How to Design a Computer Game?

Final Project for Design Computing Theory
Prof: Dr. Ellen Do

By

Mandana Sadigh

University of Washington
Fall 2002

Table of Content

ABSTRACT 3
1. INTRODUCTION 3
    1.1 Project Goals 3
2. Taxonomy of Computer Games 3
    2.1 SKILL-AND-ACTION GAMES 4
    2.1.1 Combat Games 4
    2.2 STRATEGY GAMES 6
3. The Game Design Sequence 9
    3.1 CHOOSE A GOAL AND A TOPIC 9
    3.2. RESEARCH AND PREPARATION 9
    3.3. DESIGN PHASE 9
Milestones 9
    3.4. Evaluation of the Design 11
3.5. PRE-PROGRAMMING PHASE 11
3.6. PROGRAMMING PHASE 12
3.7. PLAYTESTING PHASE 12
4. Casey's 10 Commandments of Game Design 13
5. The main elements of a computer game 15
    5.1. Graphics 15
    5.2. Sound 15
    5.3. Interface 16
    5.4. Gameplay 17
    5.5. Story 18
6. Using Flash in Computer Games 19
7. Useful Graphic tools 20
Tools for creating Games 21
8. LINKS 23
9. Designing a New Game 25
10. Future Work 25
11. References and Bibliography 26

ABSTRACT

Computer games are a relatively new innovation in the overall scheme of things. They have been around in different forms since the beginning of computers and in a lot of ways were essential in the route that computers have taken in becoming a part of our every day lives.
Computer games can be classified into different categories. Designing a computer game needs a good knowledge of developing process of a game. In This research I tried to illustrate the design process of a computer game and the main elements that we need to consider when we are designing a computer game.

1. INTRODUCTION

Game design is the creation of the structure and nature of the game and how the game works. This includes the rules and algorithms that are the foundation of the game design, but also includes more esoteric characteristics like the interface that describes how the game player communicates with the game and how the game communicates with the player. For example, in chess, the ways in which the pieces move, the alternating order of play, the rules about capturing pieces, creating new queens from pawns, and capturing the king are all part of the design. In addition, the 8x8 layout of the board is part of the design as well. But the shape of the pieces and their names are part of the game design as well. Why? Because these items, and the decisions involving them, are part and parcel to how the game is communicated to the game player.

1.1 Project Goals

My scholarship goals for this project are as follows:
Understand different categories of computer games.
To see what the development process of designing a computer Game is.
An investigate on the current tools for developing a computer Game.
To Design and develop a computer Game.

2. Taxonomy of Computer Games

Thousands of computer games are commercially available on a variety of hardware configurations. These games present a bewildering array of properties. Many show close similarities. Most possess some unique design feature. Given this large sample of games, we can learn a great deal about game design by establishing a taxonomy of computer games. A taxonomy would illuminate the common factors that link families of games, while revealing critical differences between families and between members of families. Chris Crawford has divided computer games into two broad categories: skill-and-action

("S&A") games (emphasizing perceptual and motor skills) and strategy games (emphasizing cognitive effort). Each major category has several subcategories.

2.1 SKILL-AND-ACTION GAMES

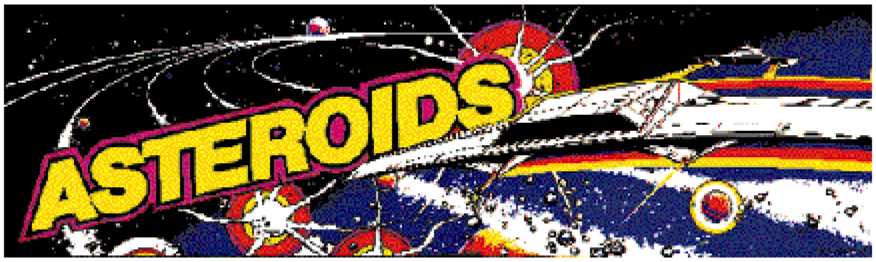
This is easily the largest and most popular class of computer games. Indeed, most people associate all computer games with skill-and-action games. This class of games is characterized by real-time play, heavy emphasis on graphics and sound, and use of joysticks or paddles rather than a keyboard. The primary skills demanded of the player are hand-eye coordination and fast reaction time.

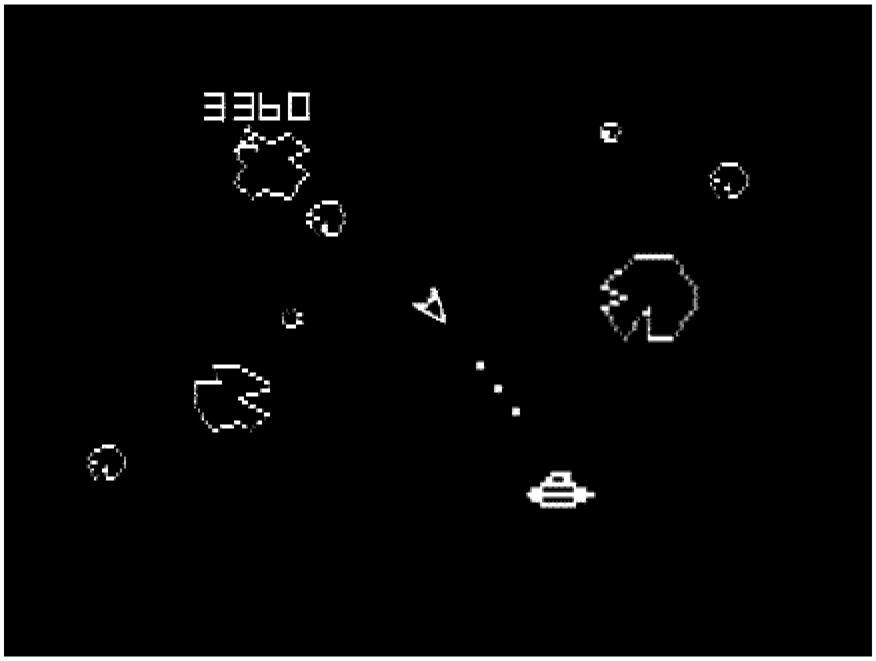
Skill-and-action games can be grouped into six categories: combat games, maze games, sports games, race games, and miscellaneous games.

