Spike: Spike Week 13
Title: Tactical Steering (Hide!)
Author: Nguyen Khanh Toan - 104180605

Goals / deliverables:

Develop a hunter-prey agent simulation involving two or more agents, where "prey" agents evade "hunter" agents by utilizing environmental objects. The simulation should:
• Feature multiple "objects" that prey agents can utilize for hiding (represented as basic circles).
• Clearly distinguish between the appearance and capabilities of "hunter" and "prey" agents.
• Display indicators (such as "x" symbols) to highlight suitable hiding spots for prey agents to choose from.
• Require prey agents to strategically assess and select optimal hiding locations to move towards.
• Avoid having prey agents hide "inside" objects; instead, they should seek hiding spots located outside or behind objects.

Technologies, Tools, and Resources used:

- Visual Studio Code
- Python 3.12

Tasks undertaken:

- Install Python 3+
- Install Pyglet 2+
- Install and setup compatible IDE for the language, e.g.: Visual Studio Code
- Pay attention to the comment of how the code work and functionality. Can use debug tool to observe the program more clearly.
- Run the code and observing the output.

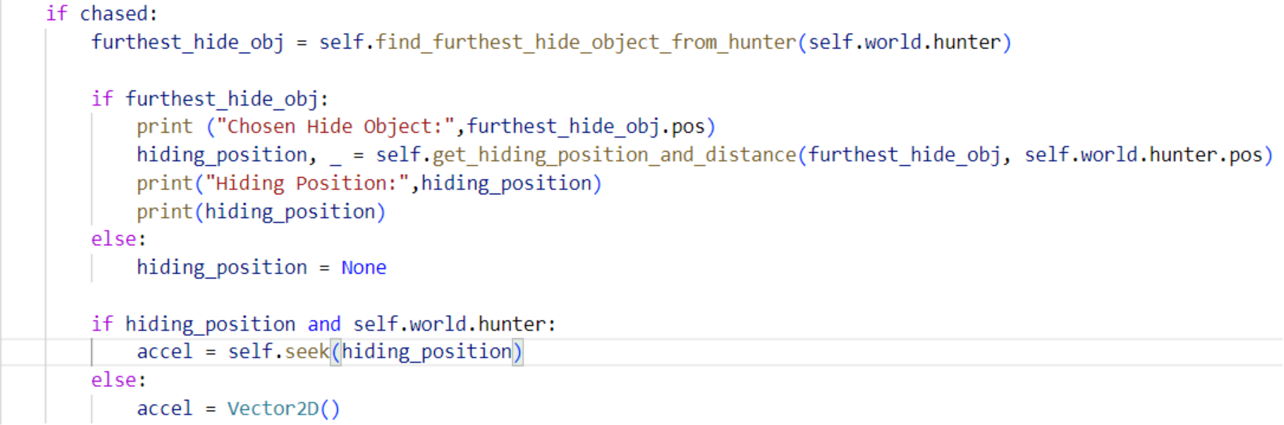
Planning Notes:

- Extend the code to support adding Hunter agents **(Press H)** and new keyboard input to add it to current world.
- Hunter can be an object of Agent class, but it needs to separate to indicate which is “prey” or “hunter”?
- Modify the flee, seek, calculation,… function which indicate behaviours of agent when there is having hunter in range.
- Create object for hiding spot

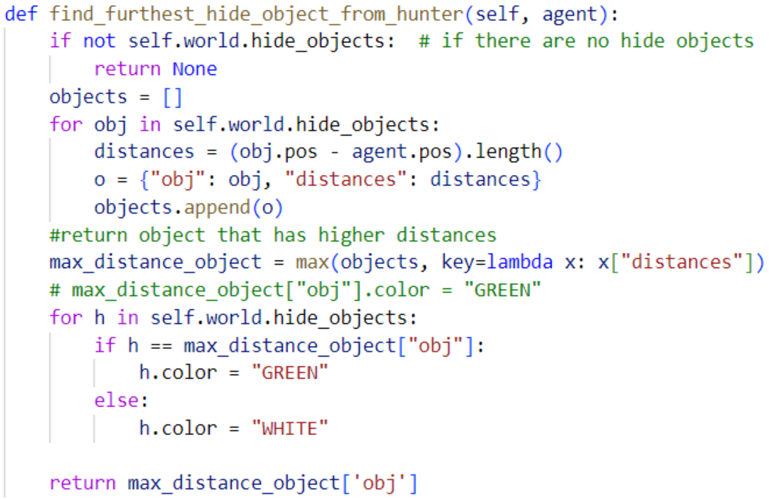
What we found out:

- When “Prey” not recognize the “Hunter” in the panic range, it will seek to the target. And when the “Hunter” is recognized, agent will automatically flee out of its panic range to the hide spot (behind the hiding object in perspective of the hunter). Then try to seek for the target.

