User interface design: porting game technology concepts to applications

Angelia Wilson

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
User Interface Design:
Porting Game Technology Concepts to Applications

By: Angelia Wilson

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology

Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

9/17/2004
Student Name: Angelia Wilson

Thesis Title: User Interface Design: Porting Game Technology Concepts to Applications

Thesis Committee

Name          Signature          Date

Evelyn P. Rozanski, PhD
Chair

Andrew Phelps
Committee Member

Anne R. Haake, PhD
Committee Member

8-19-2004

8/19/04

8/30/04
Thesis Reproduction Permission Form

Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

Master of Science in Information Technology

User Interface Design: Porting Game Technology Concepts to Applications

I, Angelia Wilson, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction must not be for commercial use or profit.

Date: 10/12/04 Signature of Author: Angelia Wilson
# Table of Contents

1. INTRODUCTION ........................................................................................................... 5

2. THE EVOLUTION OF APPLICATIONS AND VIDEO GAMES
   2.1 The Personal Computer ......................................................................................... 7
   2.2 The Video Game Console
      • Challenge ............................................................................................................. 8
      • Socialize .............................................................................................................. 9
      • Dynamic Solitaire .............................................................................................. 9
      • Bragging Rights ................................................................................................. 10
      • Emotional Experience ....................................................................................... 10
      • Fantasize ............................................................................................................ 10
   2.2.1 The Progression of the Video Game Console ..................................................... 10

3. THE EVOLUTION OF THE USER INTERFACE
   3.1 Applications ........................................................................................................... 12
      3.1.1 Screen Design ............................................................................................... 12
         • 1970s .............................................................................................................. 13
         • 1980s ............................................................................................................ 14
         • 1990s ............................................................................................................. 15
   3.2 Video Games ......................................................................................................... 16
      3.2.1 Screen Design ............................................................................................... 16
         • 1970s .............................................................................................................. 17
         • 1980s ............................................................................................................ 18
         • 1990s ............................................................................................................. 19

4. APPLICATION DESIGN VS. VIDEO GAME DESIGN
   4.1 Design Elements/Expectations ............................................................................. 20
      4.1.1 Consistency ................................................................................................. 20
      4.1.2 Expected Outcomes .................................................................................... 21
      4.1.3 Real World Expectations ............................................................................ 22
      4.1.4 Feedback .................................................................................................... 22
      4.1.5 Accomplish a Task Incrementally ............................................................... 23
      4.1.6 To Fail ....................................................................................................... 24
      4.1.7 Not Get Hopelessly Stuck ........................................................................... 25
   4.2 Requirements vs. User Expectations ..................................................................... 26
      4.2.1 Experience-based Education ....................................................................... 27
      4.2.2 Experience-based Communication ............................................................... 27
      4.2.3 Experience-based Therapy ......................................................................... 28
5. DESIGN CONCEPTS POPULARIZED BY VIDEO GAMES

5.1 Overview ................................................................................................................. 29

5.2 Personified Help ........................................................................................................ 29

5.2.1 Video Games ........................................................................................................ 30

5.2.2 Case Studies .......................................................................................................... 31

- The Legend of Zelda: Ocarina of Time ................................................................. 31
- Master of Orion III ..................................................................................................... 32
- Metal Gear Solid 2: Sons of Liberty ........................................................................... 32

5.2.3 Applications .......................................................................................................... 33

5.2.4 Case Studies .......................................................................................................... 34

- Microsoft Bob ............................................................................................................ 34
- Office Assistants ........................................................................................................ 35
- Windows XP: Rover .................................................................................................... 36

5.2.5 The Future ............................................................................................................. 36

5.3 Customization ......................................................................................................... 37

5.3.1 Video Games ........................................................................................................ 37

5.3.2 Case Studies .......................................................................................................... 38

- Diablo II ...................................................................................................................... 38
- Time Splitters 2 .......................................................................................................... 39
- Sims ............................................................................................................................. 39

5.3.3 Applications .......................................................................................................... 40

5.3.4 Case Studies .......................................................................................................... 41

- my.yahoo.com ............................................................................................................ 41
- Microsoft Office 2000 ............................................................................................... 41
- Window Blinds ............................................................................................................ 42

5.3.5 The Future ............................................................................................................. 42

5.4 Personification ....................................................................................................... 43

5.4.1 Video Games ........................................................................................................ 43

5.4.2 Case Studies .......................................................................................................... 44

- Tony Hawk Underground ........................................................................................... 44
- Sims Online ................................................................................................................... 45
- Black & White ............................................................................................................. 45

5.4.3 Applications .......................................................................................................... 46

5.4.4 Case Studies .......................................................................................................... 47

- AOL Instant Messenger 5.5 ...................................................................................... 47
- Orkut ............................................................................................................................ 47
- CokeMusic.com .......................................................................................................... 47

5.4.5 The Future ............................................................................................................ 48

5.5 Sensory Feedback .................................................................................................. 49

5.5.1 Video Games ........................................................................................................ 49

5.5.2 Case Studies .......................................................................................................... 49

- Rainbow Six: Raven Shield ...................................................................................... 49
- Mario Kart .................................................................................................................... 50

5.5.3 Applications .......................................................................................................... 50

5.5.4 Case Studies .......................................................................................................... 50
• DigiScent iSmell .................................................................................................................. 50
• Logitech iFeel Mouse ......................................................................................................... 51
5.5.5 The Future ..................................................................................................................... 51

6. CONCLUSION .................................................................................................................... 52

APPENDIX A
Personal Computer Timeline .................................................................................................. 53
Video Game Console Timeline ............................................................................................... 55
Application User Interface Timeline ....................................................................................... 58

REFERENCES .......................................................................................................................... 60
Images ...................................................................................................................................... 63
1. INTRODUCTION

Gaming technology has consistently been more advanced than its application counterpart. Although the first computers were created in the 1930s, the average user didn’t have the accessibility to even use one until the first personal computer was released in 1974. Despite that it was possible for anyone to purchase one of the personal computer kits, only those that were interested in electronics tended to buy one. It wasn’t until the 1990s that computers started to become a more common household item, yet even by the year 2000 only about 50% of households owned a computer (Bellis). Video games have a drastically different timeline in that they were available to homes two years earlier than a personal computer in 1972 and they were already popular by 1977 when Atari released the 2600 (CNN.com). Due to the nature of video games, players are always seeking out a new game once they have beaten the one that they are playing. For this reason video games are released quite frequently and so they are able to take advantage of the newest technologies available.

Many of the advanced concepts that were popularized by video games have since been moved to the application domain. Although many of these concepts succeeded once moved to an application, there are also many that failed. In some instances a certain concept may have been attempted in an application prior to its success in a video game, although they may not have succeeded because users were not prepared for the new technology. Video game developers are able to refine a specific concept to the point where it works extremely well for the player. Applications that convert these concepts and introduce them slowly tend to be more successful, however those that integrate the concept in an almost identical manner as the video game can be too far advanced for many users.

While there are many concepts that have gone back and forth between video games and applications, four important ones are: 1) Personified Help – help tools that have human characteristics such as Microsoft’s Paperclip, 2) Customization – the ability to personalize different aspects of a game or application, 3) Personification – adding a deeper level of human characteristics that specifically mimic the user/player and 4) Sensory Feedback – feedback that can be interpreted by one of the five senses. These concepts have been refined to a point where they are extremely successful in video games because developers have been using them for some time. This has allowed the developers to enhance aspects that did not succeed the first time in order to create the end product that integrates seamlessly with the video game. Many concepts, such as Customization, have been around for so many years that video game developers have perfected its use so players fully understand how to use these concepts properly. Other concepts, however, like personification, have been around for a much shorter period of time and so there is still some debate between developers regarding how to best integrate it within the game.

Throughout the road to success for video games, applications may also have been attempting to integrate the concept with varying degrees of success. In some cases the concept simply did not work because the developers were not able to integrate the concept seamlessly with the application. Sometimes concepts that make complete sense in video games do not always make sense as part of an application. There is a fine line between concepts that will work in an application and those that will not. In the example of Customization, since it has been
around so long and is so successful in video games, applications have been able to determine how to best integrate the use of customization so that users are able to take advantage of it. Personification, on the other hand, is not as successful in applications yet because it is still being determined how it is best used in video games. Yet given the number of applications that exist that are attempting to integrate personification, it is clear that this concept will continue to grow.

All four of these concepts are good examples of how video games have taken a concept and perfected it prior to an application being able to reach the same level of perfection. Since video games are released so frequently they have more capability of perfecting a concept than an application. A video game’s sequel may be released within a year if not months of the first game, however applications tend to only be released yearly if not less frequently. This allows video games more opportunity to take a concept that may not have worked in the first game and bring it to a higher level. Applications are able to utilize this as a test bed for new concepts and can therefore pick and choose from those concepts that have been successful. It allows applications to integrate new technology without having to test it on a new market themselves since the feedback of its success in a video game already exists. As video game technology continues to advance it will become the basis for even more application interface concepts.
2. THE EVOLUTION OF APPLICATIONS & VIDEO GAMES

2.1 The Personal Computer

The Merriam Webster Dictionary defines a personal computer as “a general-purpose computer equipped with a microprocessor and designed to run especially commercial software (as a word processor or World Wide Web browser) for an individual user” (m-w.com). Computers have been around since the first one was created in the 1930’s, however it wasn’t until 1974 that the first personal computer was created (Bellis). At that time the personal computer made it possible for the average home computer user to have a computer that didn’t take up a full room and cost a fortune. The personal computer is well on its way to becoming a mainstream item, just over half of all homes in America own at least one. In 1981 only 1% of homes owned a personal computer; that number grew significantly to 51% in 2000 (Grant). Personal computers allow users to browse the Internet, receive and send email, and stay in touch with family members. Personal computers have revolutionized the way that a lot of people live by supplying them opportunities that they otherwise would not have had. The Internet itself has opened up a wide array of information that used to be difficult for the average person to come by. The personal computer has completely revolutionized the way that a lot of people live and do business.

2.1.1 The Progression of the Personal Computer

The personal computer has gone through many changes since it was first introduced (see Appendix A: Personal Computer Timeline). The Scelbi computer, which was a personal computer kit released in 1974, has the honor of being the first real personal computer, however the Altair 8800 that was released the following year was the kit that really sparked interest among consumers due to its expandable parts and reasonable cost. The Apple I, released in 1976, was the first personal computer that was reasonably priced that came fully assembled and therefore sold very well. Over the next five years other companies tried to capitalize on the success of the Apple but none were able to create something that was considered exception. It was not until 1981 when IBM released their “PC” that there was a lot of excitement over a new computer. The PC was fully expandable, unlike the other computers on that market at the time. This was extremely appealing to computer owners since they had the opportunity to upgrade their computers should they be interested in it. The PC also came packaged with its own operating system, MS-DOS 1.0, created by Microsoft (Bellis).

PCs ruled the market for years with plenty of companies popping up that would create a fully customized computer for the consumer. Companies like Gateway and Dell have been very successful in creating personal computers for the average consumer by creating a computer with the newest technology based upon the consumer’s requirements. Apple made a slight comeback when they released their iMacs in 1998. This line of computers sold well for Apple due in great part to the multiple colors that the cases came in. Consumers liked the new look of personal computers as it gave a fresh face to the market and allowed them to select the color that suited them. Apple’s success with the newly
designed PC prompted other companies to start making their products more interesting. PCs still have a huge lead over Macs and their popularity cannot be denied, however Apple has slowly been making strides at gaining more market share and is one company that has truly stayed the test of time.

2.2 The Video Game Console

A game is defined by the Merriam-Webster dictionary as an “activity engaged in for diversion or amusement” (m-w.com). Video game consoles have been appearing in people’s homes since the early 1970s and have continued to grow in popularity ever since. Some companies forecast that by 2005 70% of homes in the United States will own some sort of video game console (Grant). The growth in technology has made it possible for video games to become more and more involved and detailed. Video games introduced color in 1975, but it was not until 20 years later with the release of the Sega Genesis in 1995 that gamers got their first glance of 3D graphics (CNN.com). Video games have become so detailed not only with their graphics, but also with their storylines that they provide the player with a virtual reality experience in which they are able to portray another character.

Video games create a fantasy world where the player is the hero who must overcome challenges in order to win. The most popular and memorable video games provide a compelling storyline that is able to draw the player into the game world. Players who are able to beat a video game feel a sense of achievement and typically come away from the situation having learned something. What particularly stands out from typical games and video games is their level of interaction with the player. Regular games typically revolve around interaction with other players, while the majority of video games revolve around how the game reacts to the player, which can be a very compelling reason to play a game. There are also other concepts that draw a player to a video game, including the challenge, to socialize, to play dynamic solitaire, for the bragging rights, for the emotional experience, and to fantasize. All of these elements help to create games that are unique and entertaining and are the reasons why players continue to come back for more (Rouse III).

- Challenge

Players can be entertained for hours by the challenge of a video game. The challenges that a player faces while playing a game is what keeps them interested in trying to beat it. Any time that a person is faced with a challenge that they are able to overcome they walk away from the situation having learned something, a video game is no exception. Oftentimes the lessons that are learnt from a video game revolve around how to beat a level, but in some cases the lesson can be applied to other parts of the player’s life. For example they may be able to “apply problem solving methods to their work, use their improved spatial skills to better arrange their furniture, or perhaps even learn greater empathy through game role-playing”. When a player is able to find a solution to a challenge they are filled with a great sense of satisfaction. Challenging video games
gives the player more than just an entertaining experience, it provides them with an opportunity to learn and grow (Rouse III).

