

# **FINAL PROJECT**

# **COSC-111**

**PRESENTED BY : SIRA PEL, RITHPANHA SOK, REIKA ITO**

# TABLE OF CONTENT



OVERVIEW



RESULT/DEMO



OBEJECTIVE



CONCLUSION

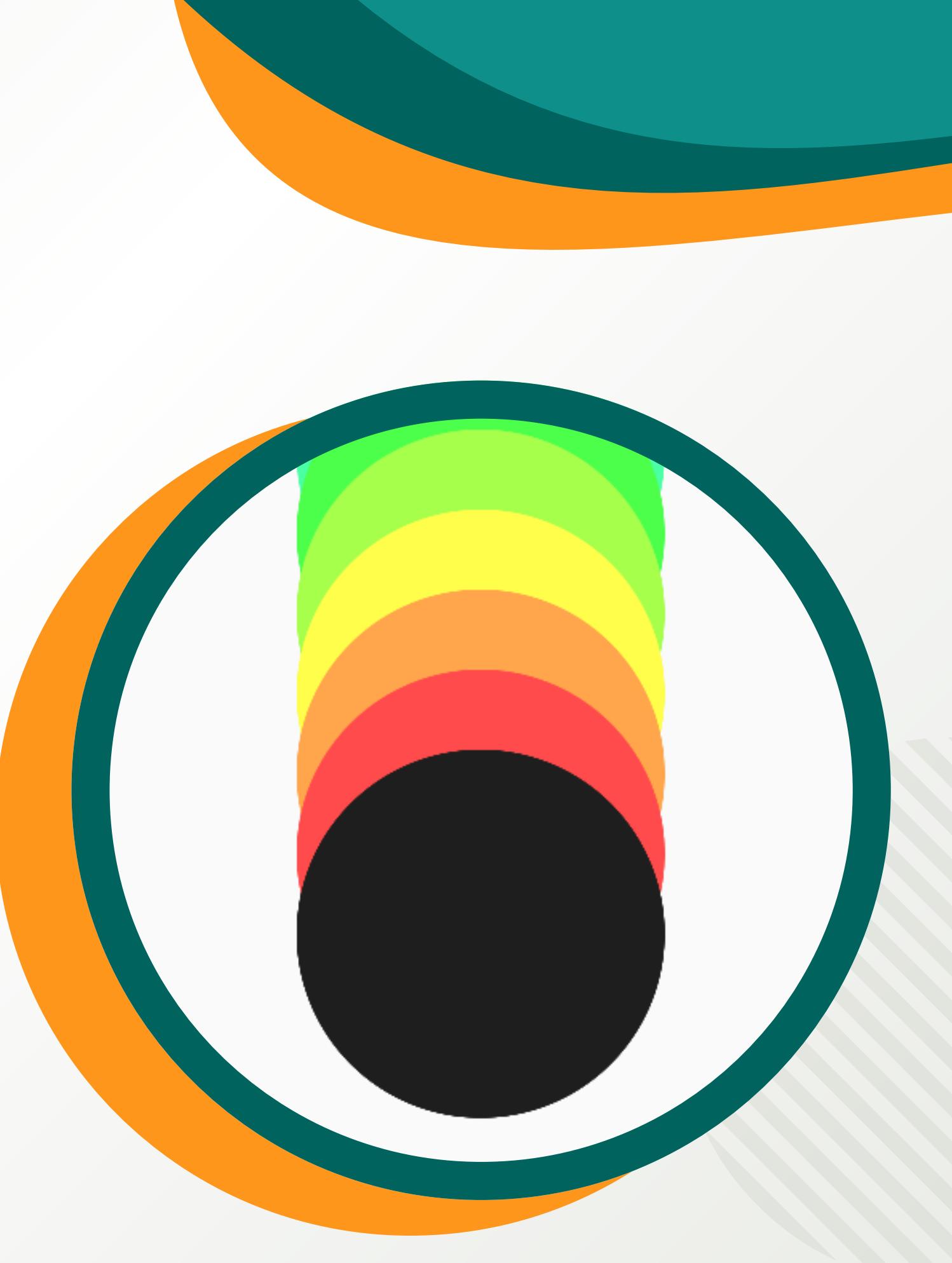


MYTHODOLOGY

# OVERVIEW

TO CREATE THIS GAME WE NEED:

