



Game Institute Student Challenge 9

Team Charleston GC9 Game Specification

<http://www.gameinstitute.com>

Updated: October 31, 2008

Contents

Game Challenge 9 Objective	4
Game Challenge 9 Judging and Scoring	4
Game Challenge 9 Start and End Date	4
Team Charleston	4
Game Concept and Storyline	4
Game Title and Genre.....	5
Key Objective for Winning the Game Challenge.....	5
Game World.....	5
Game Sequence	6
Development Tools and Libraries.....	7
Asset Management	7
Assets (Graphic File)	8
Assets (Sprite)	8
Assets (3D Meshes)	9
Asset (Sound File).....	9
Game Level	9
Game States.....	12
Player Metrics	12
Merit Badges.....	13
Little Miss – Merit Badges	14
Merit Badge Cut Scene	16
User Interface - Game Play.....	16
Save Game	17
Game Design Process	18
Configuration of the Program.....	18

Layers..... 20

Class Diagrams 21

Appendix A: Checklist Coding 23

Appendix B: Checklist 2D Art..... 24

Game Challenge 9 Objective

The theme for this game challenge is "Special Powers". Entrants are completely free to pick a style and genre for their project (i.e. shooter, puzzle, action, RPG, racing sim etc.) but a particular emphasis must be placed on including special powers that augment the player's "normal" abilities.

Some examples might include; weapon or smart-bomb powerups in a side-scrolling shooter, Jedi force-like powers in a first person shooter, offensive or defensive spells learned / collected in an RPG, turbo-boost or weapon abilities in a racing sim. The more fantastical the power / ability the better!

Wherever possible, these special powers / abilities should be gained or collected during game-play rather than having the player start with such abilities.

Game Challenge 9 Judging and Scoring

Judging will be conducted by Jon Hoult and Drew Sikora (aka. Gaiiden). Scoring will be divided into four sections: programming skills (40%), game design (40%), resources (10%), and judge's bonus (10%). The two judges' scores will be averaged together.

Game Challenge 9 Start and End Date

Start Date: October 18th, 2008, 12:00 am EDT

End Date: December 12th, 2008, 11:59 pm EDT

Team Charleston

Team Charleston is the name given to the two original misfits who have teamed up to create a GC9 submission. They both live near Charleston, South Carolina. Additional members may join the team. The name will probably remain the same. As of this revision the roles and team members are as follows.

2D Artist – Greg Rexroad

Level Designer – Greg Rexroad

3D Artist - Xenthral

Lead Programmer – Chuck Bolin

Game Designer – Chuck Bolin

Game Concept and Storyline

Cindy is a 10 year-old Rainbow Girl (similar to the Girl Scouts). She is the player's character. Cindy has been working very hard to earn merit badges. She has only one more merit badge to earn in order to get promoted to "Little Miss". She must sell 100 boxes of cookies to earn the "Cookie" merit badge. As the story begins, Cindy has sold 99 boxes of cookies.

Cindy is selling cookies in front of a store located near the Enchanted Forest. Suddenly, an Ogre comes by the table with the last box of cookies and steals it. He runs into the Enchanted Forest.

Surprised and angry, Cindy grabs a nearby object (several choices) and begins pursuit of the Ogre into the Enchanted Forest. Her objective is to catch up with the Ogre and retrieve the box of cookies. This may be done by catching the Ogre. The Ogre's will to resist will decrease as Cindy gets near him.

Along the way Cindy encounters all sorts of troublemakers. These include witches, fairies, and elves. Cindy must deal with these creatures with force. Hidden throughout the forest are items unfamiliar to a typical magical and enchanted forest. These items consist of weapons of varying degree of power and usage. Some weapons are baseball bats, pistols, shotguns, machine guns, bazookas, rocket launchers, and flamethrowers.

As Cindy masters the various weapons, she earns corresponding weapon-based merit badges. These merit badges are not typical for Rainbow Scouts but they do increase her skill level and overall points.

As a Rainbow Scout, Cindy is bound by her vows of service to those who are in need. Failing to serve others decreases her overall strength. Cindy **MUST** serve to maintain her strength. Throughout the game, an 'old woman' appears. It is in the best interest of Cindy to help frequently.

The Enchanted Forest is full of dangers and hidden things. It is nearly impossible for Cindy to catch the Ogre without help. Help is provided within the forest by a cute and cuddly creature. The purpose of the creature is to advise Cindy. The creature will hide or run away in times of danger. At times he is critical of Cindy. However he does keep her informed of threats, power-ups, and the location of interesting items.

Game Title and Genre

The game is titled "*Cindy and the Cookie Stealing Ogre*". It is a top-down "shoot 'em" game. The player (Cindy) is free to blast anything that gets in the way as she pursues the Ogre.

