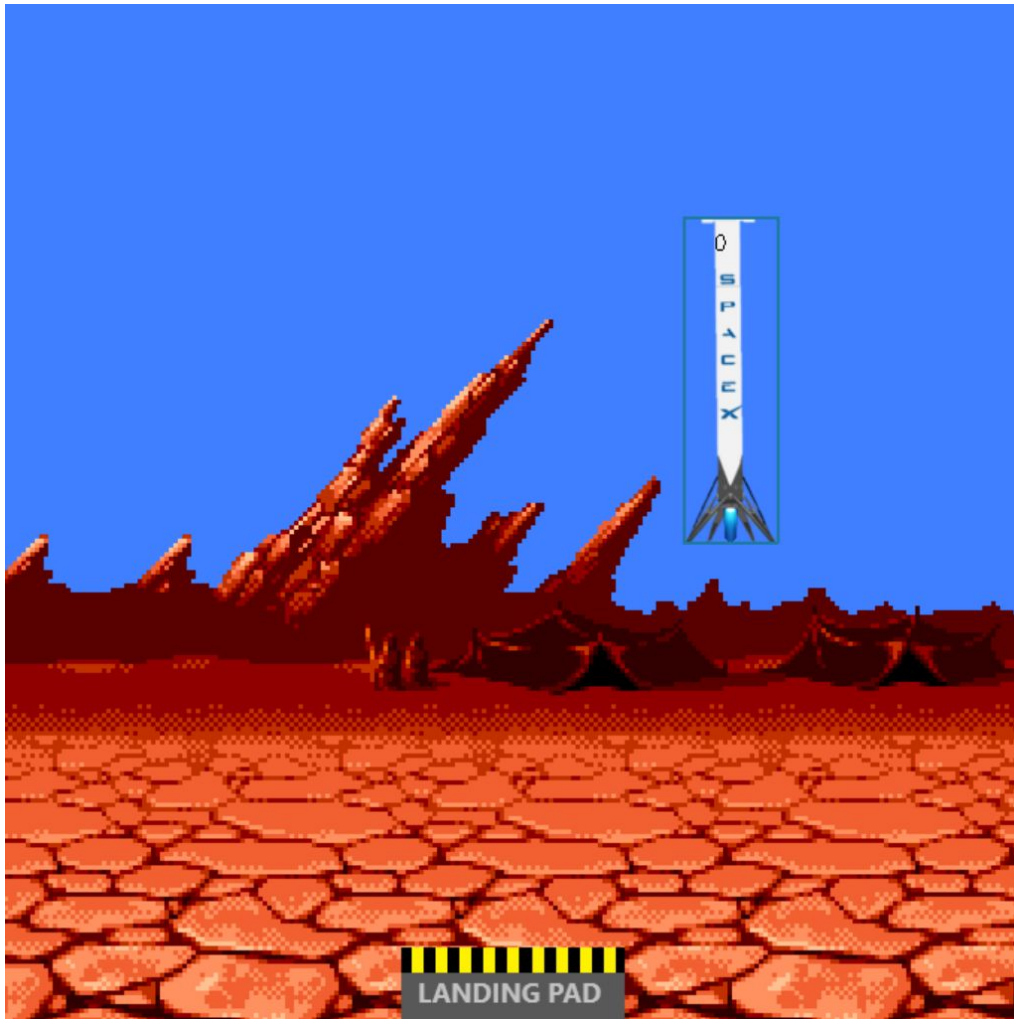


Rocket Booster Agent

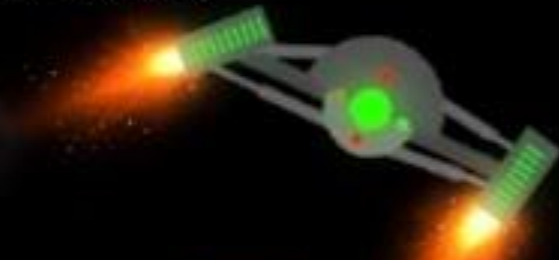
Harth Majeed
3547 Intelligent Agents &
Reinforcement Learning



Motivation



Train a *Neural Network*
to operate



Drones



Objectives

- Build an animated Rocket
 - Pygame
- Implement AI
 - NEAT
 - NeuroEvolution of Augmenting Topologies



Pygame Implementation - Difficulties and Challenges

- Loop structure
- Physics and Clock
- Boundaries
- Landing Pad
- Crashing outside landing pad



