Bubble Shooter

Multimedia Systems Project (Group 4)

Under Dr.Priyambada Subudhi

Team members :

➡Abhishikth Boda (S20210010044)

➡Manikantha Rayudu(S20210010230)

➡Preetham Jayam (S20210010096)

Abstract

- This report presents the design and implementation of a bubble shooter game utilizing multimedia system techniques.
- The project aims to create an engaging and interactive gaming experience through the integration of various multimedia elements such as text, sound, and Image and audio.
- The report covers the design process, including the selection of appropriate multimedia technologies and the development of game mechanics.
- The game allows players to shoot bubbles in a given direction, with the player's score increasing upon successful elimination of bubbles.
- Bubble Shooter game made with *pygame* module some of the important features are :
    - Game Start Button
    - Bubble elimination upon matching colours
    - Dynamic bubble generation upon game starting
    - Score tracking and display
    - Game over upon completion of bubbles
    - Bubble Shooting mechanisms

PROBLEM STATEMENT :

Develop a Python program using the pygame module to create a 2D Bubble Shooter game with advanced multimedia functionalities. The game should incorporate various multimedia elements such as images, audio, and text .

REQUIREMENTS :

- **Python** : Python is a high level programming language it is easy to understand the code by everyone
- **Pygame** : Pygame is a Python library designed for creating video games and multimedia applications.
- **Any Code Editor** : IDE's like VsCode , Sublime test

Packages Used :

1. pygame
2. math
3. copy
4. time
5. Sys
6. os
7. random
8. pygame.gfxdraw
9. pygame.locals

Code Explanation :

- **In Our Project , there are main 3 classes :**
  - **Bubble**
  - **Arrow**
  - **Score**

```
class Bubble(pygame.sprite.Sprite):
    def __init__(self, color, row=0, col=0):
        pygame.sprite.Sprite.__init__(self)

        self.rect = pygame.Rect(0, 0, 30, 30)
        self.rect.centerx = int(strx)
        self.rect.centery = strY
        self.speed = 10
        self.color = color
        self.radius = bubblerad
        self.angle = 0
        self.row = row
        self.col = col
```

```
class Ary(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)

        self.angle = 90
        arrimg = pygame.image.load('Arrow.png')
        arrimg.convert_alpha()

        arrowRect = arrimg.get_rect()
        self.image = arrimg
        self.transformImage = self.image
        self.rect = arrowRect
        self.rect.centerx = int(strx)
        self.rect.centery = strY
        self.arrow_length = 1500  # Length of the pointing line
        self.speed = 1  # Reduced speed
```

```
class Score(object):
    def __init__(self):
        self.total = 0
        self.font = pygame.font.SysFont('merlin', 35)
        self.render = self.font.render(
            'Score: ' + str(self.total), True, black, white)
        self.rect = self.render.get_rect()
        self.rect.left = 5
        self.rect.bottom = winhgt - 5
```

Main Functions :

1. RunGame() :

This function implements the main gameplay loop for a Bubble Shooter game, handling user input, bubble launching, collision detection, and scoring.

It utilizes Pygame for graphics, sound, and event handling to create an interactive gaming experience.

2 . Creating Game Environment:

✦ **MakeBlankBoard() creates the empty board .**

✦ **SetBubbles(array,gameclrlist) populates the game board array with Bubble objects of random colors from the provided color list.**

✦ **DrawBubbleArray() this function iterates through the provided 2D array representing the game board and draws each non-blank bubble object onto the display surface.**

✦ **SetArrowPosition() adds the arrow into the game**

3 . Updating the Game :

✦ **UpdateColourList() extracts unique colors from the bubble array and returns them as a list.**

✦ **The main purpose of this is to remove already completed bubbles in the board.**

4 .Shoot Bubbles:

✦ **The function ShootBubbles(bbarr, newbb, launchbb, score) handles collision detection between the launched bubble and existing bubbles.**

✦ **Updates the game board accordingly, triggers bubble popping sound effects, and manages score updates based on bubble elimination.**

5 .Pop Bubbles:

- **PopBubble() recursively identifies and collects bubbles of the same color connected to the given bubble position (row, col) within the bubble array, storing their positions in the delList list.**
- **CheckFloatingBubbles() this function will check the floating bubbles and removes from bubble array.**
- **PopFloatingBubbles() recursively removes floating bubbles connected to a given bubble position (row, col) within the bubble array, maintaining stability in the game environment.**

