Spike: 6
Title: Soldier on Patrol

Author: Ryan Chessum - 102564760

Goals / deliverables:

Create a “soldier on patrol” simulation where an agent has two or more high-level FSM modes of behaviour and low-level FSM behaviour. The model must show (minimum)
(a)High level "patrol" and "attack" modes
(b)The "patrol" mode must use a FSM to control low-level states so that the agent will visit (seek/arrive?) a number of patrol-path way points.
(c)The "attack"mode must use a FSM to control low-level fighting states. (Think “shooting”, “reloading” –the actual states and transition rulesare up to you.)

Technologies, Tools, and Resources used:

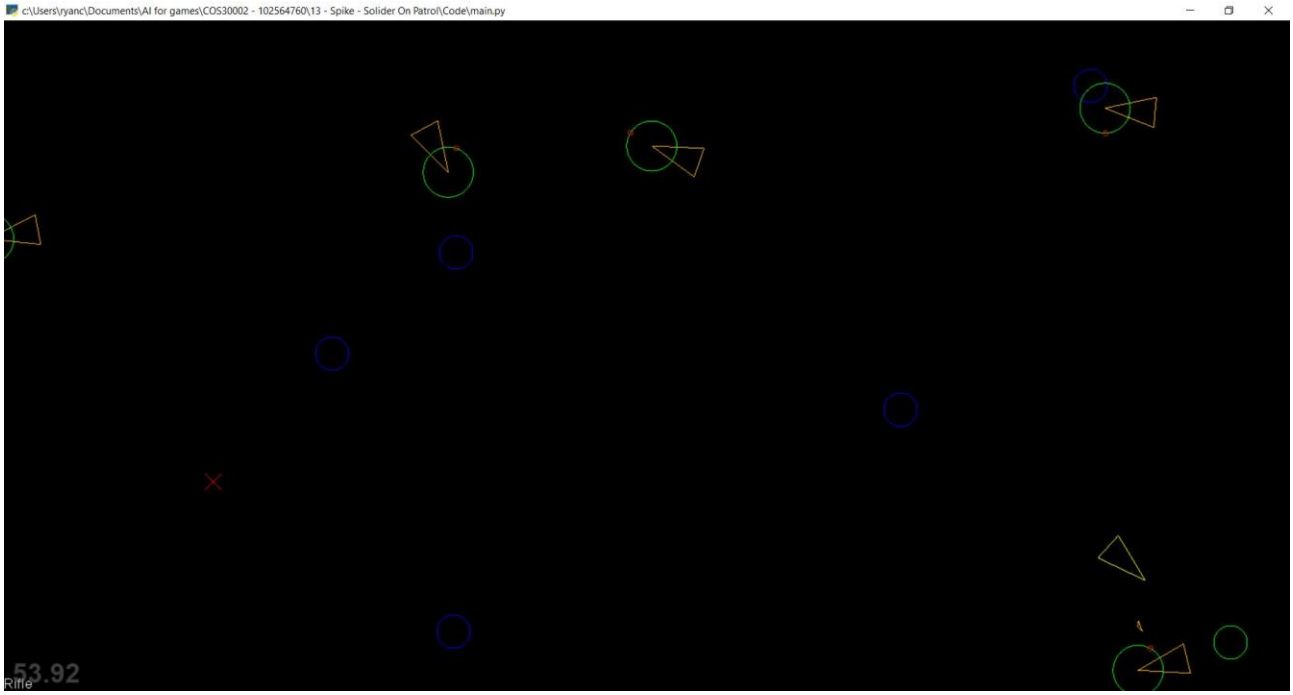
- Visual Studio Code
- Python 3.8.0
- Pyglet
- Lecture Notes
- Code from Labs
- Code from Previous spikes

Tasks undertaken:

- Import necessary code from previous tasks
- Add a new state machine to the marksman’s calculate function that controls the high level “attack” and “patrol” states
- Add a patrol behaviour that uses a state machine
- Add an attack behaviour that uses a state machine
- Add a function that switches the high level state machine between patrol and attack based on if a target is nearby

What we found out:

Finite state machines are a great way to control agent behaviour and create more dynamic agents.



The marksman has 2 high level states, patrol and attack. When in the patrol state the marksman will go from point to point on it's patrol route. If it encounters a target nearby in it's detection range on the way it will switch it's High level state to Attack where it will approach the target while trying to shoot it.

```
for target in found_targets:
    if Vector2D.distance(self.pos, target.pos) < shortest_dist:
        shortest_dist = Vector2D.distance(self.pos, target.pos)
        closest_target = target
if closest_target != None:
    self.mode = 'Attack'
else:
    self.mode = 'Patrol'
return closest_target
```

When in the patrol state it has 2 other low level states, stopped and walking. When walking the marksman will walk to the next point in it's patrol route which is marked green. When it arrives at this point it will switch its state to stopped for a second then change it's state back to walking and move on to the next state.

```
def patrol(self):
    force = Vector2D()
    if self.patrol_mode == 'Stopped' and time.time() - self.last_stopped > 1:
        self.patrol_mode = 'Walking'

    if self.patrol_mode == 'Walking':
        force = self.arrive(self.waypoints[self.current_waypoint], 'slow')

        if self.pos.distanceSq(self.waypoints[self.current_waypoint]) < 25:
            self.next_waypoint()
            self.last_stopped = time.time()
            self.patrol_mode = 'Stopped'
    return force
```

When attacking it has another 2 low level states, loaded and reloading for it's weapon. After firing it's weapon the marksman's state will go from loaded to reloading. It will not be able to shoot again until it has switched it's state back to Loaded. The time it takes the agent to reload it's weapon depends on the cooldown time of the equipped weapon.

```
def update(self, delta):
    target = self.find_target()
    force = self.calculate(delta, target)
    if time.time() - self.last_fired > WEAPON_COOLDOWN[self.weapon]:
        self.attack_mode = 'Loaded'
    if target != None and self.attack_mode == 'Loaded':
        self.shoot(target)
        self.attack_mode = 'Reloading'
```

Keys:
A – spawn new target
R – Move marksman to random position and randomise patrol area
1 – Set Weapon to Sniper Rifle
2 – Set Weapon to Rocket Launcher
3 – Set Weapon to Hand gun
4 – Set Weapon to Hand grenade
P – Pause