- **Socialize**

People often play games (specifically board games) with a group of players in order to get together and socialize. They play these games because they enjoy spending time with their friends or family and they welcome the opportunity to participate in something more social than going to a movie. Most video games are designed to be single player games, but there are also plenty of multi-player games available. A group of friends will often get together to play games like *Quake III* over their LAN (Local Area Network), making this experience very social as players shout to one another across the room. Even when these games are played over the Internet, when the other players are not in the same room, they have the ability to socialize using chat tools. Many players will send even single word chats during the most intense moments of a game, which is a testament to their desire to have a social experience (Rouse III).

- **Dynamic Solitaire**

For all the reasons that many players are looking for a social experience, exactly the opposite is true for people looking for a completely anti-social experience. Sometimes friends are not available, or sometimes players simply do not feel like socializing with other people, so during these times players want a single player game. Video games differ from other activities that people might do by themselves, such as reading, in that they “fake” human like experiences. Video games often require players to interact with other characters in the game, which imitates socialization between the player and the video game. This type of socializing, however, can be less intimidating or stressful than interaction with another human being. Video games offer a means of interaction that is completely scripted where there isn’t any pressure on the player to act in a certain way. Oddly enough this can be seen as a dynamic, interactive experience that is completely antisocial (Rouse III).

- **Bragging Rights**

A player who has beaten a game loves the opportunity it gives them to brag to their friends that they beat it first. Oftentimes the player who was able to beat a video game first is viewed as the most skilled and determined player. Similarly, someone who has won a battle in a multi-player game also gets to brag about it and hold that over his friends. It also, in some cases, wins the player some respect from their friends because it shows that they are more talented at playing the video game. Even if the player does not brag to their friends about beating a game, there is a certain satisfaction that they feel knowing they were able to complete something that was challenging, like obtaining a special item or a prestigious title. Players who have beaten a challenging video game have shown to themselves that they are good at something, which makes them feel good about the experience. There is a sense of pride that comes with winning a video game that is unlike other team-based experiences because winning is based solely on the
individual player’s skills. There can be no questioning that a player who is able to beat a video game is skilled at it (Rouse III).

- **Emotional Experience**

Video games, such as *Doom*, provide players with an adrenaline rush that is unmatched in many other game experiences. Unfortunately, many video games’ emotional experience is limited to this type of excitement and tension and does not go beyond an adrenaline rush. Arcade games are designed in such a way that they are unbeatable, yet players love to continue to try and win. Although players are continually defeated, they do not stop playing the game, so these types of games are an excellent lesson in defeat. The feeling of helplessness that a player feels while trying to beat the unbeatable cannot be ignored. Game designers have been wary of adding too much emotion into a game, so there are few examples, but the video game *Planetfall* is one exception. In this video game a beloved friend sacrifices himself for the player, causing a deep sense of sadness over a character that the player has become attached to. This experience has stuck in the mind of many gamers, and should therefore show to other game designers that involving emotional experiences in a video game can make for a very memorable experience (Rouse III).

- **Fantasize**

The art of storytelling revolves around creating an interesting set of characters and a plotline that is enjoyable to follow. These stories allow the listener to fantasize about an exotic world where the characters do exciting things. Storytellers help their listeners become part of a world in which they have only ever dreamed of becoming a part of before. Video games have the ability to take this a step further by completely immersing the player into the storyline as the main character. As this character the player gets to be the best at whatever activity they are performing, whether that is being a swordsman or a thief. Players get to act out doing activities that they would never actually get to do in their real lives, which provides them with a huge amount of entertainment (Rouse III).

**2.2.1 The Progression of the Video Game Console**

The first video game console was released in 1972 and was called the Magnavox Odyssey (see Appendix A: Video Game Console Timeline). The console was so basic that the graphics needed to be placed onto the TV screen and the score needed to be kept on paper. Atari released a few different version of the game *Pong* from 1975 to 1977, making it increasingly more advanced by adding color and on screen scoring. The Atari VCS/26000, released in 1977, allowed video game players to play different video games through the same console and was so popular that it was on the market until 1990. Nintendo released its first video game console in 1985, which started fans long running appreciation for the *Super Mario Bros* series. In 1989 Sega released the Genesis, which was the first video game console to be released with a 16-bit processor. This became very popular with video game players due to its speed. Many video game designers left
other companies to join Sega in order to develop games that utilized the latest technology (CNN.com).

The next decade became a battle for the top between Nintendo, Sega and Sony, who all continued to release new versions of their consoles with new games in the hopes of coming out on top. For five years, from 1989 through 1994 the Nintendo NES and Sega Saturn shared the top spot for sales and offered players a wide array of video games. In 1995 Sega released the Dreamcast, which initially had high sales, but dramatically fell off the charts when Sony released the Playstation later that same year. By 2000, one in every four homes owned a Sony Playstation. The battle for the best console continues now between Nintendo, Sony, and Microsoft’s Xbox, which was released in 2001. Sega dropped out of the market after the failure of Dreamcast and now focuses on making video games for the other consoles. Each console has technologies that players find appealing, and each has their own set of games that players have become attached to. Many players often own more than one of the preceding consoles so that they can play the games that are only offered to that specific console. The games that have become unique to the console is what helps each console maintain its market share because it entices players to purchase each console so that they can play their favorite games (CNN.com).
3. THE USER INTERFACE

3.1 Applications

The user interface design is a small portion of the study of Human Computer Interaction (HCI). Human Computer Interaction is the “study, planning, and design of how people and computers work together so that a person’s needs are satisfied in the most effective way” (Galitz). When a user interacts with a computer, whether through sound, touch, or some other method, they do so through software that they are able to understand how to interact with, this is the user interface. The user interface is made up of two main components, input and output. The input is what comes from the user, whether in the form of typing, clicking, or speaking. The output is what comes back from the computer, whether in the form of some sort of sound, or more commonly, a screen that displays the results of the user’s actions. A graphical user interface takes a step further by defining how the user interacts with the user interface, typically through some means of pointing (Galitz).

User interfaces have grown significantly since the first one was introduced at Xerox PARC in 1973 (see Appendix A: Application User Interface Timeline). This user interface contained many components that are still in use today, like menus. The concept of pointing and clicking things with a mouse was also developed by Xerox and was created in 1981. The Apple Macintosh, released in 1984 was the first computer to have a graphic user interface (GUI) so that screens were made up of more than just text. In 1985 Microsoft answered back with their version of a GUI, Microsoft 1.0. Apple released the Macintosh II in 1987, which was the first personal computer to feature a color monitor. Ever since Microsoft released Windows 3.1 they have virtually dominated the market of operating systems (Galitz).

3.1.1 Screen Design

Screen design and layout have come a long way since the first GUI screens were created in the 1970s. When the first computer applications were released very little time or thought went into the design of the screen. Screens were often archaic and difficult to understand due to developers shortening field names to acronyms that did not always make sense. The 1980s proved to be more focused on screen design as fields were laid out in such a way that users could easily determine what data was supposed to go in which fields. Fields that were similar were grouped together so that users will be able to see which fields belonged together, making it easier to sometimes determine the context of certain fields. The 1990s went a step further by utilizing the 3D controls that were now available. Since the 1970s it has been proven how important it is for users to be able to enter data quickly and efficiently. User interface design has become a forefront issue with regards to screen design and therefore it has become an important aspect when designing a new system (Galitz).
In the 1970s screen design and layout was not much of a concern to programmers writing the code, especially since the need for user-friendly screens was not even an issue that was known. These screens typically were made up of many fields that flowed one after the other, often with difficult to remember or understand acronyms that were used in order to try and save space. The help systems of these screens were also very archaic consisting of error codes that provided no instruction as to what the error was (See Figure 1). Users needed to look up the error code in manuals in order to determine where the actual error arose. These screens took a lot of learning and a lot of patience to be able to use (Galitz).

![Figure 1](image.png) 1970s style screen design (Galitz)
The 1980s brought about guidelines that developers could implement in order to create more user-friendly screens. These guidelines helped to create screens where similar concepts were grouped together and meaningless acronyms were removed. Commands that the user could execute were also placed prominently on the screen so that the user was not required to remember every step that needed to be taken to execute a program properly. User entry became a much higher priority during this time and so screens included hints as to what type of data was expected (such as the class type in figure 2). These screens were not one hundred percent user-friendly, but they were a huge improvement to the screens of the 1970s (Galitz).

![Figure 2 - 1980s style screen design (Galitz)](image-url)
In the 1990s operating systems were now able to display different types of graphics. This was helpful to designers because they could group items together in an even more obvious way by using borders and headings to display meaning. Function keys were also no longer needed since the new GUIs could display buttons that could be clicked to execute a command rather than the necessity to press an arbitrary key on the keyboard. Additionally, a multitude of new data entry fields were made available including drop down boxes to display accurate data selections. Screens became simpler since data could be split between multiple screens, which had been impossible to do before (See Figure 3). Developers were able to do this because computers were now fast enough to display new screens in a reasonable amount of time (Galitz).

Figure 3 – 1990s style screen design (Galitz)
3.2 Video Games

Some of the original games released in the early 1970s were solely text-based where players would be presented with a text explanation of their surrounding and were then asked how they wanted to proceed. The reason that this was done was mostly due to the difficulty in creating quality graphics. In many cases creators did not even have graphics capabilities since they were developing the game in a Unix system, so text was the only option they had available. By using text explanations the developers could create games that were much more involved than games that had graphics because the graphics were so limited. Eventually some games started to use alphanumeric characters laid out in designs in order to generate graphics in conjunction with text explanations. This was seen as the first real bridge into creating video games that had true graphics as well as a good storyline. As the graphics capabilities progressed around the 1980s, developers started to create screens that could scroll. These screens had anywhere from two different screens to six and created the concept of off-screen movement (Wolf).

In the 1970s developers used depth perception as a way to portray three dimensions, for example to ensure that items that were supposed to be closer to the player were larger, while those that were supposed to be further away would be smaller. In the 1980s developers started using three dimensional looking graphics as the background of the video game, however the characters that were used to play the game never got larger or smaller to show any movement along a z-axis plane. It was not until the 1990s when the Sega Genesis was released that video games started using 3D graphics that were generated based upon movement within the game. Three-dimensional graphics added a lot of realism to video games that did not exist before. These days, developers can create worlds that appear almost identical to the real world since this technology now exists (Wolf).

3.2.1 Screen Design

Screen design in the early 1970s, when video games were first released, was based solely on what the developers had available for options, which was extremely limited based on the technology available. Pixel sizes needed to be large simply because that was all that was available, and since pixels were so large everything needed to be based off a square. As technology progressed the pixel size started to get smaller and by the mid 1980s it was so small that it was not even noticeable. These days the pixel size can be so completely unnoticeable that developers have been able to create 3-D graphics that are so good many people consider them to be extremely realistic. Video games today have a sense of realism in that when a character moves the player can actually distinguish movements that real humans make. In the 1980s developers were happy if they could make the legs move so that the character even looked like it was walking. Advancements in technology have made screen designs what they are today, as technology continues to progress one can only assume that video games will become more and more realistic looking as time goes on (Poole).
• 1970s

The first video games that were released in the early 1970s featured very basic graphics that usually consisted mostly of lines. These games had very few moving parts and had very simple actions that the player could perform. Despite the simplistic appearance of these screens, they were still challenging to users and provided a lot of entertainment. During this time elements were not selected based on how the designer wanted them to appear, rather it was based on what the designer had for choices, which was very limited (See figure 4). The ball in the game *Pong* was actually a square ball not because the designers wanted it that way, but because that was the only option they had available. By the end of the 1970s screens were more complex than their predecessors just a few years earlier. These screens were in color and included graphics that consisted of more than just lines and dots. These video games were made up of characters, all be it simple compared to today’s standards (Mayfield).

![Figure 4 - 1970s style screen design](image)
1980s

The early 80s produced many games that were similar to the games released in the late 70s simply because the video game consoles that were released during this time were not much more advanced than those released in the late 70s. In the mid 1980s Nintendo released its first console, the NES, and drastically changed the way that video games looked. *Super Mario Bros*, which is one of the most popular video games of all times, included graphics that were far more advanced than those of the 70s (See Figure 5). In the late 1980s Sega released its first console, the Genesis, which also made a lot of advancements in video games technology, not only in graphics, but in speed as well. The graphics of the games in the 80s were far more advanced than those from the 70s and brought about a huge change in how video games were structured by creating the idea of a detailed storyline. Storylines made video games more entertaining to play by giving players an end goal that they were trying to reach.

![Figure 5 - 1980s style screen design](image)
Games of the 90s have grown so much that they have “evolved to create 3-D virtual worlds with lifelike sounds, elaborate graphics and complex plots” (Mayfield). These games had continued to grow increasingly life like to the point that some people insist that if they had walked into the room to see a game of *Madden Football* on the screen that they would have assumed it was a real game they were watching. Video games of the 90s continued to push the envelope regarding how complex the graphics could be, and they were very successful (See Figure 6). Many of these games are so engrossing that it started to cross the line between reality and fiction. The video games of the 90s had graphics that were so compelling that players would play for hours or even days and not even realize how long it had been since they started playing. These video games paved the way for the graphics that exist today, which continue to get increasing more realistic.