- The Hunter will have constant acceleration until it recognize a “Prey” in range and will use the seek function to get to the “Prey”. Its mass is also modified so that it is harder to turn. The shape of the hunter also bigger than agent and color differently from Agent (Red)
- When being chased, the agent will always try to seek for hiding spot in that further from the hunter.

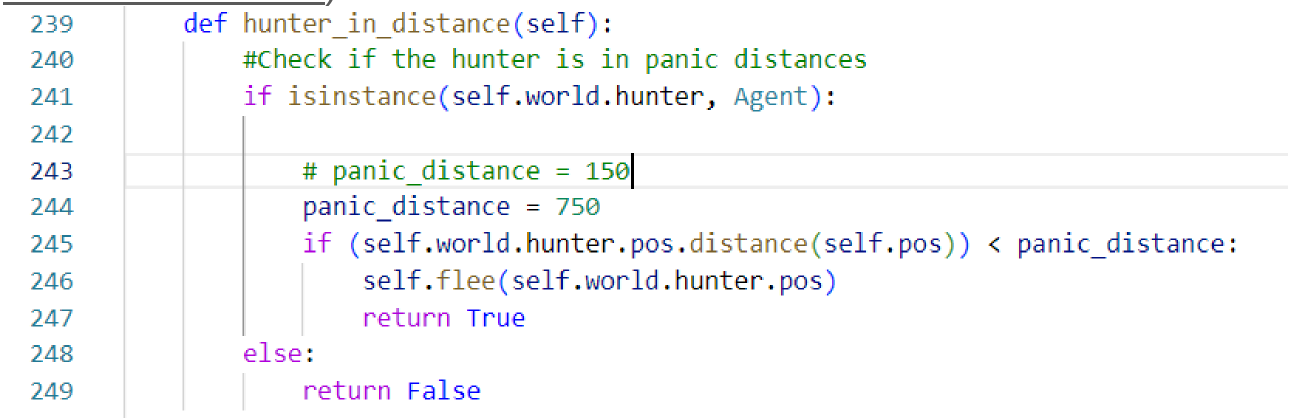


```
if chased:
    furthest_hide_obj = self.find_furthest_hide_object_from_hunter(self.world.hunter)

    if furthest_hide_obj:
        print ("Chosen Hide Object:",furthest_hide_obj.pos)
        hiding_position, _ = self.get_hiding_position_and_distance(furthest_hide_obj, self.world.hunter.pos)
        print("Hiding Position:",hiding_position)
        print(hiding_position)
    else:
        hiding_position = None

    if hiding_position and self.world.hunter:
        accel = self.seek(hiding_position)
    else:
        accel = Vector2D()
```



