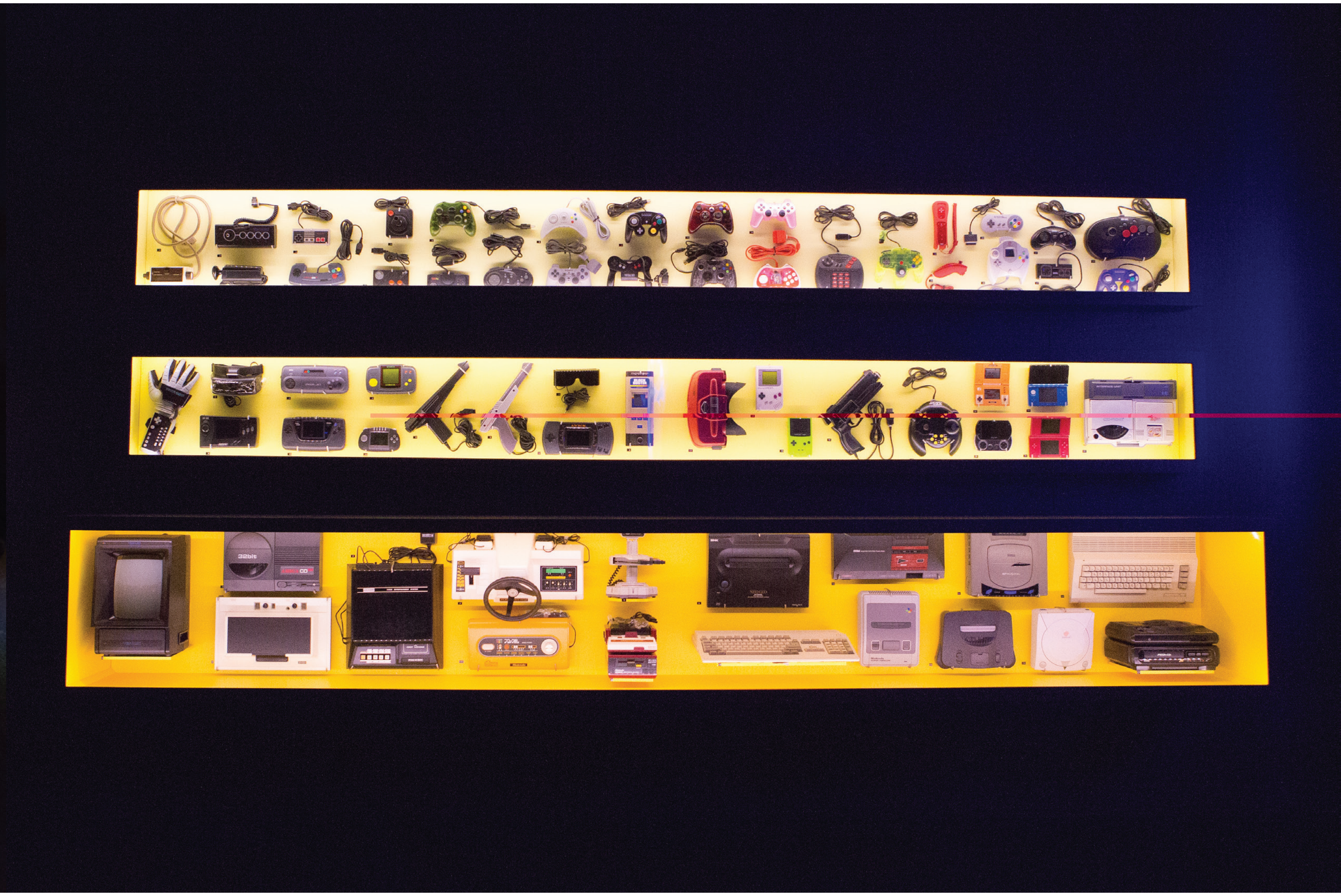
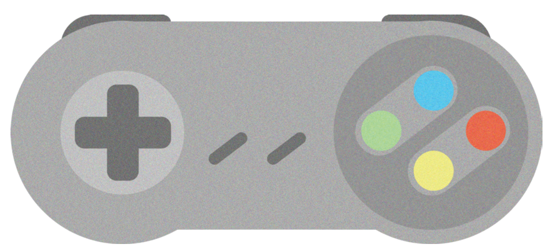


Figure 1.2 Game-Masters-Image-Gallery-02. From Games Masters, Inside the exhibition. Photography by Mark Gambino, Image Courtesy ACMI. Reprinted with permission.

Chapter 2

METHODOLOGY



Research and Development Methods

The research undertaken to pursue the above aims and objectives adhered to the following methods. Utilising the framework of text mining I followed the protocols laid out by Ah-Hwee Tan in his article "Text Mining: The State of the Art and the Challenges" (1999). Text mining involves the process of extracting data that is both "interesting and non-trivial" (Tan, 1999, p.65) from "unstructured text documents" (1999, p.65). This method was utilised for the gathering and sorting of data and it is this data that the design output was built upon. The text mining framework is split into two components, text refining and knowledge distillation. Text refining converts gathered data from unstructured text into a chosen intermediate form, either document based or concept based. Document based refers to a document while concept based represents an object or "concepts of interest in a specific domain" (1999, p.66). Knowledge distillation then deduces patterns from the gathered intermediate forms. Utilising concept based intermediate formats; I was able to deduce information related to the objects of home video game consoles from unstructured texts that I chose to mine. A majority of the texts being mined were online websites and documents. These initially consisted of Wikipedia pages as they provided an easy accessible point from which to extract large amounts of statistical data. However with home consoles history it is likely that not all information was necessarily available on Wikipedia; therefore other online resources, such as forums (neogaf.com and forum.digitpress.com) were utilised where information was either unconfirmed or missing. The readings and chapters utilised for this thesis were also mined where relevant information was found with their data also taken into account.

This framework was chosen due to its open minded approach to suitable texts, and its formatting suiting the compiling and cataloguing of gathered video game data. It should be noted that this approach is often used in

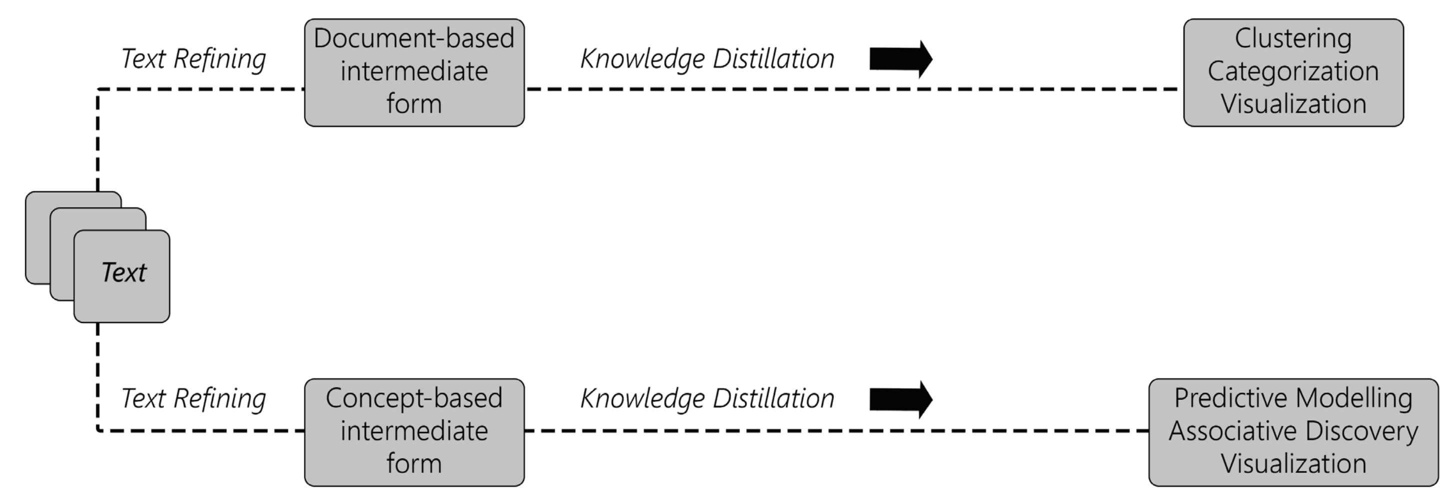


Figure 2.1. The text mining framework, as outlined by Tan (1999, p.2).

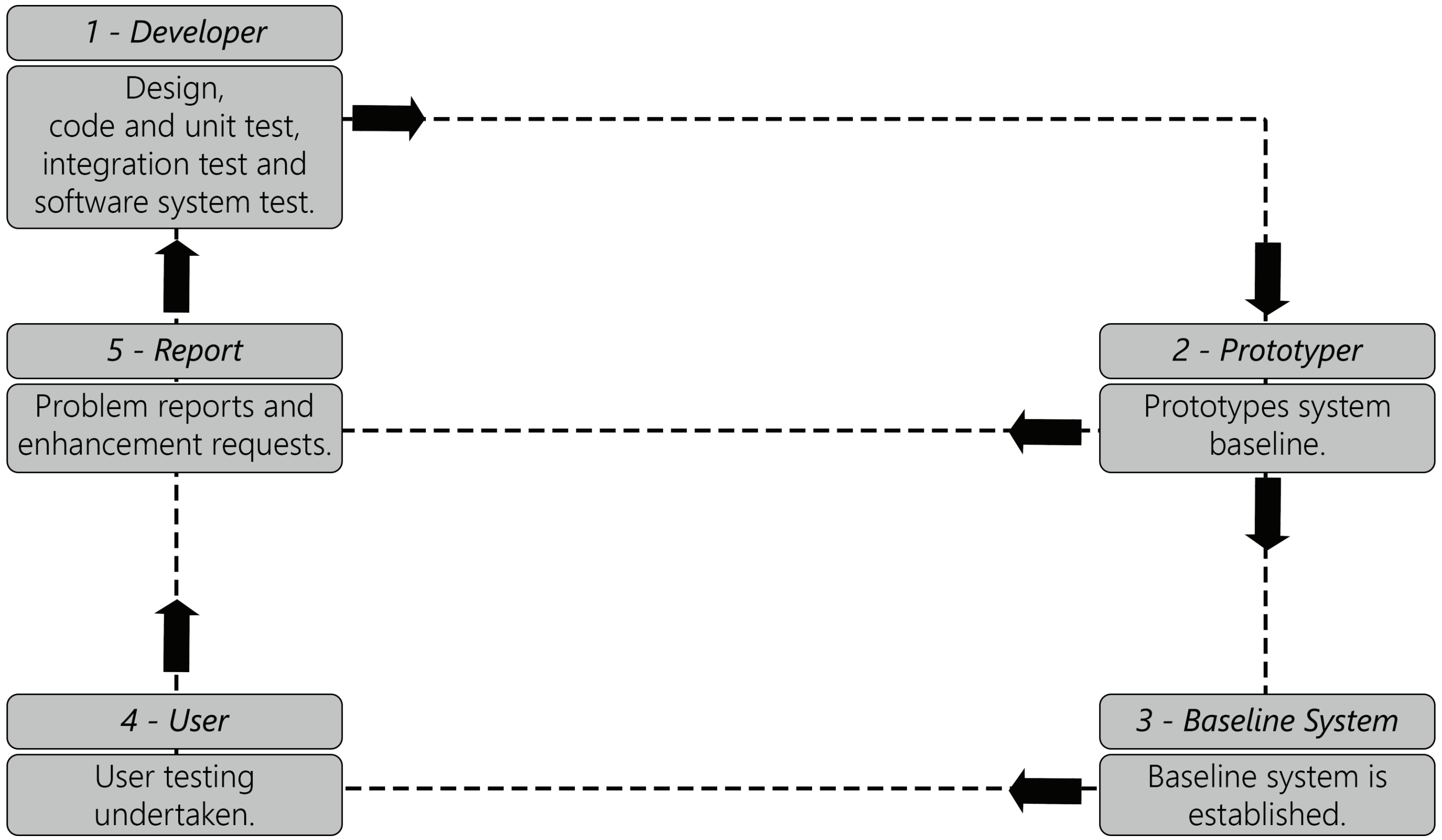


Figure 2.2. Operational Prototyping diagram, as outlined by Davis. (Davis, 1992, p.74)

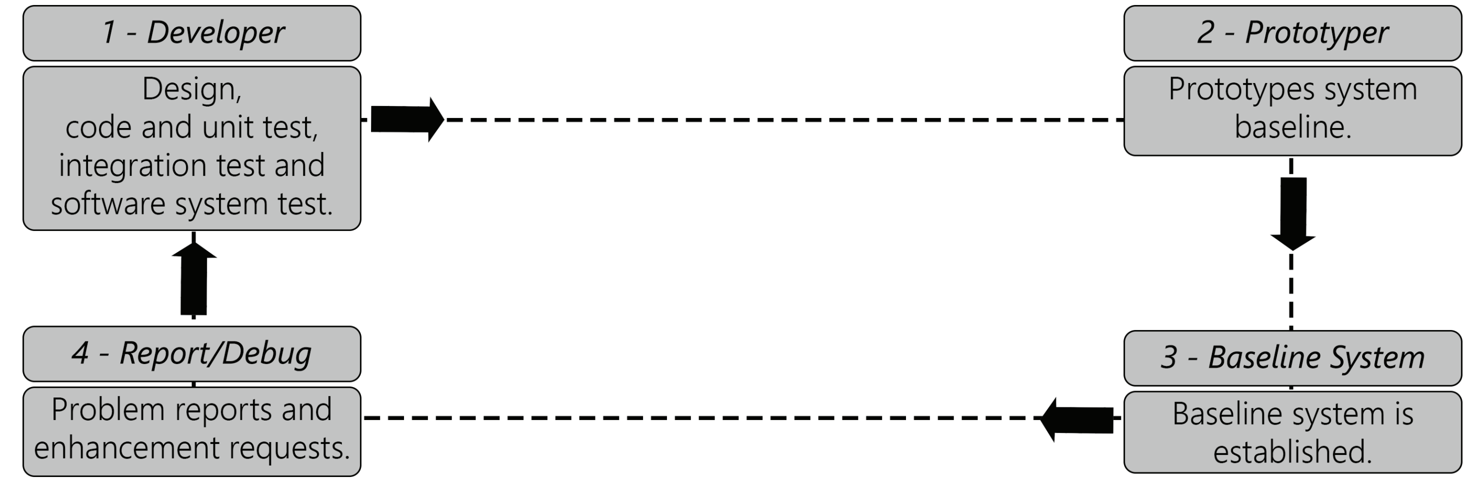


Figure 2.3. Operational Prototyping method revised to suit the project needs.

conjunction with text mining or data mining software, however Tans clear documenting of the framework allowed it to be easily adapted to the needs of this thesis without the needs of topic modelling.

The building and construction of the interactive visualization itself was developed through the process of prototyping, specifically Alan M. Davis method of operational prototyping (1992). To construct both the game aspects and digital library this method was used to establish the software foundations that were built upon and iterated into the final digital output. This development was conducted following the work outlined by Heather Maxwell Chandler & Rafael Chandler (2009), from the book *Fundamentals of Game Development*, and utilised the process of Davis (1992), from the paper "Operational Prototyping: A New Approach". This method was utilised, again in a slightly altered manner, to better suit the requirements of the project. Prototyping is a key part of any design process; however the benefits of operational prototyping and how it can be utilised in game design were what led to its selection. Operational prototyping consists of working from "an initial prototype that is refined in iterations until it becomes a final working version" (Chandler & Chandler, 2011, p.135). Davis' framework (Davis, 1992, p.71) utilises the combined approaches of evolutionary prototyping, a prototype "built in a quality manner" that implements "only confirmed requirements", and throwaway prototyping, a prototype that is "built as quickly as possible" and discarded once the "desired information is learned" (Davis, 1992, p.71) to build a full working system. Though established for software development as well as the implementation and testing of said software, this method of operational prototyping was deemed suitable as it allowed for a balanced but efficient approach, utilising throwaway prototypings speed and efficiency, alongside evolutionary prototypings considerations of quality.

The basic process begins by establishing a baseline through evolutionary prototyping. Once this baseline is established the features are implemented as "quick-and-dirty throwaway" (1992, p.74) prototypes "on top of the quality baseline software". This aspect is crucial in that it allowed work and construction to begin on the interactive design process as soon as possible, avoiding "time on work that" (Chandler & Chandler, 2011, p.135) would otherwise "be discarded", thereby saving both time and resources. Once features are deemed useful they are implemented to a suitable standard within the original evolutionary prototype, thereby establishing a new baseline. These stages are repeated indefinitely to assure quality control for the users, however I continued this process until the final visualization was satisfactory. The stages of software implementation and user testing were omitted as they were more suited towards work based tool implementation and not the continual development of an interactive visualization.

The operational prototyping framework was used in conjunction with Rudolf Kremers methods of linear and semi-linear level design from the book *Level Design: Concept, Theory & Practise.* The use of linear and semi-linear level design, as defined by Kremers, was used to structure the interactive game levels once the base systems were established. Linear level design sets the user on a strict path where progression is only possible "if the player goes through the gameplay events in the order predetermined" (Kremers, 2009, p.56), whereas semi-linear allows the player to "direct their own experiences in some instances" (2009, p.56). This played a key role in determining how aspects of the game were structured. With the game divided into specific levels corresponding to a generation of console, these methods determined the playstyles of each level, whether they be open and free flowing, or more linear and restricted. These design choices were informed by the corresponding generation of home console, the data gathered during the research process and popular games of the respective times. This in turn played a key role determining how the user could interact and navigate the constructed games as well as how information was revealed to the user.

It should be noted that Edward Tuftes discussions of "Aesthetics and Technique in Data Graphical Design" (Tufte, 2001) alongside Mark J. P Wolf and Carl Therrien's discussions of graphical evolution in video games from the respective chapters of "Z-axis Development in the Video Game" (Wolf, 2009) and "Graphics in video games" (Therrien, 2008) were taken into account. Though neither necessarily outlines specific methods, each strongly informed the aesthetic choices of the visualization, with Wolfs work playing a major role informing both the visual and interactive design of the game components as hardware development shifted game aesthetics from 2D to 3D. Tuftes work in the formatting and discussions of "attractive displays of statistical information" (Tufte, 2001, p.177) were key when sorting and filtering through extracted data, outlining methods with which to display information in ways that were "particularly accessible and open to the eye" (2001, p.183). Wolf and Therriens work highlights both the graphical evolution of video games and development of real time 3D imagery within the video game medium. Their detailed focus on these aspects gave my own designs precedents from which to work from, ensuring I, as faithfully as possible, recaptured the graphical and technical qualities of each generation of home console history.

