Retro Game Programming

How this book got started?

This idea for this book began after I purchased a book with a similar title in order to learn how to build a game on the original Apple or C64 systems. I wanted to like the book given how touching and personal I thought the Acknowledgements were. But, I just couldn't. I did not find the book "meaty" enough for professionals nor did it contain enough material of use for the types of folks who would want to really use it – beginners. There was not enough information to teach you how to really program or how to build a game on any of the retro systems the book referenced. Moreover, the reality is that you and I will have a hard time getting our hands on the computer systems highlighted in chapter one of the book – TRS-80, Atari 400/800, Commodore 64, or Apple II, unless we are willing to check out e-bay and wait for a great deal – oh yeah and have lots of money to buy those old computers and the games. What made more sense to me was to discuss how we can experience these computer systems today using software applications that are freely available online. These software applications are called emulators and they not only allow you to create your own programs as if you had the real machine in front of you but these same emulators can be used to play the games of yore as well. In addition, there are emulators available for free. Today by searching the Internet you can find the tools to play games from your favorite home computers and game consoles – Atari 2600, Nintendo Entertainment System (NES), Super NES, etc.

This book is my attempt to provide you a version of the book I thought I purchased and make it available to everyone on the Internet. In fact, I couldn't recommend the book when I read it years ago but the book now can serve as great overview on topics I cover in detail in this book. So go ahead and buy it but keep this book nearby so you can get the details, exercises and game code that will surely make you the retro game programmer you want to be.

Note: (4/6/2014)
These notes have not been updated since 2009. I am in the process of returning to these notes given the fact that there is an interest to learn how to write programs for old systems such as the Atari 2600, Apple IIe, C64 and NES systems. All these systems used a common/similar microprocessor the 6502 so be prepared to learn how to write for this former powerhouse of a processor!

My plan is to have updates every three to four weeks. The date on the heading and version number should let you know if this is an updated version. If you are not sure visit brainycode.com and check the retro-programming link on the home page.

Don't hesitate to send mail with corrections and ideas!

N Y G U E R R I L L A G I R L "AT" B R A I N Y C O D E "DOT" C O M

Introduction

Figure 1 - Donkey Kong

What is a retro game?

A game is considered a retro game if it was developed during the rise of the video game industry which for us covers the period from 1972 to the early 1980s. It is open to debate about what constitutes a retro game, but for our purposes we will consider games as "retro games" that were developed and played on video, arcade and computer game systems such as the Atari 2600, the Nintendo Entertainment System (NES), and early computer systems such as the Apple IIe, Commodore 64.

The games from this time period were easier to play and develop than today's console games for the Xbox 360 and Playstation 3. Many of these games were developed by one programmer or a small team and unlike today's typical game most did not require millions of dollars to produce and promote. Budding programmers from their bedroom or garages developed games using their computer systems. You have probably played retro games like Pong, Pac-Man, Space Invaders, Donkey Kong, and Super Mario Bros. The games that we revisit as retro games were all fun, and easy to get started playing. Let me not kid you, there were also many awful games made on these platforms and game systems as well, but we are talking about are games that were notable, innovative and fun to play.

It is easy to see the elements of the game that makes them fun to play. You have the villain (see Donkey Kong taking our damsel in distress[1] away to the next level in Figure 1), a boss (an enemy that is very difficult to kill), a good guy (representing you the player), and your best friends as opponents or comrades in arms, monsters, burning tires, aliens, and missiles to dodge or destroy. The story lines and graphics are relatively

---

[1] In many retro games the formula of hero saving damsel comes up over and over again. We don't attempt to address the social and political implications of these formulas, the only issue of us is – is the game fun.

simple compared to today's 3-D realistic games but the games are just as much fun to play.

The final testimony to the long lasting playability of retro games is how many have been converted to play on current console systems, such as the Xbox 360 and Wii, or packaged into anthology game packages with the top games from game companies such as Activision, Atari, Konami, and Nintendo.

Figure 2 - The 6502 microprocessor

What are we trying to do?

My goal with this book is to discuss video game history, teach you how to develop games by learning the computer language used by programmers to develop games of yesteryear.

We will play games, study the elements that make them fun for the player and create clones of popular games for our favorite game systems. The machines we will target are Atari 2600, the Apple IIe, and the NES. We will develop real games that will play on these systems.

The key feature each of these machines share is they are all based on the same computer chip to make spaceships move, missiles fire and keep score – the 6502 microprocessor (see Figure 2). You will be learning details about these machines and the microprocessor that powered them all. For all these machines, the effort involved to get a spaceship across the screen is different from the concerns and details a programmer creating today's games faces. But, the core elements - the logic, the step-by-step tracking of information, the representation of objects that move, blow up, and spin, the requirement that the programmer take these ideas and covert them into computer instructions – is still the same. So once you have completed this book, I recommend you try the second book in our series – learning to write 2D Platformer game engine using Java.

What do you need?

You will need a computer and an Internet connection. I am using three types of computers to test all the software out, a machine running Windows Vista, Windows 7 and

Windows 8. The basic theme here is that you will need a computer with the Windows operating system from running Windows Vista[2] or higher. All the software we use in this book will run on any one of these operating systems and probably any version in between.

You will not need to buy or acquire an Atari 2600, Apple IIe or NES machine. We will be using a software application called an emulator. "An emulator duplicates the functions of one system using a different system, so that the second system behaves like (and appears to be) the first system." What that translates to is that we will be obtaining software from the Internet to install on your PC that shall be software versions of the hardware machines of Atari 2600, Apple IIe and NES. For example, we will use the AppleWin program to emulate an Apple IIe. We will also obtain software development tools from the Internet that will be used to convert our programs to machine code and a file, disk or cartridge format that each of these emulators will be able to execute.

All the tools and applications you will need are available free online. All you need is an Internet connection to obtain it. The website, www.brainycode.com will have all the tools you will need to complete all the exercises and programs in this book.

Figure 3 - The game Hard Hat Mack (first EA game) running on AppleWin emulator

---

[2] The fact of the matter you should be able to run all the code using Windows XP and maybe Windows 98 but I will only verify the operating systems mentioned in the text.

What should you know?

I am writing this with a young person in mind as the reader; hence the language and explanation will be informal, detailed and down to earth. This is in no way intended to turn off any one who is not just out of middle school or as old as I am. I am writing this for two people – me and you. I wanted to consolidate in one place all the information that is out there in the Internet but not in a form that makes it easy to find, understand or learn from on your own unless you had years of programming under your belt. I have over twenty years programming experience and still found I scratched my head on occasions or uttered the word, "Huh?" more than once when trying to figure out how to use material I found on the Internet. I hope this effort proves useful.

I am going to assume that you

- Can download programs and files from the Internet using a browser
- Know how to install programs
- Can follow instructions
- Feel comfortable with a little math
- Know a high-level programming language

I don't know what characteristics make a great game designer or programmer. I do know that the people that I admire and love to work with are curious, love to learn new things, like to read, have a good sense of humor, and have the discipline to complete things. If you too share these attributes than you should be able to do quite well.

What's the plan?
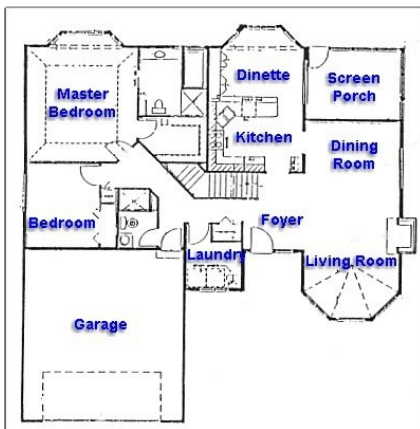
If you play the games we suggest, read, do the exercises, follow along and ask questions[3] you will be end up with the ability to program, specifically the 6502 chip and understand how to build games similar to Pong, Crisis Mountain and many more. In addition, you will be ready to learn other programming languages and build more advanced games.

Part I: Chapters 1-2. [50]
We present a short history of the games on video systems, since you really can't appreciate what goes on today in the building and playing of games unless you really know the past. We then show you how to obtain, install and use emulators for the Atari 2600, Apple IIe, NES and MAME for classic arcade games.

Part II: Chapter 3-4. [150]

---

[3] You can post questions directly to the author or join the forums (TBD) at the web site www. brainycode.com.

A look at what computers, programs are and what programmers do for a living. Chapter 4 presents the programming language we will be using for our game programming. Almost all of the games of old were built using assembly language…which if you don't know programming yet you can't appreciate it (or hate!) yet. The game systems we are using in this book were all based on the same microprocessor – the 6502 chip. In this part of the book you will become a programmer.

Part III: Chapters 5-8. [50]
We will examine the components of video games – the display, the input devices, the game play physics (e.g. balls bouncing), artificial intelligence and sound effects. The concepts presented here will be general enough to apply to even today's games. We will apply and learn the specifics for each of these topics (video and sound) for each game system we tackle.

Part IV: Chapters 9-11. [75]
This section of the book will be all about programming the Atari 2600. We present all the concepts, and build the game Pong step-by-step. We then give you the reader an assignment – you build the game Atari Tank. We provide ideas for graphics and game play but the final game is left to you.[4]

Part V: Chapters 12-14. [75]
This section of the book will be all about programming the Apple IIe computer. We present all the concepts, and build my favorite game Crisis Mountain step-by-step. We then give you the reader an assignment – you build the game Apple Robot Wars.

Part VI: Chapters 15-17. [75]
This section of the book will be all about programming the NES computer. We present all the concepts, and build my most frustrating game Abadox step-by-step. We then give you the reader an assignment – you build a Mario Bros-like game – Sam's World.

Part VII: Chapter 18. [5]
This section is a peek on what it will take to convert one of your games to run on a modern game console like the Xbox 360.

As a bonus, at the online web site www.brainycode.com we have other programming projects and assignments that you can use to learn how to build text-based adventure games, flight simulators, and more. You are also invited to share your retro games and ideas with me.

---

[4] A working version of all game assignments shall be available online with more detailed solutions on how it was built…but the fun is trying to build it your own version!!

Chapter 1: The Early History of Video Games

I am not going to pretend that I can write a complete or comprehensive review of the video game industry in one chapter. There are many comprehensive books listed in the bibliography on the history of the video game industry that do a wonderful job. I have an asterisk next to every book I recommend you read. Some books are great when it comes to coverage of game systems; others books are more directed on the topic of the games themselves. We will use the term video game to refer to games developed for coin-op arcade machines, home video game consoles and home computer systems.

What I hope to do a little bit differently from the books in the bibliography is to spend a little time on noting the games that presented something new and innovative to the gaming world. Many of "first" games qualify -- the first paddle game, the first game with space ships, the first games where you played against the computer, your best friend, the first shooter, the first text-adventure, the first fighting game, strategy, simulation, etc. When you scan the landscape of games developed during this early period you find so many rich and novel ideas were used to create games. When you compare the range of games of yesteryear to today's game universe where the only differences between the latest first-person shooter may be the graphics and the use of some new fangled graphic technology but today rarely do you go "Wow!" or "Gee, what was that designer thinking!" as often as gamers did in the early years of video game development.

My hope in playing and analyzing these retro-games of old that you get inspired to think of a new game idea that make your game "the must have" game of today.

Let's start to have fun.

Just Having Fun

A germ of an idea

In 1951 the idea to integrate video games with television sets first occurred to Ralph Baer while working for Loral while trying "build the best TV set in the world!" Mr. Baer will enter our historical overview later when he actually gets a chance to implement his idea ….15 years after he conceived it! But, I should note that he got his idea for a video game console at time when only 10% of households had television sets and the show "I Love Lucy" just started on TV later that very year.

In 1952, A.S. Douglas created a program as part of his doctoral thesis on "human-computer interaction". The game he developed was a version of tic-tac-toe called Noughts and Crosses for the EDVAC Vacuum Tube Computer.
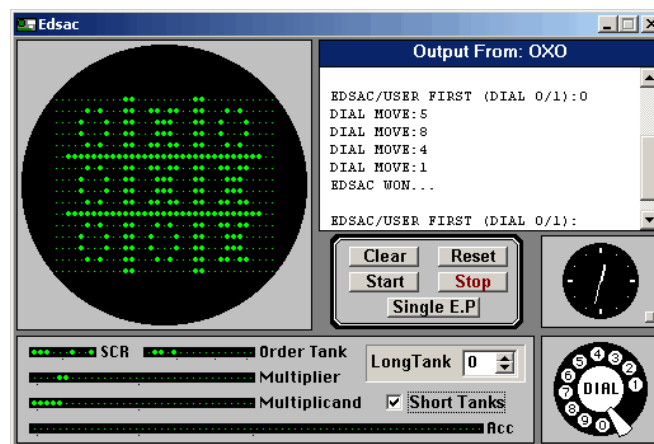
Figure 4 - The EDVAC Computer screen

This program demonstrates a repeating theme in gaming history - given a computer (regardless of cost) there is always someone who wants to write a program on it to play a game.

The First Pong Game
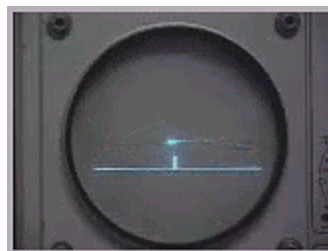
Figure 5 - Tennis for two

The first video game was not built from software programs but was built similarly to the first generation arcade machines with electronic circuits. The physicist, Willy Higinbotham wanted to create something for the annual open house at Brookhaven National Laboratory. The open house was intended to show the public how well spent their taxes were on projects that went into nuclear research, but the displays were usually boring and beyond the comprehension and appreciation of the general public. In 1958, he used a small analog computer to graph and display the trajectory of a pixel (see Figure 5) that appeared as a small moving ball that went over a vertical line that served as the net.

Figure 6 - Willy Hinginbotham

The display device was an oscilloscope and the input device was a controller with dials to change the angle of the "ball." Our first joystick!

So here we have our first video game, the game play was simple, the users were challenged to curve the ball over the net (no computer opponent) and keeping score was a matter of having users use their memories. The game was called *Tennis for Two*. You can see a video demonstration of the game and more details on this construction at this web-site: http://www.bnl.gov/bnlweb/history/higinbotham.asp.

The key thing to note about this version of Pong was that unlike the popular versions of Pong released much later this version used the real physical properties of gravity to move the ball across the screen and the controller was used to apply force and angle to the ball.

Spacewar!

The next notable event to occur related to video games happened at MIT. In the early 1960's computers were very expensive and used primarily by the Department of Defense or big companies for rather specialized purposes. In 1960, the company Digital Equipment Corporation (DEC) produced the PDP-1 (Programmed Data Processor-1). The machine cost around $120,000 which at the time was a relative bargain compared to an IBM 7090 with a price tag of $2,900,000. As a result of the price, the PDP series of machines started to appear on a few campuses around the country. At MIT a special group of men – part of the Tech Model Railroad Club – decided to create a game for the new machine. These fellows were highly inspired by the writing of the science fiction writer E.E. "Doc" Smith.

"..way back to the 1920s. It was on a hot summer night in Washington D.C. that Edward Elmers Smith (better known as E.E. or Doc, Smith), his wife, and a few close friends were marinating in the slow heat of

the Smith apartment. During the course of idle conversation, Smith mentioned how wonderful it would be if they were in the absolute zero temperature of space. Needless to say, it was a fascinating topic, and the group spent the rest of the evening engaged in a conversation dominated by fantasies of outer space. Finally, Mrs. Garby convinced Mr. Smith that he should write a book about adventures in outer space. Even though he initially refused the idea saying, "Got to have a love story to write a book, and I don't see how a love story would fit in with that kinds of stuff," nevertheless, she was able to convince him to change his mind by agreeing to write the "love stuff" herself if he would write the "wild stuff". The rest is history. "The Skylark of Space," "Gray Lensman," and all the other works of "Doc" Smith became seeds that would take some 40 years to grow."[5]

Smith was considered the grandfather of the science fiction genre. He wrote the Skylark and Lensman series which was the first modern Space Opera. The members of the Tech Model Railroad Club used the sci-fi stories as inspiration by creating a game taking place in space.[6]

In 1962 the main task to write the game that eventually came to be called Spacewar! was given to Steve "Slug" Russell. They developed the idea to "pit two spaceships (one called Needle the other Wedge) with limited fuel supplies against each other in a missile duel around a star. The other members added more features to the game – a realistic star field background, accurate gravitational effects from the sun that lies in the middle of the screen, a hyperspace feature that provides the capability for the player to press a button and make his spacecraft disappear and re-appear at a random location on the screen.

The game was notable for several reasons:

- The hacking ethos,
    - Creating software and sharing it with each other
        - the game did not belong to one person but was the effort of a team of students
    - Placing a high value on freedom of inquiry
        - the members did hesitate to explore new ideas – imagine no one ever did anything like this before!
    - Upholding the right to fork or take a copy of the current program and add to it
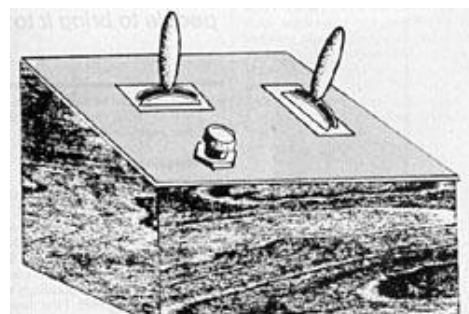
[5] Carey, Earl J., Retro Game Programming, Unleashed for the Masses. Premier Press.
[6] I don't think it is a coincidence that many budding programmers and game developers are into science fiction!

- ▪ the members did not hesitate to add new features to the game
  - o Playful cleverness
- real physics
  - o the programs used real mathematical equations to simulate being pulled into the sun if you got too close

It would take over a decade to create a game equal to Spacewar! for the general public to play. The game inspired many young men to duplicate the efforts of the Tech Model Railway Club.

You can go online today and play the original game!

The first two notable games in our history were invented just for fun with no other reason than to demonstrate the equipment and abilities of the programmers. What is more impressive is the fact that it would be years before the general public would get their chance to play games without the need for a laboratory or expensive equipment. The two games discussed introduced the following concepts that are still applicable to games today:

- visual representation of the player or his actions (ball moving, space ship moving)
- real physics
- input devices
- two players
- vector graphics

What is vector graphics?

One way to represent an image is to use a series of mathematical equations based on points, curves and shapes or polygons. We will see in the chapter on game graphics that vector representation of image has many advantages over using a bitmap. Vector graphics requires more processing power than a game based on bitmaps and hence it would take more time for microprocessors to be fast enough to handle a vector based game in fact, it would take until 1979 with the game *Asteroids* to create a game equivalent to the features of *Spacewar!* for the general public to enjoy a vector graphics game.

Unlike the game *Spacewar!* many games use what is referred to as raster graphics or bitmap graphics to display images on a screen. A raster graphic is a bitmap image representing a grid of pixels. A pixel is a point on the television or computer screen that can be a distinct color. Bitmap images are quite common for games and images you see on a web page. Each pixel "contains specific color information. A typical bit map image can consist of hundreds or thousands of pixels. The contents of the image can only be seen when the image is magnified.

RED 80%	RED 36%	RED 93%
GREEN 80%	GREEN 36%	GREEN 91%
BLUE 77%	BLUE 13%	BLUE 0%
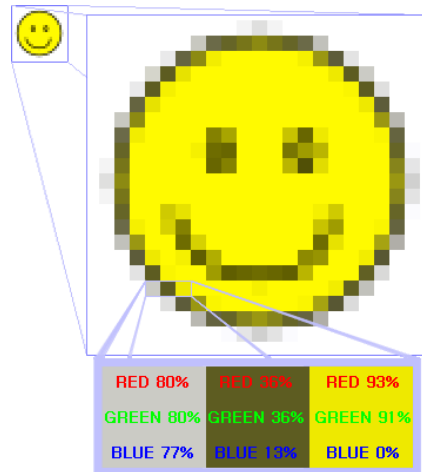
Figure 7 - A raster graphics - RGB bitmap image

The image above[7] (see Figure 7) illustrates how the top left image of the smiley face really appears when you take a look at the underlying pixels that make it up. It will take some years before the first arcade machine is introduced that uses vector graphics technology rather than the easier to display raster graphics.

First Generation

The father of the video game system

In the 1960's the only people playing games on computers were students that had access to the few PDP machines or IBM mainframes on college campuses. It would require a man with a vision to expand to a device that was in most people's living rooms – the television set.

Figure 8 - Ralph Baer

Ralph Baer emigrated from Germany to the United States with his parents in 1938 because of the Nazi regime's harassment of men and women of Jewish ancestry. He served in WWII for the US Army. Like many men of that era he went back to school to get a college education. In 1949, he graduated with a degree in a cutting-edge technology of the time – a B.S. in Television Engineering. In, 1951 he conceived of the idea as an employee for electronics communication company Loral, of building an interactive game for the television. He was a man way ahead of his time, since TVs were just starting to be

---

[7] From http://en.wikipedia.org/wiki/Raster_graphics

introduced into American households in fact in 1950 only 10 percent of American households had a television set. It wasn't until he became a lead engineer for the defense contractor Sanders Associates that Baer revisited the idea of building a TV-based videogame system.
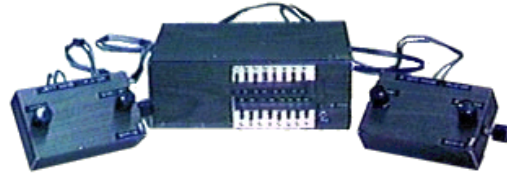
In 1966, Baer wrote a paper outlining a low-cost machine that will attach to a standard television set. He also outlines some game ideas that fall into the category of action, puzzle, instructional and sports.

In 1967, Baer with the help of Bob Tremblay, Bob Solomon and Bill Harrison worked on a multi-game system. The system (Home TV Game) provided the capability for users to play chase games, use a light gun and other simple games. Bill Rush joined the team and created a device that uses three spot generators to produce two onscreen paddles and a ball.
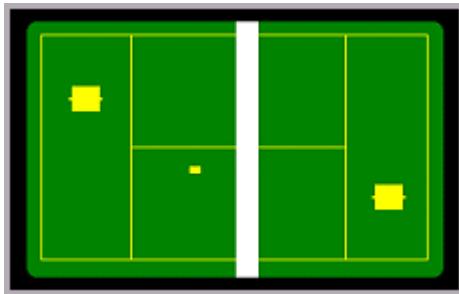
This was the first commercial Pong game as we know it today. The technology allowed different varieties of games to be developed – volleyball, handball, hockey and even shooting games. The shooting games used a rifle that today we call a light gun. The version of the device that Baer and company demoed and showed to television manufactures was called the Brown Box. It was not until 1971 that the company Magnavox saw the possibility in mass producing and selling this device.

Magnavox re-designed the machine not to include color circuitry in order to save money but instead used TV overlays – which were plastic sheets that users placed on their TVs in order to simulate background graphics. The system initially came with twelve games, chips, scoreboard, cards, dice, and other accessories.

FOOTBALL. Just like the pros. Plan your own strategy. Pass, run, even kick. Touchdown!

SKI. Race the clock as you schuss in and out between the flags.

STATES. A fun maze of riddles and rhymes that helps you learn the states and their capitals.

ROULETTE. Fascinating casino action. Put your chips down on your lucky number and try to break the bank.
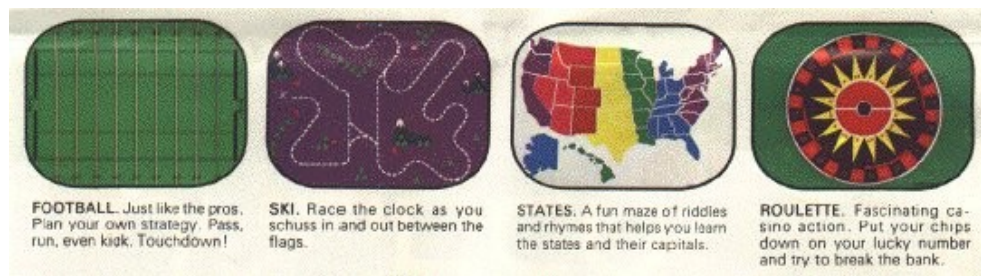
Figure 9 - Some Odyssey overlays

15

The Magnavox Odyssey replaced the Brown Box's use of game-selection switches with plug-in programming cards and was released in 1972. This allowed them to develop and sell new games for the system. Unfortunately, the marketing of the system as mishandled by only allowing Magnavox dealers to sell the system and misleading the public into thinking the device only worked on Magnavox TVs. Magnavox did manage to sell 80,000 units but it is possible many more could have sold if it was marketed correctly.

Figure 10 - The Magnavox Odyssey (1972)

The two notable games that came from this game system was the Pong game (with sound and all) and the use of a light gun in video games.

How do light guns work?

A light gun looks like a toy gun or rifle and is used as pointing device for computers, arcade or video consoles. Users aim the light gun towards the TV screen or cathode ray device in order to shoot at moving targets. Light guns were actually used quite early (before TVs) in the devices created in 1930's using vacuum tubes. Light guns became popular in early arcade shooting games. The early light gun was aimed at small moving targets (e.g ducks) onto which a light-sensing tube was mounted; the player used a gun (usually a rifle) that emitted a beam of light when the trigger was pulled. If the beam struck the target, a "hit" was scored. Today, modern screen-based light guns work quite differently – the sensor is built into the gun itself, and the on-screen target(s) emit light rather than the gun. Traditional light guns can only work with standard CRT monitors.

The father of the video arcade/video game industry

Figure 12 - Nolan Bushnell

The man regarded as starting the video arcade business is Nolan Bushnell. As a student in the University of Utah he got to play endless games of *Spacewar!* on one of the few PDP-1 machines available at universities in the country. Bushnell was perfectly positioned to see the possibilities of the game and the commercial possibilities of creating machines for the general public to use since he spent is summer months working at local arcade amusement parks.

Figure 11 – Computer Space cabinet

Bushnell had a sense of what people enjoyed playing and what they would pay to play. Bushnell worked with Ted Dabney to develop a version of *Spacewar!* into a coin-operated arcade game. Bushnell realized there were not enough quarters to justify building a game machine from a $100,000 PDP-1 computer so he built his version called *Computer Space* using electronic circuits designed to only play a knock-off of *Spacewar!* Bushnell sold the design to the company Nutting Associates. In 1971 they built 1500 units with a futuristic design[8] and fiberglass cabinet. This was one of the first[9] video arcade game machines. The game was also the first total arcade failure. They never sold all the units created. Why? The game did well in places near college campuses but was regarded as too complicated for the average Joe trying to relax at the local bar, since it required that you read the instruction manual in order to start playing.

Bushnell realized the game's flaws and decided his next creation would be simpler for the user to get started in playing. The next game was highly influenced by a demonstration he saw of the Magnavox trade show of the Odyssey game *Pong*. Nolan Bushnell started his own company with Ted Dabney and Larry Byron (who quickly bows out of the new venture). They pool their resources of $250 each (their profits from *Computer Space*) and at first call the new company Syzygy[10]. The company name was already taken by another

---

[8] It was so futuristic looking it was actually used in the sci-fi movie Soylent Green.

[9] The first known coin-operated computer game was Galaxy Game. Only one unit was built. It was also a version of Spacewar!