- A window with a basket at the bottom
- Falling "balls" (blue circles)
- Player uses arrow keys to move the basket
- When a ball touches the basket → score increases
- When a ball hits the ground → lose a life
- Three lives available
- If lives = 0 → Game Over



# OBJECTIVE

Through this project, we will learn:

- How to use tkinter for building graphical user interfaces
- How to handle keyboard input and game loops
- How to implement collision detection and real-time game updates
- How to use Pillow to load and display custom background images

# FLOW CHART

[https://aupp-my.sharepoint.com/:b/g/personal/2025303pel\\_aupp\\_edu\\_kh/EfGE1Zh54hBNkza3vu6Q2gsBdnKnVY0Tu-TXhx7mKJdW6g?e=bK6uhB](https://aupp-my.sharepoint.com/:b/g/personal/2025303pel_aupp_edu_kh/EfGE1Zh54hBNkza3vu6Q2gsBdnKnVY0Tu-TXhx7mKJdW6g?e=bK6uhB)

# SETUP AND WINDOW CONFIGURATION

```
● ● ●  
1 import tkinter as tk  
2 import random  
3 from PIL import Image, ImageTk  
4  
5 # Set up main window  
6 root = tk.Tk()  
7 root.title("Catch the Balls")  
8 root.geometry("500x700")  
9 root.resizable(False, False)  
10  
11 canvas = tk.Canvas(root, width=500, height=700, bg="skyblue")  
12 canvas.pack()  
13  
14 bg_img = Image.open("Background.jpg")  
15 bg_img = bg_img.resize((500, 700))  
16 bg_tk = ImageTk.PhotoImage(bg_img)  
17 canvas.create_image(0, 0, anchor="nw", image=bg_tk)
```

# GAME VARIABLES



```
1 # Game variables
2 balls = []
3 score = 0
4 lives = 3
5 game_running = True
6
7 # Player basket
8 basket = canvas.create_rectangle(150, 650, 250, 670, fill="brown")
9
10 # Score display
11 score_text = canvas.create_text(10, 10, anchor="nw", font=("Arial", 16), text=f"Score: {score}")
12 lives_text = canvas.create_text(400, 10, anchor="nw", font=("Arial", 16), text=f"Lives: {lives}")
```

# BASKET MOVEMENT



```
1 # Move basket
2 def move_left(event):
3     canvas.move(basket, -25, 0)
4
5 def move_right(event):
6     canvas.move(basket, 25, 0)
7
8 #Bind key
9 canvas.bind_all("<Left>", move_left)
10 canvas.bind_all("<Right>", move_right)
```

# CREATE BALL



```
1 # Create a falling ball
2 def create_ball():
3     x = random.randint(10, 470)
4     ball = canvas.create_oval(x, 0, x+30, 30, fill="#00FFEE")
5     balls.append(ball)
6     #create ball every 1.5 seconds
7     if game_running:
8         root.after(1500, create_ball)
9 
```

# COLLISION DETECTION



```
1 # Check collision between two objects
2 def check_collision(obj1, obj2):
3     pos1 = canvas.coords(obj1)
4     pos2 = canvas.coords(obj2)
5     overlap = not (pos1[2] < pos2[0] or pos1[0] > pos2[2] or pos1[3] < pos2[1] or
6                      pos1[1] > pos2[3])
7     return overlap
```