Figure 2.4. Linear level structure, as outlined by Kremers. (Kremers, 2009, p.56)

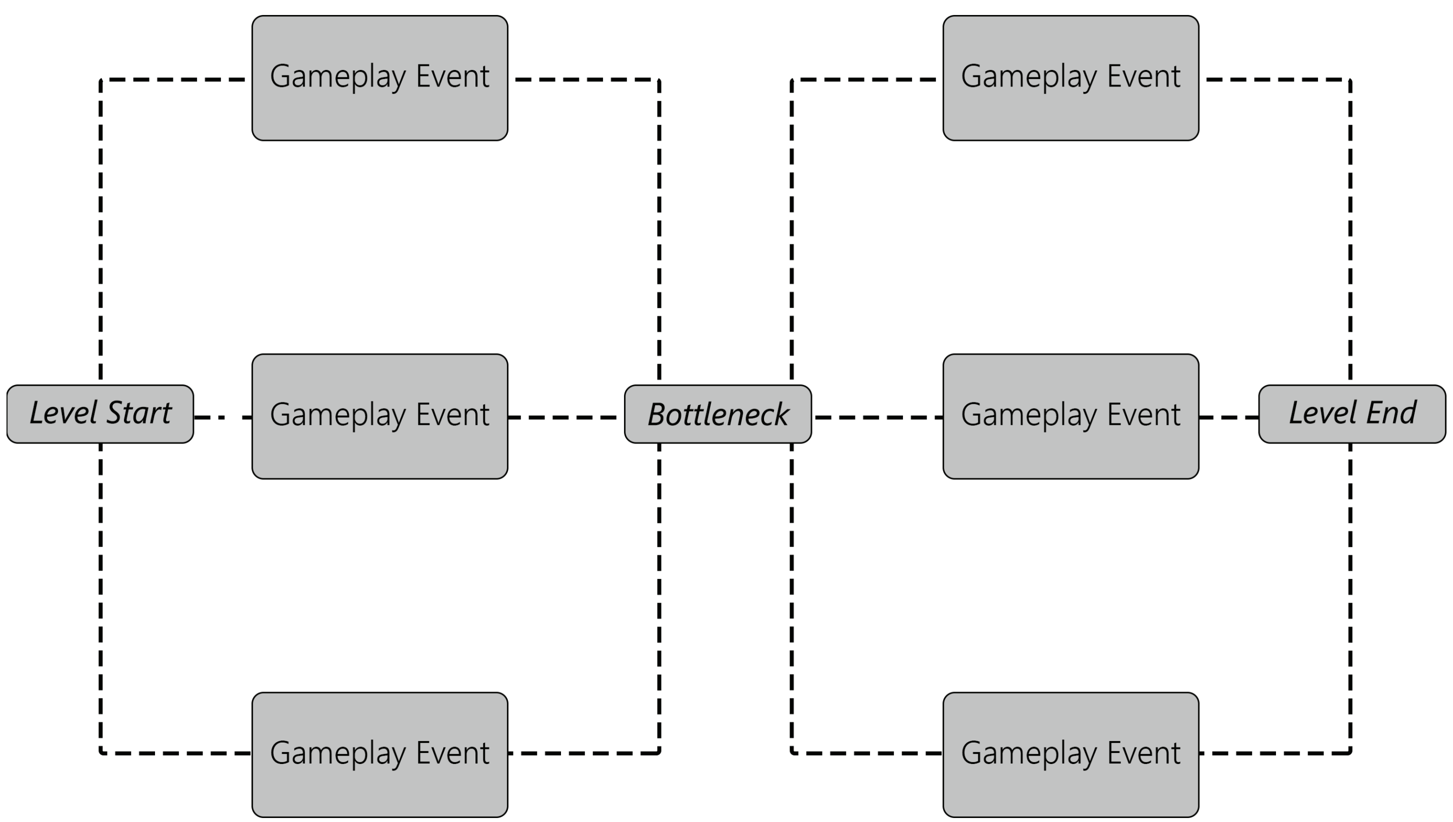
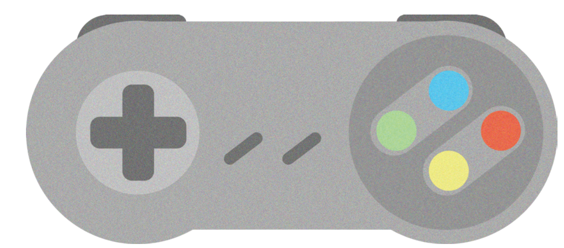


Figure 2.5. Semi linear level structure, as outlined by Kremers. (Kremers, 2009, p.56)

Chapter 3

LITERATURE REVIEW



Data Visualization within Video Games

Video games are a broad topic with literature that covers a wide range of fields, with the evolution, development and general history of video games being discussed on numerous occasions and in many different contexts. Data visualization can be defined as a general term which describes the process in which information is made visual so that its significance is easier understood. Interactivity with regards to video games, in Wolfs "Assessing Interactivity in Video Game Design", can be defined as the "player's choices" which "determine the course of the game" (Wolf, 2006, p.79) . The following literature focuses on the history of video games, some of their key technological developments and their overall influence, while also highlighting and analysing instances where video games have met with the topics of data visualization and interactive design.

Literature concerning visualization related to video games is discussed in varying contexts, however three common themes appeared. The first being the use of video games and video game engines to create visualizations to serve a purpose in another field, the second being the analysis of information visualizations commonly seen in video game user interfaces and the third being the extraction of video game data and its eventual visualization.

In 2008, Friese, Herrlich and Wolter highlighted attempts to map out geographical and archaeological data using various game engines, with each of these resulting in outcomes of differing quality. Their paper discusses three separate instances where video game engines were used to interpret gathered data and from that data construct real time 3D visualizations. Two of these constructions were of the interior space of cave sites with the third being of a landscape visualization. Each of these made use of different game engines, the first using the Quake 3 engine (released in 1999), second the Unreal Tournament 2004 engine (2004), and the third the Far Cry engine (2006) to build real time visualizations.

In 2014, Herwig and Paar also proposed a similar use of game engines for the purpose of visualization, in this instance the visualization of landscape architecture. The concept of high quality visualizations as a means of planning landscapes is common throughout this field with the use of 3D CAD software. However the pair felt that landscape architecture "does not take full advantage of the rapid developments of computer graphics" (Herwiig & Paar, 2014, p.2). The pair noted the use of game engines could allow for 3D and real time movement, offering "immediacy in the medium" (2014, p.3), as opposed to the limited pre rendered walk or fly through common in Architecture at the time of publication. Herwig and Paars methods remained similar to Friese, Herlich and Wolters (2008) as terrain data was prepared and imported into the Unreal Engine.

Over time game engines become have become more capable, as processing power continues to develop and increase, thereby improving the quality of the visualizations, however, both parties discovered each engine had limitations and therefore the visualizations were restricted. These restrictions included limited polygon count, and a lack of compatibility between their tools and the game engine. Most critically Friese et al (2008) found that the lack of documentation for the engine coupled with its relatively short life span limited its use stating "the visualization of scientific data is possible and leads to promising results [but] one should always keep the drawbacks mentioned" (2008, p.21) in consideration. Herwig and Paar felt their implementation of interactivity was also limited. Both parties came to the similar conclusions that a game engines potential for visualization could outweigh the drawbacks and limitations considering the low cost of the engines. Herwig and Paar would note that "it is promising to further explore and test impacts and possibilities of computer graphical assisted landscaping while the technology is further advancing"(Herwiig & Paar , 2014, p.10).

Each of these articles offered a precedent to work from, with each outlining their own process and workflow for utilising data within video game engines. Each of these would also highlight the challenges they faced and how they overcame these. The research undertaken here informed my own work and allowed me to develop my own approach based on their design and development processes. Their analysis of the technical limitations in particular offered great insight into how I developed my own designs and prepared me for the various difficulties that I encountered throughout the design process.

A 2011 article by Medler and Magerko highlights the second context of visualization related to video games. They discuss the analysis of play and examine the information visualization that takes place within games and how these visual experiences can promote or encourage gameplay. Their study highlights the concept of "playful visualizations" (Medler and Magerko, 2011, p.1) and how these visualizations can allow one to "play with data" (2011, p.1). A key example of this being FIFA Earth, an info Visualization tool developed for Electronic Arts FIFA 10 video game. The tool can calculate the total score of players within a country and determine how each country is faring within the game, ranking each accordingly and promoting friendly competition. They note that "aggregating the data of a country's players and comparing their wins created a separate medium through which competition was encouraged" (2011, p.6). In conclusion they noted how data "can create storytelling opportunities" and "cause competitive situations" (2011, p.11) relative simply to how it is displayed, whether through comparison to other sets of data or simply highlighting gathered sets.

The concepts of "playful visualizations" (Medler and Magerko, 2011, p.1) would inform my own data displays. Data displayed in a manner that encourages gameplay and allows users to "play with data" (2011, p.1) was an interesting concept I noted throughout my design process. To build data visualizations that are engaging and could "create storytelling opportunities" (2011, p.11) became a key focus with the intention being to make this information accessible and comprehensive. Though I did not necessarily intend to develop info visualizations that encouraged gameplay, there was a clear intention to make them engaging and that was informed by the work discussed in this article. However the potential for storytelling opportunities through data visualization was a concept that I noted as through my own visualizations I intended to create a narrative visual history of the home console.

Zammitto describes a similar example, highlighting the visualization of on screen information within games themselves, such as user interfaces and HUD systems. Noting how video games "rely extensively on visual information displayed to the players" (Zammitto, 2008, p.267), Zammitto states that certain visualization techniques are more suited to specific game genres (2008). She highlights genres such as the FPS, using Half Life 2 as an example, adopting a more simplistic minimal interface where "information is visualized only if the element, besides health, is being used or changed" (2008, p.269). This is due to the direct action focused gameplay, while an RTS game (real-time strategy), using Command and Conquer (Electronic Arts, 2007) as an example, has a focus on simultaneous tasks such as managing resources, commanding units and exploring the map, therefore "several information visualization techniques are implemented"(2008, p.273). Zammitto concludes that the differences in game mechanics and the requirement of varying information across video games highlights why certain visualization techniques are more appropriate. She concludes that information visualization in video game is not an especially developed field, "lacking an established terminology" (2008, p.275), but states that this can be altered through further research.

The highlighting of how information has been visualized within games would again inform how I chose to display my own gathered data. Though the article mainly discusses how in-game player information is displayed and what interface styles were used it does offer examples of the different types of aesthetics used. It is the demonstration of these aesthetics and how and when they were used that would inform my own visual designs for displaying information to the user.

The third context highlights the use of visualization in a more statistical analysis of video games. In 2008 Lizuka and Thawonmas proposed “an approach for visualizing players’ action behaviours using classical multidimensional scaling (CMDS)” (Lizuka & Thawonmas, 2008, p.1). Using CMDS the pair locates clusters of players before using action coding, dynamic warping and eventually Key Graph to discover the relationships among text based data. From the data they are able to see the varying types of players based on behaviour and then divide them into varying groups based on those behaviours (2008). They conclude by noting the importance of “understanding player behaviours” (2008, p.8) with regards to “improving the service of quality of online games” (2008, p.8). Though this study delves into areas far more relevant to computer science it still highlights a relationship between visualization and video games.

Though not as influential to my own designs, the research undertaken in this paper offered a perspective on data visualization and the development process unique to others I had read. The work discussed allowed me to view a design process and see what aspects I could utilise in my own work and how I could further my own designs. However the more statistical approach, its use of visualization tools and the differing intentions indicates a strong contrast to my own work. Therefore little of their work was taken into account with regards to my

Virtual Navigation and Video Game Evolution

own designs though it remained useful to understand how visualization had been applied previously.
The previously discussed chapters focused on areas where video games had been utilised within the fields of data and information visualization. The following chapters highlight areas informing the visual and interactive aspects of the thesis in a stand-alone context, focusing solely on their topics of video games and interaction, with no association to the field of data visualization. The subsequent papers discuss concepts of navigation in virtual space, the evolution of video game graphics and hardware, and the resurgence of vintage styled video games.

Lev Manovich’s discussions of navigable space in *The Language of New Media* highlights visualization in a way distinct from the previous sources, while also highlighting innovations in technology and methods of interactivity. In his work he identifies the video games Doom (id Software, 1993) and Myst (Cyan inc, 1993) as contrasting examples of navigable space. Discussing the differences in their aesthetics of navigation, Manovich points out that in “Myst the navigation is more free form” in which the player or user is “slowly exploring the environment” (2001, p.1). This contrasts with the faster paced navigation of Doom, where the “player is moving in straight lines” and “abruptly turning at right angles to enter a new corridor” (2001, p.1). The highlighting of navigable space shows methods of moving through and working with 3D visualizations of all forms. This demonstrates more than the control of a video game but the navigation of virtual space, where “navigable space can be legitimately seen as a particular kind of an interface to a database” (1998, p.7). The concept of navigable space being an interface to a database indicates the possibilities of data being housed and visualized in playable or interactive formats. Much like the environments or worlds of Doom and Myst, data visualization, with this concept in mind, can be seen to move beyond the standard user interface.

This book chapter’s discussion and the concept of navigable space played a crucial role in the development of my video game component. The comment of navigable space as an “interface to a database” (2001, p.7) was a key concept, one I kept in mind throughout the development of my own game and visualization outputs. The games I built were developed to act as both visualizations and an interactive interface allowing one to view and navigate the gathered historical data. This concept played a major role, informing my use of video games as a tool for the navigation of gathered data, as opposed to simply developing a user interface to view data.

It should be noted that the books *The Video Game Theory Reader 2* and *The Video Game Explosion A History from PONG to PlayStation and Beyond*, both edited by Mark J.P Wolf, were often points of reference throughout the research and design process. Each held a great wealth of knowledge regarding video games, their history, their influence and their innovations. These books played an integral role in my research as a multitude of the authors, including Brett Campers, Carl Therrien, Leonard Herman, Dominic Arsenault and Eric Pidkameny, and their respective chapters were all utilised throughout the research and design process, with all informing the construction of my design output. The following chapters highlighted are those that played a notably more significant role informing my own work. Brett Campers chapter of “Retro Reflexivity La Mulana, an 8-Bit Period Piece” in Wolf and Perron’s *The Video Game Theory Reader 2* discusses the rise of retro style video games and makes interesting observations with regards to interactive design, noting a trend where retro style games adhere to earlier home consoles visual and technical limitations. This is done while adopting “a style of play that does not simply replicate its classical models, but adapts and evolves them” (Camper, 2009, p.181). The article highlights La Mulana (GR3 Project, 2005), an independent game built in the style of the MSX, and how it improves upon the flawed designs common in games of the time period. Milons Secret Castle (Hudson Soft,

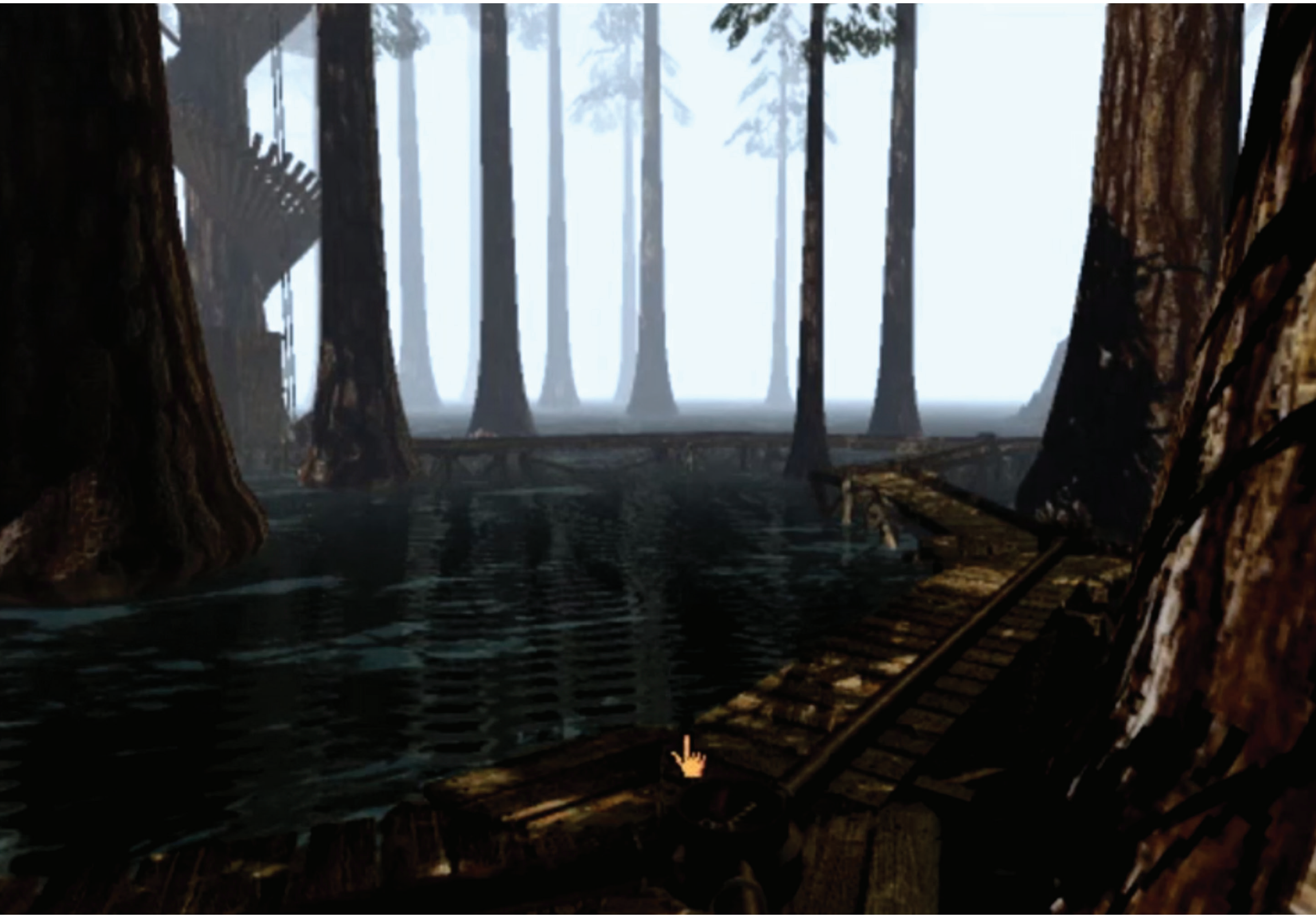


Figure 3.1. The more free form navigation of Myst (Cyan Worlds, 1993).

Figure 3.2. The faster paced forms of navigation in Doom II: Hell on Earth (ID Software, 1994).

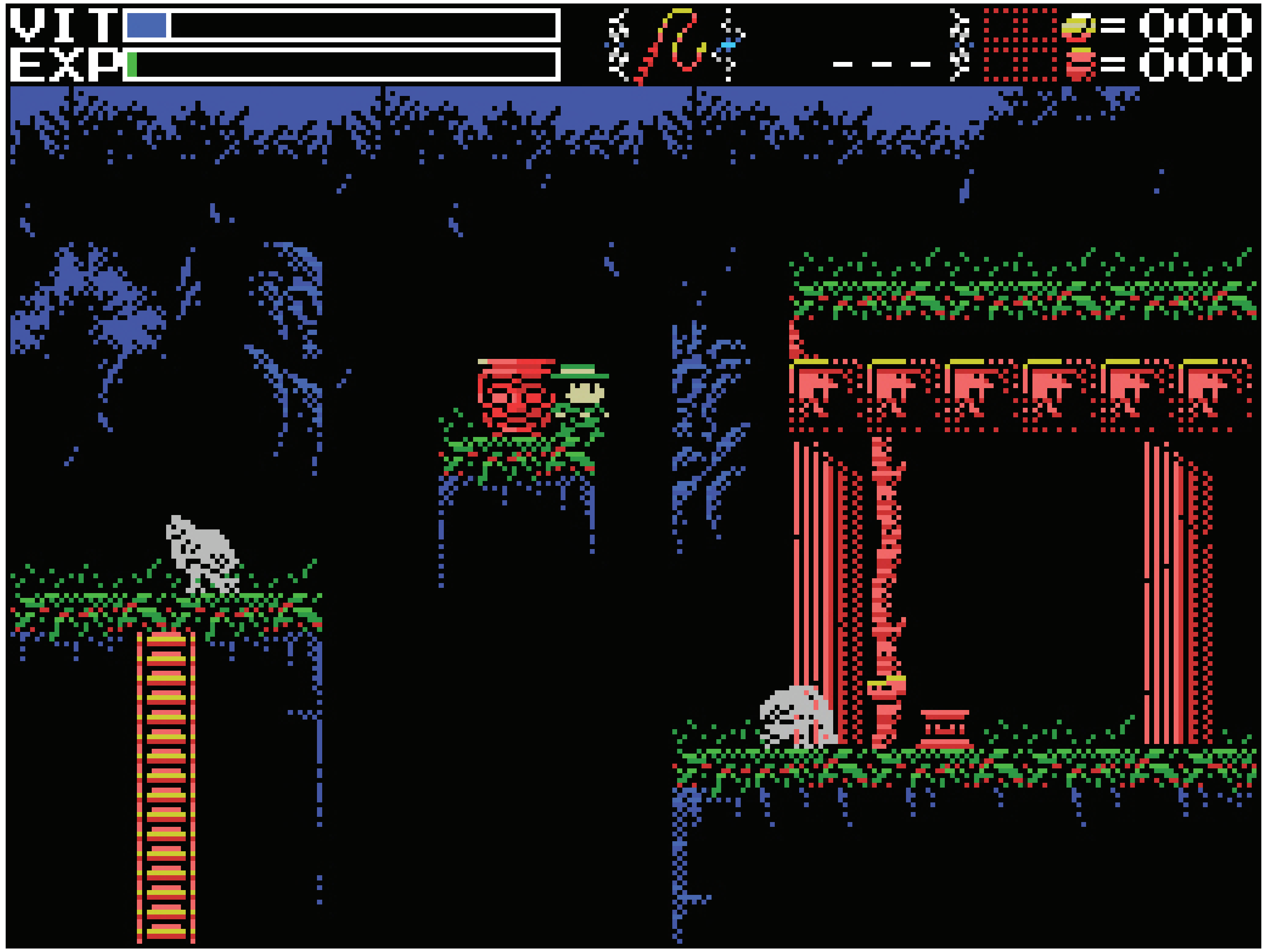


Figure 3.3. The retro styled aesthetics and gameplay of La-Mulana (GR3 Project, 2005).