![Figure 6 - 1990s style screen design](image-url)
4. APPLICATION DESIGN VS. VIDEO GAME DESIGN

4.1 Design Elements/Expectations

Video games have grown considerably in many different ways since the first arcade games were introduced in the 70s, everything from graphics to storylines. Despite this, there are some central concepts of video games that still remain the same. Applications have also grown considerably by adapting new technology, but there are still central concepts that users expect when interacting with an application. These expectations are what solidify an application or video game in the eyes of the user or player. When their expectations are not met users/players will be extremely disappointed with their experience. When Xerox was designing the GUI for STAR, which was released in 1981, they laid out several design elements that they felt users would expect from the system. These design elements ranged from maintaining consistency to being able to reverse actions that produced an unwanted result (Galitz). These design elements are still important today to both video games and applications, otherwise the expectations of the user/player will not be met.

4.1.1 Consistency

In order for the user to be able to learn how to use a new application appropriately, the GUI must be consistent. This helps the user understand where they can find specific elements when navigating between screens. Consistency can take the form of using the same font, shapes, sizes and location of specific elements on the screen to an overall consistent appearance for the application. Consistency is a necessary part of an application because it aids the user in learning where to find its key components. Users will become frustrated with applications that are inconsistent because they will not be able to quickly locate the necessary items to complete their tasks. It must be easy for a user to learn where certain components are in order for the application to work effectively for them (Galitz).

Players of video games expect to be able to understand why a specific action happens. For example, if they are playing the game Tekken they expect that if they miss a kick while fighting it will be because their opponent jumped away or blocked the kick. If they miss for no apparent reason, then the world they are in will not make sense to them. When a player does not understand what actions produce what reactions then the world is inconsistent to them and they cannot appreciate the outcomes. The more a player interacts with a game the more they will begin to learn what actions cause what reactions, so it is safe to assume that a beginner will not have a full understanding of this. A player must be able to determine the consistency of the world quickly however, or they will lose interest in the game (Rouse III).

Consistency in both applications and video games is a necessary design element because it helps the user/player become acclimated with the new environment. Inconsistencies can include visual design elements that do not match on all screens, including fonts and colors, but more specifically, where specific items are located. These inconsistencies can
make a huge impact in both applications and video games because it could mean failure for the user/player. It can be even more frustrating for users/players when a certain action does not generate the expected outcome. The user/player will typically have preconceived notions of what a specific action will do, so when that outcome does not happen it can be extremely disruptive to the user’s/player’s perception of how to interact with the application or video game. When this happens they will typically give up and discard the application or video game in exchange for one that works better for them.

4.1.2 Expected Outcomes

Objects that are selectable or that can be manipulated must be clearly evident to the user. The user must also be able to understand what it is that they are supposed to do with those objects. For example, if there needs to be a way for the user to execute a task they must know what to do in order to execute that task (like press a button), so that button must be clearly identifiable by the user and they must know that they are intended to click it. Additionally, they must also be able to understand what the consequences are for pressing the button. For example, if a button on a form said, "submit", then the user must know 1) that they can click the button, 2) that clicking the button would submit the form and 3) when the appropriate time would be to click it. A screen must be intuitive to the user in order for it to function properly, otherwise the user will end up executing tasks incorrectly, causing the application to not work properly (Galitz).

It should be rather intuitive to a player what actions are allowed in a game and which are not. It is not necessary to understand exactly what actions need to be performed in every instance, but it is necessary for them to understand what actions can be performed. For example, in the game Doom players learn early on that the point of the game is to shoot and kill all demons without being hit or killed themselves. It would have been completely counter-intuitive for the game designer to have a demon at a later point in the game that the player was supposed to talk to before killing because the player would never understand that due to the expectations laid out in the early parts of the game. The player expects that the actions performed in early parts of the games will relate to actions that need to be performed at later points in the game (Rouse III).

Understanding the expected outcome for performing a certain action can mean success or failure to the user/player. There is obviously a learning curve for a brand new user/player to an application or video game, especially if this is their first experience. Once this person understands the basic functions, for example what a button looks like and the concept that it executes some sort of task, then they will have expectations regarding what the outcome of pressing it would be, i.e. that it would "do something". These expectations can also include something as simple as what a specific icon would mean from application to application, for example if a designer were to use a disk icon (which has the connotation of saving) to mean something like to access a disk, then the user would be confused when pressing the button did not save their current work. Designers either need to stick with accepted standards or make it obvious what will happen when they do not.
4.1.3 Real World Expectations

An application typically utilizes what the user already knows about the real world in order to portray actions in a different medium. A user has the ability to manipulate these objects (like windows) in a way that makes sense to them so that they can easily adapt to the new system. Application environments and actions are based on real world tendencies and needs. For example, an application like Microsoft Word that offers users an electronic medium for writing is based upon the need to physically write something on paper. In that sense, users are able to perform many of the same actions in the application that they can in the real world, like writing in different colors or deleting text that they did not intend to write. Since applications are an extension of the real world it is necessary for the user’s understanding that the application not stray so far from the real world that it no longer makes sense to them (Galitz).

Many video games that are based around the real world tend to have a problem with making their solutions intuitive to everyone. As a player solves puzzles in a video game they come to expect that certain solutions will work. Real-world puzzles often have more than one reasonable solution that will work, however video games often only allow one of the solutions to actually solve the puzzle. This constraint is mostly due to the fact that in order to incorporate all solutions the game creator would need to program each solution into the video game, which can be very processor intensive. This can become extremely frustrating to players who know that the solution they have provided would work in the real world, however it fails in the video game. Video game designers need to be explicit in making sure that all possible solutions to a puzzle will actually work in their video game, otherwise it will appear to the player as if the game is faulty or has bugs (Rouse III).

Users and players have come to expect a great deal about the real world simply from living in it everyday, so it is imperative to their understanding that the environments in which they work or play function on those same expectations. Even when a video game creates its own virtual world there are still many elements that function in the same way as the real world, for example limbs that move the same way that animals or humans would, grass and water. Providing these elements helps the player feel as though they are in an understandable world, despite other areas that may be completely different. In the same way, metaphors used in applications are all based on the real world, for example opening a file is portrayed in the same way as finding a folder and opening its contents. Although each metaphor might not be obvious at first, the user/player is eventually able to discern the meaning and can typically relate it to something in the real world.

4.1.4 Feedback

When a user performs an action in an application it is important to provide them with feedback so that they know if their action succeeded or failed. This allows them to determine what steps to take next, if they succeeded they can move on to another task and if they failed they can fix the problems that arose. This feedback should be provided
throughout various steps during an action so that the user can continue in an incremental pattern until they have reached success. Without any sort of feedback users will become lost and frustrated regarding what has happened. This will often lead the user to making mistakes, even if they have performed every step appropriately because they will not know if it succeeded or not. This could very easily happen if a user is adding a new customer to their database and they press the “submit” button and nothing appears to happen. In this case the user would most likely press the “submit” button again, adding the same customer twice, if not more, to their database. Feedback provides users with a sense of awareness regarding how they are doing in completing their task and also lets them know when something has gone wrong (Galitz).

One element that makes really good video games great is that it gives the player the opportunity to make a lot of their own choices. At the same time the video game needs to provide enough direction to the player that they are capable of determining what they are supposed to be accomplishing next. The player expects enough direction to understand the plot of the game and to be told, no matter how subtle, what task should be performed next. This is a difficult balance to maintain in a game, giving the player enough direction to understand what to do next without telling them explicitly what they need to do. Players want to create a solution to a game that is all their own so that their experience does not mimic everyone else’s, in order to do this they need to be able to complete tasks in their own order. While allowing users the freedom to complete tasks out of order, they also will need direction as to what tasks need to be completed in order to stay on the right track (Rouse III).

Feedback is a necessary component for any successful application or video game. Feedback not only tells the user/player if they have made a mistake, it also tells them when they have completed something successfully. This helps inform the user/player that the task they are accomplishing is either on the right track or has derailed and they must fix something. Without feedback it would be common for the user/player to make a mistake. The task would most often be completed incorrectly if the user/player were left without any guidance whatsoever. It would be as if someone was given all of the ingredients to make a casserole but was not given directions on how to mix or bake them. It would be possible that an experienced person would get it right, but very unlikely that a novice would be able to figure it out.

4.1.5 Accomplish a Task Incrementally

While using a screen certain elements are important at different times based upon how the user is interacting with the screen. It is important that at the proper time these elements stand out to the user so that they understand their importance. This structure helps the user to navigate properly on a screen so that they can easily identify what step they are to complete next. Color or sound is a popular way that these types of cues are portrayed to the user. For example, when a user is updating a product in their database it does not make sense that they can update the record until they have actually changed a piece of data. Once they have changed something, however, it would make sense for the “update” button to become active. This helps the user understand that when the “update”
button is active they have the ability to update the record, but until that time they cannot update it. The flow of these steps helps the user know what to do next (Galitz).

There is typically one overall goal that the player is trying to accomplish when playing a video game. In order to make a video game entertaining and lasting, it is necessary to make the end goal difficult to reach, so there needs to be sub-tasks that the player must accomplish before they can complete the main task. These sub-tasks also serve as a method of positive reinforcement to the player that they are on the right track to winning the game. Without these hints that they are doing the right things, it can become confusing and frustrating to a player, even if they are doing the right things, because there is nothing telling them either way. It can make the video game feel as if the end is nowhere in sight, so the player will often give up (Rouse III).

Breaking down a task into sub-tasks helps users/players complete one larger task in an incremental order so that the one large task does not seem so daunting. In order to complete tasks incrementally, the application or video game must provide hints or clues as to what needs to be accomplished next. This helps provide an order to the game that otherwise would be difficult to understand. Video games are typically good at breaking a game down into sub-tasks, for example collecting a key before a treasure box can be opened. Applications also provide similar incremental tasks, for example clicking on an “edit” button before the user is actually allowed to edit a record. Sub-tasks are an excellent means for providing the user/player with guidance towards completing the overall task. It helps them not only know what they should be doing next, but it also helps them know that they are on the right path towards completing the larger task at hand.

4.1.6 To Fail

Failure is something that should never occur when using an application and if it does it should be very minimal. It is expected that users will occasionally make mistakes when using an application, i.e. user error, however it is imperative that the application handle these situations gracefully. Failure is certainly not something that is seen as a good trait in an application, so it must be accounted for in virtually every situation. It is very common for a user to type incorrect data into a form, so it is a necessary precaution that the designers of the application create a solution to this problem. Applications should not fail or crash because the user makes an error, the application should provide an explanation for why the data entered is incorrect and than suggest a solution on how to fix the problem. When a user makes a mistake it should not be brought to their attention in such a way that it is their fault, in fact correctly designed applications will have very few user errors because the user will understand how to avoid them.

Players want games to be challenging enough that they cannot beat it the first time they play it. So in this sense a player expects to fail some puzzles that they are faced with in a game. Games that are so easily beaten tend to be boring for the player and they will not be interested in playing it again. At the same time, a game cannot be so challenging that when a player fails they cannot understand what they did wrong. A player expects to fail
certain puzzles throughout a game, but they do not expect to fail in understanding how to play the game. Any time that a player fails they need to be able to understand their shortcomings so that the next time they are faced with the same problem they are able to solve it. Another technique is to allow the player to easily win at the beginning so that they feel good about themselves and the game. Once the player is doing well and has an understanding of the game, slowly increase the difficulty level for each puzzle. This keeps them feeling good about what they have accomplished while keeping the video game challenging enough to be interesting (Rouse III).

Failure is rarely seen as a good thing, especially in applications, but it is sometimes bad in video games as well. Clearly applications or video games that continually crash are unacceptable to the user or player because it disrupts them from interacting with it. Users of applications are not looking for any sort of challenge, and therefore expect that all of the tasks they are trying to accomplish will be very straightforward and easy for them to understand. On the other hand, players expect a challenge when they are playing video games because otherwise it would not be very interesting to them, so in this case failure would be acceptable to a certain extent. Failure beyond comprehension would be frustrating because the player would not be able to come up with a solution that would solve the task at hand. Failure levels are obviously something that is very different in applications and video games, however they have the same premise. Failure is an expected outcome, however both users and players expect that they will be able to get around the failure in order to come up with an acceptable solution to the task at hand.

4.1.7 Not Get Hopelessly Stuck

It is important for a user to be able to reverse any actions that they may have completed in case they made a mistake. Making it easy to reverse an action allows users the comfort of knowing that if they make a mistake it can be fixed. Not allowing the user to fix an error leaves them stuck where they are with no solution to their problem. If the user gets stuck at a specific point in the application with no way to fix the problem that has arisen then the application has failed them. In many cases it is unlikely that the user will have direct access to the person who wrote the application that they are using, so they will not be able to come up with a solution until the developer fixes the problem, if they even choose to do so. This could force the user to either start over from scratch with the application that they are using, or force them to start using a new application. For this reason it is necessary to leave the user an easy out of any situation they might get themselves into when using an application. This will save the user and developer what could possibly be hours upon hours of headache (Galitz).

