**Spike:** 14

**Title:** Emergent Group Behaviour

**Author:** Ben Holmes, 103024841

**Goals / deliverables:**

Create a group agent steering behaviour simulation that is able to demonstrate distinct modes of emergent group

behaviour. In particular, the simulation must:

• Include cohesion, separation and alignment steering behaviours

• Include basic wandering behaviours

• Use a weighted-sum to combine all steering behaviours

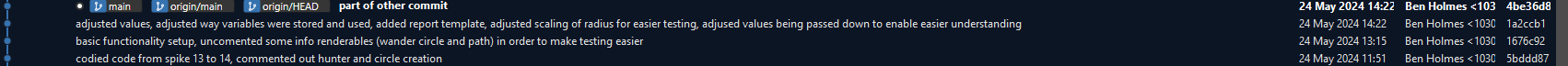
• Support the adjustment of parameters for each steering force while running

• Spike outcome report and working code (with key instructions).

**Technologies, Tools, and Resources used:**

* Visual Studio Code
* Python 3.12.2
* Pyglet

**Tasks undertaken:**

****

* Copied code from spike 13
* Commented out hunter and circle creation code, as well as wander circle render and path render for ease of use
* Added the 4 needed group functions of tag\_neighbour, separation, alignment and cohesion

A computer screen shot of text

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

A computer screen shot of text

Description automatically generated

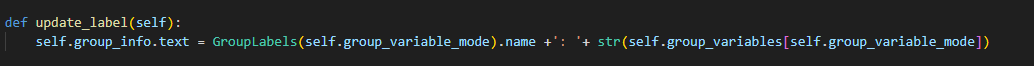
* Added necessary world.py variables for scaling of the vectors to enable weighted sum

A black screen with green text

Description automatically generated

* Added label functionality for the types of parameters as well as group label enum





A screen shot of a computer program

Description automatically generated

* Added functionality to change parameters on key presses ( all but radius increase and decrease by 1, radius by 10, they all cannot go below 0)

A screen shot of a computer program

Description automatically generated

* Added overall group behaviour calculation

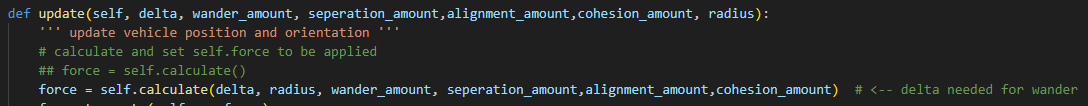
A screen shot of a computer code

Description automatically generated

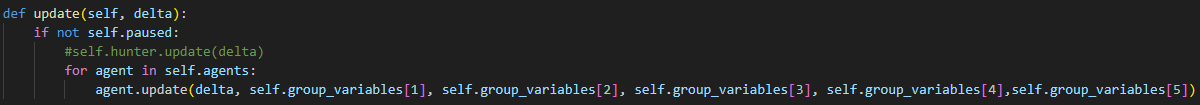
Calculate function:



Update function:



World.py update function:



* Then tested and adjusted any mistakes

**What we found out:**

The basic functionality was relatively simple to implement however I had some issues around testing and various other parameter implementation, mainly to do with creating the dictionary and not understanding exactly what was being stored there.

The testing was seemingly rotating around the same point constantly, which I figured out was first the parameters not adjusting as a result of the dictionary mistake and second, the radius increments were too large for the scale.

**Testing buttons:**

M switches parameter (or group variable modes) increasing through the list, with N doing the inverse

Left decreases and Right increases the parameter value