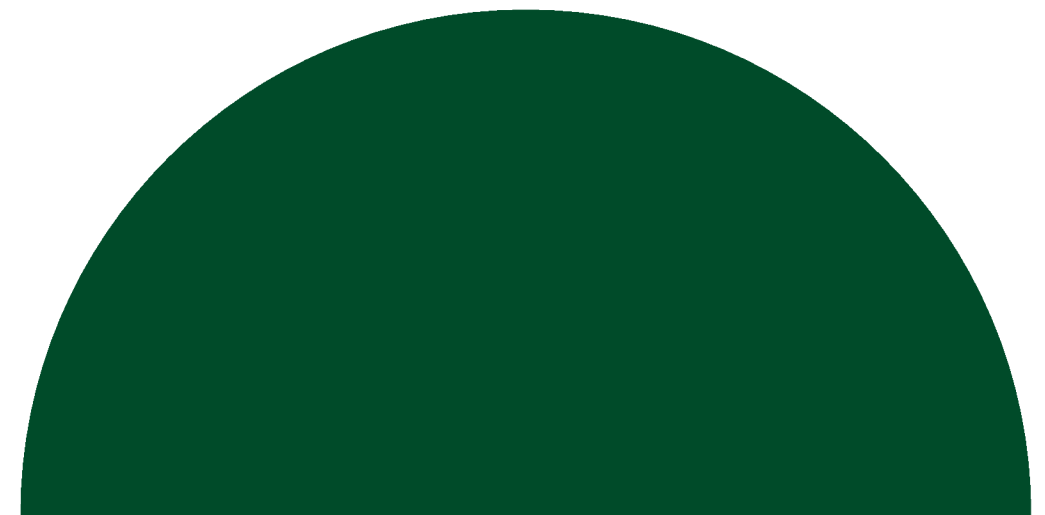
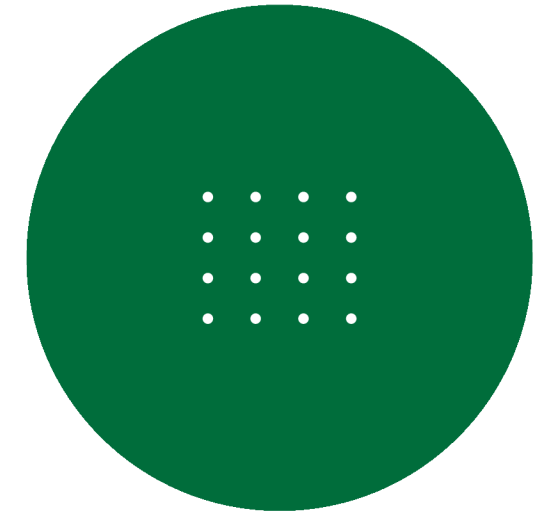


Project Proposal: Ping Pong Game

- Youssef Mohammed Hassan Farid
- Makkah Mohammed Samy Elmesery
- Amr Masoud El-Sayed


Dr/Sara El-metwally





Project Summary

This project is a digital recreation of the classic Ping Pong game using the Python turtle module. The game features two paddles controlled by players and a bouncing ball. The game supports real-time paddle control, scoring, and ball-paddle interaction. It's ideal for beginners to understand real-time game logic, basic collision detection, and event-driven programming in Python.



PEAS Description (Performance, Environment, Actuators, Sensors)

Component

Description

Performance

Players gain points by hitting the ball and preventing it from passing their paddle. Higher score indicates better performance.

Environment

A graphical window (800x600 pixels) containing two paddles, a ball, and a score display

Actuators

The paddle objects move up and down based on keyboard input (W/S for Player 1, Up/Down for Player 2). The ball moves automatically.

Sensors

Game checks for keyboard inputs and collisions between ball, walls, and paddles.

ODESDA Analysis (Objectives, Design, Environment, Sensors, Decision-Making, Actuators)

Objectives

- Build a functional and interactive ping pong game.
- Practice using Python's turtle module for graphics.
- Implement basic game mechanics: movement, collision, scoring.
- Create an engaging two-player experience.

Design

- Use turtle module to render game objects.
- Design includes a center line, paddles, ball, and score display.
- Infinite game loop updates movement and game state in real-time.
- Collision detection for ball interaction with boundaries and paddles.
- Score tracking displayed dynamically at the top of the screen.
- Game loop structure.

ODESDA Analysis (Objectives, Design, Environment, Sensors, Decision-Making, Actuators)

Environment

- 2D graphical interface in a Python Turtle screen.
- 800x600 pixel window size.
- Black background with colored elements for visibility

Sensors

- Keyboard inputs for paddle control.
- Ball position tracking to detect border and paddle collisions

ODESDA Analysis (Objectives, Design, Environment, Sensors, Decision-Making, Actuators)

Decision-Making

- If the ball hits a wall, it bounces back (direction is reversed).
- If the ball hits a paddle, it also bounces back.
- If the ball passes a paddle, the opponent scores a point.
- Ball resets to the center after a score.

Actuators

- Paddle objects move up/down in response to player input.
- Ball object moves autonomously based on its direction vectors.

ODESDA Analysis (Objectives, Design, Environment, Sensors, Decision-Making, Actuators)

Conclusion

This project not only reinforces Python fundamentals but also introduces key concepts of real-time interaction, object control, and simple AI decision-making. It serves as an engaging introduction to game development and graphical programming.



Thank You



+123-456-7890



hello@reallygreatsite.com



www.reallygreatsite.com



123 Anywhere St., Any City