Nowadays it is unlikely that a designer will create a situation where players cannot back themselves out. It is, however, common for massively multi-player online games (MMOOGs) to have situations where a player can fall into an area where they are unable to get out without the help of someone running the game. Many older adventure games were famous for making it impossible to continue if the player had missed a key part earlier on in the game. This oddly made a game where the player could not fix the error they had made and yet the game continued, although it could not be beaten. Some games
provide places where players can fall into, so it is important to allow the player to escape from these situations, no matter how difficult the designer deems these situations should be. Allowing the user to get stuck in a situation that is impossible to get out of is counterproductive to the point of the game, so if a user is allowed to go somewhere in a game, they need to be allowed to get back out (Rouse III).

Allowing the user/player to back out of any situation that they get themselves into is another imperative design element that must be taken into consideration when developing a new system. Users/players who wind up getting stuck with no way out will be completely lost, and under no circumstances should they be forced to start over from the beginning. If a designer allows their users/players to get into a situation, no matter what it is, then there must be a foreseeable way out of it. It is common for users of an application to make a mistake, so these mistakes should not force the user to either complete the task incorrectly, or to start over in order to complete it correctly. Obviously a well-written application or video game would never let a user get into a situation that they cannot get themselves out of, but it is difficult to foresee all situations that a user might get himself/herself into, so it is important to have some workaround for these types of situations.

4.2 Requirements vs. User Experience

Software applications and video games are often designed with very different intentions for the final product. Software designers must be conscious of the real world by creating applications that seamlessly integrate with real world concepts. Video game designers, on the other hand, do not have to worry about creating a game based on the real world and often create their own worlds. Additionally, the design for a typical software application is often based upon a set of requirements that the application must be able to accommodate, while video games are designed based solely with the user experience in mind. While it is important for software applications to be based in a real world environment, there is no reason that an application cannot be developed with the user experience as a top priority (Swartout and van Lent).

The experience of the user is extremely important when designing a good video game because if users do not enjoy the experience then they will not play the game. There are two key components to a user’s experience that developers strive to create: 1) the goals that the user is trying to reach and 2) the environment in which they must reach them. There are three levels of goals that the designers try to create which are short-term goals that can last seconds, medium-term goals that can last minutes, and long-term goals that last the length of the game. Every component that goes into a video game’s creation helps to tell the story of what the user is trying to accomplish. A good storyline is created by the way all of these components are mixed together so that the game is not simply a sequence of events, but a complex environment that the user must fully engage in to win (Swartout and van Lent).

The storyline of a video game is one of the most important aspects to take into account when developing a new game. The story must be entertaining enough to captivate the
user’s attention so that they will remain interested until the game has been won. A solid storyline is not enough though, to keep the user interested if there is not enough freedom left for them to make their own decisions regarding the game. There is a difficult balance between having too much storyline and not enough freedom or having no storyline and too much freedom. A good video game has a solid storyline while allowing the user enough freedom that they feel they are interacting with the video game and not simply watching a movie. A solid storyline, however, often dictates that there are a limited number of choices that the user can make. It is for this reason that it is important the video game has the appearance that the user is making many of the choices based upon their own decision and not those the game has laid out for them (Swartout and van Lent).

It is possible to create some software applications with the user experience being an important aspect of design. Experience-based systems (which are applications that require the user to go through some sort of experience, such as education, communication, and therapy) can easily be designed with the user experience considered. Educational applications that have the ability to immerse the user into the experience have been proven to create “a profound sense of motivation and concentration conducive to mastering complex, abstract material” (Swartout and van Lent). Mental health professionals have also begun to use virtual environments in order to immerse their patients into experiences where they may have phobias or mental disorders as a method for treatment (Swartout and van Lent).

4.2.1 Experience-based education

Education and training are fields where user experience design could make a real impact on how well students are able to learn. An Army training tool that has been created, called Full Spectrum Command (FSC), is a system based on current ‘real-time strategy games’. The game is a simulation that is used to help educate recruits about combat in urban areas. Its goal is to “teach cognitive skills, including decision making, synchronization, and leadership, to light infantry company commanders” (Swartout and van Len). The game is played in the first person as players are led through standard military simulations that will help them learn more about real combat. The difference with FSC compared with traditional simulation games is that FSC makes an effort to ensure that the urban areas are more challenging and complex than rural areas, as is the case with real life combat. As the recruits get better with the simulation, the missions get increasingly complex (Swartout and van Lent).

4.2.2 Experience-based communication

Another video game that the Army has developed centers on communicating different aspects of the army to those playing the game, America’s Army. The game is a “first-person shooter” game whose aim is to “inform players about the Army’s core values and support recruitment” (Swartout and van Lent). The game is more realistic than many other “first-person shooter” games by encompassing aspects such as guns that jam and the concept of needing to reload a weapon. This video game is specifically tailored towards the audience that would be interested in joining the Army by providing missions
that utilize weapons while at the same time teaching them about the Army. This video game, however, does not portray any parts of Army life that may be considered unpleasant to the players, so in reality it does not portray every aspect of Army life (Swartout and van Lent).

4.2.3 Experience-based therapy

In experience-based therapy, patients are exposed to virtual environments that have situations where they have anxieties. The realism in the virtual environments is continually increased as the patient becomes increasingly desensitized to the situation that is causing them the anxiety. For example, in early sessions patients may be exposed to a cartoon representation of their anxiety, while in future sessions they may start using realistic virtual environments. Allowing the patient to slowly progress to increasingly more realistic situations allows them to cope easier with their fears. Without proper care and a fully tested system however, these types of systems could just as easily produce the opposite outcome by making the patient’s phobias even worse. It is still an area of debate whether or not these systems are tested enough to know if they will produce the wanted and expected result (Swartout and van Lent).

There are many differences between standard software systems and game systems, and it may be possible to bridge the gap between the two by starting to develop more standard software systems using the design techniques for video games. It has been shown that for educational and training systems it is useful to design the system based on how the user will interact with it. In order to maintain realism in these systems, however, they must still be designed using a realistic environment that the user is able to believe in. For games it is more important to have a realistic character than a realistic environment, but for learning and training purposes the user must be able to immerse themselves in the system. Additionally, educational and training systems tend to allow more freedom of choice than video games, which is also important so that players are able to experience the system in a more realistic, real world way. In video games players are satisfied with being limited to a number of choices, however real world situations tend to be freer. By utilizing these design techniques it is possible to make realistic systems more memorable and entertaining to its users, which will make the system a much more productive tool (Swartout and van Lent).
5. DESIGN ELEMENTS POPULARIZED BY VIDEO GAMES

5.1 Overview

Video games have always been on the cutting edge of technology by striving to take advantage of all the advancements that have been made since its design inception. Video game players have come to expect that each video game that is released will be better than the last because it will be able to take advantage of new technology that may not have been available before. Hardware technology is an industry that has grown rapidly and therefore better and faster technology is always available. Applications, on the hand, are not usually on the cutting edge of technology for two main reasons. First, applications take a long time to develop, so at the point when the development starts the designers are working with the latest technology. Unfortunately, by the time they are able to finally release their application years could have passed, making the technology they are working with old if not obsolete. The second reason that applications do not use the latest technology is because an application is more about its functionality then how flashy the graphics are. Application designers are more concerned about creating an application that meets all of its functional requirements than one that utilizes all of the latest technologies.

The video game player population is also a lot smaller than the application user population, so there are a large number of people out there who do not know what the latest technologies are or how to use them. This group of users is more than happy to have their interfaces remain stagnant and the functionality remain the same. In order to keep non-video game player users happy, applications try to introduce new concepts slowly. This gives them the opportunity to introduce new technology without overwhelming some users. Technology concepts that have proved successful in video games tend to make their way into applications. These concepts typically prove to be successful, but there are also plenty of instances where they have not been as successful. Four of these technologies and concepts that were originally made popular by video games are personified help, customization, personification, and sensory feedback. Although some of these concepts may have been attempted in applications prior to their success in video games, it has been video games that have refined the concepts and made them work. Once they have proven to be successful in video games, applications will usually take another stab at integrating that concept or technology back in.

5.2 Personified Help

Personified Help refers to help systems that utilize some sort of character who mimics human characteristics or traits in some way. These types of help systems give the user/player a character that they can relate to that try to aid them in learning about the application or video game. The idea behind a lot of personified help systems is that they help present the data in such a way that the user/player will be able to comprehend it better and where they may be willing to seek out help more often. Help systems in applications are particularly unfriendly and difficult to navigate, so personified help systems are meant to help alleviate the frustration that users feel when using them.
implementations of personified help failed miserably with the majority of users because it actually caused more frustration than it fixed. Video games, on the other hand, have quite consistently used personified help in order to portray information and it is often successful. Help information often comes from other characters in the game and players have become accustomed to expecting to learn about the game in that way. Depending on how it is implemented into the application or video game will determine how successful it is because personified help does not always make navigating and learning the help information easier.

5.2.1 Video Games

There are two main forms of help that a user might receive from a video game depending upon the type of game they are playing. For puzzle or solitaire type games, help is usually supplied through a brief tutorial at the beginning of the game. If the player finds him or herself confused during the game they can access the help information through a drop down menu. Action video games, on the other hand, will usually supply hints throughout the video game through the use of personified help. This information is often supplied by other characters in the game in the form of a hint. Help systems in action video games differ drastically from those in standard solitaire games because in order to make the game entertaining to the player it is imperative not to tell them exactly how to accomplish each task.

Help systems for solitaire games are often very basic because of the content of the game. It is very easy for players to learn how to play these games because they are repetitive and therefore once the player understands the rules to the game they no longer need to consult the help. Occasionally games like chess will provide players with a “teacher” who will help them understand what move should be made next and why. This type of tutorial helps players learn more detailed theories about a complex game by allowing them to play the game with a human-like challenger. This type of help is very informational and supplies the player with more of a lesson about the strategies in the game than details on what they need to do next.

Help systems in action video games differ greatly because the information supplied to the player is in the form of a hint, rather than explicit information on how to play the game. Action video games differ from solitaire games in that a solitaire game is played with the same rules but will have a different outcome each time the game is played. Instead, action video games have users complete tasks incrementally and the outcome of the results typically does not change how the game is played. For this reason the player is supplied with hints about how to complete a task in order to make the game more challenging and entertaining to play.

Action videos games supply hints to the player by weaving them into the storyline and having them supplied by other characters in the game. These characters are usually very easy to identify so that the player knows which characters they can get information from and which ones are trying to kill them. Occasionally they are characters who are always with the player, or that the player can contact through some sort of communication device.
at any point in the game. Other times this information comes directly from characters that can be found in towns or other areas that are considered “safe” within a game. These characters appear very non-threatening so that the player can easily identify them as someone who is not going to harm them. They do not carry weapons and often have a smile on their face.

Help systems in solitaire games are straightforward and easy to understand, it is fact-based information about how to play the game. Action video games, on the other hand tend to be a lot more vague about the information that they are supplying. This difference exists in order to ensure that action video games remain entertaining and challenging, because if the game told the player how to complete each task then they would not get any satisfaction out of playing it. It is necessary to provide players with the right amount of help so that they are able to complete the task and yet not give them so much help that they are just following directions and get bored with the game.

5.2.2 Case Studies

- The Legend of Zelda: Ocarina of Time – Nintendo released The Legend of Zelda: Ocarina of Time for the Nintendo 64 in 1998. This was an extension of the extremely popular The Legend of Zelda games and brought the game into the third dimension. The hero, Link, meets up with a fairy name Navi during the game whose purpose is to help guide him and provide him with information that he needs to succeed in the game (ign.com: The Legend of Zelda). The fairy is weaved into the storyline, as are the other characters that provide Link with additional help, in order for Navi’s presence to make sense throughout the entire game (Sterling).

To a certain extent the user has the ability to determine how active or passive the fairy is. If the user feels that they need a lot of help then they can keep the fairy out with them at all times. When the fairy is out he will help the player by pointing out things that might be of interest. The fairy will not tell the player how to perform certain tasks or what exactly they need to do, but he will help the player by giving them hints that will help point them in the right direction. Players who are more advanced may find that the fairy is frustrating and in the way. For example, if there are a lot of things that the fairy finds interesting then he will move all over the screen and can become very distracting. Even when the fairy is put way he will occasionally come out to tell the player something, so it is not possible for the player to completely turn the fairy off. The player does have the ability to ask the fairy to repeat a hint that he just told the player, but they cannot ask the fairy specific questions. Although the fairy is helpful to a certain extent, he can oftentimes become very annoying and therefore ineffective (Sterling).
- **Master of Orion III** – *Master of Orion III* was released in 2003 with much the same premise as the first two versions, to become ruler of the "universe". The player starts out as the ruler of one planet with its own unique species and the player must then expand their rule onto other planets by defeating some rulers while generating pacts with others (Polak). In order to do this the player is provided with a group of advisors who will give the player information upon request. Also, at specific point the game will provide the user with information that the developers felt was necessary to continue (Sterling).

Although this game does provide the user with some help, it is very passive and therefore the player must hunt down the information that they need. So even though the panel of advisors exist, the player must know who to speak with in order go get the information that they are seeking. Contrary to many other games, the advisors are not part of the storyline because there is no real storyline to the game, nor are they part of the game unless the player seeks them out for their advice. The only reason that they exist is solely to advise the player on what moves they might make next. This passive help can be frustrating and confusing to new users, who might have a difficult time tracking down the answers they need to move on in the game (Sterling).

