**Tamagotchi – Final Report**

As a precautionary measure, the Tamagotchi game was a change of heart from the initial plan of the Fish Tank project. Due to a careless blunder, I had all of my progress from my Fish Tank project wiped, if not most of the crucial parts of the project, due to the “blue screen of death”. This rerolled my code back to a previous save, which was near the beginning of when I started the project. I then chose to swap over to the Tamagotchi project since reworking all of the systems in the Fish Tank project, such as the currency system, all of the tetra and other animals I had planned to put into the fish tank, were erased. At the time, I did not have the heart to go back and recode all of the possible variables and deal with the errors that may arise.

However, even in the Tamagotchi project, I still had differing ideas while I was coding. For example, I initially wanted to give the user the option of naming their pet. This meant that I would be able to either have the name set to “Your pet” and have the code run off of that. However, I decided against it and thought that the pets should at least be given a name, hoping this would bond the user to the pet. Another differing idea would be to have multiple pets in the Tamagotchi. This came from my original Fish Tank idea, as some people have multiple pets and may like the option to have multiple pets in a Tamagotchi. However, I also scrapped this idea as the Tamagotchi project swap was last minute and I was afraid I did not have the time to add this idea in time of the deadline.

Transitioning to the construction of the Tamagotchi program, the program consists of five files, including the main class. These classes are the Pet class, Dog class, Cat class, and Hamster class. The Pet class is the super class, and this class holds the Dog, Cat, and Hamster class as subclasses. The main file holds the menu, which provides the user with a starter menu to either load from a save or create a new save by selecting a new animal and providing it with a name, and a game loop, which loops around options allowing the user to play, feed, talk, put their animal to bed, pass time, save their game, and exit the program. As long as the user does not exit this loop, they may interact with their animal however they wish. However, the condition that the animal must have their hunger level below 100 is crucial, as this is the end game condition. Once the hunger level reaches 100, the animal will die, and the user will lose the game. However, this may be bypassed, as the animal dying does not immediately save the animal’s data. The user can take advantage of this by reloading a previous save and taking measures to prevent the animal from getting closer to death.   
  
Link to Video: <https://www.youtube.com/watch?v=YqCzrM5F9S0&ab_channel=WilliamNgo>