[10] Syzygy is a term used in astronomy meaning the perfect alignment of the Earth, Moon and Sun.

company so they go and rename the company Atari.[11] In 1972, a 27-year-old Nolan Bushnell officially starts the company Atari. Nolan hires a brilliant engineer named Al Alcorn to build Atari's first arcade machine. He gives Alcorn the assignment to build a simple tennis game where there are two players on each side controlling a paddle and knocking a ball back and forth.
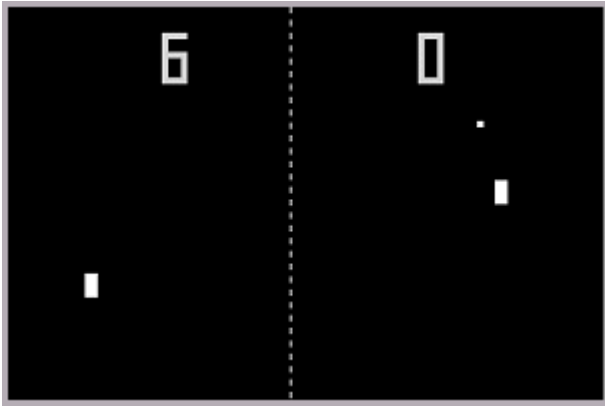
The game was built using solid-state electronic circuits and had very simple instructions for the players – "AVOID MISSING BALL FOR HIGH SCORE". Acorn incorporated some novel features for the game:

- the unique sound when the ball hits the paddle
- the ball speed increasing as the game progresses
- how the ball angled differently depending on which part of the paddle it came off on
- onscreen scoring

Bushnell assigned the game *Pong* to Al Acorn only as a test before pursuing more complex and "real" games. Atari installed the game at a local bar in Sunnyvale California called Andy Capps. Soon after installing the game machine Alcorn gets a phone call from the irate bar owner complaining that the game is broken and he should remove the machine. Alcorn goes over to inspect the machine and discovers that the game was not broken at all but it was jammed with too many quarters. The game is a smash hit and pulls in over $100 a week (Bushnell only expected to garner $25/week!). Bushnell did try to sell the game to the leading pinball arcade companies but changed his mind after seeing how successful the game was. Atari sold over 8,500 machines at a time when selling 2,000 pinball machines was considered a very good run. Atari could have gone on to sell more

but for the fact that many clones of the game were sold by other companies. It is estimated that over 38,000 Pong machines were made and sold. At this time, Ted Dabney sold his portion of the company to Nolan Bushnell due to his concern over competition and bootleggers selling so many variations of *Pong* machines (even their old partners Nutting Associates sold a version of *Pong*.)
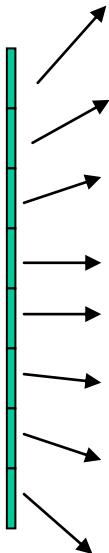
An interesting side story is the fact that the game *Pong* leads to one of the first patent infringement lawsuits in the video game industry. It was not lost on Magnavox the

---

[11] Atari is a term used in Bushnell favorite game Go, it is equivalent to the term "check" in chess.

similarity of the video arcade game *Pong* to their table tennis game for the home system Odyssey, even if the changes from Acorn made the video arcade game version more playable. Atari smartly settles with Magnavox for what becomes a small fee for the license - $700,000.

Nolan Bushnell had a terrific solution to all the "jackals" that just cloned all of the early Atari coin-op games – innovate, innovate, innovate. He encouraged his engineers to create new games as quickly as possible – to always let the competition eat their dust.

Innovations added by Acorn to the game Pong

- Divided paddle into eight segments and the collision with the ball would determine how the ball came off the paddle
- After a certain number of exchanges the speed of the ball would progressively increase
- The ball sound – "Pong" which came from extra circuitry already in the design of the game

---

Innovative Atari Games or "Life After Pong"

Atari's second game machine was again developed by Al Acorn in 1973. It was the game *Space Race*. The game had one player (vs. time) or two players racing through space from the bottom to the top trying to avoid asteroids and meteors. When a player gets hit by asteroids they start over at the bottom, when a player gets to the top they get a point. The player could only move vertically up. The key new component in the game was the use of time for the player to complete a task.

The next original game from Atari in 1973 was the game *Gotcha*. This was the first maze arcade game. There was also plenty of controversy associated with the game. Since

joysticks on most of the Atari arcade machine resembled a phallus, the Atari engineers decided to make a "female game" where the input device resembled pink boobs that were squeezed in order to control the action. Later versions of the game replaced the boobs with more standard joysticks. The notable feature of the game was the fact that two players navigated an on-screen maze, attempting to catch other.
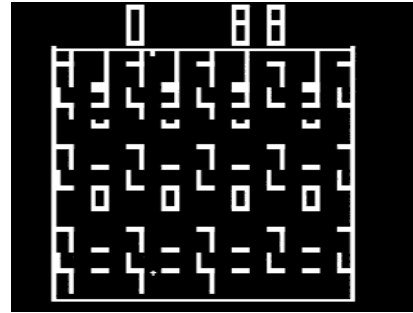
The next original game for Atari was the first racing game – *Gran Trak 10*. *Gran Trak*

10 was a single-player racing arcade game released by Atari in 1974. The player raced against the clock, accumulating as many points as possible.
It was one of the first games to use a ROM chip to store the game images, and was therefore the first game to have defined characters beyond a simple box (as with Pong) or a collection of dots (as with *Computer Space*). The primitive diode-based ROM was used to store the sprites for the car, the score, game timer, and
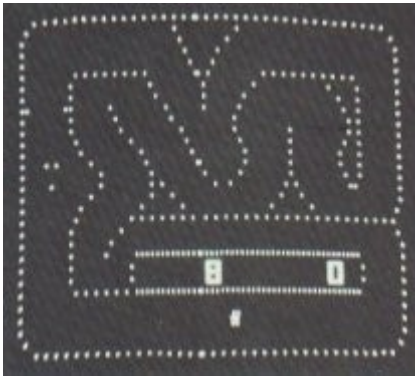
the race track. The game's controls — steering wheel, four-position gear shifter, and accelerator and brake foot pedals — were also all firsts for arcade games. Atari ended up losing money on the game *Gran Trak 10* due to an accounting error that had the machine selling for $995 whereas it cost $1095 to manufacture. They sold plenty at a lost.
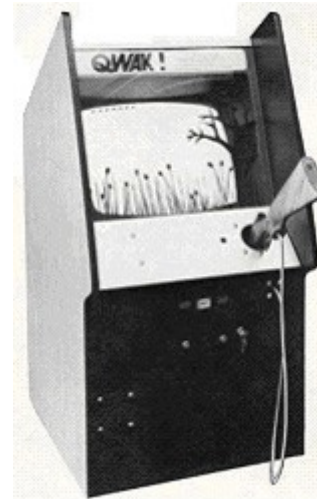
Gran Trak 10 was one of many games designed by the think tank in Grass Valley, California called Cyan Engineering. They provided many great design and technical ideas for games but in the case of *Gran Trak 10* the game was designed was "impossible" for Atari to build without the assistance of Al Alcorn. The delay in manufacturing and miscalculation in the actual cost to sell it lead to a loss of $500,000 for Atari in 1975.
 Another unique game for Atari in 1974 was the game *Qwak!* The game involved players firing at flying on-screen ducks with a rifle (light gun).
Qwak!

Kee Games

Figure 13 - Joe Keenan and Nolan Bushnell

Nolan Bushnell was a very bright guy and he did two important business savvy moves to ensure the success of Atari. The first was to lock in chip development deals with many different microprocessor manufactures. The main purpose was to keep the innovations created by these companies out of the hands of this competition for the arcade and home game systems. The second was more important in how arcade game machines were distributed. The pinball industry business practice of setting up exclusive distribution deals for arcade game machines limited the sales of Atari arcade machines. Atari was prevented from selling to more than one distributor in the same area. To get around this limitation and increase Atari's market share Bushnell talked his best friend and next door neighbor, Joe Keenan to create a company named Kee Games using 'so-called' defectors from Atari. At first Kee Games built and sold clones of Atari games to distributors that Atari was unable to sell to. But, they created their own innovative games that competed with Atari. One original and very successful Kee Game designed by Steve
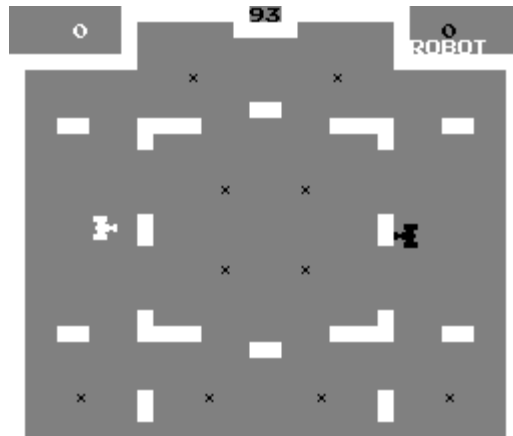
Figure 14 - Ultra Tank

Bristow in 1974 was the game *Tank*. Players move their tanks through a maze on the screen, avoiding mines and shooting each other. The players are represented by one black and one white tank sprite, and mines are denoted by an "X". Points are scored by shooting the opponent or when a player runs over a mine, the player with the highest score at the end of the time limit winning the game.

The great success of the game *Tank* helped Atari to get out of the red in 1975 and lead Atari to reveal its ownership of Kee Games. Another benefit was that distributors no longer insisted on exclusivity since so many of them wanted to get in on all the successful video arcade games.

The image shown is actually an image of *Ultra Tank*. Kee Games eventually was merged back with Atari and Joe Keenan became Atari's president.

Innovative Games from other companies.

The first game to use a microprocessor was the game *Gunfight* (1975). You had up to two players

21

Figure 15 - Gunfight

represented by Old West cowboys squaring off in a duel. The microprocessor was an Intel 8080. The game was originally designed by the Japanese company Taito. The game was greatly improved by Dave Nutting of Nutting Associates and distributed in the U.S. by Midway.

The first game based on a big movie hit was *Shark Jaws*. The game featured animated characters. The manufacturer was the company Horror Games by Nolan Bushnell. The game was not manufactured under the Atari label in order to avoid licensing issues with Universal Studios but it was trying to profit from the popularity of the movie Jaws.
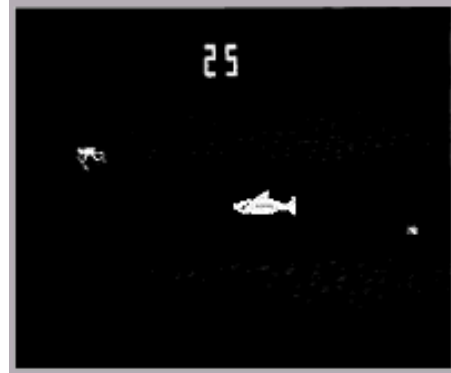
Figure 16 - Shark Jaws

At this point in time innovative games were still being developed on computers located on college campuses. On the PDP-10 the first early computer role-playing was Don Daglow's *Dungeon*. In the spirit of using an idea from another source the game was based on the game Dungeons & Dragons. The game was mostly text-based but did have some graphics of dungeon maps. In the same year William Crowther develops the game *Colossal Cave Adventure*, the first interactive fiction or text adventure game for the PDP-10. By the time, I got started programming someone converted the game to play on an IBM mainframe computer I was supposed to be using for my computer course. I managed to spend many hours lost in a twisty maze of passageways, all alike… The game used text to describe your location and what was going on and you entered one of two word phrases to move or do things.

You are standing at the end of a road before a small brick building.
Around you is a forest. A small stream flows out of the building and
down a gully.

> enter

You are inside a building, a well house for a large spring.
There are some keys on the ground here.
There is a shiny brass lamp nearby.
There is food here.
There is a bottle of water here.

> get lamp

Ok.

> xyzzy

>>Foof!<<
It is now pitch dark. If you proceed you will likely fall into a pit.

> on

Your lamp is now on.

You should be able to find a version online to play on your machine. All you need is your imagination and quest boots to play. We will see these types of games become very successful on home computers.

The home version of Pong

Figure 17 - Home version of Pong

In 1975, Atari partners with Sears to create a home version of the game *Pong* that sells for $100. Atari successfully manufactured and sold 150,000 units of the machine. Sears sells the machine and it becomes a huge hit for Christmas 1975. The following year Atari decided to build the game on their own but needed money in order to expand production facilities. Nolan Bushnell arranged for a 10 million line-of-credit to expand from the venture capitalist Don Valentine.

The following year Atari makes available a new and improved version of *Pong* with a special chip developed just to play the game, but as usual many other companies have their own version of Pong available for the 1976 Christmas season. The most successful of these companies is Coleco since they were the only company that was able to obtain their shipment of a new chip that was called "Pong-on-chip" developed by General Instruments. Their machine the TELSTAR sold for half as much as Atari's and started the home video game competition. Since 1976 many companies and home game systems have come and gone.

Figure 18 - Coleco TELSTAR

Atari goes on create many versions of Pong – *Pong Doubles* (4-player game), *Super Pong* (had 4 games), *Ultra Pong* (16 games and up to 4 players).

The story behind the game – Breakout

The game *Breakout* was conceived in 1975 as delightful variation of the game *Pong* that

was conceived by Nolan Bushnell and Steve Briskow of *Tank* fame. The idea was to have users break down a wall with a ball.
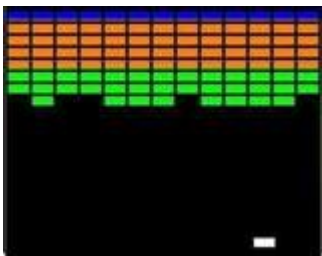
It was in the year 1974 that a young college drop-out named Steve Jobs became employee number 40 for Atari. The engineers had mixed feelings about him since he had the dress habits of a hippie – walking around shoeless dirty feet, not bathing often, and always talking about going to India in addition to being totally obnoxious.

Figure 19 - Steve Jobs and Steve Wozniak

Steve Jobs was put on the evening shift to remove him from contact from "regular" employees. His main job was to evaluate and suggest improvements to game concepts developed by the Cyan Engineering at Grass Valley. He took the opportunity to invite his best friend and engineering whiz Steve Wozniak (Woz) to join him at Atari after putting in a day's work at Hewlett-Packard. Woz spent is evenings playing games and checking out the design of games. Nolan Bushnell offered a bonus to anyone who could come up with a design for Breakout with the smallest number of integrated circuits (IC). At the time the main cost in the manufacture of a video arcade game was the number of ICs it

required. The typical game required about 150 chips, a less complicated games around half that. Steve Jobs talked his friend Woz into coming up with the design in four days. Woz came up with a design that required an astonishing 46 chips! Steve Jobs split the bonus of $750 with Woz. The only problem was that the bonus was actually $5000 not $750. Unfortunately, Woz's final design had to be redesigned by the folks at Cyan Engineering since engineers responsible for translating it into a manufacturing spec could not figure out how it worked and Steve Jobs could not really assist them since he did not know as much about electrical engineering as folks thought. It wasn't until years later that Al Alcorn learned that Steve Jobs knew less engineering then he let on and actually had Woz do all the work while he took all the credit.

The game Breakout became a big hit for Atari and reappeared as a bit hit on their home video system.

Of course we mention the Jobs and Wozniak since they team up to co-found a new computer company based on Woz's design for a personal computer - The Apple. The company went on to create the revolutionary Mac and of course the ubiquitous iPhone. I should note that when Woz found out many years later about how much he was short changed for his Breakout design he was too happy about his less than honest partner.

The Fairchild Video Entertainment System/Channel F

In 1976, Fairchild Semiconductor released a unique home video game system the Channel F. the home system used Fairchild's F8 microprocessor. The game system has some original ideas for the time:

- detached controllers with a grip design connected to the system via wires
- the console was programmable – which means many different games could be created for the system
- plug in cartridges what would contain ROM chips – the beginning of the video game cartridge

The Channel F game system set the bar for future home video game system to meet or exceed.

First Game Controversy – Death Race
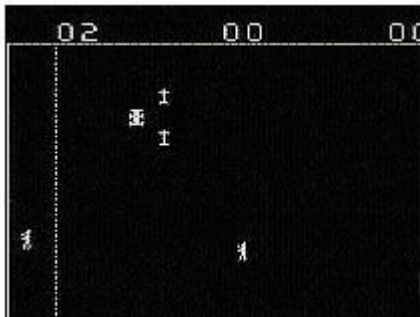
The game Death Race was released by company Exidy in 1976. The game was a driving game where the player drove over animated stick gremlins to score points. When a player successful ran over a gremlin a scream could be heard and a tombstone would appear where the gremlin was killed.

The game created a lot of controversy since the graphics were not clear enough to distinguish between the car mowing down a gremlin or of a person. There was much protest and the hullabaloo over the game that was well covered in the media. Rumor has it that incited groups of people would storm arcades and drag the unit out to burn it. It did not sell as many arcade machines but the game actually spurred a lot of interest in the video arcade game industry as the industry went on to see it greatest years.

Night Driver

In 1976, Atari introduces the first-person racing game. It is one of the first examples of a game having real-time first person graphics. The player uses a steering wheel to control a car along a road at night. The night time setting allows minimal use of graphics for the surrounding area. The game was later ported to the Atari 2600 game system. I mention this because I spent many hours playing this game and still enjoy the simple idea behind it.

The most popular arcade game for 1976 was Midway's *Sea Wolf*. The game was another creation by Dave Nutting and introduced an unusual cabinet and game play as the player used a periscope while manning a submarine and shoots torpedoes with a thumb button.

Second Generation

The history of the 6502 chip
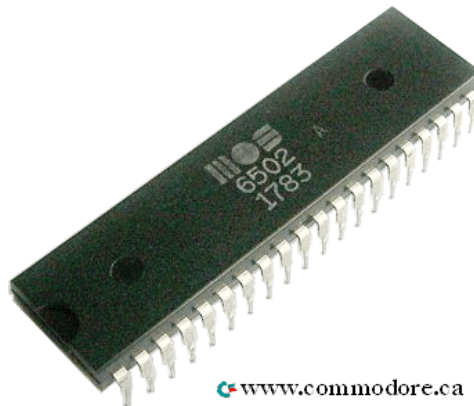
What made the second generation in the history of games possible was the use of computers called microprocessors like the 6502 in the construction of video arcade games, home video systems and personal computers. The 6502 chip went on the power many of the most popular personal computers and home video game systems like the Atari 2600, Apple IIe and the Nintendo Entertainment System.

The father of the personal computer industry
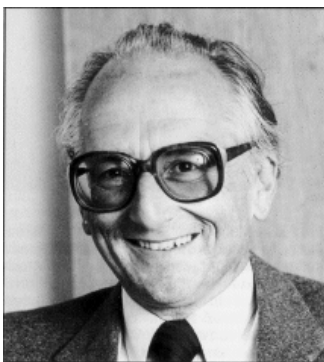
Figure 20 - Chuck Peddle

Chuck Peddle can be regarded as the father of the personal computer since his invention – the 6502 microprocessor went on to power the most popular personal and best selling person computers, such as, the Commodore 64, Apple IIe, and Atari computers.

While working for Motorola he participated in the design of the leading Motorola microprocessor – the 6800 and its support chips. He tried to sell the Motorola on the idea of creating a more inexpensive microprocessor as compared

to the $300[12] price tag of the 6800 chip. Motorola did not want to pursue such a project so Peddle took a team from Motorola to the company MOS Technology with the goal of building a low-cost chip. They created a chip (the 6501) for the unheard of price of $25. One of ways they were able to reduce the cost of the chip was to improve manufacturing yields. In the 1970's chip manufacturers typically yielded only 30% of non-defective chips from a chip run. MOS created a technique that produced a 70% success rate. This manufacturing breakthrough allowed the creation of the $25 computer chips. Seeing the success and competition Motorola sued MOS for patent infringement. One key factor in Motorola's favor was the fact that the 6501 was pin compatible to their chip the 6800. MOS redesigned the chip and created a non-pin compatible chip – the 6502. Peddle designed two 6502 trainers the TIM-1 (Terminal Input Monitor) and KIM-1 (Keyboard Input Monitor) to teach engineers how to use the chip. Thousands of these trainers would be sold to budding engineers since they saw the potential for these systems to be single board computers. In addition, many engineering hobbyist like Apple engineering legend Steve Wozniak would get their hands on the 6502 and use it to build their own personal computers.

The Atari 2600 (VCS)

Figure 21 - Atari 2600 with paddles, joysticks and games

There were many home video game systems developed around 1976-1978 but the most successful video game system of this era is the Atari Video Computer System (VCS) later known as the Atari 2600. In fact, I was surprised to learn that many of the students in my game design course played their very first video game on this system – this is in 2006!!

In 1976 Atari started research on developing a home video game system that was flexible enough to play more than one game. The new system used some of the ideas first seen on the Fairchild Channel F machine – the use of game cartridges. The use of cartridges to hold the game code instead of having the games inside the machine in some ROM device would allow the manufacture to make money on additional games after the

---

[12] This would be equivalent to about $1300 in today's dollars.

customer got the game system home. The system was developed by Cyan Engineering[13] and the prototype machine named *Stella*[14].

By this time in Atari's history Bushnell was rather tired and spent running multiple enterprises like the video arcade business, the home console systems and his idea for a family friendly arcade environment named Chuck E. Cheese. Atari did not have the funding necessary to develop enough game systems to take advantage of the anticipated demand for Christmas 1977 so Nolan Bushnell decided to sell Atari to Warner Communications for $28 million. Nolan stayed on as CEO and Joe Keenan stayed on President.

The machine was bundled with two joysticks, a pair of paddle controllers for pong games, and one game – *Combat*, based on the popular arcade game Tank. The console had a total of nine cartridge games for sale – *Combat*, *Air-Sea Battle*, *Basic Math*, *Blackjack*, *Indy 500*, *Star Ship*, *Street Racer*, *Surround* and *Video Olympics*.
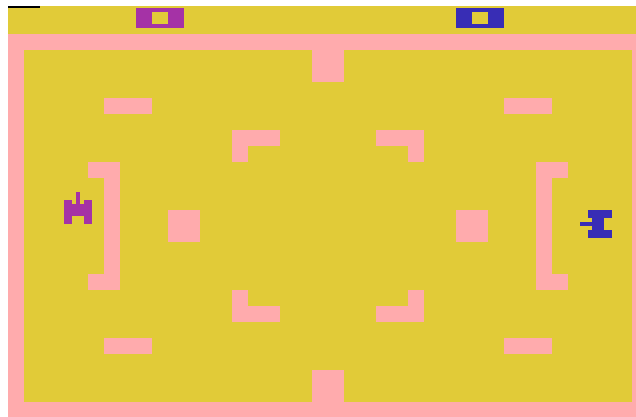
Figure 22 - Screenshot of Combat

---

[13] By this time Cyan Engineering as part of Atari

[14]Rumor has it that Atari named many of their machine prototypes after females the engineers admired for one reason or another. In the case, of the VCS, the name was supposedly the name used by one engineer's bicycle….maybe we should ask who was his bike named after?
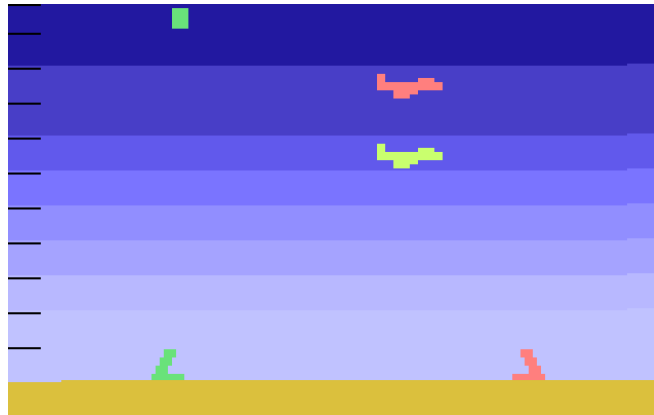
Figure 23 - Screenshot of Air-Sea Battle
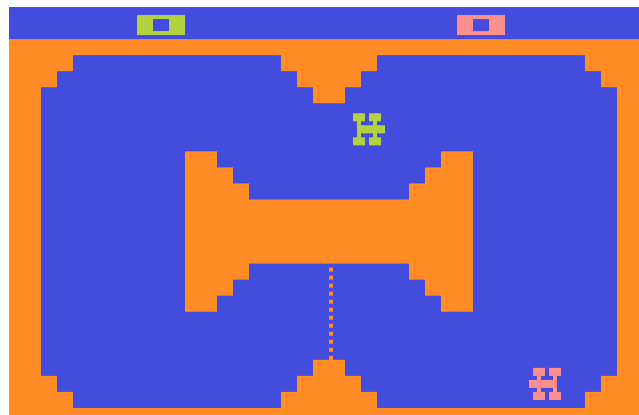
Figure 24 - Screenshot of Indy 500
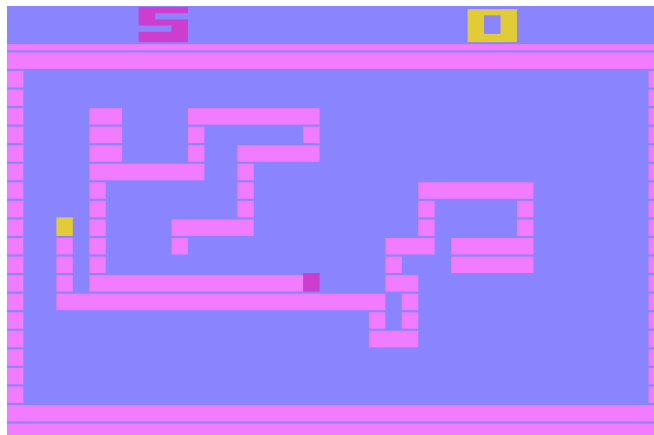
Figure 25 - Screenshot of Surround

The CPU used to power the Atari 2600 was a version of the 6502 known as the 6507, which was just a cheaper version of the 6502. The device had a combined RAM-and-I/O chip and a special chip used for display and sound called the Television Interface Adaptor (TIA) chip.

In the Christmas season of 1977 when the game console was first sold to the public, Atari had competition with the Fairchild Channel F, many discounted pong clones that knew their days were numbered given these more powerful and versatile machines like the Channel F, and handheld machines. Sales were disappointing for Atari and the parent company Warner Communications on their $199 machine since Warner had sunk 100 million dollars in Atari, and created many VCS machines in hopes of overwhelming the competition (Bushnell tactics) that Christmas. Nolan Bushnell was still head of Atari and decided that the solution was to move on to develop a bigger and better machine. This did not sit too well with Warner executives, since they manufactured many machines that did not sell and they felt the machine had a lot of promise and potential, so they decided to make a change and ease Nolan Bushnell out of Atari. New management came in and for Christmas 1978 Atari did quite well with new games such as *Outlaw*, *Space War* and *Breakout* and by this time there was a realization by the public that the system was not just a *Pong* machine but unlike other machines was extendable by the purchase of additional game cartridges.
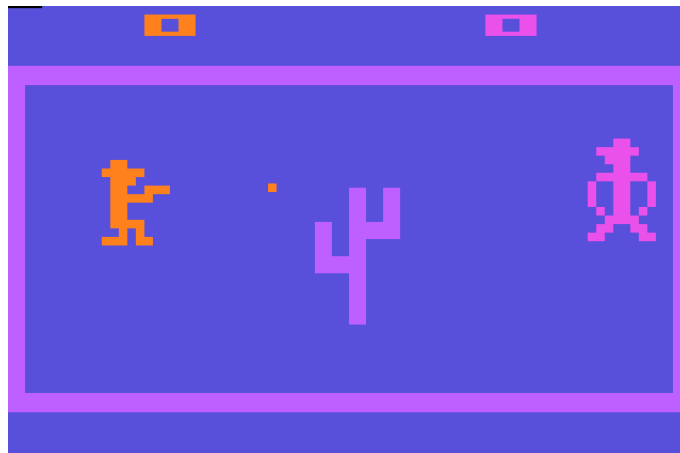
Figure 26 - Screenshot of the game Outlaw

Figure 27 - Screenshot of the game Space War

Figure 28 - Screenshot of the game Breakout

The greatest selling point for the Atari 2600 was that it allowed gamers' to play a version of their favorite video arcade game right at home. The machine helped to springboard the multi-billion dollar home video game industry we have today. The machine was "the must-have machine" to have throughout the 1980's. It went on to sell over 30 million consoles all over the world.

By 1980 there were other companies competing for the video game market – Mattel Intellivision and Magnavox Odyssey2. The Intellivision did quite
well and was an impressive machine. The Magnavox Odyssey2 didn't quite match up to either machine and did not fare as well in the market.

All these home video game machines mostly reproduced variations of the games that were popular at the video arcade.

It was at this time that console manufacturers initiated the idea of selling the console as inexpensively as possible (today many sell at a lost) and making up the cost by having customers purchase games for their systems. This is the "give them the razor, and let them buy the razor blades" philosophy of business.

Notable Video Arcade and Handheld games 1977-1979

Many innovations in games were still occurring in the video arcade industry and soon the release of the very same games on home video game systems would generate as much money as the arcades.

1977 and 1978

The game *Space Wars* was developed by Larry Rosenthal for Cinematronics. The

company was a pioneering video arcade game developer and released games using vector graphics display. The idea of using vector graphics for game display was first seen in 1962 with Steve Russell's *SpaceWar!* on the PDP-1. The game Space Wars was the first vector graphics arcade game. This game really took off selling over 30,000 units. The company Cinematronics goes on to build and sell some very popular video games based on vector graphics technology – *Warrior* (1978), *Rip Off* (1979) and many more.
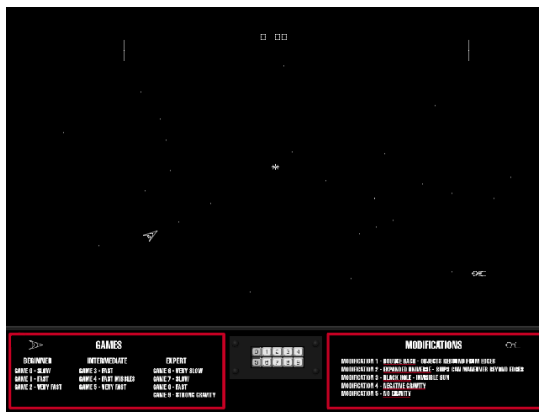
In 1977, Mattel started to release handheld games. One very popular and novel game was Missile Attack. The game imagined for the player a City resembling New York City where the player must protect from ICBM's raining down. The games were popular but the LED display did not compare to the video arcade and home video look and feel of games. In addition, today handheld machines play many different games and are powerful enough to play games that were originally designed on 32-bit home video games systems.