Key Objective for Winning the Game Challenge

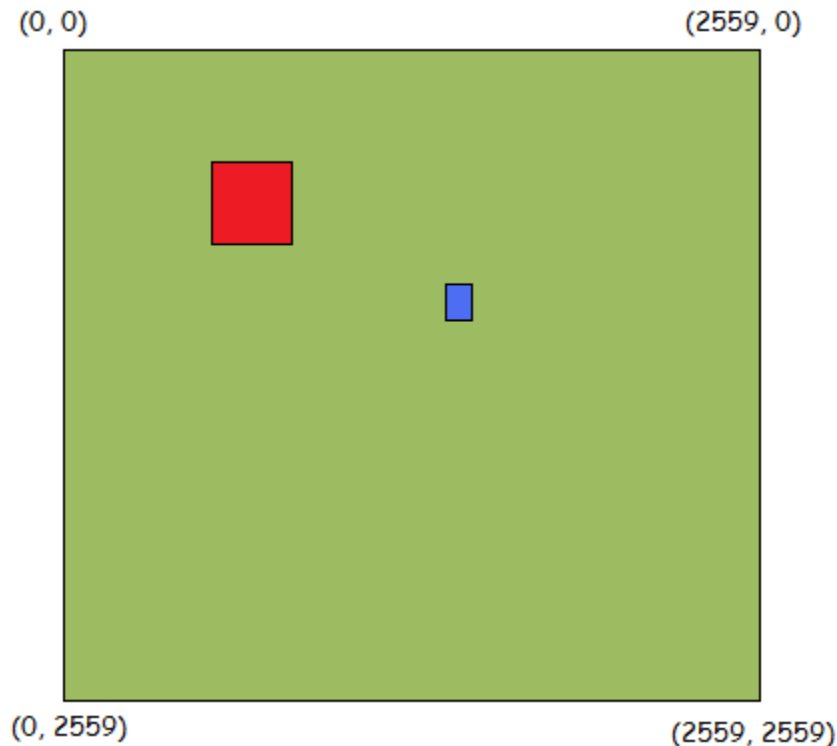
The Game Challenge has 17 participants (individuals and teams) as of this writing. Several participants will complete the challenge and submit functional and impressive games. Therefore, the single most important requirement of the Team Charleston submission is that it is **ADDICTING, FUN, and OUTRAGEOUS**. Team Charleston's game will not win if this requirement is not met.

Game World

The game world is the Enchanted Forest. It is a flat terrain consisting of trees, walls, buildings, and streams. These items are arranged in such a way as to create a very large maze. The player may explore the world by navigating around the obstacles or simply blast them out of the way.

The game world is constructed of sprites in various sizes. Areas within the game world without items are colored by the setting of the background color. The game world dimensions are 25.0 x 25.0. For reference, a 512x512 pixel map measures 5.0 x 5.0 units. Characters are typically contained in a 32x32 pixel sprite. Sprites may be of any size. The game world coordinates are relevant for placement of 3D meshes. Sprites use pixel coordinates.

The pixel size of the game world is 2560 x 2560. 2560 is calculated by multiplying 512 x 5. The illustration below indicates the 2D coordinates of the game world.



Game World Coordinates

Game Sequence

The game consists of a series of cut scenes, game play (blasting and exploring), helping others, receiving merit badges, and winning (or losing). In order to progress through the game Cindy must earn merit badges.

Step and Type	Description
1: Cut scene	Top-down view of Cindy in front of a store. Several people are walking by her table: police officer (9 MM handgun), young baseball player (with ball bat), and a Karate student (with Samurai sword). An Ogre runs from off screen to the table and snatches the Rainbow Girl cookies and runs toward

	the Enchanted Forest.
2: Player Control (Level 1)	Player battles elves in the Enchanted Forest. Elves 'kick' the player. Weapons available for use are the baseball bat, and pistol in addition to punching.
3: Cut scene	Transition. Merit badge for baseball, pistol, or punching.
5: Player Control (Level 2)	Player battles fairies in addition to the elves. Fairies fly around the players and 'whacks' Cindy on the head. Weapons available are shotguns, bow and arrows, and swords.
6: Cut scene	Transition. Merit badge for archery, shotgun, or fencing.
7: Player Control (Level 3)	Player battles witches in addition to elves and fairies. Witches hurl fireballs. Weapons consist of machine guns, flame thrower, and rocket launcher.
8: Cut scene	Transition. Merit badge for machine gun, flame thrower, or rocket launching.
9: Player Control (Level 4)	Player battles large numbers of fairies, witches, and elves. Additional weapons consist of a tank. A helicopter is used to travel across body of water.
10. Cut scene	Transition. Merit badge for operating a tank or helicopter. Victory cut scene.

Development Tools and Libraries

Language: C++ (game framework) and VB (game editor)

IDE: Visual Studio 2003

Sound: FMOD

Graphics: DirectX

Artwork: Paint.net, Paint Brush, TileStudio

Asset Management

Assets consist of graphic files, 3D meshes, sound files, and sprites. All four types of assets are defined in a data file named "assets.dat". The program loads the asset file once at startup. New assets may be added to the project and defined within the "assets.dat" file. The data file is a comma delimited file that supports comments preceded by two forward slashes.

