Spike: Spike_3
Title: Spike_Emergent Group Behaviour

Author: Le Bao Duy Nguyen 102449993

Goals / deliverables:

- Short report titled “Spike_Emergent Group Behaviour”
- Modify agent to include cohesion, separation, alignment with alterable value adjustment
- Include basic wandering and weighted sum of all steering behaviours

Technologies, Tools, and Resources used:

- Visual Studio Code
- Python 3.0+

Tasks undertaken:

- Download and install Visual Studio Code
- Download and install Python 3.0 & above
- Download and install Python extension within Visual Studio Code
- Read the codes and guidelines. Researching Canvas materials as well as Google, YouTube, etc.

  Bot functions for Cohesion, Separation, Alignment. TagNeighbours to check if the agent has any close by agent to steer with.

```
def tagNeighbours(self,bots,radius) :
    self.neighbours.clear();

    for bot in bots :

        if bot != self :
            bot.tagged = False
            to = self.pos - bot.pos
            gap = radius + bot.bRadius
            if to.length_sq() < gap**2 :
                bot.tagged = True
                self.neighbours.append(bot)

def Seperation(self,group):
    SteeringForce = Vector2D()
    for bot in group:
        if bot != self and bot.tagged:
            ToBot = self.pos - bot.pos
```

```
            SteeringForce += ToBot.normalise() / ToBot.length()
        return SteeringForce

    def Alignment(self,group):
        AvgHeading = Vector2D()
        AvgCount = 0

        for bot in group:
            if bot != self and bot.tagged:
                AvgHeading += bot.heading
                AvgCount +=1
        if AvgCount > 0:
            AvgHeading /= float(AvgCount)
            AvgHeading -= self.heading
        return AvgHeading

    def Cohesion(self,group):
        CentreMass = Vector2D()
        SteeringForce = Vector2D()
        AvgCount = 0

        for bot in group:
            if bot != self and bot.tagged :
                CentreMass += bot.pos
                AvgCount += 1
        if AvgCount > 0 :
            CentreMass /= float(AvgCount)
            SteeringForce = self.seek(CentreMass)
        return SteeringForce
```

Weighted sum that adds 3 steering behaviours together.

```
def sumBehaviours(self,delta):
        self.tagNeighbours(self.world.agents,self.neighbourR)
        if len(self.neighbours) == 0 :
            return self.wander(delta)
        cohesion = self.Cohesion(self.neighbours) * self.CohesionWeight
        alignment = self.Alignment(self.neighbours) * self.AlignmentWeight
        seperation = self.Seperation(self.neighbours) * self.SeperationWeight

        return cohesion + alignment + seperation
```

Default value

```
            self.cohesion = 0.0
        self.seperation = 0.0
        self.alignment = 0.0
        self.radius = 10.0
```

Control value of each steering behaviour

```
elif symbol == KEY.A:
        for agent in world.agents:
            agent.SeperationWeight += 0.5;
    elif symbol == KEY.S:
        for agent in world.agents:
            agent.SeperationWeight -= 0.5;
    elif symbol == KEY.T:
        for agent in world.agents:
            agent.CohesionWeight += 0.5;
    elif symbol == KEY.Y:
        for agent in world.agents:
            agent.CohesionWeight -= 0.5;
    elif symbol == KEY.D:
        for agent in world.agents:
            agent.AlignmentWeight += 0.5;
    elif symbol == KEY.F:
        for agent in world.agents:
            agent.AlignmentWeight -= 0.5;
    elif symbol == KEY.N:
        world.separation = 0
        world. alignment = 0
        world.cohesion = 0
        world.radius = 10
```

Parameters of each value are shown on screen

```
if self.showinfo:
            infotext = ', '.join(set(agent.mode for agent in self.agents))
            egi.red_pen()
            egi.text_at_pos(0, 0, infotext)
            infotext = ': '.join(('Cohesion', str(self.cohesion)))
            egi.red_pen()
```

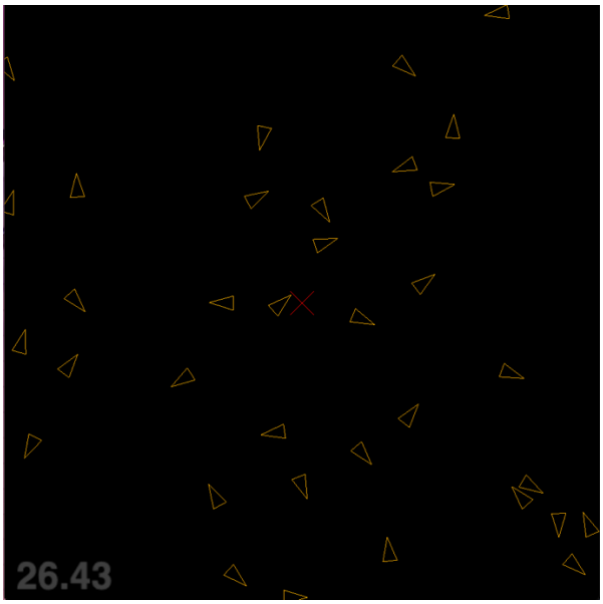
```
        egi.text_at_pos(0, 487, infotext)
        infotext = ': '.join(('Separation', str(self.seperation)))
        egi.red_pen()
        egi.text_at_pos(0, 448 , infotext)
        infotext = ': '.join(('Alignment', str(self.alignment)))
        egi.red_pen()
        egi.text_at_pos(0, 461, infotext)
        infotext = ': '.join(('Radius', str(self.radius)))
        egi.red_pen()
        egi.text_at_pos(0, 480, infotext)
```