2.1.1 Combat Games

Combat games all present a direct, violent confrontation. The human player must shoot and destroy the bad guys controlled by the computer. The challenge is to position oneself properly to avoid being hit by the enemy while shooting him. These games are immensely popular; they are Atari's forte. There are many variations on this theme, most arising from variations on the **geometry** of the situation or **the weaponry** of the opponents.

STAR RAIDERS and SPACEWAR can be compared on these bases of geometry and weaponry. ASTEROIDS is another example of combat game too.





MISSILE COMMAND is another combat game with several interesting twists. First, the player must defend not only himself but also his cities from descending nuclear bombs. Second, the game is a purely defensive game in that the player never has the opportunity to attack his enemy. Third, while shots in other games are very rapid events, the shooting process in this game is slower and takes time to develop because the missiles must fly to their targets before detonating. Because the time between firing and impact is so long, the player must plan his shots with greater foresight and make use of multiple explosions. Thus, although this is a skill-and-action game, there are more strategic elements involved than in many games of this category. Combat games have always been at the heart of computer gaming. Players never seem to tire of them; it appears that they will be around for a long time to come.

2.1.2. Maze Games

The second subgroup of S&A games is the set of maze games. PAC-MAN (trademark of Namco) is the most successful of these. The defining characteristic of the maze games is the maze of paths through which the player must move. Sometimes one or more bad guys pursue the player through the maze. The appeal of maze games can be attributed to the cleanliness with which they encapsulate the branching structure that is a fundamental aspect of all games.

2.1.3. Sports Games

These games model popular sports games. They are anachronisms derived from the early days of computer game design when computer games had no identity of their own. People without original ideas for games fell back on the sports games as models around which to design. This also served a useful marketing purpose: why would a conservative consumer buy a game with a title and subject completely alien to his experience? Better to offer him a game he is already familiar with. Thus we have games based on basketball, football, baseball, soccer, tennis, boxing, and others. All of these games take liberties with their subject matter to achieve playability. The most enjoyable aspects of the computer game have very little to do with the real game. This is fortunate, for a slavish attempt at replication would have produced a poor computer game. Only by substantially altering the original games were the authors able to produce a decent design. Even so, sports games remain the wallflowers of computer gaming.

2.1.4. Race Games

Some computer games involve a straightforward race. Most of these games allow the player to move at constant speed, but extract time penalties for failure to skillfully negotiate an assortment of hazards.

2.1.5. Miscellaneous Games

There exist a number of games that do not fit into this taxonomy very well. So we put them in this category.

2.2 STRATEGY GAMES

Strategy games comprise the second broad class of computer games. These games emphasize cogitation rather than manipulation. I do not mean to imply that S&A games are devoid of strategic content; some S&A games do indeed have a strategic element. The major distinguishing factor between strategy games and S&A games is the emphasis on motor skills. All skill-and-action games require some motor skills; strategy games do not. Indeed, real-time play is rare in strategy games. Strategy games typically require more time to play than S&A games. Strategy games can be divided strategy games into six categories: Adventures, D&D games, wargames, games of chance, educational games, and interpersonal games.

2.2.1Adventures

These games derive from one of the oldest computer games, called "Adventure". In these games the adventurer must move through a complex world, accumulating tools and booty adequate for overcoming each obstacle, until finally the adventurer reaches the treasure or goal. Scott Adams created the first set of Adventures widely available for personal computers; his software house (Adventure International) is built on those games.

Adventures are closer to puzzles than to games. Puzzles are distinguished from games by the static nature of the obstacles they present to the player. Adventures present intricate obstacles that, once cracked, no longer provide challenge to the player. It is true that some adventures push closer to being games by incorporating obstacles such as hungry dragons that in some way react to the player. Nevertheless, they remain primarily puzzles.

2.2.2. D&D Games

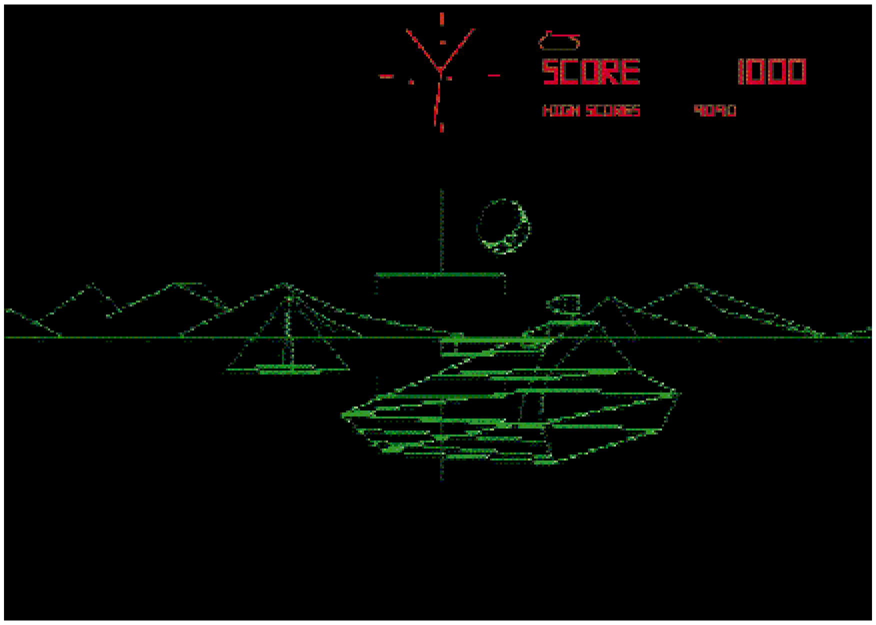
A completely independent thread of development comes from the D&D style games. Fantasy role-playing was created by Gary Gygax with Dungeons and Dragons (trademark of TSR Hobbles), a complex noncomputer game of exploration, cooperation, and conflict set in a fairytale world of castles, dragons, sorcerers, and dwarves. in D&D, a group of players under the guidance of a "dungeonmaster" sets out to gather treasure. The game is played with a minimum of hardware; players gather around a table and use little more than a pad of paper. The dungeonmaster applies the rules of the game structure and referees the game. The dungeonmaster has authority to adjudicate all events; this allows very complex systems to be created without the frustrations of complex rules. The atmosphere is quite loose and informal. For these reasons, D&D has become a popular game, with endless variations and derivatives.