Each asset is indicated by a type and unique id number. This id numbers are used in the level files.

Assets (Graphic File)

The format for defining a graphic file contains four parameters.

graphicfile, file ID, filename, description

Examples are as follows.

graphicfile, 0, menu.png, Main menu file

graphicfile, 6, trees.png, Trees used for game world.

Assets (Sprite)

Sprites are rectangular shapes located on a graphic file. A graphic file must be defined in the “assets.dat” file. The format for a sprite is as follows.

sprite, sprite ID, file ID, left, top, width, height, layer, radius, scale, alpha, angle, max frames, time betw, description

Examples are as follows.

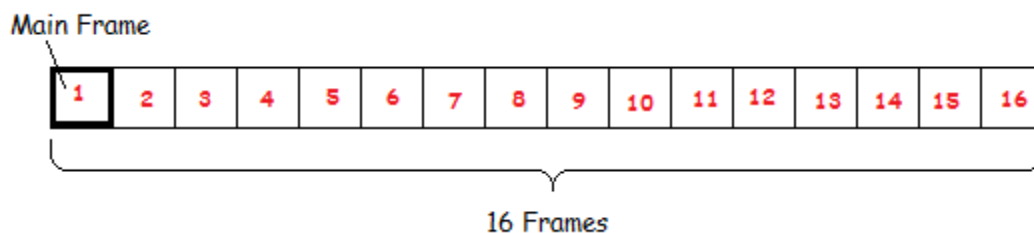
sprite, 6, 0, 1, 235, 288, 64, 7, 0, 1, 255, 0, 1, 0, Play Game

The above line specifies sprite number 6 is defined on graphic file number 0. The left and top position of the sprite located on the graphic file is (1, 235). The corresponding width and height is 288 and 64. The scale is 1. Alpha is set to 255 (no transparency). Although the sprite is drawn at an angle of 0 radians, it may be drawn in the game at any angle located between 0.0 and 6.28 radians. In this case it is displayed at 0 radians. The maximum number of frames is 1. This means it is a static (not animated) sprite. The time between frames is set to 0 because it is not used. This sprite is the “Play Game” sprite seen in the menu.

sprite, 24, 10, 0, 0, 32, 32, 6, 0, 1, 255, 0, 16, 0.1, Explosion

The above example is a 16 frame animated sprite that simulates an explosion. Sprite number 24 refers to 16 frames or animated sprites. It is only necessary to define the “Main Frame” indicated below.

NOTE: An important rule is that the other sprites must follow the main frame and be the same width and height size. The sprite will be animated by the game framework. The time between frames is 0.1 seconds. Therefore the animation sequence is 1.6 seconds.



Animated Sprite Format

Transparency or alpha is set in the graphics file. PNG files are used for the game for all graphics. The alpha channel requires the artist to select a color to be used to indicate transparency. The alpha setting in the sprite definition determines how much transparency to apply. A value of 0 makes the sprite invisible.

Assets (3D Meshes)

The game framework supports “.X” files and textured surfaces. The format for defining a mesh is as follows.

mesh, mesh ID, filename, scale X, scale Y, scale Z, description.

An example is defined as follows.

mesh, 15, box.x, 0.2, 0.2, 0.2, Standard box used in game.

Mesh number 15 is loaded from “box.x” file. It is scaled in all three axes from 1.0 to 0.2.

Asset (Sound File)

The format for a sound file is as follows.

soundfile, sound ID, filename, description

An example of a sound file is as follows.

soundfile, 78, speedkills.mp3, Background racing music.

Game Level

Each level within the game is defined by a comma delimited data file. The convention for naming level files is “level1.dat”, “level2.dat”, etc. Comments are added using two forward slashes. The level file is loaded when required by game play. At startup “level1.dat” file is loaded.

Each line consists of a keyword for the specific game object. Each object type may have a different number of required parameters. The game object line is ignored if the number of parameters does not match the requirement.

Game Object Keyword	Description of Keyword
Mesh	Places a 3D mesh within the game world. A sprite covers a mesh.
Sprite	Places a 2D sprite in the world. Usually this is used for interface graphics. More specific game objects are defined. Sprites used for game objects are