By this time computers were connected together via networks, where software could easily move from machine to machine. "The global Internet's progenitor was the Advanced Research Projects Agency Network (ARPAnet) of the U.S. Department of Defense." [15] This network provided student with a love of playing and constructing their own games the opportunity to share their games with others across the country. William Crowther's game *Colossal Cave Adventure* was made available via this network for many students to play and improve. The game made the rounds at MIT in 1977. It was at this time that Marc Blank, Bruce Daniels and Tim Anderson got together to create a new game involving maps, problem solving and of course the cast of characters expected in a computer adventure game, "..it had the thief, the Cyclops, the troll, the reservoir and dam, the house, part of the forest, the glacier, the maze,…"[16] the game was named Zork and it

---

[15] http://www.dei.isep.ipp.pt/~acc/docs/arpa.html
[16] http://www.csd.uwo.ca/Infocom/Articles/NZT/zorkhist.html

too ran on a PDP-10. The game became very popular and well played from the user community called 'net randoms' that infested the MIT systems. The game was improved and redone for Apple II and Commodore 64 to much popularity and success.

At this time Kelton Flinn begins development on a text-based aerial combat game called *Air*, which is a precursor to the 1987 game *Air Warrior*, the first commercial massively multiplayer online game with graphics. "If Air Warrior was a primate swinging in the trees, *AIR* was the text-based amoeba crawling on the ocean floor. But it was quasi-real time, multi-player, and attempted to render 3-D on the terminal using ASCII graphics. It was an acquired taste." [17]

1978

In Japan, the game responsible for an acute shortage of the Japanese equivalent of quarters and endless lost hours in 1978 was the game *Space Invaders*. The game was originally created by Taito. There is a story that components of the game originally served as test for incoming programmers and was not intended to be a game. The game was distributed in the United States by Midway. Space Invaders was one of the first shooting games or as we shall call them SHMUP (short for "shoot 'em up") . The graphics were simple and the aim was to move the user's laser cannon left or right, while dodging behind ever deteriorating protective blocks on the screen while trying to destroy the ever faster rows of aliens coming down. The game was created, designed and programmed by Tomohiro Nishikado. Two notable features introduced by the game was

- use of game "demo" screen or "attract" screen showing users how to play and enticing a passerby to drop a quarter into the slot
- popularized the concept of maintaining a "high score"
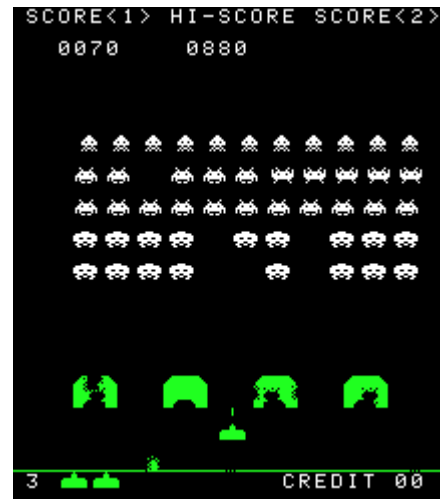
The game is addictive as you try to clear the screen of the aliens all the while your barriers missiles are eaten away by both your own and alien the arsenal. The sweat on your brow increases as well as your heart rate as you notice the aliens taking more frequent and better aim as the ships march down the screen attempting to destroy your cannon. Once in a while a red "mystery ship" with a markedly different sound appears at the top tempting you to stop shooting at the white intruders and try to go for the points that will be rewarded for taking down what must be the mother ship. When you play the game you have to take into consideration that nothing like it was seen in arcades machines and this time period many themes involving aliens and space ships was becoming an ever increasing part of popular culture.

---

[17] http://en.wikipedia.org/wiki/Kelton_Flinn

The technical details of the game are that it is a two-dimensional vertical shooter game where each level consists of clearing the screen. At each level the user is presented with faster and more accurate alien ships. The graphics are simple and the sounds are equally so. This game is truly a classic video games and a blockbuster for Taito and Midway generating over a half billion dollars in sales.
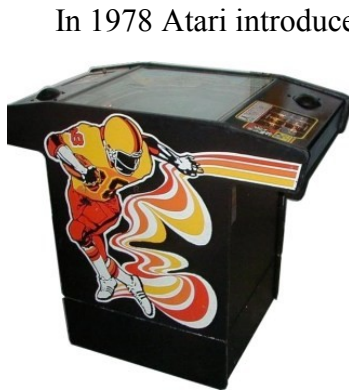
In 1978 Atari introduces the first use of a trac-ball for the game *Atari Football*. Players were represented as X's and O's on a black and white screen. Since the speed of the player in the game depended on how fast the player moved the trac-ball. Many players suffered hand injuries playing this game.
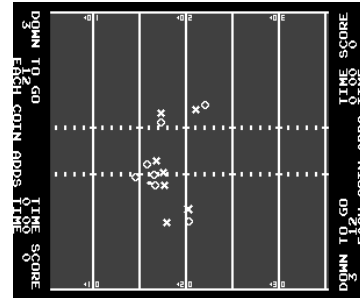
It was very popular but did not last after football season or compare in the long-term popularity to the game *Space Invaders*.

Figure 29 - Galaxian
Space Invaders started the game genre of shooter games in which one or more players control a vehicle or character and fights off large numbers of enemies in the form of monsters or aliens. The enemies appear in wave after wave and after clearing all the enemies the user gets to the next level.

The next game in the genre to expand and improve the play was *Galaxian* (1979). The game expanded on the formula pioneered by *Space Invaders* and added a new feature where the aliens move out from their formation and make kamikaze-like dives at the player's ship.[18]

Galaxian was very successful for Namco and introduced several "firsts". Although true color (as opposed to a color overlay for a game that was otherwise black and white) began appearing as early as 1975, *Galaxian* took graphics a step further with multi-colored animated sprites and explosions, a crude theme song, different colored fonts for the score and high score, more prominent background "music" and the scrolling star field, and graphic icons that showed the number of ships left and how many rounds the player had completed. These elements combined to create a look/feel that would set the standard for many other 1980s arcade games such as *Pac-Man*.

Cinematronics introduced another vector based game, *Tailgunner* in 1979. The premise of the game is that you are the tail gunner of a large space ship. Enemy ships approach

---

[18] http://en.wikipedia.org/wiki/Galaxian

the vessel for attack and you must aim you crosshairs and shoot them down before they slip past your cannons. The game ends when 10 ships get past your cannon. The game had a first-person perspective. It was the first video arcade game to feature three-dimensional animated objects.

Figure 30 - Tailgunner

Atari released their out vector-based coin-op game *Lunar Lander* in 1979. The object of the game is to pilot a lunar landing module to a safe touchdown on the moon.
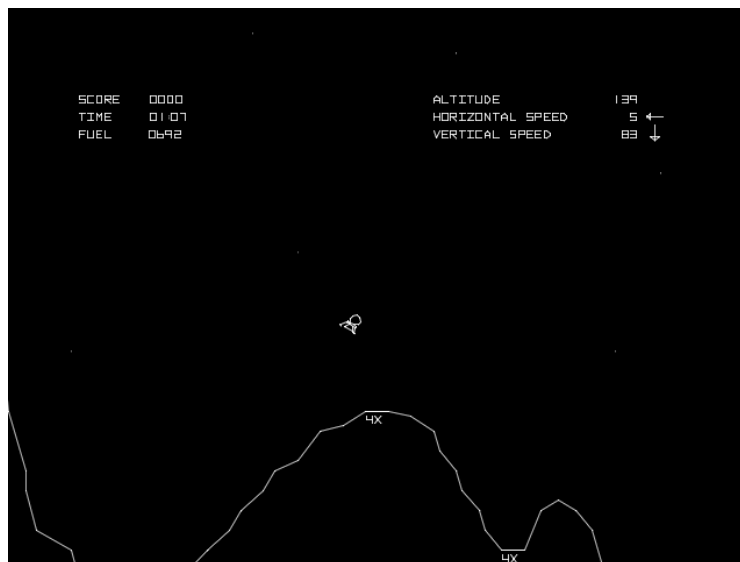
Figure 31 - Lunar Lander

The terrain is very jagged and has only a few flat areas appropriate for landing. These areas are highlighted with a flashing bonus multiplier, which is higher for smaller areas. If the player successfully lands the module, he or she is awarded points based on how good the landing was and the difficulty of the landing site.

36
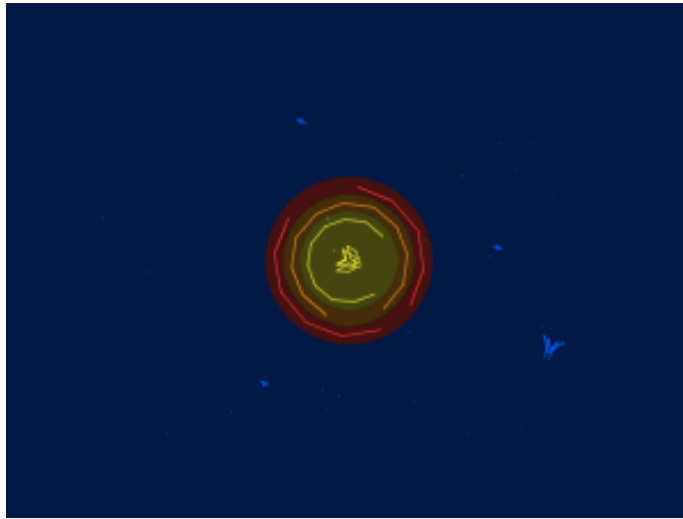
Figure 32 - Cinematronics Star Castle

Another popular vector based graphics game released by Cinematronics in 1980 was Star Castle. The object of the game was to destroy the enemy cannon which sits in the center of the screen surrounded by concentric circle. This game inspired the Atari 2600 game named Yar's Revenge.

Atari takes off!

By 1980 the home video version of the game *Space Invaders* helped to usher the Atari 2600 from one of the many competing home game machines to a console mega-hit.

What was different about the VCS for the user/player?

- Its games were more colorful
- Its control – the introduction of the joystick to the home video market
- It has game selector switches
- It allowed the player to select difficulty settings

The Atari 2600 was originally designed to handle simple games like *Pong* and *Combat*, no one anticipated the incredible variety of games this system would support. The reason the game console managed to support many different games was because significant functions were moved out of hardware and into software, this helped to

- Create a more open system
- It put the functionality in the hands of creative people who went far beyond what the original designers had ever envisioned

37

Things were initially great at Atari with Ray Kassar at the helm. Kassar replaced Nolan Bushnell in his 1978 ouster. Many other executives of the type that ran Warner Communications or similar industries started to take charge. These managers did not understand arcade engineering culture. The new management introduced dress codes, and time cards. Talent slowly started to leave Atari to form third party companies such as Activision. The newly formed companies competed with Atari in the construction of games for the VCS. In fact, the game programmers were derisively referred to "high strung prima donnas" by Ray Kasser. But, the fact is, the "Golden Age" of Atari was due to the creativity, innovations and great games developed by those prima donnas.

From: http://www.erasmatazz.com/library/JCGD_Volume_1/Egotism.html
By Chris Crawford

Volume #1 Issue 5 December 1987

Many years ago the CEO at Atari referred to the company's game designers as "high-strung prima donnas". The comment was not meant to be derogatory, but the designers responded with a T-shirt that proclaimed, "I'm just another high-strung prima donna from Atari." Perhaps the CEO was right.

I have known many game designers; they encompass a broad range of personalities. Yet all these disparate people share one common trait: they all sport towering egos. Each one is absolutely certain that his own talents are the purest, truest, most brilliant talents of any game designer in the world. I myself am given to introducing other game designers as, "...the second best game designer in the world."

Why is egotism so rampant among game designers? Is it perhaps the self-indulgence of a pampered elite? I think not. Consider, for example, my own case. Was Chris Crawford spoiled by too much press attention? The fawning masses, the rivers of adulatory prose, the screaming nubile nymphs (OK, so I exaggerate a little!) have all these things gone to my head to make me the hopeless egomaniac I now am? No, a thousand times no! Chris Crawford is too big a man to be spoiled by such trivial things! I was already spoiled long before any of this happened to me. Mine is a mature egomania refined and developed since the day I emerged from the womb and took a bow.

I think that egotism lives in game designers because of a selection effect. Game designers without healthy egos will never achieve as much as their better-endowed colleagues. The egomaniac sets higher goals for himself than he can reasonably expect to achieve. This may sound idiotic, but in a poorly defined field such as game design, it is the stuff of creativity. A civil engineer doesn't get too experimental with the bridges she designs, because it is easy to reliably calculate what will and what won't work. But we don't know as much about computer games. We don't know where the limits are. So we need these foolhardy egomaniacs who blindly plunge into the darkness, boldly going where no one in his right mind has gone before.

The egomaniac has another gigantic advantage over the more emotionally balanced person. In the darkest hours of a project, when the problems seem overwhelming and there is no rational basis for hope, a reasonable person would start casting about for ways

to scale down or even abandon the project. But the egomaniac lies face down in the mud of his own failure and then draws himself up, proclaiming, "I am ze magnificent Crawford! I weel find ze way!" Egotism, of course, takes a back seat to reality, and sometimes he fails; but when he succeeds, it seems like magic to the rest of the world.

There are, of course, liabilities created by egotism. There is the deadly difference between pre-project egotism and post-project egotism. The former serves to inspire the designer to greater heights of achievement. The latter convinces him that he has already scaled those heights. Post-project egotism blinds the designer to the flaws in his work and robs him of the ability to learn and improve.

Then there are the embarrassing consequences of an ego that is foisted on other people. It is one thing to smile inwardly in secret appreciation of your untouchable superiority; it is another thing entirely to tell it to other people. The mature, genteel egomaniac keeps to himself the untold story of his towering intelligence and blinding creativity.

So don't feel embarrassed by that ego of yours. Go ahead stand on the craggy mountaintop, lightning bolts playing about you, head held high as the furious wind hurls rain in your face. Laugh scornfully at the elements that doubt your greatness. Shout lustily into the tempest, "I am ze greatest game designer in all ze universe!"

Then crawl back into your cave and get back to work.

Figure 33 - Asteroids
Ed Logg worked on many seminal arcade games for Atari. He co-developed with Lyle Rains the arcade game *Asteroids* in 1979. It was one of the "most popular and influential games of the Golden Age of Arcade." The game used vector graphics and a two-dimensional view that wraps around in both screen axes. The player controls a spaceship in while asteroids of various sizes fly by and periodically a mean little flying saucer tries to take down the player. The object of the game was to shoot at everything and try not to let anything hit your spacecraft.

The game Asteroids became Atari's most popular arcade game.

Like many games of its time, *Asteroids* contains several programming bugs that were mostly the result of the original programmers underestimating the game's popularity or the skill of its players. The maximum possible score in this game is 99,990 points, after which the score rolls back to zero. There was also an oversight in the small saucer's programming which gave rise to a popular strategy known as "lurking" — because the

saucer could only shoot directly at the player's position on the screen, the player could "hide" at the opposite end of the screen and shoot across the screen boundary, while remaining relatively safe. This led to experienced players being able to play indefinitely on a single credit. This oversight was addressed in the game's sequel, *Asteroids Deluxe*, and led to significant changes in the way game developers designed and tested their games in the future.

On some early versions of the game, it was also possible to hide the ship in the score area indefinitely without being hit by asteroids.

The Atari 2600 or the VCS sales really took off when Atari started to convert their video arcade hits into VCS cartridges.

The notable games developed for the Atari 2600

Many of the successful VCS games released where actually versions of coin-op video arcade games, like *Pong*, *Breakout*, *Space Invaders*, *Asteroids*, but one the best game made just for the VCS was the game *Adventure*[19].

Combat (Atari – 1977)
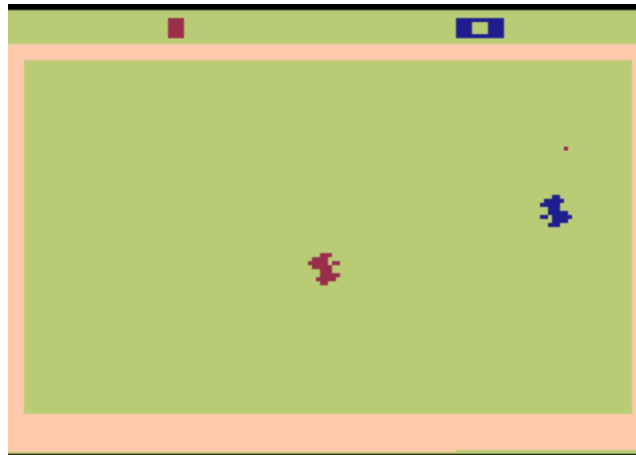
Figure 34 - Atari 2600 game of Combat

The game Combat was part of the original set of games released in 1977 for the new VCS. The game was based on the earlier coin-operated arcade game by the same name released in 1974. The game was sold with the system therefore it set expectations on playability and what the VCS could do with the hardware and software. The game advertised 27 games in one game cartridge – variations of Tank, Tank-Pong, Invisible Table, Bi-Plane and Jet.

---

[19] This game will always be in the VCS top-five.

Adventure (Atari - 1979)

In my opinion, the Atari 2600 game of *Adventure* is without question the best VCS game ever made. It was an inspired graphical version (a rather simple version) of the classic computer adventure based game of *Colossal Cave Adventure*. It had all the elements that make a game fun – you the hero with a quest, "An evil magician has stolen the Enchanted Chalice and has hidden it somewhere in the Kingdom. The object of the game is to rescue it inside the Golden Castle where it belongs…" The game has three Dragons that pursue you and hinder your quest, one dragon meaner than the next. There are three castles in the Kingdom – the White Castle, the Black Castle and the Golden Castle. The castles require that you obtain the corresponding colored key in order to open the gate to the castle.
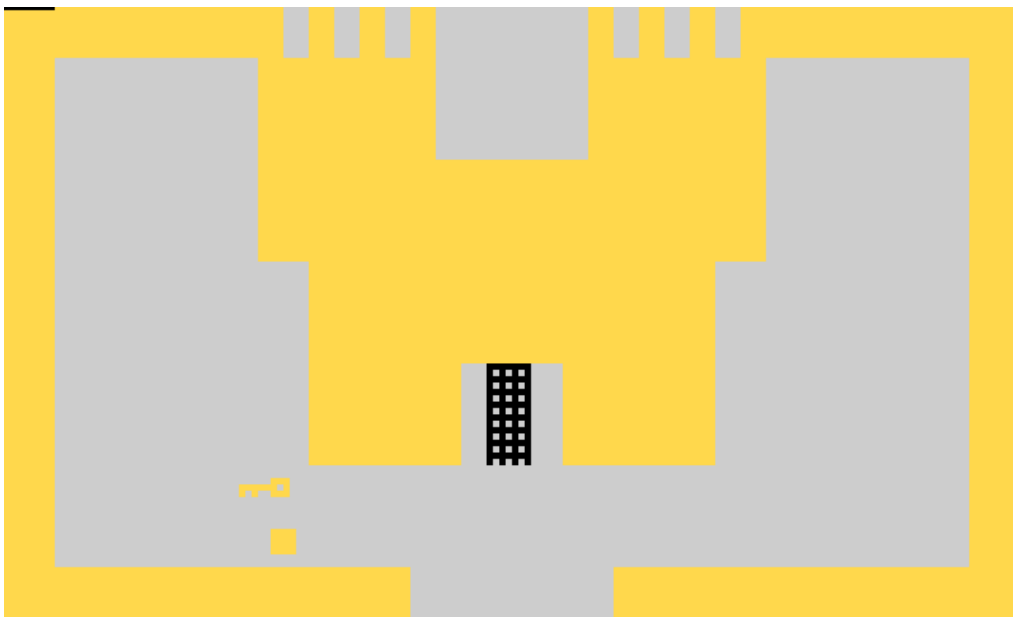
Figure 35 - Adventure screen shot

The screen shot above shows you the user (that yellow block!) holding the Golden Key in front of the Golden Castle.

Figure 36 - Adventure screen shot #2

The screen shot above shows you the user (red block…gee you change colors depending on the room) holding your sword (okay, it looks like an arrow ←) and the Black Key.
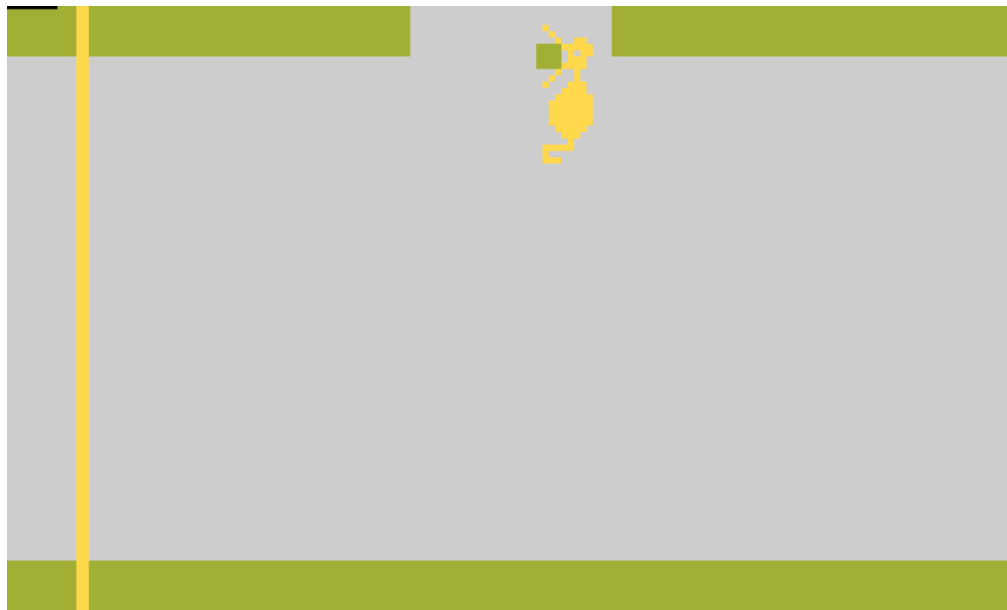
Figure 37 - Adventure screen shot #3

The above shows Yorgie the Yellow Dragon (okay so they dragons look like ducks) having lunch – you are the meal.

I remember playing the games for hours trying to make my way through the maze and catacombs searching for the right objects while avoiding the dragons and that stupid bat that flew around moving things around.
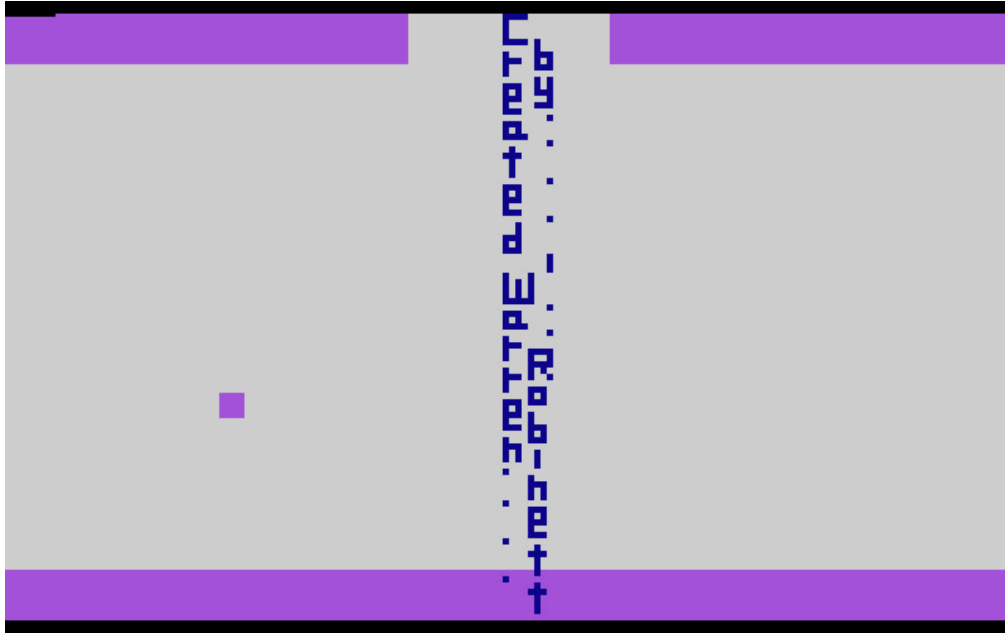
Figure 38 - Adventure screen shot of Easter Egg

The game was developed in 1979 by Warren Robinett for Atari, it went on to sell millions of copies. Since Atari, did not give credit to individual game designers or programmers Robinett introduced the first ever Easter egg. An Easter egg is a hidden message in a software application that can only be invoked or seen if player's go through a special sequence of steps. The Easter egg for this game displayed Robinett's name. Another notable feature of this game was the fact that the game allowed players to pick up (e.g. sword) items and drop them and move about the game with the item. Robinett soon after left Atari (as many of its developers) feeling unappreciated and uncompensated for his efforts. Let's compare the fact that Atari sold over a million copies of Adventure at $25 each and Robinett's salary was only $22,000 a year you can see why the great developers started to leave to form their own companies.

"I was tired of working, and Atari management didn't value the 2600 designers," he says. "Boy were they stupid, because the designers all quit and started competing companies." Years later, he notes with some enthusiasm, the company "came crashing down, like a whale dropped from a 747 at 30,000 feet".

- Warren Robinett from http://www.gamasutra.com/php-bin/news_index.php?story=13280

Space Invaders (Atari/Taito - 1980)
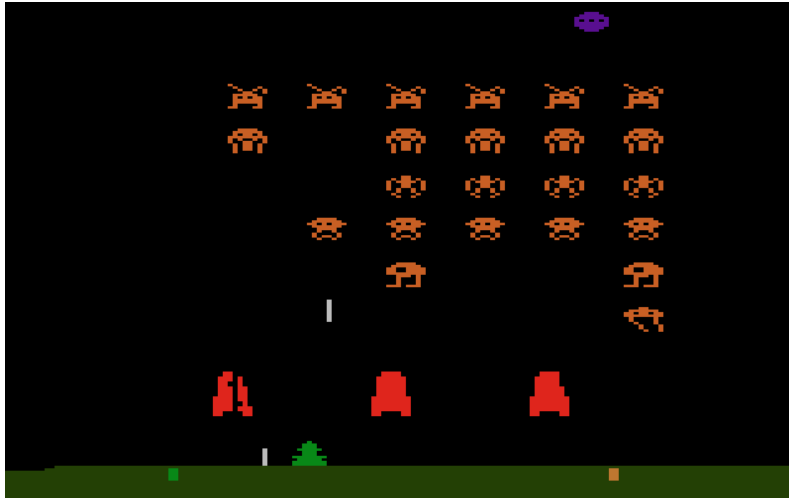
Figure 39 - Atari 2600 Space Invaders

The one game that made many people go out and buy an Atari 2600 was the game Space Invaders. It was converted into a home video game in 1980. It was the mega-hit Atari was waiting for in order to knock-out the competition and make the VCS the "must have" system. Gamers loved the game in the arcades and even more when they could just play over and over again at home. This game was "the" reason folks and went out and purchased an Atari 2600.

Each time you cleared the screen of those pesky aliens a new set would appear but slightly closer to planet earth. The game was actually un-winnable for the player but a lot of fun.

Asteroids (Atari – 1981)

Figure 40 - Asteroids for the Atari 2600

Atari translated its most popular selling coin-op into an equally popular home video game.

FROM THE MANUAL

On a quiet serene evening the Cosmic Space Patrol sets out for the usual night cruise through the boulevards of space. This beat was always the same; calm, no action and no excitement. For some reason this night feels different. Shortly before 0200 hours some form of intergalactic material is sighted through the visual particle counter. The material is too large a mass to measure. It's drifting closer. Lookout, it's a giant asteroid boulder and it's headed straight for the Cosmic Spacecraft. The only chance for survival is to dodge the boulder or destroy it. Destroying it doesn't mean just breaking it up, it means vaporizing it. Small asteroid boulders are equally as fatal as large ones.
Whew, the boulder just missed colliding with the Cosmic Spacecraft, but suddenly the Cosmic Space Patrol find themselves surrounded by thousands of the deadly asteroids. The Cosmic Space Patrol must act quickly to save their spacecraft and spare their lives. The spacecraft is equipped with photon torpedoes, hyperspace, shields, and flip control. The Cosmic Space Patrol is highly trained to handle this situation. Could you do as good a job as the Cosmic Space Patrol? How would you protect yourself if you were caught in a deadly asteroid belt? This is your big chance to fly throughout the dimensions of space and fend against asteroid boulders. The longer you survive, the more space hazards you'll encounter.

