

# ghost

Jessica Wong  
Advisor: Zoë Wood

Cal Poly San Luis Obispo  
Senior Project  
Winter 2010 – Spring 2011

# Table of Contents

Introduction .....	3
Overview .....	3
Related Work .....	4
Algorithms Overview .....	4
Algorithms Details.....	4
PNG texture support.....	4
HUD.....	5
Animated Textures.....	5
Hierarchically Modeled Objects.....	6
Bookshelf.....	6
Paintings and Windows.....	6
Fishbowls.....	7
Opening, Transition, and Ending Scenes.....	8
Images.....	9
Music and Sound Effects.....	9
Layout.....	9
Maps .....	9
Results.....	10
Conclusion.....	10
Future Work.....	10
References .....	11
Appendix .....	12
Feedback Form Data .....	12

## Introduction

3D games have grown to be the standard format for 1<sup>st</sup>-person shooter and adventure games played on the PC and game consoles. Players now expect excellent 3D graphics in addition to great game play. 1<sup>st</sup>-person shooter games are immensely popular with game series such as the Call of Duty by Activision. Adventure games like Assassin's Creed by Ubisoft are also very popular. These games provide the unique experience of escaping reality and taking part in another. In the 1<sup>st</sup>-person shooter and adventure genres, players expect the worlds to be fantastical but at the same time nearly as realistic as the real world around them. In Call of Duty, players take the roles of soldiers and submerge themselves in a world of battles. In Assassin's Creed, the player is thrown into a world of the past and fantasy, weaving together real historical events and fictitious assassinations. In Ghost, we want to create a fantastical version of the world we see most often, a world set on Cal Poly campus. The end goal of our project is to present our game at open houses and show an alternate reality of Cal Poly, one riddled with ghosts. I decided to create a 3D interactive video game in the 1<sup>st</sup> person shooter and adventure genre.

This senior project was done in CPE 476++, a combination of the original CPE476 (Real-Time 3D Computer Graphics Software) and senior project. Topics covered included optimizations, shading, collision detection, game play. All teams of 3-7 people were required to develop a game in the span of two quarters.

In the 1st quarter, the base of the game was established, while in the 2nd quarter, the team polished and made additions to the game.

### 1st Quarter Ghost Team Members

- Nick Feeney
- Billy McVicker
- Jessica Wong
- Seokjune Hong
- Michael Boyd

### 2<sup>nd</sup> Quarter Ghost Team Members

- Jessica Wong
- Seokjune Hong
- Michael Boyd

## Overview

Ghost is a 3D 1<sup>st</sup>-person shooter and adventure game. The player is trapped in Cal Poly's Kennedy Library and must battle enemy ghosts and gather items for the library's archivist to escape. The items (laptops and books) contain research the archivist needs to create an exit for the player. The goal is to collect 4 items from the first floor and 4 items on the second floor. Once all items are collected, the game is won.

Ghost is set in present time and the layouts of the floors are simplified representations of Cal Poly's Kennedy Library. A dimly lighted environment makes exploration more challenging and provides a more mysterious look and feel. To counter the darkness, the provided flashlight lights up small areas of the environment around the player. Ghost also has a dark cartoony look. The game also features real photos from Cal Poly in the 1930s. These photos are provided by Cal Poly's Liberal Arts and Engineering and Kennedy Library's Archives.

Enemy ghost NPCs will attack you to lower your health and prevent you from collecting items. Also serving as a weapon, the flashlight can shoot "light bursts" which destroy the ghosts. The player has a limited number of shots restricted by the amount of battery power the flashlight has. However, the battery power will self-replenish if given enough time. This makes the game more challenging and encourages the player to shoot only when necessary. The game currently consists of 2 floors and grows more challenging as the player makes their way up. A few second floor ghosts can steal items already collected from the player.

Ghost is programmed in C++, OpenGL, and glut. The models were made in Blender and Milkshape and saved as .md2. Sound was added with irrKlang, and sound effects and music were all in the OGG format. We used 'png.h' to import the PNG image files used in textures. Ghost was developed on Linux machines.

## Related Work

Below is a list of games that we referenced when designing our game. The main genres we referenced are horror, adventure, and 1<sup>st</sup>-person shooter. Since our game features ghosts, horror games were necessary references. Other genres of games were also included because they had certain elements we wanted to incorporate into Ghost.