Issue:

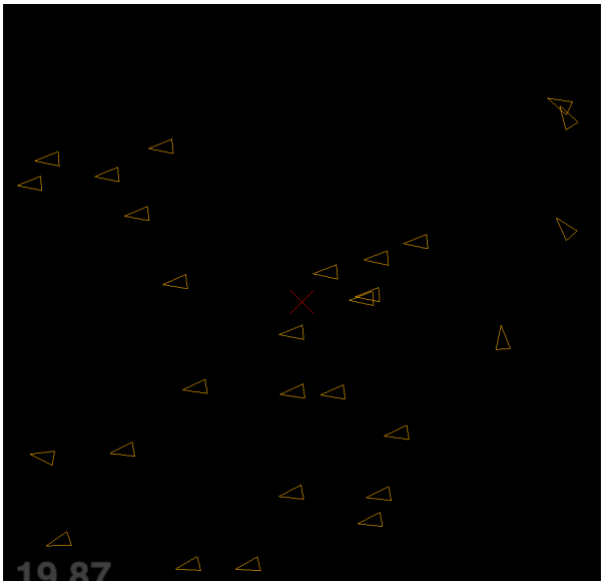
Parameters cannot be displayed on screen. Egi.[ ]_pen is not working. This can be due to Mac issue since it has occurred in previous labs before.

Output:

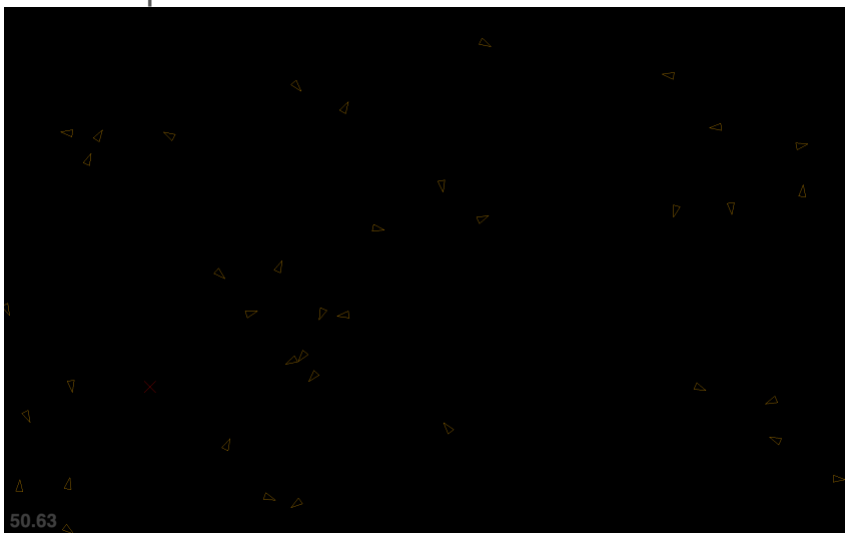
Normal screen



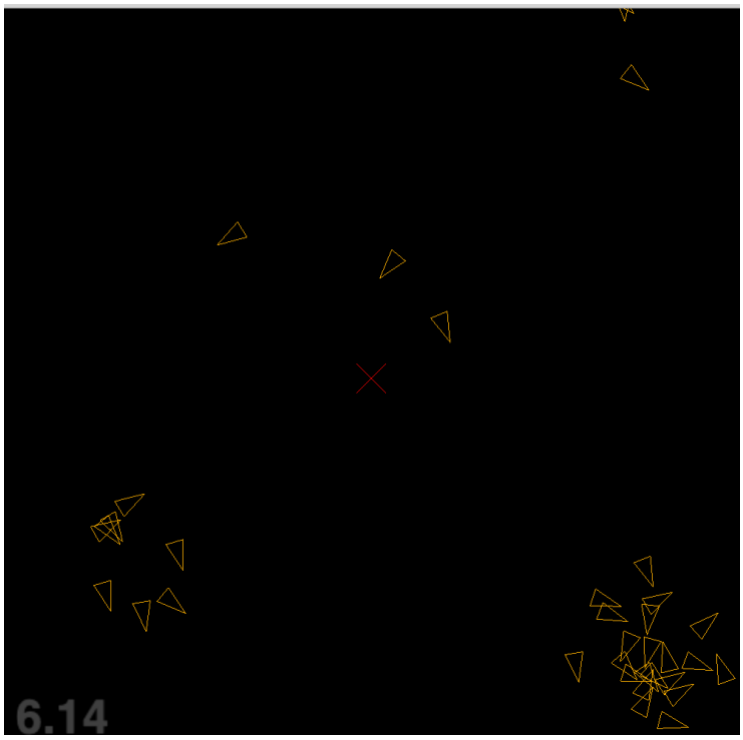
Alignment enabled



Separation enabled



Cohesion enabled


6.14