15-112 TP Design Proposal

AndrewID : hyeleek
Name : Hyelee Kim (Hailey)

Project Description

"Artificial Ecosystem" is a complex self-running virtual 2D ecosystem where user can have interaction with the environment (spawning food, killing species, etc) and can explore the ecosystem. Each species will have basic behaviors of moving, eating, attacking, avoiding, reproducing, mutating, and dying.

Competitive Analysis

1. Past 15-112 Project "Evolve" byy Mina Nowroozi
   https://www.youtube.com/watch?v=14zm7Z8k3lo
2. Interactive video game "Spore" by EA
   https://www.youtube.com/watch?v=bTC8QPjI3YI
3. Interactive Art Installation "Archipelago" by Artificial Nature
   https://vimeo.com/120987833

	Evolve	Spore	Archipelago	TP
Genetic Algorithm	V			**V**
Customizing own species	V	V		
Analysis of ecosystem	V			**V**
Explorability		V	V	**V**
Different environments	V	V		
Game End	V	V		
Abstract Shapes / Aesthetics			V	**V**
User Actions (attack, movement)		V		
User Actions (Influence the ecosystem)			V	**V**

Above is the table of features seen from the reference projects. Three projects have different characteristics. "Evolve" is an application that simulates multiple generations of species that user customizes in one of the environments. With some probability of mutations happening to these group of species, the app shows how the population changes over generations. "Evolve" focuses on genetic algorithms and how it evolves. On the other hand "Spore" focuses on user-centered experience, as user customizes the initial species and play as that species throughout the

experience. User moves around, eats, attacks others, and with the food collected, can customize further with the energy collected from food. As “Spore” is a complex multi-player game, complexity and diversity of the world is impressive.

“Archipelago” has a different style compared to the other two. The most distinct difference is in art style, which is more abstract. User influences the environment and the whole ecosystem, but does not specifically creates or act as a species. User is more like an observer of the complex system, and having some influences on the ecosystem from time to time.

My project would be some combination of different elements from each project. It would be explorable ecosystem where user can have some influence. There will be different species in the environment, where each organism can eat, grow, attack, reproduce, and mutate. Each species would have visually-pleasing abstract shapes.

Structural Plan

Each class will be in different file.

```
# object World
# attribute Species[]
# attribute Foods[]
#
# object DNA
# attribute genes
#
# object Species
# attribute DNA
# attribute location
# attributes predators[]
# attributes prey[]
# attribute age
# function move
# function eat
# function attack
# function reproduce
#
# object A inherits from Species....
# object B inherits from Species....
# object C inherits from Species....
# ..... so on..... depending n design......
#
# object Food
# attribute location
# attribute energy
# attribute life
```

Algorithmic Plan

```
############################################################
# Algorithmic plan for Flock Behavior
############################################################
```

```
# First : Generate a set of animals
#       - different list of species
#
# Second : Make basic steering force template
#       - steering force : desired velocity - current velocity
#       - apply force
#
# Third : Make three different steering forces
#       - avoidance : try not to bump into each other (with same species)
#       - align : head in same direction as neighbor
#       - cohesion : staying in the group
#       - seek : seek for food and prey
############################################################
```

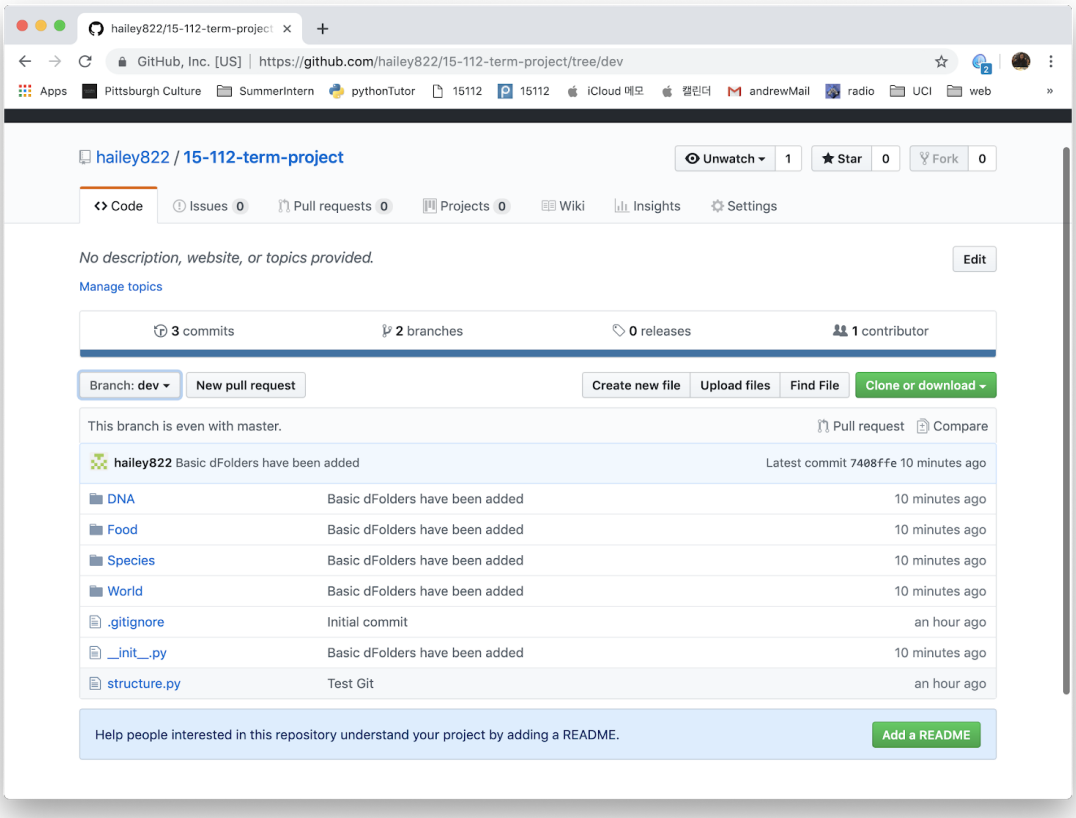
Timeline Plan [5 pts]: A timeline for when you intend to complete the major features of the project.