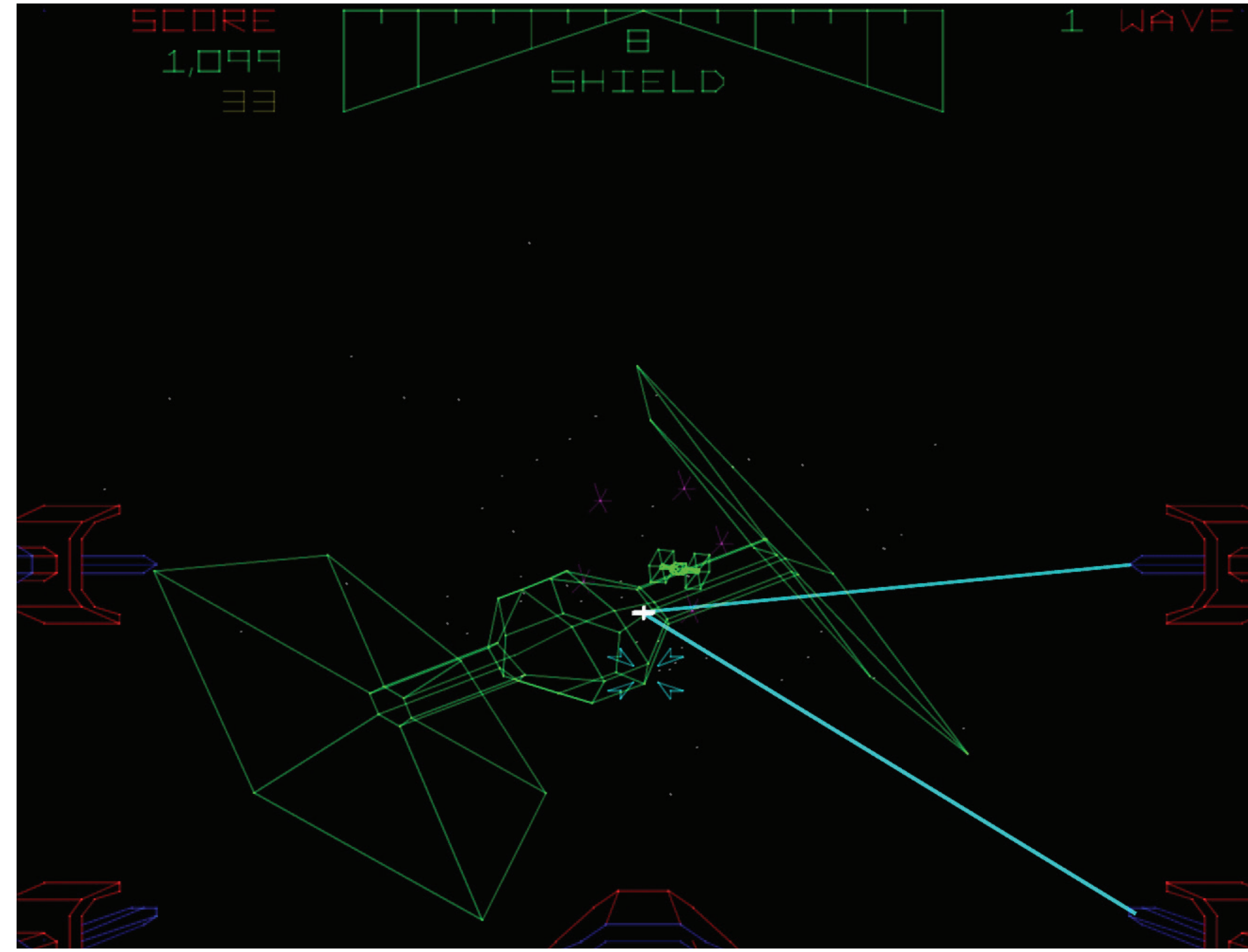


Figure 3.4. The arcade based vector graphics of Star Wars (Atari, 1983).

1988) was a prime example of this flawed construction. It was a game known for its difficulty and challenge, however these were predominantly due to its illogical and frustrating design choices. Camper highlights the powerful nostalgia that gamers have developed for games they have grown up with, which in turn has led to independent developers responding "with a trend of 'retro' styled-but original-video games" (Camper, 2009, p.170). It is interesting to note however that in their attempts to replicate the games of their youth, the influence of modern game design has led developers to "evolve past" (Camper, 2009, p.183) the traits of older games.

The outlined development of La Mulana played a key role informing both the visual and gameplay development of my own games, as well as the design process as a whole. La Mulanas adherence to visual and technical limitations of the time was a key influence as I also followed the limitations of each console generation as closely as possible. Each game I built was developed to appear as or represent a game of a specific time period. However, unlike La Mulana which attempted to improve upon the flawed designs of earlier games; I attempted to replicate the game mechanics of each generation as closely as possible. Where La Mulana attempted to "evolve past" (Camper, 2009, p.183) its key influences I tried replicate these as closely as possible.

In "Z-Axis Development in the Video Game", Wolf focuses on key technological advances of hardware, highlighting the evolution and implementation of 3D imagery within gaming. Even in the earliest forms, he states, games were attempting 3D imagery, such as PONGS (Atari, 1972) "white ball and paddles" (Wolf, 2009, p.152) appearing "in front of the black background" (2009, p.152) while graphics and the suggestive use of vanishing points were employed in others. It is interesting to note the development of technologies and how aesthetics would develop or regress relative to their hardware, an example being vector graphics. Vector graphics were one of the earliest methods to produce real time 3D graphics in gaming. Arcade games such as Speed Freaks (Vectorbeam, 1979), Battle Zone (Atari, 1980) and the Star Wars (Atari, 1983) arcade cabinet all utilised this technology. However vector graphics would eventually fall out of favour as "three-dimensional computation began appearing with filled polygons" (2009, p.156). Wolf concludes that as video games pursuit of photorealism continues the medium remains at the forefront of interactive imagery as games continue to create virtual spaces for players to enter and be convinced of these virtual worlds. Wolfs discussions of the development of 3D graphics again highlights designing within limitations, however, contrary to Campers work, we are shown interactive design choices made out of necessity, not necessarily by choice. An example of this would be the design process of "breaking up a games world into sectioned spaces" (2009, p.165) and "dividing interiors into rooms and hallways" (2009, p.165). This was a common feature of the Tomb Raider and Silent Hill series in an effort to conserve processing power.

Wolf's work played a major role informing all facets of

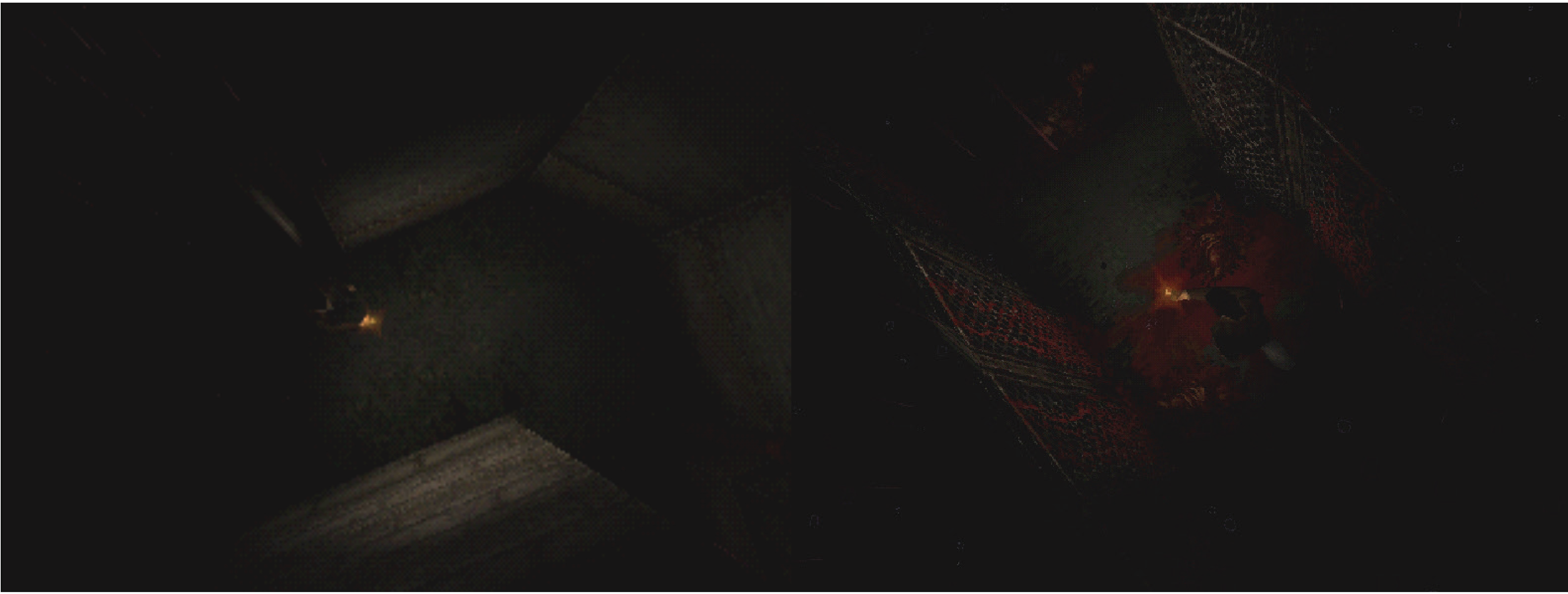


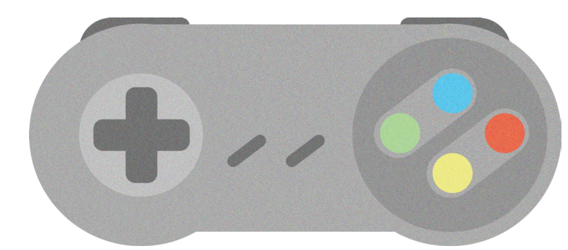
Figure 3.5. The dark and enclosed spaces of Silent Hill (Konami, 1999), demonstrating the dividing of "interiors into rooms and hallways" (Wolf, 2009, p.165).

each game I developed and the overall design output. The discussions of graphical and technical innovations and how they determined a games visual appearance would influence my projects designs from a visual, gameplay and technical standpoint. The article highlights key limitations from certain eras and what methods and techniques were used to overcome or hide these, with a key example being the use of fog in early 3D games. This information allowed me to more faithfully recreate the visual styles of each generation and develop more convincing games and visualizations. This chapter was often used as a reference point throughout the projects development to ensure that the games developed successfully matched the generation they represented. The highlighting of technical innovations also influenced how some games were programmed as I would attempt to replicate some of these features as closely as possible within the limitations of the Unreal Engine.

The literature discussed in this chapter does not discuss all that has been covered in either the fields of data visualization, interaction design or video game history. However it discusses and highlights instances in which these three topics have shared a relationship, or been utilized in a manner that is relevant to the research question. The above literature gives an overview of what work has been attempted, reviewed or discussed in relation to the subject matter of the thesis and allows one to gain an understanding of what has come before and acts as the basis of this papers research.

Chapter 4

SITE ANALYSIS



Thesis Outline and Considerations

This investigation was primarily split into two components. The first of these components was the research and extraction of data from previously described resources (Wikipedia, journals, books, webpages and forums), while the second was the construction of the interactive visualization. The reason for this was that the development of the interactive components could not begin without having the gathered data already established, as the interactive systems were built around this information. Site analysis then led me to focus on each of the two stages. The first stage being the resources from which the data was extracted. There are numerous amounts of resources detailing video game history, ranging from books, journals, online pages and websites, documenting its development in great detail and highlighting the mediums evolution. This leads one to question what information should be used and what should not. As stated previously a complete catalogue of home console history, with the allocated time and resources considered, was not feasible. Therefore, it was decided to make a selection based on what would give a more functional and complete overview of the topic, allowing for the construction of an interactive application that could be at once selective in content at the same time as it could be comprehensive in its reference to key developments in game history. To keep the research within scope I limited my research focus strictly to the statistics of consoles and games of each time period, while highlighting technical and visual developments where appropriate. The adherence to the text mining framework and knowledge extraction process played a key role in the sorting and selection of appropriate information.

The interactive application also had to be approached with the previously mentioned limitations in mind. Video games are an inherently interactive visual medium; therefore the use of interactive design and data visualization would appear to be a natural choice. What better way to learn about the evolution of home consoles than by simply interacting visualized video game data, the construction of a game that educates on and preserves its own history and legacy? As stated previously, this was undertaken utilising the Unreal Engine 4 software. Again, however, there were varying factors to be taken into account with the use of this software. The restrictions in this area bordered more on the technical and production details, however these were aspects that would greatly influence the design process therefore they had to be taken into immediate consideration.

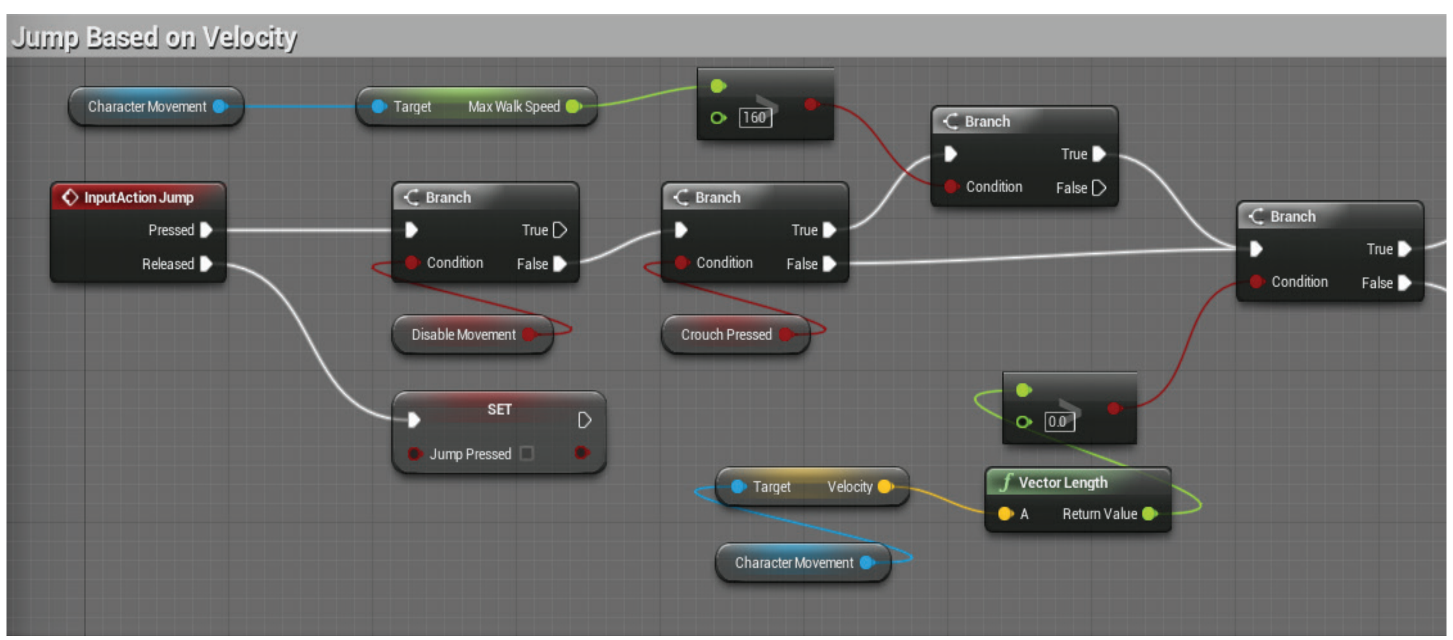


Figure 4.1. An example of the blueprint system within Unreal Engine 4.

I am familiar with the Unreal Engine 4 software and have worked with it previously; therefore, from experience, the first limitation to consider was processing power, both my own as the developer and the users. Unreal Engine can be taxing on systems, with this and the user in mind the output would need to be developed to a standard that will allow it to run on a majority of PC setups. With regards to programming Unreal Engine also houses its own visual code structure called Blueprint. Though Unreal is compatible with the C++ programming language I would be using the visual Blueprint structure as I have prior experience with it.

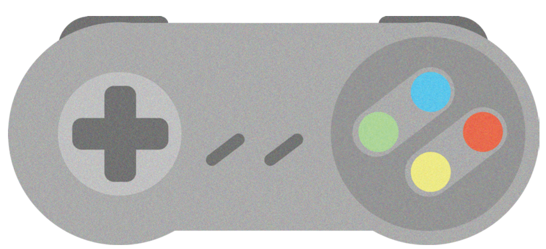
Another key factor to be considered is that the Unreal Engine does not contain all the required tools to develop the running application as intended. Game design does not only rely solely on the use of game engines but requires the input of other software. 3D modelling software, animation software, audio and image editing software, all of these are necessary for production and would be used in all stages of the design process. This shows that the design process would be spread across multiple software tools and that one must be versatile and prepared to adapt to the needs of the development process.

It should also be noted that the Unreal Engine is a modern engine used for current generation systems. It is an engine suited to the development of modern console and PC video games. With this considered, a majority of the tools offered by the engine are more suited to the development of modern games as opposed to the retro styled aesthetic. Therefore the games developed for this thesis were not completely authentic representations of the times due to the differing hardware and software utilised, however the games are as authentic a representation as can be developed within the engines capabilities.

Though much of what has just been highlighted would influence and affect the design process in some way shape or form, it is a combination of these and the limited time and resources to carry out this investigation that would truly shape the project.

Chapter 5

PROJECT REVIEW



Game Engines for Visualization

The following project review highlights studies in which video game software or engines were used for the purpose of visualization. Both used game engines in different fields of research, with one in landscape architecture, and the second in archaeological research. Each case attempted to use game engines as a cheaper more efficient alternative to architectural visualization or animation software. However, though each yielded fairly favourable results, each study found both advantages and disadvantages to their use. With regards to my own work, the following studies offered a precedent design process to build from. Though my methods already outlined a base template to build from, the subsequent papers offered unique insights to the visualization process and their use of video game engines subsequently informed my own research.

Herwiig and Paars "Game Engines: Tools for Landscape Visualization and planning?" published in 2004, focussed on the "options and limitations of game technology" (Herwiig & Paar, 2014, p.1) and questioned how they could be implemented within the "collaborative landscape planning process" (2014, p.1). Herwiig et al noted that "3D game design and landscape planning share the interdisciplinary approach and the need of visualization" (2014, p.2). The pair felt that computer games could be seen as the "draught horse" (2014, p.1) for landscape visualization.

