Yacine Manseur

5/13/15

Professor Sable

Interactive Engineering Graphics

Final Project Write-up: Space Invaders

My Final Project was a three-dimensional version of the classic game Space Invaders. This game includes 55 initial enemy aliens and a spaceship that faces these enemies. The purpose of the game is to control the spaceship and aim lasers at the enemies. For each enemy destroyed, the player scores a point. In addition to shooting enemy aliens, the user also has to be weary of the aliens shooting back at them. The player starts off with 10 health points and loses one hit point every time the player is hit. When all 55 aliens are successfully destroyed, the player progresses to the next level, where the aliens are recreated and their lasers move faster. When the player loses all 10 health points, the game is restarted and the player’s high-score is saved locally. This high-score is initialized to zero and updates every time the player successfully kills an alien.

The player’s spaceship is comprised of rectangular prisms and is modelled after the same layout as the classic spaceship used in the arcade game. The aliens are also comprised of multiple rectangular prisms to come together as the classic alien spaceship. These alien spaceships constantly alternate in a circle, moving up, then right, then down, and then left. This movement is the alien’s way of making it difficult to be hit by the player. The player’s spaceship also holds the laser, and the laser is unattached from the spaceship when the spacebar is hit. There is also a laser constantly moving towards the player, and this laser is reset in front of one of the 11 front-facing aliens. When the aliens either get a hit or miss the player, the laser is reset. If there are no aliens in a column and that column is chosen, another column will be chosen to shoot the laser. This makes it very difficult for the player to get rid of the few remaining aliens because the laser will constantly be coming from the same spaceship. Collisions are defined by 9 different cases where a laser can enter a spaceship: the four corners of a spaceship, the four sides of a spaceship, and then the rare case where the laser manages to be fully enclosed by the spaceship. An explosion animation was created to simulate a spaceship actually blowing up. This was done by creating 50 randomly places cubes 3 different times, depicting miniature parts breaking apart. The laser also contains a line of 10 light sources that span the length of it. These light sources follow the path of the laser, illuminating its surroundings as it passes through the scene.

When the game is initially opened, the game is paused so that the player can let the scene load and have time to read the instructions. The arrow keys move the space ship in four different directions. The spacebar is used to shoot a laser from the player’s spaceship. To restart the game, the user can press Shift+N at any time. Once the player is done reading the instructions, they may press P to un-pause the game and start playing. The instructions are very intuitive, but the words take up a lot of the screen. The player can press H at any time to hide these instructions so that they do not obstruct their view of the game. When a level is completed, the game is paused once again so that the player can prepare themselves for the onslaught to come. When the player runs out of health, there is an alert that tells the player their final score and when ‘OK’ is pressed it resets the game.

The game can be found at <http://ymanseur.github.io/SpaceInvaders/>. This is my first time using github to host a project, but it seems to work very well. I will also attach the files to run the game locally, although the font used is found over the internet. To run the game locally, open index.html using either Mozilla Firefox or Internet Explorer. If you wish to run the game using chrome, you must go into cmd and type the following command: C:\Program Files (x86)\Google\Chrome\Application\chrome.exe –allow-file-access-from-files. (Note that there are two dashes before “allow”).