Note) Slides and example codes are available:
https://github.com/mint-lab/prog_meets_math

Python Lab #3: *Turtle Runaway*

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 a timer
    - Step #2) Add your more intelligent turtle
    - Step #3) Add your concept of score

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

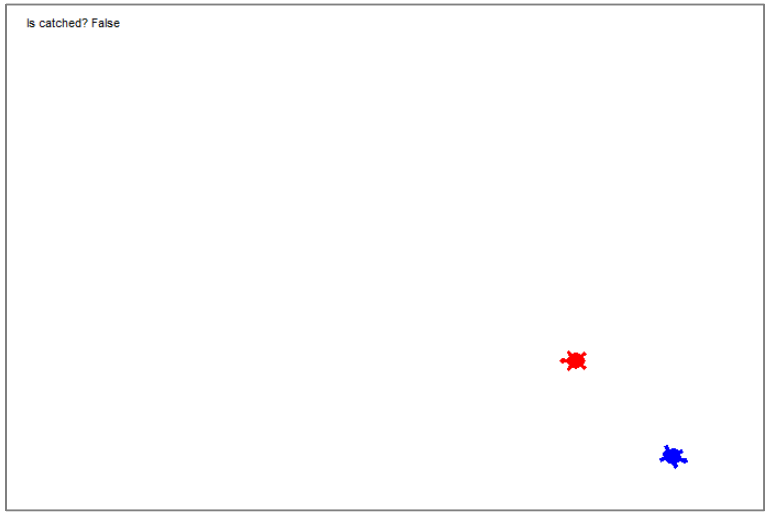
OSS Game Company

Image: ACON

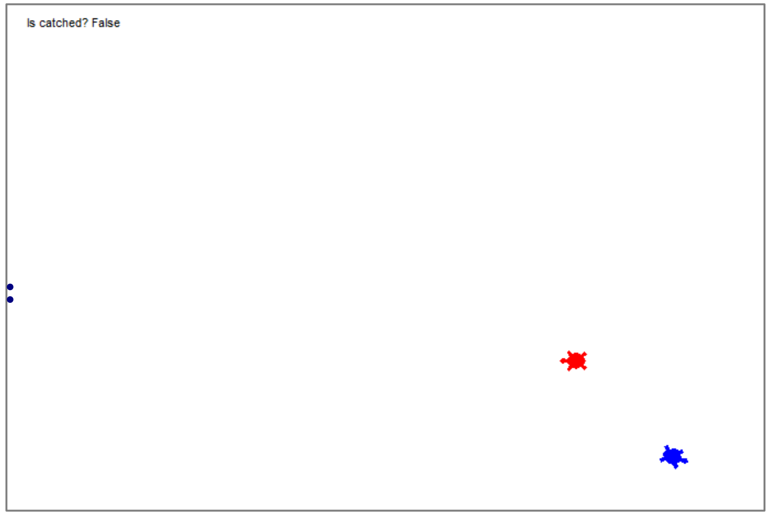
Practice) *Turtle Runaway*

- 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):
        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.chaser.shape('turtle')
        self.chaser.color('red')
        self.chaser.penup()

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

    def is_catched(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
```


Is catched? False

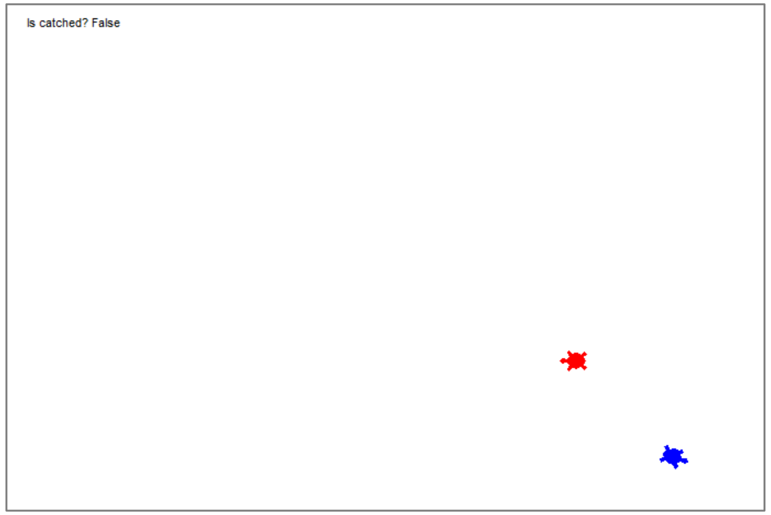
Practice) *Turtle Runaway*



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

