Mini Project 4: Cindy Sun and Logan Sweet

Project Overview

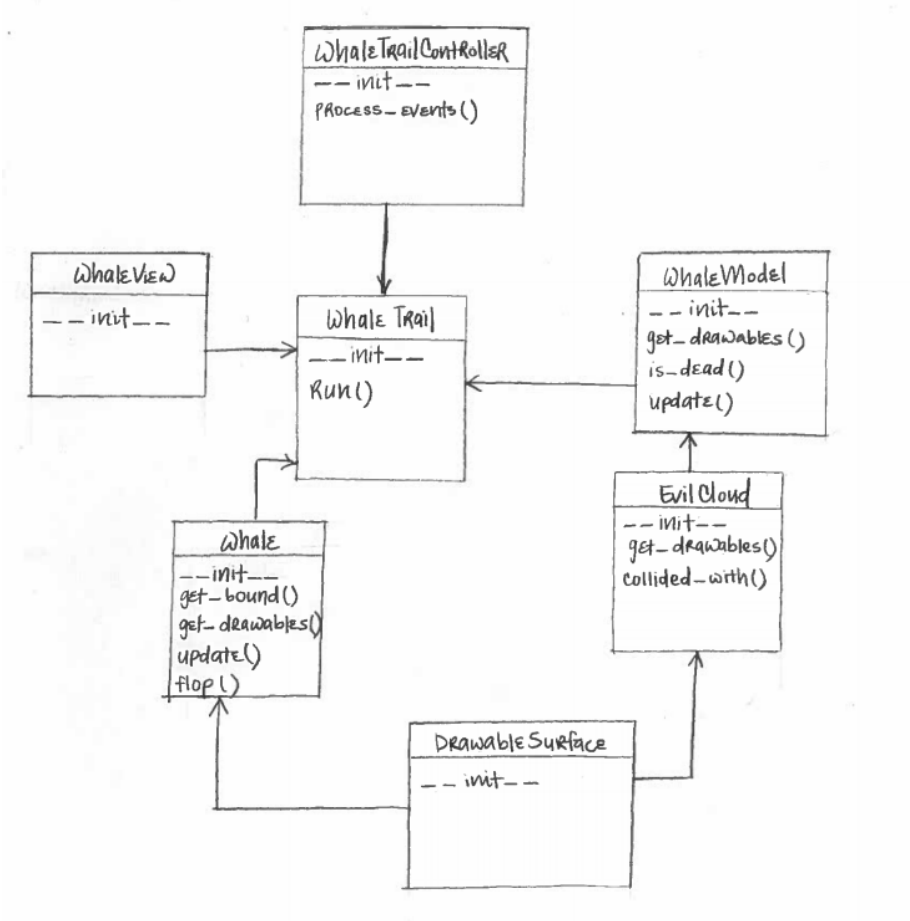
Our WhaleTrail is a python interpretation of FlappyBird with elements of WhaleTrail. The whale is controlled by the Space key and the objective of the game is to avoid the columns of evil clouds. Additionally, the background image is an image captured by the computer's camera at the beginning of the game in order to surprise the player.

Results

What we achieved in our game was closer to FlappyBird than WhaleTrail. The game we made is based on Paul Ruvolo's starter code for his FlappyBird clone. We intended to expand more on his code, but due to issues implementing our additions to the code within that pre-existing framework we did not get as far as we would have liked.

Our efforts resulted in a game that allows the player to navigate a sky full of clouds. The sky, in this game, is a candid image of the player. The image below shows our Whale Trail game.

Implementation



Our code is not structured in a hierarchical manner. Most of the classes have separate attributes and functions that are implemented into the WhaleTrail class, which holds the code to run the game. However, as shown in the UML diagram, certain class have a “HAS-A” relationship with others. For example, WhaleModel has references to EvilCloud and uses the EvilCloud object in its init function.

One important design decision we had to make was with openCV. We had several options in implementing openCV and discussed capturing an image as the background, using a continuous video image for the background, or changing the gameplay based on captured pixels. In making our decision to use a snapshot as the background, we had to consider how we had originally set up the code for the background when we built our initial code. Our starter code set an image as the background and scaled it to the window size; therefore, it made more sense to create an image file with openCV by taking a picture in the beginning and then setting that as the background.

Reflection

The final product of our game is a game flappybird-like gameplay with WhaleTrail graphics. Our initial goal was to create a gameplay more similar to WhaleTrail, which would have been achieved by altering the controls to continue the upwards acceleration if the player holds down the spacebar. We did not achieve this due to time constraints; we had trouble implementing the basic flabbybird functionality and chose to stop after we got that functioning. If we were to continue to develop our game, we would make the spacebar work like it does in WhaleTrail as well as possibly add a score counter. The best part of our game is the candid photo that is taken upon launch of the game. The quality of photo, due to the quality of installed webcams in our computers, ranges from mediocre to bad. A new player is then greeted with a (likely) ugly photo of themselves to compliment their gameplay experience.

Our plans for this project were very disorganized. If we had a better understanding of how classes wrapped before we began the project we would have achieved more of our initial goals. We did not have a solid understanding of classes at the beginning, leading us to make some mistakes in the structure of the code at the beginning. My partner and I chose to meet for longer periods of time less often, which helped us to stay focused, but got frustrating when we were stuck on a single issue for an hour or more. In future projects, we will try to do more initially planning and layout of the project, including a UML diagram beforehand.