Bohan Li

Term Project Proposal

Project Description
My project is going to be called "Survival of the Fit Enough", it allows the players to create an "ancestor" creature by placing different weights on different attributes of the creature, such as speed, health, aggression, etc. The creature is then placed into an environment with randomly generated food, and it is allowed to procreate or multiply once it gathers enough food, and pass down its best traits to the next generation. Eventually, either the creature becomes better and better at surviving, or it becomes extinct and the game ends. Some possible features include a multiplayer mode where two players make two different creatures that compete, increasing difficulty(reduced food) to make the game harder, enemies/predators that randomly appear, etc.
Competitive Analysis
Similar Project 1: EvoSim (F17)
An evolution simulator in which the user designs a wheeled machine and models how it changes over time in an environment. It features a customizable environment and the creature's initial attributes, also graphs the relevant data at the end of each generation
Similar Project 2: Evolve! (F17)
An evolution game that features three different customizable environments, customizable creatures, evolution algorithm, and visualized data.

My project will be similar to these two projects in that, it will also feature a customizable environment, and the creatures initial conditions, the twist is, the player has a limited number of points so he or she has to decide wisely how to distribute them. Another difference is that my game will feature a lot of action because it seems like EvoSim features only one active mechanical creature at a time, whereas Evolve! features multiple, but they are stationary, so the user cannot really observe the creatures in action. My game will feature multiple creatures in each generation, and they will move around, gather food, avoid predators, etc. Therefore it should be more fun to watch since the interactive part ends after the initial customizing. I might not feature data visualization heavily, however, at least not initially, in order to decrease the difficulty of implementation.

Structural Plan
My game is going to have three main modes initially, the title screen, the customization screen for initializing the creatures, and the game screen for observation of the evolution of the creatures. The main files are going to be, the Tkinter framework, the creature class file, the food class file, the splash screen, settings, etc file, the machine learning file, and the main game file. The animation is based in Tkinter and the creatures are instances of the creature class, and the

creatures evolve based on the machine learning algorithm and the main game file sets up the title screen, the game screen and the game over screen, etc.

Algorithmic Plan

The trickiest part of the project will be implementing machine learning in order to evolve the creatures, in order to do this, I must address a few things:

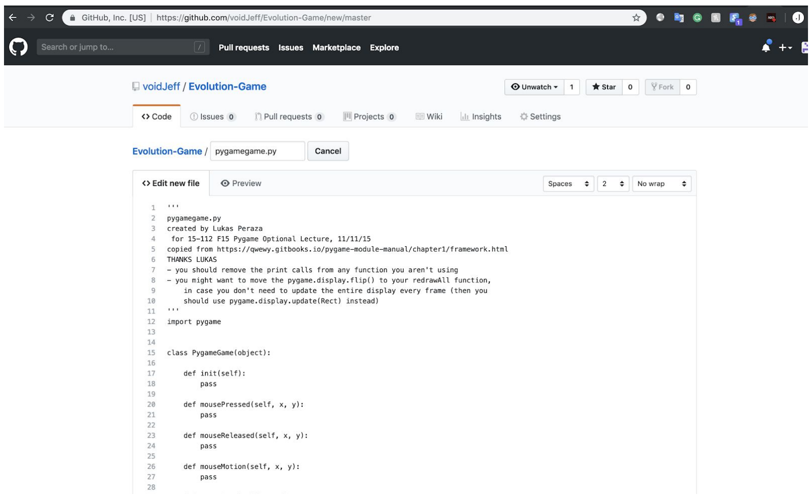
- How to measure the fitness of a given creature
  A function that takes various measurements, such as time taken to reach a certain food, average health, offspring counts, etc. then spits out a number that represents the fitness
- How traits are passed down, how to implement random mutations
  The good traits of a creature are passed down, but genetic variations must also be allowed in order to create a chance of improvements. The probability of mutations occurring should also be customizable
- How the creatures detect food and move towards it
  The creatures should have a field of vision represented by a circle of a certain radius, and if food is present within the radius, movements are made to move towards the general direction, the accuracy of the movement should also depend on the creature's attribute weight

Timeline Plan

Based on the agreed plan, (written by Komal)
TP1: The basic game, with customizable creatures and randomly generated food
Sunday, April 21st: complete AI, implement machine learning
Wednesday, TP2 (MVP): Have the Machine Learning part done, has evolution implemented with probabilistic survival rates of different varieties of creatures, etc.
Sunday, April 28th: complete machine learning, finalize project features other than cosmetics
TP3 (Final Submission): Improve the ML, improve UI, improve creatures appearance, and possibly add creature varieties.