2.2.3. Wargames

A third class of strategy games is provided by the wargames. Wargames are easily the most complex and demanding of all games available to the public. Their rules books read like contracts for corporate mergers and their playing times often exceed three hours. Wargames have therefore proven to be very difficult to implement on the computer; we have, nevertheless, seen entries.

The computer wargames available now fall into two distinct groups. The first group is composed of direct conversions of conventional boardgames. COMPUTER BISMARK, COMPUTER AMBUSH, and COMPUTER NAPOLEONICS (trademarks of Strategic Simulations, Inc.) are examples of this group of games.

The second group of computer wargames are less slavish in their copying of board wargames.



2.2.4. Games of Chance

Games of chance have been played for thousands of years; their implementation onto computers is therefore quite expectable. They are quite easy to program, so we have seen many versions of craps, blackjack, and other such games. Despite their wide availability, these games have not proven very popular, most likely because they do not take advantage of the computer's strong points.

2.2.5. Educational and Children's Games

The fifth category of strategy games is that of the educational games. Although all games are in some way educational, the games in this set are designed with explicit educational goals in mind. Educators are becoming more aware of the motivational power of computer games. Several of the classic computer games are educational: HANGMAN, HAMMURABI, and LUNAR LANDER are the three most noteworthy of these early educational games. ROCKY'S BOOTS (trademark of The Learning Company), a children's game about Boolean logic and digital circuits. The child assembles logic gates to create simulated logical machines.

2.2.6. Interpersonal Games

Is a class of games that focus on the relationships between individuals or groups. One such game explores gossip groups. The player exchanges gossip with up to seven other computer-controlled players. The topic of conversation is always feelings, positive or negative, expressed by one person for another.

3. The Game Design Sequence

Game design is primarily an artistic process, but it is also a technical process. In the first place, game design is far too complex an activity to be reducible to a formal procedure. Furthermore, the game designer's personality should dictate the working habits s/he uses. So this procedure is not as a normative formula but as a set of suggested habits that the prospective game designer might wish to assimilate into her existing work pattern.

3.1 CHOOSE A GOAL AND A TOPIC

A game must have a clearly defined goal. This goal must be expressed in terms of the effect that it will have on the player. Once you have settled on your goal, you must select a topic. The topic is the means of expressing the goal, the environment in which the game will be played. It is the concrete collection of conditions and events through which the abstract goal will be communicated. Most game designers start off by selecting their topic, with their goals subordinated to their topic.

3.2. RESEARCH AND PREPARATION

With a goal and topic firmly in mind, the next step is to immerse yourself in the topic. Read everything you can on the topic. Study all previous efforts related to either your goal or your topic. What aspects of these earlier efforts appeal to you? What aspects disappoint or anger you? Make sure that you understand the mechanics of the environment your game will attempt to represent. Your game must give the authentic feel, the texture of the real world, and this can only be achieved if you firmly understand the environment of the game.

3.3. DESIGN PHASE

You are now ready to begin the concrete design phase. Your primary goal in the design phase is to create the outlines of three interdependent structures: the I/O structure, the game structure, and the program structure. The I/O structure is the system that communicates information between the computer and the player. The game structure is the internal architecture of causal relationships that define the obstacles the player must overcome in the course of the game. The program structure is the organization of mainline code, subroutines, interrupts, and data that make up the entire program. All three structures must be created simultaneously, for they must work in concert. Decisions primarily relating to one structure must be checked for their impacts on the other structures.

Milestones

A typical set of design milestones might include the following:

Conceptual design phase: Game Overview Document (GOD), GOD review, GOD Revision

Initial Design Phase: Initial Design Document (IDD), IDD Review, IDD Revision

Expanded design Phase: Expanded Design Document (EDD), EDD Review, EDD Revision

Final Design Phase: Final Design Document (FDD), FDD Review, FDD revision

Now we define all three different structures in design phase.

I/O Structure

I/O is composed of input and output. Unlike human languages, the two are not symmetric. The computer has two means of output to the human: graphics on the screen and sound. In the future, we may see more exotic devices for output for games, but for the moment these are the two most common.

Don't make the common mistake of creating cute graphics solely to show off your ability to create cute graphics. Graphics are there for a reason: to communicate. Use graphics to communicate to the user forcefully and with feeling, and for no other reason. Plan functional, meaningful graphics that convey the critical game information while supporting the fantasy of the game.

The input structure lies at the heart of a fundamental dilemma all game designers must face. An excellent game allows the player to interact heavily with his opponent, to invest a great deal of his personality into the game. This requires that the game offer the player a large number of meaningful options, enough options that the player can express the nuances of his personality through the choices he makes. Yet, decisions must be inputted, and a large number of options seem to require an extensive and complicated input structure, which could well be intimidating to the player. Our dilemma, then, is that an excellent game seems to require a hulking input structure.

The I/O structure is the most important of the three structures in a computer game, for it is the face of the game that the player sees. It is the vehicle of interaction for the game. It is also the most difficult of the three structures to design, demanding both human sensitivity and complete technical mastery of the computer. Give it the care it deserves.

Game Structure

The central problem in designing the game structure is figuring out how to distill the fantasy of the goal and topic into a workable system. The game designer must identify some key element from the topic environment and build the game around that key element. This key element must be central to the topic, representative or symbolic of the issues addressed in the game, manipulable, and understandable.

Many games employ multiple key elements. For example, most combat games include both movement and shooting. This is not necessarily bad; if both key elements are kept simple, or if one key element retains primacy, the game can be successful. However, too many key elements violating too many of these principles will rob the game of its focus.

