Collision detection -

Collision detection was the most important part of the game. According to the rules, if there is a collision between a dragon and the egg, the score has to be updated and the egg and to respawn in a different coordinate. If there is a collision between a dragon and the border, the game should immediately quit.

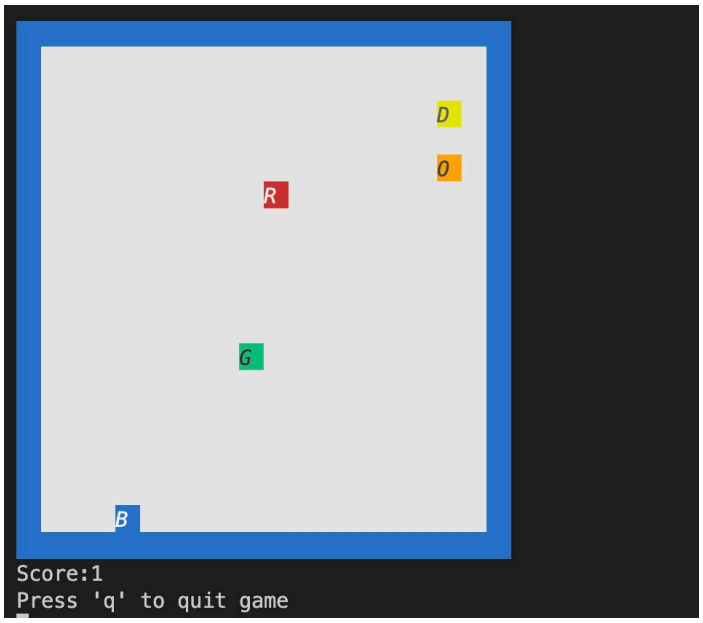
Test 1 - Collision detection between dragon and egg

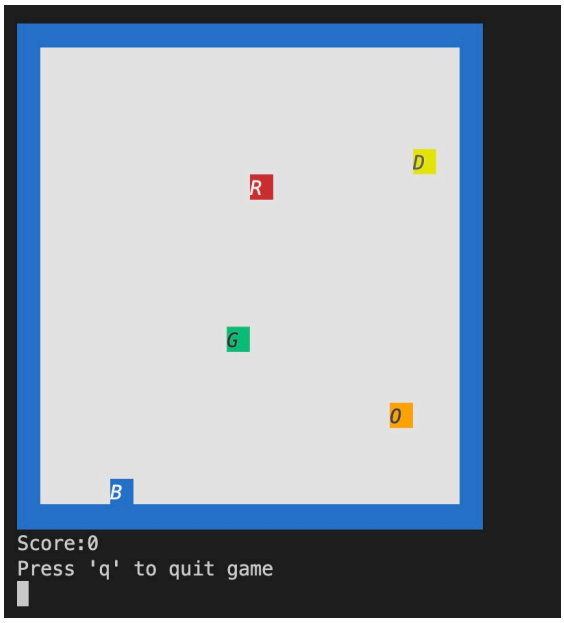
The following code was used to detest a collision between a dragon and egg -

```
// Collision detection between the dragon and OrangeEGG
else if(dragon_x == oE_x && dragon_y == oE_y){

    temp_score = p1.get_current_score();
    dragon_score = temp_score - 1;                    // subtracting

// OrangeEGG relocation after collision
    oE_x = rand() % (cols - 2) + 1;
    oE_y = rand() % (rows - 2) + 1;

}
```

The expected output would be -
- An updated score
- A respawned orange egg

The output was -





The egg respawned and the score updated successfully.

Test 2 - collision detection between the dragon and the border

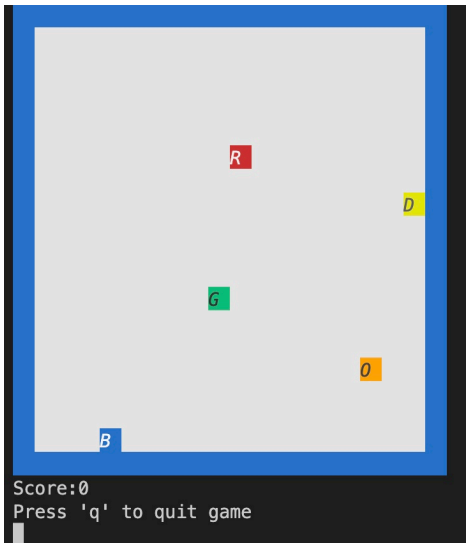
The game should end immediately if the dragon crashes into the border. The code for th following feature is -

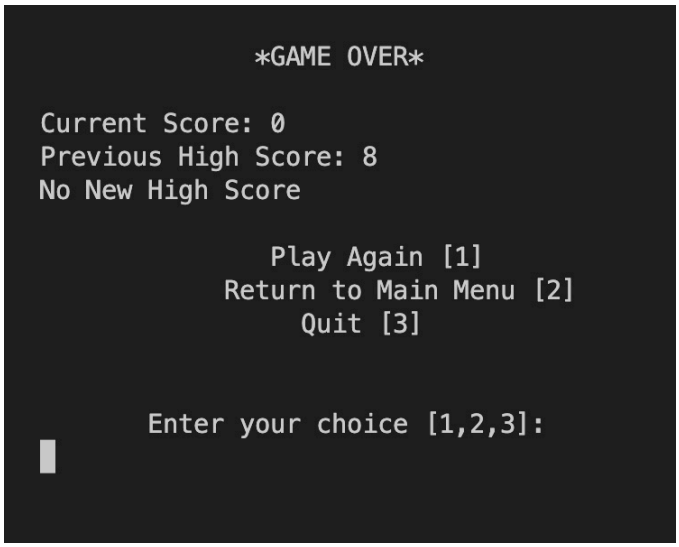
```
// Collision detection between the Dragon and the edge
// If the the dragon collides with edge, game over
if(dragon_x < 1 || dragon_x > cols - 2 || dragon_y < 1 || dragon_y > rows - 2){
    game_over = true;
}

else if(dragon_x == rE_x && dragon_y == rE_y){
temp_score = p1.get_current_score();

if (temp_score >= 2){                          // if score is greater then or equal to 2
        dragon_score = temp_score - 2;              // subtracting 2 from the score
    }
    else{
        dragon_score = 0;                           // else set score to 0

    }
```

The expected output is-

- The game should immediately stop
- An option menu should be displayed

The output was -





The game did end up quitting and the options are displayed successfully.