Python: Standard Library

Sunglok Choi, Assistant Professor, Ph.D.
Computer Science and Engineering Department, SeoulTech
sunglok@seoultech.ac.kr | https://mint-lab.github.io/

Overview

- **Prerequisite**
  - Anacodna (Individual Edition)

- **Practice: *Turtle Runaway***
  - The given skeleton code
  - Requirements
  - Practice with the skeleton code
    - Step #1) Add the timer
    - Step #2) Add a more intelligent turtle

- **Assignment**
  - Mission: Complete the game, *Turtle Runaway*

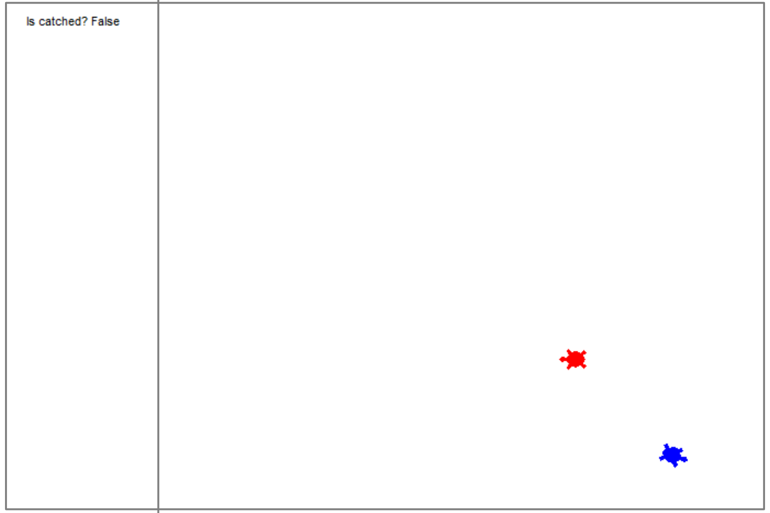
OSS Game Company

Image: ACON

Practice: *Turtle Runaway*

- The given skeleton code (file: `turtle_runaway_skeleton.py`; 1/4)

```python
# This example is not working in Spyder directly (F5 or Run)
# Please type '!python turtle_runaway.py' on IPython console in your Spyder.
import turtle, random

class RunawayGame:
    def __init__(self, canvas, runner, chaser, catch_radius=50, init_dist=400):
        self.canvas = canvas
        self.runner = runner
        self.chaser = chaser
        self.catch_radius2 = catch_radius**2

        # Initialize 'runner' and 'chaser'
        self.runner.shape('turtle')
        self.runner.color('blue')
        self.runner.penup()
        self.runner.setx(-init_dist / 2)

        self.chaser.shape('turtle')
        self.chaser.color('red')
        self.chaser.penup()
        self.chaser.setx(+init_dist / 2)
        self.chaser.setheading(180)

        # Instantiate an another turtle for drawing
        self.drawer = turtle.RawTurtle(canvas)
        self.drawer.hideturtle()
        self.drawer.penup()
```

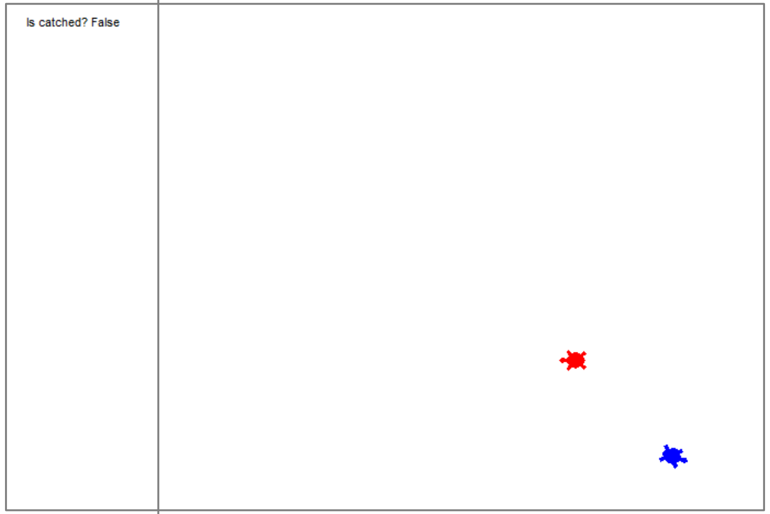


```
Is catched? False
```

Practice: *Turtle Runaway*

- The given skeleton code (file: turtle_runaway_skeleton.py; 2/4)

```
class RunawayGame:
    def __init__(self, canvas, runner, chaser, catch_radius=50, init_dist=400):
        # ...

    def is_catch(self):
        p = self.runner.pos()
        q = self.chaser.pos()
        dx, dy = p[0] - q[0], p[1] - q[1]
        return dx**2 + dy**2 < self.catch_radius2

    def start(self, ai_timer_msec=100):
        self.ai_timer_msec = ai_timer_msec
        self.canvas.ontimer(self.step, self.ai_timer_msec)

    def step(self):
        self.runner.run_ai(self.chaser)
        self.chaser.run_ai(self.runner)

        # TODO: You can do something here.
        is_catched = self.is_catch()
        self.drawer.undo()
        self.drawer.penup()
        self.drawer.setpos(-300, 300)
        self.drawer.write(f'Is catched? {is_catched}')

        self.canvas.ontimer(self.step, self.ai_timer_msec)
```