Your main problem with creating the I/O structure is overcoming constraints; your main problem with creating the game structure is realizing possibilities. Concentrate on providing enough color to guarantee that the game will convey the authentic feel of reality. Keep your sense of proportion while adding details. It will do your game no good to provide exquisite detail and accuracy in one sphere while overlooking the most fundamental elements in another sphere.

A very common mistake many designers make is to pile too many game features onto the game structure. In so doing, they create an overly intricate game, a dirty game. Designing the game structure is emotionally very different from designing the I/O structure. While designing the I/O structure, the designer must thread a precarious path between the Scylla of expressive power and the Charybdis of expressive clarity, even while the storms of hardware limitations toss her design to and fro. While designing the game structure, the designer finds herself on a placid sea stretching flat to the horizon. The challenge taunting her now is "Where do you go?"

Program Structure

The program structure is the third object of your design attentions. This structure is the vehicle which translates the I/O structure and game structure into a real product. One of the most important elements of the program structure is the memory map. You must allocate chunks of memory for specific tasks. Without such safeguards, you may end up expending excessive quantities of memory on minor functions, and having insufficient memory remaining for important tasks. Definitions of critical variables and subroutines are also necessary. Finally, some documentation on program flow is important.

3.4. Evaluation of the Design

You now have three structures in hand: the I/O structure, the game structure, and the program structure. You are satisfied that all three structures will work and that they are compatible with each other. The next stop in the design phase is to evaluate the overall design for the most common design flaws that plague games. The first and most important question is: does this design satisfy my design goals? Does it do what I want it to do? Will the player really experience what I want him to experience? If you are satisfied that the design does pass this crucial test, proceed to the next test. Design evaluation is very important, because the next step is game programming. So before committing to the programming, you should evaluate your design first.

3.5. PRE-PROGRAMMING PHASE

If the game has made it this far, you are now ready to commit your ideas to paper. Until now your documentation has been sketchy, more along the lines of notes and doodles than documents. Now you are ready to prepare your complete game documentation. First, commit all of your design results from the previous phase to paper. Define the I/O structure and the internal game structure. The tone of this documentation should emphasize the player's experience rather than technical considerations. Compare this first set of documents with your preliminary program structure notes; adjust the program structure documents if necessary.

3.6. PROGRAMMING PHASE

Both application programming and system programming is necesery for programming a game.

3.7. PLAYTESTING PHASE

Ideally, playtesting is a process that yields information used to polish and refine the game design. In practice, playtesting often reveals fundamental design and programming problems that require major efforts to correct. Thus, playtesting is often interwoven with a certain amount of program debugging.

Sometimes playtesting reveals that the game is too seriously flawed to save. A nonfatal, correctable flaw is usually a matter of insufficiency or excess: not enough color, too many pieces, not enough action, too much computation required of the player. A fatal flaw arises from a fundamental conflict between two important elements of the game whose incompatibility was not foreseen. You must have the courage to trash a game with such a fatal flaw. Patching after the game is programmed can only achieve limited gains; if the game is badly deformed, abortion is preferable to surgery.

One of the last tasks you must perform before releasing the game is the preparation of a game manual. Manuals are frequently given short shrift by just about everybody associated with computer games. This is a serious mistake, for the manual is a vital element in the overall game package. A computer has many limitations; some can be overcome with a good manual. Much of the static information associated with a game can be presented in a manual. The manual is also an excellent place to add fantasy support elements such as pictures and background stories. Finally, a well-written manual will clear up many of the misunderstandings that often arise during a game.

3.8. POST-MORTEM

Once the program is out, brace yourself for the critics. They will get their filthy hands on your lovely game and do the most terrible things to it. They will play it without reading the rules. If it's a strategic game, they will castigate it for being insufficiently exciting; if it's an S&A game, they will find it intellectually deficient. Don't let these critics affect you. Most critics are far less qualified to criticize programs than you are to write them. A very few critics with the larger publications are quite thoughtful; you should pay attention to their comments. With most critics, though, you should pay heed only to views shared by three or more independent critics. Remember also that even a good critic will roast you if your goal is not to his taste.

.

4. Casey's 10 Commandments of Game Design

Travis S. Casey, who is one game designer has 10 commandments of game designs, these are:

1. WRITE GAMES YOU LIKE.
Write games on the things YOU like and hopefully your enthusiasm will come through.

2. EXPERIENCE IS THE BEST TEACHER.
The best way to learn game design is to read a lot of games, play a lot of games, analyze those games, and design your own games or game extensions. Since my main experience is with RPG's, my examples will come from them, but the idea is applicable to all kinds of games.
3. TEST, TEST, AND TEST SOME MORE.
Play, test your games. Play them as much as possible; get other people to play them, preferably without you around, and talk to them afterwards. (Having other people play the game without your presence is called blind-testing, BTW.)
In addition, think about your rules. Consider hypothetical situations and work out the probabilities involved. Repeat the calculations under different conditions; different terrain, at night, etc. This will help you find places where you've made a mistake in your math or made a bad assumption.

4. LEARN YOUR BACKGROUND.
If you want to write a medieval fantasy game, read medieval literature and history. Read books about magic. Read existing medieval fantasy games. Similarly for any other type of game; if you're making a game set in the Vietnam war, read official histories of the war, unofficial histories, and especially analyses of strategy and tactics.
All this background is useful in several ways: for one thing, it will help you in creating realistic rules. For another, it lessens the chance that you will make a major mistake in terminology or background. And, of course, the material is often interesting in itself. If you're not interested in learning about X, why are you writing a game about it anyways?

5. FORMAL EDUCATION.
Take a class in introductory probability and statistics. Try reading some on the mathematical theory of games.
If you want to do computer games and haven't already taken any programming classes, take a few. You may not learn anything about how to program, but a good class will teach you some things about how to organize a program to make maintenance and bug-finding easier.
While you're at it, build up a "reference library." This is a set of games and books on whatever subject you're making your game on. This will help immensely when

inspiration strikes at 3 AM and the library is closed.

