

IS51016A: Audio Visual Computing

Coursework 2 Report

Brock Craft

Akira Fiorentino (33330644)

Uyen Le (33365715)

3rd April 2015

**Word Count: 732**

(excluding headings, subheadings, subtitles and references)

**Executive Summary**

Our program is a first person 3D survival horror game. The player is trapped in a creepy area and the aim is to stay alive as long as possible without falling off while escaping from surprise attacks from the giant cubes. Altogether, this serves the purpose of the game is to create a thrilling, exciting and scary experience that requires quick action from the player.

**Features and User Interactions**

Why did we develop this project?

Initially, the concept of making a first person video game has always appeared interesting to us, in terms of its graphical representation and constrained viewpoint of the player.

We discovered that there is a strong correlation between the first person platform and horror / maze-like games. In other words, this includes using 3D rendering graphics to create real-life representation of objects, an avatar-based mechanic that creates directional, limited player perspective, and positional audio to elevate the excitement of the player. Furthermore, this game type allows us to get creative by creating our own storyboard, theme, music,… All this creates a truly fascinating yet addicting user experience that we want to achieve.Therefore, as a result, a horror survival game was born.

User interactions:

The interactivity of the game is a simple control system where users press W, A, S, D keys to move around while using mouse to change their viewpoints with the exception of the y-axis (i.e. always on the floor).

Restricting the perspective of the player was achieved by adding a spotlight looking down over the player. This is so that the player can only see what’s in front to a certain distance, while the tiles and enemies get darker the further they are from the player. This creates a thrilling experience as the player cannot know direction of the boxes.

For the enemies and their functionalities, the program constantly creates and recycles these boxes with randomised speeds, directions and sizes in the dark, that won’t disappear unless they’re far away from the player. This is so that the appearance of boxes is more unpredictable and natural.

The enemies’ (i.e. boxes) job is to kill the player by pushing him off the tiles. If so, he will fall down to his death and the game over screen is displayed with the score (the amount of time stayed alive).

How does it relate to the course?

* + Graphics:

The visual appearance and graphical design are a major focus of the program. Using 3D renderer has allowed us to achieve a minimalist visual style that we like, along with colour scheme, fonts,… that follow the dark, mysterious forest theme to produce the scary atmosphere.

Also, black and red are the main colour scheme, with a bit of green for the tiles as grass to spice up the design a bit. For this we mapped the fill colour of the tiles according to the distance from the player position. We’ve also carefully chosen bold, black-letter fonts for the introduction and game over texts.

* + Lights and Camera:

We used a combination of lights functions and an external camera library called OCD. This camera isn’t centred on one object but allows for actual movement in any direction, immediately this turns the program into a first person perspective. The spotlight effect moves along with the player and is placed above, looking down.

* + Transformations and Physics:

Functions such as *rotate()* and *translate()*, *pushMatrix()* and *popMatrix()* have been fully utilised throughout the game in drawing objects and positioning them (ex: the floor at (0, 940, 0)). Therefore, the challenge was to understanding the 3D coordinate system carefully.

Physics was used in implementing the functionalities of boxes and falling action of player. If one enemy gets too close to the player, it triggers a collision. This collision then immediately triggers a sudden acceleration in a certain direction (based on the relative position of the enemy) that pushes the player away. The x-axis direction is reversed when the player falls.

* + Audio control:

Background music contributes a important part to the user experience, therefore we’ve used Minim library to add an eerie and chilling song. We used *setGain()* to reduce music volume, so that that a brown-noise effect in the background can be heard as the boxes gets close. The noise volume is based on the box that is currently closest to the player. This effect acts as a hint so the player is aware of incoming enemies even if they're behind him, and generally allows for a more immersive experience.

**Development**

Structure of code, methods and classes used, external sources:

// Akira’s part

**References**

1. First person video game: [en.wikipedia.org/wiki/First\_person\_(video\_games)](http://en.wikipedia.org/wiki/First_person_%28video_games%29)
2. Music: *Dream Maker* by Thomas Bergersen: <https://www.youtube.com/watch?v=k20NwCro8ig>
3. OCD: <http://gdsstudios.com/processing/libraries/ocd/reference/>
4. Minim: <http://code.compartmental.net/minim/index.html>
5. getLineOut(): <http://code.compartmental.net/minim/minim_method_getlineout.html>
6. setGain(): <http://code.compartmental.net/minim/audioplayer_method_setgain.html>
7. Brown Noise: <http://code.compartmental.net/minim/noise_class_noise.html>

Processing:

1. P3D: <https://processing.org/tutorials/p3d/>
2. spotLight(): <https://www.processing.org/reference/spotLight_.html>
3. millis(): <https://processing.org/reference/millis_.html>
4. pushMatrix() and popMatrix(): <https://processing.org/reference/pushMatrix_.html>