- **Metal Gear Solid 2: Sons of Liberty** – *Metal Gear Solid 2: Sons of Liberty* was released for the PlayStation 2 in late 2001. This game mimics a lot of the strategy required for game play in the original *Metal Gear Solid*, which was released for the PlayStation. The storyline behind this one-man army kind of game involves a high-tech weapon that is about to come into the hands of the wrong kind of people, a resistance group called Dead Cell. This counter-terrorism game challenges players in vast areas of stealth and weaponry in order for them to succeed (ign.com: Metal Gear Solid 2).

Much of the help that comes from this game comes in the form of other characters sending the player messages across a codec that gives them hints on what they should be doing. Sometimes the information they send is useful while other times it is not, it’s up to the player to determine which it is. Players can actively approach other characters in order to try and get information about what they should be doing. Players in the tutorial mode of the game will receive messages regarding new items they acquire which help teach them how to use the item. As long as the player knows what frequency a character is on then they are able to contact that character in order to try and get more information from them. Typically information necessary for game play comes across the codec from other players, however occasionally the player will come face-to-face with other
characters who can also supply them with the information that they need. Most of the help that the game supplies to players comes through messages that are displayed as the game sees fit, however the player can attempt to track down information if they are unsure how to proceed (Sterling).

5.2.3 Applications

Help systems in software applications typically provide information and assistance to users in situations where they either do not understand a concept or do not know what is expected of them next. For this reason help systems are typically filled with a wealth of information regarding the entire application. Unfortunately help systems are often thought of in a very negative light and therefore users are reluctant to use them. Users are very aware of the necessity of a help system but they will often only reference it when it is obvious that they have no other choice. In many instances help systems prove to be very difficult to use because the user often does not even know what it is that they are looking for. Further, what the user thinks they are looking for may actually be listed under a different term because the developer referred to a specific topic in a different way. Either way, the user is usually already frustrated because they do not understand a certain aspect of the application, so by the time they resort to using the help they are already feeling very negative about their experience (Randall and Pederson).

Over the past decade help systems have evolved quite drastically and now make themselves quite visible to the user by encouraging interaction from the first time they open the application. Many of these help systems contain an “ethos”, which refers to the “character, the image, the presence of human-like or human-appealing attributes” (Randall and Pederson). Designers of applications today strive to create help systems that act more as a teacher or an expert than as a reference guide. These help systems are meant to start helping users from the moment that it realizes they may be in need of assistance. This approach is meant to help users feel more confident about what they are doing as well as minimize the frustration that users feel when they do not know what to do. New help systems can be as nonchalant as tool tips, the tips that appear when a mouse hovers over a button or menu item, or as advanced as Microsoft’s Office Assistant who can offer advice before the user even thinks they need it (Randall and Pederson).

In order to create systems that are both helpful and have human-like characteristics designers have created help systems that tend to be overly cute and too cartoon-like. While new users tend to appreciate the cute characters and often overly simplified help, advanced users find them to be annoying and oftentimes degrading to their intelligence. Whichever way users view these help systems, their reaction is one of ethos because they realize that it is an attempt to create a human-like representation of the very static data presented in the help documents. So even though these help systems are not accepted by all users, designers have managed to take a step further in creating a help system that will be able serve every user (Randall and Pederson).

The reason that these types of help systems tend to fail with some users is because in an ideal world the user would be able to understand the interface so well that help was never
required. So even though these help systems attempt to offer users an interactive medium with the data they are searching for, it still requires them to approach the system for help and further, to know what they are looking for. The idea behind ethos is that it will be provided on a level where the user does not need to ask a question because the application supplying it already knows what information is needed. In some cases Microsoft’s Assistant tries to accomplish this by trying to interpret what the user is doing and asking if they need help. For example when *Microsoft Word* thinks that the user is writing a letter the assistant will ask if they would like some help. This can be useful to the users who do not know how to write a proper letter, however for those that do it not only interrupts their train of thought, but it also questions their intelligence. Further, if the user is not even writing a letter when the assistant asks if they need help it decreases the users confidence that the help system is knowledgeable enough to help them solve their other problems (Randall and Pederson).

Help systems that embed the help text within the application tends to create an environment where it provides the user with the necessary information without making them ask for it. Screens that embed help text typically supply a list of choices with a paragraph of text, which provides the user with an explanation for what that selection will do. This provides the user with the help data in such a way that they have the choice of reading the text or not, depending upon their level of expertise. This method of providing the user with help data seamlessly integrates with the application so that the user does not even realize that they are being provided with help. It allows the user to learn how the interface works without forcing them to search for the required information and at the same time ensures that the system is not forcing information upon them that might not even be relevant in the first place (Randall and Pederson).

### 5.2.4 Case Studies

- **Microsoft Bob** In 1995 Microsoft released their operating system called “Bob”, which was a cartoon desktop that was meant to be extremely user-friendly. It came with eight applications to help users with their home organization such as a calendar and an address book. Despite its attempts to be easy to use and friendly it never took off with users and it is considered to be one of Microsoft’s biggest failures (Wikipedia.com: *Microsoft Bob*).

  ![Microsoft Bob Image](image)

  *Figure 10 – Microsoft Bob*

The idea behind *Microsoft Bob* was to create an operating system that looked and acted more like the real world so that novice users would feel more comfortable using it. The interface was based upon actual rooms so that it would remind users of their home. Even the concepts of the Windows operating system had changed so that instead of logins for different users, there were different “homes” to go into which the user could customize according to their own preferences. The guides, similar to the Office Assistant, could also be selected based upon the user’s...
preference about what type of character they wanted to lead them around the home, the most recognizable being the dog “Rover” (TechTV: Microsoft Bob).

Within each room there are items that when clicked will launch different applications, for example the calendar on the wall would launch the calendar application, while the notepad on the table would launch a writing application. These concepts were meant to help users more easily identify with the applications on the computer rather than expecting them to learn about icons and menus. One of the main reasons that it failed so badly was because it was too cute and essentially got in the way of what the user was trying to accomplish (TechTV: Microsoft Bob).

- **Office Assistants**  With the release of Office 97, Microsoft took the positive aspects of Microsoft Bob and created an animated help system often recognized as the Paperclip or “Clippy”. The Office Assistants are cartoon characters that the user is able to select based on their preference of the characters available. The characters that are available range from the commonly used Paperclip to “The Genius”, who is a cartoon representation of Albert Einstein. Giving the user the opportunity to select which assistant best suits them allows them to decide which character will best help them communicate with the application.

The Office Assistant has become a feature that users either love or hate, with the haters making up a much larger group. It is said that one of the most commonly asked questions of the Microsoft technical support is how to turn off the Office Assistant (Wikipedia.com: Microsoft Bob). Perhaps that is why with Office XP Microsoft chose to have the Office Assistant disabled by default, although they state that it is because the interface is so much more usable that users will no longer need the assistance. "Office XP is so easy to use that Clippy is no longer necessary, or useful," explained Lisa Gurry, a Microsoft product manager. Instead Microsoft is opting to implement more “smart tags”, or tool tips, and provide more embedded help throughout the application (Microsoft.com: Farewell Clippy).
**Windows XP: Rover** – With *Windows XP* Microsoft has finally achieved the proper amount of human-like characteristics for a help character without going so far that it becomes obnoxiously cute to the user. The search function in *Windows XP* allows the user to search their computer for various types of files or data with the assistance of “Rover” who has been given an updated look from his *Microsoft Bob* days. Rover provides a dialog box which has embedded information that is useful to users who are interested in the information while still being unobtrusive enough that it can be disregarded if the user wanted to.

The user cannot ask Rover questions, as with an Office Assistant, his only function is to provide the user with the information that they need. Rover is animated as he tries to help the user as well as when he is searching through the computer for the information that the user is looking for. In this sense it provides the user with the bit of human interaction that can be useful, while at the same time maintaining a level of interaction for users that are not interested in the character but still need the information that he supplies. It also does not assume that the user needs to ask questions or make the user feel as though the help system does not understand their needs. The information that Rover provides is still available even if the user turns off the animation, which is not an available feature of the Office Assistants. This version of Rover is another step closer towards created a personified help system that all users can relate to and are satisfied with using. It provides help without shoving it the user’s face, thereby making it interact almost seamlessly with the user interface.

**5.2.5 The Future**

Video games have been able to seamlessly integrate personified help characters into the game because to the player it makes sense to have other characters that they can speak with. The virtual world of a video game typically includes interaction with multiple characters, the adversaries and the accomplices, so to have characters exist on a strictly information basis flows well with the storyline. Despite Microsoft’s attempt at creating a world in which help characters made sense, i.e. *Microsoft Bob*, it failed because users were not looking for a world in which they would navigate through in order to accomplish their day-to-day tasks. Microsoft is getting closer to perfecting the personified help character by minimizing its use, which does raise the question of whether or not the character is even useful or if in reality by going in the opposite direction of *Microsoft Bob* is the personified help character being removed from software applications altogether?

In essence Microsoft has taken an environment, *Microsoft Bob*, which was quite comparable to a video game environment and removed almost all of the aspects that made it comparable in the first place. Early efforts at creating a personified help character in a software application clearly failed. This is because when a user is typing a
document there are no human-like characters that make sense to have pop up to offer help or suggestions because an instance like that would not happen in the real world. This is why personified help is so difficult to accomplish in software applications. Although personified characters like the Office Assistants do not completely make sense in the software application world, they do still have their purpose and really are quite useful to novice users. Perhaps what all of this has shown to designers is that personified help clearly works in video games because it is designed seamlessly into the environment. Embedding help information in software applications allows the designer to relay the information seamlessly as well, and with Windows XP’s Rover they have been able to add some character to the search engine while integrating it seamlessly into the application.

5.3 Customization

Customization is the means by which a user/player can adjust certain aspects of an application or video game in order to have it meet their specifications. Customization in applications usually takes the form of changing minor aspects of the application like where toolbars are placed or what color the screen is. In more complex uses of customization, however, users can change the appearance by selecting a different skin. The different skin usually will change at least the graphics of the application, but sometimes they can actually change the entire concept for the application. Video games offer much more in the way of customization by allowing users more control over the characters they play with and the environment that they are playing in. Customization is one component that can actually make an application or video game more interesting to the user/player as well as make it more intuitive to them by creating something that they can relate to easier.

5.3.1 Video Games

Customization has been an important aspect of video games for a long time. Early video games allowed players to select a name for the characters they were playing with. As time went on video games advanced to allowing users to customize the appearance of the characters and their skill sets. Today, video games allow customization of virtually every aspect of the game including everything from creating unique characters and names to creating a unique environment in which to play the game. Customization allows for two important things in a video game. First, it ensures that the video game is unique for each player and second, it creates the opportunity for the player to play the game again with a different character in order for the game to react differently to them and essentially create a new game.

Video game players want an experience that is unique to them. A unique experience allows players to feel more involved in the game, especially when the characters are either based upon themselves or people they know. This means that there is something that they can relate to in the game, which helps them get more involved in the storyline. Players also want an experience that is different from what their friends have for an experience. This makes for interesting conversation about how each player beat a
specific obstacle since they did not all beat it in the same way. It also means that each player will have gathered a different set of skills while playing the game and if they then play the game in a multiplayer setting each player will bring to the table their own set of skills.

Video games that follow a very strict storyline are only entertaining for the player to play once. As soon as they have beaten the game there is no reason for them to play again because they will already know how to beat every obstacle and they will also already know how the story ends. Video games, such as role-playing games, allow the users to create their own characters with different skills and attributes means that each character they create would approach each obstacle in a different way. For example a wizard may cast a spell to beat a specific obstacle where a warrior would use their fighting skills. This means that as long as the player creates new characters each time they start a new game then their game will be different.

5.3.2 Case Studies

- **Diablo II** – *Diablo II* was released in June of 2000 and is a third-person role-playing game with the ability of also being a multi-player game. The excitement for the release of this game was so high that over 1 million copies were sold before the game was even released (ign.com: *Diablo II*). Since the game is often played as a multi-player game, the ability to customize the characters is important not only so that each player has their own identity, but also so that it is easy to recognize each player based on their appearance. When the player starts a new game of *Diablo II* the first thing that they do is select what type of class they want their character to be, i.e. a warrior, wizard, druid, etc, and assign them a name. The class that is selected determines the sex of the character, for example if the player selects the Amazon character then the character is female whereas if they select the warrior then they will be male. This type of character selection allows the player to have some customization over what type of character they will be and to a certain extent how they will appear (Sterling).

As the game progresses there are two things that happen that further customizes the character. First, as the player acquires new equipment the character’s appearance will change based upon which items the player chooses to use. The second form of customization that occurs is that the player will have the option to select what skills they want their character to excel in. Allowing the user to select not only what items to carry and what skills to excel in ensures that not every player will have the exact same character, even if they select the same class. This is important because it gives the player a feeling of individuality while also ensuring that each player has their own purpose since their skill sets will be so different (Sterling).
• **TimeSplitters 2** – The premise for *TimeSplitters 2*, which was released in 2002, is that there are aliens who are plotting evil things and so the main characters follow them through time to try and stop them. Each stop through time is a different location that the main character must defeat to move on (*ign.com: TimeSplitters 2*). The player has the ability to select which model they wish to play with and give them a name. This will determine what the character will look like, but this is the only form of customization available for the characters (Sterling).

