

Mehmet Doğan **2012400189**

Mahmut Uğur Taş **2011104090**

Card Match Game

The game is an image-based card matching game which let people play guess game among famous people photos. To start the game, you need to click “New Game” and gameBoard is loaded. Then, first player starts to game, it clicks one item and time starts to count and when second click arrives, time stops and time between start&stop added to user total time. The player who is active, is indicated with bold and red font. It does not matter if clicked items are equal or not, turn changes on each move and next player’s turn is active. If the selected items match, the score of user who clicked items, is incremented by one. Matched cards are disabled and they are not clickable any more and the images’ visibility is decreased. If they are not matched the cards are flipped back after showing the image for 2 seconds.

How Game is developed ?

The project is developed with 2 minor and 2 major classes.

Minor Classes

- We introduce a class named “item” which will be used for storing images url and how many times they are used while distributing on gameboard.
- We introduce a class named “player” to store ID, playTime and totalScore.

Major Classes

- We introduce a class named “thegame” which let us initialize game, checking cards(matched or not) and update user scores. In the initialization method, we add 2 players to user vectors and creates a vector which stores the items. By using this vector, the items in the list is distributed onto gameBoard. CheckMatch is possibly the most critical method, it gets time and clicked Items’ position, convert the position to matching position on gameBoard array. Then, if the clicked items match user Score is incremented and returns true. Whether it is matched or not, time is added to user play time and turn changes.
- We introduce a class named “cardmatchgame”, it includes all necessary graphic based objects that are explained below. The item clicked methods sends parameter to checkMatch method in “thegame” instance. Also, slot and signal functions are explained below.

QT Programming Explanation

For GUI part, we have 10 labels and 33 buttons. First we will explain layout structure, we have a vertical layout which keeps new game, previous game, and next game buttons. Another vertical layout keeps game no label and highest score label. There is a horizontal layout which contains these two layouts. We used a grid layout which contains score board part of game. Score board has 8 labels. Those are player names, score and time labels of players. Another horizontal layout keeps this grid layout and other horizontal layout which is at top right. Finally we create a widget to make top layout stable at top. Our bottom layout is just a grid layout which contains 30 buttons. Then we used a vertical layout to unit this layout and widget. We have 4 functions at gui part.

- First one is newGameClicked, when new game button is clicked, it calls this function and create a “the game” instance and refresh everything on window, then creates a signal mapper to find clicked button on grid layout. Finally this function create game buttons, and set current player.
- Our second function is prevGameClicked. This is called when previous game button is clicked and at every click it gets previous game’s data and updates score board, also activates next button. If it reach to first game this function also disables previous game button.
- Third function nextGameClicked is called when next game button is clicked. It does the same operations of prevGameClicked function. Only difference is it doesn’t move backward, Its operation is in forward direction. If score board shows first game and next button is clicked, this function handles activation of previous button and if last game is reached then disables next button.
- We have another function to make required operations when a card(game) button is clicked. This is itemClicked. First we get position of clicked button then take corresponding image from game board of current “the game” and sets clicked button image as this taken image from game board. If number of clicked buttons is two then function checks whether they match or not, if they match function updates score and disables these buttons otherwise they turn initial image, in each case current player is updated and current player label’s color is set as red.

PS: You may need to set build directory to project’s current directory since image paths are relative.