- Silent Hill by Konami (feel, genre: horror)
- Resident Evil by Capcom (lighting and feel, genre: horror)
- Call of Duty by Activision (damage effects on HUD, genre: 1<sup>st</sup>-person shooter)
- Alien Wake by Remedy Entertainment (flashlight, genre: psychological thriller)
- Ghost Busters by Terminal Reality (ghosts and overall game, genre: action)
- Amnesia: The Dark Descent by Frictional Games (look and feel, genre: survival horror)
- Assassin's Creed by Ubisoft (game play, genre: action adventure)

## Algorithms Overview

The following is a list of the technologies implemented in Ghost along with the team members who worked on them. Descriptions of technologies I worked on are described in the next section. The rest of the descriptions can be found in other members' papers.

- 2D placement map for collision detection optimization (Billy)
- Collision detection (Billy, Jessica, Seokjune, Michael, Nick)
- View frustum culling (Billy)
- Layout (Jessica, Nick, Michael)
- Modeling (Billy, Michael)
- Hierarchical Modeling (Seokjune, Jessica)
- Projected texture mapping for flashlight (Nick)
- Heads-up display, HUD (Nick, Jessica)
- Particle systems (Seokjune)
- Animated textures (Jessica)
- PNG texture support (Jessica, Billy)
- Sound (Michael, Jessica)
- Mist (Michael)
- Flashlight projectile, "light burst" (Nick, Seokjune)
- Ghost AI (Billy, Seokjune)
- Collectible Items (Michael, Billy, Jessica)
- Maps (Jessica)

## Algorithms Details

The following sections describe how parts of the game were designed and implemented.

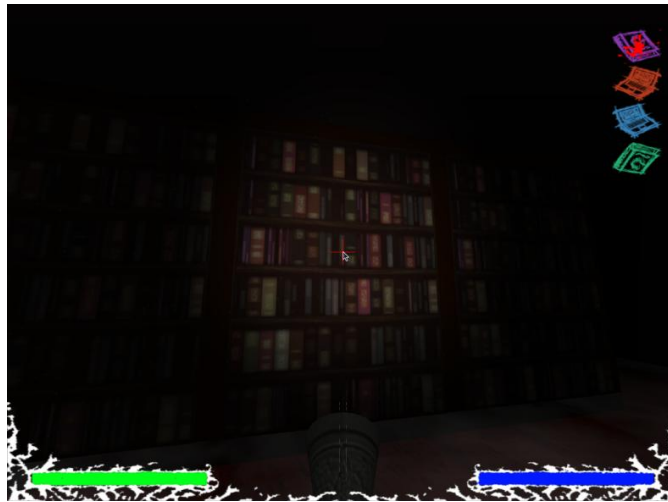
### PNG texture support

Previously, the game used 24 bit BMP images, but they did not support the alpha values needed for some of the transparent textures. The PNG image format was chosen because it supported alpha values and provided good

image quality with its lossless data compression. In order to load PNGs, the 'png.h' library was used as well as open source code borrowed from PNG Texture Loader by David Henry.

## HUD

All images used in the HUD were hand-drawn and created in Photoshop CS4 with a pen tablet. The images were then saved as transparent PNGs. The HUD is the last element to be drawn. With the exception of the flashlight (bottom center), the item icons and the battery and health bar decors were drawn with the lighting disabled to achieve the effect of a 2D layer placed on top a 3D environment.



The bottom left bar represents the player's health, and the bottom right bar represents the battery power for the player's flashlight. The health and battery bars are 2D rectangles. When health decreases, the x-coordinates on the left of the bar decrease. When battery decreases, the x-coordinates on the right of the bar increase. In play, the bars decrement towards their respective corners. The battery bar recharges in timed increments.

Each of the item icons (top right corner) represent items the player needs to collect. The images and color correspond to the item type (laptop or book) and color (purple, orange, blue, or green). The images for each item type are the same. To make the HUD seem less repetitive the coordinates of how their textures are placed were arranged differently in order to rotate and flip them. When the player has collected an item a red splatter will appear on top of the matching item icon. The red splatters are also share the same image, and like the items, their textures are rotated to avoid unappealing repetition.