Instead this game has an interesting method for customizing the game for each player. Obviously the more that a person plays the game the better they will become and *TimeSplitters 2* is no exception. The interesting thing about this game is that the better the player becomes the more options they have. For example, the player cannot move on to some levels in the game until they have been able to defeat other levels. More specifically, the better a player does on a level, for example beating it in 3 minutes as opposed to 8 gives the player a more advanced assignment level which will also open up more levels. This not only encourages the player to practice the game more, but it gives them more incentive to beat the game by utilizing more skills. Another interesting thing that *TimeSplitters 2* offers is the ability to create your own level. This allows the player to create an environment that is meaningful to them and provides a different form of customization that is typically available to video games (Sterling).

• **Sims** – *Sims* is an interesting game that was released in 2000, whose plot has nothing to do with killing anything or needing to rescue anyone. Instead, this game centers on the player creating a number of characters to live together in a home. Success is a matter for the player to decide since they have the ability to try and make the characters as happy or sad as they choose. Virtually every aspect of this game is customizable by the player including the design of the characters and the home they live in. It is up to the player to decide how sociable a character is, what type of career they have and even how well they flirt. The player than helps guide the characters to find jobs and even love, however the characters can choose not to listen to the advice of the player based upon their set of personalities, should the advice go against their traits (Lopez).

This amount of customization is already significantly more than is available in other video games and yet the creator, Will Wright, was so intent on making something completely customizable that they have made it possible to add your own skins or download those that someone else has created. And if that still is not enough for some players, it is even possible to get in and adjust the programming language in order to
make the video game run the way that they want. “Wright wants to make sure that the world can be tweaked into any form you’d like” (Lopez). Wright’s favorite analogy is that if you wanted to, you could create the Addam’s Family next to the Brady Bunch just to see what happens. It’s all up to what the player chooses to create and is limited only by what their imagination can come up with (Lopez).

5.3.3 Applications

Customization allows users to change certain aspects of an application to meet their desires or requirements. Typical examples of this would include the ability to change the appearance or color palette used in the application, the ability to select which items appear in a toolbar, the ability change the keyboard layout and even the ability to change the content available. Many websites today will even allow users to create their own version of the site, typically dubbed “my” site name. These sites allow users to customize the data presented on the site in such a way that it is most useful to the user.

Since many sites now offer their own version of the “my” site, security and privacy are issues that have come to the forefront. Users must share personal information with the site in order for it to interact properly with them. These sites know everything from the user’s name and address, which is required to get things like weather and traffic reports, to many of their personal preferences regarding music and sports. In the past companies have shared a lot of this data either with other companies or other departments within their own company. This has proven to be a disastrous move for some companies because users feel that their privacy has been violated. Anytime that a user must share sensitive data with a company they need to feel safe and secure that the data they are supplying will not end up in someone else’s hands. Users who do not feel comfortable with a site simply will not share their information or will not visit again (Manber et al).

Usability becomes a huge concern when users can customize many of the options that are available. A balance between allowing them as many choices as possible while maintaining that all options they select do not end up creating a site that is unusable is a necessity for a successful personalized site. If a site allows a user to have too many choices they may end up selecting choices that were not meant to go together. Further, when a user feels like they have too many options they typically will not experiment with the application in order to learn how it works because they will be afraid of making a wrong choice (Manber et al).

Some sites can take customization too far by trying to provide users with information that the developers think they would want. For example when displaying top news stories some sites might be tempted to supply unique news to the user based upon information they have searched on in the past. While some users might find this useful, other may find it is supplying information that they are not really that interested in. Further many see it as an invasion of their privacy knowing that the application is keeping tabs on what information they are searching for. Additionally, when it comes to reading the news, most users are only interested in reading the new stories that everyone else is reading. Clearly it is possible to go too far when customizing applications, especially when the
system tries to determine what the user wants. As long as the user has the ability to make those choices for themselves then their chances of being happy with the outcome is much higher (Manber et al).

5.3.4 Case Studies

- **my.yahoo.com** – *Yahoo!* was the first site to offer the ability to customize their site in order to create the user’s own personal view of the data being provided. *My Yahoo!,* which was released in 1996, allowed users to personalize the data that was provided. Users could select everything from how the data is displayed to what actual data they are interested in seeing. The interface is customizable by selecting the color palette, the layout and the priority given to each item on the page. Although the options are limited, allowing users to determine how the page should look gives them the opportunity to create a page that they find most appealing, not necessarily what the designer felt was most appealing (Manber et al).

Users have the ability to select from multiple different news feeds available, get stock quotes for their chosen stock symbols, get the local weather, and their daily horoscope, among hundreds of other things. This gives users the opportunity to stay on top of whatever news worthy topics they are personally interested in, whether that is the sports news or the entertainment news. This ensures that the user will be interested in coming back to their personal site because they know it will contain information that is important to them (Manber et al).

- **Microsoft Office 2000** – The release of *Microsoft Office 2000* brought about some subtle, yet important changes to customization. Many of the items that can be customized remained the same, for example, users can still decide which toolbars to display and what buttons should be on them as well as decide where those toolbars should be placed. Users can also make slight changes to the layout of the screen, i.e. toolbars can go along the bottom, the sides or even be floating rather than just being tied down to the top of the application. Additionally, if the full toolbar does not have space to display than the items that appear on the toolbar adjust based upon which items are used the most (Microsoft.com: Microsoft Office 2000).

Perhaps the biggest change that was made to *Microsoft Office 2000* was that Microsoft tracks which menu items the user selects the most and then starts to “hide” other menu items that are used less frequently. The idea behind this was that novice users would not overwhelmed by the multitude of choices available in menu items and would be able to
easily find what they needed. By simply hiding the additional menu items Microsoft Office 2000 makes it possible for advanced users to get at additional items without too much difficulty by simply expanding the menu where the item exists. When a company tracks information regarding what users are doing, people start to get a little concerned about their privacy. This instance, however, seemed to be acceptable because it is not information that a user would consider sensitive and in the long run it makes the menus easier to navigate (Microsoft.com: Microsoft Office 2000).

- **Window Blinds** – Window Blinds is one of the most popular downloads available at C-net’s site, Download.com and is currently estimated to be running on 8 million machines worldwide. Window Blinds allow users to change the appearance of the user interface for Windows 98, ME, 2000 and XP through the use of a “skin”. A skin applies new graphics to a currently existing application, or in this case operating system, in order to give it a more personalized appearance. There are thousands of developers who have created their own skins for Window Blinds, so there are many different themes to select from ranging from a school black board to Nintendo’s Gamecube (WindowBlinds.net).

Window Blinds is popular because it has allowed developers to adapt the windows theme into a wider array of options. Although the premise of the design is still based upon windows, developers can add their own elements and icons that may symbolize other things to the users. For example, the Nintendo Gamecube skin made as many components as possible be based upon the Gamecube’s exterior console design in order to relate the operating system to the console. Window Blinds generates an unlimited number of options for what the operating system’s theme can be based upon; it is only limited by what the developers come up with. Stardock, the company that produced Window Blinds, allows developers to utilize a program called DirectSkin in order to ensure that the application will be able to work properly with Window Blinds, making it possible for every application on the computer to interact with the user’s chosen skin (WindowBlinds.net).

### 5.3.5 The Future

Customization is a pretty inherent concept when it comes to video games and there are few games that exist today that do not allow the user to customize at least one aspect, even if it is as simple as adding their name. Customization is an aspect of video games that players have come to expect, and they are more than willing to take the extra time to select the characters they want to play with and ensure that they have the appropriate attributes. Players are excited when they are presented with the opportunity to uniquely generate many aspects of the game, including creating a character and their environment. Customization in video games is what makes video games interesting and exciting for the players. It is a concept that video game designers are able to completely integrate into
their games without over or under using it. Video games have mastered the art of customization.

Customization in applications, on the other hand, is not nearly as sophisticated as video games and it is an option that does not exist as much. Some applications today are making a better effort to allow customization by giving users the option to change the skin so that the application has the appearance that the user wants. Creating different skins is something that a lot of users are interested in, so applications that allow different skins typically have plenty for the user to choose from. Applications that do allow different skins will have a very different appearance from one user’s machine to the next despite that the functionality will be basically the same. Functionality is an area that applications have not explored much either. Some applications will allow users to organize the layout in any way they choose, however many will not allow the user to easily customize the functionality. Applications that do allow for the functionality to be updated usually hide it deep within menus were users cannot easily find it. Applications have a long way to come to creating an environment that is as customizable as video games are, but some key applications are on their way.

5.4 Personification

Personification is the concept of adding aspects to an application or video game that directly personify the user or player. This could be something as simple as adding their name and perhaps a short bio to something as complex as adding their own face to a character. In applications personification typically deals with information regarding the user, like setting up a profile that includes their name and interests and occasionally includes adding a photograph. In video games, personification is something that is taken to a much deeper level by really placing the user into the game. The latest video games have the ability to actually place the player’s image into the game so that the character they are playing with will look like them. Other video games allow players to select characters that closely resemble themselves by selecting faces, hair color, and body type. Allowing users/players to determine what they look like or how they are perceived allows them to feel connected to the application or video game.

5.4.1 Video Games

The storyline of a video game is a key component that either makes a video game great or sees it fail. The most essential piece of the storyline is the main character. One of the most recognized characters of all times from video games is Mario whose games are all released on the Nintendo consoles. A survey in 1990 showed that Mario was “recognized more by American children than Mickey Mouse” (Poole). Part of what makes a good video game character has to do with how they move across the screen. For example, when the player moves the character forward by pressing buttons on the joystick, the character must move in such a way that the player feels like they are actually controlling each movement that the character makes. This draws the player into the game so that they come to care about the character they are playing and will do anything to avoid them getting hurt (Poole).
In 1997 the James Bond video game *Goldeneye*, which was based on the movie of the same title, literally took the faces of characters from the movie to place on characters within the game. This blend between real actors and video game characters brought reality even close to video games. This concept is what helped players relate better with the game because visually there was something that reminded them of the movie. While playing the game they could see themselves completing the same missions that 007 did in the actual movie. Although this is good in that it helps draw the player into the game more, it is also bad because it leaves little to the imagination. In some cases this type of morphing between movies and video games reminds the players too much of the movie and therefore they do not find themselves as absorbed as they could be (Poole).

Personification takes this concept a step further by actually placing characters into the game that closely resembles the player themselves. This allows the player to fully engage in the game as himself/herself, so no longer are they controlling a character, but they are controlling a representation of them-self. Clearly no player would want to bring harm to him or herself, so this concept ensures that the player cares even more about how they perform in the game. This concept is a double-edged sword because many players are interested in games because they can escape reality, which includes their own life, for a time being. If the character they are playing is actually based upon them then they are merely projecting themselves into a different world, but they are not escaping who they are. On the same note, however, many players who place themselves into a game feel a much deeper connection with the character they are playing and in that sense have a better experience. The success of this concept rests on the experience that the player is looking to have.

5.4.2 Case Studies

- **Tony Hawk Underground** — *Tony Hawk Underground* allows for extreme personification that is unprecedented in many other video games. The idea of the game is that the skateboarder selected is based upon the actual player. The player is able to take a no-name skateboarder and take them to the professional level within the game. The player selects everything from height and weight, to the clothes they wear and the skateboard they use. There are different faces available that the player can select in order to find one that closely matches their features (Wiley).

While many games today allow players to select from a group of different faces, it is unlikely that the player will find one that looks exactly like them. What this game offers that others do not is that the player can actually send the developers of *Tony Hawk Underground* a digital image of themselves and the developers will send back a code that can then be used to download their image. Once the image of their face has been downloaded into the game the player can select hair color, beard, and other facial features.
features. Players can also personalize their own tricks and skate parks in order to create environments that are completely their own (Wiley).

- **Sims Online** – *Sims Online*, released in December of 2002, has a drastically different concept than that of its predecessor *Sims*. *Sims Online* allows the player to create one character, or avatar, who is meant to be a virtual recreation of himself/herself. Players have the ability to choose many different characteristics about themselves, such as their face and clothes in order to closely represent themselves. Unlike *Sims*, however, the player does not select their personality scores as it is assumed that the player will be infusing their own personality traits with the character (Bates).

In many senses Sims Online is just a chat room with visual avatar personalities. In other ways the game mimics personalities in an oddly eerie way. For example, there are mini tasks that come up throughout the environment that players can choose to either participate in or not. Players also get popularity ratings based upon how often their house is visited and how many times people have given them a “friendship flower”. Players have the option of doing nothing or doing everything they can to succeed in the game. Similar to real life, however, the player will most likely act in a way that is natural to them. In this sense social people will most likely be more popular while the others will remain more secluded and will only be known by a fewer subset of the online community (Bates).

- **Black & White** – The objective of Black & White, which was released in March of 2001, is to get followers to believe in the player as a god. The player has the ability to cast miracles and has powers that allows them to perform certain tasks. What is interesting about this game is that the player has the option to play a good god or an evil god. If the player performs all actions in an evil way then the game takes on an evil tone, everything from the appearance of the world to the music. If the player performs all actions in a good way then the world appears good and heavenly music is played. It is possible for players to play the game neutrally by performing good and evil and therefore the game will not strew drastically to one side or the other. This game adjusts to personify the player based upon how they play the game. The game is constantly trying to interpret the player’s actions to determine if they were meant to be evil or meant to be good. Further, even if the player is occasionally bad the game will not interpret that as being an overall evil god, instead it looks at everything the player has done in order to determine which type of god they are (Blevins).
Additionally, the player has an avatar that they raise as the game progresses. Typically the avatar will mimic the actions of the player, so if the player is good then the avatar will be good, likewise if the player is bad then the avatar will be bad. The player also has the ability to train their avatar based on how they reward or punish them for tasks that they perform, so it is possible to have a good avatar while playing the game in an evil way and vice versa. The point of the avatar is that you can impose any set of values onto them that you choose, no matter how you play the game. The game itself, however, will also try to mold the avatar based upon the actions that the player takes, whether they are good or bad (Blevins).