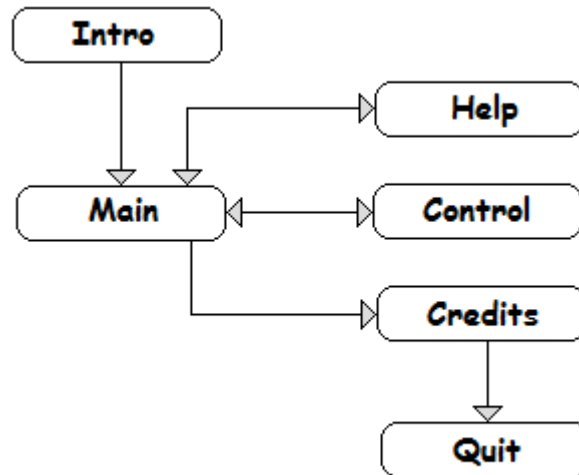
	not affected by destruction or collision. They remain neutral.
Player	Defines starting position for the player's character 'Cindy'.
Witch	Defines starting position, sprites, and behavior of a witch. The witch flies about casting fireballs. Fireballs are very threatening to Cindy and must be avoided at all costs. The witch comes and goes.
Elf	Defines starting position, sprites, and behavior of an elf. An elf is very mean to Cindy. It will run up to her and try to kick her. This is problematic when there are numerous elves attacking her. The elf continually harasses Cindy.
Fairy	Defines starting position, sprites, and behavior of a fairy. Fairies flutter about slapping Cindy in the head. Head slaps cause more damage to Cindy's health than kicking by elves.
Ogre	There is only one Ogre in a game. Ogres are large and intimidating. However, they lack courage, especially against weapon packing Cindy. The Ogre is only displayed during game play in the last level in lieu of 'destination'.
Shotgun	Weapon that includes 10 rounds.
Machine Gun	Weapon that includes 50 rounds.
Grenade	Single explosive device.
AmmoSG	x rounds of shot gun ammo.
AmmoMG	x rounds of machine gun ammo.
Rocketlauncher	Shoulder launched rocket launcher with 3 rockets.
AmmoRocket	Single rocket.
AmmoGun	x rounds of 9MM pistol ammo.
Flamethrower	Weapon that includes 10 blasts of flame.
FuelFlame	Bottle of flame thrower fuel good for 10 blasts of flame.
Tank	Tank that may be controlled by Cindy. Contains 10 rounds and 20 units of fuel.
AmmoTank	x rounds of tank ammo.

FuelTank	x units of tank fuel.
Helicopter	Used for transport and contains no weapons.
Bat	Baseball bat used for clubbing elves and fairies. Breaks after 10 whacks.
Sword	Samurai sword good for numerous swings.
Bow	Bow with 10 arrows.
AmmoArrows	x arrows.
Pistol	9MM pistol including 10 rounds of ammo.
Grass	Various types of grass sprites used for the ground. May be damaged by fire or explosion (craters).
Dirt	Various types of dirt sprites used for the ground. May be marred by explosion (craters).
Wall	Various types of wall sections used for construction. May be destroyed.
Tree	Various types of trees used to decorate landscape. May be destroyed leaving only a charred trunk.
Prize	Various types of prizes that the player may collect to increase health.
Power	Various types of power-ups that may increase Cindy's speed, accuracy for shooting, power in punching elves or fairies.
Grandma	Old woman that appears and travels from point A to point B. Cindy increases 'kindness' by helping the old woman. Helping happens simply be moving close to the old woman. A circle around the old woman indicates she is receiving help from Cindy.
Destination	Player must move within 100 pixels of this coordinate to complete the level. This applies to all levels except for the last level.

All game objects have an X and Y position, sprite ID for normal appearance, and a sprite ID for damaged appearance. Other information is added to the parameters as required.

Game States

The main program is governed by game states. The states are “Intro”, “Main”, “Control”, “Credits”, “Help”, and “Quit”. All game play and cut scenes identified in the 16 steps of the Game Sequence are controlled within the “Control” game state.



Game State Diagram

Player Metrics

Winning the game occurs when Cindy retrieves her last box of cookies from the Ogre. The Ogre will give up the cookies when Cindy is within 100 pixels of the Ogre. He will stop running and will no longer resist. Within the range of 100 the game sequences to a cut scene containing a dialog and transfer of the cookies.

Throughout the game levels, two attributes are displayed and monitored by the program. The “health” attribute (range of 0 to 1000) decreases every second of game play. Kicks by elves, slaps by fairies, and fireballs from witches each have a progressively negative effect on Cindy’s “health”. Health is set to 1000 at the beginning of each level ‘if’ the player earns the a “Merit Badge”. The game is over if health reaches 0.

Description	Amount of Health Damage/Healing
Standing or walking	-1 unit/second
Running or fighting	-3 units/second
Kicked by an Elf	-10 units
Head slapped by a Fairy	-20 units

Hit by a Witch's fireball.	-100 units
Prize (health)	Varies from +5 and higher

Health Table

The second attribute is "kindness". Cindy begins with a score of 1000. This does not automatically reset at each level. The game is over if kindness reaches 0. The only way to increase Cindy's "kindness" is for Cindy to perform an act of kindness to the "old woman". This is achieved by moving Cindy to within 100 pixels of the old woman.

Kindness decreases each time Cindy fights. Fighting is a necessity to maintain her strength. However, "Rainbow Girls" must achieve the correct balance of fighting and acts of kindness.