Yar's Revenge (Atari – 1981)
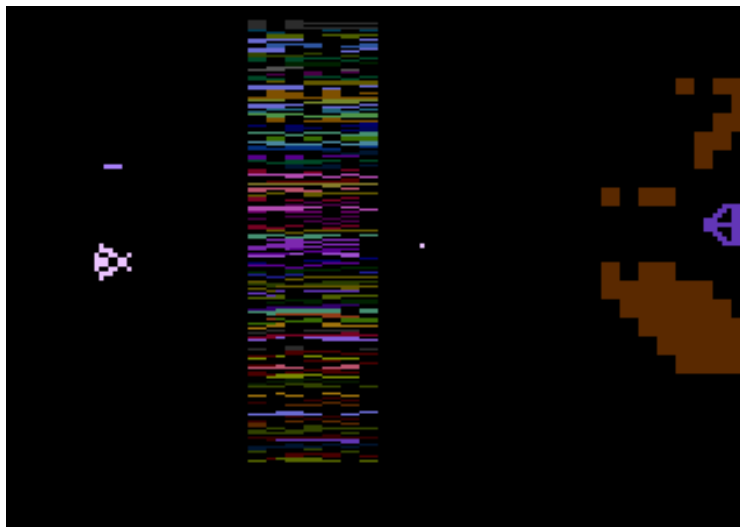
Figure 41 - Yar's Revenge for the Atari 2600

In 1981, Atari released the game *Yar's Revenge*. It was actually the best selling original title for the VCS. It started as licensed port of the Cinematronics game *Star Castle*. There were many changes to the Atari version, so much so, that the final game had very little resemblance to *Star Castle*. "In this game, the hero (a Yar) is an insect-like creature who must nibble or shoot through a barrier in order to fire his "Zorlon Cannon" into the breach and destroy the evil Qotile, which exists on the other side of the barrier. The Qotile can shoot at the Yar even if the barrier is undamaged, by turning into the "Swirl" - fortunately, the player is warned before the shot is fired, and he can retreat to a safe distance to dodge the enemy's energy blast. Also in the game is a safe area, "the neutral zone", where the pursuing enemy torpedo cannot harm him (although the Swirl can). The Yar cannot shoot from within the neutral zone."[20]

FROM THE MANUAL:

The primary objective of the game is to break a path through the shield, and destroy the Qotile with a blast from the Zorlon Cannon. The secondary objective is to score as many points as possible. See Figure below for an explanation of the objects on the playfield.

The shield is the red area in front of the Qotile base. It appears in one of two shapes, as an arch, or a shifting rectangle. The shield is made up of cells. The Yar scout can destroy these cells by firing at them with energy missiles, from any location on the playfield, or by devouring them on direct contact. (The Zorlon Cannon can also be used to destroy the cells, but this is a waste of a powerful weapon.) NOTE: To learn how to control the Yar and other aspects of the game play, be sure to read Section 3, USING THE CONTROLLERS.
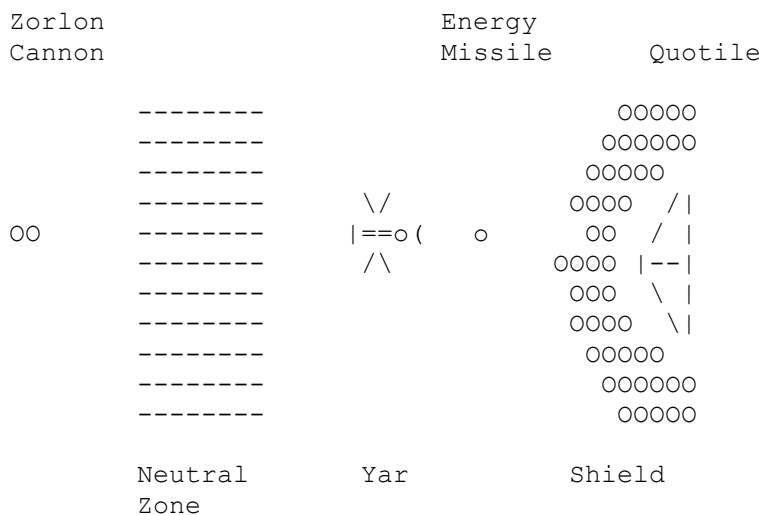
```
Zorlon                        Energy
Cannon                        Missile       Quotile


        --------                          OOOOO
        --------                         OOOOOO
        --------                          OOOOO
        --------         \/           OOOO   /|
OO      --------         |==o(    o      OO  / |
        --------         /\          OOOO |--|
        --------                      OOO  \ |
        --------                     OOOO   \|
        --------                     OOOOO
        --------                    OOOOOO
        --------                     OOOOO


        Neutral        Yar           Shield
        Zone


             Figure 1
```

Once a path has been cleared through the shield, the Zorlon Cannon must be used to destroy the Qotile. To call up the cannon, the Yar can either eat a cell, or run over the

---

[20] http://en.wikipedia.org/wiki/Yars%27_Revenge

Qotile. (See GAME VARIATIONS, Section 5, for more details on playing ULTIMATE YARS, Games 6 and 7.)

The Zorlon Cannon appears on the left side of the playfield, and moves in a direct line with the Yar. This means the Yar is in its line of fire. It is important therefore, to aim the cannon at the Qotile, fire it, and fly out of the way fast!

The Qotile shoots off two weapons: Destroyer Missiles and Swirls. The Destroyer Missiles come in a more or less constant stream, one at a time. The Yar must do his best to dodge them. Periodically, the Qotile transforms into a Swirl. This Swirl winds up and rushes off after the Yar. A Swirl can be destroyed with the Zorlon Cannon by hitting it either at its base location, or in mid-air. As a player's score increases, the Swirl becomes increasingly dangerous. (See Section 6, SCORING.)

The glittering path down the center of the screen is the Neutral Zone. This area will protect a Yar from Destroyer Missiles but not from Swirls.

While in the Neutral Zone, a Yar cannot fire any energy missiles of his own.

When a Yar is hit by a Destroyer Missile, a Swirl, or his own Zorlon Cannon, he dies. Each player has four Yars (turns) to play in a game. Additional Yars can be earned.

Solaris (Atari – 1986)

Figure 42 – Solaris

FROM THE MANUAL

Blast Those Cobra Ships, Mechnoids, and Raiders Before They Blast You!

The Zylons are back -- those spaceway sneaks, villains of Venus, Saturnian scoundrels! They're swarming through the galaxy in huge forces, attempting another takeover. They've got to go! And we need YOU to go get 'em.

But it's a hush-hush missions. If the Zylons guess you're onto them, you're a goner. So the official report says you're out to find the lost planet Solaris and rescue the Atarian Federation Pioneers stranded there. But if the Zylons reach Solaris before you do, they'll destroy it.

You've got to hyperwarp from quadrent to quadrent, facing vicious attackers such as Kogalon Star Pirates, Planet Destroyers, and Cobra Fleet. But don't worry -- your fighter, the StarCruiser, is specially outfitted with a Galactic Scanner and plenty of photon torpedoes. Just don't let the Zylons destroy a Federation Planet, or your quadrant mutates into a terrifying Red Zone.

Ready? Then hop into the StarCruiser, rev the engine, and go! And remember -- if anything flies your way, blast it!

The game was developed by Douglas Neubauer, it had you the player conducting space battles while having to refuel your spacecraft periodically, the objective of the game was to reach the planet Solaris and rescue its colonists. The game was notable for having some of the best graphics ever seen for the Atari 2600. Neubauer had gained a lot of experience before developing *Solaris* with the wonderful 8-bit computer games *Star Raiders*.

Notable Third party games – Activision and Imagic

Activision

The company Activision was founded in 1979 by former Atari game developers – David Crane, Larry Kaplan, Alan Miller and Bob Whitehead. These developers left in order to take advantage of the fact that the games they developed for Atari made millions but they did not receive credit or monetary compensation beyond their salaries and small bonus for their efforts.

At this point in home video game history game manufactures like Atari, Coleco, Mattel developed their own games for the home video game systems they sold. Activision was the first company to be formed as a third-party developer for a home video game system.

The developers that went on to form Activision were responsible for game cartridges for the 2600 that made up over 50% of Atari sales. This very fact caused Atari to file a lawsuit against Activision.[21] The lawsuit was eventually thrown out of court.

The key difference from Atari was that Activision featured their programmers on the game manuals and encouraged players to submit pictures of their high scores in order to get special patches.

Activision went on to create the following innovative and best-selling games:

Pitfall! (Activision – 1982)

Figure 43 - Pitfall!

This game sold over 2.6 million copies! It is one of the best selling games ever made for the Atari 2600. The game was created by David Crane, formerly of Atari.

Activision
Pitfall!
Instructions

Picture this! You are deep in the recesses of a forbidden jungle—an unforgiving place few explorers ever survive. But you've got courage, because you're with Pitfall Harry, the world famous jungle explorer and fortune hunter extraordinaire. The lure of hidden treasure draws you and Harry deeper and deeper into the bush. But, being a great explorer, you wouldn't think of starting such a difficult journey without reading this manual first—very carefully.

PITFALL! BASICS

---

[21] http://www.gamasutra.com/view/feature/129961/the_history_of_activision.php?print=1

The object of Pitfall! is to guide Harry through a maze of jungle scenes,
jumping over or avoiding many deadly dangers, and helping Harry grab the most
treasures in the shortest possible time.

"Its technical achievements included non-flickering, multicolored, animated sprites on a
system with notoriously primitive graphics hardware. *Pitfall!* was a massive success for
the 2600. It is considered to have been the best selling game ever made for the system,
with over 4 million copies of the game sold. Several ports were made for computer
systems (such as the Commodore 64, Atari 800 and TRS-80 Color Computer), as well as
for home consoles (such as the ColecoVision and the Intellivision)."[22]

Pressure Cooker (Activision – 1983)

Figure 44 - Pressure Cooker

This game is refreshingly different from your typical platform game or shoot'em up. In
this game the objective is to prepare hamburgers by grabbing and bouncing condiments
around the kitchen. You received points for completing orders accurately.

---

[22] http://en.wikipedia.org/wiki/Pitfall!

River Raid (Activision – 1982)

Figure 45 - River Raid

"River Raid is the grandfather of all vertical shooter games and is a great classic for the Atari 2600. You control a fighter plane going down the River of No Return and it is your goal to destroy enemy tankers, helicopters and jet, racking up as many points as possible. Avoid touching the sides of the river as that will crash your plane and watch your fuel as well, as an empty tank means will put a quick end to your flight. Fortunately there are many fuel tanks along the river, but they will become scarcer the farther along you fly. There are two types of river terrain you will encounter, light green and dark green. When the river is going through light green terrain it will go straight through without any land obstacles in the middle. Dark green terrain forces you to choose a side as the river will split in two and you cannot fly over the land. Periodically you will encounter bridges that must be destroyed in order for you to continue on your mission.

One very impressive aspect of River Raid is that the dark green terrain is quite varied, which breaks any monotony, the game could hold. The graphics are well designed and the playfield scrolls fluidly during the gameplay. The sounds consist of your planes engines, the shots you fire and explosions. And finally, the control is dead on. You can speed up or slow down your plane by pushing up or down on the joystick. By reaching a score of 15,000 or more you could join the River Raiders and receive a River Raiders patch. The maximum score that can be reached in the game is 1 million points."[23]

The game was programmed by Carol Shaw, a former Atari employee, who is said to the first female video game designers/programmers.

---

[23] http://www.atariage.com/software_page.html?SoftwareLabelID=409

Kaboom! (Activision – 1981)

Figure 46 - Kaboom!

Kaboom was designed and programmed by Larry Kaplan. The player uses the paddle control to catch bombs dropped by the "Mad Bomber". You have three buckets to use to catch a bomb, you gain buckets after every 1000 points and lose buckets when a bomb is not caught.

Barnstorming (Activision – 1982)

Figure 47 – Barnstorming

"Steve Cartwright created a number of games for Activision -- Megamania, Seaquest, and Frostbite (which would come in at number 11) -- but his top achievement is also his first: Barnstorming. Simple concept: Fly a classic aeroplane through barns. Avoid bird. Avoid

windmills. Avoid weathervanes. The whole game is a test of twitch skill set against that Activision "sunset." There's something about its breezy formula that is still winning today."[24]

This game was notable for having the first sunset in the horizon. This feat was impressive given the limited graphics capability of the Atari 2600, in fact, the code to produce the sunset was copied by other programmers in creating their own games.

H.E.R.O. (Activision – 1984)

Figure 48 - H.E.R.O.

Some gamers will tell you that H.E.R.O. was the best game made for the VCS, but due its release date it never received the attention it deserved. The game play involves guiding the protagonist Roderick Hero (also known as R. Hero to his friends) through deep dark mine shafts as he dodges flying critters, tentacles and moving walls.

The game play was fun and addictive. Try it.

FROM THE MANUAL

PERIL IS MY MIDDLE NAME

"Roderick Hero here (R.Hero to my associates), President and Chief Executive Officer of H.E.R.O., Inc., inviting you to join me in tackling a tricky maze of mine shafts rife with the kind of danger we daring types only dream of. Until now, that is! "

"Think of it: miners trapped in a mountain bubbling with lava rivers and magma deposits, mine shafts crawling with vile vermin, all lethal to the touch. Oh, it's just too awful ----- and too exciting! "

"I'm armed with equipment of my own invention. My Prop-pack carries me to even deeper depths, while I zap creepy critters with my Microlaser Beam. Dynamite demolishes walls that get in my way. If my supply lasts, that is. And my power. "

---

[24] http://retro.ign.com/articles/868/868566p1.html

"Did I mention the terrible tentacles that loom up out of lava? Or the massive walls that can crush me? What about the raft I ride on? Oh, never mind. Come along, and make yourself useful."

Chopper Command (Activision – 1982)

Figure 49 - Chopper Command

"Robert Whitehead's Chopper Command is sort of like a cross between Defender and Choplifter. You must protect rolling truck convoys at the bottom of the screen by blasting through waves of jet fighters and helicopters. The game is not easy, but a little practice gets you into the "zone." You will be able to control your fire, slamming it up into incoming enemies or fire while making an evasive turn -- essentially shooting one way while flying another. The game's appearance in pretty spartan, but the on-screen map and the Activision "sunset" are nice touches. This 1982 release is one of the better shooters for the 2600."

Dolphin (Activision – 1983)

Figure 50 – Activision

This game was rather unique in that it required that the player use both their ears and eyes to play.

"You are in control of a dolphin who needs your help in swimming through the ocean, through schools of seahorses, and being able to get the good currents (the arrows which are going in her direction) while dodging the hungry squid and the bad currents (the arrows going against her). You will hear a set of beeps before each school of seahorses appears, which is the dolphin's sonar. You must listen to the pitch to determine where you can swim through. The higher tones mean you must swim closer to the surface, while the lower tones mean you have to swim closer to the ocean floor.

Colliding with a seahorse slows your dolphin and makes it easier for the squid to catch her, but she can protect herself by guiding herself into currents that slow down the squid. Occasionally a seagull will fly by overhead, and if the dolphin touches it, she will gain the ability to drive the squid off by ramming into it.

Dolphin is a truly unique game and still remains fun to this day."[25]

---

[25] http://www.atariage.com/software_page.html?SoftwareLabelID=147

Robot Tank (Activision – 1983)

Figure 51 - Robot Tank

This game is obviously a take on Atari's video coin-op *Battlezone*. Activision created a fast-past and visually impressive version for the VCS.

Spider Fighter (Activision – 1983)

Figure 52 - Spider Fighter

This is a colorful and fast-paced shooter where you must defend you orchard against wave after wave of evil spiders. The spiders lay eggs and steal your fruit.

Imagic

Imagic was another third-party developer for the Atari 2600 started by a former Atari programmers Rob Fulop, Bill Grubb, Bob Smith, Mark Bradley and Denis Koble.

The bestselling titles were *Demon Attack*, *Atlantis* and *Cosmic Ark*.

Demon Attack (Imagic – 1982)

Figure 53 - Demon Attack

This game was developed for the Atari 2600 and other well-known game systems of the time. The premise of the game was that you were marooned on the planet Krybor and being attacked by demons from above. The demons attacked in waves with different weapons. The game was very similar to the Atari coin-op game Phoenix and prompted a lawsuit. *Demon Attack* game become quite a hit for the company.

Atlantis (Imagic– 1982)

Figure 54 – Atlantis

The game Atlantis is in the shooting game genre. The player controls the last defenses of the City of Atlantis against the Gorgon invaders. The player has to defend seven bases with one cannon at the center that shoots straight up and two cannons that shoot from the

left and right base diagonally. The player will always eventually lose, since trying to beat back the Gorgons is rather difficult, but at the end a tiny ship is seen rising from the rubble and fly's away, foreshadowing another encounter in the sequel game – Cosmic Ark.

Cosmic Ark (Imagic – 1982)

Figure 55 - Cosmic Ark

The sun of Alpha Ro is fading fast! Soon it will flicker out.
The Cosmic Ark races to save creatures from doomed planets in that solar system.
Meteor showers bombard the Ark, threatening its Atlantean crew - and planetary defense systems make this mission of mercy doubly treacherous! Time and energy slip away
- work fast or these defenseless little beasties will disappear for all time.

*****

The Cosmic Ark has traveled thousands of light years in order to preserve the many exotic species peculiar to the Alpha Ro solar system. When that sun novas, life on planets there will wither and die.

Nothing about this mission is easy. Heavy meteor activity throughout the system makes travel hazardous. And each planet has an automatic defense system that cannot be knocked out.

Can the Ark succeed in saving the many helpless creatures of Alpha Ro? Can the Cosmic Ark survive? How many planets can you reach?

The Atari 2600 was not the only video game machine in the time period we are covering but it was the most popular and well-known and many innovative games started as original Atari coin-op games that were later converted to video games for the system. The

Atari also had many third-party developers creating ground-breaking game ideas stretching the VCS to its physical limits.

The Rise of the Machines!

When microprocessors started to be used in constructing video arcade games engineers young and old started to get their hands on them to create their own projects at home. Many had the goal of making their own personal computers. In 1975 when Popular Electronics Magazine featured on their front cover Ed Roberts MITS Altair kit the personal computer revolution started to take off. The Altair was based on the 8080 microprocessor which at the time cost $300 but the kit was available to all adventurous engineers for about $400.

In California's Silicon Valley a new computer club got started, the Homebrew Computer Club. The club members at first met in a garage to discuss their experience with trying to build their own Altair computer or other computer projects but the group quickly grew into hundreds of members and had to move into an auditorium. Many members of the club went to start or work for many computer companies in the Silicon Valley.

Many great machines were built and sold during this time period. We will focus on my favorite[26] the Apple computer. The honors for the best selling machine during the retro time period will go to another machine – the Commodore 64 (C64), but Apple was the first to sell one million machines. The C64 outdid the Apple in three key categories – price, graphics and sound. Atari also got into the personal computer business with technically very good machines the Atari 400 and Atari 800. But, we only have time for one – the Apple.

Apple

http://apple2history.org/museum/computers_apple1/apple1b.html

The inexpensive cost microprocessors like the 6502 provided many engineers with the opportunity at building their own computer systems from kits or parts. Steve Wozniak (see section on *Breakout*) spent many years building computer systems using only paper and pen since the components for real machines were unaffordable for the average student. By

---

[26] The first machine I ever purchased was an Apple IIe. It still sits on the very desk I am writing this book on.

1975 as member of the Homebrew Computer club he exchanged ideas and chips with the members of the club.[27]

Most of the members of the Homebrew club were busy obtaining what was then "the machine" to have the Altair. The Altair was based on the Intel 8080 http://apple2history.org/museum/computers_apple1/altair8800.htmlmicroprocessor but that chip was $300 as compared to the newly released $25 6502 chip, so Woz decided to try his hand building a computer using the 6502. Woz decided on two key improvements for his machine, one was to use a keyboard for data entry rather using what was common to engineers at the time - front panel switches and his machine would output to a video terminal rather than the use of teletype. The keyboard was probably influenced by his job at HP designing calculators and Woz learned how to work with video while he was hanging around Atari building and playing with side projects. The Homebrew Computer Club was very impressed with his efforts at building a machine. Steve Jobs was a friend and occasional visitor who provided suggestions that helped shape the machine as a product to be sold. It was Jobs idea to sell Woz's creation as a kit to fellow members of the Club to make money. By 1976 they formed the Apple Computer Company. Woz's next creation – the Apple II really had all the functionality and ease of use that consumers required in personal computers. The machine really took off and became a big seller for the company.

Figure 56 - Bill Gates (age 27) in 1982

I should mention that a young man by the name of Bill Gates made a name for himself among computer hobbyists by writing a BASIC interpreter for the Altair. Bill and his partner Paul Allen went on to co-find the company Microsoft. The rest as they say is history.

---

[27] There are many sources for this section but I highly recommend http://apple2history.org for more details on Apple history.

Top Computer Games

Let me admit two things – compared to other computer systems of the time the Apple II sound and graphics were dismal. Despite its shortcoming the machine was the premier machine for game designers to develop games for in the 1980's. Many popular coin-op arcade games were ported over to the Apple IIe so in this list we cover only the notable and popular games. The C64 eventually went on to sell more machines. The Commodore 64 had superior sound and graphics.

As more people started to purchase home computers and home game consoles game developers started to invest their efforts to create games for these systems. Budding game designers only required a computer, some development tools and their imagination. Many a young boy and girl started their game programming careers not to long after their got their first home computer.

We will feature and discuss games that could not have been done for the arcades or home video systems since the games required a keyboard, in addition we will examine games that introduced new concepts and new features on home computer systems.

The designer's of the mainframe game Zork formed a company in 1979 to create interactive fiction for computers. The name of the company was Infocom and they developed many interactive fiction games. An interactive fiction (IF) game simulates an environment and has the player enter text commands (pick up the apple) to control the action in the game. The player reads text and responds with short sentences while trying to achieve a quest. Infocom developed the Zork series, Planetfall, Leather Goddesses of Phobos, Hitchhiker's Guide to the Galaxy and many more. These types of game reached their peak in popularity in the mid-1980's. What remains true today for all budding game designers is that the story elements of worlds built in these text adventures are still "must have" ingredients in any game today. You must play one or more to see the fantastic worlds and stories created with text.

In 1980 the company released the game Zork I: The Great Underground Empire on many computer systems.
"The game takes place in the Zork calendar year 948 GUE (although the passage of time is not notable in gameplay). The player steps into the deliberately vague role of an "adventurer". The game begins near a White House in a small, self-contained area. Although the player is given little instruction, the house provides an obvious point of interest. When the player enters the house, it yields a number of intriguing objects: an ancient brass lantern, an empty trophy case, an intricately engraved sword, etc. Beneath the rug a trap door leads down into a dark dungeon. But what initially appears to be a dungeon is actually one of several entrances to a vast subterranean land--the Great Underground Empire. The player soon encounters dangerous creatures, including deadly

grues, an axe-wielding troll, a giant Cyclops and a nimble-fingered thief. The ultimate goal of *Zork I* is to collect the Twenty One Treasures of Zork and install them in the trophy case. Finding the treasures requires solving a variety of puzzles such as the navigation of two brutal mazes and some intricate manipulations at Flood Control Dam #3. Placing all of the treasures into the trophy case scores the player 350 points and grants the rank of "Master Adventurer." An ancient map with further instructions then magically appears in the trophy case. These instructions provide access to a stone barrow. The entrance to the barrow is the end of *Zork I* and the beginning of *Zork II*."

Playing the game:

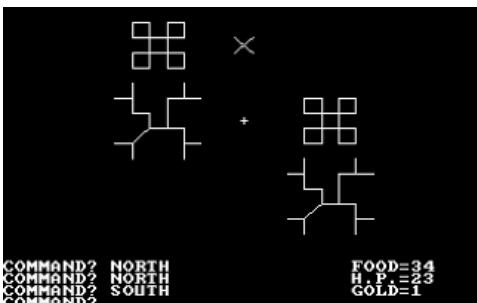**Figure 57 – Playing Zork I**

In 1979 Roberta William's husband was a programmer for IBM who came across the mainframe version of Colossal Caves and showed it to his wife. She enjoyed playing the game and others like it on their Apple II computer. She decided she could make her own adventure game and take advantage of the fact that computers allowed you to display images. Her first game as game

designer with her husband as the programmer was *Mystery House (1980)*. The game will probably be in any top 100 computer games list since it merged the art of interactive fiction with images. The game itself is a murder mystery, where you the player are stuck in a house with seven people any one of whom could be the murderer. The game was a hit selling about 15,000 copies and earning the Williams enough money to move out of Los Angeles and into Sierra Nevada. The company was named On-line systems later renamed Sierra On-Line, later Sierra Entertainment. The company suffered the classic failure of many start-ups of expanding too soon into too many platforms. By mid-1984 there were close to bankruptcy but were saved when contacted by IBM to showcase their new computer the PCjr. Roberta Williams designed a game with all the classic elements of a fantasy – "a knight would have to save a kingdom in distress by recovering three lost treasures" The game had animated color graphics and a pseudo 3D-perspective. The player could use the keyboard to move the game character around to examine objects and control his movement. "A game like this had never been made before.." The game *King's Quest* (1984) was the game that really made the company take-off. The game had many sequels. Al Lowe developed the game *Leisure Suit Larry in the Land of the Lounge Lizards* (1987)for Sierra On-Line. The plot for the game required the player to take on the role of Larry Laffer a 40-year old virgin. The game went on to critical and commercial success being named "Best Fantasy, Role Playing or Adventure Game of 1987" and selling over 250,000 copies.

One of the first examples of a computer role-playing game was Akalabeth: World of Doom. This game was released in 1980 by Richard Garriott. The game was developed by Garriott for the Apple II using BASIC. The game attempts to combine the storytelling and game play of Dungeons & Dragons and fantasy books of J.R.R. Tolkien. The player embarks on the quest to kill increasingly strong monsters that hide somewhere in dungeon.

63

Garriott started to learn how to program in High School when he developed an interest in creating his own fantasy games. He got the nickname Lord British from other students because they thought he had a slight British accent. In fact, he was born in Cambridge, England but raised in Texas. His first published game was the Akalabeth. He later developed the very popular *Ultima* series of computer games. His contribution to gaming as a premier developer leads to his induction to the Academy of Interactive Arts and Sciences' Hall of Fame.

The game developer and distributing company Electronic Arts (EA) was started by Trip Hawkins. Hawkins was the Director of Strategy and Marketing at Apple when he ventured out to start his own company in 1982 – a video publishing company. Hawkins distinguished his company from the rest by creating attractive packaging for his games[28] that featured the game designers and programmers on the cover. In addition, the company shared the profits with the developers. EA soon attracted some of the best game designers and programmers in the young industry.

The game *Pinball Construction Set* (1983) was developed by Bill Budge. The game was ported to many popular computers such as the Apple and Commodore 64. The game "created a new genre of computer games – the 'builder' or 'construction set' class of games." The user builds their very own pinball arcade machine by dragging and dropping bumpers, flippers, and other pinball items to the table. The game play on the virtual pinball machine follows the rules of gravity and physics (which were also configurable). The game was a huge seller and lead to other popular construction games such as *Music Construction Set*, *Adventure Construction Set* and *Racing Destruction Set*. This game will always be on the top list of 'innovative games'.