The HUD was designed to be minimalistic and stand out. Thus, bright colors were used to color the item icons and the bar decors. The design of the bar decors and item icons are scratchy and slightly messy. The goal was for the images to give off a slightly unsettling feeling with their unclean lines while at the same time be clear enough so that players can make out what they are.

## Animated Textures

Because translating objects across would be too expensive for such small animations, animated textures were implemented. The images would be loaded in and given a unique number (key). Each image was a frame in the animation. The frames were added to a list in the order the frames should appear. This list and the desired delay time between frames were set as parameters when Paintings and Windows (described more in depth later) were constructed. The frames were then looped through with the short delay, usually from 0.1 seconds to 20 seconds. This works like traditional animation where each frame has slight changes and creates the illusion of motion. The animations were drawn and manipulated for each frame on the archive photos with Photoshop CS4 and a pen tablet. Making the animations was very time consuming since each frame had to be created from scratch.

Sample of frames used in animated textures



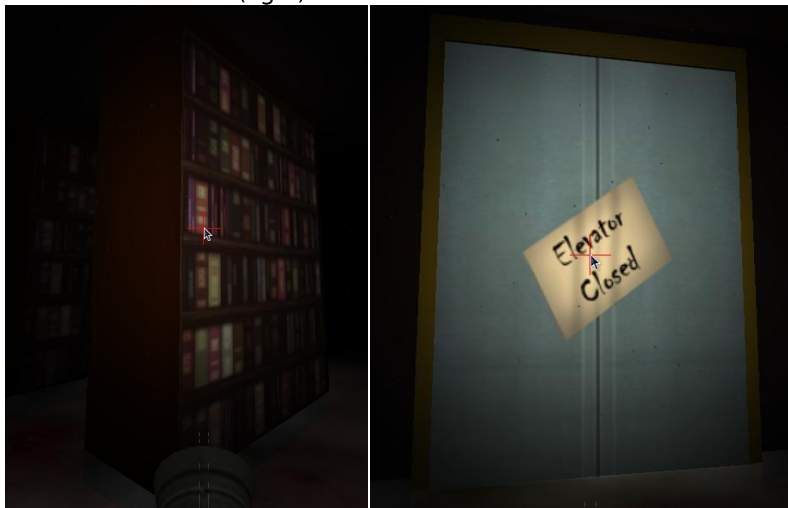
## Hierarchically Modeled Objects

The following sections describe how some objects in the game were made by combining and manipulating OpenGL and glut primitive shapes.

### *Bookshelf*

Since the setting of Ghost is in a library, bookshelves are necessary decorations. Bookshelves are made from one cube that has been scaled into a rectangular box. The rectangular box is textured with a wood texture. Then, a two textured planes sandwich the box to make a bookshelf. The bookshelves have two orientations, rotated 45 degrees on the y-axis. Different coordinates for the bounding boxes (used in collision detection) are assigned for the two positions so that they correspond correctly. Different textures can be passed in for the plane. As shown below a bookshelf can easily become an elevator with different textures.

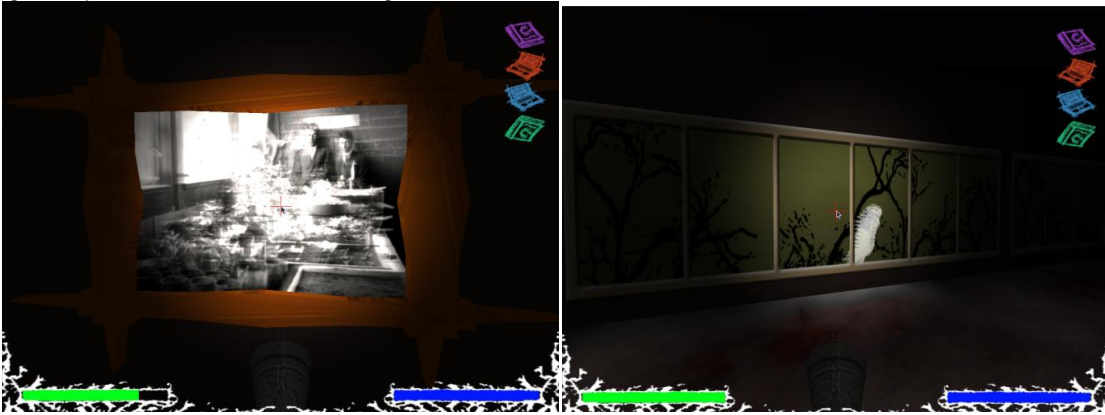
Bookshelf (left) and bookshelf as elevator (right)