```python
class RunawayGame:
    def __init__(self, canvas, runner, chaser, catch_radius=50, init_dist=400):
        # ...
    def is_catched(self):
        # ...
    def start(self, init_dist=400, ai_timer_msec=100):
        self.runner.setpos((-init_dist / 2, 0))
        self.runner.setheading(0)
        self.chaser.setpos((+init_dist / 2, 0))
        self.chaser.setheading(180)

        # TODO) You can do something here and follows.
        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.pos(), self.chaser.heading())
        self.chaser.run_ai(self.runner.pos(), self.chaser.heading())

        # TODO) You can do something here and follows.
        is_catched = self.is_catched()
        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)
```

Practice) *Turtle Runaway*


Is catched? False

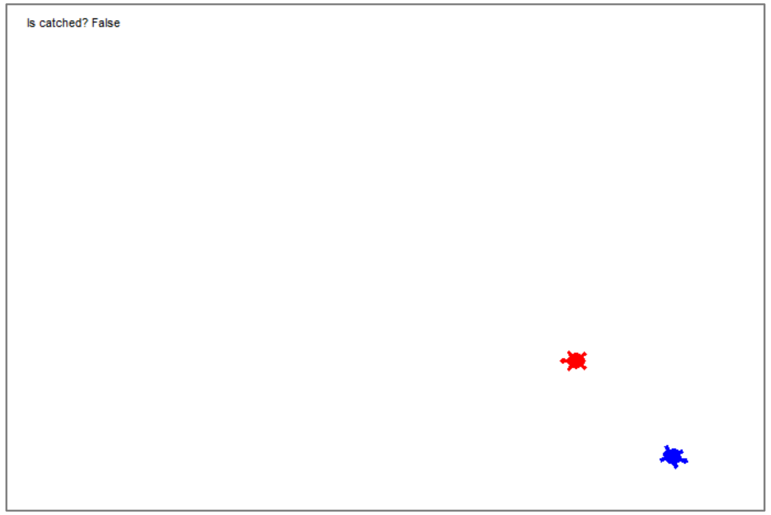
- The given skeleton code (file: `turtle_runaway_skeleton.py`; 3/4)