---
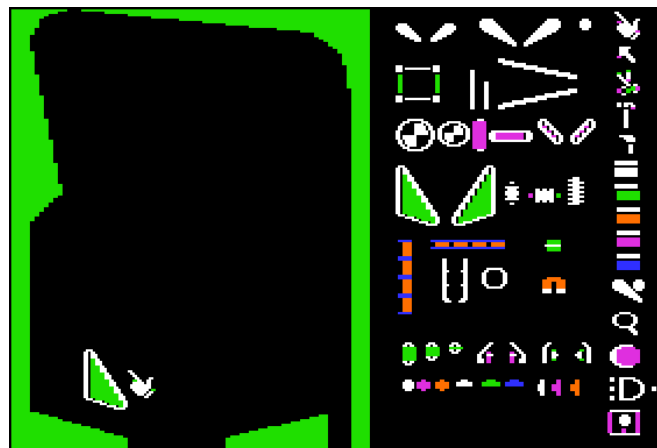
[28] This was a time when many games were packaged in Ziploc bags with type written instructions, Electronic Arts games were packaged like album covers.
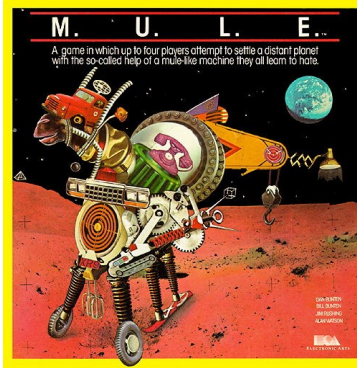
Another seminal game released by EA was the game *M.U.L..E* (1983) by Dan (Danielle) Bunten. The game would probably be considered a strategy game today. The game was ported to many computer systems (even the Nintendo Entertainment System). "Set on the fictional planet Irata (which is *Atari* backwards), the game is an exercise in supply and demand economics involving competition between four players, with computer opponents automatically filling in for any missing players. Players are provided with several different choices for the race of their colonist, providing different advantages and disadvantages which can be paired to their respective strategies. To win, players not only compete against each other to amass the largest amount of wealth, but must also cooperate for the survival of the colony." The game was one of the earliest examples of multiplayer game concept. Bunten followed up this game with many popular games including *The Seven Cities of Gold* (1984), *Heart of Africa* (1985), Robot Wars (the first modem game) and many more.
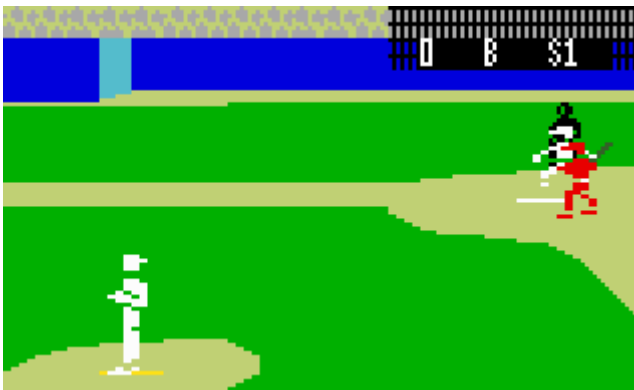
The sports genre started with a mainframe game named BASBAL (1971). The basis of the game was the use of real baseball statistics to determine what would happen next. The statistical nature of the game was the key component in graphics version of baseball games (as well as other sports games). The first notable baseball game was actually done on the Intellivision console in 1983 – Intellivision World Series Baseball. The game was notable for using different camera angles, being in 3D, and having a inset screen "to show a runner taking his lead off of first base." The game's popularity was affected by the great video game crash of 1983. Games that fall into the sports genre are Baseball games, Football games, Basketball games, Soccer games.

One of the first games in the genre that EA will later dominate *Dr. J and Larry Bird Go One on One* (1983) was a computer basketball game. The player can play as one the two most popular basketball players of the time Julius Erving (Dr. J) or Larry Bird. The graphics as shown here for the Apple II was rather poor compared to today's but the

65

game can be noted for featuring popular players, the animation for the time, moves that were unique to each 'real' player. Today EA is well-known for its basketball, baseball, football, golf soccer games (Madden NFL anyone?)
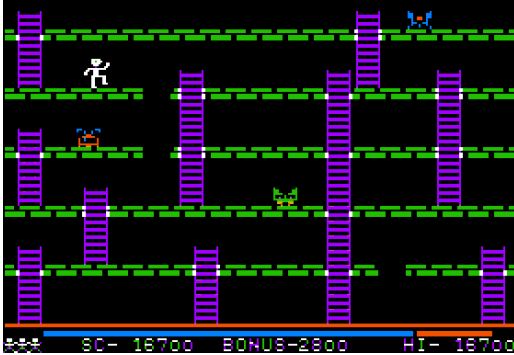
The game *Apple Panic* (1981) is a platform released for the Apple II that is based on the seminal platform game Space Panic (1980). A platform game is a game where the main character controlled by you the player "jumps to and from suspended platforms or over obstacles." At one time many games developed for computers, video machines fell into this genre. These games came before Donkey Kong a rather well-known platformer that set the tempo for story and fun.

The first stealth-based was *Castle Wolfenstein* (1981) by Muse Software. The game was initially released on the Apple II. The main character had to move about a castle searching for secret war plans. The game had a top-down perspective, but the characters were "seen upright like in a side-scroller." The player had to sneak past guards, impersonate Nazi guards and kill enemies when necessary. The notable feature of the game was the use of digitized voices for some of the game characters, where a Nazi soldier would utter "Kommen Sie!" This genre would gain popularity in the 1990's with games such as Metal Gear for the NES.

The Great Arcade Golden Age

The early 1980's was a great time for the Video arcade business as well. Arcade machines were everywhere – barber shops, restaurants, and some even showed up in doctor's office.

There were many great games designed during the period known as the "Golden Age of Arcades." I should also mention that Atari was not the only company churning out the hits, there was Williams Electronics (Defender), Namco (Pac-Man), Midway with Namco (Ms. Pac-man), (Stern Electronics (e.g. Berzerk), and many more.

There were many designer/programmers that will always be remembered for the one great innovative game or the series of hits generated (note, usually when a designer is

very prolific they usually have some duds in the mix). This was a time period when the game designer and programmer were one in the same. Today the game designer and programmer works with a team of people to develop a game for a video game console or computer.

The most popular video arcade games

Here is list of the most popular coin-op video arcade games ever made. Note the time period. This list will probably never change. Why? Games today are created and released to be played on home video consoles not coin-op arcade machines. So the days of millions of kids young and old spending the day dropping hard earned quarters into the coin-op machines are long gone.

1. Pac-Man (Midway/Namco - 1980)
2. Galaga (Namco - 1981)
3. Donkey Kong (Nintendo - 1981)
4. Star Wars (Atari - 1983)
5. Ms. Pac-Man (Midway – 1981)
6. Dig Dug (Atari – 1982)
7. Asteroids (Atari – 1979)
8. Defender (Williams – 1980)
9. Tron (Williams – 1980)
10. Tempest (Atari – 1980)
11. Centipede (Atari – 1980)

Pac-Man (Midway/Namco – 1980)

This blockbuster of a game was developed by Toru Iwatani for the Japanese arcade company Namco. Iwatani goal in developing the game was to develop a game that females would be comfortable going to the arcade to play. An apocryphal story has it that the idea for the image of Pac-Man came to the designer Iwatani when he removed a slice from a pizza pie and saw the image that was left. At the time of its release most video arcade games were space shooters such as Space Invaders. The game was distributed in the U.S. by Midway. The game was originally named Puck-Man but was changed for fear what mischievous players could do with a magic marker to the game cabinet.   The game started the new game genre of "maze-chase" games. Its

popularity was tremendous earning it a place in the Smithsonian. The ghosts, Blinky, Pinky, Inky and Clyde were well-known to any who played and each displayed a distinct and unique personality during game play.

Atari released a version of the game for the Atari 2600 in 1982. Atari made two fatal mistakes in their version of the game, they rushed the release of the game that lead to poor graphics (annoying flicker) and they produced more cartridges than Atari machines ever sold at the time in anticipation that customers would buy the console in order to play the game. It did sell over 7 million but many millions of cartridges went unsold.

Galaga (Namco – 1981)

The game Galaga is another Namco masterpiece. It was also released by Midway in the U.S. It has become an example of the best elements of a space shooter. It has great sound and explosions, a challenge stage and displayed to the player the number of hits and misses, so you can either brag to your friends about your accuracy level or hide your head in shame if you found out your were shooting mostly at air. The enemy appeared as colorful insects. The testimony to its long lasting playability is the fact that Namco frequently adds the game to its top game anthology, you can today play it on XBox Live arcade and many arcade centers may still feature a Galaga cabinet for you to store your quarters away in. The musical tune that starts each level is memorable and will always bring a smile when you hear it (when you are not playing of course since the only thing to do is shoot the alien insect scum down!).
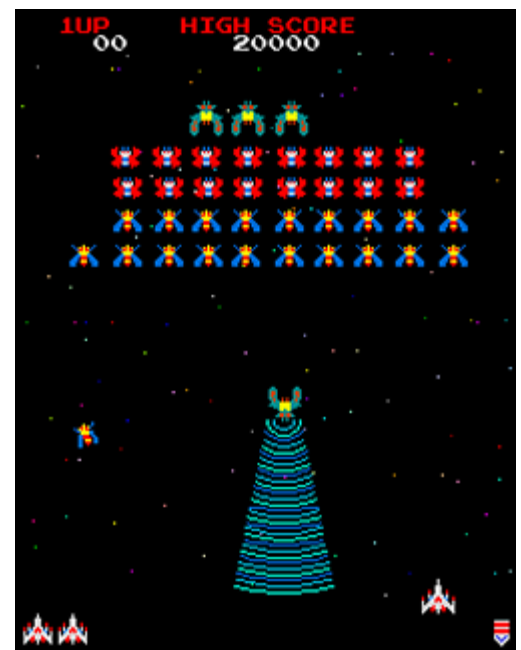
Donkey Kong (Nintendo – 1981)

This game was notable for several things – Nintendo's first successful coin-op machine, starting the wonderful game design career of Shigeru Miyamoto, starting the franchise of wonderful characters starting with Mario and Donkey Kong, and of course being a real joy to play.

The protagonist Jumpman (later renamed Mario supposedly after a less than friendly NJ warehouse guy who collected the rent on the warehouse holding all those dust collecting unsold *Rader Scope* cabinets),

68

attempts to rescue the damsel in distress, Pauline, from the giant ape who keeps grabbing her away from our hero. The story line is thin but the game used graphics to give the characters personality and cut-scenes that helped move the story along. "Donkey Kong is the first example of a complete narrative told in video game form." Donkey Kong was not the first platformer but one of the best in the genre.

Star Wars (Atari – 1983)

The game Star Wars is based on the similar named movie. The game is a first-person space simulator. You get to relive the last scene of the movie - where our hero Luke Skywalker goes up against the Death Star. The game features digitized samples of voices from the movie. The game uses vector graphics to produce a lively 3-D space environment. The game was very popular and one of the few examples of great games being inspired by great movies.
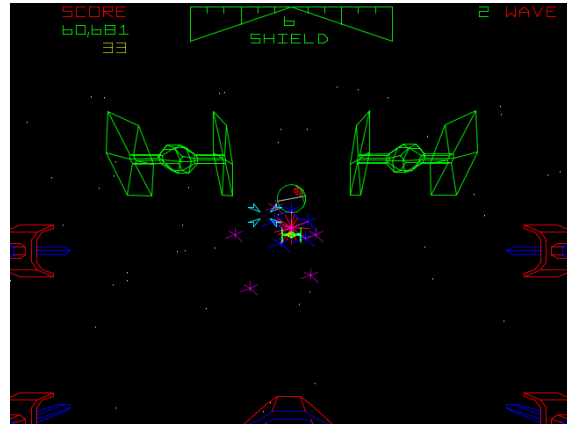
## Ms. Pac-Man (Midway – 1981)

It is hard to believe that any coin-op could sell more machines than *Pac-Man* but there was one – *Ms. Pac-Man*. The game was actually an unauthorized version of the game Pac-Man created by the company General Computer Corporation (GCC). The company was started by two college students Doug Macrae and Kevin Curran who made money on the side with the popular coin-op machine *Missile Command*. In order to extend the life of the cabinet they reversed-engineered the game and created a 'souped up" version of the game – Super Missile Attack. They were able to sell their enhancement board to other distributers who wanted to extend the life and money making capabilities of their Missile Command cabinets. Atari sued the company but the lawsuit was settled by allowing the company to develop games for Atari 2600. GCC managed to create half the games for the Atari next game console the Atari 5200. The company took on the task to create an enhancement board for Pac-Man and came up with Ms. Pac-Man. They actually made many improvements to the original Pac-Man game – a female protagonist, new maze designs, the ghosts movements were more random which made the game more challenging. GCC showed the game to Midway who were looking for a follow-up to the successful Pac-Man. Midway released the game and it went on to sell more machines

than Pac-Man. The rights were eventually given to Namco in order to avoid legal problems over the creation and release of an unauthorized version of Pac-Man.

Dig Dug (Atari/Namco – 1982)

This was another Namco classic distributed in the U.S. by Atari. The game features Dig Dug (controlled by you the player) trying to rid the place of underground monsters dwelling in the earth. Dig Dug goes up against Pookas who wear yellow googles and green fire breathing dragons named Fygars . Our hero is digs tunnels and either inflates a monster until it blows up or drops a rock on them.
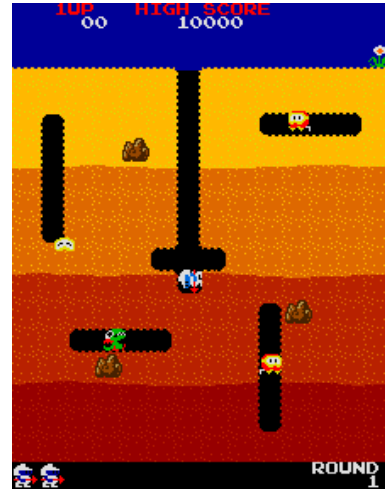
Asteroids (Atari – 1979)

Playing Asteroids today it is hard to believe how popular and influential the game was for its time. It was designed and programmed by two of Atari's best arcade game designers, Lyle Rains and Ed Logg. The goal of the game is quite simple – destroy the asteroids and those annoying flying saucers that periodically come out to shoot you down. This game went on to be one of Atari's best selling arcade games.

The game has been ported on to many consoles and computer systems.

Defender (Williams – 1980)

Defender was thought to be too complicated to have any success in the arcade world. The game required that the user use five buttons and joystick. The game was developed by the famous game designer and programmer Eugene Jarvis. The game was in fact difficult to master with the average new player lasting less than a minute as they tried to master their mission of flying their spacecraft along a mountainous landscape while trying to destroy the aliens and protect the humanoids on the planet from being taken captive. The game developed a strong following and lasted at the arcades in to the 80's.
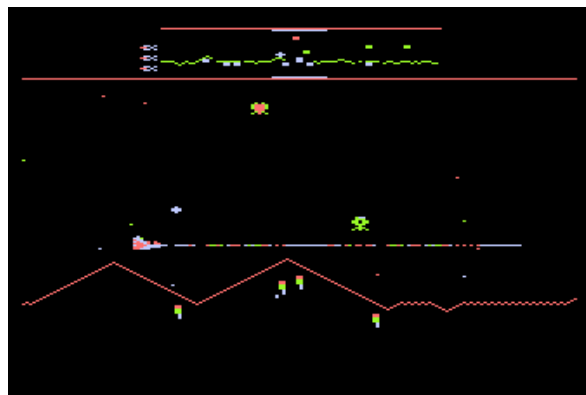
Tron (Bally/Midway – 1980)

This game was based on the popular movie Tron. The player plays four distinct games per level – Lightcycles, Grid Bugs, Tanks, and the MPC Cone. The player must complete all four games before advancing to the next level of the game. The game actually was more financially successful than the initial movie release.

Tempest (Atari – 1980)

"***Tempest*** is an arcade game by Atari Inc., originally designed and programmed by David Theurer. Released in October, 1981, it was fairly popular and had several ports and sequels. The game is also notable for being the first video game with a selectable level of difficulty (determined by the initial starting level). The game is a tube shooter, a type of shoot 'em up where the environment is fixed, but is viewed from a three-dimensional perspective." In Tempest the player goes against five enemies, Flippers, Tankers, Spikers, Fuseballs, and Pulsars. The goal was to survive and score as many points as possible.
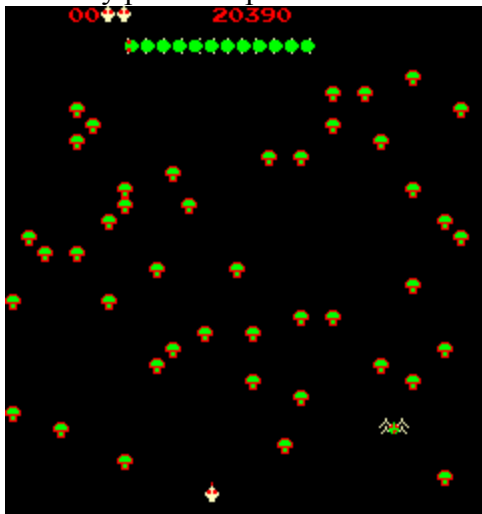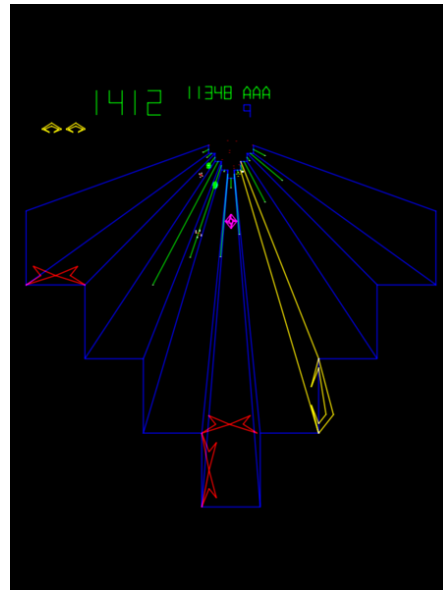
Centipede (Atari – 1980)

Centipede was conceived and programmed by one of the only female programmers at Atari in 1980 – Dona Bailey. Ed Logg was the other key developer for the game. The game was one of the few games that appealed to both sexes. The player starts at the bottom of the screen and tries to finish off a

71

centipede and swarms of insects before they get to the bottom of the screen. The game was very popular and sold many machines for Atari.

The popularity of video games was also accompanied by concern and discussion on how it was affecting our youth.[29]
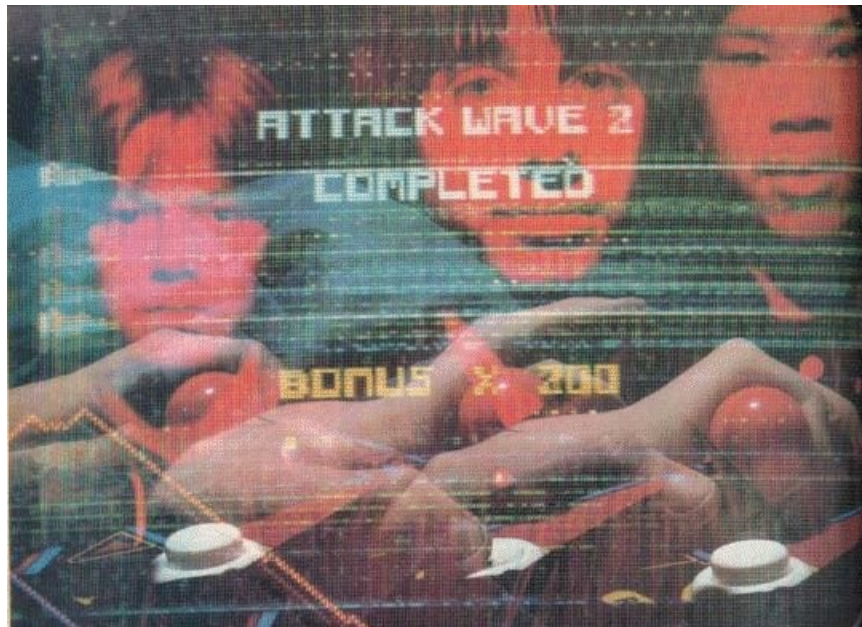
Figure 58 - Time Magazine Cover Story

In 1982 Time magazine did a cover story on the topic, highlighting the popular games, the number of quarters being spent by the young and the old in video arcades and the growing controversy. In 1982, the video arcade business was a $5 billion industry and that figure did not include the money being spent on home video consoles or computer games. Arcade machines were not just confined to arcades but were showing up at your local pizzeria, hair salon and dentist office. They were everywhere. They were making more money than the casinos in Nevada, twice as much as the movie industry and three times more that the combined revenues of baseball, basketball and football. The industry was huge! Parents were not only concerned about the money their kids were spending in on games but the time. There was also talk about what the games were teaching them – to be aggressive, violent and social retards. This debate still continues today but the focus is no longer on video arcade games but games developed for the Xbox, Playstation and Nintendo consoles.

---

[29] http://gamearchive.com/General/Articles/ClassicNews/1982/TimePg50.htm

The games created during this period were highly varied and innovative games. You had shoot'em ups, racing and fighting games of course, but you also had Q-Bert, Qix, and so many other strange and wonderful games that seem to emerge from the minds or groups of folks that would try anything that seemed different and fun.

Nintendo enters the video game business
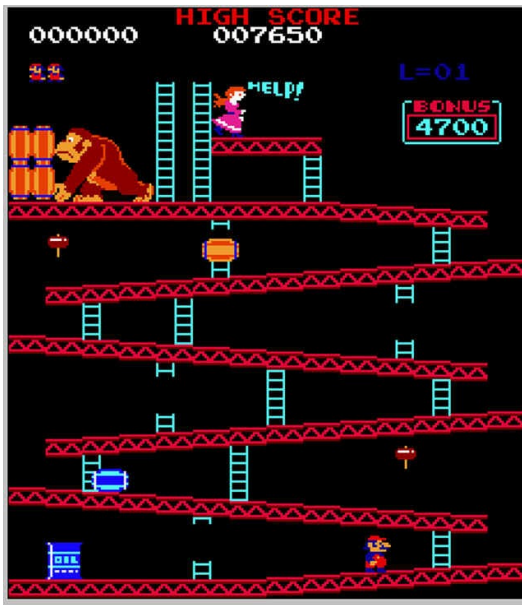
Nintendo made several early attempts to enter the video arcade market. All of its early games were not very popular. There was Sheriff (1979), Space Fever (1979) a Space Invaders clone, Space Firebird (1980) a space shooter action, and on some folks top ten worse video game – Radar Scope (1980). Radar Scope was Space Invaders meets Galaxian. Thousands of game cabinets just sat in a warehouse in New Jersey. The president of Nintendo of America, Minoru Arakawa, requested that a new game be developed that could use the cabinet hardware and just re-paint the cabinet. A young designer working for Nintendo was given the assignment – Shigueru Miyamoto. This was Miyamoto's first game assignment for Nintendo. He decided not to tweak the current game but to come up with something new. He developed a story about a character (representing the player) named Jumpman (later re-named Mario) that must rescue a damsel in distress, Pauline, from the big bad ape named Donkey Kong. The game had a story, cut-scenes (Donkey Kong moving the next level), and multiple stages. In 1981 the game went on to be Nintendo's first huge arcade hit. The game also is made for various game consoles and computer systems.

The Infamous and Dreadful Games

We just went through all many great games made for the VCS by Atari, Activision and Imagic. The games had all the limitations of an Atari 2600 with respect to sound and blocky graphics but the reason you should try to play each of the games we featured in this section is to determine why they were fun then and are still fun to play today. If the games were first released at the video arcades then I encourage you to try that version first rather than seeing a version limited by the technology of a game console or home computer.

If the games you design can capture the same elements of fun contained in these games then you will have a hit.

Let's not forget that there were many games made that were just dumb ideas, poor ports of arcade machines, or just dreadful with respect to playability. The fact remains true today - more bad games are made than hits. I discuss several of them in this section. Why? I think you should try these games[30] in order to determine why they failed.

Custer's Revenge - 1982
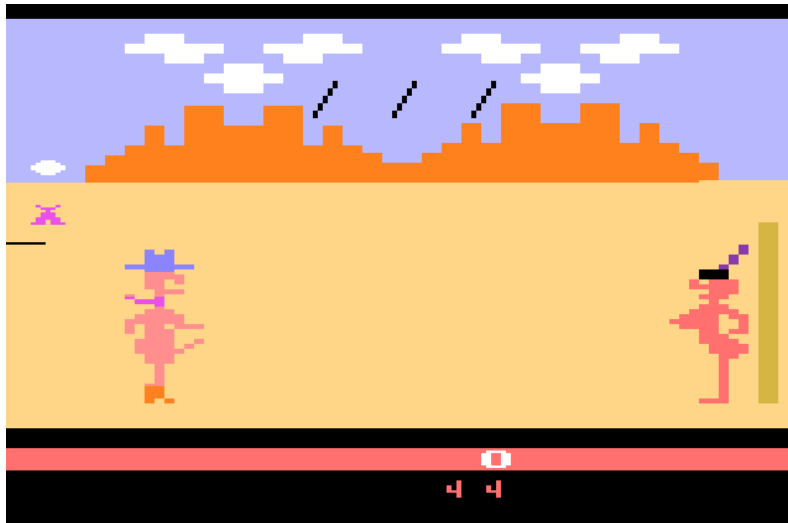
Figure 59 - Custer's Revenge - sick game.

This was an awful idea by game designers who must have wanted to create something pornographic and sick. This is the type of game developed by boys masquerading as men who just hate their Moms, sisters and or any women in their lives. The game idea was stupid – Custer tries to get across the screen dodging arrows to score with the maiden. You the player are supposed to pretend that rape is fun! The game play and idea just plain sucked.

Custer's Revenge is not the last game to try a dumb idea for a game. A lot of thought and consideration must be given by the game designer on what people will consider fun. This isn't Grand Theft Auto we are talking about. The latest lame so-called game was Columbine. It was rated one of the worse game ever made. Games can be scary, somewhat offensive and fun but games based on rape, or re-living tragic events is not fun. Well balanced people see the difference between a game like Grand Theft Auto and Cdolumbine. We can only wonder at what the game designers of these games are smoking.

E.T. the Extra-Terrestrial - 1982

---

[30] Don't bother trying to play the infamous games – they have no redeeming value.

Figure 60 - E.T. - extra terrible

The game E.T. was based on the very successful movie with the same name. Atari wanted to take advantage of the game popularity and rushed a game out for Christmas 1982. The game programmer was only given several weeks to design and complete an entire game and painfully it shows in the game play.

This game is notorious or infamous for representing the failure of Atari in maintaining standards in many of the games they released for the VCS. The marketing folks thought they could sell anything as long as it came packaged with the Atari label in a game cartridge. They found out for the Christmas of 1982 that it was not true. Millions of the unsold game cartridges were hauled in trucks to a New Mexico desert and just buried and covered with concrete.

This marked the beginning of the end of Atari as a video game powerhouse and the demise of the entire video game industry (for a time). Atari failed to meet it earnings for that last quarter of 1982 and this started a slide not only for Atari but for the entire industry. The following year Atari posted a loss of over $500 million.

The video game collapse of 1983

By 1983 and 1984 the market was just saturated. Too many arcades, too many consoles, manufactures building too much inventory anticipating the massive growth would continue. Too many mediocre to dreadfully awful games. But the good times did not continue. The quarters being dropped into coin-op machines slowed down and the game buying public slowed down purchases of game cartridges. At this time point in time Atari was the biggest player in the industry and fell the hardest and the fastest.

Why?

There were just too many game consoles and game manufactures all competing for the same market. In Atari's case there were over fifty companies creating games for the VCS. The games ranged from the few gems from companies like Activision to the hundreds that ranged from boring to just being in bad taste. The home console business also had serious competition – home computer systems. "By 1984, Commodore was selling 300,000 computers a month, and there are 4 million Commodore computers in use around the world."

Third Generation – Nintendo hits the reset button

In 1985 Nintendo released a new game console in the US the Nintendo Entertainment System (NES). In order to get stores to stock their product and people to take a look they packaged it up a less of a home video game system but more of an entertainment system. The system came with a Zapper light gun, and robot and unique controllers (see picture) rather than the typical joystick or game paddle. The machine used the 6502 for processing muscle but accompanied it with another processor a PPU (picture processing unit) to handle the graphics.

The head of Nintendo, Hiroshi Yamauchi, concluded that the downfall of Atari was based on the fact that anyone could and did develop and sell cartridges for the VCS. He decided that game manufactures must obtain a software license to create games for the NES. In addition, Nintendo decided that it would manufacture the actual game cartridges for all third-party developers. A game required the Nintendo "Seal of Quality" in order to be sold.

Nintendo initially had a difficult time convincing anyone to distribute or stock their new game console since everyone was still hemorrhaging from their losses sustained during the video game crash. They decided to test market the product in the toughest market - New York City. The game console was destined for success being packaged with one of the best games ever created for a console – Super Mario Bros. The game was designed by the same designer that started the ball rolling for Nintendo in the video arcade business - Shigueru Miyamoto. The game was a platform game that had Mario trying to save Princess Peach (or Princess Toadstool) of the Mushroom Kingdom from the evil king of the Koppas – Broswer. The game went on to be the best selling game of all time selling over 40 million copies to date. Another notable fact is that Miyamoto went on to design some more game classics and is without argument one of the best game designer that ever lived.

The only noteworthy 8-bit contender to the NES was the Sega Genesis manufactured by Sega. It was released by in 1986 and had to go mano a mano with the NES. Who doesn't remember Sonic the Hedgehog.

