Game Design Document

I didn’t know what my game was going to be until June. Planning for the game began early 2018 when the theme was released. The aspects that stood out the most from Transformation were the mutation and evolution. I had dabbled in machine learning and neural networks before and was inspired by genetic algorithm simulators on YouTube to create something similar. At first I had no clue what to make, early brainstorms developed ideas like evolving from a fish to a human, training a car how to move through mazes and even a plant growing game. ~~In spite of the continual emphasis on planning, planning, planning, at the time of writing this game document in early April I still don’t know what my game will be~~

**Game Title**

The game is named after the game’s protagonist Dr. Darwin, similar to how Donkey Kong is named after the game’s protagonist. The title is an obvious reference to Dr. Charles Darwin.

**Game Description**

Evil scientist Dr. Darwin is on a mission to evolve the perfect killing creature. Survive every generation by shooting or evading the creatures. How many generations can you survive before the creatures outsmart you?

**Audience**

Dr. Darwin is intended for all ages but primarily adolescents, teenagers. Anyone who enjoys critical thinking games and/or has an interest in evolution simulators

**Characters/Roles**

The protagonist is an evil scientist who moves about in a flying saucer. He is named Dr Darwin after Charles Darwin who invented the theory of evolution. His objective is to breed a unique creature who can effectively target the player while also dodging incoming bullets shot by the player

The player is a tank. They can move about, shooting at creatures. The player has a health bar which decreases when they shoot a bullet or are attacked by creatures or are almost outside the wall.

Dr Darwin breeds creatures who have the ability to detect the player, bullets, walls and vary its speed and direction based on its detections.

**Environment**

The environment is very dark with walls that border the gameplay area keeping the creatures and player in a confined area. The wall not only can be used by the player to eliminate creatures but also confines the gameplay to make it more intense and challenging. The player loses health if they are too close to the wall

**Theme**

The concepts that stood out to me the most from transformation were mutation and evolution. The game very much follows both these ideas as the creatures are transformed and mutated every generation.

**Objective/Goals**

The player’s objective is to survive as many generations as possible. To defend themselves they can either shoot or evade the creatures. This task gets harder as the game progresses and the creatures get smarter. A clever strategy would be to find the weakest creatures and

**Perspective**

Perspective is a fixed birds eye view of the 2D arena and the tank is simply moved around inside the confines of the game

**Controls**

The player uses the arrow keys to move the tank, left to rotate anti-clockwise, up to move forwards, right to rotate clockwise and down to move backwards. To shoot, players use the spacebar.

**Reference points/Originality**

No other games come to mind that have similar gameplay mechanics to Dr. Darwin.

**Platform**

My mentor says the game will need to be run on Windows. When I tested the game on a Windows I found the game was much much faster than when on my own MacBook laptop. From this I had to change my game to update based on the clock tick rather than per frame.

**Development Environment**

I developed the game on a MacBook laptop in Python using the Pygame graphics library as well as the Numpy library.

**System Requirements**

A keyboard is needed to control the tank and a mouse is needed to click buttons. As for graphics, a monitor of more than 1000x600 would be needed.

**Resourcing/Capability**

I will need Python installed along with the Pygame and Numpy libraries. I will also require a music making software in order to create game music and graphics software

**Style**

I want to emulate a type of minimalist retrofuturism style, one with simple, bright colours, non-complex shapes. I will use Consolas for the font throughout as it gives a techy and artificial feel.

**Process**

Because of the simplicity of the style, I won’t require powerful graphics software. The best software I can think of, believe it or not is Microsoft Powerpoint and using the shapes to create characters, environment and so on.

**Timeline/Deadline**

I aim to have the game finished by July to start game-testing and fixing any nuances in the gameplay. The creatures in the game operate using randomly generated neural networks so this would require a lot of time-consuming testing.

