

GROUP MEMBERS: MISRA YAVUZ, 2016400135 ALGI KANAR, 20161400123

PROBLEM DESCRIPTION

Problem: Writing a memory game in qt format which will give the information of how many tries and pairs was done simultaneously while playing. The game needs to be a single person game which uses clickable buttons as cards and alphabetical letters as the cards' faces in a random nature. After each try the cards must face down or disappear according to whether getting a successful pair or not. It also needs to have a reset button to restart the game with new random set of cards whenever player wants a new game.

PROBLEM SOLUTION

After opening a project in Qt, the mainly intervened parts are the ui and cpp file. The ui part is the visible design part of the game. We used QPushButton and QLineEdit to perform a memory card game visual. Then, we change the names of these buttons and lines in order to easily directly access them in cpp files. We also put them in certain layouts to obtain a tidy look.

For the cpp part, at first we defined some variables like counter, pairs and tries to keep track of the current game status. counter is changed when a button is pressed, pairs counts the matched buttons and tries basically keeps the number of tries. Since the game size is fixed at 4x6, we defined an array sized to keep all push buttons together and treat them in for loops later. To accomplish randomness, we hold a vector named myvector to keep all letters and shuffle them when a new game starts or whenever the reset button is pushed.

In order to proceed accordingly to user's actions, we implemented 2 functions: ButtonPressed and ResetPressed.

ButtonPressed slot is connected to all QPushButton which are basically the cards of the game(excluding the Reset). Counter keeps track of the opened cards. 0 means no card is opened, 1 means 1 card is opened and 2 means 2 cards are opened. We need the counter to be 2 to prevent another pressing at the time of qWait(1000). In fact, user can press another button but that signal won't be used until qWait(1000) ends and we update pairs and tries. After all updating, we change the counter to 0 to make it available for next readings. When counter is 0; we set the pressed button's text to its random value by reaching to the myvector with the index of the button. When counter is 1, we repeat the same steps for the second card. Then, if a matching happens we hide those matched buttons and increment pairs and tries. If no matching happens, we simply set the buttons' text to X and increment only the tries. By letting and checking the counter, we proceed through the game.

ResetPressed slot is connected to only Reset which is also a QPushButton. This slot starts a new game in a way. If Reset is pressed, we shuffle myvector again to accomplish randomness. Then we call show() and setText("X") on all pushButtons through the numButtons array(excluding Reset). counter, tries and pairs are all set to 0 and tries and pairs are shown on the window.