Best NES Games

Super Mario Bros. (1985)

"*Super Mario Bros.* is a platform game developed by Nintendo in late 1985 and published for the Nintendo Entertainment System. In *Super Mario Bros.*, Italian plumber Mario must save Princess Peach (Princess Toadstool in the US version) of the Mushroom Kingdom from the evil Bowser, king of the Koopas. In two-player mode, Mario is aided in his quest by his brother, Luigi. In order to save Princess Peach, the Mario Bros. must conquer the eight worlds that comprise the Mushroom Kingdom. Mario (or Luigi) must make his way to the castle in each world and defeat one of the Bowser's evil minions, thereby taking control of that world. In order to reach each castle Mario or Luigi must battle through three "sub-worlds" by either destroying or avoiding Bowser's henchmen. If Mario or Luigi successfully fights his way through the castle and defeats the evil minions, a Mushroom Retainer (later called Toad), is freed. Inside the eighth castle, the Mario Bros. will find Princess Peach." This game went on to sell over 40 million copies!

Super Mario Bros. 3 (1990)

The third game in the Super Mario Bros. series for the NES, and regarded by many as not only the best game ever made for the NES but the best 2D platformer. It has the same fun game elements as the original game released and packaged with the original NES – features the company mascot, Mario and his brother Luigi. It is a 2D platformer where the player jumps around collecting coins and points and defeating his enemies. This game featured many new elements added by the game designer Shigeru Miyamoto, a map screen, mini-games and new power-ups. The game went on to

77

sell over 18 million cartridges!

This game should be played and analyzed by any budding game designer.

The Legend of Zelda (1987)

This game is also a Shigeru Miyamoto designed game. The game play was quite different from any games Americans players were accustomed to playing at the time. The game takes place in a fantasy world of Hyrule where the main character, a young boy named Link must rescue Princess Zelda from the evil Ganon. Link must find eight fragments of the artifact the Triforce of Wisdom. This requires that Link traverse the game world buying items from merchants and exchanging information with gamblers and old ladies in order to continue his quest. Link must find nine underground dungeons, where each one is a maze requiring that Link battle creatures. Nintendo had many concerns how receptive American gamers would be given that it did not strictly fall into any genre known at the time. The game manual provided many hints and tips and Nintendo even made available a phone number user's could call if they needed more guidance. The game is notable for being nonlinear and favoring game completion rather than a high score.

Mike Tyson's Punch-Out! (1987)

This game usually comes up in any top ten NES game list. The game was based on a coin-op game Punch-Out! created by Nintendo in 1985. "The game features a boxer named Little Mac working his way up the professional boxing circuits, facing a series of colorful, fictional boxers, leading to a final fight with real-life boxer, and then World Heavyweight Champion, Mike Tyson."
Later versions of the game replaced Mike Tyson (who had lost his crown) with a fictional boxer by the name of Mr. Dream. The game was re-titled as just Punch-Out!

Metal Gear (1987)

 Metal Gear was designed by Hideo Kojima for Konami. Kojima is known as an "influential and innovative video game" designer responsible for many great games. Metal Gear is a stealth game that is notable for popularizing the

78
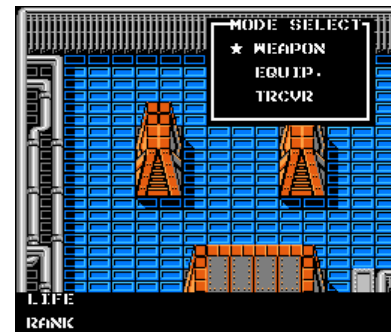
stealth game genre for the NES. In this type of game the user playing the character of Solid Snake goes through the game avoiding visual and direct contact with guards. This game and the series of games developed in the franchise remain popular today.

Metroid (1986)

Metroid is another Nintendo classic designed by Gunpei Yokoi, the creator of the Game Boy. The games has the side-scrolling aspects of the Super Mario Bros. and the item collecting and non-linear nature of The Legend of Zelda. The game became famous and notable because the player does not discover until the end that the lead character is female – Samus Aran the bounty hunter. The game combines the gameplay of a shooter, platformer and adventure game.

Contra (1988)

Contra is another Konami arcade game ported over to the NES in 1988. "The player controls a commando who battles waves of enemies including humans, machines, mutants and aliens to reach his ultimate goal." The notable feature used by the game was the two-play simultaneous gameplay. Players work together on the same screen and must work together.

Super Mario Bros. 2 (1988)
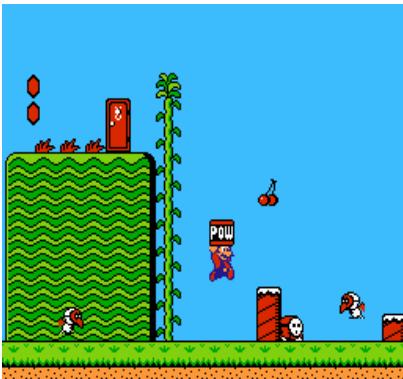
The game was actually a re-done version of another Nintendo game *Doki Doki Panic* with the game's characters replaced by Super Mario Bros characters, hence the game is quite different from the original Super Mario Bros. The game did quite well and was critically acclaimed but did not duplicate the success of the original. This is one of the first games where the player has a female character – Princess Peach to choose from to play with along with Mario, Luigi and Toad. In this version of the game the player could not kill an enemy by jumping on it but was able to pick up a

vegetable and throw it at an enemy, in addition in this version the player had a life meter, which when exhausted (four hits by the enemy) meant the player's demise.

Zelda II: The Adventure of Link (1988)

The second game in the series is an action-adventure game developed by Shigeru Miyamoto. The game combines side-scrolling and RPG element and was not as well received as the original game. The game managed to

introduce game play elements that have became common in other versions of Zelda. The game had more RPG elements, for example experience points, the use of magic, and increased interaction with non-player characters (NPCs).

Castlevania (1987)

This game was developed by Konami and released in 1987. "The series soon became a recognized landmark in the design of action platforming games. It soon became known for its elaborate design aesthetic and theatrical soundtracks. The earliest games borrowed source material from motifs in iconic horror cinema, though the series later developed a more artistic and dramatic direction drawn from Gothic fiction and dark romanticism."

"It is the year 1691...The land of Transylvania has been at peace for one-hundred years now, thanks to the efforts of Christopher Belmont. The peasants and villagers have begun to purge their minds of the memories of the times when the lands were dominated by chaos and shadows, times when the undead walked the earth...However, there are those that remember that the evil Count Dracula returns every one-hundred years to plague the land, bringing with him the forces of Hell...Thus, one evening, the Prince of Darkness rises and returns to Castlevania, his ancestral home, calling forth his minions to purge the

world of human flesh.

The people cry out for a hero — someone to defend them from the evil desires of the Count. Thankfully, they don't have to look very far, for within the land of Transylvania the Belmont line still lives, as Simon Belmont, great-grandson of Solieyu Belmont, takes up the legendary whip called the Vampire Killer, and sets forth on his journey through the darkened countryside to the dark lord's castle...

Upon arrival, the young man fights his way through legions of zombies, gigantic bats, and even faces Death himself, but in the end, he makes it to the Count and in a battle to end all battles, he comes out the victor...

The price? The evil master places a curse of death upon Simon, which will lead him into a long and dangerous journey in the very near future...”

Mega Man 2 (1989)

This is the second game in the original Mega Man series by Capcom. This game is regarded as the best game in the series. The player takes on the role of Mega Man, a robot who travels the side-scrolling levels to fight the Robot Masters of Dr. Wily. The game featured graphics and game mechanics that remained in future games in the series. This was the best selling game in the series. The games under the genre of platformer and was released during the heyday of 2D platform games on game consoles.

The Other Generations

The video game business has settled on home game consoles as the target of their efforts with the current contenders being the Nintendo Wii (yes still in the game), Microsoft Xbox 360 (yes, Mr. Gates decided to join the fun) and Sony Playstation (the dominant leader for over the last decade).

There have been other several generations of game systems.

Fourth Generation

This is the generation of 16-bit microprocessor used to power game consoles. The two big contenders are Nintendo with the Super NES (1990) and Sega with the Sega Mega Drive (1988). There were many great game created for these systems from platformers like Earthworm Jim, Chrono Trigger, Super Metroid, The Legend of Zelda: A Link to the Past, Mega Man X and many more. For the Sega we have Sonic the Hedgehog, the fighting game Street Fighter II and Mortal Kombat and many more.

Fifth Generation

This is the generation of the 32-bit microprocessors used to power game consoles. There were many game consoles in this generation but the two notable survivors were the Nintendo 64 and Sony Playstation. This was the era that slowly knocked Nintendo from its perch as the 500-pound gorilla and crowned a new game console – the Playstation.

Six Generation

The machines in this generation were fast and the graphics impressive. This generation had four contenders, Sega with the Dreamcast, Sony with the Playstation 2, Nintendo with the GameCube and a new kid on the block Microsoft with the Xbox. The console wars have been dominated by one or another Japanese company since the NES appeared on the scene (1985). In 2001 Microsoft decided that the game console was going to be more of a presence and fixture in the family living room being a device not only to play games but view movies, listen to music, and order pizza. At first it tried to convince Sony to have Microsoft use its operating system know-how to help it make it a living room fixture but when Sony declined Microsoft decided to take try its hand in the tough competitive business of game consoles.

Seventh Generation

This is the "passing" generation of consoles. The cpu's and graphics chips used in the XBox 360 and Sony Playstation are breathtakingly fast. These machine render graphics that were one in the domain of very expensive graphics workstations, in fact, you can build a supercomputer with these chips.

Eighth Generation

Since starting these notes a new generation of machines have been released the Nintendo's Wii U (2012), Sony's Playstation 4 and the Microsoft Xbox One.

The consoles get more powerful than ever and are used for more than just playing games. They are all here to stay of a very long time.

Summary

The history of video games as we know them started with one of the three games in the Smithsonian – Pong and continues today. There are many aspects of game design and development that needs to be explored the key one being – what makes a fun game? You have to play games to start answering that question. We highlighted many of the games and I hope you are eager to play some of the games we talked about. I also hope you are even more eager to learn to create your own version of one of these game classics.

In the next section we will start to play some games and do some exercises.

Project: Select a modern game (e.g. Metal Gear) that has a long history and examine the elements of the game that has made it long lasting. Discuss the various game versions and the platforms it was released on.

Chapter 2: How we can play retro games today

This chapter is all about how we can play some of the retro games mentioned in Chapter 1 on our home PC. The goal of this chapter is to:

- Get some popular emulators installed on your PC
- Obtain some game ROMS to play some of the classic games we covered in Chapter 1
- Play the games to analyze and study
  - What the game designer had in mind
  - The game "playability", what makes it fun
  - How the game takes advantage of the technology and its limitation for the time

There are labs associated with each emulator. Follow the instructions and have fun.

Multiple Arcade Machine Emulator (MAME)

MAME is an application available at http://mamedev.org/. This application can be used to run original arcade games, like Donkey Kong, Berzerk, Defender and many others.

You can find ROM files to use by just searching on the Internet with a browser. I should note that unless you know for certain that the ROM chip you are downloading is legally okay to use that you should only download ROMs chips for games you own.

The latest file I downloaded from the website www.mamedev.org was a self-extracting file named mame0126s, which I ran and extracted to C:\MAME.

I obtained the game Berzerk which I found available for free on a public web-site. The version I found did a great job at replicating the synthesized voices of the robots.

The game Berzerk is a multi-directional shooter released in 1980 by Stern Electronics. You will be the green humanoid that must kill robots or be killed by them. It featured talking enemies, including Evil Otto. Otto was named after a real person named Dave Otto from Nutting's Associates who ran the R&D section for the company. The Berzerk game designer, Alan McNeil, apparently wanted the user to feel as helpless as one of the engineers working for Mr. Otto. Evil Otto could not be defeated and would show up if the player stayed in one room too long. In the attract mode the game a

synthesized voice would entice potential game players by stating "Coins detected in pocket."
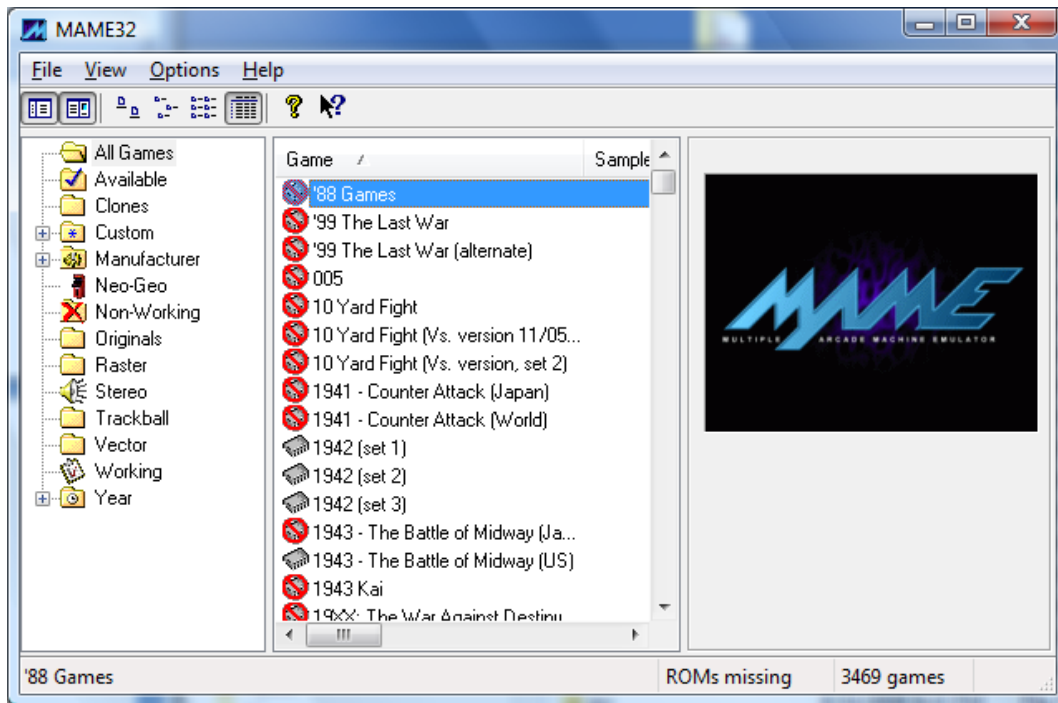Start up the program mame32.exe.

Figure 61 - MAME startup screen

The middle column indicates the games ROMS I can play and those that are were not found in the directory (C:\MAME\ROMS) I used to store all the ROMS I could play. I change some options, for example, I elected to run a game in a window rather than occupying the entire screen.
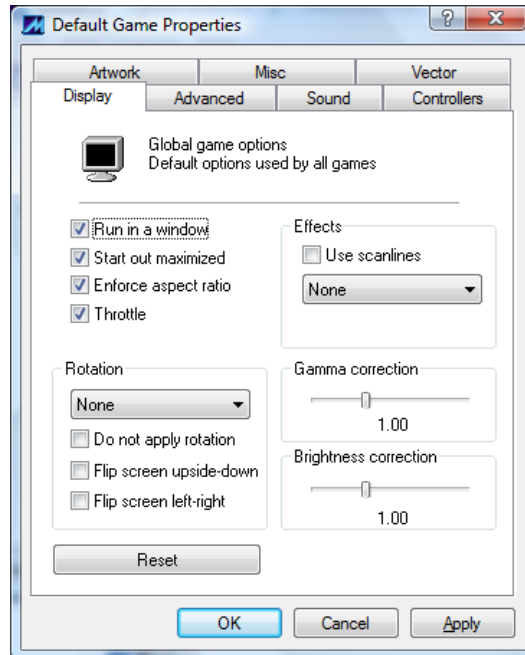
Figure 62 - MAME Display options

Select the game you want to play. For this lab it will be "Berzerk (Set 1)". Double click on the game name.
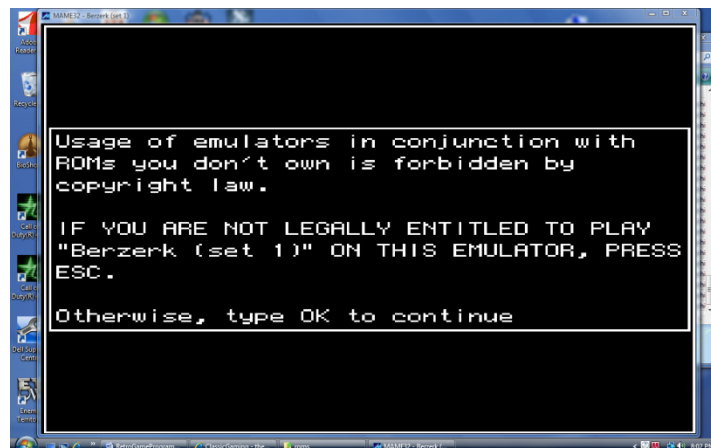
Figure 63 - Legality prompt

I picked up my copy of Berzerk from
http://classicgaming.gamespy.com/View.php?view=GameMuseum.Detail&id=243. Try to obtain the copy with sound.

You will be prompted to ensure that you are entitled to play the ROM you about to play. Enter "OK".
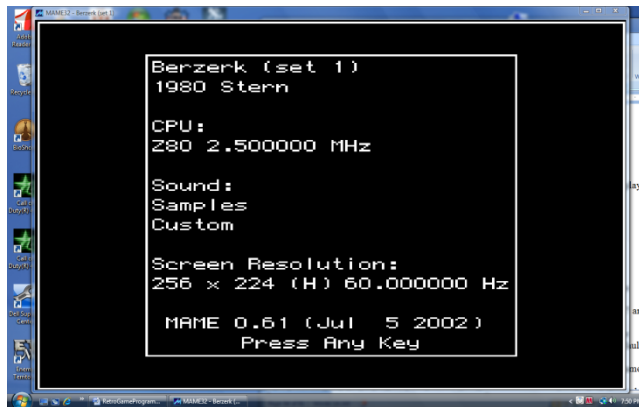
Figure 64 - Berzerk information screen

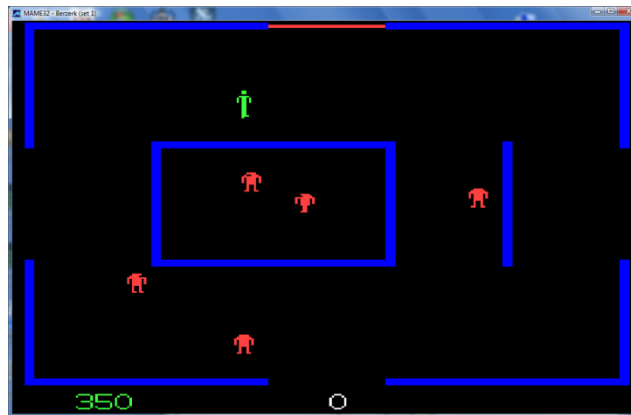Press any key to start the game from the information screen.

Figure 65 - Berzerk attract mode

FROM MAME FAQ
How do I start a game?

Remember, these are not re-creations of arcade games, but the original arcade games themselves. In order to play an arcade game, you must first insert a coin or two. By default in MAME, the "insert coin" switches are mapped to the keyboard keys 5, 6, 7 and 8.
Once you've inserted enough coins to have a credit in the game, you need to start the game. Many games had separate buttons for 1 player start, 2 player start, etc. By default, MAME maps these switches to the 1, 2, 3 and 4 keys on the keyboard.
Some games didn't have separate start buttons; instead you just pressed any of the game's action buttons to start a game. For these games you need to press one of the action buttons. By default, the first three buttons for all games are mapped to the Left Control key, the Left Alt key, and the Spacebar.

Exercise: Examine one of the games in the top 10 arcade game list and play for several hours. Describe each level and the elements that make the game fun.

Stella

The emulator I use to play my Atari games is Stella. You can get a copy by visiting the www.atariage.com. The URL http://stella.sourceforge.net/ will direct you to the latest stable versions of Stella.
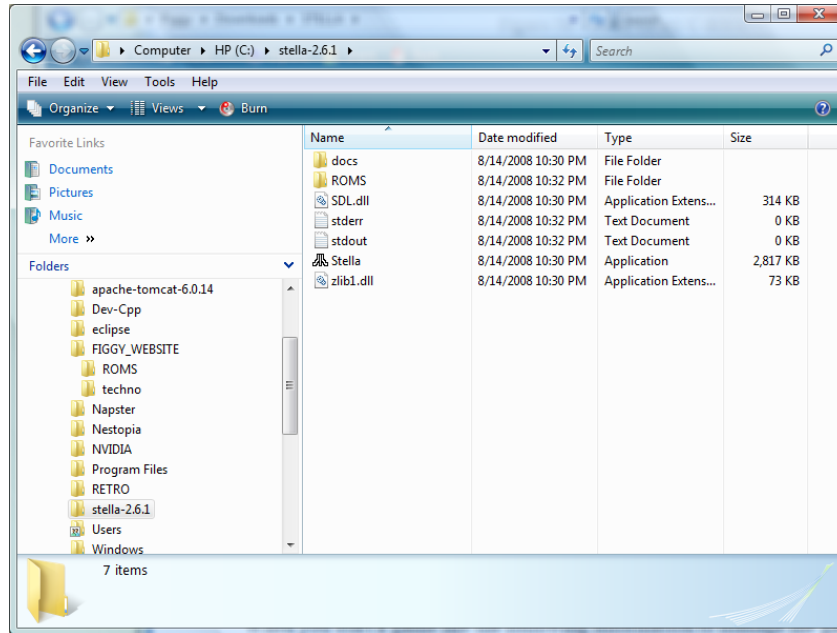
Figure 66 - Stella on my C drive

You will need to obtain game ROMS – that is the game software that resided on the cartridges. There are many places on-line to obtain the files you will need to run on the emulator Stella, I would start with the www.atariage.com.

Start the Stella application by double-clicking on the Stella icon application. What I typically do is move the application to my desktop (this avoids having to open Windows explorer and navigating to the Stella directory) by right-clicking and taking the "Send to" option and specifying "Desktop (create shortcut)."

Figure 67 - Stella Startup screen

You will need to navigate to the directory holding the Atari game ROMS, which for me will be C:\stella-2.6.1\ROMS. I double click on [stella-2.6.1] and then ROMS.
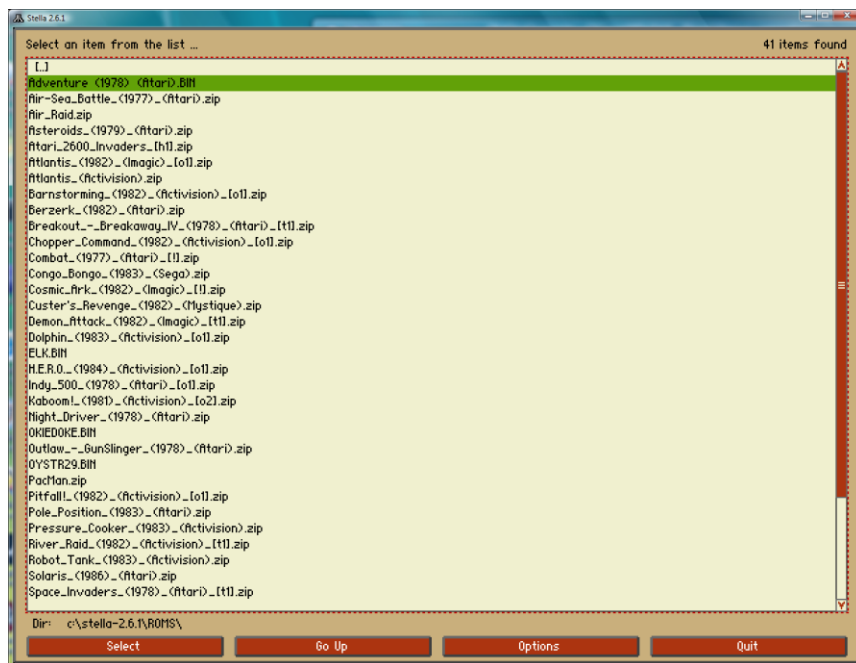

Figure 68 - Viewing files in ROMS directory

You can start a game by double-clicking on the name and use the keys below to select the game type, level of difficulty and controls.

When you start a game use the following information to manage the game once it starts up:

F1: Select Game
      Many games came with several game variations on the cartridge.
F2: Game Reset
      This will re-start the game
F3: Color TV
      These key plays the game in color
F4: Black & White TV
      This key plays the game in black and white
F5: Left Player Difficulty B
F6: Left Player Difficulty A
F7: Right Player Difficulty B
F8: Left Player Difficulty A
      Many games had two levels of difficulty
ESC: Exits the game

Left Joystick:
[TAB] or [Spacebar]   Button
[A]   Left
[S]   Down
[R]   Up
[D]   Down
or use the arrow keys

Right Joystick:
[J]   Button
[K]   Left
[L]   Down
[O]   Up
[;]   Down

Exercise: Spend time playing the top Atari 2600 games mentioned in Chapter 1.

AppleWin

We will use AppleWin to play Apple IIe games. You can obtain the emulator from any number of places. Try www.brainycode.com and follow the links to the emulator and apple games. Download the file AppleWin1.12.6.0 and unzip to your hard drive. I recommend the directory C:\AppleWin.
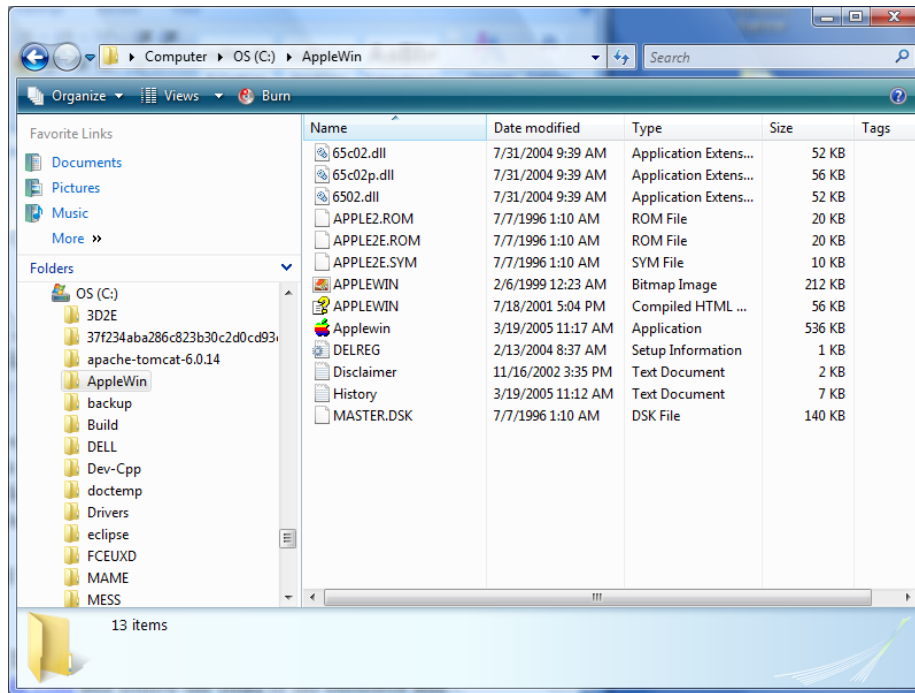
Figure 69 - Unzipping AppleWin to your hard drive

Create a directory C:\AppleWin\DISKS to place all your Apple files that represent the diskettes the games would be stored on.

Figure 70 - Apple II Disk Drive

The diskette would be inserted into the Apple II disk drive.

Figure 71 - AppleWin startup screen

The AppleWin emulator starts life as an AppleIIe by clicking on the button with the apple image on the menu. The machine will make a distinctive sound that emulates the startup sound of a real Apple machine. The lab will walk you through emulating inserting a disk into the machine and starting to play.