Using the rural areas of Strausberg, Herwiig and Paars used Unreal Engine 3 to create landscape visualizations of the site. Terrain data was prepared using other software before being properly formatted so that it would import to Unreals level editor. Photographic textures were applied to the geometry while other assets, such as trees and other vegetation, were modelled and imported to the scene. There were notable limitations however with higher amounts of geometry and higher poly counts leading to drops in the frame rate, thereby losing the "impression of continuous movement" (2014, p.7).

Herwiig and Paars found during the process that trial and error was "often necessary to import terrain, 3D objects and other external data" (2014, p.9). They also felt that "a fundamental knowledge in 3D designing" (2014, p.9) was necessary to effectively use the 3D game engine and ensure that engine restrictions and performance were taken into consideration. Regardless of these issues, the pair still felt the potential of game engines in the field remained concluding that with game engines low cost and optimization they remained an interesting alternative to professional landscape software.

A key focus here is the pairs use of the Unreal engine. Their techniques and methods would be notably similar to my own. The restrictions encountered here, specifically the limitations regarding performance and geometry played a major role influencing how I would design as these are issues I was aware I would face. Their solutions for optimization were techniques I took into account and used myself when building my own levels throughout the design and development process of my game components.

Friese, Herrlich and Wolters work highlighted three projects undertaken by students where game engines were utilized to visualize archaeological data. Their paper "Using Game Engines for Visualization in Scientific Applications", published in 2008, documents the use of three different engines over separate instances to visualize both the interior space of caves, based on laser scan data, and as a landscape visualization tool. Their study would analyse the potential benefits and limitations of each software. Discussing the three projects undertaken, the first two used man made caves as a base while the third focused on landscape visualization. With the first two projects, the original intention was to create a re-construction as accurate as possible of the caves with the intention of then exporting these reconstructions to CAD software.

The first approach utilized the Quake 3 engine, an engine that at the time of use was relatively old and had been available for a number of years. Developed by ID Software the Quake 3 engine supported "shaders, curved surfaces, 32-bit color" (Friese, Herrlich & Wolter, 2008, p.15) and hardware rendering. Upon the construction of geometry severe limitations were found with the amount of triangles that could be generated. Closed surfaces were implemented to maintain a low polygon count, as the "maximal number of triangles was limited" (2008, p.16), and keep the visualization within the engines restrictions.

The second project would use Unreal Engine 2. Unreal Engine 2 supported the PC, PS2 and XBOX and was a far more capable engine, able to support resolutions up to 2048 x 2048, dynamic light sources and up to 150, 000 triangles in view at any one time (2008, p.17). Similar to the first project the gathered data was converted to be compatible with the engines level editor. The scan data originally "consisted of several thousand triangles for the cave surface" (2008, p.13) leading to high computational costs for the engine. The solution required was to reduce the number of triangles. Again limitations were found and had to be overcome through "trial-and-error methods" (2008, p.18).

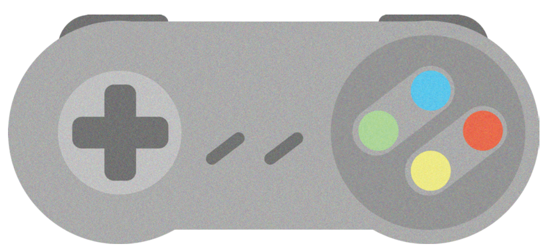
The third project highlighted in the study focuses on landscape visualization and planning using the CryEngine. The intention of this project was to build "the prototype of a visualization tool for landscape architects" (2008, p.18) by applying the "visualization and interaction capabilities of modern game engines" (2008, p.18). Again the benefits focused on the "low additional cost" (2008, p.18) game engines. This project used the CryEngine as it supported large outdoor areas while providing a "comprehensive set of tools for terrain shaping and vegetation placement" (2008, p.19). Again geographical data was converted into a format recognized by the engine, before being altered and modified in real-time, generating a map that could be used in-game. The study found the results to be successful, with "convincing results concerning realism and visual quality" (2008, p.20). However this project was also not without its drawbacks. This project encountered more difficulty with the data conversion process than previous studies had found, as a manual conversion process was required so that data could be used within the engine.

Friese at al came to the overall conclusions that the use of game engines for the purpose of visualization was very much a matter of preference. Citing the major advantages as the "state of the art graphics" (2008, p.20), high optimization and price. The low budget acts as the biggest advantage when compared to the "10, 000 US$ per copy" (2008, p.20) professional visualization tools, however they note the serious flaws that one may encounter. Limited documentation, engine restrictions, short life span and inability to do more than what the games can do are cited as the major issues concerning the use of this video game engines. The software becomes a viable option as long as the above restrictions are considered and the potential of the project does not greatly exceed the capabilities of the engine (2008, p.21). Regarding my own research, the area of this paper I found most insightful was their highlighting of the data conversion process and making data compatible with their engine. Prior to development I had predominantly focused on the development of the game component, overlooking how I would import not only data, but general assets into the Unreal Engine. However I was fortunate enough that the engine provides a multitude of methods for the importing of required assets. Regardless I would still come to find issues when importing various 3D rigs and models as well as real time physics data. Utilising their trouble shooting methods and by finding appropriate formatting I was able to find a way around these issues.

Each of these studies demonstrates the application of both my research aims, with each project compiling gathered data before developing the data into an interactive application. Each project also highlights the potential for game engines to serve in the field of visualization, highlighting the possibilities and benefits that this type of software can offer outside of the gaming medium. The documentation of their experiences and their process provided an invaluable foundation from which I structured my own design process. Though the engines they were developing with are older iterations, with far more limitations than I encountered, the approaches and techniques that were applied in their studies remained relevant. Processing power and limitations of both software and hardware remained a defining factor during my development and had to be carefully considered at all times. Therefore the optimization techniques undertaken in both studies, though out of necessity, were of crucial importance to my own research and prepared me for my own interactive development. Their hardware concerns also influenced my own hardware considerations as well as the users; because of this optimization was an early component of the design process. These studies would leave me with a strong understanding of what must be done to achieve my research goals and to complete the design process in an effective and timely manner.

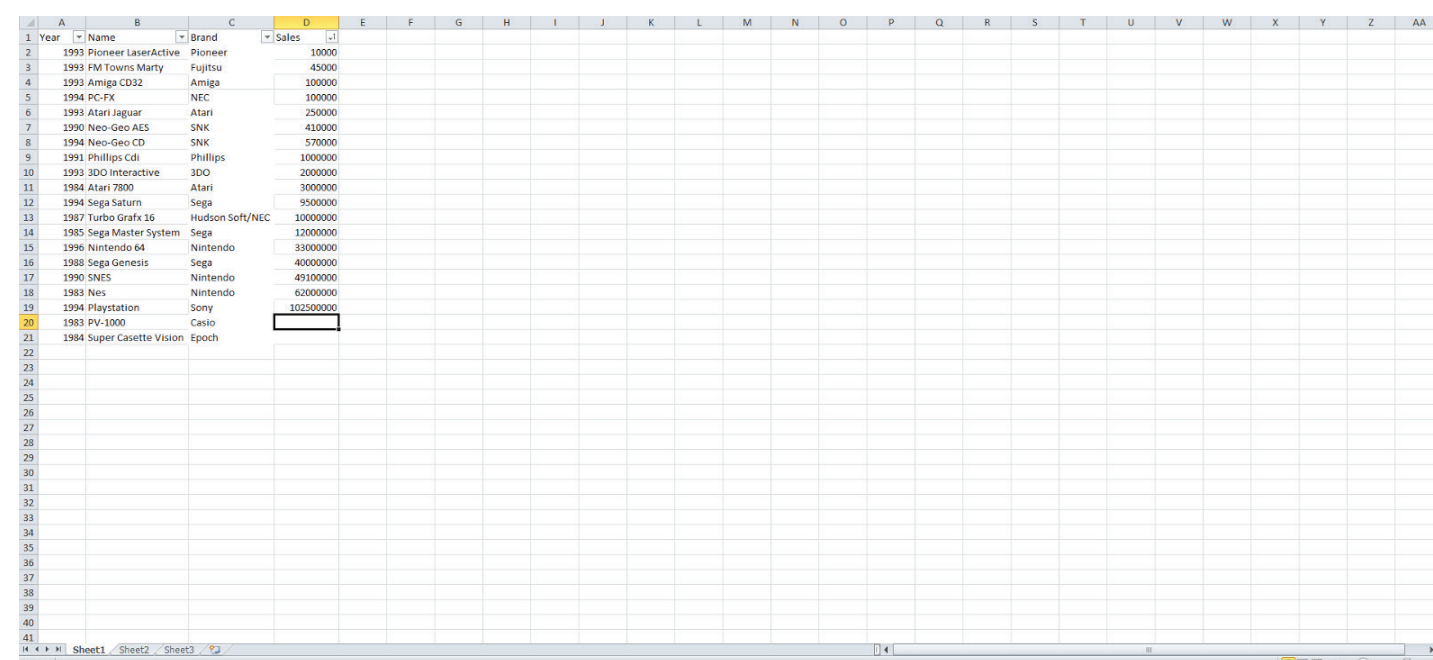
Chapter 6

PRELIMINARY DESIGN



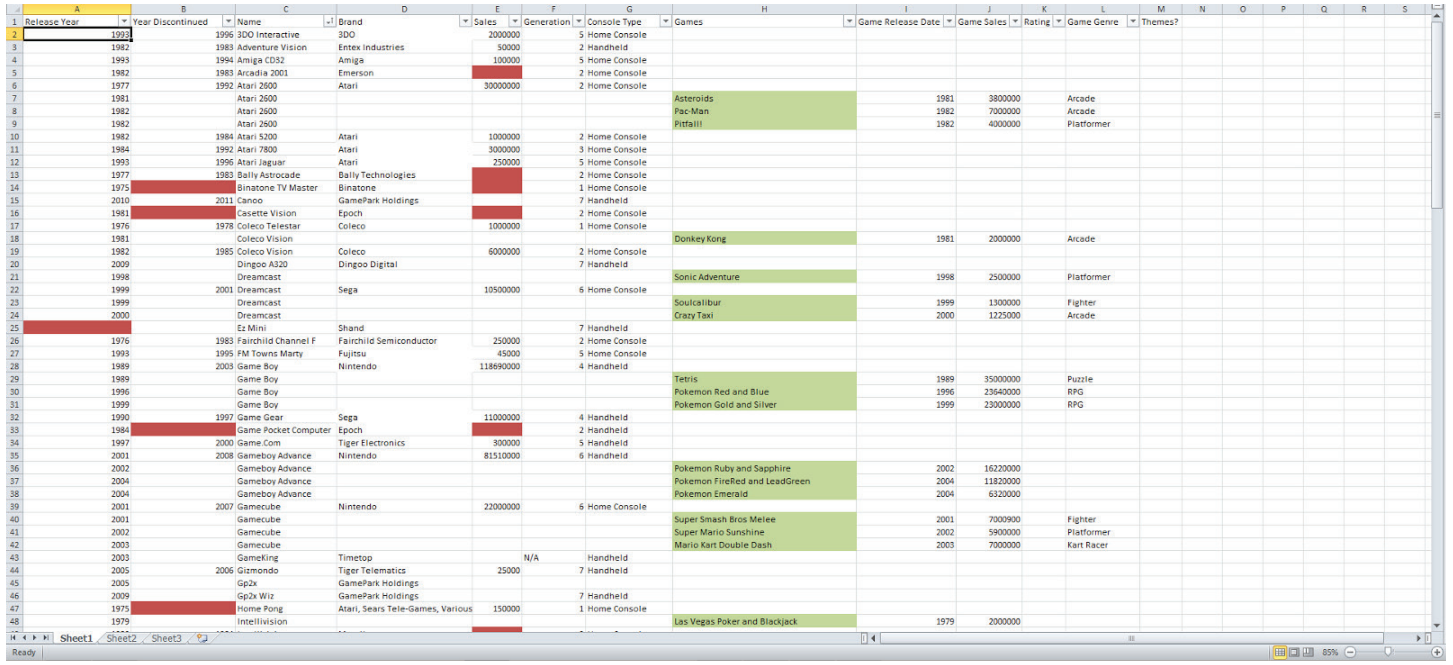
Text Mining and Knowledge Distillation

The design process began with the collection of data. Following the text mining framework I started with the process of text refining, where information relating to video game home consoles was gathered from text documents. These documents encompassed various readings, websites and books. This information was extracted into a concept based intermediate format (the concepts in this case being the separate home consoles such as the Atari 2600 or the Sony Playstation) and stored in an excel spread sheet. All gathered data was built up over time and eventually sorted through the process of knowledge distillation, where "patterns or knowledge from the intermediate form" (Tan, 1999, p.2) are deduced. The process of knowledge distillation began with the adding of filters to the excel spread sheet, allowing one to sort the data according to various categories that would be added later. Much of the initial data was gathered from Wikipedia as it contained specific information such as release dates, unit sales, games sales and technical specifications. Other data was gathered from the works of Wolf's books *The Video Game Explosion* and *The Video Game Reader 2*, with some data taken from forums and various websites. During the research it became notable that some data could not be found at all. For instance, information regarding more obscure or less successful consoles was notably difficult to find. Though there would be documentation indicating certain consoles did indeed exist and were sold, in some instances very little data to none at all could be found, with the exception of either a Wikipedia page or obscure web page. An example of this being the Neo Geo AES, though not as obscure as other consoles, its sales figures could not be confirmed, with only forum estimates to go by. Its Wikipedia page noted only a combined sales figure of the Neo Geo MVS, AES and CD at over one million units sold, therefore a definite number could not be found. Another notable example is the Casio PV-1000. Released in Japan in 1983, the PV-1000 was reportedly removed from distribution not long after its initial release (Video Game Console Library, n.d.). Due to this short period of release it is noted for its rarity and very little data is available concerning its sales.



Year	Name	Brand	Sales
1993	Pioneer LaserActive	Pioneer	10000
1993	FM Towns Marty	Fujitsu	45000
1993	Amiga CD32	Amiga	100000
1994	PC-FX	NEC	100000
1993	Atari Jaguar	Atari	250000
1990	Neo-Geo AES	SNK	410000
1994	Neo-Geo CD	SNK	570000
1991	Phillips Cdi	Phillips	1000000
1993	3DO Interactive	3DO	2000000
1984	Atari 7800	Atari	3000000
1994	Sega Saturn	Sega	9500000
1987	Turbo Grafx 16	Hudson Soft/NEC	10000000
1985	Sega Master System	Sega	12000000
1996	Nintendo 64	Nintendo	33000000
1988	Sega Genesis	Sega	40000000
1990	SNES	Nintendo	49100000
1983	Nes	Nintendo	62000000
1994	Playstation	Sony	102500000
1983	PV-1000	Casio	
1984	Super Casette Vision	Epoch	

Figure 6.1. Text refining being undertaken within an excel spreadsheet.



Release Year	Year Discontinued	Name	Brand	Sales	Generation	Console Type	Games	Game Release Date	Game Sales	Rating	Game Genre	Themes?
1993	1996	3DO Interactive	3DO	2000000	5	Home Console						
1982	1983	Adventure Vision	Entex Industries	50000	2	Handheld						
1993	1994	Amiga CD32	Amiga	100000	5	Home Console						
1982	1983	Arcadia 2001	Emerson		2	Home Console						
1977	1992	Atari 2600	Atari	30000000	2	Home Console						
1981		Atari 2600					Asteroids	1981	3800000		Arcade	
1982		Atari 2600					Pac-Man	1982	7000000		Arcade	
1982		Atari 2600					Pitfall!	1982	4000000		Platformer	
1982	1984	Atari 5200	Atari	1000000	2	Home Console						
1984	1992	Atari 7800	Atari	3000000	3	Home Console						
1993	1996	Atari Jaguar	Atari	250000	5	Home Console						
1977	1983	Bally Astrocade	Bally Technologies		2	Home Console						
1975		Binatone TV Master	Binatone		1	Home Console						
2010	2011	Canoo	GamePark Holdings		7	Handheld						
1981		Casette Vision	Epoch		2	Home Console						
1976	1978	Coleco Telestar	Coleco	1000000	1	Home Console						
1981		Coleco Vision					Donkey Kong	1981	2000000		Arcade	
1982	1985	Coleco Vision	Coleco	6000000	2	Home Console						
2009		Dingoo A320	Dingoo Digital		7	Handheld						
1998		Dreamcast					Sonic Adventure	1998	2500000		Platformer	
1999	2001	Dreamcast	Sega	10500000	6	Home Console						
1999		Dreamcast					Soulcalibur	1999	1300000		Fighter	
2000		Dreamcast					Crazy Taxi	2000	1225000		Arcade	
		Ez Mini	Shand		7	Handheld						
1976	1983	Fairchild Channel F	Fairchild Semiconductor	250000	2	Home Console						
1993	1995	FM Towns Marty	Fujitsu	45000	5	Home Console						
1989	2003	Game Boy	Nintendo	118690000	4	Handheld						
1989		Game Boy					Tetris	1989	35000000		Puzzle	
1996		Game Boy					Pokemon Red and Blue	1996	23640000		RPG	
1999		Game Boy					Pokemon Gold and Silver	1999	23000000		RPG	
1990	1997	Game Gear	Sega	11000000	4	Handheld						
1984		Game Pocket Computer	Epoch		2	Handheld						
1997	2000	Game.Com	Tiger Electronics	300000	5	Handheld						
2001	2008	Gameboy Advance	Nintendo	81510000	6	Handheld						
2002		Gameboy Advance					Pokemon Ruby and Sapphire	2002	16220000			
2004		Gameboy Advance					Pokemon FireRed and LeadGreen	2004	11820000			
2004		Gameboy Advance					Pokemon Emerald	2004	6320000			
2001	2007	Gamecube	Nintendo	22000000	6	Home Console						
2001		Gamecube					Super Smash Bros Melee	2001	7000900		Fighter	
2002		Gamecube					Super Mario Sunshine	2002	5900000		Platformer	
2003		Gamecube					Mario Kart Double Dash	2003	7000000		Kart Racer	
2003		GameKing	Timetop		N/A	Handheld						
2005	2006	Gizmondo	Tiger Telematics	25000	7	Handheld						
2005		Gp2x	GamePark Holdings									
2009		Gp2x Wiz	GamePark Holdings		7	Handheld						
1975		Home Pong	Atari, Sears Tele-Games, Various	150000	1	Home Console						
1979		Intellivision					Las Vegas Poker and Blackjack	1979	2000000			

Figure 6.2. The filtering and categorizing of data through knowledge distillation.

Beginning the process of knowledge distillation, where data is "organized according to their content, for visualization and navigation purposes" (Tan, 1999, p.3), filters were applied in the excel spread sheet to sort the gathered data according to applied categories. The chosen categories were common factors shared amongst all home consoles. The initial sets of categories were release years, discontinuation years, console manufacturers, unit sales and console generation (the era of console), as this data, in most cases, could be found consistently. This is a key concept as much of the data would be differentiated by console generation. It should also be noted that handheld console data was initially gathered but never utilized in the final project due to time constraints. Following this more data was filtered into the excel spread sheet and more categories were added. The added categories were top selling games (relative to each console), game release years, game sales and game genres. From here the data could be sorted and various patterns and trends could be established through the filtering process, such as a manufacturers growing or diminishing popularity, increases or decreases in sales over time and various other statistics.

Commercials and advertisement videos of each time period were also extracted. This was decided as they offered a unique insight to the different generations of home consoles, again showing not only consoles development and evolution but the developments of marketing and promotion in video games. Each video extracted was categorized according to the home console it was advertising. As stated previously where some figures and statistics could not be found, instances of official advertisements could not be found for some of the more obscure or lesser known consoles. In these cases, videos of the consoles gameplay were substituted in their place while instances where no footage could be found at all were left blank. These videos were then sorted according to their generations and made accessible through the UI within the project.

