MINOR ASSIGNMENT-06

Game Programming with C++ (CSE 3545)

Publish on: 10-05-2025 **Submission on:** 15-05-2025
Course Outcome: CO_5 **Program Outcome:** PO_4 **Learning Level:** L_5

Problem Statement:

Experiment with SFML **`sf::View`** (i.e. 2D camera that defines what region is shown on screen) to divide the game up into layers and development of player & zombie classes for the Zombie Arena game.

Learning Objectives:

Students will be able to scroll, rotate or zoom the entire scene without altering the way that their drawable objects are drawn and also able to design the required classes for building the Zombie Arena game engine.

Answer the followings:

1. The graphical assets make up the parts of the scene of the Zombie Arena game. State how many graphical assets are present in our game. Also give a count of different images in `background_sheet.png` file.

Count of graphical assets and images

2. The sound files in Zombie Arena game are all in `.wav` format. These are files that contain the sound effects that will be played when certain **events** are triggered. Write the sound file names and when it would be played.

Sound files & purpose

3. Write the code snippet to create a view from a rectangle as well as a vew from its center and size.

Create view

4. The view in SFML is like a 2D camera. It controls which part of the 2D scene is visible, and how it is viewed in the render target. The new view will affect everything that is drawn, until another view is set. Write the SFML-C++ statement(s) to set a view to be displayed in the window and draw everything related to it.

Create view

5. Write the appropriate variable name(s) and/or function name(s) in the place of **?** symbol assuming the image **player.png** has a resolution of 50×50 pixels.

```
 class Player{
   private:
      Texture ?
      Sprite ?
   public:
      Player();
};
Player::Player(){
        ? . ?("player.png");
        ? .setTexture(?);
        ?. setOrigin(?,?);
        ? . ?(120,234);
}
```

Complete the code snippet

6. Design a player class with optimal number of private and public members to draw the player sprite at the center of the defined view of the window.

Player class

7. The following Figure:1 shows event handling by polling. Develop a code snippet to instantiate an object of `Event` type (`Event` is a SFML class type) to poll for the system events. Additionally, include a loop with the condition `window.pollEvent(...)` to keep looping each frame until there are no events to process and display the appropriate message when there is a state change.

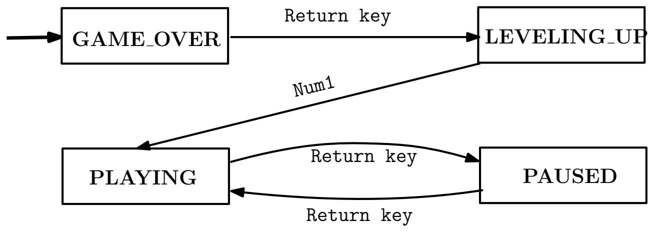

Figure 1: Game state transition for handling events by polling

Code Snippet

8. Re-write the `spawn` public member function for the **Player** class to spawn 5 players as shown in the below Figure-2. Also state the SFML statements to call the **`spawn(...)`** function and **`window.draw(...)`** for the player.

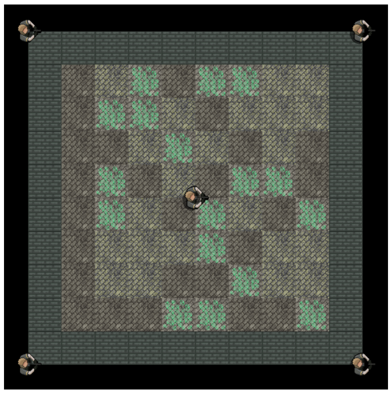

Figure 2: Player spawn at different position of the arena

Code Snippet

Code Snippet

9. Design a public member function, **`void spawn(float startX, float startY, int type, int seed)`**, for the **Zombie** class to draw the 3 kinds of zombies over the arena as shown in the below Figure-3. Also write **THREE** function call statements to invoke the **`spawn(...)`** function with three independent objects of that class **Zombie**.

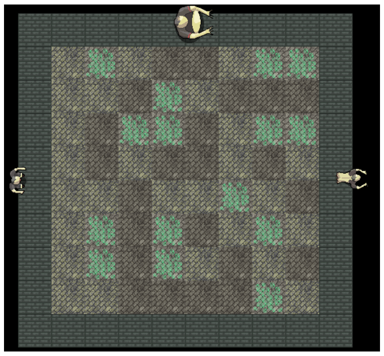

Figure 3: Zombies spawning over the arena wall

Code Snippet

10. Redesign the above code snippet to use array of objects rather than 3 individual objects of the **Zombie** class. Don't write the **`spawn(...)`** function again.

Code Snippet

11. Redesign the above code snippet to use dynamic allocation of objects (using new) rather than array of objects of the **Zombie** class. Don't write the **`spawn(...)`** function again.

Code Snippet

12. Write SFML-C++ statement to compute the angle between the player location **`(x1,y1)`** to the BLOATER position **`(x2,y2)`**. Additionally, set the rotation of the BLOATER zombie sprite (i.e. `m_Sprite`) to that angle.

Code Snippet

13. Assume that a zombie sprite, `m_Sprite`, is to the left of the player's position (i.e. `Vector2f playerLocation`). Write SFML-C++ statement to update the zombie position variable (`m_Position`) w.r.t. the player.

Code Snippet

14. State the code segment to keep the player (**`m_Position.x`** & **`m_Position.y`**) is NOT beyond any of the edges of the current arena (`m_Arena`) with the surrounding wall tile size, `m_TileSize=50`.

Code Snippet

15. Write the code segment to generate a random number between 80 and 100.

Code Snippet