Nestopia

You can locate the latest copy of Nestopia using an Internet search engine, download and unzip the machine to your PC. I placed my copy at C:\Nestopia and all NES game ROMS at C:\Nestopia\GAMES. I own a copy of Metroid so I downloaded a ROM image to play on my NES emulator.
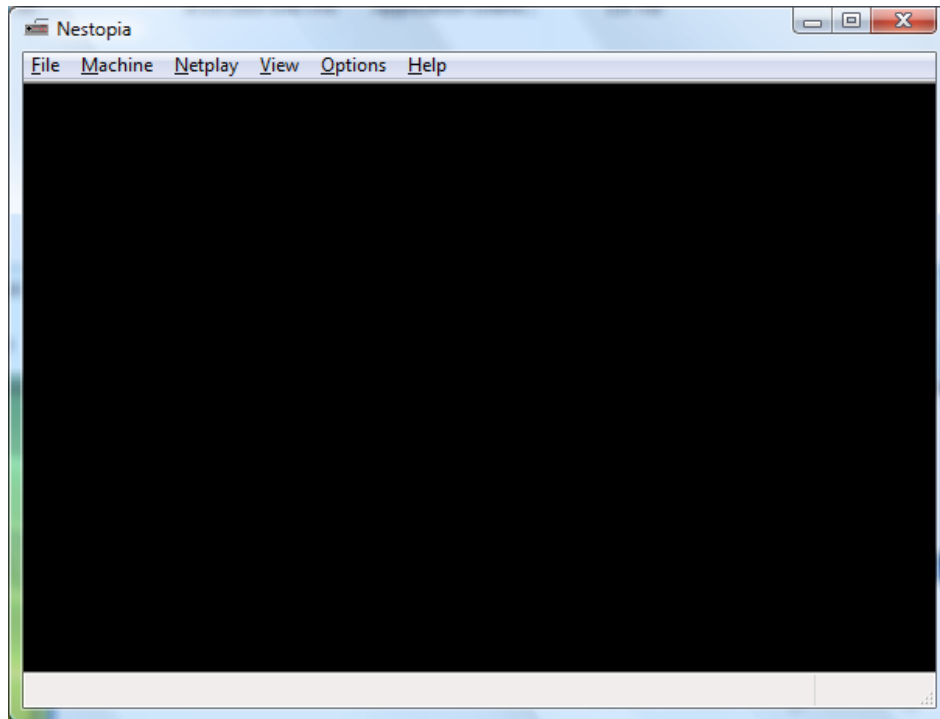
Figure 72 - Nestopia startup screen

The default screen size when you startup it up is rather small but it can be resized using the View|Screen Size.

You can start a game by selecting File | Open and doubling clicking one of the games in your game directory.

The labs below will guide in how to use the emulators above to play some favorite games.

Lab 2.1 – Using MAME to play some classic SHMUPS

The purpose of this lab is to play a series of classic shoot-em ups and compare how the game play evolved for moving a space ship on the screen and shooting at aliens.

Step 1: Obtain ROM copies of the following games
- Space Invaders (1978)
- Galaxian (1979)
- Galaga (1981)
- Gradius (1985)
- R-Type (1987)

 A web site to visit to download these ROMS is www.romworld.com.

Step 2: Play each game for at least 20 minutes. Try to allocate the same amount of time to each game even if you have played it in the past.

Step 3: Write a review for each game comparing the game play, levels, aliens, challenges, and new feature(s) introduced by each particular game to advance the shump genre. Also, discuss the possibility of the game still being playable today.

Lab 2.2 – Using STELLA to play Adventure and Pitfall!

Step 1: Obtain copies of the following Atari 2600 game Adventure and Pitfall! at www.atariage.com.

Step 2: Play Adventure using skill level 1. This will be quite easy to complete. Try to complete the game with skill level 2.

From the manual:

SKILL LEVELS

Level 1 This is the simplest Skill Level. When you depress the game reset switch to begin play, you will see the Key to the Golden Castle. Unlock the Castle and enter. You will find the Sword inside the Golden Castle. The Key to the Black Castle is being guarded by Grundle, the Green Dragon. Yorgle, the Yellow Dragon is roaming free and may or may not be found guarding the Enchanted Chalice, which is hidden with the Magnet inside the Black Castle.

Level 2 This Kingdom is much larger than Level 1. There are Catacombs (in which you can see only part way). The Key to the Golden Castle is hidden here. You must pass through the Catacombs to reach the White Castle. The Key to the White Castle is hidden in the Blue Labyrinth. Inside the White Castle is the Red Dungeon. There is a Secret Room in the Red Dungeon where the Key to the Black CAstle is hidden. To get to the Secret room you must use the Bridge.

To get to the Black CAstle you must pass through the Blue Labyrinth. Behind the first room of the Black Castle is the Grey Dungeon, which is similar to the Catacombs. The Enchanted Chalice is hidden here, guarded by Rhindle, the Red Dragon.

All objects, the Dragons, and the Black Bat will start in the same place in the Kingdom each time you play the game at Level 2.

Level 3 The Kingdom is the same as Level 2, but is more difficult to play as the Evil Magician has placed all the objects and the Dragons randomly within the Kingdom. You will never know for sure what is in the next area of the Kingdom until you enter it, nor will you know for sure where the Enchanted Chalice may be hidden. The Dragons could be inside any of the Castles.

You can get a complete copy of the game manual at www.atariage.com.

Step 3: Play Pitfall!

See if you can get the hang of jumping on the vines and on top of the crocs heads. The goal is to collect all the treasure (all 32) in under 20 minutes of game play.

Step 4: Write a review of the game play, graphics and sounds associated with these two Atari 2600 games. Take into consideration the hardware limitations and what was available for home video systems in 1977.

Lab 2.3 – Using AppleWin to play Crisis Mountain

Step 1: Obtain a copy the game Crisis Mountain from www.brainycode.com and save in a directory you will create named GAMES at C:\AppleWin1.14.2.
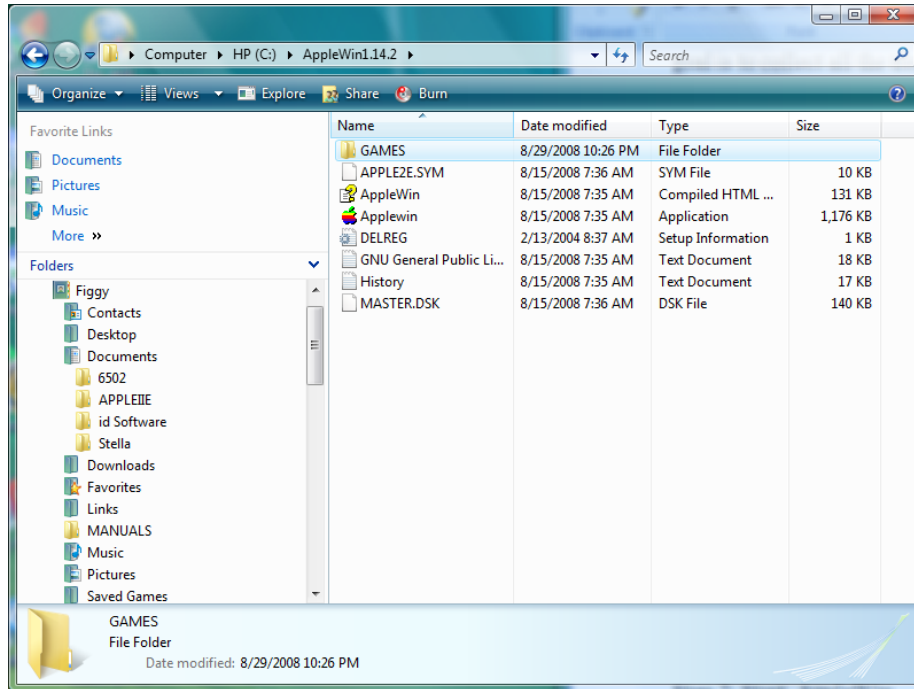

Figure 73 - Location to save AppleWin Games

Step 2: Start AppleWin by double-clicking on the icon in C:\AppleWin.

Figure 74 - Startup screen for AppleWin

Step 3: Click on the disk icon labeled "Master". A dialog box opens up where you should navigate to the directory and highlight the file for Crisis Mountain.
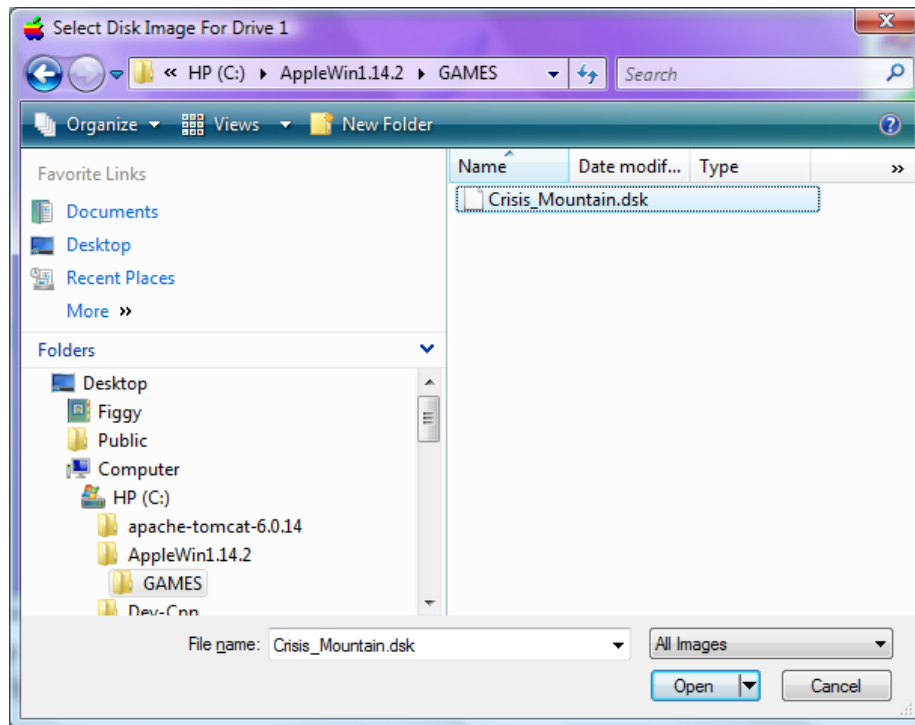
Figure 75 - Navigate to Crisis Mountain.dsk

Step 4: Click on the Apple icon to load and startup the game.

Play the game though at least the first screen. See if you can stop the two clocks in time. You must pick up and use what looks like a white shovel in order to stop the clocks. Don't forget that you can jump over boulders and crouch.

Step N: Read the game review below from Compute! Issue 40, September 1983 by Patrick Parrish and write a paragraph on whether you agree with the review.

Crisis Mountain For Apple And Atari

Patrick Parrish, Editorial Programmer

Crisis Mountain, programmed in machine language by Ron Aldrich and David Schroeder, is an excellent, exciting game, requiring an Apple II or Apple II Plus with 48K RAM (also available for the Atari 400/800 with 48K) and a disk drive. This one-player contest from Synergistic Software can be played with either a joystick or the game paddles.

   The scenario of the game is that a group of terrorists was hiding out in the caverns of a

dormant volcano in the Pacific Northwest. The volcano erupted unexpectedly, forcing the terrorists to abandon their hideout. As they fled, they left behind their loot and supplies - and several nuclear bombs. To save the West Coast from impending disaster, *you* must venture into Crisis Mountain, dig up and defuse the bombs while avoiding numerous hazards.

Nine Skill Levels

Crisis Mountain alternates between two cavern scenes as you progress through nine skill levels. In the beginning of the game, you are given three lives. And if you're skillful enough you can earn a life at 10,000, 30,000, and 50,000 points. On each level you are presented with a labyrinth of passageways, precipices, and fiery lava pits which sporadically spew rocks and debris.

Scattered about the cavern, in addition to innocuous objects left by the terrorists, are active bombs positioned randomly in one of five locations. Each displays a time, also randomly chosen, before detonation. As you advance from one skill level to another, you are challenged with more bombs and less time to defuse them. Thus, picking the appropriate route through the maze of passageways becomes more and more critical.

Points are awarded for the completion of several tasks. Nominal scores are given for gathering the loot, gun caches, and boxes left by the terrorists. Once you've collected all items, certain bonus forms appear in random positions about the cave.

Another way to score points is to leap boulders. The larger the boulder, the more points you receive. Being struck by a boulder, on the other hand, diminishes your strength. The strength level is indicated with a number from one (weakest) to three (strongest). When you are weakened, your point scoring abilities are significantly impaired. In fact, at strength level one, scoring becomes secondary to mere survival since you can rarely manage to leap boulders in this weakened condition. Fortunately, there are several safe nooks around the cavern where you can recover.

Your running figure (center) leaps a tumbling boulder in Crisis Mountain.

Treacherous Caverns

There are other ways you can be destroyed in the game. You can fall or be knocked into a

lava pit by a boulder, a bomb can detonate, or you can be bitten by the deadly bat, Bertrum.

It is obvious that tremendous effort went into designing this game's high-resolution graphics. Each form is drawn in intricate detail. The frothing lava pits and tumbling boulders are remarkably realistic.

The Deadly Bat

But the most remarkable graphic element of the game (and the most confounding to any player) is Bertrum, the bat. Bertrum flits about the cavern in a way that resembles a real bat. If a boulder is blasted from a nearby lava pit, Bertrum will dart toward it for a quick inspection, determine the rock is not prey, and fly off to another part of the cave.

But Bertrum is more than just a visual success. His presence adds a degree of chance to the game which makes it faster and more challenging. This dreaded bat has a knack for determining where your player is at any moment in the game. Sometimes, you can avoid Bertrum with a last minute duck or leap. At other times, escape is simply impossible. I've yet to discover a foolproof way to evade this creature, though there may be a tactic.

There are several other excellent features of this game. For one, the ESC key allows you to halt or resume a game at any time during play. With *Crisis Mountain*, a game can sometimes last an hour or more. A break during such a prolonged period of play, beyond being a convenience, is often essential for maintaining your concentration. (No "save game" option is offered.)

Although the sound effects are very good, you may want to turn them off occasionally. If so, you can cancel output to the Apple speaker with CTRL-S. On the other hand, if you want an engulfing, environmental audio effect, output can be sent to external speakers via the cassette port. You can also store on disk, and subsequently display, the high score to date.

Overall, *Crisis Mountain* is a superior programing achievement and a thoroughly entertaining game.

Crisis Mountain
Synergistic SOftware
830 N. Riverside Drive
Suite 201, Ronton, WA 98055
$34.95

Lab 2.4 – Using Nestopia to play Metroid

Step 1: Obtain a copy of the game Metroid.

Step 2: In order to get started I recommend you find one of the many helpful guides online. If you want to savor the fun of discovery in navigating through the game then avoid the detailed walkthroughs. The following is a nice description of the game put together in a walkthrough by Matt McGrath.

In the year 2000 of the history of the cosmos, representatives from the many different

planets in the galaxy established a congress called the Galactic Federation, and an age of prosperity began. A successful exchange of cultures and civilization resulted and thousands of interstellar spaceships ferried back and forth between planets. But space pirates also appeared to attack the spaceships. The Federal Bureau created the Galactic Federation Police, but the pirates' attacks were powerful and it was not easy to catch them in the vastness of space. The Federation Bureau and the Federation Police called together warriors known for their great courage and sent them to do battle with the pirates. These great warriors were called "space hunters." They received large rewards when they captured pirates, and made their living as space bounty hunters.

It is now the year 20X5 of the history of the cosmos, and something terrible has happened. Space pirates have attacked a deep-space research spaceship and seized a capsule containing an unknown life-form that had just been discovered on planet SR388. This life-form is in a state of suspended animation, but can be reactivated and will multiply when exposed to beta rays for 24 hours. It is suspected that the entire civilization of planet SR388 was destroyed by some unknown person or thing, and there is a strong possibility that the life-form just discovered was the cause of the planet's destruction. To carelessly let it multiply would be extremely dangerous. The Federation researchers had named it "Metroid" and were bringing it back to Earth when it was stolen by space pirates!

If Metroid is multiplied by the space pirates and then used as a weapon, the entire galactic civilization will be destroyed. After a desperate search, the Federation Police have at last found the pirates' headquarters, the fortress planet Zebes, and launched a general attack. But the pirates' resistance is strong, and the Police have been unable to take the planet.

Meanwhile, in a room hidden deep within the center of the fortress, the preparations for multiplying the Metroid are progressing steadily. As a last resort, the Federation Police have decided on this strategy: to send a space hunter to penetrate the center of the fortress and destroy the Mother Brain. The space hunter chosen for this mission is Samus Aran. He is the greatest of all the space bounty hunters and has successfully completed numerous missions that everybody thought were absolutely impossible. He is a cyborg: his entire body has been surgically strengthened with robotics, giving him superpowers. Even the space pirates fear his space suit, which can absorb any enemies' power. But his true form is shrouded in mystery.

Step 3: Play the game long enough to make your way through BRINSTAR. Find the Maru Mari (start by going left), this will enable your character Samus to transform into a ball to make it underneath small passages. There is also a long beam that lengthens the range of Samus's beam in order to kill enemies from a longer distance. In addition, pick up the bomb in order to blow up certain blocks. If you want to experience the elements of a good side-scrolling game – then play until you discover the big secret about Samus.

Summary

This chapter was all about installing and using the emulators to play games designed for the arcades using MAME, the Atari 2600, Apple II and the NES. The labs helped to explore how shoot'em ups evolved and some interesting platform games for all three non-arcade machines.
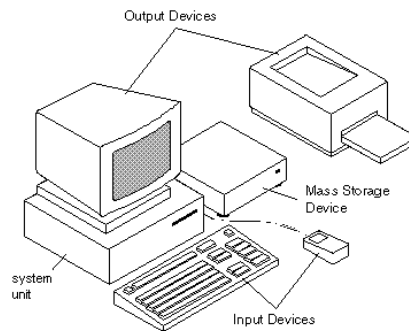
Chapter 3: Computers, Programs, and Programming

Figure 76 - A typical computer system

How we use computers

You probably have a lot of experience using a computer to chat online, browse the Internet, send and read e-mail, create documents, or playing computer games. If you were to look inside your desktop computer you would see the following components or parts:
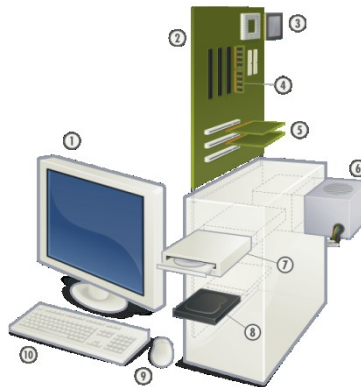

Figure 77 - Components of a computer system

The parts you are probably very familiar with are:
1- The monitor or computer screen
   This is the part that visually displays what is going on inside the computer on your behalf. You may have windows open listening to music on your favorite Internet station, other windows open to your chat client waiting for you best friend to log in so you can gossip, etc.

7 – The CD or DVD devices that you use to insert game disks
   The CD or DVD disc is where programs, images, sounds are stored. For the most part you can regard the information on these devices as permanent, that is, they don't disappear when the machine gets turned off. The same thing is true for

information on game cartridges.

10 – The keyboard

This is one of the devices you use to input information to the programs, like URLs (www.brainycode.com) to browsers. You also use a mouse and if you are lucky to own a gamepad or joystick you can use them to play games.

The parts you may not readily recognize since they are all inside your computer are:

Figure 78 - Computer motherboard

2 – The motherboard.

This is where the key components of your machine reside.

3 – the central processing unit (CPU) is where all the processing takes place. The CPU on the machine I am currently using to write this document is an Intel® Core ™2 Quad CPU[31]. It happens to be pretty fast and impressive. It actually has four CPUs on the chip. The CPU we will be learning to use the 6502 is one simple microprocessor. The key thing about a CPU (at least it used to be key when computers only had one CPU) is the speed. Let's compare the speed of various machines and consoles from yesteryear and today. The key measurement we will be considering is the number of cycles or clock ticks per second. This speed correlates to the average number of instructions or operations it can execute per second. Each operation corresponds to the execution of a CPU instruction. It is with each instruction that aliens get moved across the screen, missiles fired and sound affects achieved, so the more instructions a CPU can execute per second the more graphics, sound, game elements can make the game more realistic.

Table 1 - Comparison of machine speeds

Machine/Console	Year	Speed/machine clock size	Interesting Fact…
PDP-1	1960	200 KHz	Was the machine used to make the game SpaceWar!
IBM System/360	1964	16.7 MHz	This was the machine I first

---

[31] This was written around 2006

101

			to learned to program on and played the text-based game Adventure.
Atari 2600 – MOS 6507	1977	1.19 MHz	The first major game console system that used a version of the 6502 chip.
Apple IIe - 6502	1983	1.023 MHz	The computer that got the home video computer business started and the first computer I owned.
NES - 6508	1985	1.79 MHz	A version of the 6502. Games were fast due to the specialized graphics chip.
SNES – 65C816	1990	2.68 MHz and 3.58 MHz	Not until Donkey Kong Country did it really become the "must have" machine.
Playstation	1994	33.8699MHz	The console to beat Nintendo in the game console business.
Nintendo Game Cube	1996	93.75 MHz NEC VR4300	Nintendo sold over 11 million Super Mario 64 games. A great game!
Playstation 2	2000	Emotion Engine 294.912 Mhz	So far my favorite system with my favorite games… God of War and God of War 2 and Ms. Lara Croft.
XBox	2001	Custom 733 MHz	Halo anyone?
Nintendo Game Cube	2001	PowerPC Gekko 485 MHz	It is actually backwards compatible with the Game Boy Advance!
XBox 360	2005	3.2 GHz PPC Tri-Core Xenon	I just love XBox Arcade.
Wii	2006	IBM PowerPC-based, 729 MHz	The innovative Wii Remote which has a motion detector may make this "souped" up Game Cube the winner in this round of console wars.
Playstation 3	2006	3.2 GHz Cell Broadband Engine	IBM designed the CPU for the top three consoles for this generation.

The way to understand the speed value is to start with the Hz, which stands for hertz. A hertz is a measure of frequency or the number of times something happens per second. 1 Hz is equal to one cycle per second. KHz stands for kilohertz. When the frequency is 1000 hertz represents or 1000 cycles/second we use the shorthand 1 kiloHertz or 1 KHz.

So the PDP-1 ran at 200 KHz or 200,000 cycles per second. MHz stands for 1 million cycles. So a CPU that runs at 1 MHz cycles at 1,000,000 cycles per second. Today's consoles (except the Wii) have CPU's that execute in the GHz range where 1 GHz is 1 billion cycles per second. So how much faster is an XBox 360 over an Atari 2600?

$$\frac{\text{XBox 360 speed}}{\text{Atari 2600 speed}} = \frac{3.2 \times 10^9 \ \text{Hz}}{1.19 \times 10^6 \ \text{Hz}} = 2.7 \times 10^3 = 2700 \text{ times faster}$$

The calculation above is actually meaningless. Can you believe that an XBox 360 is actually way more than 2700 times faster than an Atari 2600 since it gets a boost by a graphics chip that runs at 500 MHz and the CPU chip actually has more than one CPU working to make Gears of War look so good. It is hard to compare machines by just taking into consideration the speed of their heartbeat – the CPU. If you look at the screen of an Atari 2600 game vs. a game designed for the XBox 360…

.. well it is like comparing an amoeba to a human.

The systems we are going to learn in detail Atari 2600, Apple IIe, and NES all these systems use the same type of CPU the 6502. The machine cycles are 1.19 MHz, 1.023 MHz, and 1.79 MHz, respectively. The speed of these machines look puny compared to today's game console CPU's. The fact, is the speeds are rather dismal but the games constructed on them were simpler to understand and easier to duplicate, hence the reason I think it would be easier for you to work with learning to program and create your own games.

4 – memory

The memory chips are where the program you are running is placed. The CPU obtains its instructions from memory. So that program that manages your spaceship and responds to the joystick fire button resides in this part of the computer. Memory also holds data that is used to track how many lives you have left, or the number of monsters left on that level you are trying to get pass. Instructions and data are going into and out of the CPU as

your program is processed. Unlike the contents of CD and DVD disks when the machine is turned off the information in memory disappears.

5 – extension cards

You can expand the capabilities of your computer by installing network cards, video cards, modems, etc. I should note that many inexpensive computer systems have these components already installed on the motherboard.

6 – power supply supplies the power to the computer
8 – hard drive

This is the component that has the operating system, e.g. Microsoft Vista and other software that you probably installed from CDs/DVDs or the Internet. The information on this device does not go away when you turn off the machine.

The Hardware and Software

The last section went into details about the different components in your desktop computer. The key component for anyone who wants to make games is the CPU since it controls and manages what is going on in the computer and its speed will pretty much dictate the constraints of the game you create. Two other key components is the graphics chip (if any) and sound chip (again, if any). The graphics and sound chips will assist the CPU in getting the job done of rendering the game to the user.

The CPU executes the program that resides on the game cartridge. The flow of data within the home computer or game console is the following:
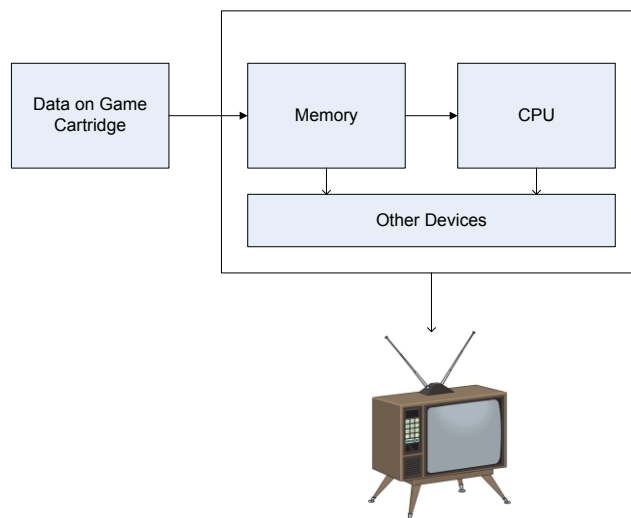

Figure 79- How that game gets on your television

The program and the data elements it displays must be placed into computer memory (RAM). The CPU then processes the machine instructions. The instructions may correspond to reading in the game button just pressed by the user, saving the value for the number of monsters on the screen, sending data to the video device so the gamer will actually see something on their screen. The "other devices" will be different for the three machines we will be studying but the final output to your television set is the same – a wonderful game!

The data on the game cartridge consists of a program designed to execute for that CPU[32] and data that holds the images and sounds that we see when we play the game. The data on the cartridge is stored on a memory device called ROM which stands for read-only memory. The data remains on the cartridge. The cartridge may also contain another type of memory that holds your score or save state while playing the game. This type of memory (e.g. WRAM) may require a battery in order to sustain the information for a reasonable length of time.

I recently purchased some NES games from eBay. I was rather disappointed to find out the battery inside the cartridge had died and hence not allowing me to save game state! I have to keep the NES running so I don't have to start all the way from level 1.

Programmers are the folks that create the programs we use in our computers or consoles. They create computer programs for devices as small as my digital watch, my cell phone, game boy advance or as big as my microwave, computer, notebook, XBox 360, or automobile. The programs programmers write to get the machines to do their thing look like strange incantations that only these wizards can understand.

Below is a small piece of code from a larger program that consumed hours of my time.

```
//
// D_DoomMain
//
void D_DoomMain (void)
{
    int        p;
    char       file[256];

    FindResponseFile ();

    IdentifyVersion ();

    setbuf (stdout, NULL);
    modifiedgame = false;

    nomonsters = M_CheckParm ("-nomonsters");
    respawnparm = M_CheckParm ("-respawn");
    fastparm = M_CheckParm ("-fast");
    devparm = M_CheckParm ("-devparm");
    if (M_CheckParm ("-altdeath"))
            deathmatch = 2;
    else if (M_CheckParm ("-deathmatch"))
            deathmatch = 1;
```

---

[32] Not all CPUs are created equal. Each one has a unique set of instructions that they handle.

```
switch ( gamemode )
{
 case retail:
        sprintf (title,
                       "            "
                       "The Ultimate DOOM Startup v%i.%i"
                       "            ",
                       VERSION/100,VERSION%100);
        break;
 case shareware:
        sprintf (title,
                       "            "
                       "DOOM Shareware Startup v%i.%i"
                       "            ",
                       VERSION/100,VERSION%100);
        break;
 case registered:
        sprintf (title,
                       "            "
                       "DOOM Registered Startup v%i.%i"
                       "            ",
                       VERSION/100,VERSION%100);
        break;
 case commercial:
                       :
                       :
                       :
```

You may be able to recognize the program code above as small portion of the game
Doom. So now you know the hours I spent were not spent programming but playing the
game. If you don't know the programming language used in the above sample some parts
will be readable and understandable and others more like Greek. The program or code
above is written in the programming language called C. The popular computer
programming languages today are C, Java, Objective C(Apple products),C++, C# and
Visual Basic[33], etc.

In order to program the games on the systems we plan on using in this book we will need
to learn Assembly Language specifically intended for the 6502 microprocessor.

So let's begin.

---

[33] http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Chapter 3: Learning Assembly Language

This chapter is going to be pretty long since we will have to cover enough about programming in order to start building games. The chapter will cover what programming is about and how programmers use a language such as assembly language to control a computer and make Mario jump on a mushroom, or Samus fire a weapon, and Pitfall Harry swing on a rope.

What is Computer Programming?

A computer program or program is a set of instructions that a computer follows to perform some action. A computer program is typically written in a computer programming language such as C++, Java, or Visual Basic. These programming languages are known as high-level programming languages. A high level language can look more like the languages we naturally communicate in but not exactly since a computer has to have clear unambiguous instructions, for example:

```
if ( numberOfMonsters == 0 ) {
        advanceUserToNextLevel();
}
```

The above reads "if the number of monsters is equal to zero, then advance the user to the next level." A program written in a high-level programming language has to be translated into the language used by the CPU you are using. The CPU only understands machine language. An example of what machine language looks like follows:

101010010000010010001101000101100000001010101001000001101...