### *Paintings and Windows*

The Paintings and Windows are used as decorations in the game. Both use animated textures as described earlier. They draw attention to themselves so that the player gets distracted from the ghosts. Story-wise these the photos displayed in the painting frames are animated because The Archivist has trapped dangerous ghosts in them. When the player makes it to the second floor, these ghosts have escaped and follow the player to the second floor. The windows are placed on the second floor to decorate the wall along the bookshelves. Like the bookshelves, the paintings and windows can be placed in two positions, 45 degree difference and rotated on the y-axis. As such, they also have different bounding boxes for each position.

Painting with photo (left) and window (right)

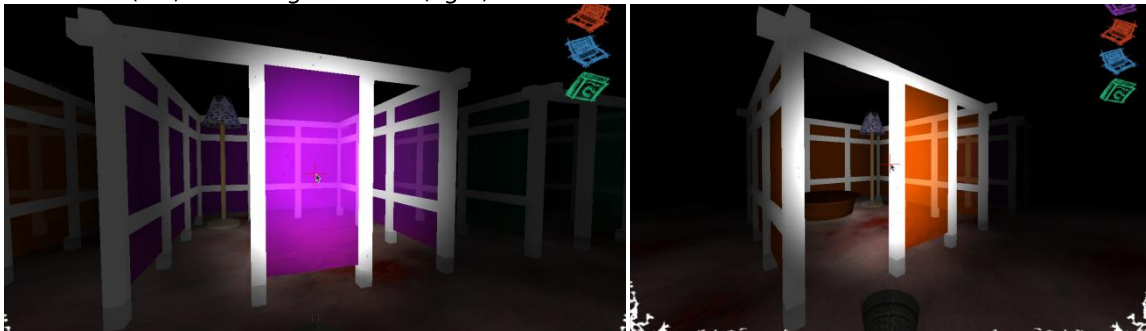


The frames for the paintings are made from cubes that have been translated, rotated, and scaled to achieve their look. The cubes are textured with a wood texture. The photos (animated textures) are textured onto a plane that is positioned in the frame. The windows are planes with animated window textures.

### *Fishbowls*

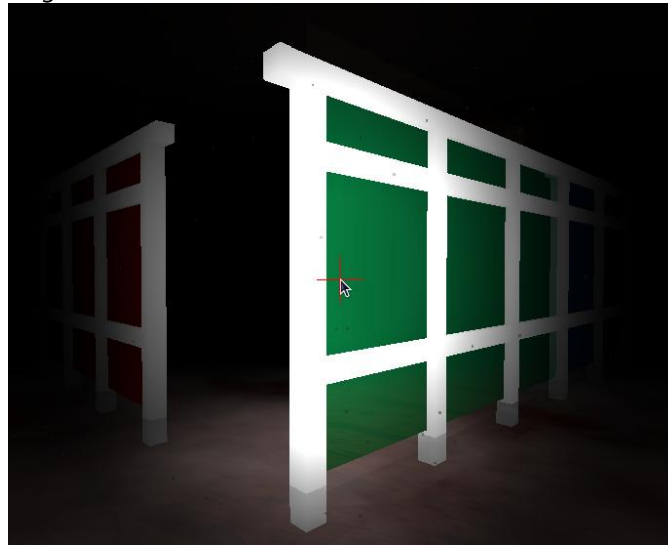
Transparent fishbowls are a big part of Kennedy Library's second floor, so they have been represented in the game as well. In order to have collision detection work correctly, each wall of the fishbowl is separate. The player needs to walk in and out of the fishbowl so one large bounding box that wraps around the whole fishbowl would prevent the player from entering. Adjusting the bounding box code for this one thing would also be too much work. As a solution, the fishbowl is split into separate elements with a bounding box for each. The fishbowl class offers three types of walls, ones with outside frames, ones without, and doorways. The ones with frames and without make it so that when the fishbowls are drawn, they won't have overlapping and unnecessary frames. The fishbowls are structured much like paintings. The frames are made from scaled and translated cubes. These cubes have a white material set. A plane is positioned inside the frame and textured with transparent colors. The fishbowls are colored differently so the player can orient themselves and also provide some more visual stimulants. As seen below, the player can see through the transparent fishbowl walls.