6. Remaining Simple Functions :

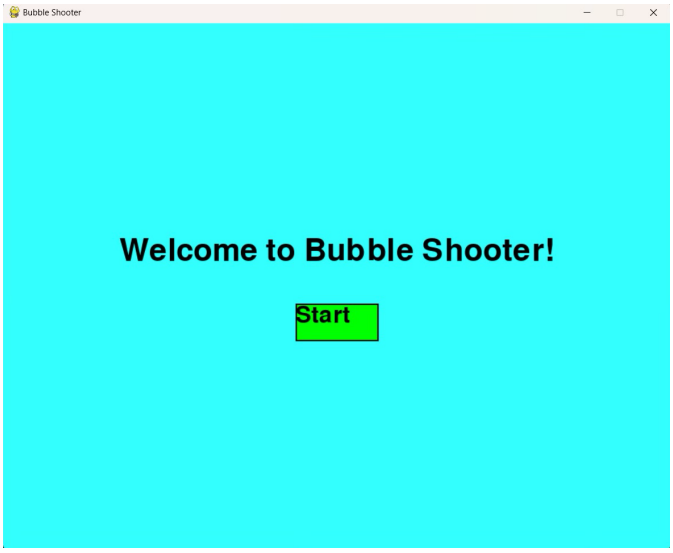
- **ColourOfNextBubble() this function is responsible for creating the next bubble at the bottom right corner**
- **DrawBubblesArray() drawing the each non blank bubble in the bubbles space.**

7. End Screen :

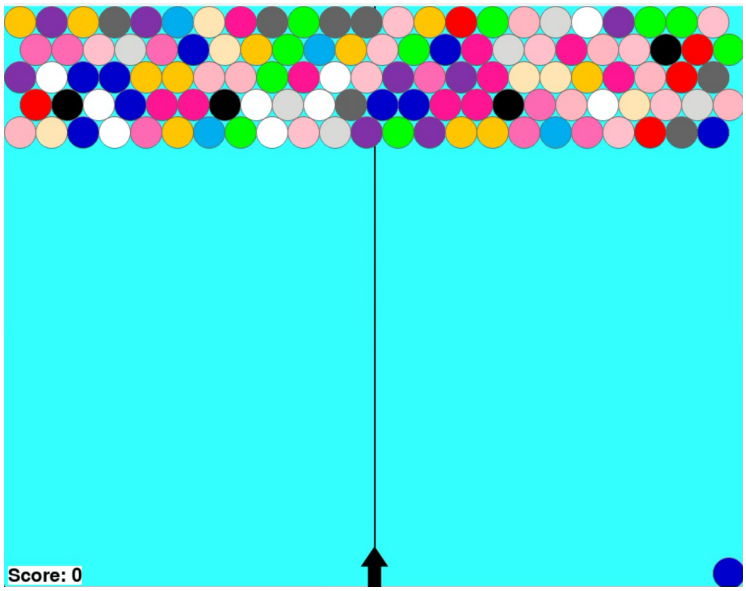
- **The function endScreen(score, winorlose) displays an end screen message indicating whether the player won or lost, along with their final score, and prompts them to either play again or quit the game.**

Screenshots of our Project :

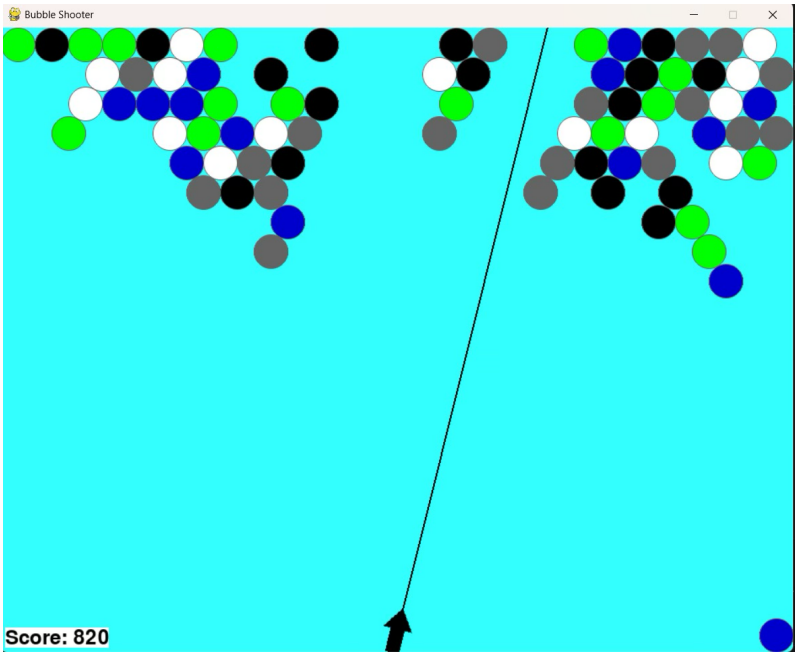
1) Start Screen :