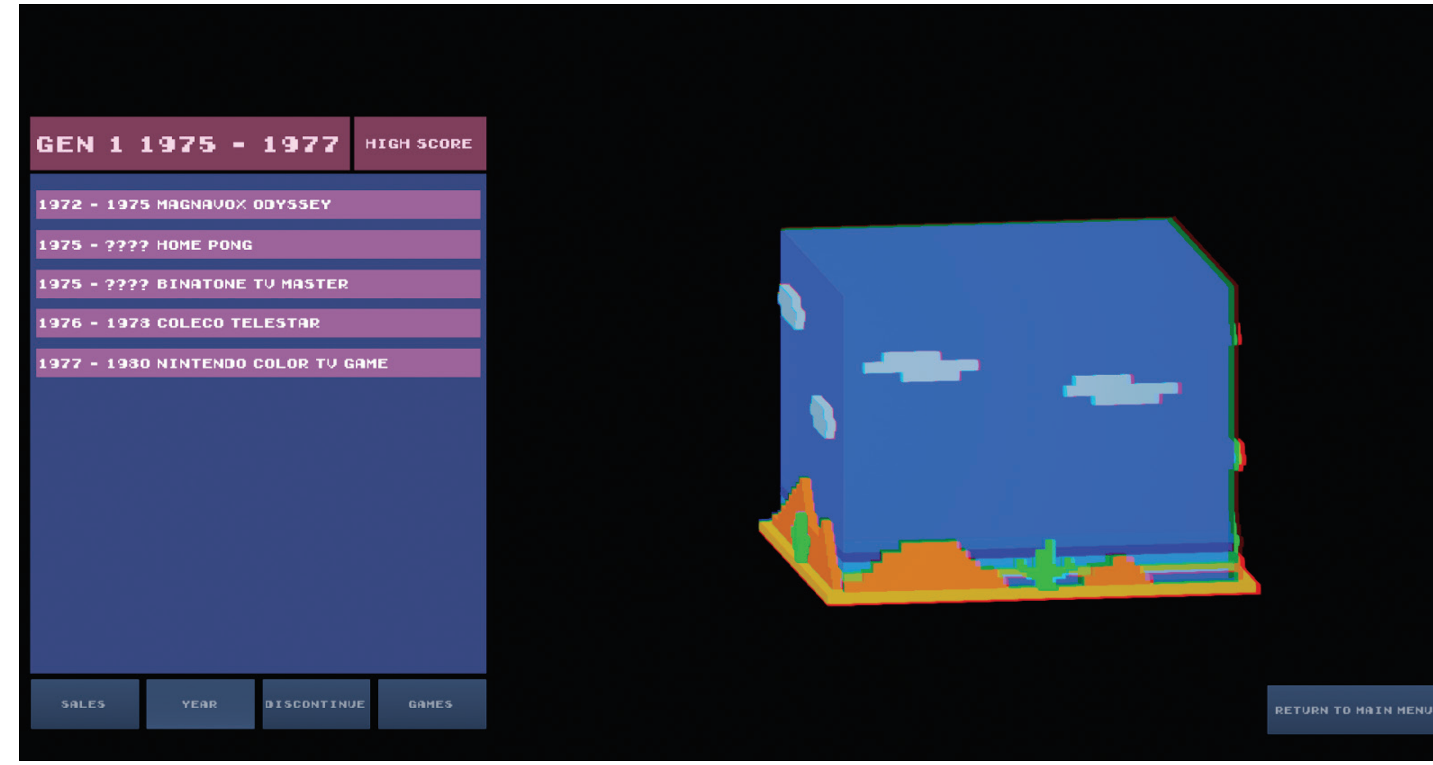


Figure 6.3. An early iteration of the user interface within Unreal Engine 4.

Operational Prototyping and UI Development

With statistical data gathered, development of the visualization would begin according to prior research on the visual and graphical aesthetics of the time as well as the technological developments of home consoles. The first phase of developing the visualization was building a system in Unreal that allowed the user to sort, select and view the gathered data in a manner similar to how excel sorted the information. This first step followed the operational prototyping framework outlined by Davis, establishing an "evolutionary prototype" (M. Davis, 1992, p.74) to act as a baseline in development. As stated previously, the design process followed a slightly altered Davis framework with certain stages of the method not being used, namely those involving prototype distribution and user system testing. These were discarded as the development of an interactive visualization does not necessarily require these. The base system was constructed using Unreal Engines blueprint (Unreals built in visual coding system) and allowed users to select a generation of home consoles and filter through the data of their selection. From here, with the exclusion of prototype distribution and user testing, test features were implemented "on top of the quality baseline software" (M. Davis, 1992, p.74) in a "quick-and-dirty throwaway prototype" (1992, p.74) manner. The following features were roughly implemented with animations developed using Unreals Widget system and a basic system built to access and sort data. Using buttons and selecting a category allowed for the data to easily transition in and out of the scene. This initial system was built to be flexible so that data could be easily entered and changed if required, and intended to be used for each generation of home console. The extracted commercials and videos were integrated, with a section of the UI offering an interface where they could be viewed. This system functioned but there were issues with audio playback. Video files functioned properly with the user able to view them, however the audio often would not play

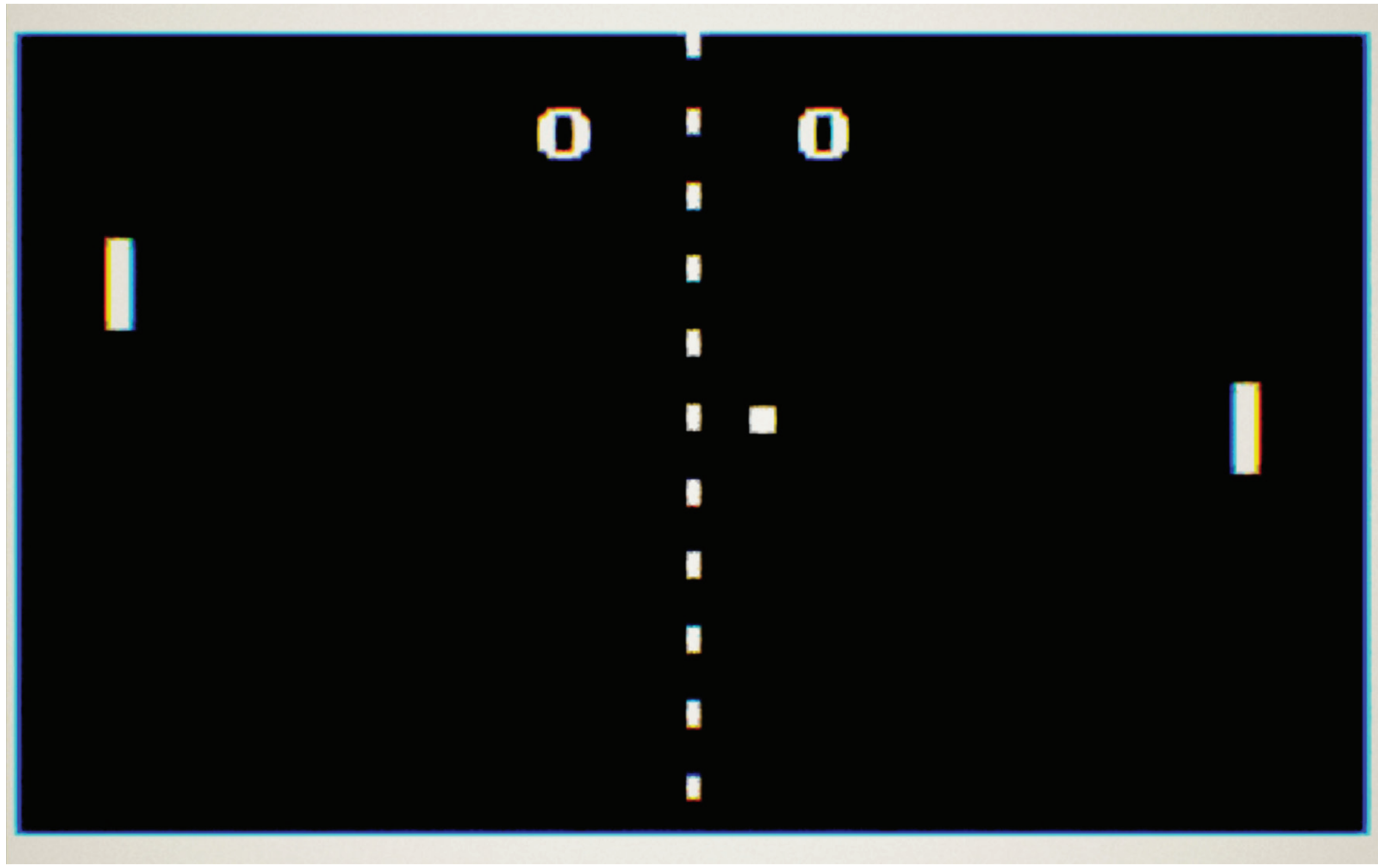


Figure 6.4. A screenshot of the generation one PONG (Atari, 1975) inspired level.

and when it would there were difficulties synchronizing picture and sound. Regardless, this feature remained as it was already integrated and near completely functional. The layout and display of the system would follow Tuftes outlines for "the friendly data graphic" (Tufte, 2001, p.183). The "basic structures" (2001, p.178) of "the sentence, the table, and the graphic" (2001, p.178) were used to make the data "particularly accessible and open to the eye" (2001, p.183). A simple table was developed and used to display data in a sequential order that would alter according to the user. At this point in development, the base system remained a combination of an evolutionary prototype and throwaway prototypes as not all features were complete or fully implemented.

From this the development of the game interaction would begin. This game, as previously stated, would offer a visual history while also linking to the previously constructed data sorting system. Utilising the edited work of Wolf, informing me of the visual and graphical evolution of video games, along with precedent video games of respective generations, multiple levels were built. Each of these levels was linked to a generation of home console history and would represent the time periods of their corresponding generation in both their aesthetic and their gameplay.

Through operational prototyping the first stage was to build the basic navigation systems, or player controllers. These would be the basic inputs for the user and act as the base evolutionary prototypes that each level would be built upon. With each level corresponding to a generation of home console, five different methods of control and navigation were built using Unreal Engines blueprint. Two of the five would navigate in a 2D manner, one a side scrolling controller and the second a top down controller. The second two of the five would navigate 3D space, one from a first person perspective and the second from a third person perspective. The fifth and final system was a basic Pong controller built specifically for the first generation of home consoles. The reason for these different modes of navigation was so that each would be representative of the gameplay style and control of the assigned time period. An example being that in the third generation, games of the mid-eighties to early nineties, the user would navigate the visualization in a 2D platformer format, due to the popularity of 2D sidescrollers at the time.

With the exception of the Pong system, each control system was partially built upon template work that the Unreal Engine offered, thereby using some of Unreal's defaults as the baseline prototypes. The 3D third person and 2D top down controllers were each heavily revised versions of Unreal templates. The Pong controller, however, was built as a stand-alone system. The reason for the specific Pong controller was that during data collection it was noted that most games of the first generation of home consoles were predominantly Pong. In many cases game were either Pong clones or variations of Pong, therefore a specific prototype had to be built to represent this.

The next stage was deciding the basic genre or style of each level and establishing what features would be prototyped and subsequently implemented. Generation one would be built in the style of Pong, while generations two and three would be built as 2D platformers. Generation four would be constructed as a top down action game while generations five, six and seven would be third person platformers. Initially generations seven and eight were to be built in the first person adventure or shooter genre, but due to time and resource constraints these concepts were removed, with generation eight being omitted completely. The genre decisions were made due to the popularity of these styles in their respective time periods. From the base systems I implemented key features based on the selected genre in the "throwaway prototype" (Davis, 1992, p.74) manner before finalizing changes to the base system. This iterative process would then utilize either linear or semi linear level design depending on which generation I was building for and which genre of game that generation had been assigned. Earlier generations of video games were often very linear in terms of design and gameplay; due to the arcade style nature of some of the games and the technical limitations of the time, therefore the earlier levels of the visualization were to be constructed with a linear approach in mind.

Level Construction

The first generation prototype required the least amount of work as the system for Pong was quite simple. The baseline system established was more or less complete in its first evolutionary prototype form. With the basic design informed by Ataris PONG (Atari, 1975), where the "white ball and paddles appear to be in front of the black background" (Wolf, 2009, p.152), the main features were already established with the user able to control the paddle and bounce the ball upon collision. The only changes required were the adjusting of the CPU paddle component, as its initial form was too difficult, and the basic ball physics initially being too unruly and fast.

The development of the second and third generation levels required various feature implementations and went through numerous iterations. Each level was built upon the 2D side scrolling controller system but would result in very different outcomes.

The second generation level would be built in the style of an Atari 2600 platforming game, games where the "primary objective requires movement through a series of levels" (Wolf, 2008, p.270), and made to resemble the classic game Pitfall! (Activision, 1980). This level would adhere to the limited sprite aesthetics and "overlapping plane graphics" (Wolf, 2009, p.152) of the Atari 2600. The first features to be implemented were the basic game rules. These were programmed in blueprint, and included player collision, player camera, enemy collision, ladder systems and game conditions. These were gradually developed with code being applied and removed before final changes were made to the base system. Some of these features proved problematic to implement due to my limited programming knowledge, with the ladder system and enemy collisions proving most difficult. The camera system is a notable feature in that it does not scroll. Though the Atari 2600 featured games that utilised a scrolling camera, games were often restricted to single screen experiences due to the limitations of the 2600s graphics chip, known as 'Stella'. Pitfall! Was an early game that "expanded beyond the initial one screen limit" (Therrien, 2008, p.241) by "displaying multiple adjacent spaces" (p.241). With this in mind the camera switches between static scenes to mimic this effect. Once the base system was revised basic sprite assets and animation frames were drawn in Photoshop. These visual assets were developed to imitate the 2600s graphics, with its "large uniformly coloured 'blocks' " (p.240). This style of graphics was due to the 2600s "inability to manipulate detailed bitmaps" (p.240). When these assets were completed they were imported to the Unreal Engine. Backgrounds were drawn and imported to the engine using the tile set tool. This tool allows for images to be divided into grids by the engine, with the grid size being allocated by the user. From these grids the developer can select which division or piece of the grid they wish to draw from and add to the background. This feature allows the developer to build multiple backgrounds from a single imported image, therefore when drawing background assets repetition and continuity must be taken into account. Once all assets were imported and working correctly they had to be applied to the blueprints so that they functioned. At this stage more additions were made to the system with the implementation of character animations to the player controller being the first step in applying graphical assets. Background assets and enemy animations were then implemented in a similar manner. With these assets complete and properly functioning they were refined, finalized and properly implemented, establishing a new baseline. From here level layout became the major priority as the new base system was now in a stable functioning state. The level followed a semi-linear approach; notable considering the early generation of home console. Pitfall!, though a 2D platformer in which the player moves across the horizontal axis, features somewhat open ended gameplay with the player attempting to gather all the treasure scattered throughout the game. The player must revisit and explore areas and the goal is not achieved simply by moving from left to right. With this noted, the generation two level constructed follows a similar route, with the user able to decide their approach until being "led to a bottleneck" (Kremers, 2009, p.57) that will conclude the level.

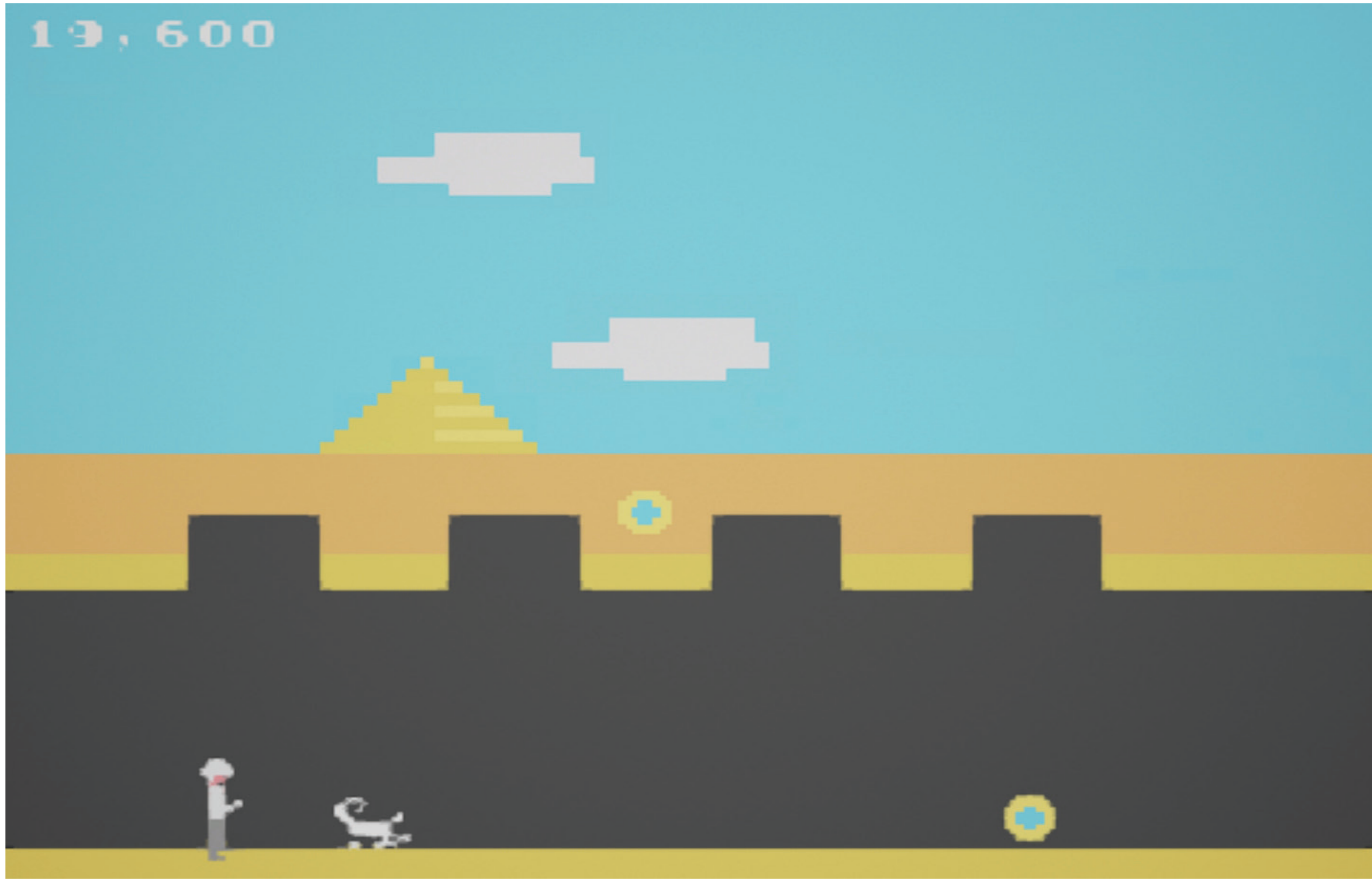


Figure 6.5. Pitfall (Activision, 1980) inspired generation two level that was developed.

Figure 6.6. Pitfall (Activision, 1980) for the Atari 2600, an early example of the 2-D platformer.

Figure 6.7. Screenshot of the generation two platforming level.

