Senior Project Paper

James Kautman

April 22nd, 2011

For my Senior Project I chose to create an 3D Virtual World. I chose this subject since I was interested in creating a 3D world, and to learn the basics of a first person shooter game. I made this Windows based, and I think it can be run on Linux and Mac but I am not sure about that. I chose to create this in C++ using OpenGL and SDL to help with the graphics. Why I chose these tools is because after learning how to use them in my Video Games class I thought that it would be the easiest to help me create the world I am making. Within my world I have creatures/robots (depends on how you look at them) that wander around, when the player get close to the creatures they will then begin to chase the player until the player is either dead or how killed the creature. The player can move forward and backwards as well as from side to side (strafing), and can also shoot a gun. The goal of the player is to reach the end of the level(s) without dying. As the player wonders the levels the player can pick up keys to help them on their search of the end of the levels. I know that this is not really a great story for a game to have, but I was more so focusing on the game play over the storyline.

**Visuals**

The visual of my game are my main focus of the project, getting things like lights and textures to work properly and to make it look decently realistic. At first I thought lighting would be hard, remembering back from Video Games class it felt like I need a lot of things to get the lighting to work, until I found a tutorial online that helped my with OpenGL lighting and the lighting effects can out good. One problem that I have come across is when the player kills a monster near the light source some of the monsters around it go dark. I figured out that I was clearing the screen when I was not suppose to clear the screen and that fixed the bug. A big thing I learnt about lighting is to use “glNormal3f(-1.0f, 0.0f, 0.0f);” command with 3D objects, the reason is because as the creatures turn the light gets all mixed up on which surface of the monster to shine on, and you could have the opposite sides being lit up when only on side is suppose to. After getting that command to function properly the creatures look like they are lit up decently and do not have any weird lighting patterns, (except for the on bug I found). Also I found out another thing about lighting and that is with OpenGL you can only have eight light sources on at one time. So if you want the visuals of more lighting you have to use textures that look like there is a light source even though there is no light coming from that place. On how I used the lighting is that as the player progress though the levels the player will reach these “checkpoints.” The player will not notice that it is a checkpoint, but the game will use it as an indicator to change the light. Since I only use one light source, this helps to achieve different things in the game, like it being creepier or that you are coming to a safe area.

Since textures are also part of the visuals I included them in the main interest for the creation of the game. I tried to give the monsters a texture, but I found out that is was way hard, and for me near impossible to get them to look right. So I went in another approach and created a complex object with a head, a body and legs. To determine the direction the creature is facing I gave the “face” a green color while the rest of the creature is orange. Now for other parts of my world the textures were pretty easy on getting them to work. For the walls I have a brick pattern that is repeated, and for the floor I have a tile pattern repeated throughout. I originally planned to get different types of floors but I am running out of time to finish some of my other items I need to get done.

The last part of the visuals that I had worked on was the HUD. A HUD is a place where the player can see what their health is at, the amount of armor they have, how many lives are remaining, which guns they have, and other objects like that. I first created the health and armor bars, these are just standard colors, they can go up and down depending on how much health or armor the player has. Every monster can cause damage to them and the health will go down, if the health hits 0 the player loses a life and gets a refill on life, so the bar goes back to full. As the player loses lives they can see a “little man” in the middle disappear, as the player hits 0 lives they lose the game, and can restart or exit the game. The one thing I did not finish in this game was the use of other guns. I felt that if I had enough time I would place them in. I did leave in the visuals of having weapons so the HUD did not feel empty. Lastly on the HUD I have a score counter, it adds 20 to each creature kill. I was going to add points based on which weapon you were using, but since I did not implement weapons the scoring for that fell through. Now the background of the HUD is all textured, it uses a diamond plate as its texture, and I also created bullet holes in it by blending in a bullet hole with a white background on top of the diamond plate. This gave the HUD an interesting look to it and I was happy it turned out that way. As I was creating the background I had a problem getting it to blend right, I was using Nehe’s online tutorial to blend it, and it was not coming up right, the white background of the bullet hole was not being blended with the background. I found out that the mask I was using was getting blended right, but not the image I wanted to see, so I removed the mask and the image then was blending right. To go along with the HUD, I also have a Pause Menu which pauses the game, and allows the user to exit the game, save their place in the level, or to load their last save file. The background for the Pause Menu is also like the HUD and has the diamond plate with bullet holes in it. Overall the visuals came out pretty well and I am happy for that.