Purple fishbowl (left) and orange fishbowl (right)



The fishbowl walls are also used to create walls within the second floor. Their colors help players orient and locate themselves.

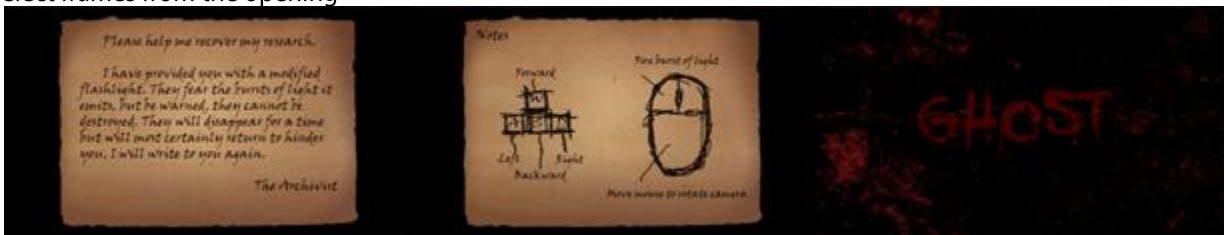
Room entrance with red and green fishbowl walls



### Opening, Transition, and Ending Scenes

The opening scene comes at the very beginning of the game. The transition plays when the player has collected all items on the first floor and is sent to the second floor. This scene is when the player is in the elevator that brings them to the second floor. The ending scene shows when the player has collected all items on the second floor and the game is completed. For a game over, only the last two frames of the ending are shown. These scenes set the story up as well as provide some information on how the game is played.

Select frames from the opening



Select frames from the transition (elevator)



Select frames from the ending





### *Images*

The opening, transition, and ending scenes also used animated textures. Like the Windows and Paintings, the frame keys were put in a list in the order they should be shown. The one difference is that the frames now have different delay times because there is text for the player to read. Each frame was checked for their delay times in a set of conditionals that determined when frames would change.

### *Music and Sound Effects*

Music and sound effects were set in the conditionals mentioned above. The sounds needed to be timed to the frames.

The timing for the frames and sound effects required a lot of adjustments. Trial and error was the main process used to find the most appealing combination of images and sounds. The music was found through internet searches for royalty free music, and sound effects were found at freesound.com.