6. TAKE TIME OFF.

A game is like a child; when it's first born, it's parents think it's perfect. Take some time away from your game to keep from getting burnt out and to get a fresh perspective on it. Repeat this from time to time.

7. KEEP RECORDS.

Make sure you have more than one copy of your game. If you're typing the rules on a computer, keep one copy on the hard drive, one on a floppy, and a printout of a fairly recent version (say, print it out once a month, or once a week if you're working really fast).

In the same vein, keep copies of older versions as well. You may find in playtesting that your new idea isn't as good as the old one was, and what are you going to do now if you've trashed the old copy? Keep at least one copy of the last version around, in addition to the copies of the current version.

8. DON'T FORGET THE INCIDENTALS.

Great rules and writing are nice, but a good visual presentation will do wonders for your sales. If you're doing it yourself, learn something about desktop publishing, and either find some ready-made illustrations (for example, in the Dover clip art stuff or US government publications) or find someone to draw a few
illustrations for you.

Find a printer and talk to him/her; discuss ways to do what you want as inexpensively as possible. A lower price will help sales some, and lower expenses will help your profits.

9. REMEMBER, IT'S ONLY A GAME.

Don't ignore real life to work on your game. If someone doesn't like your game, don't take it personally. Don't get worried about people stealing your ideas. Remember rule #1 and have fun with what you're doing.

10. More Advice

And, here's some extra advice from Tom Lehmann, president of Prism Games:

Incremental innovation often works best. If everything in your game is familiar, it will feel stale. If everything is very different, it may feel strange. A single clever twist on a familiar theme is good but may result in your game being viewed as a "variant"; TWO clever twists on familiar ideas makes a game feel fresh while still easily accessible. So don't try to re-invent the wheel. Instead, try to present existing ideas cleanly and simply while extending a few key concepts in new and interesting directions.

Revise and Polish your game ideas. Testing serves not only to clean up bugs in the game system and rules presentation but also as the forum in which the game designer may discover the game that he or she *really* wanted to put forth, as opposed to the one they actually have put together. If you leave testing to the end, this discovery may not do you any good. If you test early and often with an eye towards trying to figure out just what the game really is about, you can often improve a game considerably.

Both Alpha and Beta testing are strongly recommended in games developing.

"Alpha" testing can be viewed as asking the questions: "Is there a game here?" and "Have I found it yet?" "Beta" testing can be viewed as asking the questions: "Is this the best way to achieve this effect?", "Is this game mechanic essential -- or can it be simplified or eliminated?" and "Are all the major game systems working together to impart the game experience I want?" "Gamma" testing asks the question: "How can I improve game balance and presentation?" Too many designers stop after Alpha (producing an intriguing but shoddy game) or go from Alpha to Gamma, skipping Beta (producing games that are ok but not great).

5. The main elements of a computer game

The computer games can be break into 5 elements. Those are graphics, sound, interface, gameplay and story. I'll briefly go over a definition of each one and then go into each one more deeply.

5.1. Graphics

Graphics are the first thing that will strike a player (or potential player) when they see your game. Graphics consist of any images that are displayed and any effects that are performed on them. This includes 3D objects, textures, 2D tiles, 2D full-screen shots, Full Motion Video (FMV), statistics informational overlays and anything else that the player will see.

Every game wants to be different, but at the same time if you are too different then the learning curve to learn how to control your game may be too high and you may lose potential players. Like almost everything in life this is a tradeoff and will simply take testing to figure out what works best for your game.

2D games have an immediate advantage in this regard as drawing 2D screens are almost always significantly less then 3D. If you want your game to be centered on gameplay and your game's internals rather than a fancy 3D interface this is still a viable option.

5.2. Sound

Sound consists of any music or sound effects that are played during the game. This includes starting music, CD music, MIDI, MOD tracks, Foley effects (environmental sound), and sound effects.

Sound is just as crucial as any other point in your game. Designers now have the capabilities are CD quality music and sound effects and can finally use them.

Sound is more immersive than graphics. Sound is more immersive than graphics. While graphics will draw you in to a scene, the sound going on in the background will create a reality in the player's mind that can never be done with graphics alone.

In technical terms the sound is going straight to their "subconscious mind" while the visual parts of the game are being picked at by the "conscious mind". To explain this detail simply, what they are not paying attention to is affecting them more because they're not paying attention to it.

5.3. Interface

The interface is anything that the player has to use or have direct contact with in order to play the game. The interface is not as straightforward as the above definitions as it goes beyond simply the mouse/keyboard/joystick, which is only the first contact the game has with the player. The interface includes graphics that the player must click on, menu systems that the player must navigate through and game control systems such how to steer or control the pieces in the game. Half of a game's "AI" (Artificial Intelligence) is also The key to any interface is in its simplicity. You want buttons that are next to each other to have similar functions. Without this level of organization in your interface your game can become difficult to play and that is the ultimate breach in immersivenessrelated to interface. To keep an interface simple you have to think about someone who has never used a computer before and you are trying to explain what to do. You want to group things in ways that make sense and keep the controls down to an absolute minimum.

While all 5 aspects to video games are important, this one you, as a designer, need to pay special attention to. This is the connection between the player and the game. The graphics may impress them, the sound may wow them, but if there isn't a properly designed interface they may as well be watching a movie.

Interface AI

Where the interface and game AI combine is how your game handles and interprets player input.

When the player in Command & Conquer selects a unit and then tells it to go a destination he is expecting it to go the same way he would go to that destination from the unit's position. This is an important key to remember. Most of game AI is about the player's expectations. You are trying to simulate things that happen in real life in your game. The moment that you start deviating from the player's expected reactions you are breaking the immersiveness.

Losing immersiveness with the interface is actually worse then losing immersiveness with any of the other elements of a game. When a player is playing a game they are connecting with it. They are using the game's controls as an extended portion of their body, in much the same way as a baseball player uses the bat or glove as an extension of his body during a baseball game. When your game's interface is not allowing the player to do what he wants or expects to do then you are creating a barrier between the game and the player.

