Spike Outcome Report

Number: 02

Spike Title: Emergent Group Behaviour

Personal: Peter Argent (7649991)

Goals:

Create a group agent steering behaviour simulation that can demonstrate distinct modes of emergent group behaviour. The simulation must:
• Include cohesion, separation and alignment steering behaviours
• Include basic wandering behaviours
• Use weighted-sum to combine all steering behaviours
• Support the adjustment of parameters for each steering force while running

Technologies, Tools, and Resources used:

- The code completed in Lab06
- Visual Studio 2017 with Python 3 installed
- Pyglet Documentation here: http://pyglet.readthedocs.io/en/pyglet-1.3-maintenance/
- Help from peers.

Tasks undertaken:

- Download and Open the code from lab 06

- Added Variables to the world.py for the weighting of cohesion, separation and alignment steering behaviours.

```
# Cohesion/Sepertion/Alignment Variables ## Added 2018-04-16
self.cohesion = 0.0
self.seperation = 0.0
self.alignment = 0.0
self.radius = 10.0
```

- Added the neighbourhood mode to agent.py.

```
AGENT_MODES = {
    KEY._1: 'seek',
    KEY._2: 'arrive_slow',
    KEY._3: 'arrive_normal',
    KEY._4: 'arrive_fast',
    KEY._5: 'flee',
    KEY._6: 'pursuit',
    KEY._7: 'follow_path',
    KEY._8: 'wander',
    KEY._9: 'neigbourhood',
}
```

In __init__()

```
# If Tagged is true, We are part of a neighbourhood
self.tagged = False
```

In agent.calculate()

```
    force = self.wander(delta)
elif mode == 'neigbourhood':
    self.find_neighbours(self.world.agents, self.world.radius)
    force = self.wander(delta)
    force += self.seperation(self.world.agents) * self.world.seperation
    force += self.cohesion(self.world.agents) * self.world.cohesion
    force += self.alignment(self.world.agents) * self.world.alignment
```

- The Weighted Sum for the Force for the neighbourhood calculation requires separation(), cohesion() and alignment() methods, Which in turn require a find-neighbours() method.

```
def find_neighbours(self, bots, radius):
    for bot in bots:
        bot.tagged = False
        # get distance between self.pos and bot.pos
        dis = Vector2D.distance_sq(self.pos, bot.pos)
        if dis < (radius + bot.bRadius)** 2:
            bot.tagged = True

def alignment(self, group):
    avgHeading = Vector2D()
    avgCount = 0

    for agent in group:
        if self != agent and agent.tagged:
            avgHeading += agent.pos
            avgCount += 1

    if avgCount > 0:
        avgHeading /= float(avgCount)
        avgHeading -= self.heading
    return avgHeading
```

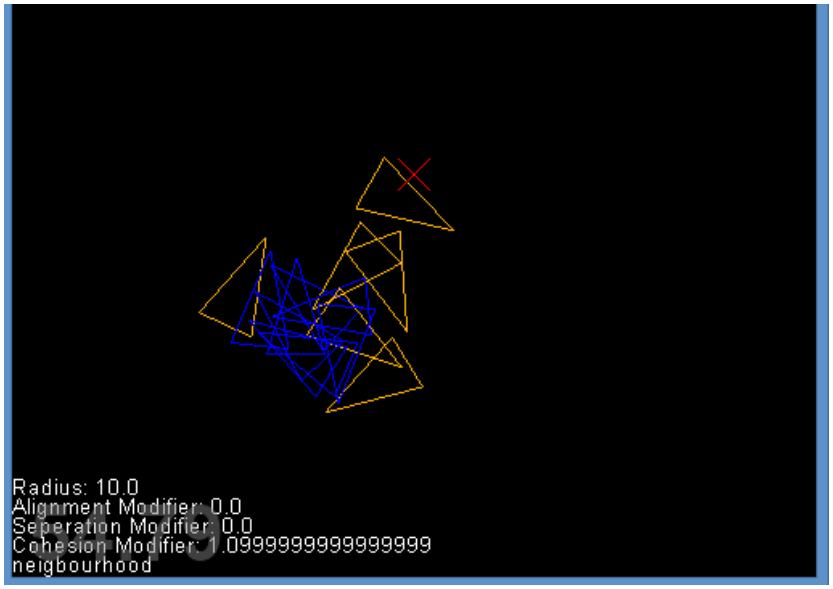
```
def cohesion(self, group):
    """
    This returns a steering force towards the center of the group
    """
    centerMass = Vector2D()
    steeringForce = Vector2D()
    avgCount = 0

    for agent in group:
        if self != agent and self.tagged:
            centerMass += agent.pos
            avgCount += 1

    if avgCount > 0:
        centerMass /= float(avgCount)
        steeringForce += self.seek(centerMass)

    return steeringForce
```

```
def seperation(self, group):
    """
    This returns a steering force away from the center of the group
    """
    centerMass = Vector2D()
    steeringForce = Vector2D()
    avgCount = 0

    for agent in group:
        if self != agent and self.tagged:
            centerMass += agent.pos
            avgCount += 1

    if avgCount > 0:
        centerMass /= float(avgCount)
        steeringForce += self.flee(centerMass)

    return steeringForce
```

