Arjun Bhan

Professor Chung

COM 110 Final Project

Due 12/17/18

Memory Game Project Problem Specification and Design Document

This program is a memory game. The user’s goal is to click on two cards and match their numbers. After clicking on the introduction screen, the user is tasked to select a difficulty level which determines the number of pairs that need to be matched. Matching a pair awards 100 points; whereas, making an incorrect match takes away 25 points. The game ends once all cards are successfully flipped. Once the game ends, the user has the option to either quit the program or play again.

Program functions:

* Basic Memory Game Functions / Components.
  + Try to get matches
  + Has 3 difficulty levels
  + Record number of attempts
  + Point gained if two cards match, otherwise points lost
* Main function
  + Calls in other functions into the program.
  + Calls the intro function to make a introduction to explain to the user how the code works.
  + Calls on the grid function to make cards and will tell the function which difficulty the user chose. The difficulty will decide on how many cards are in the game. The difficulty decides how many matches are needed to win a game. There is a variable that changes with difficulty. This variable counts how many matches have happened in a game.
  + The code calculates what type of card the user selected by using the getRank function from the grid class.
  + A point system is used in the game. If a match is found, the point system adds 100 points. If the two flipped cards are not a match, the point system takes away 25 points. The point system was created by changing the value of a variable that held the points and redrawing the points when two clicks occurred.
  + In order to prevent a card from repeatedly being clicked for points, if two cards have the same x and y coordinates that they will subtract the points that were given by the cards having the same rank.
  + The code is able to restart itself by recursively calling on the main function.
  + There are also two exit buttons in the game. One is for the difficulty select menu and the other one is for the game,
* Additional functions
  + Colsiz function
    - This function sets the color and size for the text used in the program.
    - Although this function is very small, changing the color and size of text is used throughout a lot of the program.
  + Option function
    - This function resets the coordinates for the GUI.
    - This function also uses the grid class to change the difficulties of the code.
  + Intro function
    - This function is used to introduce the code. This function is important as it allows the user to know what the code does. The reason why this code is a function is to better organize the code, despite only being called once.
* Grid class
  + Initializer function
    - This function allows for the random assortment of cards.
    - This function makes sure that there will always be two matching cards in the deck.
    - This function sets up points for the cards to be placed.
    - This function makes sure that the matching card suits will be either a spade or a heart.
    - This function decides the height and width of the cards.
  + drawGrid function
    - This class generates the cards on the screen. The card positions are based on the points in the list of centers.
* matchObject
  + Flip function
    - This code uses Boolean values to change the image of the card.
    - If the function returns True, the function flips up the card. If the code returns False, the function turns the card face down.
  + Activate function
    - Tells the clicked function that the card should be able to be clicked since it is active. The function does this by returning true to the clicked function.
  + Deactivate function
    - Tells the clicked function that the card should not be able to be clicked. The function does this by returning false to the clicked function.
  + drawObj function
    - This function draws the card images.
  + Clicked function
    - This function will give the user the ability to be able to click a card by returning the x coordinate and y coordinates of the card.
    - By using Boolean variables to check whether the card is active or not this function is able to make the deactivate and activate function work. If the card is activated then it can be clicked otherwise it won't.
  + setUpVals function
    - This function sets the variable for suit of a code and the rank of a code. This function sets up a variable that will be used to hold the card images. It also lets the variable be used throughout the code.
  + Initializer function
    - This code makes sure that all the cards are naturally active and face down. It does this through the use of Boolean variables which have set the cards being face up as False and the card being active as True.
    - This code also sets up variables that will be used later in the code.
* buttonclass
  + This is the button class we made during class.
* Graphics
  + Zelle graphics
* Time
  + Import sleep in order to view the flipped cards for a second before they flip back over.

User inputs:

* Program opens and gives user instructions
  + Choose between three difficulties by selecting a button
    - Easy (8 cards), medium (16 cards), and hard (24 cards)
* The game begins
  + Create an quit button into the GUI, but the user has to complete the game before quitting.
  + Call the Grid class
    - Uses the grid class to create a grid of cards that are made by the matchObject class
      * The grid class creates a grid of buttons.
        + The buttons are changed into flipped over cards and their positions are randomized
      * The user clicks on two different cards
        + They flip over as they are clicked
        + The ability to flip a card will be decided by Boolean variables.

If they match, both card buttons remain flipped up in the window.

100 points are also added to the score

If the do NOT match, both card buttons are flipped back over to their original positions.

25 points are removed from the score

The game continues until all cards are matched.

Once the game is concluded the code will give the user two buttons one will close the game and the other will restart the game by recursively calling main into itself.

Cool and interesting modifications that I would want to make to a future version include adding:

* A High Score List.
* Background Music.
* Sound effects that play depending if you make a correct / incorrect match.
* The ability to have games with more than two suits of cards.
* A more detailed card flipping animation.
* A restart button that does not recursively call main().