5.4.3 Applications

Applications usually deal with personification by allowing users to add their name and create a bio that they can share with other users. Microsoft Outlook is a good example of an application that allows for personification, although it is on a very basic level. In Microsoft Outlook users can add their names, email addresses, calendar information, as well as store information about their friends and family. All of this information is personal and makes the applications specific to them because another user most likely would not be interested in someone else’s address book. Although this is a simple example it is one that has been perfected. Users have the flexibility to do virtually anything they may be interested in doing and it’s very simple for them to do so.

Applications that allow users to communicate directly with other users via the Internet are the ones that allow for the most amount of personification. Clearly this is an important piece to these applications because the users want to be able to feel as though they know the person they are conversing with, if they do not already know them. They also want to be able to get a feeling for what that person’s actual personality is like based upon information that they supply about himself/herself. These types of applications need to personify the person using them because that is the reason they exist.

It is difficult to say if personification would be useful in other situations since it does not really exist in other applications, however it is clear that anytime a user feels more connected with an application the more likely they are to feel comfortable using it. Most applications could apply some form of personification even if it were to simply be storing information regarding the user and their preferences. Beyond that though, it would definitely be on a case-by-case basis how useful it would actually be. For example, in an application like a calculator there is no reason to keep any personal information about the user, but in an application used to keep information about the user’s wedding plans, personal information is imperative.

Personification is a concept that has not been used much in applications, so there is not much data on which to determine its success or failure. Personification is successful when it is used to store information regarding the user, like Microsoft Outlook does, and it is also successful when it comes to personifying the user like an instant messenger application does. In video games, personification is used to engage the player by placing them directly into the action of the game. In applications there are rarely instances where
users are actually made part of the execution of tasks where they can visually see themselves doing so. Applications where users interact with one another are starting to take advantage of this concept, however there are potentially lots of applications that could take advantage of this concept by using it in different way.

5.4.4 Case Studies

- **AOL Instant Messenger 5.5** – AOL’s *Instant Messenger* allows friends and family to talk in real time over the Internet. *AOL Instant Messenger* includes the ability to set up a personal profile including likes and dislikes, select an icon that somehow represents the user, whether it be of an interest of theirs or an actual picture of himself/herself. Users can share images and files with one another through the application as well as talk with a larger group of people in a group chat. Users can also personalize other parts of the application, like the color and font that their message will show up using. Most recently, *AOL Instant Messenger* offers the ability to video chat with other users, allowing them to actually see each other.

- **Orkut** – *Orkut* is an online community that is quite unique because it will not allow new users to join unless someone else specifically invites them that is already part of the community. This makes it a community full of people where everyone is connected to everyone else through at least one person. New users create a profile that includes a picture of himself/herself along with a short bio of information. Users can add photos to an album, share notes with each other, view their friend’s profiles and even view the profiles of friends of friends. This community makes people feel more comfortable because the idea that your information is being shared with friends of your friends, and not merely strangers, is a comforting feeling.

- **CokeMusic.com** – CokeMusic.com allows users to join an online community that is dedicated to music. The site allows users to navigate through studios and to create their own music by using the mixers available on the site. The users can then share their music with other users as well as distribute their “CD” through the site. Users can go to different rooms and play their music for all of the other users to hear. Anytime that someone
else likes a user’s music they will give them points, which allows users to buy things like furniture and art for their own personal studio (CokeMusic.com: Help).

In order to access the account the user must great a “V-ego” which is an online version of himself/herself. The user is allowed to choose from a selection of faces, hair colors and clothes in order to create a character that they feel represents them. Users also create profiles about themselves to share with other users as well create their own rooms where other users can visit. This site is a unique way of allowing users to share music while also creating an environment to get to know each other. Users can be musicians, producers or fans, allowing them to decide their level of interaction on the site (CokeMusic.com: Help).

5.4.5 The Future

The future of personification for video games will probably include the ability to do a full body scan so that users can put all aspects of themselves into the video game instead of just their face. Currently choices tend to be limited for some games, so giving players more options would certainly be one way that video games could allow for more personification. Video games have done an excellent job at allowing users to personify the characters that they play with. Even allowing users to choose between different characters allows them a certain amount of personification because users may choose a character that resembles them or who even simply is the same sex as them. Video games that are successful are the ones where the player can feel a connection with the character they are playing, so a character that resembles them may allow this connection to be even deeper. Simply having characters that may resemble players is not a complete recipe for a successful game though; so all games that allow for personification will not necessarily be successful.

Personification within applications still has a lot of growing left to do since there are currently few examples of this. It is possible that users may be more productive working in a virtual environment that is more like a game where they have to control the user, i.e. themselves, to perform specific tasks. This may make mundane everyday tasks more interesting to users, however it may also detract from what they are trying to accomplish. There is certainly a lot more that can be done to encompass more personification into applications, however at what level it will be successful is still to be seen. Video games are already very successful at allowing players to personify the characters so that they are very similar to themselves or are an enhanced version of themselves that they are interested in portraying. Perhaps the only way for video games to enhance the concept would be to extend the amount of personification that is available and allow even more options and more selections so that the characters can become even closer versions of the player.
5.5 Sensory Feedback

Sensory feedback supplies people with feedback that helps them understand what their actions do. Sensory feedback is something that a person can touch, taste, smell, hear or see. For example, when a person steps on the gas in a car they expect the pedal to push back, when they put an orange in their mouth they expect to taste orange, when they put a flower in front of their nose they expect to smell its fragrance, when they slam a door they expect to hear a bang, and when they throw a ball they expect to see where it goes. All of these forms of feedback allow a person to know that the expected outcome actually occurred, and if it does not occur then the person will know that something went wrong. Besides sight and sound, the rest of the senses were for a long time largely ignored in video games and applications. Video games were the first to branch out and start to explore using touch as a form of feedback when they started using things like rumble packs. These proved to be extremely successful for video games, so in the past few years both video games and application have been exploring utilizing the other sense in order to provide users with a more realistic experience.

5.5.1 Video Games

Sensory feedback in video in games has always existed, even if it was as basic as displaying on screen what was happening, or playing sounds for the user to hear. These two forms of sensory feedback were all that existed for a long time for video games. In the 1990s force feedback in the form of "rumble packs" made their way onto the market. The rumble pack was an add-on for the joystick that would make the joystick vibrate based upon what was going on in the game. Originally rumble packs were limited in what they did, however these days rumble packs are a good source of feedback to understand what's going on in the game. Video games that have a lot of action, like racing games, specifically use the rumble packs to help the user know when certain things are happening, like they have been hit by another player or are driving on dirt.

These days, video games are starting to make good use of other senses, like sound. Although sound obviously exists in video games they are starting to get more advanced by utilizing the surround sound technology. Surround sound not only puts the user directly in the center of the action based upon the sound, but it also helps them identify things that may be going on around them. This adds another level to games that previously did not exist while also increasing the level of realism for specific games. Basic sounds in a video game are clearly meaningful and will help the user along with understanding what is happening, but surround sound adds more to the game by allowing the user to experience the game as if they were in it.

5.5.2 Case Studies

- Rainbow Six: Raven Shield – Raven Shield is a counter-terrorist game that utilizes surround sound in order to create a more realistic environment. The surround sound is of huge importance in this game because it allows the player to hear where their opponents are in the environment. This allows them to pinpoint exactly where they are standing in
order to shoot accurately or to be aware when someone may be trying to sneak up on them. It also helps players get a clearer picture of whether or not their shots were accurate or where they may have gone. It also informs them of where bullets may be coming from if someone else is shooting at them. Surround sound helps the game become more realistic to the player and adds another layer of complexity and interest to video games (Sterling).

- **Mario Kart** – *Mario Kart* and other Nintendo games are classic examples of games that utilize force feedback information. Originally the rumble packs were additional pieces that players could get for their joysticks, but these days it comes built in. *Mario Kart* will supply feedback to the player throughout multiple points in the game. For example if another player or an obstacle hits them, then the joystick will rumble. Likewise, if the player drives over items that they should not the joystick will also rumble. This feedback helps give users information when they are doing something wrong as well as if something has happened to them. This allows video games to encompass the tactile feedback that used to be lacking in video games and allows users to have a more solid gaming experience.

### 5.5.3 Applications

Applications have been very similar to video games in that their progression into other senses has been very minimal. Sight and sound are obviously two senses that are widely used within applications, however unlike video games, surround sound is not something that has been used much. Besides multimedia applications, there are few applications on a computer that would require the use of surround sound. In fact using surround sound to provide a beep to the user when they do something wrong would be useless information because the user is not meant to be placed in a specific environment where that type of error occurs, so it does not add any level of information that they did not already know.

Keyboards and mice are traditional forms of tactile feedback that help the user know that they are pressing keys or have pressed a mouse button. Beyond these two items, though, there are not many cases where another attachment has been used that supplies force feedback to the user. Smell and taste are obviously two areas that have been relatively ignored when it comes to applications. This is most likely due to the fact that many applications have no use for these types of feedback.

### 5.5.4 Case Studies

- **DigiScent iSmell** – *iSmell* is a USB device that hooks up to a computer and releases scents into the air based upon information that is sent to it. The device works by heating certain oils that together make up the requested scent and then blowing them into the air with fans. Signals are sent to the device based upon information that is linked into a file, so for example a picture of an orange may produce a scent of an orange peel, or a picture of fire would produce the scent of burning wood. This tool would also allow users to create new scents that they could share with their friends, or allow them to attach a scent
to a love letter. Scent is a largely ignored medium when it comes to computer technology and this is one solution to help bring it into the forefront (Platt).

- **Logitech iFeel Mouse** – Immersion Technology is the company responsible for the majority of products that support force feedback like joysticks, and that same technology has gone into creating the *iFeel* mouse produced by Logitech. The mouse provides pulses to the user when they perform tasks like rolling over buttons or menu items within *Windows* and links within a browser. For some users, especially novice users, this technology can be extremely helpful in letting the user know when they are hovering over something that is important. For more advanced users the mouse tends to be distracting because in a *Windows* environment they are constantly rolling over something that the mouse feels is important enough to supply feedback about. Although this mouse is not for everyone it does provide useful information to those users who are not completely familiar with *Windows*. Those that are familiar with *Windows* and browsers will most likely find the information useless (Logitech).

**5.5.5 The Future**

Smell and taste are the two senses that have been ignored the most for both applications and video games. Despite that there has been a product developed to allow users to smell different scents while working on their computer, the *iSmell*, it has not taken off with the majority of computer users, showing that they either are not ready to use this type of technology, or the technology is not what they want. Most likely users are not looking for something that can create a smell for them while working on their computers and so they simply are not purchasing the product. The *iFeel* mouse is another product that has not sold as much as the producers had hoped. Part of the reason that these two products are failing is probably in part due to the fact that users are not accustomed to them. Users also need the time to get acclimated to new products that are so drastically different from everything that they currently use.

Smell is probably a sense that is ignored just as much in video games as it is in applications. The failure of this type of product probably could have been predicted given the failure of movies that released scents into the ventilation system of movie theatres during specific parts of the film. For example if the movie showed a couple at a Chinese food restaurant then the audience would get a whiff of Chinese food through the vents in the movie theatre. These types of movies were not very popular and so their success was extremely short lived and was really just a novelty. Touch is a sense that a lot of video games have started to explore. Most joysticks these days include their own rumble pack and even games that use steering wheels have come up with their own way to make the joystick rumble to signify different things to the player. Video games will probably come up with more advanced ways of portraying touch in the future; however what they currently have is satisfying most players.
Applications have had varying degrees of success when it comes to using interface concepts that were popularized by video games. As times has gone on and video games have gotten more complex, applications have been able to draw from many different concepts that have been successful in video games. Video games have become the testing ground for new concepts and once they have been solidified, applications are able to take advantage of those concepts that have proven themselves time and again in video games. Clearly Personified Help and Customization are two concepts that work extremely well in video games and therefore application developers have been able to use those as examples for how to integrate them into their application. Although they have not always been successful, as was the case with Microsoft Bob, developers have learned from their mistakes and now are able to use concepts from video games in a more clearly defined manner. Personification and Sensory Feedback are relatively new concepts that therefore do not always work in new applications because they have not been completely refined within video games. As these new concepts become more integrated into video games, application developers will be able to determine how they would be best used within their application. Many application developers have come to realize that video game concepts are exactly what are needed in order to create applications that users are more interested in using. Within the next few years even more application developers will come to realize that video game concepts make applications easier to use and allow more options, so they will attempt to integrate even more concepts. As technology progresses and video games become even more complex, applications will have more to draw from and therefore they will end up becoming more like video games with every new release.
APPENDIX A

Personal Computer Timeline

1974 – Scelbi

The Scelbi (SCientific, ELectronic and Biological) was the first personal computer kit available and came with 1K of programmable memory for $565. For an additional $2760 a user could purchase another 15K (Bellis).