Description	Amount of Kindness Decrease/Increase
Kicking	-1 unit each kick
Shooting bow and arrow	-2 units each arrow
Shooting pistol	-2 units each round
Shooting shotgun	-3 units each round
Shooting machine gun	-5 units each round
Swinging bat	-1 unit each swing
Swinging sword	-2 units each swing
Grenade	-10 units each explosion
Rocket	-20 units each explosion
Tank	-30 units each explosion
Flame Thrower	-50 units each fire blast
Prize (kindness)	Varies from +5 and higher
Helping Old Woman	+20 units each second of assistance

Kindness Table

Merit Badges

Merit Badges may be earned by Cindy during each level. Earning a merit badge automatically increases the player's health at the beginning of the very next level. Following are a list of badges that may be

earned. Multiple badges may be earned in a level. A merit badge for a weapon increases the accuracy of using that weapon in subsequent levels.

Merit Badge Name	How Merit Badge is Earned
Kick Boxing	Kicking elves 20 times.
Kindness	Providing 60 seconds of assistance to the old woman in one level.
Sword Fighting	Killing 10 elves or fairies.
Archery	Killing 4 elves or fairies.
Baseball	Whacking elves or fairies 20 times.
9mm Pistol	Killing 10 elves, fairies, and/or witches.
Shotgun	Killing 10 elves, fairies, and/or witches.
Machine Gun	Killing 20 elves, fairies, and/or witches.
Grenade Throwing	Killing 10 elves, fairies, and/or witches. Destroying 5 objects (wall sections, trees, buildings).
Rocket Launching	Killing 10 elves, fairies, and/or witches. Destroying 10 objects (wall sections, trees, buildings).
Flame Thrower	Killing 10 elves, fairies, and/or witches. Destroying 20 trees.
Tank Command	Killing 10 elves, fairies, and/or witches. Destroying 20 objects (wall sections, trees, buildings).

Little Miss – Merit Badges

Retrieving the cookies from the Ogre allows the player to win the game. The ultimate accomplishment is to win the game and to get promoted to Rainbow Girl “Little Miss”. This is accomplished by earning 7 out of the 11 merit badges.

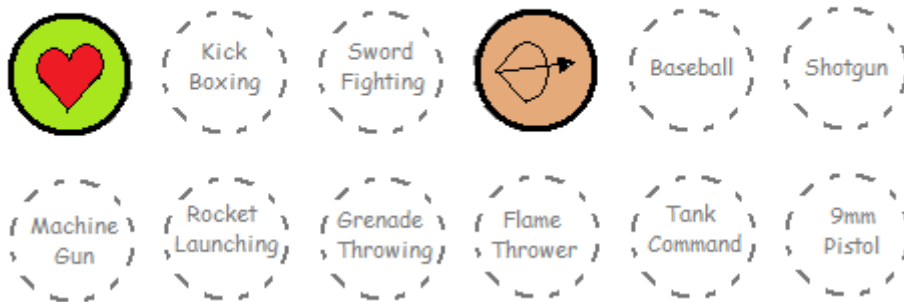
A display similar to the one below is shown between levels. The cut scene presents the Merit Badge.

MERIT BADGES



Merit Badge Display – No Badges

MERIT BADGES



Merit Badge Display – Two Badges (Kindness and Archery)



Sample Little Miss Patch (created at <http://www.logosoftwear.com>)

Merit Badge Cut Scene

When a level is completed, it will continue to the very next level. However, if a Merit Badge has been earned, a cut scene will occur between levels.

The 3D Mesh of the Merit Badge will start small at a distance in the center of the window. It will grow in size. It is coin shaped. During the movement it will be rotating about the X and Z axis. When the badge is large enough to be seen it stops for a moment. The Merit Badge Display illustrated on the previous page is displayed with the additional badge.

User Interface - Game Play

The user controls the game with the mouse and the keyboard. The LEFT mouse button performs the following functions.

LEFT MOUSE BUTTON

Inventory List – Selects weapon.