This results in the player fighting the interface to play the game, and nothing could make the player angrier at your game or walk away faster then an interface that isn't letting him play. If you have ever been in an arcade when a joystick or button wasn't functioning properly I'm sure you've seen someone who was really into their game flip out and kick the hell out of the machine because they lost the game over an interface glitch.

Easy to start but hard to master

A perfect example of this is Street Fighter 2 (SF2). SF2 allows anyone to play for the first time and theoretically they can beat an extremely good player, just by hitting buttons and intuitively moving around. The controls for SF2 are impeccable, they are fluid and allow you to move from one move to another in a very graceful manner once you and your hand learn the timing of it. SF2 is one of the rare games that scales to all levels of players.

5.4. Gameplay

Gameplay is a fuzzy term. It encompasses how fun the game is, how immersive it is and the length of playability. The second half of the game's AI is related to game-play.

Most of the successful games are the games that anyone could walk up off the street and play, but to really play well takes a great deal of work. The interface allows the player to grow into the game, like SF2.

Player's do not like to do the same thing over and over again. Games that players appear to be doing the same thing over and over again may actually be deceiving. Something is going on that allows them renewed enjoyment with each game or they would stop playing.

AI in Gameplay

The difference between game AI in the interface and game AI in gameplay is that the interface AI controls what the player is trying to do and gameplay AI controls how computer plays against the player. AI in games is not the usually same as scientific AI that you may have heard about. Although quite recently some game developers have been using real AI forms such as Neural Networks and Genetic Algorithms in games they are predominately not used in games yet as they normally require more time and processing power then can be currently allotted to them. What game's AI is instead is just the behaviors of objects in the game to simulate their intelligence.

When designing an artificial intelligence (AI) for a strategy game, one must keep clearly in mind the final goal, i.e. winning the game (actually the goal is to entertain the customer, but the sub-goal of trying to win the game seems more appropriate here). Normally, winning the game is accomplished by reaching a set of victory conditions. To achieve these victory conditions, the computer needs to control a divergent set of resources (units and other decisions) in a coordinated and sophisticated manner.

5.5. Story

The game's story include any background before the game starts, all information the player gains during the game or when they win and any information they learn about characters in the game.

Immersiveness is very important in games. Immersiveness is the ability of a game to capture the player's attention and make him feel like he is actually in the game. Books and movies have immersive qualities as they can make their audience feel involved in the story by immersing them in the story's world and making them empathize with the characters. In designing games you must try to use the known methods of immersion that books and movies use as well as qualities that are unique to computer games.

Configurability

It's good to have a lot of options in your game that players can configure. This way they can tune your game in ways that suit them better than the shipped configuration. However, configuring is for advanced players, not beginners. Do not expect or ask your beginning players to configure anything you can avoid. They may not understand the implications of what you're asking them and all they want to do is play the game.

Do not expect your players to pick up your manual before they start your game. Design everything so that the first time user can figure things out without ever touching your manual. When designing your standard interface find the options that are the easiest to use for the greatest amount of players, this may not always be the same as the easiest for advanced players.

Variety

Player's do not like to do the same thing over and over again. Games that players appear to be doing the same thing over and over again may actually be deceiving. Something is going on that allows them renewed enjoyment with each game or they would stop playing.

To use Quake death matches as an example, these players run around killing each other over and over again for hours, sometimes with only a few seconds of gameplay between deaths. To the casual onlooker this may seem like a tedious time wasting event at best, but to the players there is an interaction taking place for supremacy and companionship at the same time.

The players are fighting to see who is the best and at the same time they enjoy the rival company of other people who have similar interests with them.

Real AI in games

The subject of real AI has come up a lot recently with regards to being placed in computer games. There are several different types of AI's but Neural Networks and Genetic Algorithms (or evolution) are normally the two I hear about most. Both of these either seem to require more resources than current machines can offer or take too long to learn and adapt to the players actions and so have rarely been seen in any games.

It is my assertion that for quite some time still making your own AI routines based on your own intelligent decisions will give you better output then having an AI routine eventually do it for you.

Characters

People tend to identify with other people. A lot of people like to identify with things they would like to be which is why the hero character is so popular. There are many different kinds of hero's though, and more recently the vulnerable hero has come to rise as one of the preferred hero types. What is a vulnerable hero? It's someone who could be an ordinary person but is put in extraordinary circumstances and through fate is forced to survive. More strictly defined it is someone who does something heroic but is not a superman. He will get hurt, he will have doubts about his abilities, he will wish he didn't have to do things, he will be every that the stereotypical tough guy won't do. In short he will be a real person with fears, doubts, good sides and bad.

Goals

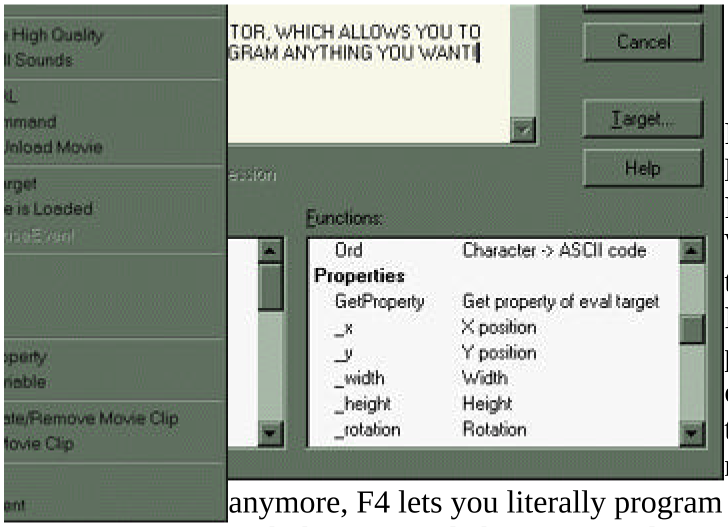
Player's like to feel like their accomplishing something. If you can give them a goal such as saving the world, gaining some great artifact or piece of information or just saving their own lives they feel like they've accomplished something when their done.