# MOVING BALL AND GAME LOGIC

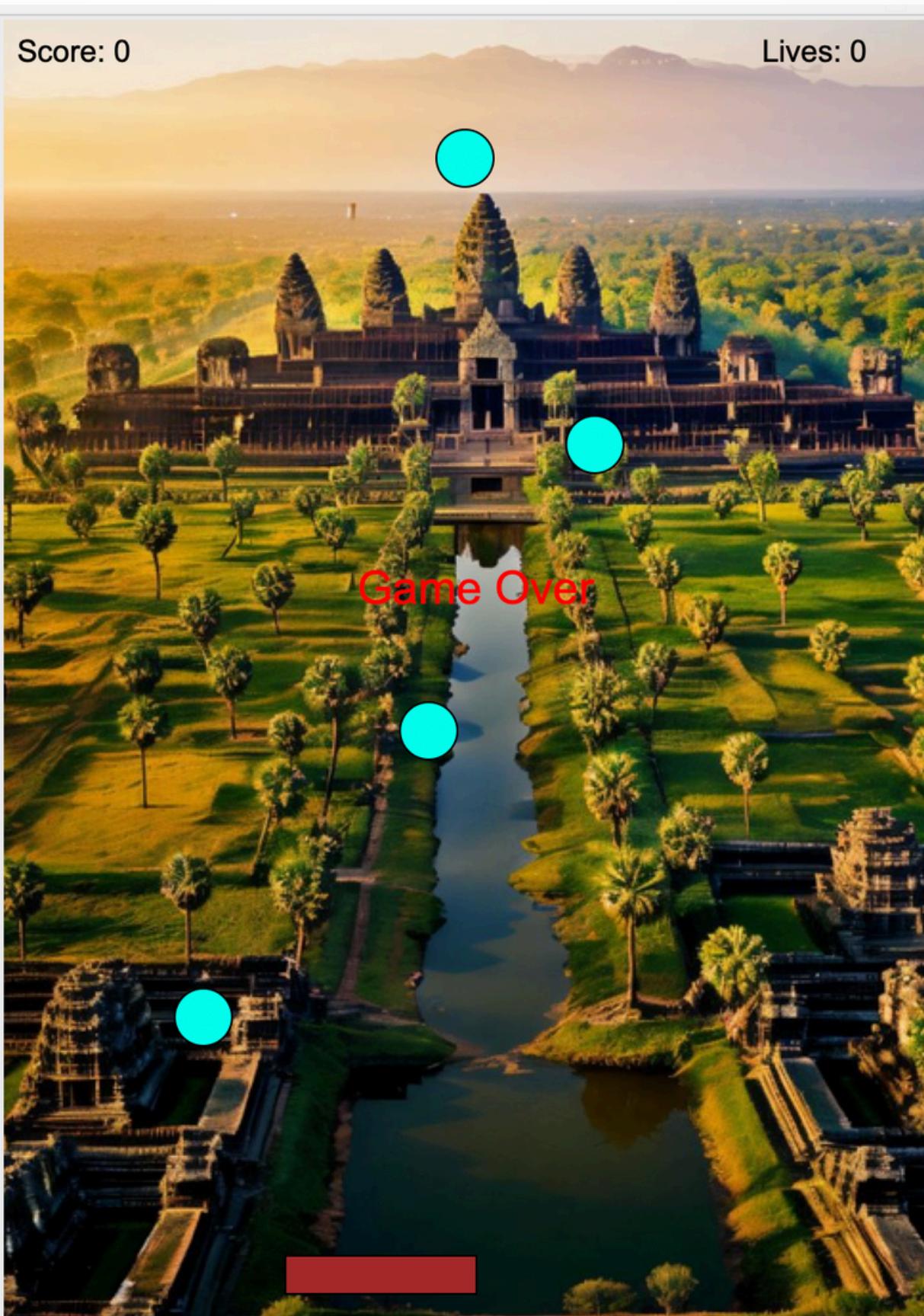
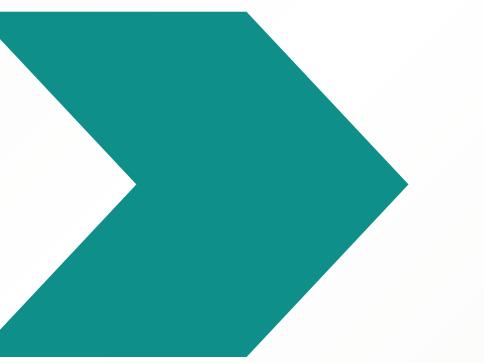
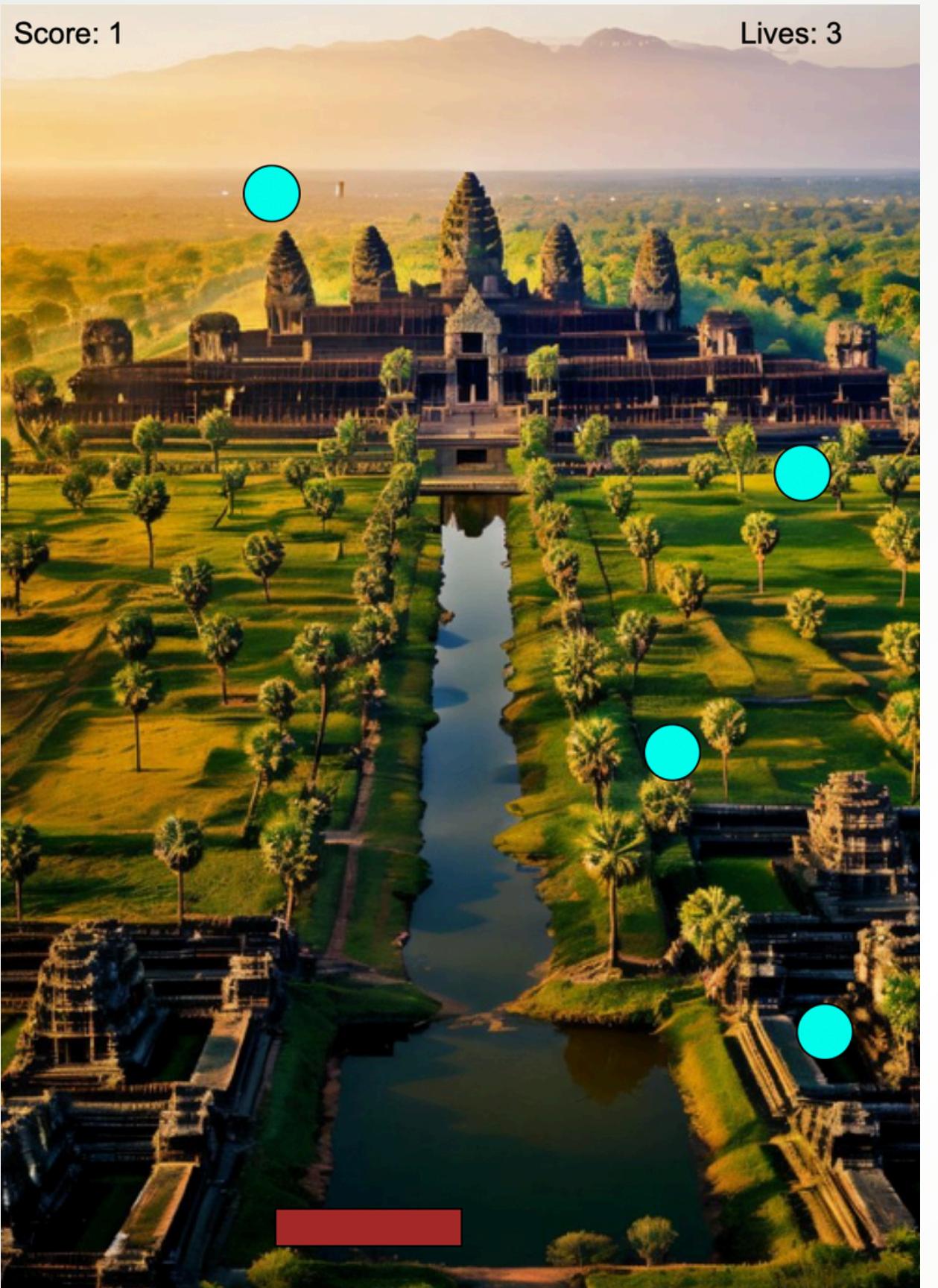


```
1 # Move ball down and check for collision
2 def move_balls():
3     global score, lives, game_running
4
5     for ball in balls[:]:
6         canvas.move(ball, 0, 10)
7         pos = canvas.coords(ball)
8
9         if pos[3] >= 700: # bottom
10            balls.remove(ball)
11            canvas.delete(ball)
12            lives -= 1
13            canvas.itemconfig(lives_text, text=f"Lives: {lives}")
14            if lives == 0:
15                game_running = False
16                canvas.create_text(250, 300, text="Game Over", font=("Arial", 24), fill="red")
17                return
18        elif check_collision(ball, basket):
19            score += 1
20            canvas.itemconfig(score_text, text=f"Score: {score}")
21            balls.remove(ball)
22            canvas.delete(ball)
23    #Move down every 0.1 seconds
24    if game_running:
25        root.after(100, move_balls)
```