1974 – Mark 8

Jonathan Titus designed the second personal computer that was available, the Mark 8. The magazine Radio Electronics provided instructions on how to build your own version of the Mark 8, which readers were ecstatic about (Bellis).

1975 – MITS Altair 8800

MITS (Micro Instrumentation Telemetry Systems) was a company based out of Albuquerque, New Mexico that had been in the calculator business until Texas Instrument overtook the market. When the first two personal computer kits were released the owner Ed Roberts, along with friend Les Solomon, decided to try to create their own kit. The Altair was shipped with an Intel 8080 CPU and 256K of RAM for $400. Due to its reasonable cost and easily expandable parts, the Altair was a huge hit. The Altair was the kit that really got the home computer revolution started (Bellis).

1975 – IBM 5100

IBM released their own version of the microcomputer in 1975. The computer came fully assembled with BASIC, 16K of RAM, a tape storage unit, and a built in 5 inch screen. The IBM sold well to the small business and educational sectors, however due to its $10,000 price tag none of the hobbyists buying the Altair could afford to purchase the 5100 (Bellis).

1976 – Apple I

The Apple I was the first affordable computer released that came fully assembled for $666.66. The computer came with 8kb of RAM, a keyboard, a visual interface, and a 6502 processor (Bellis).

1977 – Apple II

The Apple II was released in early 1977 for $1298 was based on the Apple I and included color graphics, an audio cassette drive for storage of data, and originally came with 4kb of RAM, which was later increased to 48kb (Bellis).
1977 – Commodore PET

The Commodore PET (Personal Electronic Transistor) was based on the same 6502 processor as the Apple II and sold for almost half the price, $795. It also included 4kb of RAM, a keyboard, a video display, and an audiocassette drive for data storage (Bellis).

1977 – Radio Shack TRS-80

Also in 1977, Radio Shack released the TRS-80, which was based on the Zilog Z80 processor and included 4kb of RAM and 4kb of ROM. The TRS-80 also came with an audiocassette drive for data storage. Due to many of the problems with the TRS-80 it was often referred to as the “Trash-80” and therefore despite strong early sales, did not fare as well as other machines (Bellis).

1981 – IBM PC

The IBM PC (Personal Computer) was available for $1565 and included 4.77 MHz Intel 8088 processor, 16kb or memory which was expandable to 256kb, a 160kb floppy drive, and a color monitor. What made the PC different from other computers of the time was that it was fully expandable using open architecture and it came with the new operating system created by Microsoft (MS-DOS 1.0). Time Magazine actually named the PC “man of the year” (Bellis).

1983 – Apple Lisa

The Lisa was the first personal computer released to feature a GUI (Graphical User Interface). Xerox Corporation’s PARC (Palo Alto Research Center) had developed a GUI in the 1970s, but it was not until the Apple Lisa that it became widely popular. Due to the price tag on the Lisa, Apple was never able to sell more than 10,000 machines. Even when the Lisa 2 was released, also called the Macintosh XL, sales never reached what had been projected (Bellis).
Video Game Console Timeline

1972 – Magnavox Odyssey

The Magnavox Odyssey is considered the first video game console that was commercially sold to homes. The game had no memory, so players had to keep score on paper and due to the primitive graphics the console included an overlay that was placed on the TV screen to make the game more entertaining (CNN.com).

1975 – Atari Pong

Different versions of the Atari Pong game were released between 1975 and 1977 that included advanced technology such as color and on screen scoring. The Atari console included two built-in controllers and could only play the game Pong (CNN.com).

1977 – Atari VCS/2600

The Atari VCS (Video Computer System), also called the 2600, was Atari’s 8-bit answer to allowing different video games to be played through the same console. The 2600 was able to maintain its marketability since Atari continued to release new games for the system. This console was on the market until 1990, giving the 2600 the longest market time in video console history (CNN.com).

1978 – Magnavox Odyssey 2

The Magnavox Odyssey 2 was Atari’s closest competitor in the 8-bit sector, but was still never able to gain the momentum it needed to overtake their market. The Odyssey 2 was the only console released at the time that included a built-in keyboard (CNN.com).

1980 – Mattel Intellivision

The Intellivision soon became Atari’s closest competitor as they marketed themselves as a more sophisticated machine. The Intellivision was the first 16-bit console, so its graphics were far better than Atari’s. This console was also the precursor to the keypad controllers that were released with game consoles in the 1990s, although they were not a popular feature of the Intellivision (CNN.com).

1982 – Coleco Colecovision

The Colecovision was a very unique machine at the time it was released because it was as powerful as a computer, so for this reason Coleco had plans to release an add-on that would actually turn the console into a home computer. These plans were later dropped due to the video game crash of the 1980s. This console was sold with the popular Donkey Kong game, whose plumber character would later become Nintendo’s ever popular “Mario” (CNN.com).
1982 – Atari 5200

Atari’s 5200 was their attempt to regain much of the market they had lost to more technologically sophisticated systems that had been released since the 2600. The 5200 was as powerful as the Atari 400, Atari’s personal computer that included 16K of RAM. Unfortunately the console contained many problems, especially not being backwards compatible with the 2600 game cartridges (CNN.com).

1985 – Nintendo NES

The NES was an 8-bit system with 52 color capabilities and 2K of RAM, clearly not as technologically advanced as other systems on the market and yet it became hugely popular. The console quickly became the best selling console since the Atari 2600. Its third installment of the Mario game Super Mario Bros. 3, released in 1988, sold over 7 million copies (CNN.com).

1989 – NEC TurboGrafix – 16

This console featured an 8-bit CPU, a 16-bit graphics chip, and was the first console to feature a CD player attachment. Despite its advancements, the console was overshadowed by the Sega Genesis, which was released later the same year, due to fewer game selections and a less powerful 8-bit CPU (CNN.com).

1989 – Sega Genesis

The Sega Genesis, first released in Japan as the MegaDrive in 1988 was the first console to compete with the Nintendo. The Genesis featured a 16-bit CPU, which some game developers felt was so advanced that they left Nintendo for the opportunity to develop more advanced games. Sonic the Hedgehog was released in 1991 and was a huge success due to the ability to utilize the speed of the 16-bit processor (CNN.com).

1991 – Nintendo Super NES

The Super NES was Nintendo’s answer to the 16-bit game console. Although it initially lagged behind the Sega Genesis, it was eventually able to recapture its popularity. For five years the Sega Genesis and Nintendo Super NES shared the top spot in the video game market and were able to provide gamers with a wide array of game selections (CNN.com).

1995 – Sega Saturn

Initially the Sega Saturn sold well amongst the gaming community due to its 32-bit processor, CD player, 3D graphics and digital sound. According to Sega the Saturn sold 100,000 consoles in the first 3 days on the market. The console was eventually surpassed by the release of Sony’s Playstaton later that same year (CNN.com).
1995 – Sony Playstation

The Sony Playstation was initially an attachment for the Super NES to play CDs, however when Sony and Nintendo could not agree on how to market the attachment, Sony decided to develop the system into its own console. Sony released the 32-bit console in September of 1995 and immediately took off in popularity. As of May 2000 one in every four homes in the United States owned a Playstation (CNN.com).

1996 – Nintendo N64

Nintendo chose to bypass the 32-bit era and went right into building a 64-bit console. The N64 was able to provide gamers the 3D look that they had come to love while enhancing the games that the classic games they were already drawn to. Silicon Graphics, the company responsible for bringing tornadoes and dinosaurs to life in the movies, created the imagery in the N64 games (CNN.com).

1999 – Sega Dreamcast

The Dreamcast stepped past the Nintendo N64 by creating a 128-bit architecture that greatly surpassed all of the consoles on the market by featuring technology only found in computers at that time such as a 200 MHz processor, a 64 channel audio chip and 26 MegaBytes of RAM. The Dreamcast also included a 56K modem to allow players the ability to play their games online. All of this technology made the graphics of the games smoother and more detailed than any other system on the market (CNN.com).

2000 – Sony Playstation 2

The Sony Playstation 2 includes faster playing speeds, higher quality graphics, while also allowing for the viewing of DVD movies through the console. Additionally, the console is also available with a network adapter to allow gamers to play over the Internet (CNN.com).

2001 – Microsoft Xbox

The Xbox was Mircosoft’s first step into the gaming arena. The Xbox comes fully loaded with a “stripped down version of the Windows operating system, a 700 MHz processor, an 8 gig hard drive, 64 MB of memory, and Ethernet port and a DVD drive” (CNN.com).

2001 – Nintendo Gamecube

The Gamecube is Nintendo’s step into the 128-bit arena, including a 405 MHz processor. This video game consoles accepts only 3-inch CDs in order to try and reduce piracy, so it does not include the DVD playback like the Sony Playstation 2. Nintendo has always had strong game titles, and the Gamecube is no exception with new releases of Pokemon, Mario and Zelda (CNN.com).
## Application User Interface Timeline

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>Xerox PARC</td>
<td>Xerox was the first company to generate the modern components of today’s GUI at their PARC (Palo Alto Research Center) (Galitz).</td>
</tr>
<tr>
<td>1974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1975</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1976</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1977</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1978</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1979</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>Xerox STAR</td>
<td>The Xerox STAR introduced the concepts of the mouse with pointing and clicking (Galitz).</td>
</tr>
<tr>
<td>1982</td>
<td>Apple Lisa</td>
<td>The Apple Lisa featured drop down lists and menus (Galitz).</td>
</tr>
<tr>
<td>1983</td>
<td>Apple Macintosh</td>
<td>The Macintosh was the first widely released personal computer that included a GUI (Galitz).</td>
</tr>
<tr>
<td>1984</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1985</td>
<td>Microsoft Windows 1.0</td>
<td>Microsoft releases their version of the windows style GUI (Galitz).</td>
</tr>
<tr>
<td>1986</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1987</td>
<td>IBM Presentation Manager</td>
<td>IBM’s Presentation Manager was meant to be a replacement for DOS style operating systems (Galitz).</td>
</tr>
<tr>
<td>1988</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>NeXT NeXTStep</td>
<td>NeXTStep was the first to produce three-dimensional graphics (Galitz).</td>
</tr>
<tr>
<td>1989</td>
<td>UNIX GUIs</td>
<td>UNIX releases GUI versions of the open source code (Galitz).</td>
</tr>
<tr>
<td>1990</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1991</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td>Microsoft 3.1</td>
<td></td>
</tr>
<tr>
<td>1993</td>
<td>Microsoft NT</td>
<td></td>
</tr>
<tr>
<td>1994</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>Microsoft 95</td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>IBM OS/2</td>
<td>IBM releases their OS/2 operating system (Galitz).</td>
</tr>
<tr>
<td>Year</td>
<td>Event</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>1997</td>
<td>Apple Mac OS 8</td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>Microsoft 98</td>
<td></td>
</tr>
</tbody>
</table>
| 1999 | Apple Mac OS X  
Apple releases OS X, which is based on the UNIX operating system (Galitz). |
| 2000 | Microsoft Windows 2000 and ME |
| 2001 | Microsoft Windows XP |
| 2002 | |

Microsoft NT 4.0.  
Microsoft releases the upgrade to NT with version 4.0 (Galitz).
REFERENCES


<http://pc.ign.com/articles/131/131754p1.html>


<http://www.wired.com/news/culture/0.1284.45146.00.html>


<http://www.wired.com/wired/archive/7.11/digiscent.html>

<http://pc.ign.com/articles/378/378268p1.html>


Images

Figure 4 1970s Screen
<http://www.pong-story.com/atpong2.htm>

Figure 5 1980s Screen

Figure 6 1990s Screen
<http://www.gamespot.com/n64/sports/maddenfootball64/screens.html?page=4>

Figure 7 The Legend of Zelda: Ocarina of Time
<http://media.ign64.ign.com/media/000/000437/img_1194554.html>

Figure 8 Master of Orion III
<http://media.pc.ign.com/media/014/014808/img_1663832.html>

Figure 9 Metal Gear Solid 2: Sons of Liberty
<http://media.ps2.ign.com/media/014/014538/img_1384764.html>

Figure 10 Microsoft Bob
<http://www.cs.unm.edu/~dlchao/flake/doom/chi/bob.jpg>

Figure 11 Office Assistants
<http://www.csun.edu/itr/guides/excel/2000/images/beg06-fimg01.gif>

Figure 12 Microsoft XP: Rover
<http://www.dickinson.edu/resnet/xp-search4.jpg>

Figure 13 Diablo II
<http://media.pc.ign.com/media/010/010629/img_1347667.html>

Figure 14 TimeSplitters 2
<http://media.cube.ign.com/media/479/479934/img_1573367.html>

Figure 15 Sims
<http://media.pc.ign.com/media/011/011754/img_1314974.html>

Figure 16 my.yahoo.com

Figure 17 Microsoft Office 2000
<http://www.robinupton.com/computer/software/microsoft/word/view_toolbars.gif>

Figure 18 Window Blinds
Figure 19 Tony Hawk Underground
<http://media.cube.ign.com/media/545/545801/img_1776252.html>

Figure 20 Sims Online
<http://media.pc.ign.com/media/015/015970/img_1548945.html>

Figure 21 Black & White
<http://media.pc.ign.com/media/003/003897/img_1273413.html>

Figure 22 AOL Instant Messenger
<http://www.jasonjay.com/papers/telementoring_files/image006.jpg>