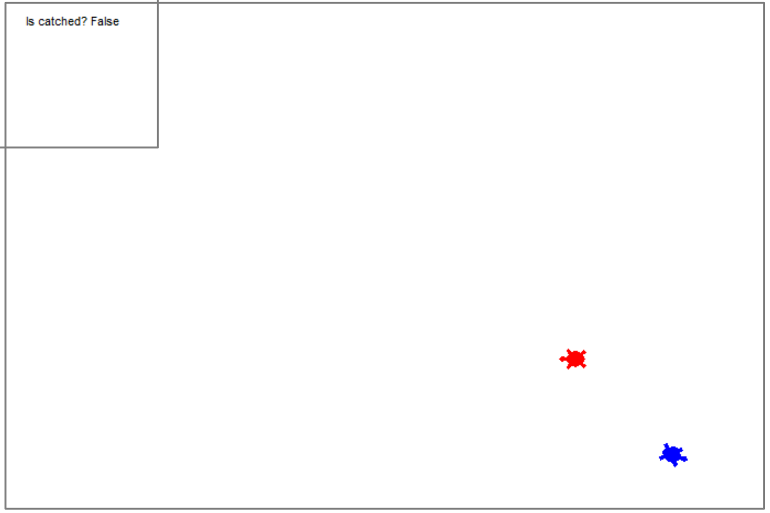


```
Is catched? False
```

Practice: *Turtle Runaway*

- The given skeleton code (file: `turtle_runaway_skeleton.py`; 3/4)

```python
class ManualMover(turtle.RawTurtle):
    def __init__(self, canvas, step_move=10, step_turn=10):
        super().__init__(canvas)
        self.step_move = step_move
        self.step_turn = step_turn

        # Register event handlers
        canvas.onkeypress(lambda: self.forward(self.step_move), 'Up')
        canvas.onkeypress(lambda: self.backward(self.step_move), 'Down')
        canvas.onkeypress(lambda: self.left(self.step_turn), 'Left')
        canvas.onkeypress(lambda: self.right(self.step_turn), 'Right')
        canvas.listen()

    def run_ai(self, opponent):
        pass
```

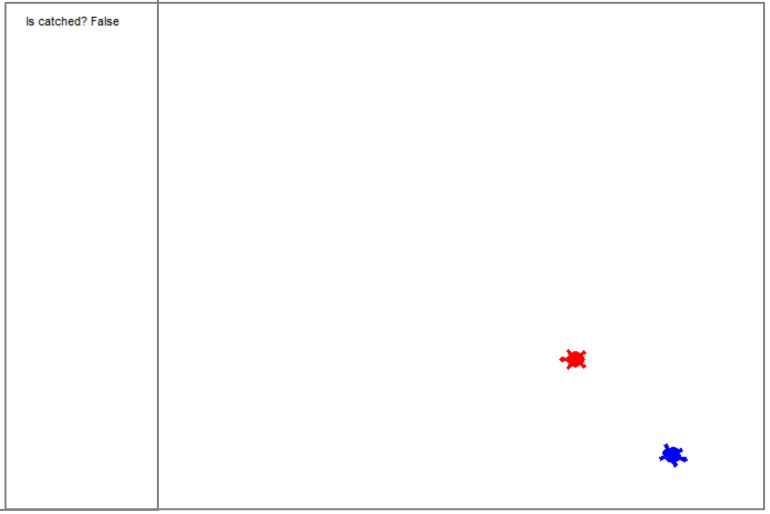


Is catched? False

Practice: *Turtle Runaway*

- The given skeleton code (file: turtle_runaway_skeleton.py; 4/4)

```
class RandomMover(turtle.RawTurtle):
    def __init__(self, canvas, step_move=10, step_turn=10):
        super().__init__(canvas)
        self.step_move = step_move
        self.step_turn = step_turn

    def run_ai(self, oppoenent):
        mode = random.randint(0, 2)
        if mode == 0:
            self.forward(self.step_move)
        elif mode == 1:
            self.left(self.step_turn)
        elif mode == 2:
            self.right(self.step_turn)

if __name__ == '__main__':
    canvas = turtle.Screen()
    runner = RandomMover(canvas)
    chaser = ManualMover(canvas)

    game = RunawayGame(canvas, runner, chaser)
    game.start()
    canvas.mainloop()
```



Is catched? False

Practice: *Turtle Runaway*

- Requirements
  - **Mandatory (base point: 5 points)**
    - **Add a timer (5 points)**: You can freely choose an up/down timer for your purpose.
    - **Add your ~~intelligent~~ Turtle (5 points)**: You can assign a role, *runner* or *chaser* or both.
    - **Add a concept of score (5 points)**: You can define the score by yourself.
  - Optional
    - Change the window title to *Turtle Runaway*
    - Add a concept of stages
    - Add opening, closing, and ending
    - Fix a bug (e.g. switching colors)
    - Anyway, you can do whatever you want if the game is more fun.

- Practice with the given skeleton code
  - Step #1) Add the timer
  - Step #2) Add a more intelligent turtle

Assignment

- Mission
  - Complete the given skeleton code (`turtle_runaway_skeleton.py`)
  - Submit your code (`turtle_runaway.py`) and its explanation (`turtle_runaway.md`) with a screenshot (turtle_runaway.png)

- Condition
  - Please follow the above filename convention.
  - You can start from scratch (without using the given skeleton code).
  - You can freely change the given skeleton code if necessary.

- Submission
  - Deadline: **October 13, 2021 23:59** (firm deadline; no extension)
  - Where: e-Class > Assignments
  - Score: Max 20 points