Version Control Plan

I will use GitHub to back up my code and store them in repositories. Multiple files can be added and they can be easily organized. Here's an example:

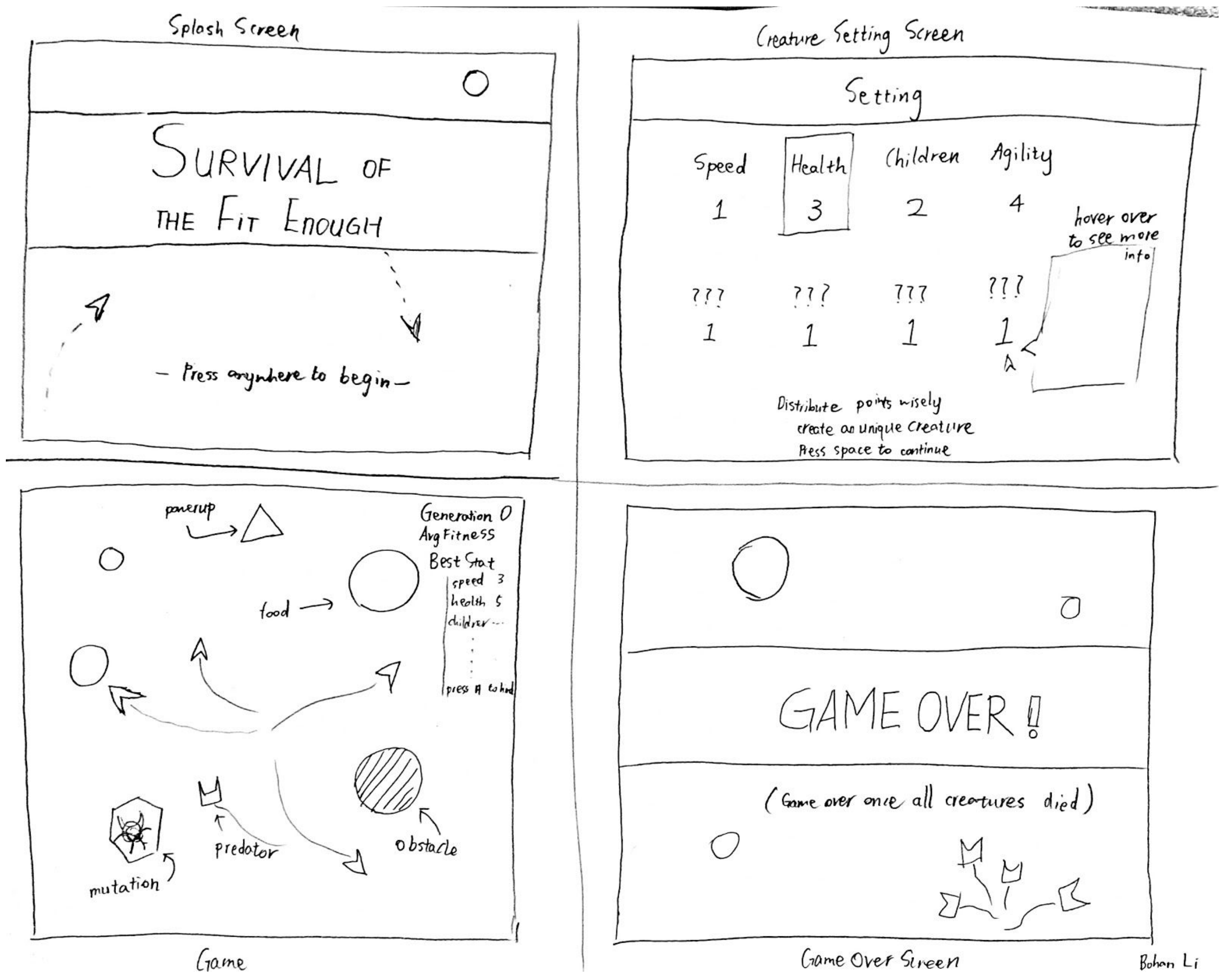


```
'''
pygamegame.py
created by Lukas Peraza
 for 15-112 F15 Pygame Optional Lecture, 11/11/15
copied from https://qwewy.gitbooks.io/pygame-module-manual/chapter1/framework.html
THANKS LUKAS
- you should remove the print calls from any function you aren't using
- you might want to move the pygame.display.flip() to your redrawAll function,
    in case you don't need to update the entire display every frame (then you
    should use pygame.display.update(Rect) instead)
'''
import pygame


class PygameGame(object):

    def init(self):
        pass

    def mousePressed(self, x, y):
        pass

    def mouseReleased(self, x, y):
        pass

    def mouseMotion(self, x, y):
        pass

```

Module List

- Pygame
  For the most part, pygame is the only external module. However, for now, I am going to stick with Tkinter unless certain features demand using pygame.
  Also, I suspect that machine learning frameworks would also be borrowed from online sources

Storyboard


Splash Screen
SURVIVAL OF THE FIT ENOUGH
– Press anywhere to begin –
Creature Setting Screen
Setting
Speed
Health
Children
Agility
1
3
2
4
???
???
???
???
1
1
1
1
hover over to see more info
Distribute points wisely
create an unique creature
Press space to continue
powerup
food
Generation 0
Avg Fitness
Best Stat
speed 3
health 5
children ...
press A to hide
predator
obstacle
mutation
Game
GAME OVER!
(Game over once all creatures died)
Game Over Screen
Bohan Li