# GAME INITIALIZATION

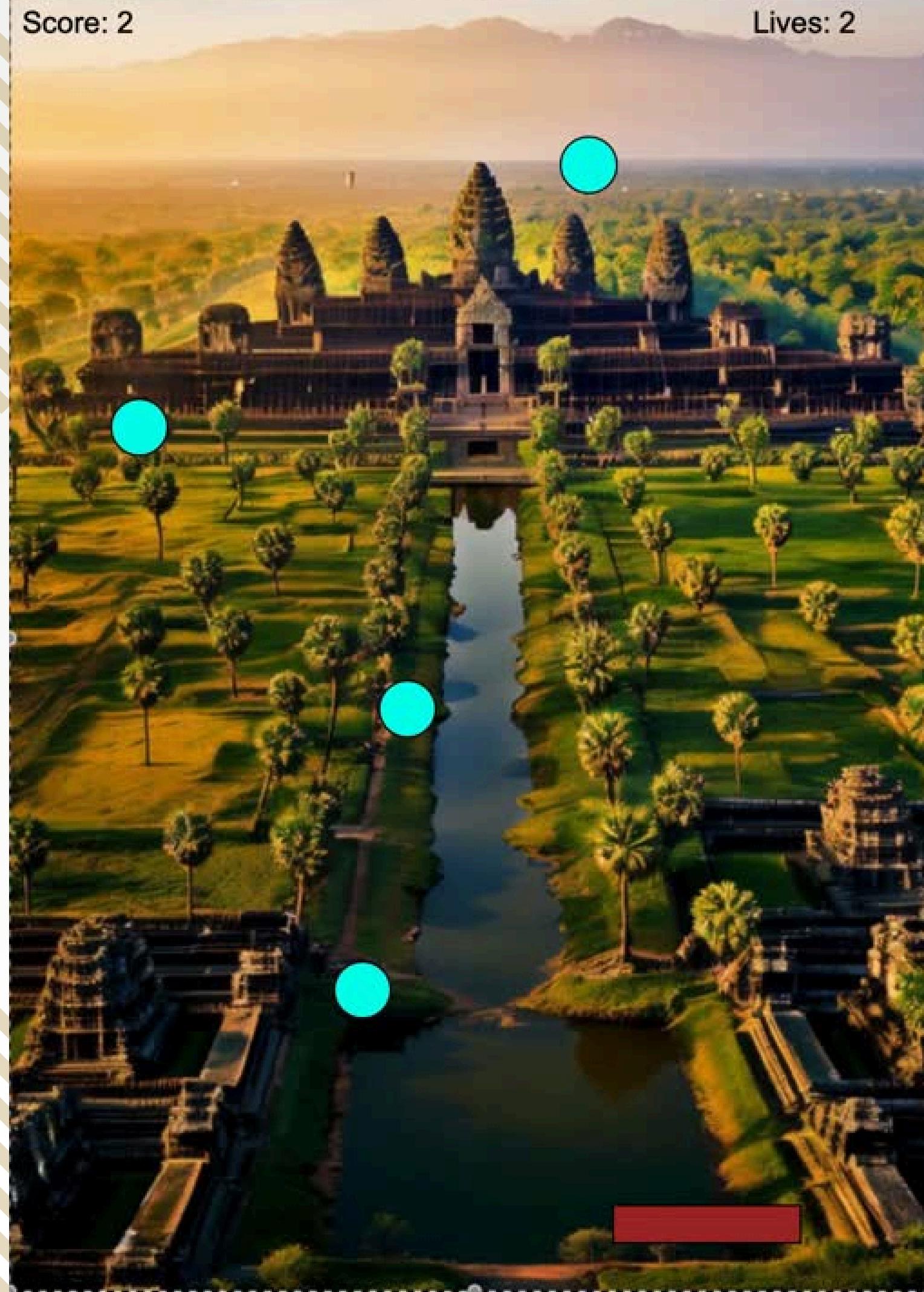
```
1 # Start the game loop  
2 create_ball()  
3 move_balls()  
4  
5 root.mainloop()  
6
```

# DEMO/RESULT



Score: 2

Lives: 2



# CONCLUSION

This project was a simple game created using Python and Tkinter, with image support from Pillow (PIL). The game challenges players to catch falling balls with a basket, tracking their score and remaining lives.

# THANK YOU

