IBCP A.I GAMES

THE TURTLE RACE

An Exciting Python Turtle Graphics Adventure

PRESENTATION OVERVIEW

- Introduction
- Explanation of code
- Explanation of how game works

IBCP A.I GAMES

WHY WE CHOSE THIS NAME

As the AI students of IBCP, we chose the name IBCP AI GAMES to show how the turtle code that was coded in Python is simply a game created by the brilliant students of Artificial Intelligence.

INTRODUCTION

IBCP A.I GAMES

It's not just about winning bets; it's about experiencing the joy of bringing code to life in a fun and interactive way. Join us as we dive into the world of Python Turtle Graphics, where racing and coding collide for an unforgettable adventure!

SCREENSHOTS

```
from turtle import Turtle, Screen
import random


is_race_on = False
screen = Screen()
screen.setup(width=500, height=400)
user_bet = screen.textinput(title="Make your bet", prompt="Which turtle will win the race? Enter a color: ")
colors = ["green", "black", "gray", "purple","orange","magenta"]
y_positions = [-60,-40,-20,20,40,60]
all_turtles = []


#Create 6 turtles
for turtle_index in range(6):
    new_turtle = Turtle(shape="turtle")
    new_turtle.penup()
    new_turtle.color(colors[turtle_index])
    new_turtle.goto(x=-600, y=y_positions[turtle_index])
    all_turtles.append(new_turtle)

if user_bet:
    is_race_on = True

while is_race_on:
    for racer in all_turtles:

        if racer.xcor() > 550:
            is_race_on = False
            winning_color = racer.pencolor()
            if winning_color == user_bet:
                print(f"You've won! The {winning_color} turtle is the winner!")
            else:
                print(f"You've lost! The {winning_color} turtle is the winner!")

        #Make each turtle move a random amount.
        rand_distance = random.randint(0, 10)
        racer.forward(rand_distance)

screen.exitonclick()
```

PESUDOCODE

1) Import the necessary turtle module and set up the screen for the race.

2) Ask the user to make a bet on the winning turtle's color.

3) Define a list of turtle colors and their starting positions.

4) Create 6 turtles with different colors and starting positions.

5) If the user entered a bet, start the race.

6) While the race is ongoing:
a) Check each turtle's position.
b) If a turtle reaches the finish line (x-coordinate > 550), end the race.
c) Determine the winning turtle's color.
d) If the user's bet matches the winning color, print a ‘victory’ message; otherwise, print a ‘lose’ message.

7) Display the race outcome and winner.

8) Allow the user to click on the screen to exit the program.

CODE

```
from turtle import Turtle, Screen
import random
```

- *'from turtle import Turtle, Screen':* Turtle and Screen class is imported from the turtle module as turtle is used to create the graphics using turtle and screen is used to the blank window where turtles will move.
- *'import random':* Importing random module helps us to work with random numbers and their operations.

```
is_race_on = False
screen = Screen()
screen.setup(width=500, height=400)
```

- *'is_race_on = False':* The variable specifies that the race during the initial program is not active.
- *'screen = Screen()':* Creates a turtle screen object.
- *'screen.setup(width=500, height=400)':* Sets up the dimensions of the turtle screen with a width of 500 pixels and a height of 400 pixels.

```
user_bet = screen.textinput(title="Make your bet", prompt="Which turtle
```

'user_bet = screen.textinput(...)': Displays a pop-up window to take user input for the color of the turtle they are betting on.

```
colors = ["green", "black", "gray", "purple","orange","magenta"]
y_positions = [-80,-50,-20,20,50,80]
all_turtles = []
```

- *'colors':* A list of colors that will be assigned to the turtles.
- *'y_positions':* A list of y-coordinates where the turtles will be initially placed on the screen.
- *'all_turtles':* An empty list to store turtle objects

CODE

```
#Create 6 turtles
for turtle_index in range(6):
    new_turtle = Turtle(shape="turtle")
    new_turtle.penup()
    new_turtle.color(colors[turtle_index])
    new_turtle.goto(x=-600, y=y_positions[turtle_index])
    all_turtles.append(new_turtle)
```

The loop creates six turtle objects with different colors, lifts the pen (so they don't draw while moving), sets their color, and positions them at the starting line on the screen.