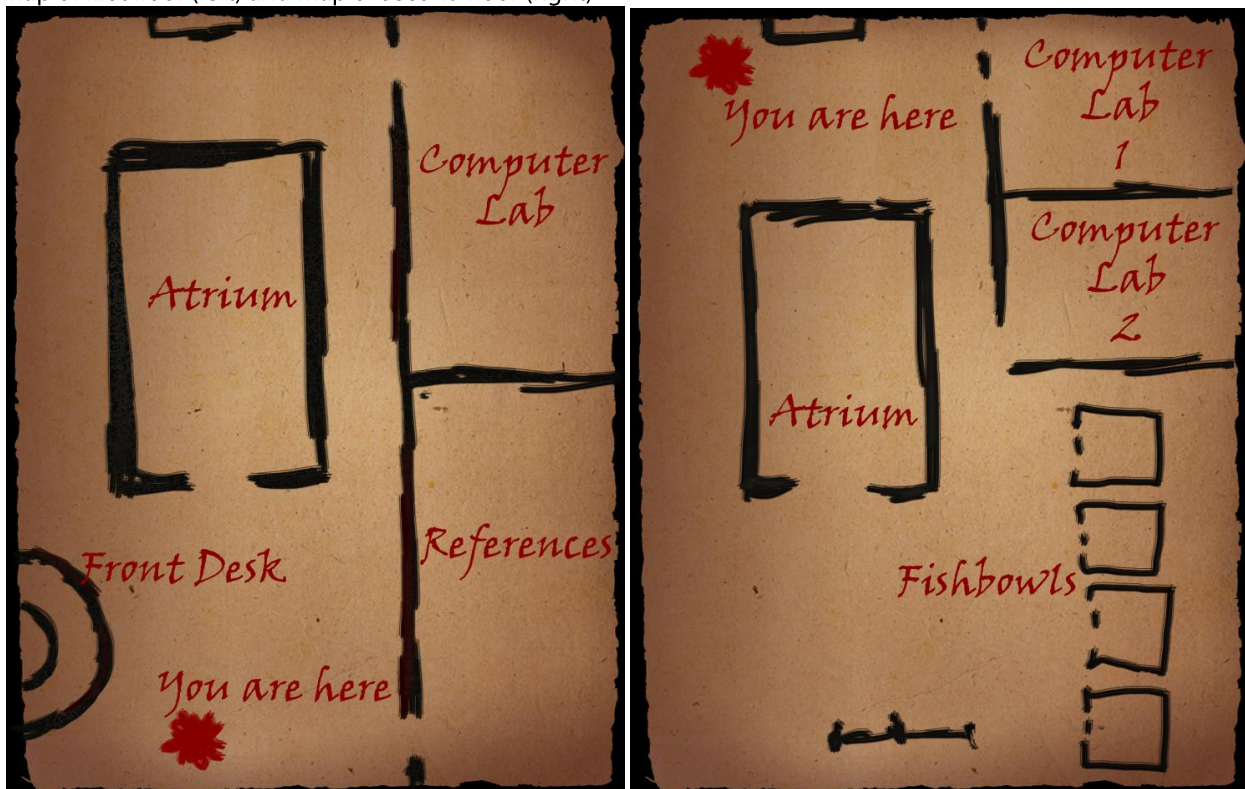
### **Layout**

The first floor layout and second floor layout were designed to be simplified representations of Kennedy Library. The designs were drawn from the Kennedy Library's blueprints. Representing all of the library's features and rooms would make the game too difficult, so key elements were chosen to be included in the game. The first and second floors were populated by placing objects at specific coordinates.

### **Maps**

Two maps are placed on the first floor, and two more are placed on the second floor. The maps are positioned at the front of the second floor and back of the second floor (the "You are here" locations are where the corresponding maps are positioned). From user feedback, one of the most desired features were maps. In order to keep the players wandering but not too lost, maps were provided.

Map of first floor (left) and map of second floor (right)



## Results

At the end of 2 quarters, we were able to create large polished prototype of our game. We're very proud of the effect it has on people. We wanted our game to be fun but also frightening. After working on the game for so long, we were no longer scared by anything in the game. It was great to see that players were shocked when they saw a ghost and that they felt the environment was spooky. We're also very happy with how well the sounds and music matched our game. It really made the game feel complete.

As we developed more ideas for the game, some code did have to be replaced by new code that could handle the new complexity of the technology we introduce to the game. At other times new code had to be added to existing objects so that a technology could work properly. Projected texture mapping for example required that all the object classes with textures also provide a function that would draw the object without textures enabled. If this was not done the spotlight would not properly display on the objects.

From feedback we found that our game can be very challenging when there are a large number of ghosts. We have been continually adjusting the number of ghosts and how they respawn in order to balance the game play. Additionally, from multiple requests for a map, we have placed maps on both floors so that players can find their way more easily. Adjustments also were made on the rate of recharge for the battery bar and how much health a ghost can take from a player based on feedback.

For the first quarter, Ghost's game development process started out with a playable game at the first milestone. After, we focused on adding more technologies and objects to make the world richer. In the second quarter, we added a second floor with new objects. We put more focus on game play and added more sophisticated ghosts (item stealing ghosts). We also limited the number of ghosts on a floor and had ghosts spawn when an item is collected on the second floor.

The work was distributed based on the expertise of the member and also time. Nick and Billy were strong coders, so they did some more complicated parts of the game such as the projected texture mapping, setting up the collision detection, and setting up the layout. Seok was familiar with particle systems, so he created the dust and the explosion effect of when a ghost is destroyed. Michael learned how to model and modeled the collectible items as well as some objects placed in the world. I was familiar with image manipulation, so I made the animated textures and created most of the textures used in the game.

