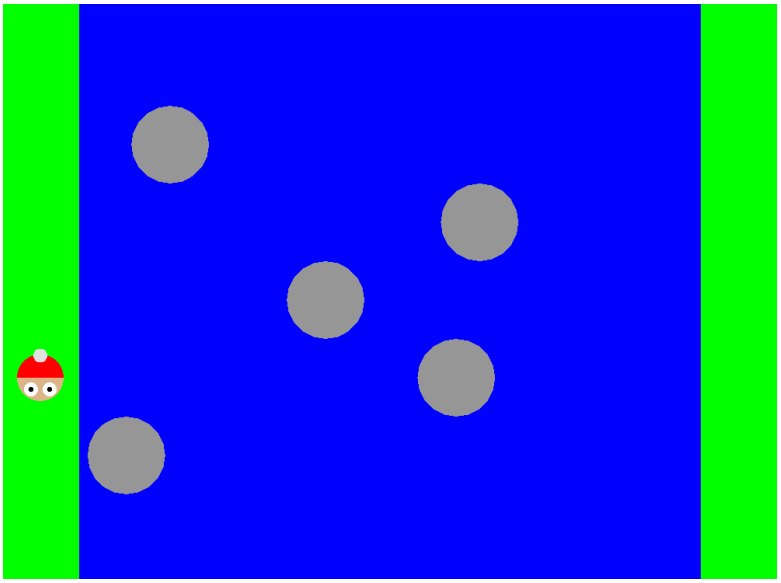
CSL7450: Computer Graphics
Assignment: 1

Instructor: Dr. Avinash Sharma
Release Date: 31 / 01 / 2025

Submission Date: 11 / 02 / 2025
Total Marks: 100

Create a 2D game titled "Portal Adventurer" using OpenGL. Following is the brief description of the game:

- Rick, while sitting in his room, was suddenly teleported to an unknown part of the world via a portal. He needs to find a way back home.
- The game should feature a bird-eye view stationary camera (top down)
- At each level, the player starts at one point of the map (teleports in via a portal at that point), and has to carefully step across moving platforms to reach the endpoint of the level (to teleport out via an exit portal).
- If the player falls off the platforms, he/she respawns at the start of the level.
- The player will have to collect keys along the way (placed on certain platforms) in order to unlock the exit portal for that level.
- Aside from the risk of falling off the platforms, the player will have to watch out for moving enemies (Enemy must be appropriate to the environment). If the player comes in contact with an enemy, player health should decrease.
- Create 3 levels, each containing a different biome.



Example Biome: River

Detailed discussion of each component

1. **2D World:**
    a. The world consists of 3 maps. The player starts at the entry-point of map 1, and progresses through the maps one by one (exit portal of map 1 teleports the player to the entry-point of map 2, and so on)
    b. Once the player travels through the exit portal of map 3, they have completed the game.
    c. Each map should be a different biome, with start point, end point, platforms and enemies appropriate to the theme of the biome. For example: If the map is a river, the start point can be the left bank, end point the right bank, platforms can be moving rocks and enemies can be alligators.
    d. Use your creativity to come up with interesting maps.
    e. The platforms on the map must be constantly moving as per your choice, but allow the character to eventually reach the end point via switching between them
    f. Each map should have 3 keys to be collected, placed on different platforms. The exit portal should only allow entry after collecting the 3 keys.
    g. The enemies movements should be randomized on every run of the game (Example: An alligator periodically jumping across a platform can be randomized to target different platforms on each run)
    h. All the map assets (geometry) to be created from scratch however external textures if used (Example rock texture applied on the platform or water texture applied on the river) are permitted.
    i. The camera should be stationary, and top down (can be slightly angled as long as whole map is visible at once)
    j. When a player exits a map (to enter the next one), the camera should teleport to the next map to show the player spawning in.

2. **Character:**
    a. The character design choice is up to you. All the vertices, indices, colours and textures (if any applied) for the character should be created by you, and be suitable for the camera angle chosen.
    b. The character can move around by pressing W,A,S,D. Additional keys for other actions can be used (explained in section 4).
    c. The character should have 3 lives associated with them, and a starting health of 100. Should their health be reduced to zero (by falling off the

platforms or being harmed by the enemies) they will lose a life, and spawn at the start of the map they were on (Each map start can be considered as a checkpoint). However, losing 3 lives will result in a GAME OVER, and the player will have to start from the beginning (map 1) again.

d. The player's health and remaining number of lives should be visible on the screen at all times during the gameplay.

3. **Screens:**
    a. Main Menu:
        i. Should contain 2 options: NEW GAME and LOAD GAME.
        ii. NEW game starts a new game,
        iii. LOAD game loads the checkpoint of the last game (if it had been closed before completion). To achieve this whenever the user closes the game, save the state variables of the current checkpoint such as map number, lives remaining, and health into a text file (or any other format).
    b. **(Optional)** Cutscene on starting a new game, showing the character getting sucked into a portal before the game begins.
    c. Game screen:
        i. Shows the view from camera
        ii. A HUD at the top of the screen to display the player health, no. of lives left, current map number, and total time passed since the start of the game.
    d. GAME OVER and YOU WON screen
        i. GAME OVER screen will have the option to go back to the main menu.
        ii. YOU WON screen will show the total time required to complete all 3 maps.
    e. **(Optional)** Cutscene on completing the game before YOU WON screen to show the player reaching back to their room.

4. **Unique Mechanic:**
    a. Implement a unique mechanic for each map. For example in a river biome the mechanic could be the ability to jump in order to reach different rocks, for a jungle environment the mechanic can be to swing across a wine to reach a key, for a space biome the mechanic can be antigravity to float around freely, etc.
    b. Use your creativity according to the biome.

c. Can use an additional input key other than WASD for this mechanic (ex. SPACEBAR, E, etc.)

Marks Distribution

- 2D World **(60 marks)**
  - For each map,
    - Start, end, keys and platforms logic *(10 marks)*
    - Enemies movements and collision logic *(5 marks)*
    - Creativity and map aesthetics *(5 marks)*
- Character **(10 marks)**
  - Character design and creativity *(5 marks)*
  - Movement, health, lives, respawn logic *(5 marks)*
- Screens **(15 marks)**
  - Main menu, YOU WON, GAME OVER *(5 marks)*
  - Checkpoint save and load *(5 marks)*
  - HUD: ie- player stats, time elapsed, etc. *(5 marks)*
- Unique Mechanic **(15 marks)**
  - Each map unique mechanic
    - Creativity *(2.5 marks)*
    - Implementation *(2.5 marks)*

Submission Guidelines

1. Programming language: Python (PyOpenGL) / C++ (OpenGL)
2. OpenGL version: Legacy / Modern (recommended)
3. Submit the code as a single zip file titled ***"RollNumber_A1.zip"*** with all the code files and assets neatly organized. The zip file should also contain a readme file specifying in detail how to run the code.