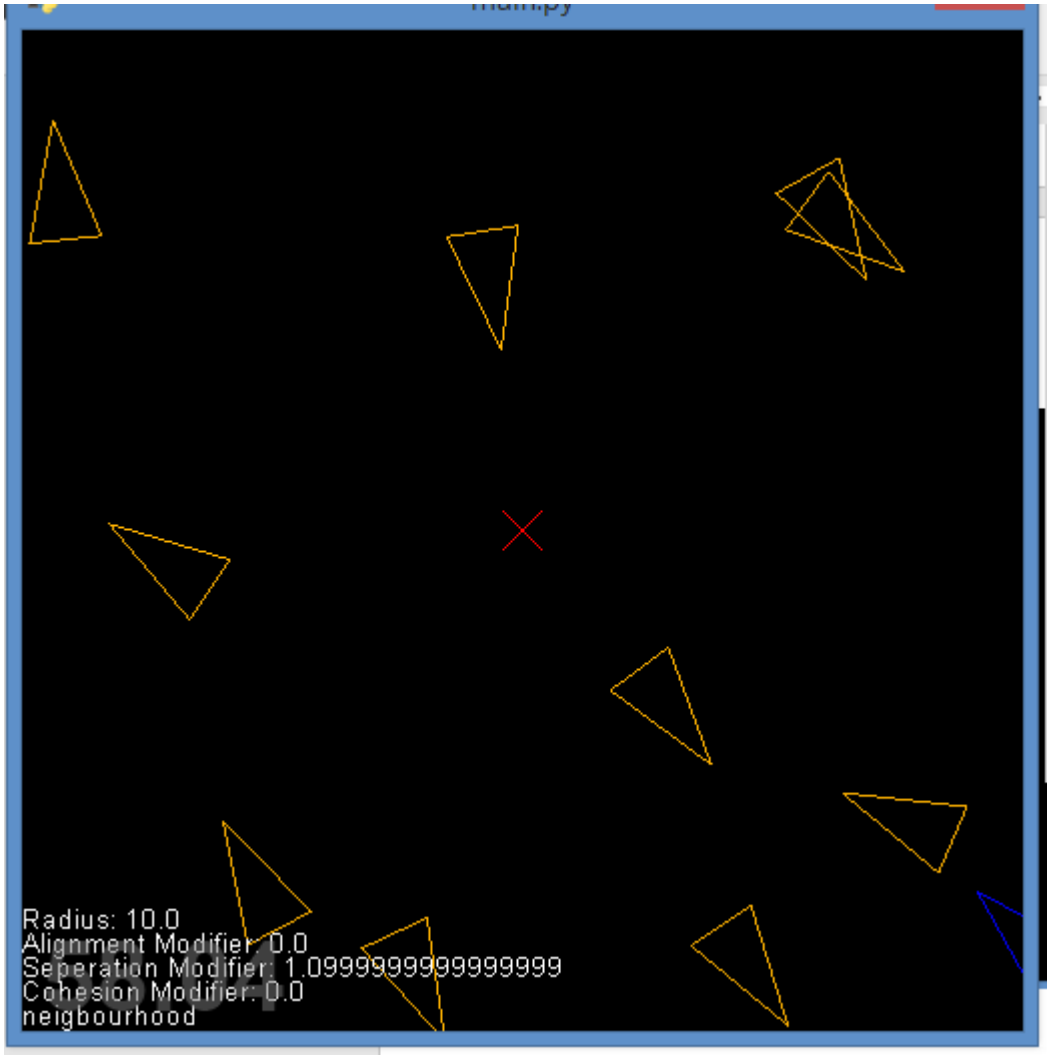
- Then Add Controls to change the weighting of each of the component parts of the weighted sum and how within how large of a radius each agent will consider others to be their neighbours.
I used the brackets keys to increase or decrease the modifiers and I used Shift, Ctl and Alt to change which modifiers were being modified

What we found out:

- High Cohesion but very low Alignment and Separation cause the agents to flock together and mill about

