

Assignment #1

Instructor: Avinash Sharma

Released: 19th March 2021

In this assignment, we will build a 2D game. In doing so, we aim to get comfortable with making objects of greater complexity with OpenGL, working with many interacting components and understanding how applications are built with OpenGL.



Figure 1: Among Us

In this task, you will be required to create a 2D maze game derived from the **Among Us** game, in the form of a single-player variant. To help you get started, you can use the [Hello-World](#) template given in the previous assignment. You must use **OpenGL(version 3+)**.

1 Task

This game's objective is to exit a maze you have been dropped into after finishing two tasks. You must hurry because reactor meltdowns do not fix themselves.

1. World: A procedural maze (different maze layout on every game startup). Maze needs to have a **single exit**, and if the player reaches this point after completing **two tasks**, the player can exit the maze. The walls of the maze are visible as lines on the screen (2D).
2. Player: A single-player with some assigned health value can navigate the generated maze with the keyboard.
3. Light: The lighting of the game can be turned on or off using a keypress. When switched off, you should have a small light source illuminating the area around you, which allows you to see a small part of the world around you. Bear in mind that the light cannot penetrate the maze walls, and the rest of the maze should remain in darkness. The number of seconds spent in dark mode should boost the points of the player.
4. HUD (Heads Up Display): A box of rendered text on the top left of the screen, present at all times, detailing the following:
 - Health: This indicates the player's points, which can increase or decrease based on events/interactions in a manner defined by you.
 - Tasks: 'Tasks completed / Total number of tasks available.'

- Light: 'On/ Off' indicated the status of the central lighting system.
- Time: A countdown from some X number of seconds. When the clock runs out, end the game.

5. Tasks

- An imposter character (enemy) exists at a random location in the maze and begins to get to your position as soon as the game begins. A button (a circle drawn in the maze path) to vaporise (make the imposter disappear) the imposter exists at another random location in the maze. The task here is to reach the button before the imposter reaches you. Use a pathfinding algorithm to make the imposter reach you as you move around. Coming in contact with the imposter ends the game.
 - A button to release power-ups and obstacles exists in a random part of the maze. The task is to reach the button and then collect the power-ups to increase your score. You can represent power-ups/ obstacles using geometric 2D shapes. Coming in contact with an obstacle decreases your score.
6. Characters: Draw and colour the player and imposter like real Among Us characters with basic shapes like a semicircle plus a rectangle stacked for the body and another rectangle for the glass part of the helmet.



Figure 2: Player / Imposter

2 Bonus

1. Use a shortest path algorithm instead of brute force for the imposter to find the player(you).
2. Draw and colour the power-ups and obstacles to signify some real-world object like a coin and a bomb.
3. Add a 'You Won' / 'Game Over' / 'You Lost' screen to enhance gameplay.

3 Marking Weightage

Marks will be given out of 100.

- **World:** 20
- **Player:** 10
- **Light:** 10
- **HUD:** 20
- **Tasks:** 20
- **Characters:** 20

We have given less weight-age for the lighting since we understand that it is challenging to implement. However, do try your best; having lighting effects is rewarding and makes your game instantly pop.

4 Instructions

1. Please write well-commented code in C++, and make sure to submit your own work.
2. The submission directory should be structured as:

```
1 .  
2 |- src  
3 |- assets  
4 |- CMakeLists.txt
```

Name the directory after your roll number and submit a zip before **11:59 PM on March 27th**.

All the best!