POPL-2 CONCURRENCY PROJECT DOCUMENTATION

Trap Shooting Model Game

Description of the game:

Enemies are randomly generated and sent towards the player, the player has to shoot the enemies. If the enemy crosses the border on the player's side, health of the player gets decremented. The score of the player gets incremented when the enemy is shot.
Upward key to rotate the rifle up.
Downward key to rotate the rifle down.
Space Bar to release bullets.

main():

This function declares a pointer which points to the game object.
It displays the loading GIF for three seconds and then calls the function to display the main window.

Game class:

This is subclass of QGraphicsView having class variables:

QGraphicsScene *scene
MyRect *player
Score *score
Score2 *health

In the **constructor**, we create the scene dynamically and set its size and background.
It has the following member functions:

void displayMainMenu() : This function displays the menu with the title "TrapShooting Model Game" and options Play and Quit.

void starti() : This function creates the player,score and health dynamically and adds them all to the scene in their respective positions. It also sets the focus on to the player using setFocus() function. It also plays the background music of the game in a loop. It then spawns the enemies every two seconds.

MyRect class:

It is a subclass of **QObject** and **QGraphicsEllipseItem**. It has a member function **keyPressEvent()** which takes the input from the keyboard and sets the rotation of the rifle and releases the bullets. If space key is pressed, then a new bullet is added into the scene and we try to play the sound of bullet being shot in a separate thread.

It also has a public slot **void spawn()**, which creates an enemy and adds it to the scene.

Bullet class:

It is a subclass of **QGraphicsPixmapItem** having a public slot **void move()**. In its constructor, we set the **QTimer** for the bullet to move periodically by connecting the timer with the **move()** slot.
In the **move()** function, we keep a track of the colliding items and check whether this bullet is colliding with any enemy. If yes, we delete both the bullet and the enemy. Otherwise we move the bullet forward in its original path by a small amount.

Enemy class:

It is a subclass of **QGraphicsPixmapItem** and has the public slot **void move()**. In its constructor, we randomly set the y coordinate and set the image of the enemy. We then set the **QTimer** for the enemy to move periodically by connecting the timer with the **move()** slot. We check the score of the player and set the speed of the enemies accordingly.

In the **move()** function, we move the enemy forward by incrementing the x coordinate. We also check whether the enemy has crossed the boundaries of the scene. If yes, then we delete the enemy.

Score class:

It is a subclass of **QGraphicsTextItem** and has member functions **void increase()** and **int getscore().** In its constructor, we initialise the score to zero, set the font and size of the text item and add it to the scene. The **increase()** function, increments the score by one and the **getscore()** function returns the value of the score.

Score2 class:

It is a subclass of **QGraphicsTextItem** and has member functions **void decrease()** and **int gethealth()**. In its constructor, we initialise the health to ten, set the font and size of the text item and add it to the scene. The **decrease()** function, decrements the score by one and the **gethealth()** function returns the value of the health.

Button class:

It is a subclass of **QGraphicsRectItem**. It has three member functions,**void mousePressEvent()**, **void hoverEnterEvent()** and **void hoverLeaveEvent()**. The **mousePressEvent()** checks if the mouse has been clicked and emits the signal **clicked()**. The **hoverEnterEvent()** checks if the mouse pointer has entered the button and changes the color of the button. Similar functionality is shown by **hoverLeaveEvent()**.