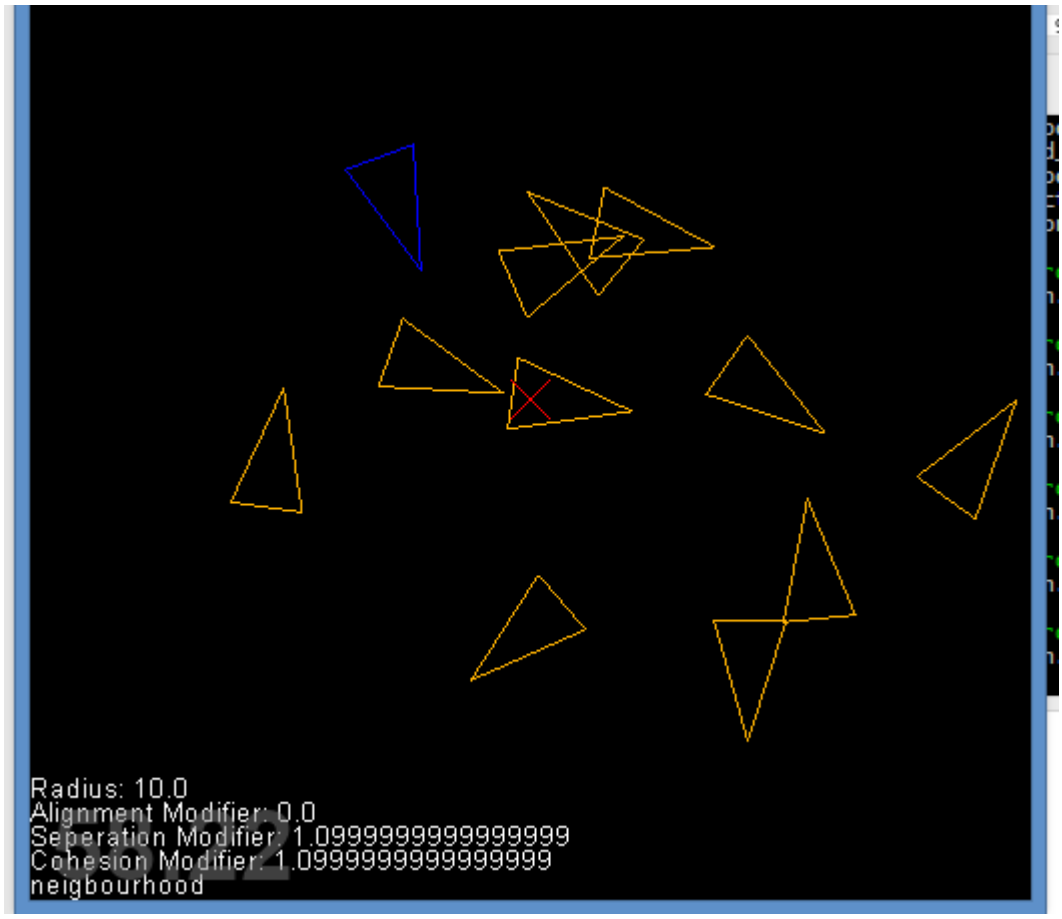


- High Separation but low Alignment or Cohesion causes the agents to tend towards the corners as they wander