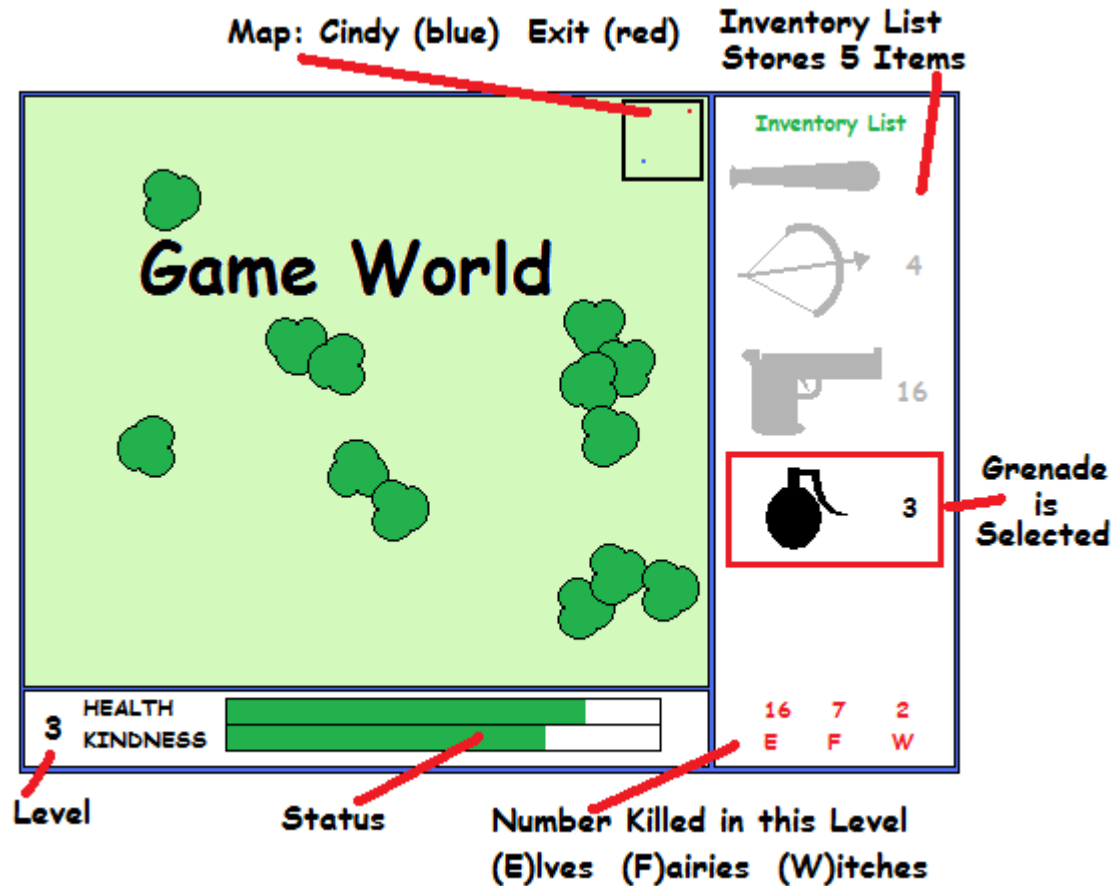
Game World – Fires weapon.

Keys W,A,S,D

Game World – Moves Cindy forward and backward (W and S). Rotates Cindy (A and D).

Key P - Pauses game.

Key S – Saves game.



Game Play Window

Save Game

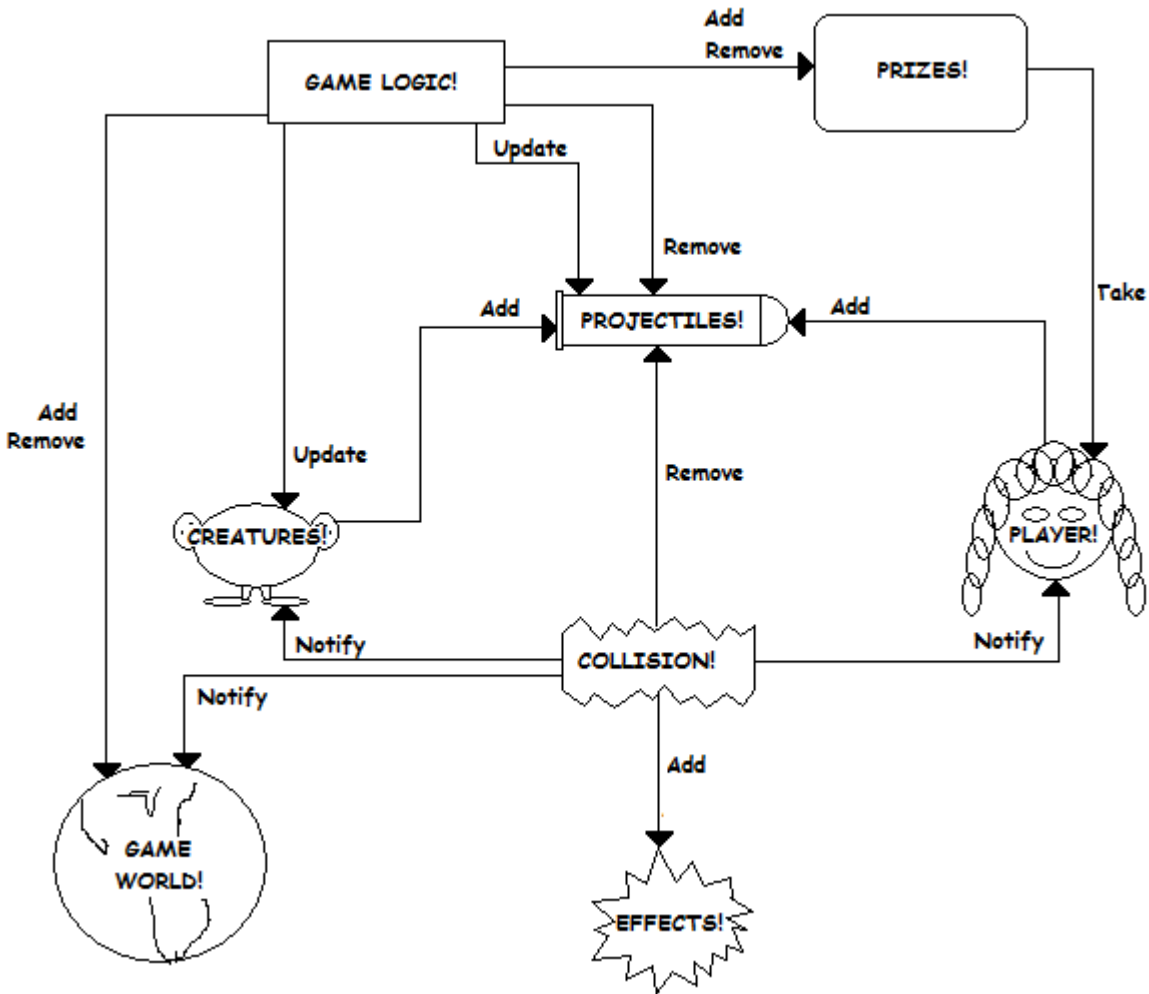
Prior to exiting a game, it may be saved by pressing the "S" key. In fact, this key may be pressed throughout the game play to save the latest progress. The following information is saved.