	date	content
TP1	04/09 Tue	- Design Proposal
	04/10 Wed	- Initialize all the files with classes
	04/11 Thurs	- Write basic big classes (dna, species, world, food)
	04/15 Mon	- Initialize World : Set up datas (set food, species) and Move/Draw
	04/16 Tues	- Finish up Design Proposal
TP2	04/17 Wed	- Write different types of inherited species
	04/18 Thurs	- Set predator/prey relationships
	04/19 Fri	- User Interaction Improvement : mouse interaction to create food generator with polygon
	04/20 Sat	- Simulate group behavior (flocking) : align, cohesion, avoidance
	04/21 Sun 04/22 Mon	- Meaningful mutation, inheritance, reproduction values
	04/23 Tues	- Adjusting the values to control the population
TP3	**04/24 Wed** **04/25 Thurs**	**- Improving visualization of elements** **(each species to have unique shape that looks more life-like)**
	04/26 Fri **04/27 Sat** **04/28 Sun**	**- Population control scenarios** **1) when total population is over certain level** **2) Spawn viruses and slow down the animals** **3) Viruses move around and attack animals with the highest**

		population **4) Viruses kills and becomes the most least populated animals** **5) When population goes down to certain level, the emergency mode ends**
	04/29 Mon **04/30 Tues**	**- Collision detection optimization** **- UI improvements**

Version Control Plan

https://github.com/hailey822/15-112-term-project



For the term project, I will use github to control versions. I have made two different remote branches “master”, and “dev” and made initial commit with folders and python files.

Module List

- Tkinter
- PIL (TBD)

Storyboard

- Attached in separate file

TP2 Update

After TP1 meeting, I have decided to add data visualization part to show changes in population and meaningful implication of it. At this point, I am thinking of including 1) population size 2) mutated population ratio for each of the species and as a population as a whole. Possibly, I will use line graph to do so.

I have started to use more **modules** since TP1 : **numpy** and **shapely.geometry (Polygon, Point)**

- **Numpy** : For vector calculation. However, as all the functionalities don't look very different from using normal list, so it is possible to move back to remove numpy module after TP2
- **Geometry** : This library is used for the convenience of generating random points within the bound of irregular polygon. Food generator and death traps are irregular polygons, so this library was particularly useful to get the bounds of the polygon and whether the point is in the irregular polygon.

TP3 Update

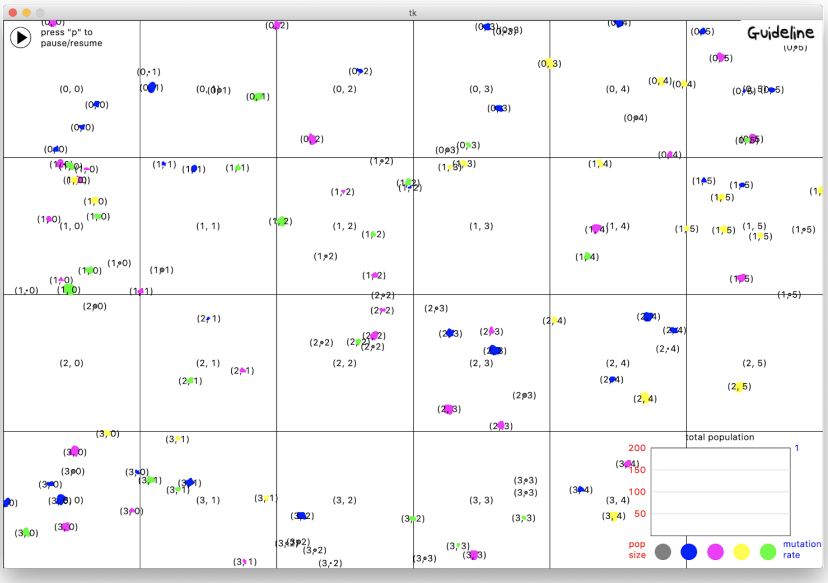
After TP2 meeting, there were two focuses.

1) Optimizing the collision detection algorithm
2) Deal with leg happening due to over-population and turn it into bring more complexity.

Both of the problems were complex.

< Optimized collision detection - Grid System>

In order to deal with collision detection, I came with Grid System. The whole screen runs with grid system which has several grids. Each grid holds "food" and "species", as what each grid contains are updated as these objects move. Each species will collision detection with only the food and other species that are in the same grid. Following image is a visualization of grid system and how each food and species belong to each grid.



```
############################################################
# Algorithmic plan for Grid System
############################################################
# First : Initialize Grid System
#       - Grid System holds grids divided into the size of the grid given.
#
# Second : Initialize what grid holds
#       - As the initial population is added
# - each food and species holds zone information
# - each zone also contains the information of entities belonging to the zone
#
# Third : Update grid system
#       - update grid information as the entities move
# - situations to consider
# - : eaten food, eaten species, new species born
# - : new food spawned from wetland
# - : new food born as species die
#
# Fourth : Utilize grid system
# - optimize collision detection by only checking entities in the zone
############################################################
```

< Virus : controlling over-population and leg >

Virus appears at certain point when the whole population exceeds certain level. All the species freezes while all the viruses are gone. Virus has target prey to eat the most populated species. Once that mission is accomplished, it turns into least-populated species. This emergency situation let the ecosystem to run and balance itself endlessly, adding complexity to the ecosystem.