NEAT Implementation - Difficulties and Challenges

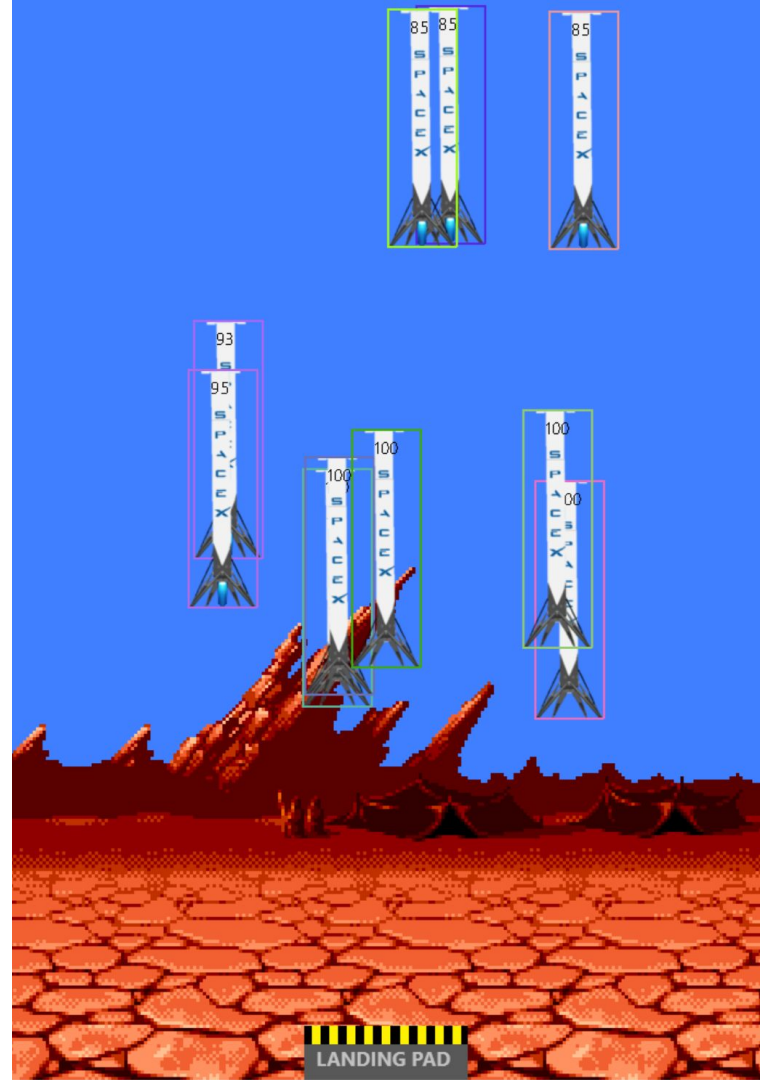
- Required overhaul of the code
- Application code must accommodate the NEAT code architecture

```
if output[0] > 0.5:
    agent.thrustBool = True
if output[1] > 0.5:
    agent.leftThrustBool = True
if output[2] > 0.5:
    agent.rightThrustBool = True
if output[2] > 0.5:
    agent.idleThrustBool = True
```

```
#controls of the agent rocket
def control(self):
    if self.leftThrustBool and self.fuel > 0:
        self.set_image(rocket_left)
        self.x_acc = -acc_rate
    if self.rightThrustBool and self.fuel > 0:
        self.set_image(rocket_right)
        self.x_acc = acc_rate
    if self.thrustBool and self.fuel > 0:
        self.set_image(rocket_thrust)
        self.y_acc = -acc_rate
    if self.idleThrustBool:
        self.set_image(rocket)
        self.x_acc = 0
        self.y_acc = acc_rate
    self.thrustBool = False
    self.leftThrustBool = False
    self.rightThrustBool = False
    self.idleThrustBool = False
```

Results

- Results were poor
- Tuning the configuration of NEAT
 - Inputs 3,4,7,10
 - Outputs 3,4
 - Hidden Layers 0,1,2,4,10
 - Up to 500 generations
- The issue lies in 2 places
 - The scoring system
 - The physics



Conclusion

- Project was too ambitious for a beginner
 - Beginner in animation
 - Beginner in animation physics
 - Beginner in applying Genetic Algorithms like NEAT
- Future work
 - Start simple
 - Build up