That's right the computer understands 0's and 1's. The 1's and 0's represent the state of switches within the machine. Each switch is either on or off. We represent this for ourselves by using 1 for 'on' and 0 for 'off'.   The machine language or code above represents only three machine instructions for a program that actually requires a total of eight machine instructions that just add two numbers together and stores the result into a third number. The equivalent program written in a high-level language such as C would only require one instruction:

sum = a + b;

You can read the above as "Add a and b and store the result in sum".   It is fairly safe to say that a program written in a high-level language looks more like math or science formulas than machine language version of a the same program. Programming in machine language ….using 0's and 1's is tedious and error-prone. The first hobbyist computers required users to program them this way.
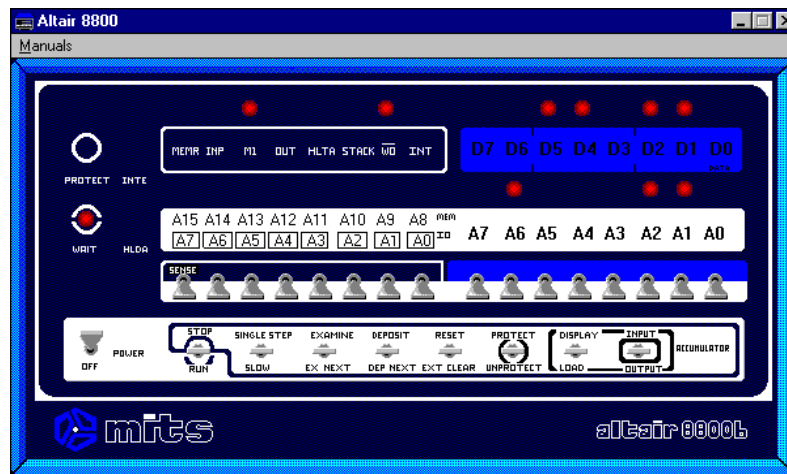
**Figure 80 - Windows Altair emulator**

The image above is actually an illustration of a Windows program depicting the front panel of one of the first hobbyist machine – the Altair. Users had to enter programs using machine language. The front panel had switches that would represent the up ('1') and down ('0') state of a switch.

I should note that I am being very loose in using the word "computer." The fact is a computer consists of many distinct elements. The part of the computer that actually processes machine language (your program) is called the central processing unit (CPU).

The middle ground between the tediousness of machine language (all those 0's and 1's) and a high-level language is assembly language. This is what we will be a learning. An assembly language designed to generate instructions for the 6502 microprocessor.

The series of '0's and 1's is called binary and in the next section we will discuss how computers represent numbers and characters.

Learning how computers represent information
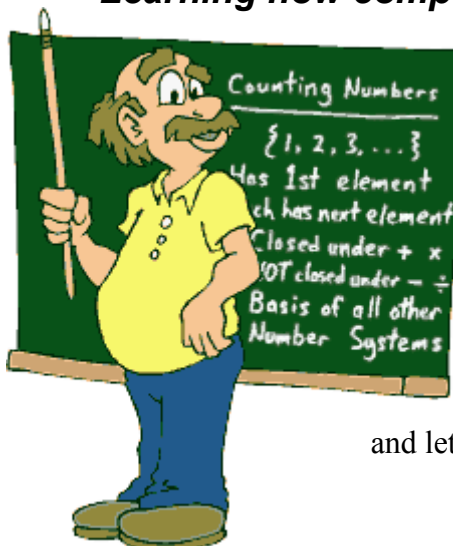
This is the part of the book that I found the toughest to do since it was both the most essential and the most boring. How could I possibly make learning to count and manage numbers the way a computer does exciting? I don't recall where I was or who taught me how to count, add and subtract and figure out how to tip the waiter, but somehow I don't think I thought it was exciting, it was something I was told I had to learn or else (at least no one could threaten to take my game console away if I didn't since it wasn't invented it). Well, here it goes, "You now have to learn computer arithmetic." Now get the groaning out of the way and let's start…

The information stored in memory and processed by the CPU is grouped into *bits* (0's and 1's). IBM with the invention of the wonderful IBM/360 standardized the use of 8-bits used as a distinct group called a *byte*. The 6502 microprocessor is an 8-bit microprocessor since it fetches data from memory 8-bits or 1 byte at a time.

Example of a bit: 0
Another example of a bit: 1
Example of a byte: 01010101
Another example of a byte: 11111111

In order to start learning to use bits and bytes we need to learn binary arithmetic, but first let's examine our good ol' number system.

Our base-10 system

We learned in grade school the base-10 (or decimal) number system which is a positional number system. In our natural number system we have ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The ten digits naturally come from the fact that we have ten fingers. All our numbers are constructed from these ten digits. For example, given the number

123

We read this as "one hundred and twenty three". Each digit has a weight that contributes to the value of the number. The '1' is in the hundreds position, the 2 in the tens position and the 3 in the ones position. When given any number each digit in the number from right to left increases by a power of ten. We can re-write the number as:

$123 = \underline{1} * 10^2 + \underline{2} * 10^1 + \underline{3} * 10^0$

In general given any number in our base with the digits

$d_n d_{n-1} \ldots d_2 d_1 d_0$ we can

compute its value by multiplying each digit starting with d_0 with an increasing power of 10.

$$d_n * 10^n + d_{n-1} * 10^{n-1} + \ldots + d_2 * 10^2 + d_1 * 10^1 + d_0 * 10^0$$

Also, note that when we are counting up by one 0, 1, 2, 3, 4, 5, 6, 7, 8, 9….the next number wraps around and continues as $\underline{1}0$…we then increment the one's again, 11, 12, 13, 14, 15,16, 17, 18, 19, and note that two things happen the ten's go up by one and the one's re-start at the beginning again…2̲0, 21… etc. When we get to 99 we move into the hundreds position now and restart all digits to the right back to 0…1̲00. We picked up this pattern ages ago and do it without much thought.

The above should be a review our beloved number system. Knowing the underlying principles will help us understand number systems using a different base. Since we will

be working with numbers from different bases (base-10, base-2 and base-16) we will end each number not identified directly in text with a subscript to indicate the base so 123 in base-10 will be represented as 123_{10}.

Binary World – base-2

The world within a CPU is constructed from switches that have one of two states – 'on' or 'off'. Hence in this world it is natural to use only two digits to represent a number – 0 and 1.

So for a computer to count from 1_{10} to 10_{10} in binary will go through the following sequence:

0, 1, and now we get the rollover we did with our base-10 system so the next number is 10, and then 11, now we get a rollover to the next position 100, 101,110, 111, etc. Compare the chart below listing a number in decimal notation and its equivalent in binary:

Table 2 - Decimal/Binary numbers from 0-15

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

The first thing that is obvious is that the binary equivalent of any decimal number will generally require more digits to represent, since a binary number only has two digits to work with.

Converting from binary to decimal

Not too many programmers can look at a sequence of 1's and 0's and tell you what decimal number it represents unless it is small number (see Table 2). We will need to learn to convert from binary to decimal to determine what number we are working with.

Converting from binary to decimal is rather easy just expand the binary number where you multiply each binary digit by an increasing power of 2 from right to left.

You will need to memorize this table – the powers of 2 (or at least figure out how to generate it):

Table 3 - Powers of 2

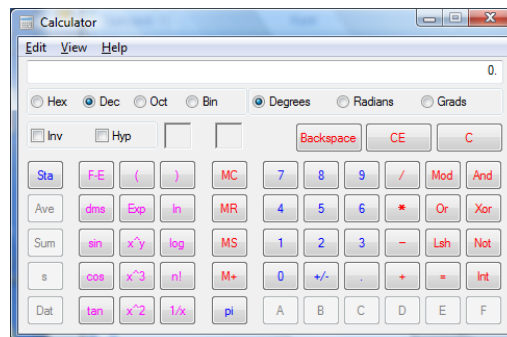|	
2^0	0
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256

Given the binary number 10110101 it is equal to the following decimal number:

10110101 $= 1 * 2^7 + 0 * 2^6 + 1 * 2^5 + 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0$

quickly drop all the terms that multiply against a 0 bit

$$= 1 * 2^7 + 1 * 2^5 + 1 * 2^4 + 1 * 2^2 + 1 * 2^0$$
$$= 2^7 + 2^5 + 2^4 + 2^2 + 2^0$$
$$= 128 + 32 + 16 + 4 + 1$$
$$= 181$$

You can double check you work by using the Microsoft Windows Calculator.

Figure 81 - Software calculator

If your calculator does not look like the one above make sure that the View | Scientific is selected.   Select the "Bin" option for binary and enter the binary number.
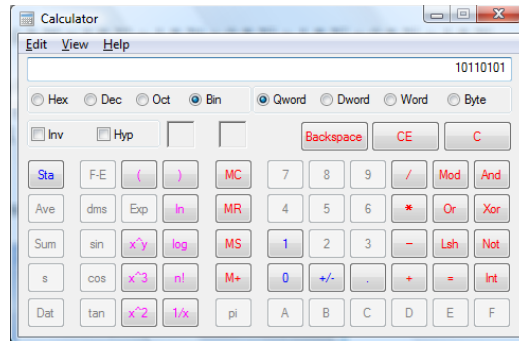
Figure 82 - Calculator with binary number

To quickly convert the number entered to decimal just click on the "Dec" option for decimal:
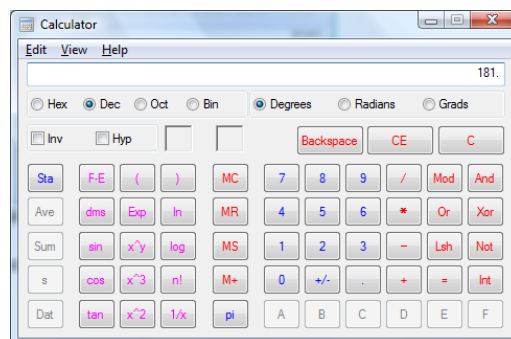

Figure 83 - Converting binary number to decimal

The value shown by the calculator matches the value we computed the old-fashioned way.

Converting from decimal to binary

With the Windows Calculator it is easy to convert from decimal to binary. I will let you figure it out.

The pencil and paper technique requires that we repeatedly divide the decimal number by 2 and take the successive remainders as the next binary digit building from right to left of the binary number we are constructing.

For example, given the decimal number 201 the number in binary will be:

201 / 2 = 100 r **1**
100 / 2 = 50 r **0**
50 / 2 = 25 r **0**
25 / 2 = 12 r **1**
12 / 2 = 6 r **0**
6 / 2 = 3 r **0**
3 / 2 = 1 r **1**
1 / 2 = 0 r **1**

So the binary equivalent to 201_{10} is 11001001_2.

Adding in binary

You should remember the following when adding numbers larger than one bit

```
  0     0     1     1
 +0    +1    +0    +1
 --    --    --    --
  0     1     1    10
```

Note, the last $1_2 + 1_2 = 10_2$, that is, we get a carryover of one bit into the next binary position.

Example: Add the following two binary unsigned numbers:

```
  00111001
 +10110011
 ----------------
```

```
        1                        11            11   11
  00111001                 00111001            00111001
 +10110011      =>      +10110011      =>   +10110011
 -----------            -----------         -----------
          0                       00               101100
```

```
   11   11
  00111001
 +10110011
 -----------
  11101100
```

Note, that in binary $1 + 1 + 1 = 11_2$

All our binary numbers have been unsigned numbers.

The convention we will use when referring to the bits in a number is to designate a number position for each bit.
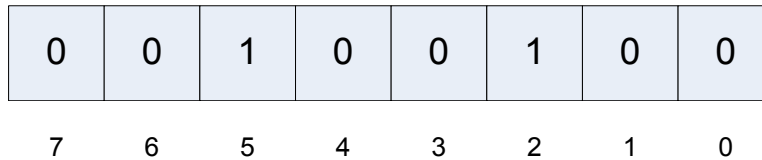
0	0	1	0	0	1	0	0
7	6	5	4	3	2	1	0

Figure 84 - bit positions in a byte

When working or talking about the bits in the byte value shown in Figure 84 we refer to the least-significant or right-most bit as bit position 0. We increase the bit count by one from right to left. The most-significant or left-most bit position has bit position 7. In general given a CPU that uses n-bits to represent data the bit position will be labeled starting with 0 in the least-significant position and move up to n-1.

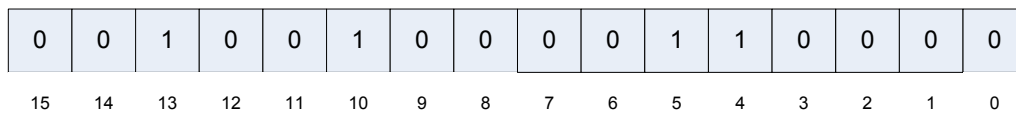The 6502 CPU uses 16-bits to address memory. In this case, a typical memory address will appear as:

0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 85 - 16-bits in a memory address

How to represent positive and negative numbers

The CPU has to represent positive and negative numbers. We will examine this topic assuming that the CPU will be using a fixed number of bits to represent all numbers. That is, if the computer is using 8 bits to represent its numbers the number 0_{10} will be represented as 00000000 in binary. If the CPU is using 8-bit numbers then it can represent 256 distinct values. The formula to use to figure out how many distinct values can be represented with n bits is 2^n. Therefore, for n=8 we get 2^8, which is equal to 256. The 6502 uses two's complement to represent a negative number. In decimal notation the negative of 100 (which is assumed to be positive) is -100. In binary, this is handled by taking the binary number and complement each bit, which just means if a bit is 1 make it 0, if a bit is 0 make it 1 and adding 1 to the complement.

Example: Complement the binary number 01101000.

01101000 ➜ 10010111

To negate or find the negative a binary number, complement it and add 1_2 to the result.

Let's take the example 100_{10} converted into an 8-bit binary number is 01100100. This represents +100 in decimal. The twos-complement of this binary 8-bit number is -100.

$+100_{10} = 01100100_2$ ➜ complement ➜ $10011011_2 + 1_2 = 10011100_2 = -100_{10}$

114

It should also work the other way, negate -100_{10} should result in $+100_{10}$.

$-100 = 10011100_2$ ➜ complement ➜ $01100011 + 1_2 = 01100100_2 = +100_{10}$

As you can see we get the same number we started with.

It is easy to tell from examination if a number is positive or negative. If the left-most digit is 0 then the number is positive, of the left-most digit is 1 then it is negative.

```
+100₁₀  =  01100100₂
           ^
           |_____  left-most is 0, therefore number is positive.
```

```
-100₁₀  =  10011100₂
           ^
           |_____  left-most is 1, therefore number is negative.
```

Example: What is the number 11001001_2 as an unsigned number? as a number in two's complement?

To figure out the decimal equivalent of 11001001_2 as an unsigned number, just expand using powers of 2.

$11001001_2 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^0$

$\qquad = 128 + 64 + 8 + 1 = 201_{10}$

To figure out the decimal equivalent of the signed number   you must first determine if the number is positive or negative. If the number is positive than you can use the same technique as above, otherwise if the number is negative you have to convert to positive and use the same technique.

Since it is negative we will compute the two's complement in order to negate it.

11001001_2 ➜ complement ➜ $00110110_2 + 1_2 = 00110111_2$

$00110111_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 16 + 4 + 2 + 1 = 55_{10}$

which means 11001001_2 must be -55_{10}

Also, note that $+100 + (-100) = 0$. The same should be true if the addition is being done is binary.

```
+100₁₀  =  01100100
-100₁₀  =  10011100
           ---------
          100000000
```

Since our CPU uses eight-bits the result will be 00000000 there is what is called a carry of '1'. You could think that would be a cause of concern but it is not since one of the numbers was negative and the other positive we know that the sum could not be larger than our machine could represent. What is the largest positive number our 8-bit machine could represent? Well, this is easy enough to figure out. Since the number is positive the most significant-bit must be 0, and all 1's in the remaining 7-bits, 01111111. What is this number? The answer can be computed using the binary expansion or we can re-phrase the question, how many distinct values can you represent with 7 bits? We have a formula for that,

2^n, where n=7 so, $2^7 = 128$. So the largest positive number that can be represented with 8-bits is 128 (gee, not much!). In general the formula to use to compute this, is given an n-bit number you can represent 2^{n-1} positive numbers. What is the smallest negative number? This may upset you, but we can represent more negative numbers with the same number of bits than positive numbers.

$128_{10} = 01111111_2$

$-128_{10} = 10000001_2$

There is one more negative number $-129_{10} = 10000000_2$

This is the nature of using two's complement to represent positive and negative numbers. The formula to use to determine the number of negative numbers that can be represented with n-bits using two's complement is $2^{n-1} + 1$

Subtracting in binary

The CPU uses the same hardware to perform subtraction that it uses to compute the two's complement of a number and addition. The reason this works is because:

a – b will give you the same result if you did a + (-b)

Therefore, when subtracting two binary number $b_1 - b_2$, you simply negate b_2 and add to b_1.

Example: Subtract the binary number 00110011 from 01101001.

```
01101001                              01101001
00110011     ➔ two's complement       11001101
```

```
------------                           ------------
                          1|00110110
```

The result (not including the bit that carried over) is 00110110.

Let's do the subtraction in decimal to check our answer:

01101001_2 ➔ 105_{10}
00110011_2 ➔ 51_{10}

$105_{10} - 51_{10} = 54_{10}$ ➔ 00110110_2 which matches the result we got.

Hexadecimal World – base-16

Having to work with long binary numbers can be cumbersome. One byte may not be much but when you start dealing with 16-bit memory addresses, for example, 0010110011110010 you can start to get errors in dealing with so many binary digits. Computer scientists deal with this by using hexadecimal notation or base-16 as a shorthand notation for binary values since it is a very simple to go back and forth from hexadecimal to binary and back.

In the base-16 number system you have 16 digits:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Yes, the number system uses A to represent the equivalent to 10_{10}, B to represent 11_{10}, … F to represent 15_{10}.

The table below shows you the first 16 values in decimal, binary and hexadecimal.

Table 4 - Decimal - Binary - Hexadecimal values

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E

15	1111	F

Converting from binary to hexadecimal

When programmers talk about the value residing in a register in a CPU or in memory they don't sling binary digits around. They could but they don't. They know how error prone it is. If you look at the table above you can see it is faster to write the value 15_{10} in hexadecimal (F) than binary (1111). If you had to look at or write 8-bit, 16-bit, 32-bit numbers all day which would you prefer?

If is quite easy to convert from binary to hexadecimal. Let's assume when we are called to do this that the number of binary bits is always a multiple of 4, that is, it will be 4, 8, 12, 16, …4*n bits in the number.

Given any binary number the easiest was way to convert it to hexadecimal is to replace every four bits with its hexadecimal equivalent from the table above.

Example, given the binary number:

0110111100110001

partition the binary number into groups of four bits

0110   1111   0011   0001

replace each number with its hexadecimal equivalent

6   F   3   1

so the binary number 0110111100110001= 6F31 in hexadecimal.

Converting from hexadecimal to binary

To convert a number from hexadecimal to binary is just the reverse from what we learned in the last section – replace each hexadecimal number with its 4-bit binary equivalent.

Example: Convert A045 into binary.

A = 1010
0 = 0000
4 = 0100
5 = 0101

$A045_{16} = 1010000001000101_2$

Converting from hexadecimal to decimal

Converting from hexadecimal to decimal will require a little bit if work. Since the weight of each hexadecimal digit is a power of 16 then we will need to expand the hexadecimal number using that fact. To help us with the power of 16 let's use the following table:

Power of 16	Value
16^0	1
16^1	16
16^2	256
16^3	4096
16^4	65536

Example: Convert the hexadecimal number 14D to decimal.

$$14D = 1 * 16^2 + 4 * 16^1 + 13 * 16^0$$
$$= 1 * 256 + 4 * 16 + 13 * 1$$
$$= 256 + 64 + 13$$
$$= 333$$

You can use your Windows calculator to double check your work.

Converting from decimal to hexadecimal

Converting a given decimal number to hexadecimal requires that you divide the number by 16 until the quotient is 0. Each remainder is the hexadecimal value from right to left. For example, lets convert 333 to hexadecimal:

333 / 16 = 20 r **13 or D**
20 / 16 = 1 r **4**
1 / 16   = 0 r **1**

From the above you can conclude that $333_{10} = 14D_{16}$

Adding hexadecimal numbers

Subtracting hexadecimal numbers

Exercises

Note: All solution are available in Appendix XXX. If you want to see the details on how a problem was solved then go on-line to
www.brainycode.com/retroprogramming/ex_answers.htm

1. What is the sum of binary 01100010 and 00100111 ?
2. Translate the signed binary number 10001110 into decimal.
3. Translate the unsigned binary number 10001110 into decimal.
4. Subtract 11100111 from 01110110.
5. Write the hexadecimal representation of binary 0100110001110001.
6. What is the sum of hexadecimal 0ABC and 3320?
7. What are the smallest and largest signed integers that will fit into a 16-bit register?

A Detailed look at a microcomputer

The microcomputer is a wonderful invention. The microcomputer is the entire computer system. The microprocessor is where the CPU for a modern computer and console is stored. The one we will be working with are version of the 6502 microprocessor.

A high-level view of a computer/console system created with a CPU is shown below:
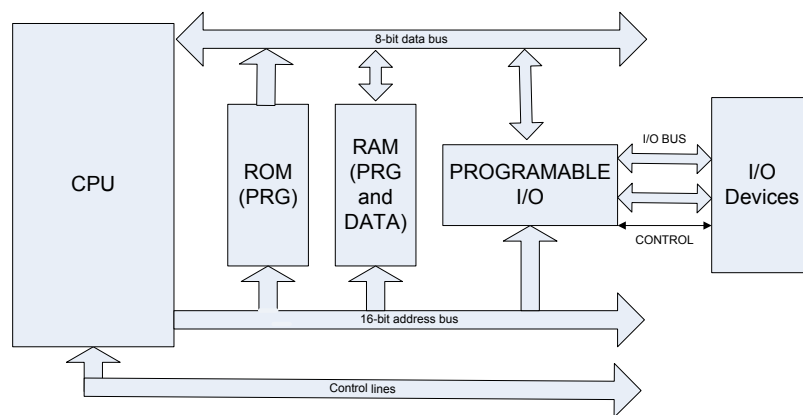
How does a microcomputer works?

The basic components of a microcomputer are:

- The microprocessor – this is the soul of the machine. This is where the program executes – numbers get added, subtracted, compared, moved around.
- Clock – the clock synchronizes all the components in the machine. The clock is the heart of the microcomputer. You can imagine it sends out a pulse and something happens to get information processed on the machine. For a typical 6502 microprocessor the clock beats at about 1MHz, which means it beats about 1 million times a second.
- Memory (ROM) – This is called Read-only memory. It stores permanent information, usually programs you can start using right away as soon as you turn on the machine. Some typical types of programs are monitor (small operating system) and the programming language BASIC. The information in ROM is there and remains there even if your machine is on or off.
- Memory (RAM) – This is called Random Access Memory (or R/W for Read/Write Memory). This is the place where you load your favorite game from a tape drive (old way, no longer used), disk drive (also old-fashioned), hard drive, or flash drive. When you turn off your machine all the information stored in RAM disappears.
- Input/Output devices – There are many ways of getting information to and from the microcomputer. For "input" we use the keyboard, joystick and mouse. For "output" we use the computer monitor or printer.
-

Chapter 4: The Video Mode and Video Buffer

Chapter 5: Game Graphics

Chapter 6: Player Input, Physics, and AI

Chapter 7: Sound Effects

Chapter 8: Programming the Atari 2600

Chapter 9: Creating the game Pong on Atari 2600

Chapter 10: User Assignment: Atari Tank

Chapter 11: Programming the Apple IIe

Chapter 12: Creating the game Crisis Mountain on the Apple IIe

Chapter 13: User Assignment: Apple Robot Wars

Chapter 14: Programming the NES

Chapter 15: Creating the game Abadox on the NES

Chapter 16: User Assignment: NES Sam's World

Chapter 17: Building retro games on Xbox 360

Top 100 Video Arcade Games

1. 1971 - Computer Space
2. 1972 - Pong
3. 1974 - Tank
4. 1976 - BiPlane
5. 1976 - Death Race
6. 1978 - Sea Wolf
7. 1978 - Space Invaders
8. 1979 - Asteroids
9. 1979 - Galaxian
10. 1979 - Lunar Lander
11. 1980 - Asteroids Deluxe
12. 1980 - Battlezone
13. 1980 - Bezerk
14. 1980 - Centipede
15. 1980 - Defender
16. 1980 - Eagle
17. 1980 - Missle Command
18. 1980 - Pac-Man
19. 1980 - Phoenix
20. 1980 - Rally X
21. 1980 - Star Castle
22. 1980 - Tempest
23. 1980 - Warlords
24. 1980 - Wizard of War
25. 1981 - Donkey Kong
26. 1981 - Frogger
27. 1981 - Galaga
28. 1981 - Gorf
29. 1981 - Mousetrap
30. 1981 - Ms. Pac-Man
31. 1981 - Qix
32. 1981 - Stargate
33. 1981 - Vanguard
34. 1982 - Black Widow
35. 1982 - Burgertime
36. 1982 - Dig Dug
37. 1982 - Donkey Kong Jr.
38. 1982 - Jungle King
39. 1982 - Joust
40. 1982 - Moon Patrol
41. 1982 - Pengo
42. 1982 - Pole Position
43. 1982 - Q*bert

44. 1982 - Robotron 2084
45. 1982 - Space Duel
46. 1982 - Time Pilot
47. 1982 - Tron
48. 1982 - Xevious
49. 1983 - Dragon's Lair
50. 1983 - Elevator Action
51. 1983 - Gyruss
52. 1983 - Major Havoc
53. 1983 - Spy Hunter
54. 1983 - Star Wars
55. 1983 - Tapper
56. 1983 - Track and Field
57. 1984 - Punchout
58. 1985 - Choplifter
59. 1985 - Gauntlet
60. 1985 - Ghosts 'n Goblins
61. 1985 - Indiana Jones & The Temple of Doom
62. 1986 - Arkanoid
63. 1986 - Outrun
64. 1986 - Rampage
65. 1987 - 1943 The Battle of Midway
66. 1987 - Double Dragon
67. 1987 - Rastan
68. 1987 - Road Blasters
69. 1987 - Street Fighter
70. 1987 - Zaxxon
71. 1988 - Bad Dudes vs. Dragon Ninja
72. 1988 - Cyberball
73. 1988 - Ghouls 'n Ghosts
74. 1988 - Tetris
75. 1989 - Golden Axe
76. 1990 - Golden Tee Golf
77. 1990 - Raiden
78. 1990 - Rampart
79. 1990 - Smash TV
80. 1991 - Captain America and the Avengers
81. 1991 - King of Monsters
82. 1991 - Terminator 2: Judgement Day
83. 1991 - X-Men
84. 1992 - Lethal Enforcers
85. 1992 - Mortal Kombat
86. 1993 - NBA Jam
87. 1994 - Daytona USA
88. 1995 - Area 51
89. 1995 - Soul Edge

90. 1996 - San Francisco Rush
91. 1997 - House of Dead
92. 1997 - NFL Blitz
93. 1998 - Gauntlet Legends
94. 1998 - House of Dead 2
95. 1998 - Time Crisis II
96. 1999 - Carnevil
97. 1999 - Ferrari F355 Challenge
98. 1999 - Hydro Thunder
99. 2000 - 18 Wheeler: American Pro Trucker
100. 2000 - Gauntlet Dark Legacy
101. 2001 - Ridge Racer V

Chapter 1
1. The First Video Game - http://www.bnl.gov/bnlweb/history/higinbotham.asp
2. Player 1 Stage 1: Bits From the Primordial Ooze - http://www.emuunlim.com/doteaters/play1sta1.htm
3. Lecture on Low Bit Games – http://www.timesup.org/Obsolete/lectureBolt.html
4. http://www.Pong-story.com/intro.htm
5. http://www.computerspacefan.com/SpaceWarSim.htm
6. http://en.wikipedia.org/wiki/Spacewar%21
7. Cohen, Scott. Zap! The Rise and Fall of Atari. McGraw-Hill (1984).
8. Kent, Steven L. The Ultimate History of Video Games. Three Rivers Press 2001.
9. Herman, Leonard. Phoenix: The Fall & Rise of Videogames. Rolenta Press January 1997.
10. Sheff, David. Game Over. Random House 1993.
11. Levy, Steve. Hackers: Heroes of the Computer Revolution. Penguin, 2001.

Chapter 2
1. http://www.mess.org/
2. TBD