```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, opp_pos, opp_heading):
        pass
```

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, opp_pos, opp_heading):
        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__':
    # Use 'TurtleScreen' instead of 'Screen' to prevent an exception from the singleton 'Screen'
    # ...
    # TODO) You can do something here and follows.
    runner = RandomMover(screen)
    chaser = ManualMover(screen)

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

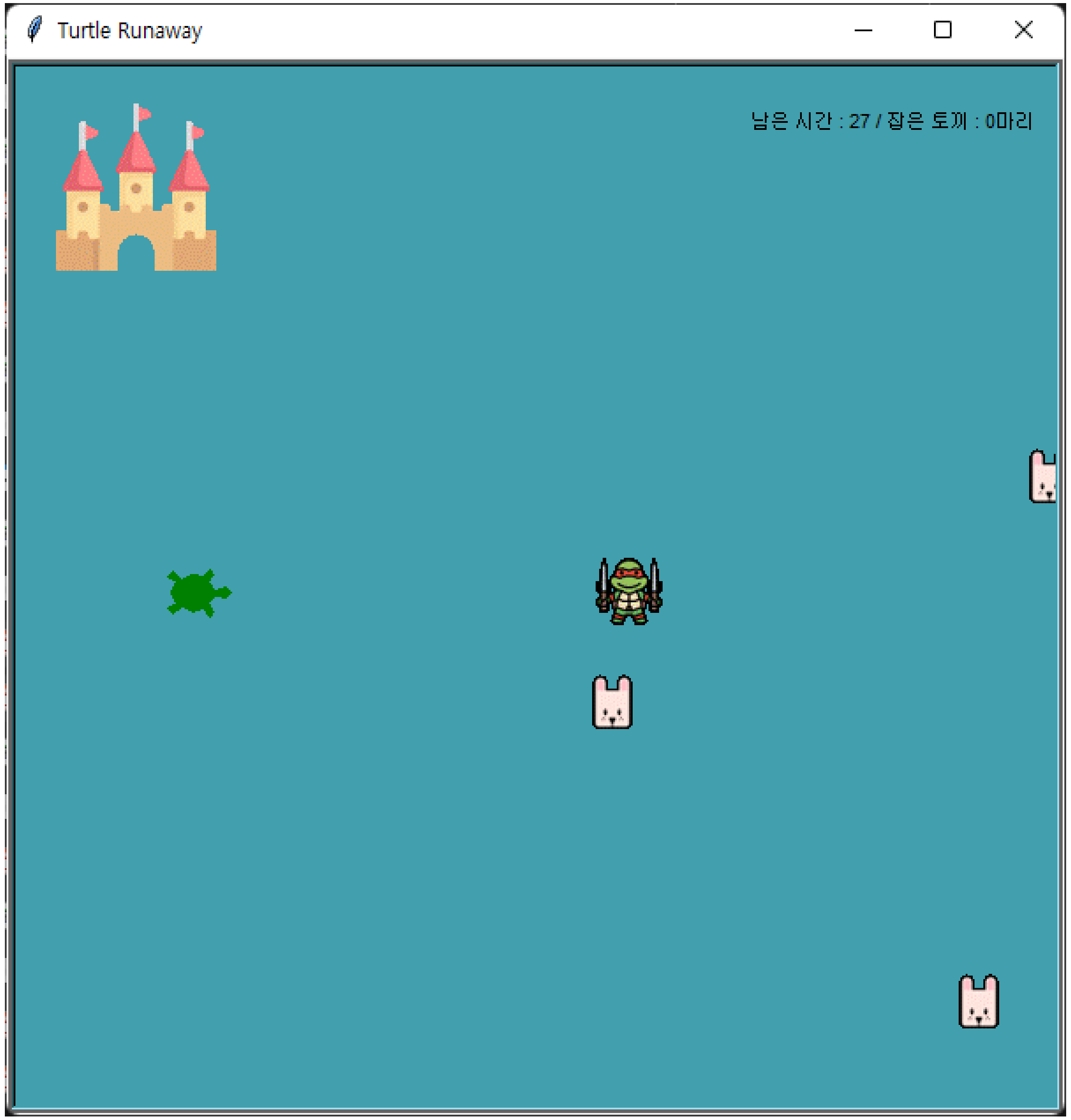

Is catched? False

Practice) *Turtle Runaway*

- Requirements
  - **Mandatory**
    - **Add a timer (5 points)**: You can freely choose an up/down timer for your purpose.
    - **Add your ~~intelligent~~ Turtle (8 points)**: You can assign a role, *runner* or *chaser* or both.
    - **Add your concept of score (7 points)**: You can define the score by yourself.
  - Optional
    - Change the window title to *Turtle Runaway*
    - Add background or game arena
    - Add a concept of stages
    - Add opening, closing, and ending
    - Fix a bug (e.g. switching colors)
    - Anyway, you can do whatever you want to make the game fun.

2021 Best Work (by 정의진)

```python
import turtle, random, time

class RunawayGame:
    def __init__(self, ...):
        # ...
        self.chaser.shape("bad_turtle.gif")
        # ...

if __name__ == "__main__":
    screen = turtle.Screen()
    screen.setup(600, 600)
    screen.title("Turtle Runaway")
    screen.bgcolor("#429FAD")
    screen.addshape("rabbit.gif")
    screen.addshape("castle.gif")
    screen.addshape("bad_turtle.gif")
    runner = ManualMover(screen)
    chaser = ChaseMover(screen)

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



남은 시간 : 27 / 잡은 토끼 : 0마리

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, 2023 23:59** (firm deadline; no extension)
  - Where: e-Class > Assignments
  - Score: Max 20 points