Try to make your story so that the extreme things do not stand out too far from anything else. Make sure that you could believe the series of events that happen after your normally unbelievable event. It can be assumed that tomorrow Dog Men from outer space will not invade Earth, but if that is your game's premise following it up with them doing the cha-cha with Clint Eastwood is not going to make any player more receptive to your story.

Attention to detail is one of the other keys of developing a story, do not leave any holes in it, as your player will notice them and they will bother him. If in your story the villain recovers the lost artifact and locks it away in his fortress which is then destroyed the next scene should not contain you finding the artifact in a tree. If this does happen your story then you need an explanation, such as it was blown out of the fortress with the destruction and must have landed there. Highly unlikely, but at least you have covered a plot hole.

6. Using Flash in Computer Games

The tools and techniques of Flash are far more welcoming to the Photoshop- and vector-literate designer than HTML, while its powerful ActionScript facilities provide you with the means to create advanced interactive content that assembles on the fly, in response to how and where your viewer clicks their mouse.
The stuff is so powerful, in fact, that you can create captivating games within 50K odd, which is more than reasonable for a phone line, and it's a fact that a well-published game can drive your site stats up in exponential leaps.
The art of creating great Flash games is to keep things simple, at least initially, so you don't get lost in ever decreasing circles of complexity.



TOR, WHICH ALLOWS YOU TO
GRAM ANYTHING YOU WANT!

Ord	Character -> ASCII code
Properties	
GetProperty	Get property of eval target
_x	X position
_y	Y position
_width	Width
_height	Height
_rotation	Rotation

Flash is the perfect tool for creating things like interactive menus and animated cartoons. Flash and Director work very well together. We use Flash content inside of Director to take advantage of the strong points of both. Flash 4 (F4) opens the doors to flash programming. New scripting capabilities offer you a real tool when creating games. In the past, complex, fun games have been restricted to Macromedia Director. Not anymore, F4 lets you literally program like you would in any other language, but with the ease, and object oriented interface. Actions have never been better in F4. The new actions simplify some task that may have taken hours to do. They also offer new possibilities for interactivity and animation. Actions and scripting are major hot new features in F4.

F4 now has the option of rounded squares. In the past you would have to create the rounded corners with circles, and very carefully place them in the deleted corners of the squares. Not anymore, now you can set the radius of the corner, and F4 will do the rest! Definitely a HOT new feature that advances your F4 drawing.

7. Useful Graphic tools

Here is a list of some of the graphical tools which could be helpful in developing graphical environment of games.

Adobe Photoshop 7.0
Corel Draw 8
Eye Candy Plug-in
Macromedia Fireworks 2
Macromedia Fireworks MX
Macromedia Flash 4
Macromedia Flash 5
Macromedia Freehand 10
Paint Shop Pro 6
PhotoImpact 7
Ulead COOL3D
Ulead GIF Animator
Ulead WebRazor Pro
XARA 3D
XARA Webster

Tools for creating Games

If you are interested in designing and developing a game here are some tools and some useful references. I have listed some of them here.

3D Game Maker
The 3D Gamemaker is the easiest 3D game maker ever! It requires no programming experience and yet still allows quality results to be created.
3D GameStudio
3D GameStudio is the leading authoring system for 2D and 3D computer games. It combines a programming language with a high-end 3D engine, a 2D engine, a map and model editor and huge libraries of 3D objects, artwork and pre-made games. It was never easier to create 1st person games, 3rd person games, role playing games, side scrollers, flight simulators, spaceship simulators, board games, sports games, 3D pinball... real-time presentations, v irtual exhibitions... and 3D applications never seen before!
3DRad
3D Rad is a programmable, user-friendly, 3D real-time engine. It is not a SDK, a GDK, a C++ library or a wrapper, so you don't need a C++ compiler! It is a complete developing tool, producing stand-alone executables!
Amit's Game Programming Information
Game Programming topics including AI, path finding, game design, objects, hexagonal tiles, threads, and text games.
CDX
CDX is a free GDK (Game Development Kit) which is comprised of a set of C++ wrapper classes for writing Windows games. It is built on top of Microsoft Windows and DirectX technology and offers simple to use C++ wrappers for all aspects of game development. This includes things like sprites, tiles, scollable maps, alpha blending and even 3D primitives (using Direct3D). CDX takes care of the low-level details of using DirectX, providing you with an easy to use toolkit for implementing your own games using simple yet flexible C++ classes. With only 2 dozen classes, CDX is quick to learn.
CharacterFX
CharacterFX is a shareware skeletal animation package, created for game developers and mod makers. Greatest features: inverse kinematics, weighted vertices and integrated scripting language for quick and easy automation and extensions.
Digital Game Developer
Digital Game Developer is a free online publication dedicated to game development professionals. News, tutorials, reviews, profiles and industry statistics for console, PC, Mac, arcade and web game developers and game publishers.
Discreet's Open Source Projects
Discreet's new Open Source Projects site resides on a WebBoard dedicated to the download and upload of source code components. Next Generation-relevant portions of 3D Studio MAX are now available from Discreet as Open Source projects and will eventually be available as direct downloads from the Discreet Game Zone. According to Discreet, programmers are free to use, modify, and integrate this software into any application they are developing, according to the terms and policies of the Open Source license. In the future, the game developers' interests will dictate what other portions of

3D Studio MAX, as well as Character Studio and Lightscape authoring tools, will be released to Open Source.

Druids GL Journal

This site follows chronicles the problems, and solutions through a game development project. Contains some excellent articles about OpenGL programming, including terrain generation, voumetric fog, and more.

DXWrapper

DXWrapper is a free GDK (Game Development Kit) which contains a set of classes which wrap DirectX's and Win32's functionality. It serves as a layer between your application and low level DirectX and Win32.

East Coast Games

East Coast Games is an excellent resource for game programmers and people who want to become game programmers.

Game Creation System

The Game Creation System is a complete system for making 3D action games which you can sell or give away for free. The Game Creation System is an incredibly fun program which allows ordinary people to make 3D action games. No programming is necessary.

