Queen Mary University of London

Department of Electrical Engineering & Computer Science

ECS657U/7003P – Multi-platform Game Development

Final Game Report

SIMULA

Hasan Emre Erdemoglu

Mate Krisztian

Xsara Baptiste-Wilson

7 December 2020

Table of Contents:

1. **Game Repository:** https://github.com/heerdemoglu/ecs7003-simrunner
2. **Gameplay Video:** https://www.youtube.com/watch?v=zyu9WipMYg4
3. **Game Concept:**

An application called Simula is found on a PC. The application is password protected; a password is requested for authorization. Upon a clicking the button to enter the password the application throws you into a simulation. The application is corrupted and overrun with bugs, there is no way to terminate the simulation manually, to survive the player has only one option to play the Simula and try a complete the simulation. When the simulation begins it recognizes the player is unauthorized and starts to collapse, the player must move forward traversing across platform to avoid the collapse and survive.

The game should be fun and exhilarating as you need to keep moving forward. The simulation will try to disorient you throughout your journey. The simulation misleads the player into believe that they could reach an exit when the simulation finishes; however, there is no end nor exit. Eventually the player will get consumed by the simulation, either you will fall, or get caught in the collapse. The only thing that you can do is survive for as long as you can.

4. **How to Play:**

The game has tank type control. A, D buttons are used to rotate character body to the left and right respectively. W, S can be used to move forward or back. Sprinting is available by LEFT SHIFT button and jumping is available by SPACE button. If you are tired or need to change game options button P can be used to pause the game and to access these menus. Beware that you cannot jump once you are off the ground, so think twice while you want jump through an obstacle; your lack of coordination may lead to your early demise.

5. Feature List:

5.1. Environment Features:

The aim is to develop an environment which is cyber looking; dark and unstable. The shards and platforms light the way, illustrating the path the player is on.

The scripts for this part can be found in Scripts/PCG and the prefabs can be found under Prefabs/Tiles.

5.1.1. Platforms:

There are two types of platforms: solid monochrome and translucent multi-colored. Solid monochrome platforms do not have any light emitting from them do not disintegrate. Multi-colored platforms are fragile and disintegrate into shards as you step on them. (This disintegration is not implemented into final game as it was buggy at the time of submission).

These two tiles represent stability and instability of the simulation that the player is in, while also directing the direction the player must go.

5.1.2. Shards

Shards are illuminating pieces of tiles that once formed the instable platforms. In game story they represent the broken platforms from previous player trials, floating and drifting in the environment. As game environment does not have any light emitting; they are also the only source light in the simulation other that the zone light when in range.

5.1.3. Path blocks

Path blocks are hand crafted sets of platforms which are grouped together. These blocks can later be sent to PCG routine to construct endless valid paths for the player to traverse.

5.1.4. Procedural Endless Environment Generation

The level design is linear and endless. New shards (as light source and environment) and platforms are procedurally generated as the players continue their journey. The paths are constructed from pre-defined set of path blocks which are stitched together.

While this implementation uses a linear path design, with new pathing block design a semi-linear game world can also be generated. The game mechanics is very compatible for such changes/additions.

5.2. Gameplay Features:

5.2.1. Player Controls

The game uses tank style controls. The body of the player is rotated by A (Left) and D (Right) and the movement is achieved by W (Forward) and S (Backward). Left Shift is used for sprinting and Space is used for jumping. Necessary scripts are under Scripts/Player Scripts. Also note that player can pause at any time using “P”.

5.2.2. Platform Interaction

The platforms grapple the players when they are within their vicinity. The platform will force the player to change its rotation (with the camera) to disorient the player. Note that the gravity still works the usual way. The necessary code can be found under Scripts/Player Scripts.

5.2.3. Accessibility Settings

Players can choose to quickly pause the game at any time using the keyboard key “P”. This will also prompt an options panel, where the master audio and sensitivity options of the player can be changed and saved. This is particularly useful with different screen resolutions which may affect movement mechanics. This menu is built-in to the Game Scene under the name “GameCanvas”. Note that the home screen is a different scene.

5.2.4. The Zone & Difficulty Levels

In the game’s story, the player is an intruder to the system and the system will react to their existence. A zone will chase the player from behind eliminating the player if it catches the player. The zone will progressively get faster to ensure that even the most skilled player failing at some point. The zone will show a visual indication of its distance via the zone light as it gets closer to the player. Necessary scripts can be found under Scripts/Game Control Scripts

5.3. Non-gameplay Features:

5.3.1. Home Screen

As soon as you open the application the initial scene loads and there is a fade in effect via display as the camera transition from alpha 1 transparent to 0.

