Design Paradigms

Since we were creating the game from scratch, but most of us had played flappy bird before, we had some idea of which direction we wanted to take this project in. The project design is loosely based on Object Oriented Design as we used Python, which is an object oriented language.

We started with the Top-Down Functional Decomposition as we thought it would give us an overview of what we required for the game and how to implement it. We started by creating some classes that would hold the main functionality of the game, and created more classes as we thought they were required. We thought about the game and decided on a few basic classes such as game, bird, constants and pipes. Once we started building the game, we quickly saw that we needed a new class for the background and so we created the background class. Once the main functionality of the project was done, to add text to the screen, we created the class score. Therefore, we ended the game with 5 different classes, all holding some part of the functionality. The classes we ended up with were - game, bird, pipes, background and score - and one file named constants that keeps track of constants and variables needed in the game for the different classes.

We then moved on to using Component Level Design to design the individual classes. On discussing and brainstorming earlier, we had a broad idea of what we wanted each class to do and after learning pygame through websites and videos, we had a good understanding of what functions we needed to implement the game. We created various functions and variables in the individual classes to help us format the code and improve functionality in the game, and make it easier to scale the game for Project 4. A more detailed overview of this is given in the UML Modeling Diagram.