Game Developer Search Engine

Search Engine, and category link listings covering: 3D & Graphics, 3Dfx, DirectX, Engines, OpenGL Articles & Tutorials, Books, FAQ, Magazines Design & Programming, AI, Game Companies, Publishers, Retailers, Reviewers Hardware, Sound, Peripherals, Video, Forums, Groups, Mailing Lists Languages & API, C/C++, Delphi, Java, VB and more!

Game Exchange

Nichimen Graphics' Game Exchange is an open standard for describing 3D objects, textures, and animation in an easy to understand ASCII format--a format that can accommodate the data requirements of just about any 3D engine. Generate content once, then convert that data to each of your target platforms, using different converters. Use the Game Exchange API (written in C++, and completey revamped for version 2.1) to create your own!

GameDev.net

Gamedev.net is the ultimate online resource for game developers. Featuring a huge reference library, daily news updates, the most active game development forums anywhere, and an online game development workshop!

Genesis 3D

Genesis3D is a state of the art 3D graphics engine built for high performance real-time rendering. Genesis3D offers advanced lighting features, 3D modeling support, seamless soft-skin polyonal characters, and many other innovations. Genesis3D provides the depth and versatility required to produce more than just great games. It is intended for many types of applications including education, business, and the internet.

GL Game Developer

Game development using OpenGL. Some tutorials, and code.

International Game Developers Association

The International Game Developers Association is an Independent, non-profit professional organization for developers of entertainment software.

International Game Developers Network

IGDN is an unincorporated membership association for the game developer community.

Photon Effect
libCON - is a C++ API including classes abstracting 2D and 3D graphics, sound, and input. The main target application for the set of libraries presented here is Games ranging from 2D arcade, to board games to 3D flight simulators. Other applications that require graphics & sound can easily be created as well.
DirectX
DirectX is Microsoft's answer to graphics and animation for Windows. It has several parts, including DirectDraw, DirectInput, Direct3D, DirectSound, etc. It has some advantages: drivers are provided by hardware manufacturers and support hardware acceleration features. It has some disadvantages as well. The biggest disadvantage, in my opinion, is how convoluted and tricky it is to use. There is a significant learning curve associated with DirectX.
OpenGL
OpenGL is an alternative strategy to DirectX, with some differences in features and strategy. OpenGL is available on a variety of platforms. It particularly focuses on 3D programming.
Fastgraph
Fastgraph is a general purpose graphics library that works with Windows or DOS. You can use it with or without DirectX. It is written in assembly language, so it is very fast. Fastgraph is good for beginners because it is so easy to use. It is also powerful enough to satisfy the demands of advanced programmers. For example, to display a pcx file and paste it to the screen, your code would look something like this:
```
fg_showpcx("game.pcx",0);
fg_vbpaste(0,639,0,479,0,479);
```
There is no function in DirectX or OpenGL that simply displays a pcx file. Fastgraph gives you lots of functions to do animation in 2D and 3D

8. LINKS

Below is a listing of various resources available for game design on the web.

Game Design Organizations

Computer Game Developers' Association Home Page

International Game Developers Network Home Page

The Game Manufacturers Association (GAMA) Home Page

The Academy of Adventure Gaming Arts & Design

Game Design

Artificial Intelligence in Games

Articles and Book Chapters by Diana Gruber

Game Developer
An on-line version of the magazine.

Gamasutra - The Art & Science of Making Games
A joint work between Game Developer magazine and the CGDC.

IntFicLinearityVsBreadth
A discussion on interactive fiction design.

Game Design 101
A good article from C/Net's Game Center.

Legal and Copyright Issues

Ten Big Myths About Copyright Explained

The Copyright Website

Interesting Games

Hundred Years War
Jim Dunnigan's current online project.
Hundred Years War

Newsgroups

rec.games.design

rec.games.programmer

Other

Even Brook's Wargame Reviews, Analysis and Rumor/News
A wealth of information on published wargames.

MILNET: Open Source Military Information Database

University of Waterloo Museum and Archive of Games

XYZZYnews Home Page
Fan magazine for interactive fiction (Infocom games)

9. Designing a New Game

One of the main goals of doing this project was designing a new computer game. So after researching on different computer games and understanding the design process of a computer game, I started developing a new game.
This game is in the designing phase.

Story

This Game has only one player. It is a skill & Action game.
Player needs to find his/her way through a maze. But the difference with the other games is that the player will be told which way s/he should go.
Game has also have a research purpose to figure out which symbols, with which color gets the player attention better.
The player needs to pay attention to the symbols on the screen, and the more he gets them the better is his or her performance. *So it can improve reading skill as well.*

Graphic

This Game has both 2D and 3D objects. The road and the car are 3D objects. The symbols are either 2D or 3D objects. It is tried to have good graphic in this game. Different graphic software, like flash, Photoshop, and freehand will be used to create the objects.

Interface

The main input device in this game is the keyboard. Function keys will be used at the easiest way.
The interfaces are tried to be as user friendly as possible.
Different colors and different symbols are used in this game.

Software Requirements

For some graphic and also some part of the game story I will use Flash software.
For developing a game there are many different tools. I would like to use java and OpenGL in this project, since they have powerful graphic options.

10. Future Work

The next step is implementing this game. Before starting implementation, as we understood from the information in this research, I need to test the design phase to make sure that it is the game that I had planned originally to develop. It has been seen that sometimes in this step some games have been stopped from going to the other phases. It means sometimes the design test process fails and it is more advisable in this situation we abort going to the programming and implementation the game. So the next step would be testing the design phase and after that implementing the game.

11. References and Bibliography

1. Crawford, Chris, The Art of Computer Game Design.
2. David Joffe, Game programming with DirectX,1997.
3. Edited by: Francois Dominic Laramee, Game Design Perspective, Charles River Media, INC., 2002.
4. Rouse, Richard, game Design Theory & Practice, Wordware Publishing,INC., 2001.
5. Walker, Chad and Eric, Game Modeling using low polygon techniques, Charles River Media, INC., 2001.
6. Mark Lewis Baldwin and Bob Rakosky, Project AI