5.3.2. Pause Panel

The pause panel is activated through the user pressing the “P” key, when activated the time scale for the game is set to 0. When the game is paused the user has the option to return to the Home screen, Restart the simulation and access a settings panel.

5.3.3. Setting Panel

When paused the setting panel can be activated via a button to allow the user to make available adjustment to personalize their game play, such as the games music volume, player acceleration and player rotation speed. The user has the choice of closing this panel and returning to the pause screen by clicking a button or for a quicker way to return to game play press the “P” key again.

5.3.4. Death Panel

Upon the death of the player is presented with two buttons, presenting options to play the simulation again or return to home screen. Upon the selection of one of these buttons there is a fade out before the corresponding scene is loaded.

5.3.5. Scoreboard

Players can enter their names to record their name to the Hall of Fame. Note that this functionality is not fully implemented hence disabled. The top players who lasted the longest in the simulation (highest timer score) can be displayed in a high score table.

5.3.6. Fade

Upon entering any scene in the game there is a fade in affect to add to the ominous atmosphere of the game and the notion of entering and leaving the simulation.

5.3.7. Exiting the application

The user can exit the game at any time during game play by pressing the “Escape” key.

5.4. Audiovisual Features:

5.4.1. Background Music

The background music is created by us using the Cubase Digital Audio Workstation and a MicroKORG MIDI synthetizer. The intense pulsing of the music helps to emphasize the tenseness of the game world fiction. The volume of the background music can be manually adjusted by the user in the settings menu by adjusting the volume slider. This value then is passed to the AudioManager through the PlayerController object.

5.4.2. Character Animations & Sounds

For character animations the game makes use of free assets from Adobe’s Mixamo.com. The model and the animations are all from this site, but all are modified to our needs. All animation avatars were converted to humanoids, so any model with the same bone structure could be used.

The switching between the different animation states is handled by a multi-dimensional blend-tree. The animator object keeps a list of attributes (float and boolean) to get the current value at each frame. The movement related values (velocity x-y-z) are floats and depending on their values they blend towards the state with a higher value. This creates smooth transitions between the states rather than a harsh sudden change.

The handling of audio files and clips are done by using a custom-built Audio Manager script. This singleton class keeps a single collection of sounds and for each sound the clip, source, volume, pitch, and loop attributes.

Sound effects were sourced from fresound.org and a free speech-to-text converter. As mentioned above are handled by the AudioManager. The volume of the footsteps is dynamically set by the value of the forward velocity.

5.4.3. Light Sources & Ambiance

The light sources are based on the shards and multi-colored platforms. These sources are point sources illuminating their local areas. The multi-colored platforms are used to cue player about the path that they must take. The shards light sources add depth to the environment, creating a dim atmosphere. The zone also has a strong directional light indicating that it is close to the player. When the player is in range of the zone, the zone light is made visible, acting as a warning system to make the user aware of its approach. The distance between the player and the zone determines the zone light's intensity. If the distance between the zone and the player is decreasing intensity increase, while if the distance increases the zone light's intensity will decrease until not visible. The dynamic adds a layer of adrenaline and excitement to the game play.

5.4.4. Camera Effects

Camera has rotational effects. The camera is rotated whenever player displaces itself to another tile with a different orientation. This effect intends to disorient the player while also making the progression more challenging.

Camera also has fade in and out effects while transitioning from home screen to the Game Scene. When fading in and out the alpha value of the black texture applied to the camera is modified in accordance with the transition. The same effect is also applied when restarting the game from Pause or Game Over states to allow a smooth transition between game states.

5.4.5. Materials

To promote an almost-broken simulation feeling; prototype texture packs are used. The transparency of these materials is increased, and materials are given light sources of respective colors to give light to, otherwise dark, environment.

5.4.6. Simula “Speech” Audio

To add to the game concept of Simula being an old corrupted application the simulation “voice” is distorted and irregular. This speech audio is used in the home menu and game over screens.

6. External Asset List:

This list can be found in this project’s GitHub repository with more detail.

6.1. Tile Textures:

For the final project “Grid box Prototype” textures were used.

6.2. Player Model & Animations:

For the final project, “YBot” model is used from this website. The animations are from the same website; bundled with the model.

6.3. Audio & Voiceovers:

Voice lines are generated by Text to Speech by using this website. Game sounds (such as background) is made by Mate Krisztian. Human walking sound effect is taken from this link. Human Breathing effect is taken from this link. Landing and Jumping effects are here and here respectively.