**Responsibility**

I’m a one man team

**Submission Guidelines**

I’ve had a look through all the guidelines and theme criteria to ensure the submission is valid

**Other**

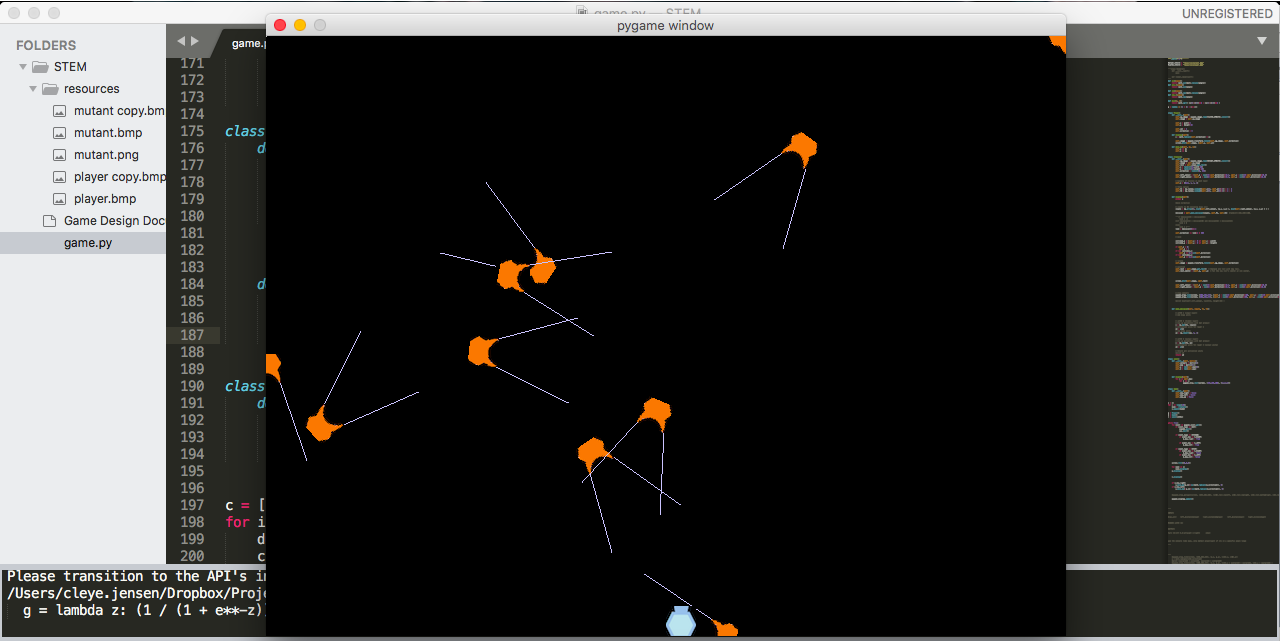
The creatures in the game operate using neural networks so this would require a lot of testing

**Timeline**

My priority is to test the creature’s behaviour so the first designs of both the player and creature were prototypes. The idea I had in mind for the creature would be menacing while also simple to reemphasise the idea that they are a replicated species. The creatures ended up looking like the head of an ant which was suitable. The player was intended to be a kind of robotic killing machine and turned out as a weird looking hexagonal object.

The first problem I encountered was learning how to rotate the images. In Pygame, the size of the rectangle container of an image was changed when the image rotated making the character’s movement very erratic. A week later I fixed it by having the rectangle translated based on the the image’s direction so that rotation was much more smoother. Now I can move on to making the creatures to make decisions

I started testing the neural network of the creatures. I made 10 creatures each with their own neural network



As game development went on, ideas that were too hard to implement were scrapped and new ideas were taken on board

Firstly I planned what inputs and outputs the creature’s neural network should have

