15-112 Fundamental of Programming
Term Project Proposal

Qian Zhang
(qianz1)
Section: KK

11/22/2015

Main Idea:

In this project, I am planning to "recreate" a classical game: Angry Bird. However, the new created version will have a major different compared with the original game: it will be enable users to choose the game modes: normal mode, self design mode, and multiplayer mode (if possible). Besides, it will have a function called "recording", which enables users to record their games and then replay the "videos".

If the normal mode is chosen, the game will act just like the original one; in self design mode, an user can decide where to launch a bird and where the pigs are, how everything is arranged so that this game never becomes boring; and under the multiplayer mode (if possible), one player will control birds and other one will control the pigs so that when a bird is trying to shoot a pig, they can somehow dodge or defense.

Problems Identified:

To make the idea realized, there are mainly two problems to overcome:

1. To simulate the real physical collision when a bird hit a pig;
2. To enable the information exchanges between different computers.
3. To keep track of the mouse or keyboard events and re-run the code with these data when a user wants to replay the "video".

Modules Needed:

To solve the problems, I need to import three modules:

1. pygame:

"pygame" is a package gives a better performance than tkinter in Python when creating and running a game, it's faster, and images can be used more robustly.

In order to create a game in pygame, it would be necessary to create a screen and then start a while loop which keep track of the mouse and key events, redraw all corresponding activities as long as the game is not end, the structure is basically same as the barebone we used for tkinter.

2. **pymunk:**

"pymunk" is a package enables to simulation of the collisions among rigid bodies, it should works on the pygame environment.

With this package, I can set each bird and pig as a circle body, initiate it with the mass, moment, position, and then compute the impulse. And of course each of them will be an instance of the corresponding class, so that python can handle the collisions with the attributes I set to each of them.

3. **socket:**

"socket" is a built-in package inside the standard Python that enables the information exchanges among different computers so that it can enables the multiplayer mode.

To realize the multiplayer mode, what I need to do is to set up a server that acts as a media to send and receive information between clients: Once a player (let's call him/her A) made a move, this move will be send from his/her client to the server, and then the server send this message to the other player (let's call him/her B), upon the message is received, corresponding actions can be made on B's computer, and vice versa. So all actions from two players are synchronized. However, I found some

dramatic delay of the user operations and responses between different computers, this might have some negative influence on the game experience, I will decide if it can be applied in an actual game, so hopefully, there will be a multiplayer mode.

Notifications:

Please note that the self design mode is actually a backup plan for multiplayer mode because of the uncertainties of the lag caused by socket, the final deliverable will contain at least one of them. I will, of course, try to have both of them in my project if I can, the project is subject to changes according to the progress I made.