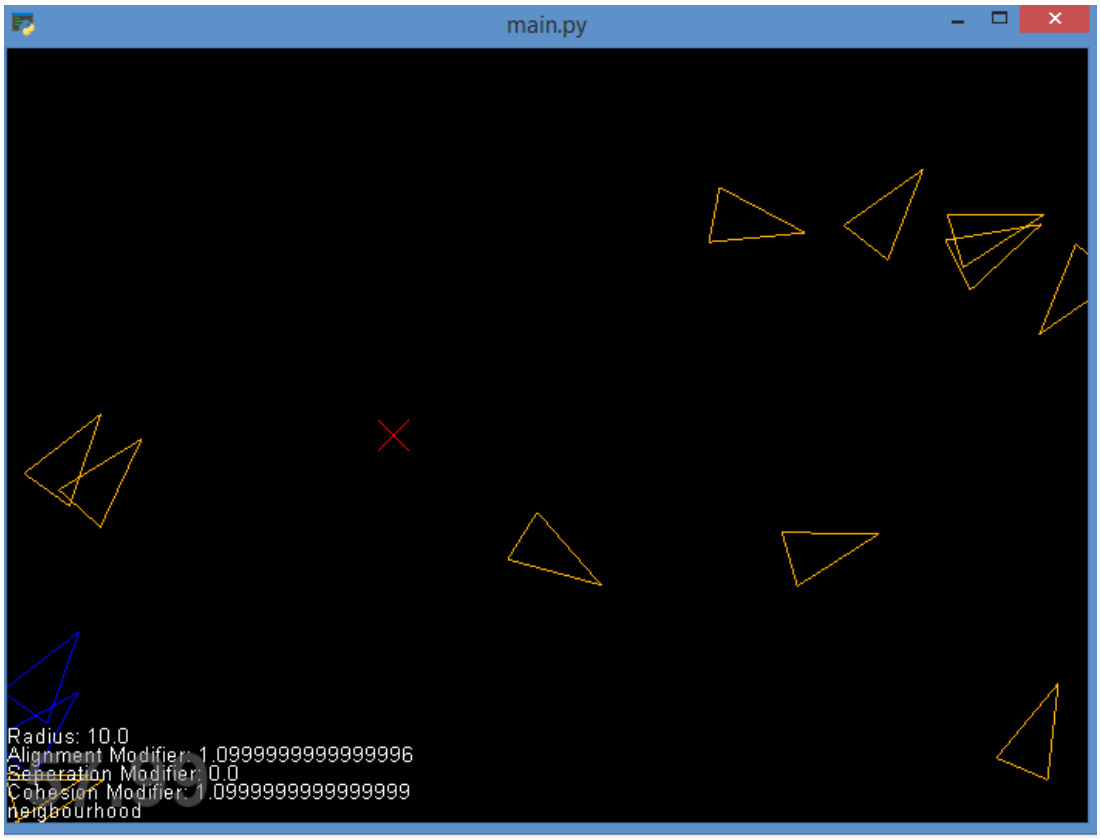


- High Alignment with very low Separation and Cohesion cause nearby agents to move in the one direction for a little bit before they wander away from each other

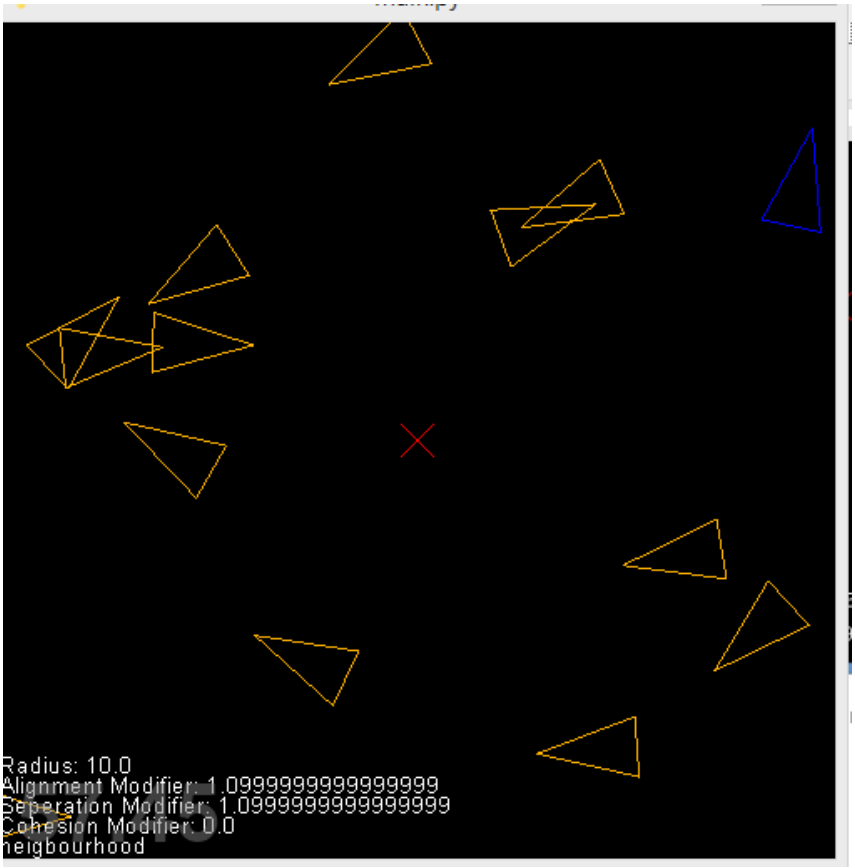
- Low Alignment but high Separation and Cohesion cause the agents to spin around in circles when near each other simultaneously trying to get away and clump together.



- Low Separation but high Alignment and Cohesion cause the agents to group together and move in one direction until they go off screen, but they tend to clump too much



- Low Cohesion but high Separation and Alignment cause the agents to run away from each other on the edges of the screen. Unlike purely high separation, the agents tend to avoid the middle of the screen even harder.



Recommendations:

- Additional spikes may be used to explore the addition of other steering behaviours to the weighted sum, including wall avoidance, object collision and other similar behaviours.

Notes:

- Controls for the Emergent Group Behaviour.
    - [decreases cohesion
    -] increases cohesion
    - Shift-[decreases radius
    - Shift-] increases radius
    - Ctl-[decreases separation
    - Ctl-] increases separation
    - Alt-[decreases alignment
    - Alt-] increases alignment