MINOR ASSIGNMENT-03

Game Programming with C++ (CSE 3545)

Publish on: 22-03-2025 **Submission on:** 30-03-2025
Course Outcome: CO_3 **Program Outcome:** PO_3 **Learning Level:** L_4

Problem Statement:

Experiment with Timber game mechanisms; *Time is always running out, Getting more time by chopping the tree, chopping tree causes the branches to fall, the player must avoid the falling branches and repeat until time runs out or the player is squished* using C++ with Simple and Fast Multimedia Library(SFML).

Learning Objectives:

Students will be able to learn arrays, functions, enum class, texture, sprite, text, font, rectangle shape and keyboard handling for player's input etc. in SFML-C++

Answer the followings:

1. Write the code snippet to create two objects for each of the given classes; **Text**, **Font**, **SoundBuffer**, **Clock**, **RectangleShape** and **FloatRect** respectively.

Code Snippet

```
Text text1, text2;
Font font1, font2;
SoundBuffer buffer1, buffer2;
Clock clock1, clock2;
RectangleShape rect1, rect2;
FloatRect fr1, fr2;
```

2. Fill out the places marked with the symbol, **?**, in the following code snippet.

```
Texture ?;
?.loadFromFile("sample.png");
Sprite ?;
?.setTexture(?);
?.?(960,540);
```

Code Snippet

```
Texture texture;
texture.loadFromFile("sample.png");
Sprite sprite;
sprite.setTexture(texture);
sprite.setPosition(960, 540);
```

3. The **`pollEvent`** function of **RenderWindow** class puts data into the **event** object of the **Event** class that describes an operating system event. There would be many events stored in the queue. Construct a while loop to check for the **event** types **KeyPressed** and display a message, `Keypressed Detected` on the `std`, if that event has occurred.

Code Snippet

```
Event event;
while (window.pollEvent(event)) {
    if (event.type == Event::KeyPressed) {
        std::cout << "Keypressed Detected" << std::endl;
    }
}
```

4. Write the code snippet to detect the Keyboard input, **`enter`** key, from the user using **Keyboard** class and also display a message, *Enter Pressed* on the output stream, if **`enter`** key is pressed.

Code Snippet

```
if (Keyboard::isKeyPressed(Keyboard::Enter)) {
    std::cout << "Enter Pressed" << std::endl;
}
```

5. SFML plays sound effect using two different classes; `SoundBuffer` and `Sound`. Write the code snippet to set up the sound effect that would be played on the Timber game event *player runs out of time*.

Code Snippet

```
SoundBuffer outOfTimeBuffer;
outOfTimeBuffer.loadFromFile("out_of_time.wav");

Sound outOfTimeSound;
outOfTimeSound.setBuffer(outOfTimeBuffer);

// Play the sound when the player runs out of time
outOfTimeSound.play();
```

6. Create a program to generate a random 1-D array of size 10. Additionally shift each of the array elements 2 positions to right and filled the beginning two places with random elements from the range 20 to 30. Finally display both the arrays.

Code Snippet

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    srand(time(0));
    int arr[10];

    // Generate random array
    for (int i = 0; i < 10; ++i) {
        arr[i] = rand() % 100;  // Random values 0–99
    }

    // Display original array
    cout << "Original Array: ";
    for (int i = 0; i < 10; ++i) cout << arr[i] << " ";
    cout << endl;
```

```
 // Shift right by 2
    int newArr[10];
    newArr[0] = 20 + rand() % 11;  // Random
between 20–30
    newArr[1] = 20 + rand() % 11;

    for (int i = 2; i < 10; ++i) {
        newArr[i] = arr[i-2];
    }

    // Display shifted array
    cout << "Shifted Array: ";
    for (int i = 0; i < 10; ++i) cout << newArr[i] <<
" ";
    cout << endl;

    return 0;
}
```

7. In Timber game, when the player chops, `logActive` is set to be `true`. Write the SFM-C++ code snippet in a block that only executes when `logActive` is `true` to handle a flying log.

Code Snippet

```
if (logActive) {
    // Move the log
    logSprite.setPosition(
        logSprite.getPosition().x + (logSpeedX * deltaTime.asSeconds()),
        logSprite.getPosition().y + (logSpeedY * deltaTime.asSeconds())
    );

    // If the log moves off screen
    if (logSprite.getPosition().x > 1920 || logSprite.getPosition().x < 0) {
        logActive = false;
    }
}
```

8. We know that the timber game must end badly with either the player running out of time or getting squashed by a branch. Write the code snippet to detect and execute when the player is squashed by a branch and also display message **`Player Expired-Game End`** at the center of the screen.

Code Snippet

```
if (branchPosition[playerSide] == side::LEFT || branchPosition[playerSide] == side::RIGHT) {
    // Player is squashed
    paused = true;
    acceptInput = false;

    messageText.setString("Player Expired-Game End");

    // Center the message
    FloatRect textRect = messageText.getLocalBounds();
    messageText.setOrigin(textRect.left + textRect.width / 2.0f,
                      textRect.top + textRect.height / 2.0f);
    messageText.setPosition(1920 / 2.0f, 1080 / 2.0f);
}
```

9. State the algorithmic steps to express the actions that would be taken while the player press the **right key** when `bool activeInput = true;` in timber game.

Code Snippet

1. Check if activeInput is true.
2. Check if Right key is pressed.
3. Increase the score.
4. Move player sprite to the right.
5. Update the log sprite direction.
6. Set playerSide = RIGHT.
7. Reset timeRemaining.
8. Set acceptInput = false.

```
if (acceptInput)
      {
if (Keyboard::isKeyPressed(Keyboard::Right))
        {
            // Make sure the player is on the right
            playerSide = side::RIGHT;
            score++;

            // Add to the amount of time remaining
            timeRemaining += (2 / score) + .15;

            spritePlayer.setPosition(850, 520);

            spriteAxe.setPosition(AXE_POSITION_RIGHT,spriteAxe.getPosition().y);

            // Set the log flying to the left
            spriteLog.setPosition(810, 720);
            logSpeedX = -5000;
            logActive = true;

            // Update the branches
            updateBranches(score);

            acceptInput = false;
        }
```

10. Write the sequence of SFML-C++ statements to draw the sprites; **X**, **Y**, **Z**, and **Zombies[5]** on to the game window.

Code Snippet

```
Sprite X;
X.setTexture(xTexture);
window.draw(X);
window.draw(Y);
window.draw(Z);

for (int i = 0; i < 5; ++i) {
    window.draw(Zombies[i]);
}
```

11. Consider the following C++ code snippet;

```
class A{private:int x,y;
  public:void setXY(int x1, int y1){x=x1;y=y1
      ;}
    void getXY(){cout<<x<<" "<<y<<endl;}
    ~A(){cout<<"END"<<endl;}
};int main(){A a1,a2;
   a1.setXY(10,20);a2.setXY(40,50);
   a1.getXY();return 0;}
```

Output

```
10 20
END
END
```

12. Consider the following C++ code snippet;

```
enum colour{
    blue, red, yellow
};
int main(){
    enum colour c;
    c=yellow;
    cout<<c<<endl;
}
```

Select the Correct one

- ❑ 1
- [✓ 2]
- ❑ 3
- ❑ None of the above

13. Consider the following C++ code snippet;

```
enum hello{
    a,b=99,c,d=-1
};
main(){
    enum hello m;
    cout<<a<<" "<<b<<" "<<c<<" "<<d<<endl;
    return 0;
}
```

Select the Correct one

- ❑ 0 1 2 3
- ❑ 0 99 100 101
- [✓ 0 99 100 -1]
- ❑ None of the above