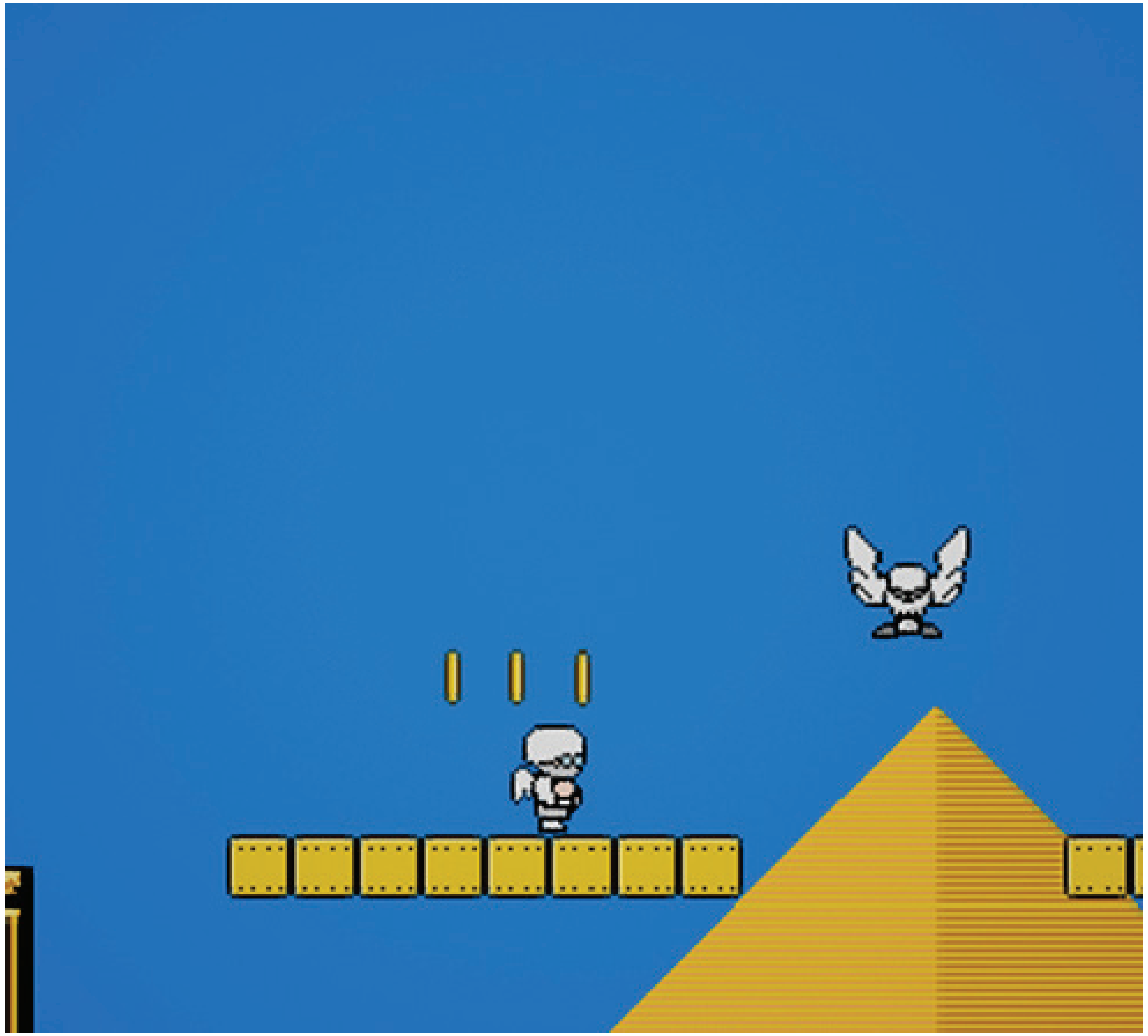


Figure 6.8. The 8-bit inspired third generation level taking its inspiration from the popular Mario series.

The third generation level would follow a similar path to generation two, but with more complex artwork and gameplay required. The significant popularity of platformer games such as Super Mario Bros (Nintendo, 1985) and Alex Kidd in Miracle World (Sega, 1986) would lead to the 2D controller system being used again. The level was built to imitate the 8-bit processors and "2.5 dimensional graphics" (Wolf, p.157, 2008) of consoles such as the NES and Sega Master System. While the Atari 2600 platformers offered basic control, the controller for this level required more complex features, such as the ability to move the character at multiple speeds, jump at differing heights and utilize power ups. Power ups were a common occurrence in games during this generation; therefore it was decided that they would be implemented in this level. Many of these features were roughly implemented on top of the initial controller system, with the most difficult feature to implement being variable

Figure 6.9. Super Mario Bros 3 (Nintendo, 1988) one the most popular titles for the NES.

jumping heights. This was achieved by prototyping an instance where world scale gravity is altered when the jump button is pressed and released, thereby giving the impression of variable jumping height. Basic enemy AI was constructed by moving enemies between given coordinates over an allocated period of time. This was developed using the timeline system, an Unreal Engine feature which allows events to occur over, or at a time allocated by the developer. Again, when all features were functional they were integrated with the base system. From here the required art assets were developed. With the improvements in sprite technology, "sprites were coloured and large enough for more detailed characters" allowing for "different sprite designs" that could "animate characters turning in different positions" adding "a sense of depth" (Wolf, 2009, p.157). Due to this, higher quality assets needed to be developed again in Photoshop. The NES was able to display up to

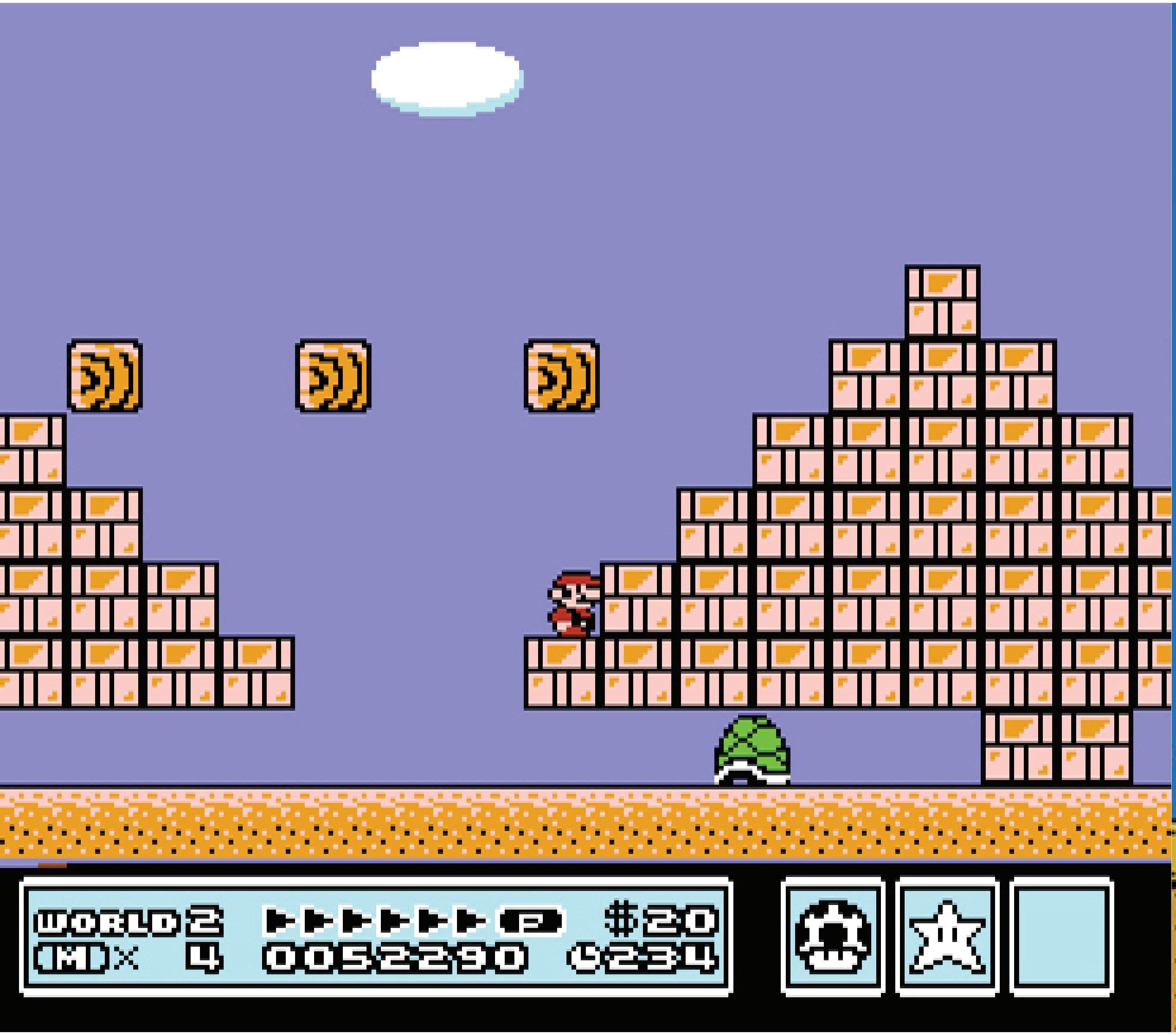


Figure 6.10. Screenshot from Super Mario Bros 3 (Nintendo, 1988).

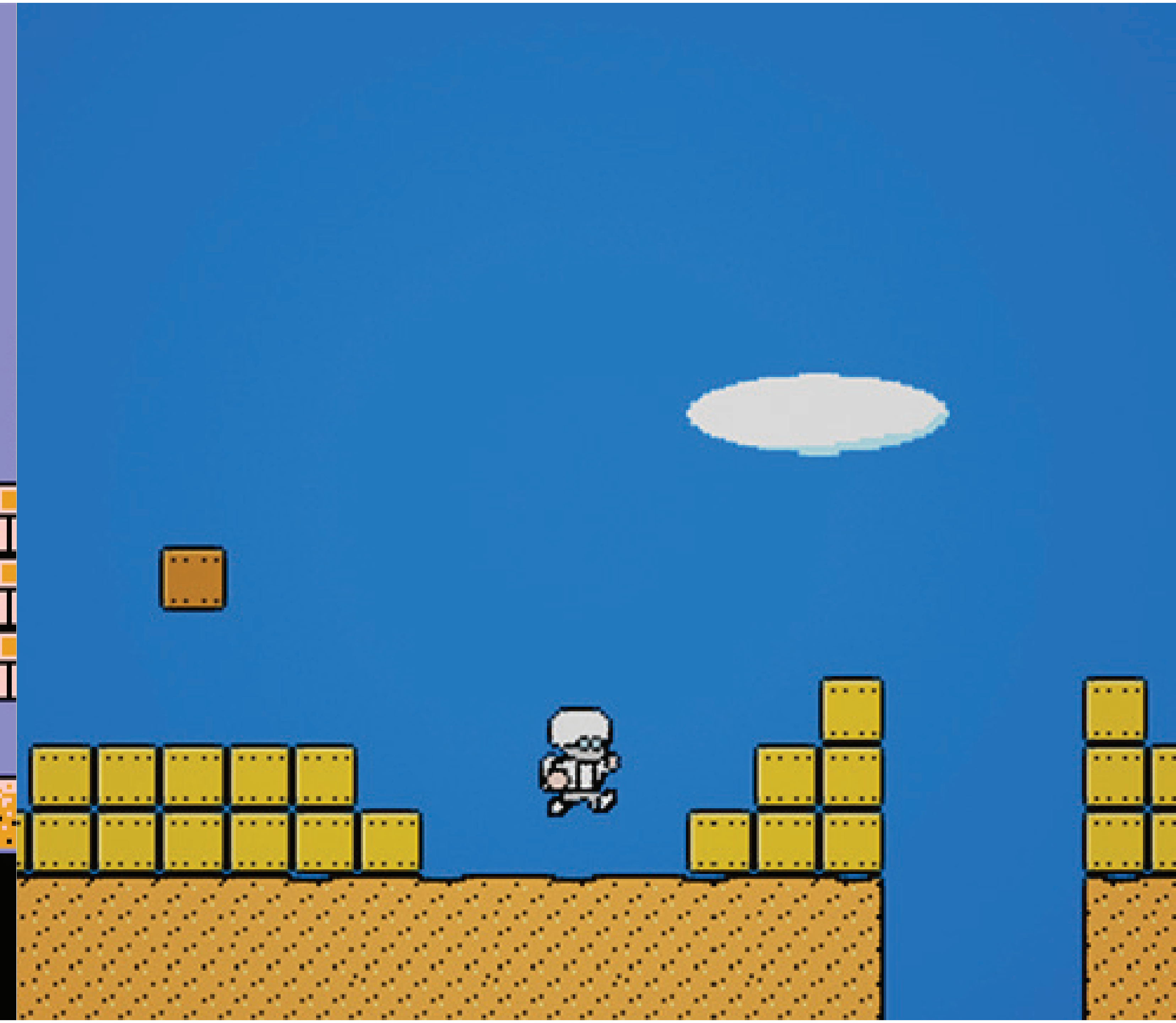


Figure 6.11. Screenshot from the third generation level.

Figure 6.12. The Legend of Zelda: A Link to the Past (Nintendo, 1991), the key precedent for the fourth generation level.

sixteen colours and up to sixty four sprites at a time at a screen resolution of 256 x 240 (Arsenault, 2008, p.114). These limitations were followed as closely as possible to maintain the 8-bit aesthetic; however screen resolution could be not be controlled to fully match this. Therefore a 4:3 (standard definition television) screen ratio was utilised to imitate this. The implementation of art assets began with the assigning of the character sprites to the new baseline controller system, with backgrounds being handled through the Unreal tile set tool. This level was built to follow a far more linear design to imitate the genre conventions of the era. 2D platformers were known for their linear gameplay therefore it was applied to this levels structure. The user in this case is only able to navigate from left to right, with events occurring only as intended by the developer.

The fourth generation level used the top down player controller system. The fourth generation of home consoles is often referred to as the 16-bit era and is remembered for the rivalry between Sega's Genesis console and Nintendos SNES console, with each competing in "the home console race" (Herman, 2008, p.120). 2D side scrollers again remained popular during this time period, with Sonic the Hedgehog (Sega, 1991) and Super Mario World (Nintendo, 1990) being two of the most popular games of the time. However the top down adventure and RPG (role playing games) genres also became prominent, particularly on the SNES with titles such as The Legend of Zelda: A link to the Past (Nintendo, 1991), Final Fantasy III (Squaresoft, 1994) and Chrono Trigger (Squaresoft, 1995). Following these titles, the fourth generation level was constructed as a top down adventure game, a game "made up of multiple, connected rooms" (Wolf, 2008, p.262) with the user navigating from the bird's eye perspective. The generations' technological development

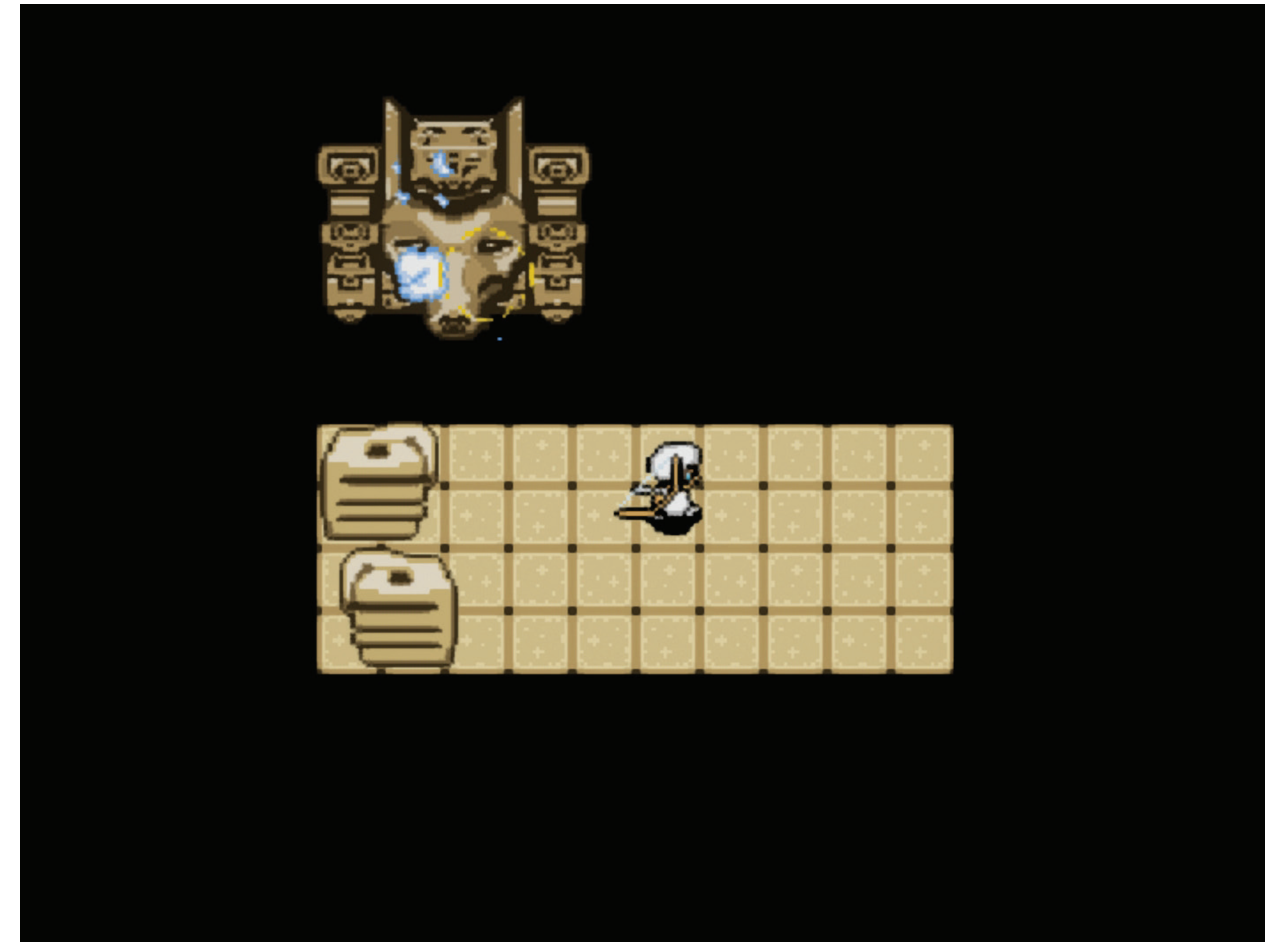


Figure 6.13. Boss battle from the fourth generation level. This level was built to represent the 16-bit top down adventure and rpg games of the early nineties.

again allowed for more complex art, animation and gameplay. With "larger memory capacities" (Pidkameny, 2008, p.253) 16-bit consoles were more capable than their predecessors, with the SNES cartridge able to hold up to six megabytes of data. Because of this notable improvements in the graphics and gameplay had to be made. Again features were prototyped on top of the base system before being integrated and finalized with the implementation of artwork, establishing a new baseline. The level itself would follow a semi-linear design as adventure games should not offer "linear progression" but allow for the "player to wander and explore the games world freely" (Wolf, 2008, p.262). In this level the user must obtain keys to unlock doors allowing further progression; however the keys do not have to be accessed in a specific order. The user can decide how to approach this before being forced in to a designed bottleneck that will lead to the levels conclusion. The development of this level would act as the last to operate in 2D as the following generation would make use of the developing 3D technologies, namely the CD-ROM.

Figure 6.14. Screenshot from The Legend of Zelda: A Link to the Past (Nintendo, 1991). Note the use of tile based sprites as backgrounds.