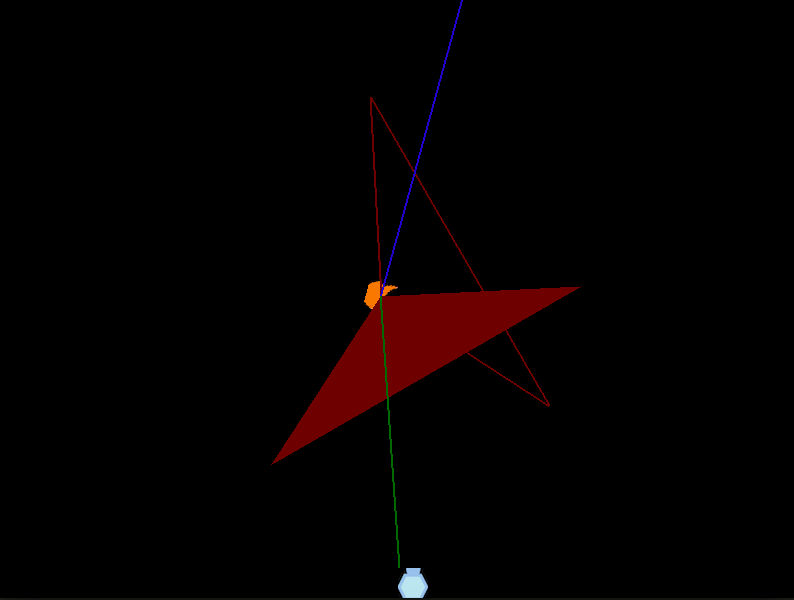
INPUTS bias\_unit left\_distance2player right\_distance2player left\_distance2wall right\_distance2wall

HIDDEN LAYER (4)

OUTPUTS turn\_left turn\_right shoot

At first I planned to make the creatures detect players by how far away the left side and the right side of the creature were from them. However if both eyes were 600pixels away from the creature, it doesn’t tell much

Instead I would use the sensors like eyes, only detect player/wall if its in a specific angle range



Another issue was the turn output had 2 different values: turn\_left and turn\_right, but this looked jittery and unnatural

The alternative was to create 0-1 turning values for the direction

The evolutionary algorithm would work by evaluating all the creatures fitnesses. Their fitness would then be taken into account when half the population is sampled weighted on their fitness (i.e. so while the fittest creature had the highest likelihood of breeding it wasn’t guaranteed to be chosed) this was to ensure diversity in the creatures genes. The chosen half of the population was breeded by mutating their genes, this was done by slightly varying the parameters in the creature’s neural network

Fitness was based on rewards and punishments, all were chosen to breed creatures that could chase and attack the player and dodge bullets.

Rewards are:

damage done on player

average distance to player

time with player in sight

--ability to sense bullets and dodge them--- too hard

Punishments are:

being stationary

running into walls

being hit by bullets

The creatures in each generation have 20s to show their fitness.

Hopefully too many creatures with computationally expensive neural networks won’t slow down the game, so that limited the population in each generation

to do:

music

storyline

gui

Other ideas that never made it:

The creature shoots back

Levels

some creatures drop powerups when they die

4 waves (a wave is 16 specimens of a species) come for each generation

species:

twirlers, sharpshooters

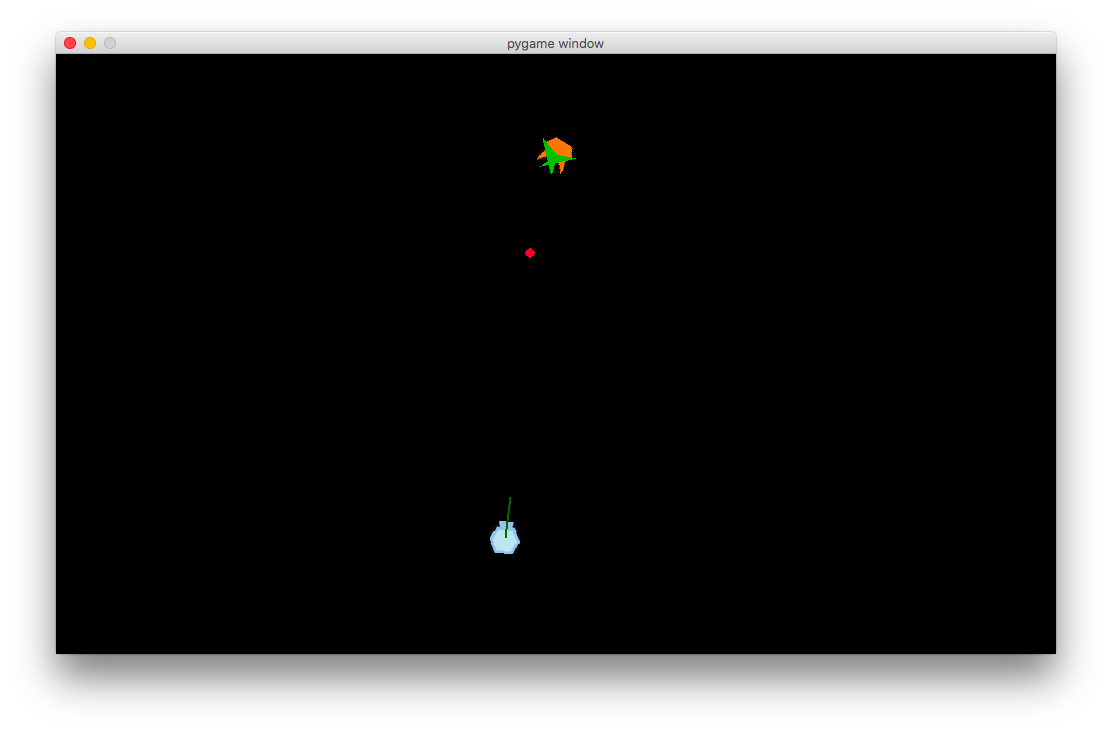
survive as many generations as possible

better start gui

borrowed button class off old program

problem: bullets were sensed even when moving away from them

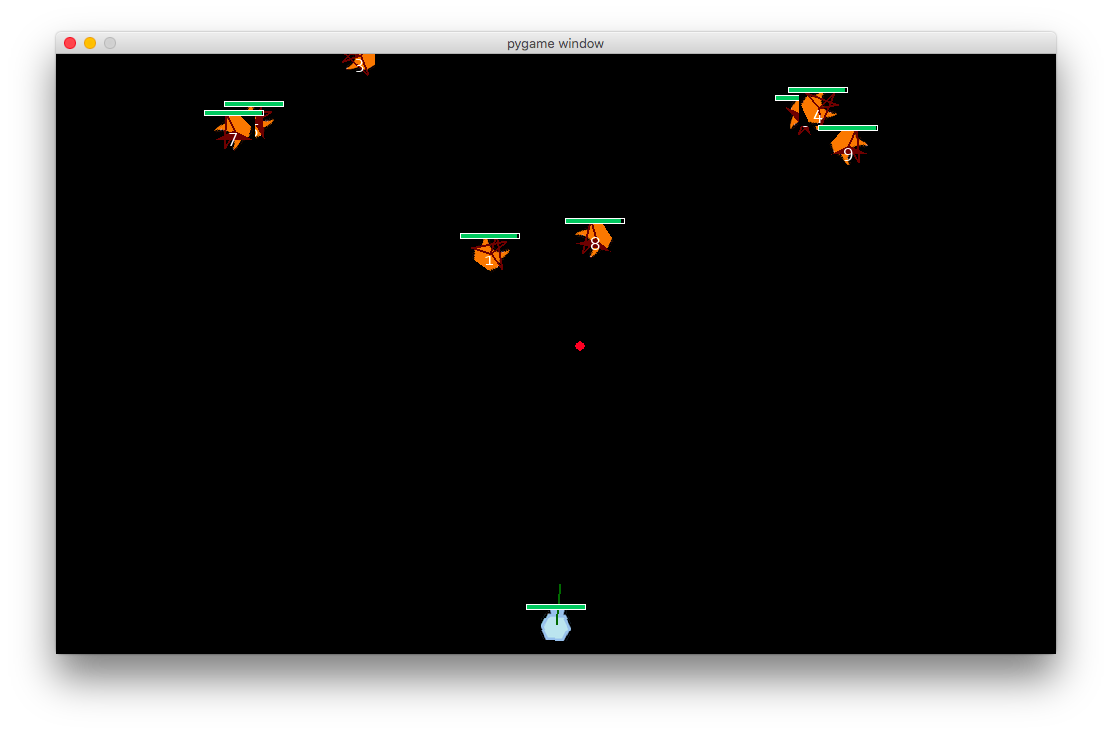
added angles to sensor, add periphery value