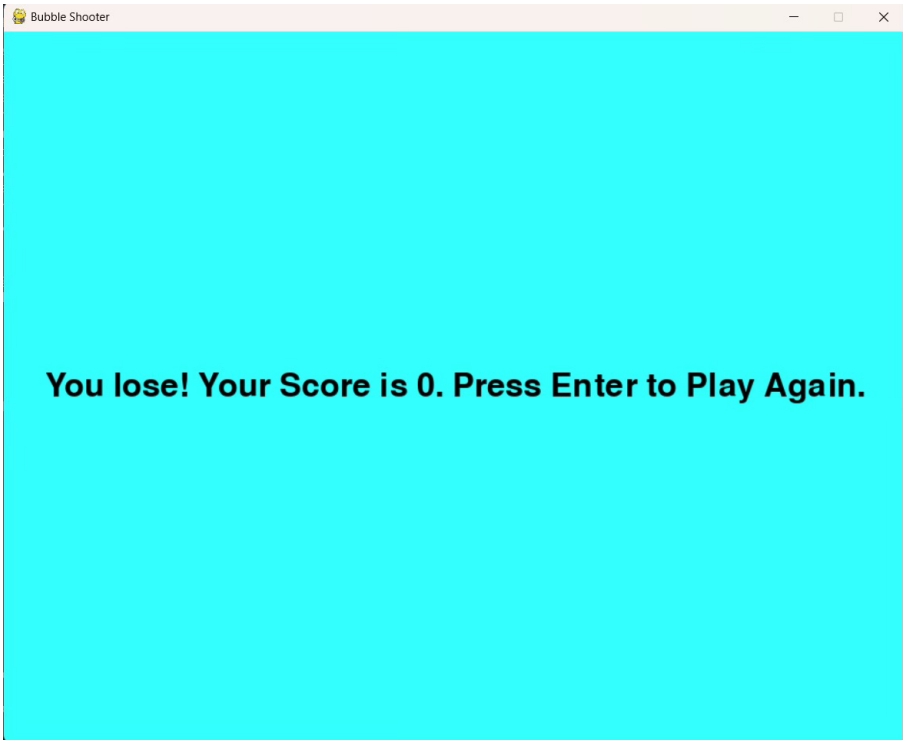
2) Game Initial Screen:

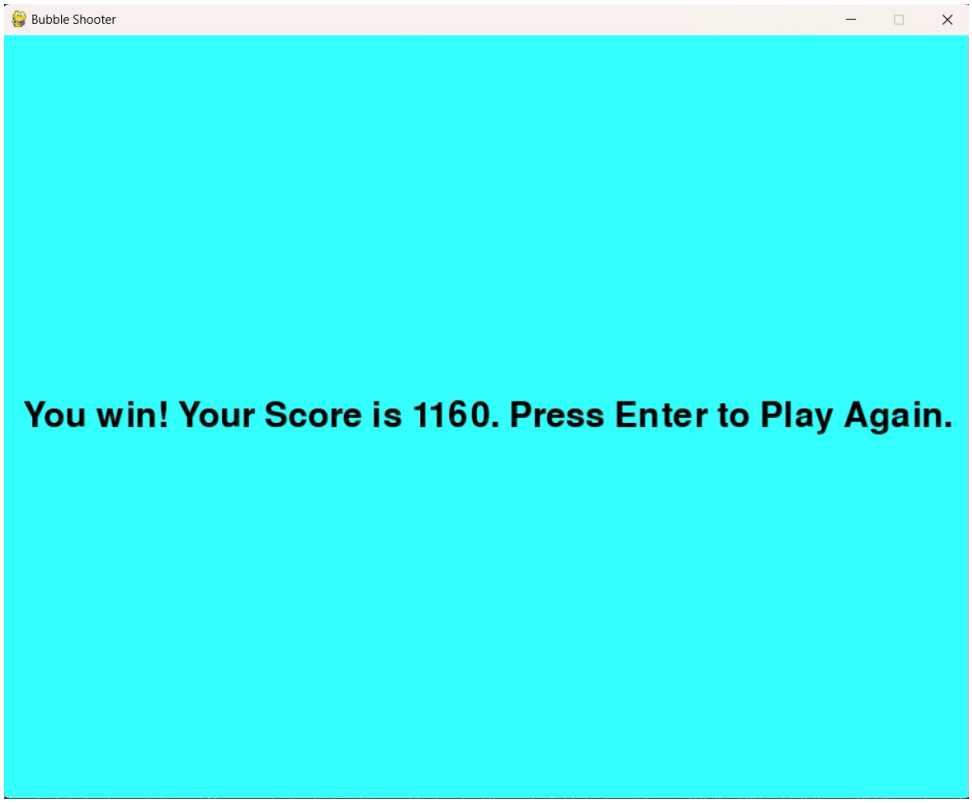


3) Middle Of the Game :



4) Final Result Screen:




Bubble Shooter
You win! Your Score is 1160. Press Enter to Play Again.

Future Improvements :

- ✦ We can try to implement difficulty levels in the game. Such that we can choose then while the start of the game.
- ✦ Easy level(has 3-4 colours) and Hard Level(has 12-15 colours)
- ✦ We can also try to include many graphics in our game like bubbles falling animations, etc
- ✦ We can also try to add a leaderboard such that all the scores can be maintained

Thank You !!!