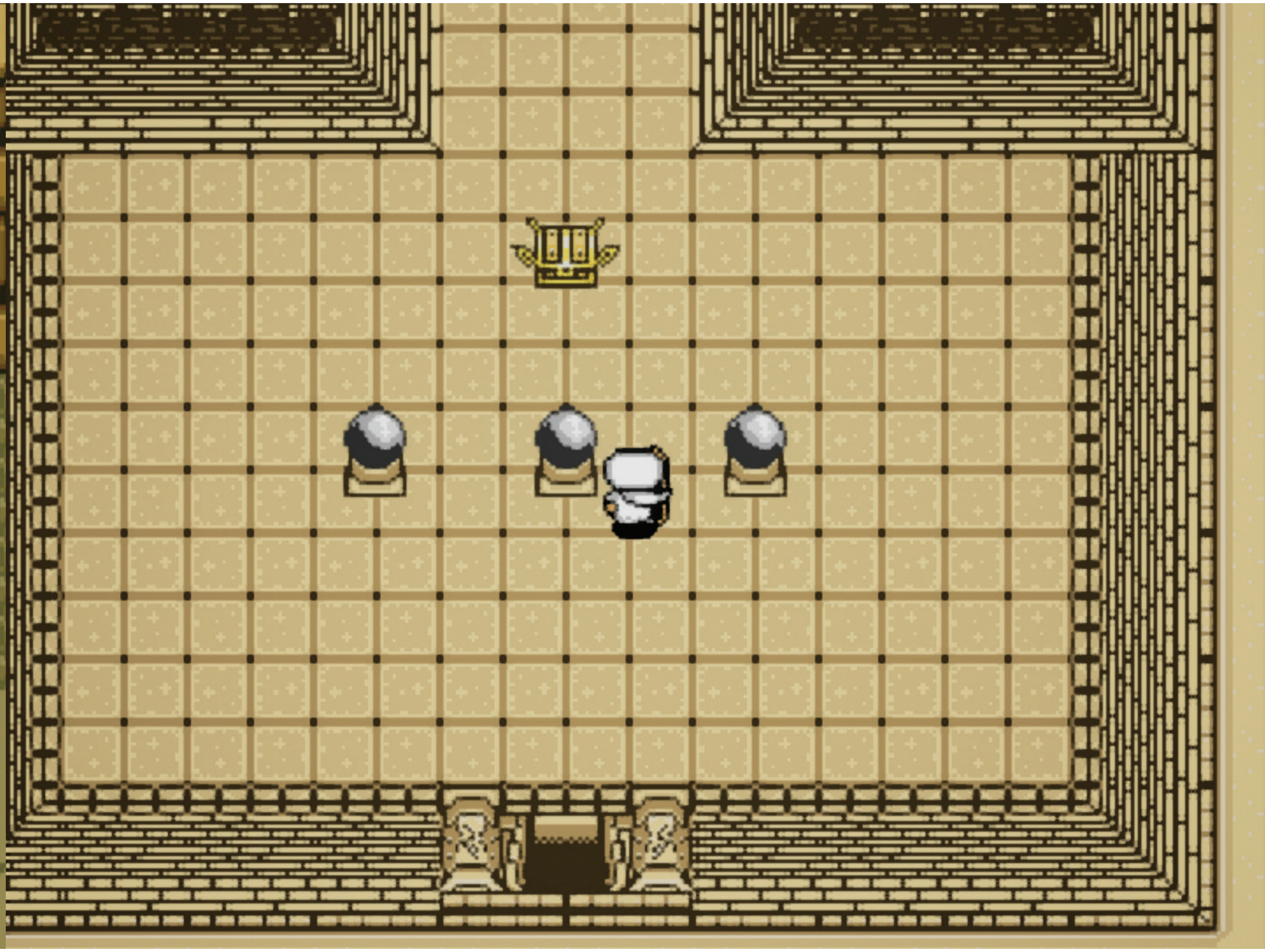


Figure 6.15. Screenshot from the fourth generation level as the character approaches a treasure chest.

Figure 6.16. Crash Bandicoot (Naughty Dog 1997) one of the earliest mascots for the Sony Playstation and key influence on the generation five level.

Figure 6.17. The fifth generation level, intended to represent the early developments in 3D home console gaming.

The fifth generation saw "3D arcade games and home games" (Herman, 2008, p.164) become "more common" (Herman, 2008, p.164) as many consoles utilised CD-ROM, a disc based media that could "hold up to 700 megabytes" (Therrien, 2008, p.121). This format would also introduce "the infamous loading screen to the console gamer" (Therrien, 2008, p.121) as data from CD could not be instantly read the way the previous generations ROM cartridges could. These changes would alter some of the development and iteration process, with the use of new software required, however the basic design structure would remain the same. The most successful console of the era was Sonys PlayStation, featuring titles like Crash Bandicoot (Naughty Dog, 1996) and Grand Turismo (Polyphony Digital & Sony Computer Entertainment, 1997), therefore this level would be built to reflect and imitate Sonys landmark console. Using the third person 3D controller the level was constructed in the style of the 3D platforming genre, popularised by titles such as Super Mario 64 (Nintendo, 1996) and Banjo Kazooie (Rareware, 1998) for the Nintendo 64, with Crash Bandicoot (Naughty Dog, 1996) for the PlayStation. Beginning with the third person controller system, little work was required as the initial baseline system met most requirements of the level design. With the user already able to run and jump, the only feature prototyped and subsequently implemented to the base system was a static camera controller. This camera was designed to imitate the more rigid systems of the time. With the base system established work began in Autodesk Maya to model, rig and animate all 3D assets required. Though consoles of this era were now capable of real time 3D there remained limitations. 3D real time graphics of the era could not "afford the same kind of refined detail, high resolution, and subtle lighting effects that pre rendered images" (Wolf, 2009, p.164) could. Due to this games had to be designed around these limitations, with the levels texture resolutions and polygon counts kept low to ensure a faithful imitation of the generations' imagery. Once the player model was completed it was rigged and animated before being imported to the engine for use. Unlike 2D assets, where sprites were imported before making animations that a blueprint could recognise, 3D animations had to be compiled into a blend space and from there structured into an animation blueprint. An animation blueprint is a specific blueprint for 3D animation that can communicate to the controller system. The basic model and rig were imported to Unreal, at which point the animations were imported to the engine. Animations were then attached to the previously imported model and from there an animation blueprint could be constructed. Once the animation blueprint was built it was linked to the player controller and properly implemented. From here all modelled environmental assets were imported and the design and structure of the level could begin. With the PlayStation unable to render large and expansive areas in real time; a "solution to this problem was to design spaces in such a way as to avoid sightlines extending deep into the distance" (2009, p.165). A common example around this was by "dividing interiors into rooms and hallways" ensuring "only a small portion" of the game "is rendered at any given time" (2009, p.165). Using this information, the level was designed in a linear fashion within enclosed spaces while also obscuring faraway distances and views. This was a common technique of the time where "receding spaces are gradually darkened" (Wolf, 2009, p.165). Like the previous third generation platformer level, this level was designed in a linear manner to more closely imitate the game play of its source inspiration, Crash Bandicoot (Naughty Dog, 1996).

Figure 6.18. Screenshot from Crash Bandicoot (Naughty Dog 1997). Note the darker areas and inability to see further in the distance saving both processing and rendering power.

Figure 6.19. Screenshot from the fifth generation level where rendering techniques were imitated to more closely resemble the early 3D technology.

Figure 6.20. The generation six level intended to imitate the larger environments and improved visuals of games such as Jak and Daxter (Naughty Dog, 2001) and Ratchet and Clank (Insomniac, 2002).

The sixth generation level would mirror the development of its predecessor. With Sonys PS2 as the most popular console of this era, and of all time, this level would imitate its gameplay and aesthetics. This level also adopted the play mechanics of a 3D platformer, imitating popular titles of the time such as Jak and Daxter (Naughty Dog, 2001) and Ratchet and Clank (Insomniac Games, 2002). The technological developments of this generation would make a notable impact on both the graphical capabilities and gameplay possibilities of home gaming. With the introduction of DVD and its ability to "store six times as much data as a CD" (Herman, 2008, p.167) more work would be required to achieve both the aesthetics and gameplay functionality of the era. Due to the higher standards of both the visuals and gameplay this level had a notably longer development period than its predecessors. More time was spent fine tuning blueprints and ensuring functionality than any other level previously. The baseline system required multiple features to be implemented in a quality manner before level design could begin; with some of these including complex player controls and 3D enemy AI. Utilising the Unreal Engine Nav Mesh tool I was able to prototype an AI control system, whereby enemies track and follow the user once they are within a certain distance of the enemies' sight collision box. The revised control scheme was also developed allowing the user to walk, run, dash, attack, jump and double jump. These features once complete would be implemented alongside the required 3D assets. These assets again included models, rigs and textures and were modelled in Maya. However the visual standards had again been raised, with higher polygon counts, smoother animations and higher quality texturing all required to complete the visualization. Once all features were properly implemented and a new baseline was established, level design could begin. The level design returned to a semi linear approach as the generations' technology was no longer confined to the enclosed spaces of the previous generation. With "the increasing power of home systems able to extend visibility quite far down the z-axis into the distance" (Wolf, 2009, p.165) larger more expansive environments were now possible. This level allowed the user to direct themselves and take their own path through the visualization before being forced to the end goal.

Figure 6.21. Jak and Daxter (Naughty Dog, 2001) was a major title for the PlayStation 2 and demonstrated the improved hardware and capabilities with larger environments and more detailed visuals.

Figure 6.22. Screenshot from the generation six level.

Figure 6.23. Screenshot from Jak and Daxter (Naughty Dog, 2001) showing the larger environments and increased draw distance.

The seventh generation level would again follow the previous development process and use the third person controller; though it was originally intended to utilize the first person system. The initial concept was to allow the user to navigate a first person perspective to imitate the popularity of first person shooters during the seventh generation. However as noted during previous levels development, the time required achieving both a suitable graphical and interactive standard was now much greater. More and more work was required to meet these high standards due to the technical improvements of every generation and because of this the first person concept was dropped. I would have been unable to complete, with resources and time considered, a level in that genre to a suitable standard. With this considered the choice was made that the visualization would imitate the aesthetics of independent games or indie games. This decision was made due to indie games smaller scale, allowing for a more feasible development period, and their rise in popularity throughout this era due to the increase in online distribution. Independent games are known for their exploration of "new aesthetic territory" (Camper, 2008, p.199), with designers "challenging how we make and play games, why we do so, and what subject matter they can portray" (2008, p.197). It was decided the level would imitate titles like Journey (ThatGameCompany, 2012), a 3D platformer, and Limbo (Playdead, 2010) a 2D platformer, both indie games known for their innovation and creativity. With this considered I returned to the 3D platformer style and recycled the base system from the previous level, which in its current state was already suitable. This would leave art and asset creation as the majority of work left to be done and as the key focus before implementation.

Beginning in Maya the player character was modelled and rigged, however unlike previous generations the model was exported into the sculpting software Zbrush. The developing technology of this generation would allow for the techniques of "bump mapping and normal mapping" (Therrien, 2008, p. 248), a technique which fakes the details of indents and extrusions on a mesh, "thus creating bumps with no additional geometry (2008, p.248). Inside Zbrush clothing details were sculpted onto the original mesh, adding more detail and subsequently more polygons. This detail was then exported to a normal map which was applied to the original model in Maya. This allowed the added detail to be shown without additional polygons being added, thereby saving processing power. A notable feature implemented was physics based clothing, to imitate the cloth physics of Journey (ThatGameCompany, 2012). This was difficult as Unreal Engine does not accept Mayas nCloth physics. This forced me to utilize Nvidias PhysX cloth. There were initially difficulties with compatibility and the cloth was prone to crashing both Maya and Unreal Engine. This was eventually overcome when settings in Maya were adjusted. With these complete, environment assets were modelled and textured. The environmental assets also made use of normal maps allowing for greater depth and detail without additional polygons. Lighting also played a larger role in this level with the "addition of ever more complex effects" (Therrien, 2008, p.249) on home consoles. The lighting built made use of Unreal Light sources and an illuminated texture applied to a mesh, giving a fog like volumetric lighting illusion. This technique was reused throughout the level as even in this generation "3D game worlds" were "still prone to repetition" with "tiled textures and reused animations" (2008, p.249) still common. The lighting technique was also developed in this manner to conserve processing power while maintaining the higher standard visuals. These features were then implemented and allowed for the level design process to begin. Though open world game design had become common in this generation linear level design was chosen as the level structure. This was due to both technical and time limitations as constructing a narrow enclosed path allowed for high quality rendering and imagery without heavily straining the hardware.

Figure 6.24. Screenshot from the Journey (ThatGameCompany, 2012) influenced generation seven level.

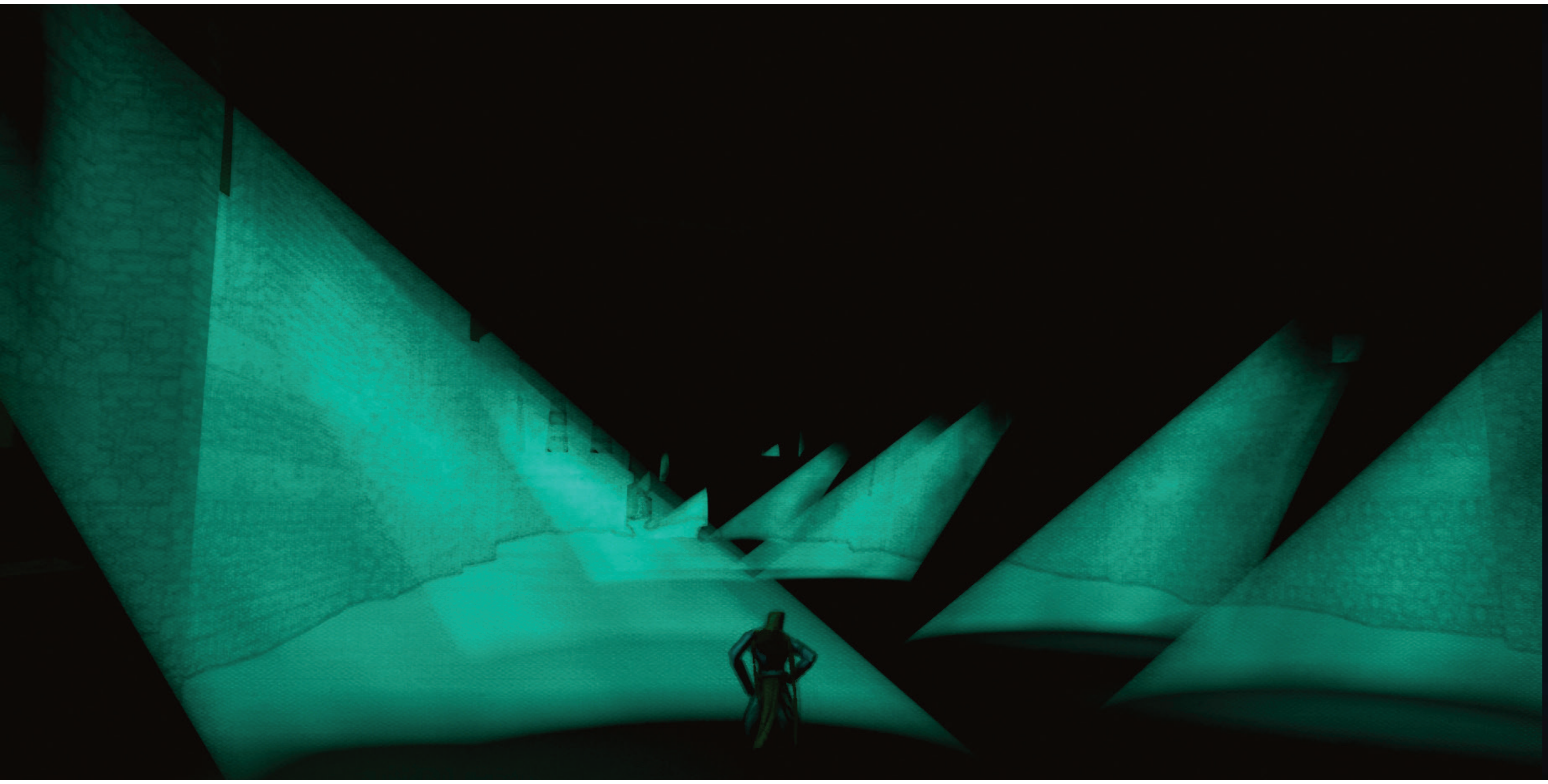


Figure 6.25. Screenshot of the generation seven level. The focus of this level was developing unique lighting and shading visuals to imitate some of the indie game aesthetics.

Figure 6.26. Screenshot from Journey (ThatGameCompany, 2012) demonstrating the the unique visual style and gameplay found in indie games.

UI Revision and Data Integration

Following the completion of all levels, the original gathered data was then integrated with the initial UI information sorting system. Extracting the data from the excel spread sheet and filling in the UI system was a simple process, however the UI visualization no longer suited the project in terms of both functionality and aesthetics. Following attempts to adjust the UI to suit the project it was decided to remove this base system and build a new prototype entirely. The original system was removed and a new system was established. The new system used some aspects of the previous UI but removed the widget system, instead being built entirely within blueprint. However this system was very much constructed in a throwaway prototype fashion with some poorly considered features being implemented due to its rushed development. All information was placed upon images which were formatted according to Tuftes “friendly data graphic” again utilising “the sentence, the table, and the graphic” (Tufte, 2001, p.178). However with data being imprinted to images, information could not be adjusted after being imported to the engine. To alter the data one had to return to Photoshop and make changes there before reimporting to Unreal, as opposed to making changes directly in the engine. The new system essentially hard coded data to the user interface. This same hard coding practise also applied to the commercial and advertisement videos due to the new system implemented. Again this system was not ideal; however it achieved its intended purpose of informing the user and was subsequently implemented.

Before data was added to the completed levels audio had to be taken into account. With the time and resources considered it was deemed unfeasible that I produce all required sound and music while developing the project itself. With the projects intention to preserve video game history the decision was made to use extracts of music and sound cues from some of the precedent games of each generation. Seeing as this project is not intended to be released nor developed for any commercial benefit, the audio would be used under the license of fair use and subsequently accredited to the appropriate composers and sound engineers. The reasoning for this was also to preserve and highlight the evolution of video game audio. Much in the same way the commercials extracted demonstrate the evolution of video game marketing, this audio highlights the development of video game sound as the technology has developed over time.

Before data was added to the completed levels audio had to be taken into account. With the time and resources considered it was deemed unfeasible that I produce all required sound and music while developing the project itself. With the projects intention to preserve video game history the decision was made to use extracts of music and sound cues from some of the precedent games of each generation. Seeing as this project is not intended to be released nor developed for any commercial benefit, the audio would be used under the license of fair use and subsequently accredited to the appropriate composers and sound engineers. The reasoning for this was also to preserve and highlight the evolution of video game audio. Much in the same way the commercials extracted demonstrate the evolution of video game marketing, this audio highlights the development of video game sound as the technology has developed over time.

The final stage would see extracted data, again following the text mining framework, from various readings used to construct a basic narrative around and within the levels. This data would appear as text windows throughout levels, showing information and history relevant to each generation. The process of knowledge distillation was again applied with information being categorized by generation before assignment to the generations corresponding level. This feature was developed within each levels system and first prototyped using the widget feature. Once iterated to a suitable standard the feature was properly implemented within the base system and the data filtered through.

This would conclude the practical component of the project with all systems developed to a high quality working standard and the interactive game properly linked and working with the UI data visualization system.

