Labyrinth of the Goblin King

Project Report

Labyrinth of the Goblin King is the First-Person Shooter I developed for my Game Design CA1. It is a dark fantasy dungeon-crawling game, where the player must fight their way through the Goblin King’s labyrinth, killing goblins and grabbing treasure along the way. To escape from the maze, the player must leap into the glowing portal at the other side!

# Code Process

The game began as a simple platform with several capsules and cubes dotted around the area. While most of these objects acted as obstacles, one of the capsules was scripted to act as a player. This was achieved by making the camera object a child of the player object, following as it moved. The camera was restricted to simulate a person’s gaze, clamping the player’s look range similarly to how the human neck is limited.

A screenshot of a cell phone

Description automatically generated

The next step was to add enemies; by applying a Nav Mesh Agent to the enemy and “baking” a Nav Mesh onto the plane used as a floor, I could tell the “enemy” objects to track and follow the player. With that completed, I added a “weapon” empty to fire “bullet” prefabs.

Using Unity’s in-built collisions system, I gave the enemies HP, which would be removed when they collided with the bullets. I also gave the player HP, which would be removed when the enemies collided with the player. Finally, if the enemies ran out of HP, they would be removed from the game. If the player ran out, the game would end.

A person standing in front of a brick wall

Description automatically generatedWith that, the basic prototype was complete! I began implementing graphics, adding textures to the ground and obstacles and swapping out my enemy capsule for a goblin mesh. The next step was adding animations; In the end, I had added several simple animations such as fade-ins and fade-outs when changing scenes, as well as more complicated ones such as the spawn and movement animations for the goblins; which required scripted triggers.

Qr code

Description automatically generatedAfter giving the game some of the aesthetics I wanted to implement in the final application, I added more in-depth features such as a Heads-Up display (HUD) and an enemy spawner. The HUD was a simple canvas element, upon which I attached several text elements and images to. Various scripts in the project would send information to these elements such as remaining time and score, which would then be displayed to the screen. The player’s health bar works similarly but converts the health “number” into a flat colour image, which is scaled to fill a “health bar” on the HUD (Brackeys, 2020).

The most important piece of my game was replacing the preset obstacles with a maze to navigate through. While I could have simply made a basic maze, I thought it would be more fun if the game featured randomly generating mazes. I managed to implement this with the addition of a recursive algorithm (Nambiar, 2020). Using this, the script sets a grid of walled cells, then draws a random path through them. After this first path, it backtracks through any unvisited cells, ensuring that every cell in the maze is theoretically reachable by the player. Using this algorithm, I then had the script apply a series of “wall” prefabs to build the maze in the scene.

Text

Description automatically generated

Above, the code checks if there is an unvisited cell to the left of the current cell. This code repeats for each direction and ensures that no cell is left unvisited, before flagging which walls will need to be filled in by the Maze Rendering script.Text

Description automatically generated

This is a sample section of the Maze Renderer script; shown above the script checks if the cell is flagged for an upper wall.

Once the maze was fully assembled, I added the “exit portals” and “treasure” pickups. Both of these use a variant of the standard colliders, using the Trigger functionality instead of the standard collisions. This means that instead of causing a collision, the object detects when another enters its radius. The treasure object adds to the player’s score before deleting itself, and the exit portal (A shimmering green particle-effect orb, (Aguiar, 2017)) loads the victory scene.

Once finishing the important elements of the game I made some simple Menu, Win, and Game over scenes with some simple animations, then moved on to tweaks and minor fixes. I added sound effects and background music, and changed my skybox from a solid black to a glowing subterranean surface, to further reinforce the aesthetic of the game.

Most sound clips including the music and most effects were created by myself on the Beepbox site (Nesky, n.d.), but I had sourced a handful of them: the goblin hit (SFX, n.d.) and death (Koenig, 2009) sounds, as well as the portal’s warbling hum. (Ch0cci, 2006)

# Imported assets used in my game

In the creation of my game, I used a variety of assets from the Unity Asset Store to decorate the world and bring life to the level. From the player’s crossbow, to the lights, to the goblins, the asset store proved to be invaluable in the creation of my game.

Crossbow Asset (Noman)

Ground tiles (Indie)

Goblin (Kozhemyakin)

Wall texture (Studios)

Crystal Lights (SineVFX)

Treasure pickup (shop)

Skybox (Yughues)

# References

Aguiar, G. (2017, May 20). *Unity 5 - Game Effects VFX - Glowing Orb*. Retrieved from Youtube: https://www.youtube.com/watch?v=ctmqr\_8esT0&t=265s&ab\_channel=GabrielAguiarProd.

Brackeys. (2020, February 9). *How to make a HEALTH BAR in Unity!* Retrieved from Youtube: https://www.youtube.com/watch?v=BLfNP4Sc\_iA&ab\_channel=Brackeys

Ch0cci. (2006, February 1). *Alien Ship Idle*. Retrieved from freesound.org: https://freesound.org/people/Ch0cchi/sounds/15347/

Indie, P. (n.d.). *Dungeon Ground Texture.* Unity Asset Store.

Koenig, M. (2009, August 13). *Gagging Sound*. Retrieved from Sound Bible: http://soundbible.com/844-Gagging.html

Kozhemyakin, A. (n.d.). *Goblin.* Unity Asset Store.

Nambiar, S. (2020, May 22). *Maze Generation Unity Tutorial*. Retrieved from Youtube: https://www.youtube.com/watch?v=ya1HyptE5uc&ab\_channel=SandeepNambiar

Nesky, J. (n.d.). *Beepbox*. Retrieved from Beepbox: beepbox.co

Noman. (n.d.). *Crossbow [Medieval Weapons Pack].* Unity Asset Store.

SFX, F. (n.d.). *https://freesfx.co.uk/Category/Rubber/382*. Retrieved from Hard Rubber Impact.

shop, F. (n.d.). *Treasure Set - Free Chest.* Unity Asset Store.

SineVFX. (n.d.). *Translucent Crystals.* Unity Asset Store.

Studios, G.-R. (n.d.). *Tileable Bricks Wall.* Unity Asset Store.

Yughues, N. /. (n.d.). *Yughues Free Ground Materials.* Unity Asset Store.