- Level
- Merit Badges
- Health
- Kindness
- Elves Killed
- Fairies Killed
- Witches Killed
- Weapons in Inventory
- Ammo in Inventory
- Player Position

The comma delimited file format will be determined later. This feature is not required for the game. It will be implemented if there is time remaining in the challenge.

Game Design Process

The game logic manages all game objects. Following is a general process flow.



Configuration of the Program

A configuration file “config.txt” is loaded during startup. The user/developer may configure various features in the program. Ideally, a graphical interface would be created to modify this file. Due to the short time allowed for the game challenge the manual file will suffice. The configuration file allows comments preceded by two forward slashes.

The program window may be positioned anywhere on the computer screen.

```
screen_left = 100    //left side of program window in pixels
screen_top = 200     //top of program window in pixels
screen_width = 1024 //width of program window in pixels
screen_height = 768 //height of program window in pixels
```

The window may be centered in the computer screen by setting the following parameters to -1.

```
screen_left = -1     //center horizontally
screen_top = -1      //center vertically
```

The program may run in windowed mode or full screen. NOTE: Full screen mode crashes if the program loses focus (e.g. pressing ALT + Tab).

```
full_screen = 0      //windowed mode, 1 = full screen
```

The program has information that is displayed on the screen. This is useful for debug. Setting the following parameter to 1 enables the display of information to the screen.

```
display_debug_info = 1 //display info, 0 = show no info
```

The music may be turned on or off.

```
play_music = 1       //play music, 0 = turn music off
```

Information related to the level may be defined as follows.

```
current_level = 1    //sets current level, useful for testing various levels
max_levels = 3       //sets maximum number of levels
```

```
//defines files used for each level (game limit is 10 levels)
level_file1 = level1.dat
level_file2 = level2.dat
level_file3 = level3.dat
```

Layers

An important concept to game objects is the 'layer'. The layer determines the order of rendering. There are 8 layers for game objects.

Layer	Description of Layer
0	Splatter or textures used as background for 3D meshes.
1	3D meshes.
2	Sprites related to the ground.
3	Sprites placed on top of the ground such as wall sections, creatures, projectiles, and the player sprite.
4	Sprites that cover other sprites such as a building, bridge, or trees.
5	Effects such as fire, explosion, and smoke.
6	Prizes and power-ups.
7	Interface graphics.