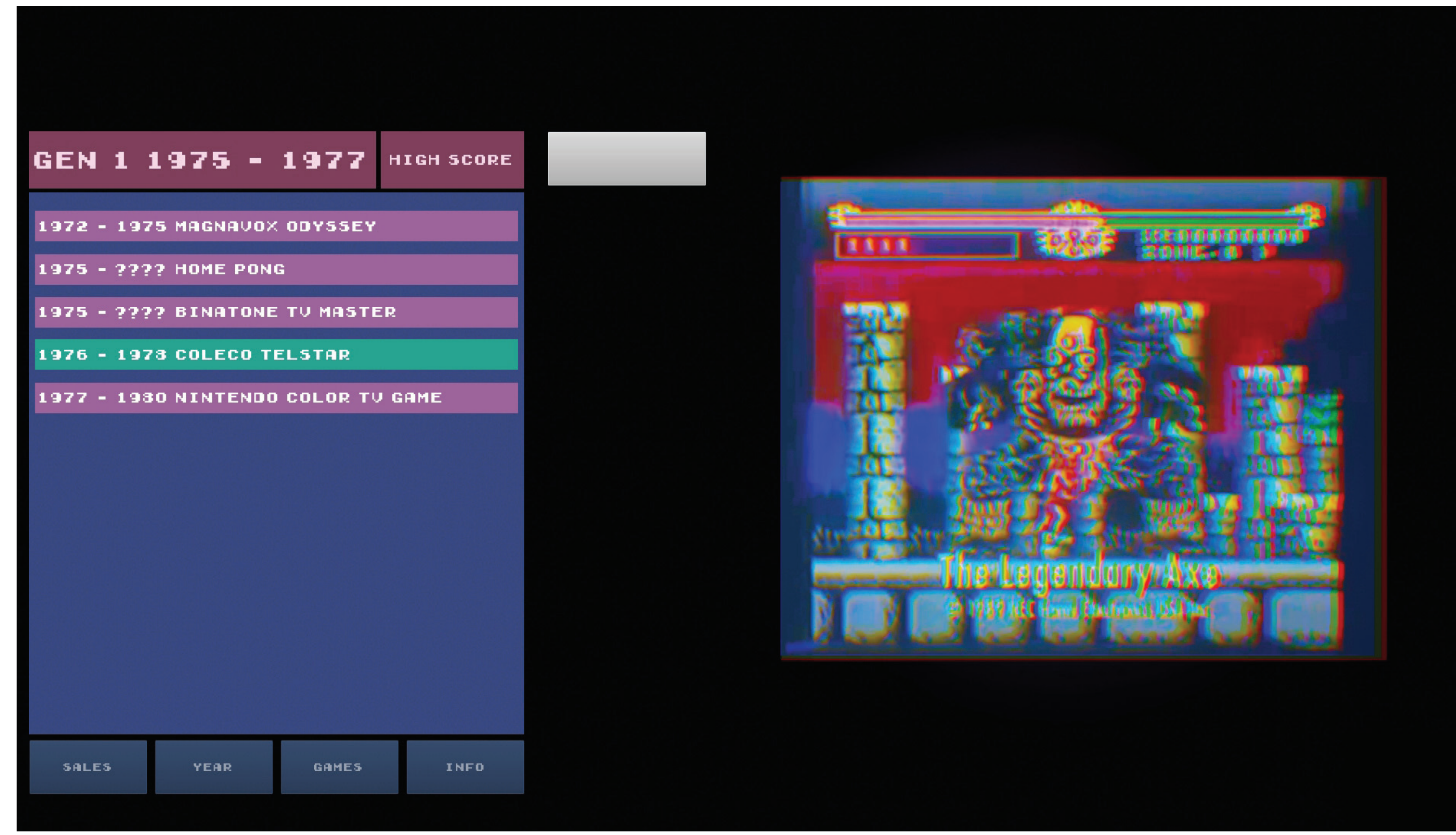


Figure 6.27. The original user interface and data visualization with the video capabilities added.

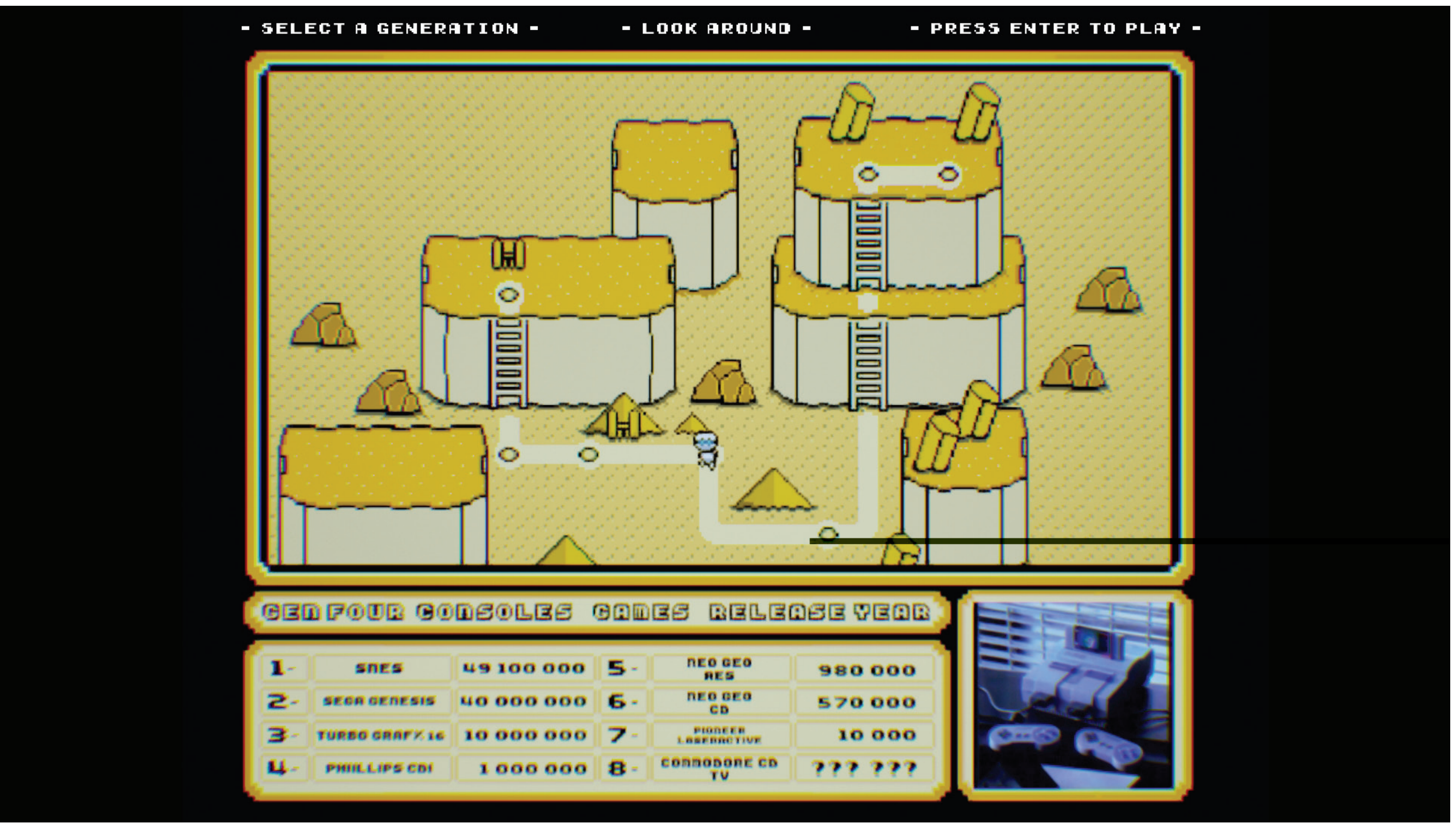
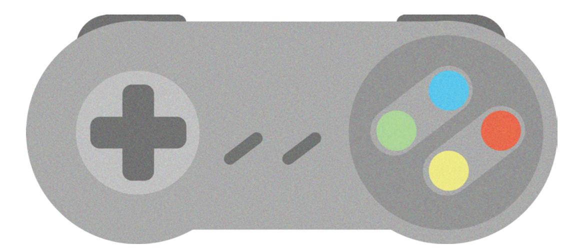


Figure 6.28. The revised user interface and data visualization with the altered video system.

Chapter 7

DEVELOPED DESIGN



Output Reflection

The completed design in hindsight achieves the aims and objectives set by the project with data visualized in an interactive library and the development of a game that interacts with said library. However there are areas that could be improved upon were the thesis to be expanded to a multiyear project.

The first area is the programming and structure of the overall project. If this project were to be advanced, greater efforts would be made to allow for easier continued development. The projects current structure would be revised to allow for development from other parties and collaboration. Though Blueprint suited my needs and experience, it is not a common or well-known programming language in the way Python, C# or java are. This limits the options of continual development to those only familiar with the blueprint system. Developing with the concept of open source or continued development in mind potentially would have led to more thought and emphasis being placed upon the projects structure.

A key area of development would be the visual representations of the generation levels six and seven. The higher quality visuals of both were not recreated as faithfully as the previous generations. The prior generations graphics and gameplay could be imitated and constructed in a timely manner, however from generation 6 onwards, the development times more than doubled. All levels, from the first generation to the fifth, required approximately two weeks' work, and in that time frame the base gameplay systems could be developed alongside all art assets to the standard required. However, generations six and seven required four to five weeks of work each just to reach a functional standard. From both a visual and interactive perspective I feel that these levels remain ripe for further development.

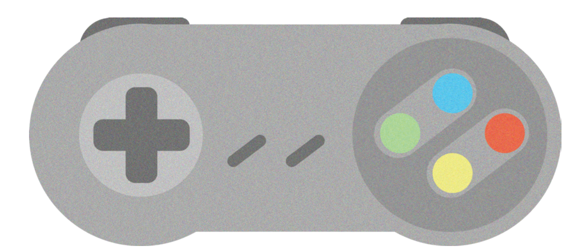
Another aspect to take into consideration is the use of operational prototyping. The choice of operational prototyping as the key design method was mostly successful; however there were minor issues with the process. The establishing of baseline systems for all player controllers, UI setups and level systems helped established key requirements early and developed a foundation to build upon. However the development of features through throwaway prototyping could lead to features of questionable quality being implemented. The method states features are developed in a "quick-and-dirty throwaway" (Davis, 1992, p.74) manner before being finalized in a quality fashion and implemented. Again however these issues were relative to time and only became prominent in the later stages of the project when longer periods of time were required for development.

The intention of making a compact and accessible format for the gathered data I feel has been achieved, and the interactive gameplay format as opposed to a more standardised user interface allows for a more engaging navigation of virtual space. Even with the scope limited to home console history there are large amounts of data and information that remain absent from the final project. Or in some cases data is present, though in a diluted format and not with the detail and insight that would have been preferred. The design output feels completed but seems as though more can be done. In its current state it appears as a streamlined and condensed collection of home console history. With the projects intention to explore the possibilities of interactive data visualization, there remains room for more development in all facets of the project.

In retrospect I feel that the project leaves great opportunity for expansion. The project in its current form shows some of the potential of video games in the field of visualization, but it has merely scratched the surface of said potential. Though completed, I feel that on a larger scale and with continued development something comprehensive and exceptional could be produced.

Chapter 8

CONCLUSION



Conclusions and Project Reflection

Video games have a long, complex and detailed history. Even when limited to the history of the home console, the intention to distil and compile all or as much of that history as possible into a single interactive visualization remains a difficult task. Though the possibilities of data visualization and interactive design were explored, and an interactive visualization was developed and completed, the question one comes to is whether the proposed aims were achieved. As an interactive visualization, the design output acts as an introduction to the development of the home video game console, while also offering insight and an overview of their history. In this sense, it does achieve what was proposed. Data visualization was explored through interactive design and produced an example of what can be achieved through their unique partnership. However there remains a wealth of knowledge that is not easily accessible to the public and it is this knowledge that could also be stored within a similar interactive visualization. As the research progressed I noted that the project would have been ideal for open source development, thereby offering the possibility of continued work. However this realisation was made too late in development to be seriously considered and would required far more resources to achieve. The design output of this project can be seen as a starting point, an example of what can be done to preserve the history of video games, in a manner that both reflects and encapsulates the medium while allowing for a greater understanding of it. It is part of a unique approach to data visualization that has not been pursued previously, and shows how data can be visualized in a way that is both unique and innovative.

In conclusion the design output and research undertaken here do not provide definitive answers regarding interactive design and its possibilities within the field of data visualization. Nor does the project provide a complete catalogue of home console history. Instead it provides a snapshot of video game history and acts as an example of what can be achieved through the use of both data visualization and interactive design. For video game history it perhaps offers a new and exciting means to preserve and catalogue the vast wealth of video game data produced over the past 50 years, offering a format that allows for a more engaging and accessible manner of learning. In its current state this project offers a starting point, a potential way to store and preserve video game history for future generations. Through visualization and interactive design, an exciting new way of learning can be innovated and this project could perhaps act as either a basis or template for future research. There is a great amount of potential in the combinations of these two fields, and it is a potential that is yet to be fully explored.

Bibliography

Arsenault, D. (2008). SYSTEM PROFILE: THE NINTENDO ENTERTAINMENT SYSTEM (NES). In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 109-114). Greenwood Press.

Camper, B. (2009). INDEPENDENT AND EXPERIMENTAL VIDEO GAMES. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 197-202). Greenwood Press.

Camper, B. (2009). RETRO REFLEXIVITY La-Mulana, an 8-Bit Period Piece. In M. J. P. Wolf (Ed.), *THE VIDEO GAME THEORY READER 2* (pp. 169-195). Routledge.

Chandler, M. C., & Chandler, R. (2011). *Fundamentals of Game Development*. Sudbury, United States of America: Jones & Bartlett Learning.

Chung, P. (2015). SINGAPORE. In M. J. P. Wolf (Ed.), *Video Games Around the World* (pp. 469–494). MIT Press. Retrieved from http://www.jstor.org/stable/j.ctt17kk7tc.35
Davis, A. (2008). OPERATIONAL PROTOTYPING: A NEW APPROACH. IEEE, 9(5), 70 -78. http://dx.doi.org/10.1109/52.156899

Friese, K., Herrlich, M. & Wolter, F. (2008) Using Game Engines for Visualization in Scientific Applications. *IFIP International Federation for Information Processing*, 279, 11-22. http://dx.doi.org/ 10.1007/978-0-387-09701-5_2

Herman, L. (2008). A NEW GENERATION OF HOME VIDEO GAME SYSTEMS. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 115-120). Greenwood Press.

Herman, L. (2008). THE LATER GENERATION OF HOME VIDEO GAME SYSTEMS. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 161-171). Greenwood Press.

Herwiig, A., & Paar, P. (2002). Game Engines: Tools for Visualization and Planning? In E, Buhmann., U, Nothelfer., & M. Pietsch (Ed.), *Trends in GIS and Virtualization in Environmental Planning and Design* (pp. 162-171). Wichman.

Kremers, R. (2009). *Level Design CONCEPT, THEORY, & PRACTICE*. Natick, United States of America: A K Peters, Ltd.

Lizuka, K & Thawonmas, R. (2008). Visualization of Online Game Players Based on Their Action Behaviours. *International Journal of Computer Games Technology*, 2008(5), 1 -9. http://dx.doi.org/10.1155/2008/906931

Magerko, B., & Medler, B. (2011) Analytics of Play: Using Information Visualization and Gameplay Practises for Visualizing Video Game Data. *PARSONS JOURNAL FOR INFORMATION MAPPING*, 3(1), 1-12.

Manovich, L. (2001). *The Language of New Media*. Cambridge, United States of America: MIT Press.

Pidkameny, E. (2008). SOUND IN VIDEO GAMES. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 251-257). Greenwood Press.

Tan, A. (1999, April). *Text mining: The state of the art and the challenges*. Paper presented at the Proceedings of the PAKDD Workshop on Knowledge discovery from Advanced Databases, China. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.6973&rep=rep1&type=pdf

Therrien, C. (2008). CD-ROM GAMES. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 121-125). Greenwood Press.

Therrien, C. (2008). GRAPHICS IN VIDEO GAMES. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 239-250). Greenwood Press.

Tufte, E. (2001). *The Visual Display of Quantitative Information* (2nd ed.). Cheshire, United States of America: Graphics Press.

Wolf, M. J. P (2006). Assessing Interactivity in Video Game Design. *Mechademia 1: Emerging Worlds of Anime and Manga*, 1, 78 -85. Retrieved from http://www.jstor.orgstable/41510879

Wolf, M. J. P. (2008). Imaging Technologies. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 19-12). Greenwood Press.

Wolf, M. J. P. (2008). VIDEO GAME GENRES. In M. J. P. Wolf (Ed.), *THE VIDEO GAME EXPLOSION A History from PONG to PlayStation® and Beyond* (pp. 259-274). Greenwood Press.

Wolf, M. J. P. (2008). Z-axis Development in the Video Game. In M.J.P. Wolf & B. Perron (Ed.), *THE VIDEO GAME THEORY READER 2* (pp. 151-168). Greenwood Press.

Zammitto, V. (2008, July). *VISUALIZATION TECHNIQUES IN VIDEO GAMES*. Paper presented at the EVA London Conference, England. Retrieved from http://www.sfu.ca/~vzammitt/papers/zammitto-EVA-Visualization_Techniques_in_VideoGames.pdf

List of Figures

Figure 1.1 Game-Masters-Image-Gallery-01. From Games Masters, Inside the exhibition. Photography by Mark Gambino, Image Courtesy ACMI. Reprinted with permission.

Figure 1.2 Game-Masters-Image-Gallery-02. From Games Masters, Inside the exhibition. Photography by Mark Gambino, Image Courtesy ACMI. Reprinted with permission.

Figure 2.1. The text mining framework, as outlined by Tan (1999, p.2).

Figure 2.2. Operational Prototyping diagram, as outlined by Davis. (Davis, 1992, p.74)

Figure 2.3. Operational Prototyping method revised to suit the project needs.

Figure 2.4. Linear level structure, as outlined by Kremers. (Kremers, 2009, p.56)

Figure 2.5. Semi linear level structure, as outlined by Kremers. (Kremers, 2009, p.56)

Figure 3.1. The faster paced forms of navigation in Doom II: Hell on Earth (ID Software, 1994).

Figure 3.2. The retro styled aesthetics and gameplay of La-Mulana (GR3 Project, 2005).

Figure 3.3. The arcade based vector graphics of Star Wars (Atari, 1983).

Figure 3.4. The dark and enclosed spaces of Silent Hill (Konami, 1999), demonstrating the dividing of "interiors into rooms and hallways" (Wolf, 2009, p.165).

Figure 4.1. An example of the blueprint system within Unreal Engine 4.

Figure 6.1. Text refining being undertaken within an excel spreadsheet.

Figure 6.2. The filtering and categorizing of data through knowledge distillation.

Figure 6.3. An early iteration of the user interface within Unreal Engine 4.

Figure 6.4. A screenshot of the generation one PONG (Atari, 1975) inspired level.

Figure 6.5. Pitfall (Activision, 1980) inspired generation two level that was developed.

Figure 6.6. Pitfall (Activision, 1980) for the Atari 2600, an early example of the 2-D side scrolling platformer.

Figure 6.7. Pitfall (Activision, 1980) inspired generation two level that was developed.

Figure 6.8. Screenshot of the generation two platforming level.

Figure 6.9. Super Mario Bros 3 (Nintendo, 1988) one the most popular titles for the NES.

Figure 6.10. Screenshot from Super Mario Bros 3 (Nintendo, 1988).

Figure 6.11. Screenshot from the third generation level.

Figure 6.12. The Legend of Zelda: A Link to the Past (Nintendo, 1991), the key precedent for the fourth generation level.

Figure 6.13. Boss battle from the fourth generation level. This level was built to represent the 16-bit top down adventure and rpg games of the early nineties.

Figure 6.14. Screenshot from The Legend of Zelda: A Link to the Past (Nintendo, 1991). Note the use of tile based sprites as backgrounds.

Figure 6.15. Screenshot from the fourth generation level as the character approaches a treasure chest.

Figure 6.16. Crash Bandicoot (Naughty Dog 1997) one of the earliest mascots for the Sony PlayStation and key influence on the generation five level.

Figure 6.17. The fifth generation level, intended to represent the early developments in 3D home console gaming.

Figure 6.18. Screenshot from Crash Bandicoot (Naughty Dog 1997). Note the darker areas and inability to see further in the distance saving both processing and rendering power.

Figure 6.19. Screenshot from the fifth generation level where rendering techniques were imitated to more closely resemble the early 3D technology.

Figure 6.20. The generation six level intended to imitate the larger environments and improved visuals of games such as Jak and Daxter (Naughty Dog, 2001) and Ratchet and Clank (Insomniac, 2002).

Figure 6.21. Jak and Daxter (Naughty Dog, 2001) was a major title for the PlayStation 2 and demonstrated the improved hardware and capabilities with larger environments and more detailed visuals.

Figure 6.22. Screen shot from the generation six level.

Figure 6.23. Screen shot from Jak and Daxter (Naughty Dog, 2001) showing the larger environments and increased draw distance.

Figure 6.24. Screen shot from the Journey (ThatGameCompany, 2012) influenced generation seven level.

Figure 6.25. Screen shot of the generation seven level. The focus of this level was developing unique lighting and shading visuals to imitate some of the indie game aesthetics.

Figure 6.26. Screen shot from Journey (ThatGameCompany, 2012) demonstrating the the unique visual style and gameplay found in indie games.

Figure 6.27. The original user interface and data visualization with the video capabilities added.

Figure 6.28. The revised user interface and data visualization with the altered video system.