If the user has entered a bet, *'is_race_on'* is set to True, indicating that the race is now active

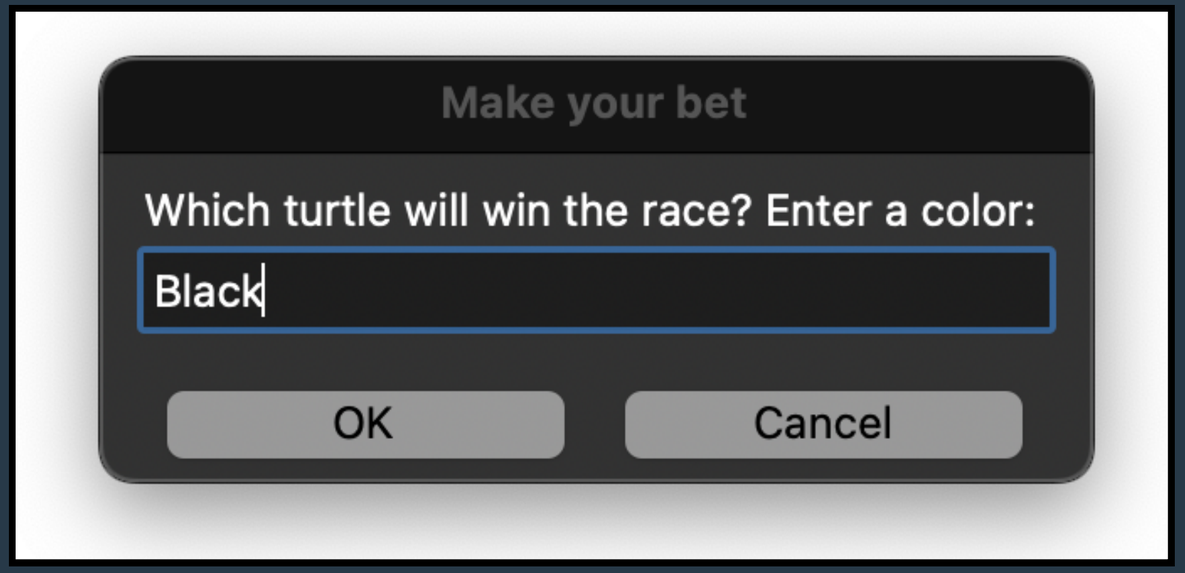
```
if user_bet:
    is_race_on = True
```

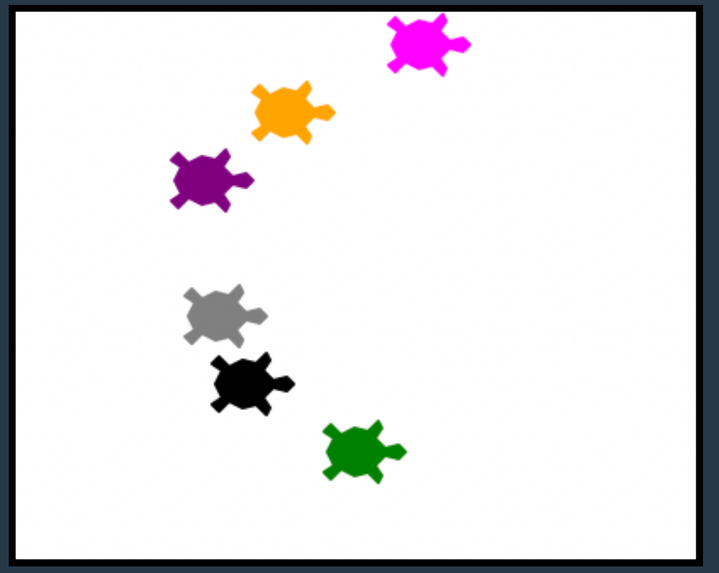
```
le is_race_on:
 for racer in all_turtles:
     if racer.xcor() > 550:
         is_race_on = False
         winning_color = racer.pencolor()
         if winning_color == user_bet:
             print(f"You've won! The {winning_color} turtle is
         else:
             print(f"You've lost! The {winning_color} turtle is

     # Make each turtle move a random amount.
     rand_distance = random.randint(0, 10)
     racer.forward(rand_distance)

een.exitonclick()
```

CODE

- The while loop continues executing as long as the condition 'is_race_on' is True.
- Implies that the turtle race simulation is ongoing.
- A nested 'for' loop iterates through each turtle in the 'all_turtles' list.
- Indicates that each turtle will perform certain actions within the loop.
- The 'if' statement checks if the x-coordinate of the current turtle 'racer' is greater than 550.
- If met, implies the turtle has crossed a finish line or reached a specific point in the race.
- If a turtle crosses the finish line, sets 'is_race_on' to 'False', indicating the race is over.
- Determines the color of the winning turtle and prints a message based on the user's bet 'user_bet'.
- These lines generate a random distance and move each turtle 'racer' forward by that distance.
- Simulates the turtles' movement in the race.
- This line exits the turtle graphics window when the user clicks on it.
- Serves as a way to end the program gracefully.