added distance to player, bullet sensing, wall sensing as parameters

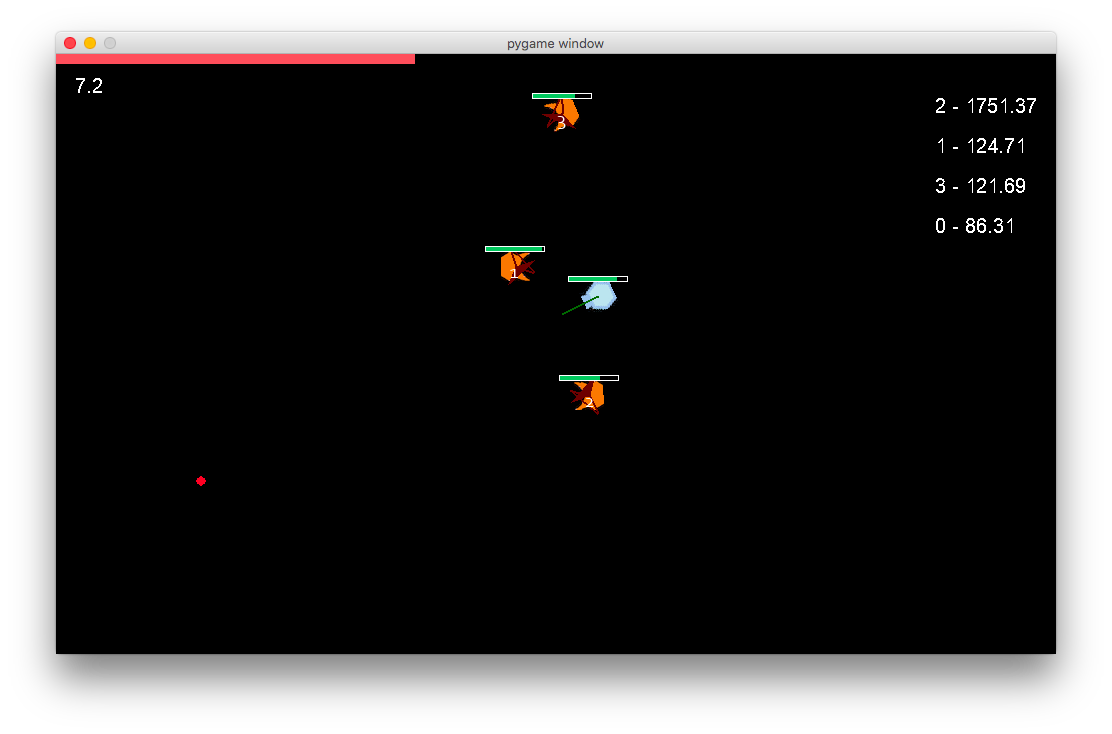
wall proximity function

added speed output



I’ve had the idea that the creature has limited energy or life and this is expended the faster it moves, and also shooting it depletes its life by a random amount

if that’s the case I need an overarching class for movable objects which player and creatures inherit from



techniques:

try lure them into walls

move around them to get beind their back

fitness starts at 0:

reward:

damage done on player

average distance to player

time with player in sight

--ability to sense bullets and dodge them--- too hard

punish:

being stationary

running into walls

hitting bullets

Creature

- update

- display

- calculate\_fitness ->

- make\_decision -> output units

interesting notes:

some creatures were attracted to the bullets,

half the creatures were really smart, the other half fairly dumb

what inputs and outputs should the nn have

INPUTS(6) bias\_unit, left\_sensor\_detect, right\_sensor\_detect, distance2player, left\_sensor\_detect\_bullet, right\_sensor\_detect\_bullet, wall\_proximity