## Conclusion

This project allowed me to bring in other skills I had that I never used in a software project before (e.g., image manipulation). The freedom to choose what kind of game to develop and how to design it was the best part. We had a lot of choices on what technologies to add to our game. We ended up choosing what was feasible, what we could accomplish in the time allotted. However, I think those technologies were sufficient. We were warned that adventure games would be difficult to complete in the time we had, but we had a full running game at the end. We knew of the restrictions and designed accordingly. We had large expectations but kept them realistic. I think that being ambitious but not overly so was what made our game successful.

## Future Work

Future work would include more models for a richer world as well as three more floors to complete the representation of Kennedy Library. There is also the possibility of porting the game and integrating it with the Ghost iOS app (a tour guide app for Cal Poly campus).

## References

"Alan Wake." *Alan Wake*. N.p., n.d. Web. 5 June 2011. <<http://www.alanwake.com/>>.

"Amnesia: The Dark Descent." *Amnesia: The Dark Descent*. N.p., n.d. Web. 5 June 2011. <<http://www.amnesiagame.com/>>.

"Assassins Creed Revelations | Age Gate | Ubisoft." *Assassins Creed Revelations | Age Gate | Ubisoft*. N.p., n.d. Web. 5 June 2011. <[http://assassinscreed.ubi.com/revelations/en-US/agegate.aspx?NextUrl=http%3a%2f%2fassassinscreed.ubi.com%2frevelations%2fen-US%2fhome%2f%3futm\\_source%3dUbi.com%26utm\\_medium%3dCarousel%2bspotlight%26utm\\_campaign%3dAnimus](http://assassinscreed.ubi.com/revelations/en-US/agegate.aspx?NextUrl=http%3a%2f%2fassassinscreed.ubi.com%2frevelations%2fen-US%2fhome%2f%3futm_source%3dUbi.com%26utm_medium%3dCarousel%2bspotlight%26utm_campaign%3dAnimus)>.

"Call of Duty®." *Call of Duty®*. N.p., n.d. Web. 5 June 2011. <<http://www.callofduty.com>>.

"Ghostbusters: Sanctum of Slime." *Ghostbusters: Sanctum of Slime*. N.p., n.d. Web. 5 June 2011. <<http://www.ghostbustersgame.com/>>.

"Konami Digital Entertainment, Inc. : Silent Hill: Shattered Memories." *Konami Digital Entertainment, Inc. : .* N.p., n.d. Web. 5 June 2011. <<http://www.konami.com/games/shsm/>>.

"Resident Evil Portal." *Resident Evil Portal*. N.p., n.d. Web. 5 June 2011. <<http://www.residentevil.com/>>

Wood, Zoe. "Zoë's Home Page." Computer Science - Cal Poly San Luis Obispo. N.p., n.d. Web. 28 May 2011. <<http://users.csc.calpoly.edu/~>

# Appendix

## Feedback Form Data

### Name

1. Tim Adam
2. Tag
3. Chad
4. Victor
5. Jordan

### Was the game stable (did not crash)?

1. Y
2. Y
3. Y
4. Y
5. Y

### Was the game fun?

1. Y
2. Y
3. Y
4. N/A
5. Y

### Was the game too easy? Too difficult? At the right level of playability?

1. Okay...
2. Seemed about right
3. Right level
4. It's fine
5. Right level

### What suggestions do you have?

1. Ghosts trap you means immediate game-over -> consider secondary emergency attack?
2. Perhaps some sort of hit or help as to where to go. Maybe not directly but somehow.
3.
  - Mini-map! I felt like I was wandering blindly in the dark w/o any direction
  - Maybe slightly more light to orient the player more
  - Balancing the # of ghosts in the player's area vs. how much ammo is available
4. Lower mouse sensitivity

5. Add running option

**What did you like about the game?**

1. It was unique and played smoothly.
2. Random ghosts spawning
3. Very good environment and the music suits the game perfectly
4. N/A
5. Good environment

**Why did you dislike about the game?**

1. Battery life of weapon...
2. It was a bit scary but that's what you were going for... The flashlight bursts aren't very noticeable so I wasn't sure I was scaring them.
3. N/A
4. Ghost AI was dumb, have it take cover from fire and stuff, looks like its just going straight for you.
5. Ghost swarming makes game too hard on second floor because there is not enough space to escape.

**Note:** Testers 1-4 did not test out the second floor, but tester 5 did.