HandyGames™ Recruitment Project

Pedro Nunes

This project is divided into three parts: Map Generation, Character Spawning, HUD Creation.

Map Generation

To generate the map, a Game Object was created to handle the spawn of all clones of the prefabs, such as the possible blocks, the ramps and all 6 different characters. This game object, named Spawner, contained the script that will handle the generation of components.

To set the base of the map, 8 blocks were added to serve as a basis, and calculate the coordinates of the components to be spawn. From that point, blocks are generated layer by layer, and in each layer, a block is added from the left column to the right one. In the first layer, each block can be generated on the ground or first level, which is randomly set. If the first level is chosen, an additional block is added, to remove whitespaces.

The remaining layers are completed after, which follow the same procedure, but allowing to reach a second level higher. The blocks in a column have a one fifth chance of being replaced with a different type, making most of the blocks in a column the same type. If a block is on a higher level than the one on its left, than a ramp is added to its left. If, instead, the block is lower, a ramp is added to the top of the block. When a ramp going down is added, the Spawner ensures that the next two blocks will be at the same level, creating one buffer between different ramps.

Due to the randomness of the map, the level of the first block in each layer is taken into account, and it is calculated the coordinates for the characters to be spawn.

Character Spawning

Every second, the Spawner randomly decides to spawn a bug or a hostage, with fifty percent chance each. Although, if there is already a hostage on the screen, or there is not possible to spawn a hostage, a bug is always spawn instead. If a hostage is spawned, two bugs are immediately spawned before and after the hostage, with reference to the hostage that they are leading.

Each character will move on the X axis, entering the screen and using its colliders to detect ramps or the destructor situated on the right side of the map. The destructor has the purpose for a character to delete itself upon entering in contact with it, indicative that the object is outside of the screen.

When a character encounters a ramp, the character is slowly rotated to match the inclination of the ramp, moving now also on the Y axis. Upon reaching the end of the ramp, the character rotates back to the normal position, and to ensure the correct course is followed, the character is forced to snap to the respective Y coordinates that it's supposed to follow.

Using a Game Manager script, with the objective of handling the information relative to the game, it can be used to maintain the status of the game, such as bullets remaining or hostages alive/killed. With the Game Manager, whenever the left mouse button is clicked, a Raycast can be launched from the camera to the cursor position to analyse which objects were selected. To verify if a character was selected, with no obstacles covering it, all colliders are analysed, and if the top character is on the same or lower layer than the top obstacle, then the character is visible, and is indicated that the character was hit, spending a bullet.

When a character is hit, the dead animation is triggered, in which they fade out of the screen and are destroyed. Additionally, if the character hit was a hostage, after fading out, they fade back in the top right corner, scaled down and rotated on the side, with its dead sprite. The exact position of where they fade in, are saved by the Game Manager, to ensure all saved/killed hostages are in the right positions.

When a bug leading a hostage is hit, it sends a message to its hostage, which upon having both leading bugs killed, the hostage is considered free. A freed hostage then directly moves to its position on the top right corner, while scaling down to the right size.

Whenever a hostage is free, killed or goes out of bounds (as in, enters in contact with the destructor and is destroyed by it), the hostage informs the Game Manager, which informs the Spawner to allow it to spawn another hostage.

Audio cues were added to represent when a character is killed, which is the same for all types, but there is a different audio cue for each type of hostage when it is saved.

HUD Creation

A pause menu was added to pause the game by pressing escape, allowing it to resume again by pressing the same key. Additionally, buttons were added that also allow to resume the game, and also start a new level, generating a new map, or to quit the game

After all the hostages are saved or killed, or when the bullets run out, the game over screen appears, with the information relative to the game over, and buttons that allow to restart the game or quit it. Additionally, a main menu before the game is started with a quit button. A background music was also added that plays from the start of the game.