HIDDEN LAYER (4)

OUTPUTS(2) turn (0-left 0.5-straight 1-right) velocity

optimum creature:

if not detecting the player, rotate

if detecting player with both eyes, move forward

if close to player, move faster

if detect bullet in one eye, move away and fast

if detect bullet in both eyes, move away and fast

if close to wall, rotate

I saved music for last because I didn’t want to hear whatever song I put in played over and over again while I’m testing the game

approaching

I had no idea how to do the music then I used noteflight and audacity to record a piano sample

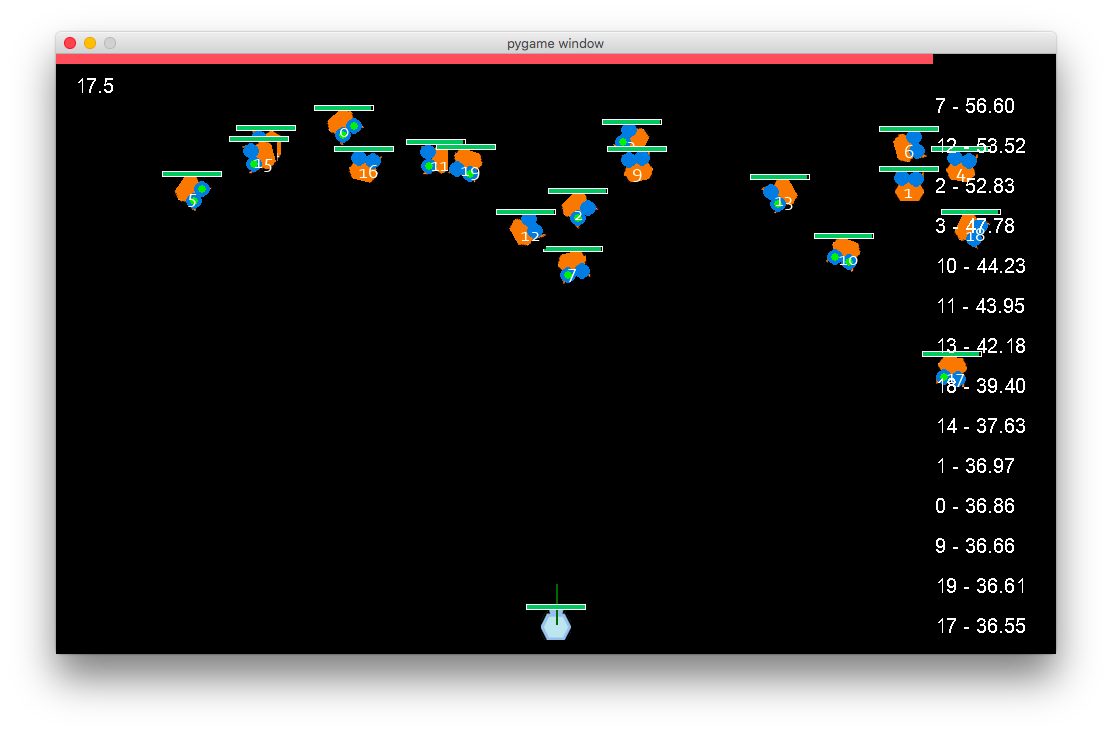
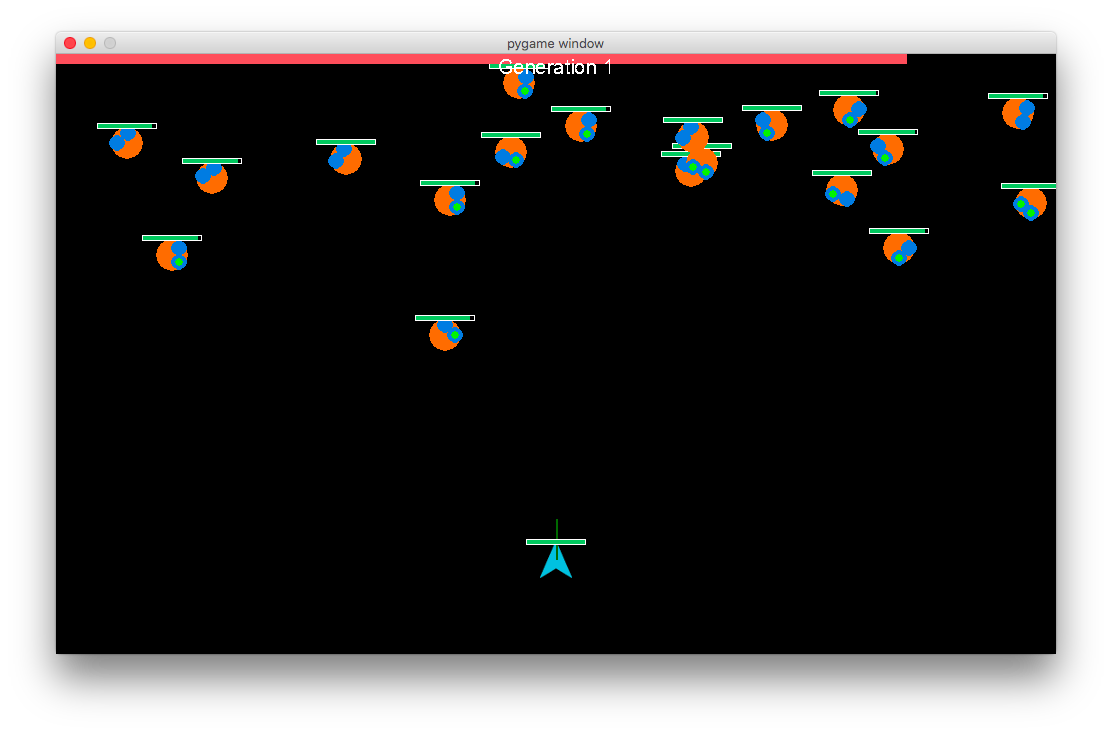
todo:

music

storyline

graphics

I experimented with the layers of the neural network and realised it was way too complex for the intended goal of following the player, I removed the hidden layer so that there were only one set of parameters thus making mutation easier.



to add difficulty increase speed and rotational velocity with generation

testing:

faster not smarter

player/creatures too close to wall, creatures kamikaze into wall

start creatures in centre

mutability don’t mutate fit creatures

win: defeat all in time limit

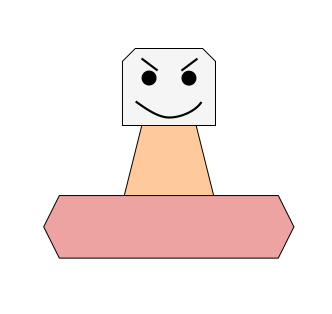
player wins back health if its bullet kills a creature

storyline:

Evil Dr Darwin is trying to breed to ultimate killing creature

Defend yourself by trying to kill them with bullets

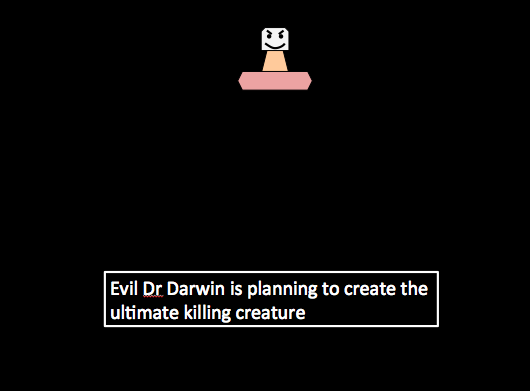
Run them into walls



had to implement scene class to control scenes

add animations like creature dying: creatures shrinks then sort of “pops” like Mario

intro to look like this



change creatures so they make sense