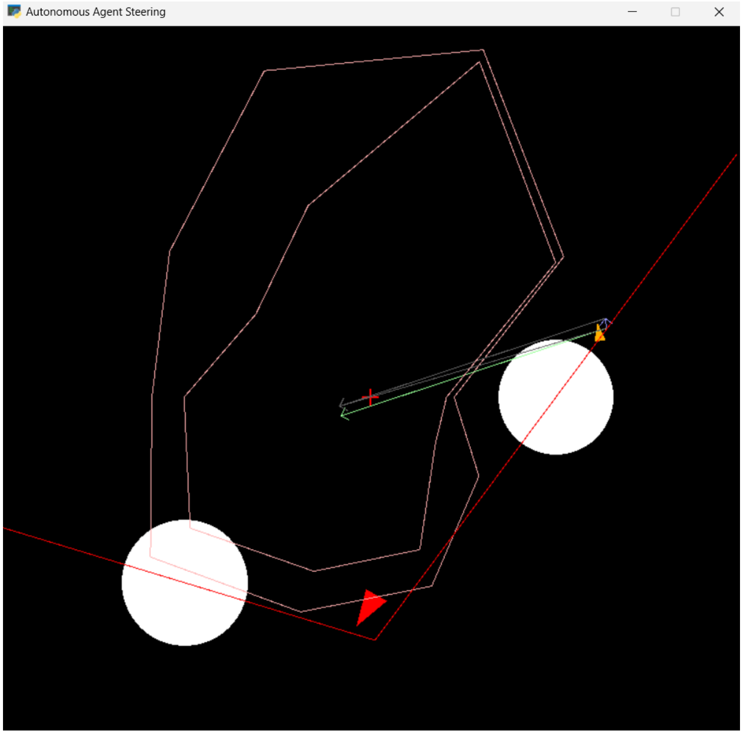
```
def find_furthest_hide_object_from_hunter(self, agent):
    if not self.world.hide_objects:  # if there are no hide objects
        return None
    objects = []
    for obj in self.world.hide_objects:
        distances = (obj.pos - agent.pos).length()
        o = {"obj": obj, "distances": distances}
        objects.append(o)
    #return object that has higher distances
    max_distance_object = max(objects, key=lambda x: x["distances"])
    # max_distance_object["obj"].color = "GREEN"
    for h in self.world.hide_objects:
        if h == max_distance_object["obj"]:
            h.color = "GREEN"
        else:
            h.color = "WHITE"

    return max_distance_object['obj']
```

- I have changed the panic distance to 750 of the agents so it is easier to see the agent hiding from the hunter (Try to make the hunter chase agent first, it is easier to see hiding behaviour of agent and the pursuit behaviour of hunter)

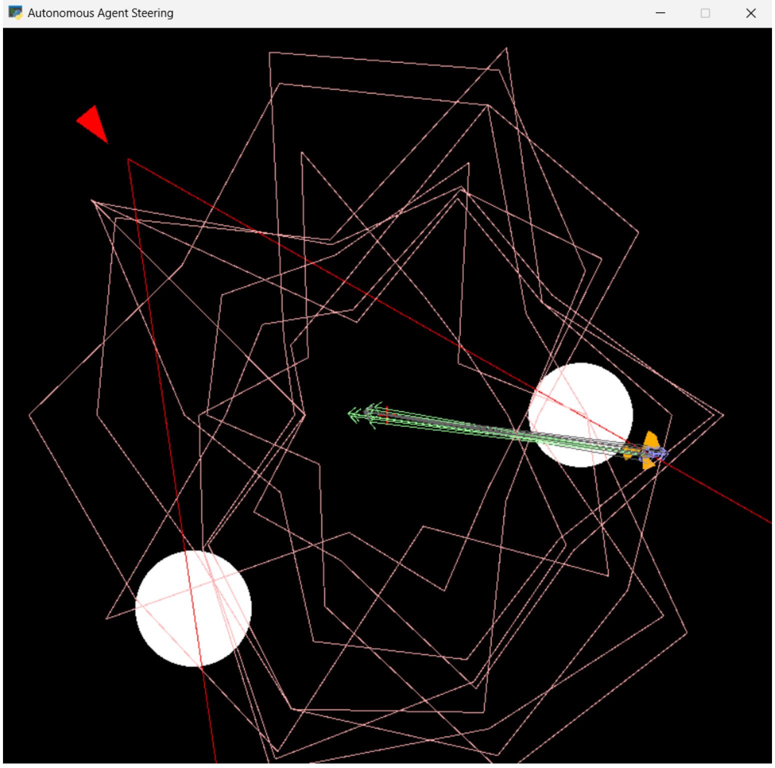


```
    def hunter_in_distance(self):
        #Check if the hunter is in panic distances
        if isinstance(self.world.hunter, Agent):

            # panic_distance = 150
            panic_distance = 750
            if (self.world.hunter.pos.distance(self.pos)) < panic_distance:
                self.flee(self.world.hunter.pos)
                return True
        else:
            return False
```

- Otherwise, they are chasing like Tom and Jerry until agent get out of the recognition of the hunter.



1 agent



Multiple agents