Alexander Munoz

Partners: Marcus Loe, Derrick Joe

CSE 165

Project Overview:

Team Name: Nerds Inc.

Project Title: Space Wars 3

The group will attempt to make a game that pays homage to old top down shooters from yester – year. It will mimic games such as Space Invaders and Galaga with some minor differences. The game has a functioning main menu with several options such as the ability to see high scores on the particular machine, as well as controls for the game. Once the game begins, alien ships will spawn on the top of the scree, and the player’s ship will appear at the bottom section of the screen. The score will display on the bottom of the screen.

All of the objects are derived from a simple “Rectangle” class for simplicity sake. Whenever two rectangle objects collide, the objects will be pushed back in the list of objects. This will make it appear as if the objects are getting destroyed.

enemySprites.push\_back(new Enemies((float)-0.8, (float)0.7, (float)0.1, (float)0.1));

The game has simple controls; the arrow keys move the ship around, shooting the missiles is spacebar, restarting the game is ”r” and returning to the menu is the escape key. The simple objective of the game is to clear the screen of the alien ships. Once the screen is cleared, the game will end and the player will win:

if(spritesKilled == 26){

gameOver = true;

state = GAMEWIN;

redraw();

}

Once the game is over, regardless of either winning or losing, the score will be checked through simple logic checks. The score will be compared to the other scores in the high score file and will be replaced if necessary. The app.cpp file will ultimately save the score in this manner.

|  |  |
| --- | --- |
| void App::saveScore(){ |  |
|  |  |  |
|  |  | ofstream file; |
|  |  | file.open("highscore.txt", ofstream::out | ofstream::app); |
|  |  | a |
|  |  | file << score << endl; |
|  |  | file.close(); |
|  |  |
|  |  | ifstream rfile("highscore.txt"); |
|  |  | string line; |
|  |  | vector<int> scoreArr; |
|  |  | while (getline(rfile, line)){ |
|  |  | int num = atoi(line.c\_str()); |
|  |  | scoreArr.push\_back(num); |
|  |  |  |
|  |  | } |

**Individual Contribution/Division of Labor:**

The artificial intelligence was an awkward obstacle to achieve. For the most part there was very minimal need for an AI to properly function in the project. The game only requires the ships to move down the screen.

|  |
| --- |
| gluBuild2DMipmaps(GL\_TEXTURE\_2D,3,theTexMap.GetNumCols(),theTexMap.GetNumRows(), |
| GL\_RGB, GL\_UNSIGNED\_BYTE, theTexMap.ImageData()); |  |
|  |  |

With the use of a simple map, the ships will be set into predetermined spots once drawn into the scene. The only time the game will successfully beat the player is if the alien ships pass up the player’s y axis point.

|  |
| --- |
| for (int i = enemySprites.size() - 1; i >= 0; i--){ |
|  |
| if (enemySprites[i]->test->getY() > -1 && paused && !checkHit[i]){ |
|  |
| enemySprites[i]->test->setY(enemySprites[i]->test->getY() - 0.0009); |
| }else if(enemySprites[i]->test->getY() > -1 && paused && checkHit[i]){ |
|  |
| enemySprites[i]->test->setY(enemySprites[i]->test->getY()); |
|  |
| }else{ |
|  |
| state = GAMEOVER; |
| break; |

Either getting physical contact by the moving enemies or if they will pass the getY value, the game will end and the game over screen will appear.

**Lessons Learned/ Additional Comments:**

One major flaw with the groupwork was the lack of iteration control. Through discord, a platform used for communication, the group would toss around ideas with one major working project at all times. The group only used github till the end to check if any major mistakes would occur on runtime with other computers. If any major errors would have occurred implementing a new function or even multiple, major issues would have arisen. Luckily, the group didn’t have any major issues with that. After attempting to make such a simple game through a combination of C++ and OpenGL, it is much clearer the necessity to have proper parent classes defined. Having easy to implement classes makes the creation of other minor objects much more streamlined. The group made a fairly archaic model of a rectangle class; although it was simple, it made other objects easier to make.