QUANG NGUYEN – PROJECT REPORT

Project description: For the final project of the CS123 course, I made a cat picture memory matching game.

Code structure: The game folder includes three python files and two subfolders (the cat images subfolder and the matching prompting image subfolder). The three python coding files are app.py, game_config.py, and animal.py.

The file app.py is the main folder running the game. It includes:

- The find_index function for working with the mouse to find the index of the images;
- A series of called methods to display the screen and the images;
- A while statement for working with the the computer keyboard and mouse;
- The for statement for flipping the images when they are clicked;
- The if statement for matching two identical images.

The file game_config.py includes all the configuration of the game, from the size of the screen, size of the tiles, and number of tiles and images. This file is linked with the app.py file, so the programmer can change the configuration of the game on game_config.py

The file animal.py is responsible for working with the Animal Class. It defines the characteristics of the animal class.

Challenges: Listing the characteristics of the Animal class; working with mouse and keyboard; figuring out the calculation for the index of the image; getting used to the methods in pygame. Fortunately, thanks to pygame, the code is much simpler and shorter than writing the game with tkinter.

Known issues: The game configuration is not freely changed. It must follow some underwritten rules, for example: the total number of tiles must equal to the double amount of the animal images, the size of the screen must fit the total number of tiles.