**Controls**

Next on my list for the game was to get the controls to work properly. As I said before the player can move forward, backward, and side to side using the WASD keys (W to move forward, A to strafe left, S to move backwards, and D to strafe right). At first I had A and S turning the player because I did not have the mouse working correctly, but now you use the mouse to turn the player left and right and also to shoot. How the player determines how to move forward and backwards is by computing the calculating where the player is looking at and then moving them. This Code shows how I did the movements:

zmove += float(cos((xRot \* 3.14159265)/180))/10;  
xmove -= float(sin((xRot \* 3.14159265)/180))/10;

You have to get the Radian of the angle you are looking at and the take either the cosine or sine of that to get the actual position you are moving to, and the dividing by 10 is so you go slower that way you are not flying across the screen. I also used these formulas, one for strafing but I had to add or subtract 90 from xRot so you move right and left respectively, and the other for when you have the walls you get bounced back farther so you done get stuck in the wall and all you have to do is get the inverse of the number. One bug that I still get, but rarely, is when the player moves into a wall, if they press the opposite direction button they may be able to go through the wall. This happens because the player moves when they are hitting the wall and it reverses the direction the wall may send them. After I got that to work for the player I also used it for the creatures so they can move around, but the creatures differ by they take one step when they move and find another direction and then repeat the move. The only problem with that is that the creatures can look like they are in the walls since it is checking the center of the creature. Right now the AI for the monsters is not working correctly. When the player gets close the creatures are suppose to focus the player and head for the player. I have used several formulas to determine the angle the creature needs to look at to be focusing the player, but each one had its own problems. At first I use the Dot Product got it working but the monster would focus at angle 0 and not at the player, I figured out since I was using the player’s vector for the monster that it would never always look at the player. The other formula, I cannot remember what it is but what was happening was that when the player got close to the creature that the creature would disappear. I figured out after awhile that the math was causing a -1.#IND000 to appear in the values of the monster. Something was not converting correctly or causing the float to infinite or not a number. The only way around this is to keep trying out different formulas to determine the direction.

**Other Concepts**

The other concepts that I am going to talk about are my level loading, saving, loading, and scoring. I talked about the scoring in the Visuals section, but I came across some problems with it. First is that I wanted to have a high score list but I found out that adding an input field in OpenGL and SDL was a lot harder then what I thought it would be. If I wanted to have one I would have to create it from scratch, and I did not know where to start and did not understand how to get it to work. If I had more time I would have tried to work on it to get it to work. Next I implemented a save file for the user to use. It records the player’s position, direction, health, armor, lives, and score. Also it keeps track of whether the creatures are alive, their position, and their direction. If you open the file it is nicely detailed and the like, so it is easy to “hack” the game to get more lives or score. With the load it goes through the save file to load the game, and updates the game to match the save file. Nothing fancy with the loading other than it updates the player first then goes through the list of creatures and updates them. Finally how I have a feature to load levels, if I had multiple levels I could change the layout of the walls, doors, items, creatures. It goes through and loads the walls into place, next to goes through the doors, then item, and then the creatures. This does take awhile for the game to fully load compared to if I were to add them in a function versus a file, but I feel that the file helps me if I wanted to add more levels then all I would have to do is make a new file with the coordinates of the objects and then I can load them into the game instead of making more lines of code to do the same thing. Last thing I worked on was sound, the only things that I used sound for was when the player shot the weapon, and when the monsters died. Since sound was not a main focus of my project I felt that this way enough for my game at the moment.

**Grade**

Last part of my paper is my grade. If you look at the last page of my paper you will see a chart of the points per object and with it how many point per object I believe I should have gotten. From this chart I have determined that I will get a 100 out of a 110 which would be an A. The chart has been updated from the proposal to exclude some objects that was too difficult to accomplish, along with that so has my grading curve. Below a 55 would be an F, 55 – 64 would be a D, 65 – 79 would be a C, 80-94 would be a B, and 95-110 would be an A. I dropped out keeping a Highscore sheet mainly because I could not implement an input text field, and I also decrease the controls by five points since I had no reason to use the number keys. If I could have gotten weapons implemented there would have been a good reason to use some of the numbers, but since there is only one weapon it is kind of hard to use the number keys. Overall I felt that this project taught me some valuable time in Time Management. There were days I would try to mainly focus on this project, and then I would fall behind on my other classes, and then as I caught up with my other classes I felt that I would just drop this project and pick it up like a week later. Learning new algorithms on how to solve simple things like movement was fun, seeing the movement work and that it kept my respective x coordinate and z coordinate sound so I could do things like collision detection and shooting was awesome when I got them to work. Overall I had fun this project and have learnt a lot of new and neat things about OpenGL, SDL, and C++.

|  |  |
| --- | --- |
| Points | |
| Lightning Effects | 20/20 |
| Controls   * Movement keys * Mouse keys | 10/10   * 5 * 5 |
| AI | 15 5/15 |
| Textures | 20/20 |
| Visual Effects   * UI * Menu Screen | 15/15   * 5 * 10 |
| 3-D objects | 10/10 |
| Sound | 5/5 |
| Saving/Loading   * Saving * Loading | 10/10   * 5 * 5 |
| Scoring System   * Keeping track during the game | 5/5   * 5 |
| Total | 100/110 |