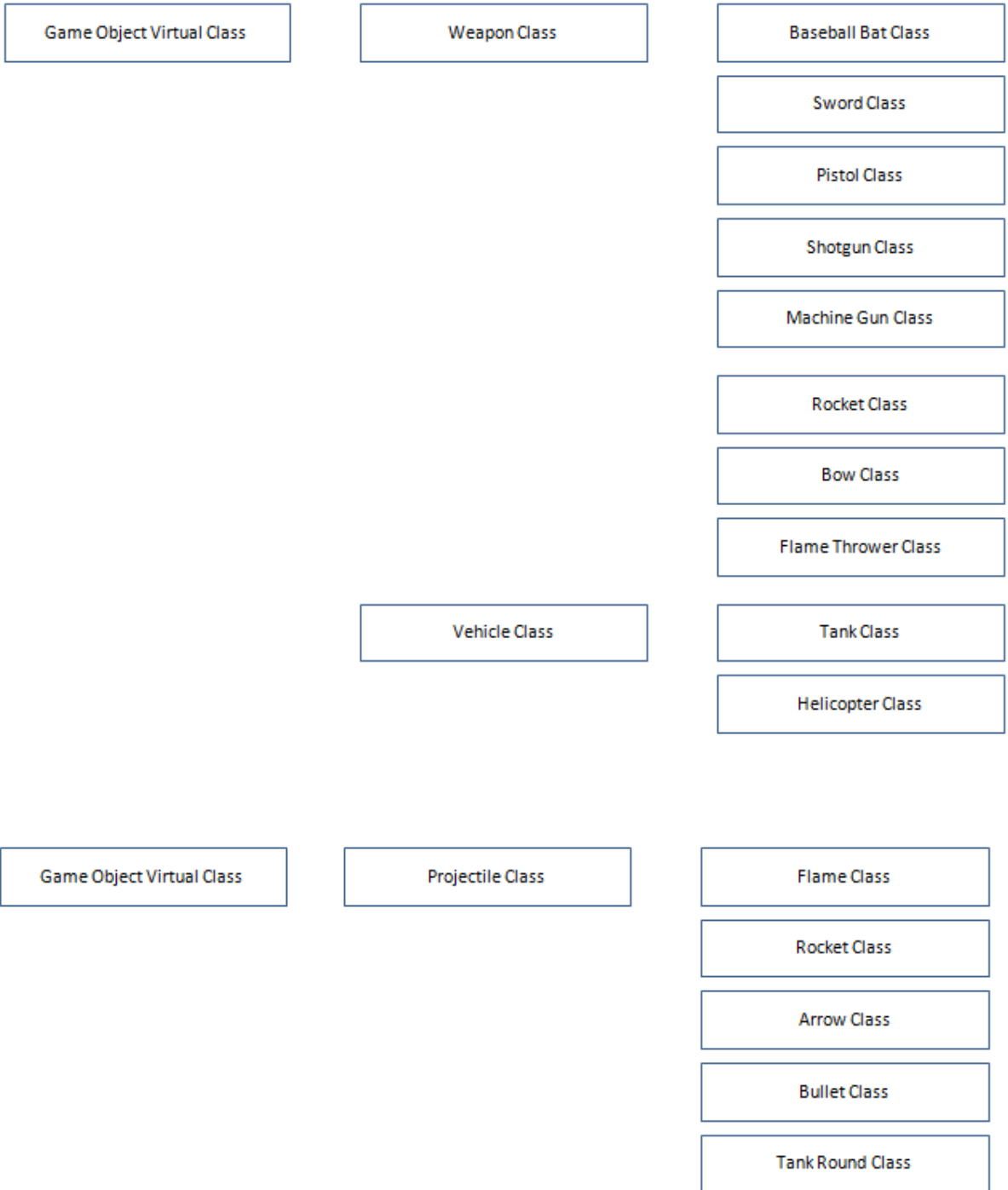
Game Object Layers and Descriptions

It is only necessary to define the layer for the sprites.

Class Diagrams

All game object classes are derived from a game object virtual class. Four major classes are derived from the game object class: landscape, character, weapon, and projectile.





Appendix A: Checklist Coding

- Create character base class.
- Create player class.
- Create elf class.
- Create fairy class.
- Create projectile class.
- Create effects class.
- Create tank class.
- Create helicopter class.
- Create woman class (old woman).
- Create prize class (includes power ups).
- Create collision management class.
- Add level 1 game state.
- Add level 2 game state.
- Add level 3 game state.
- Add level 4 game state.
- Add initial cut scene state.
- Add transition cut scene (level 1 and 2).
- Add transition cut scene (level 2 and 3).
- Add transition cut scene (level 3 and 4).
- Add final cut scene (win the game).
- Create weapons base class.
- Create GUI inventory window.
- Create GUI health and kindness window.
- Create GUI map.
- Create GUI input (mouse and keyboard).
- Create editor state.
- Open and load level file into editor.
- Save editor level data to a file.
- Add and delete game objects in the editor.
- Create scoring process.
- Create power up process.
- Glue kindness to old woman.
- Glue projectiles to weapons and characters.

Appendix B: Checklist 2D Art

- Create landscape sprites (dirt, grass, water).
- Create tree sprites.
- Create animated burning tree sprites.
- Create rocks sprites.
- Create animated rock shrapnel sprites.
- Create wall sprites.
- Create animated exploding wall sprites.
- Create various building sprites.
- Create animated exploding building sprites.
- Create animated witch sprites.
- Create animated dying witch sprites.
- Create animated elf sprites.
- Create animated dying elf sprites.
- Create animated fairy sprites.
- Create animated dying fairy sprites.
- Create animated old woman sprites.
- Create animated Cindy sprites (standing, walking).
- Create Cindy with animated baseball sprites (standing, walking).
- Create Cindy with animated pistol sprites (standing, walking).
- Create Cindy with animated punching sprites (standing, walking).
- Create Cindy with animated shotgun sprites (standing, walking).
- Create Cindy with animated machine gun sprites (standing, walking).
- Create Cindy with animated bow and arrow sprites (standing, walking).
- Create Cindy with animated sword sprites (standing, walking).
- Create Cindy with animated rocket launcher sprites (standing, walking).
- Create Cindy with animated flame thrower sprites (standing, walking).
- Create animated tank sprites.
- Create animated helicopter sprites.
- Create animated ammo sprites (pistol).
- Create animated ammo sprites (shotgun).
- Create animated ammo sprites (machine gun).
- Create animated ammo sprites (rockets).
- Create animated fuel sprites (flame thrower).
- Create animated bat sprites.
- Create animated pistol sprites.
- Create animated shotgun sprites.
- Create animated machine gun sprites.
- Create animated flamethrower sprites.