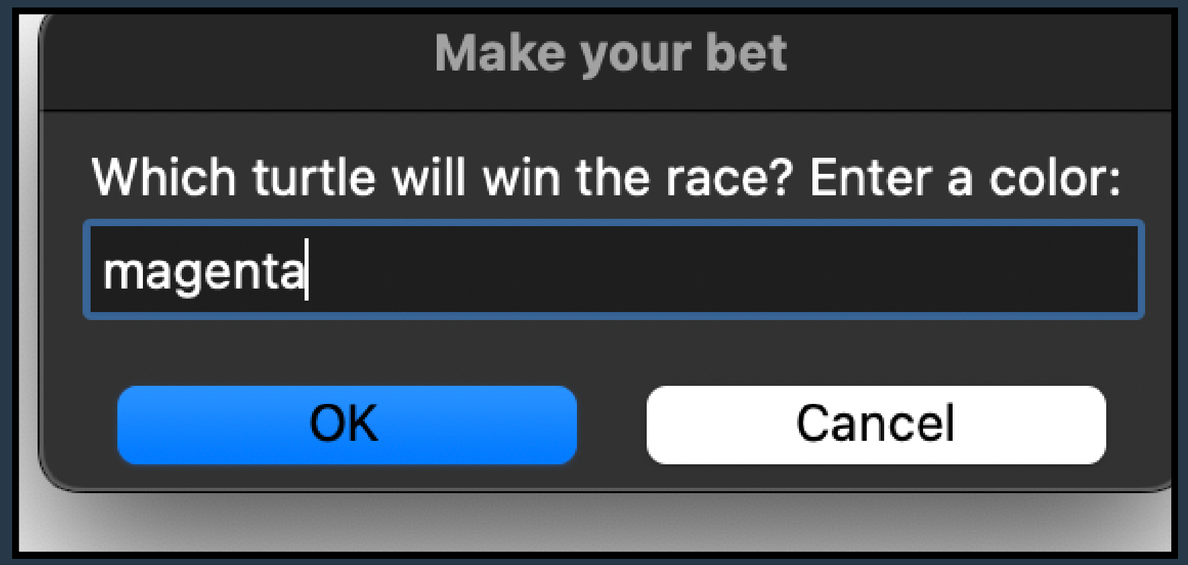
SCREENSHOTS

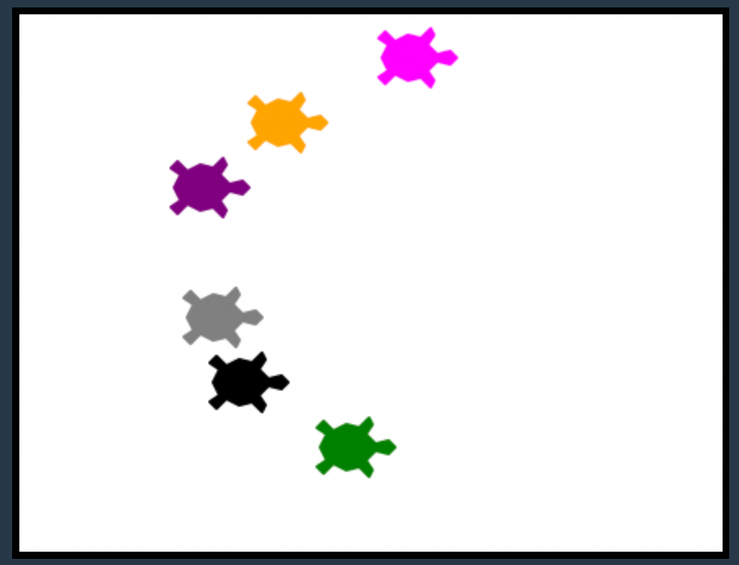




```
==========================================================
You've lost! The magenta turtle is the winner!
```

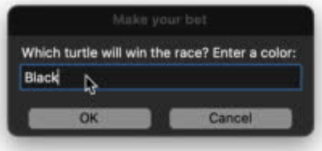
SCREENSHOTS





```
You've lost! The gray turtle is the winner!
```

DEMO


Make your bet
Which turtle will win the race? Enter a color:
Black
OK
Cancel

SUMMARY

THIS PYTHON CODE CREATES A SIMPLE TURTLE RACING GAME WHERE USERS BET ON A TURTLE TO WIN. IT USES TURTLE GRAPHICS FOR VISUALS, ACCEPTS USER INPUT FOR BETTING, AND EMPLOYS RANDOMIZATION TO MAKE THE RACE UNPREDICTABLE. THE CODE COVERS FUNDAMENTAL CONCEPTS LIKE CONDITIONALS FOR DETERMINING THE WINNER, LOOPS FOR CONTINUOUS GAMEPLAY, AND BASIC TURTLE MODULE USAGE. THIS PROJECT SERVES AS A HANDS-ON INTRODUCTION TO PYTHON PROGRAMMING, EMPHASIZING GRAPHICS, USER INTERACTION, AND GAME LOGIC IN A STRAIGHTFORWARD MANNER.

Q & A

THANK YOU!

FOR LISTENING

Presented by Akshat, Navya,
Presha and Azmah