

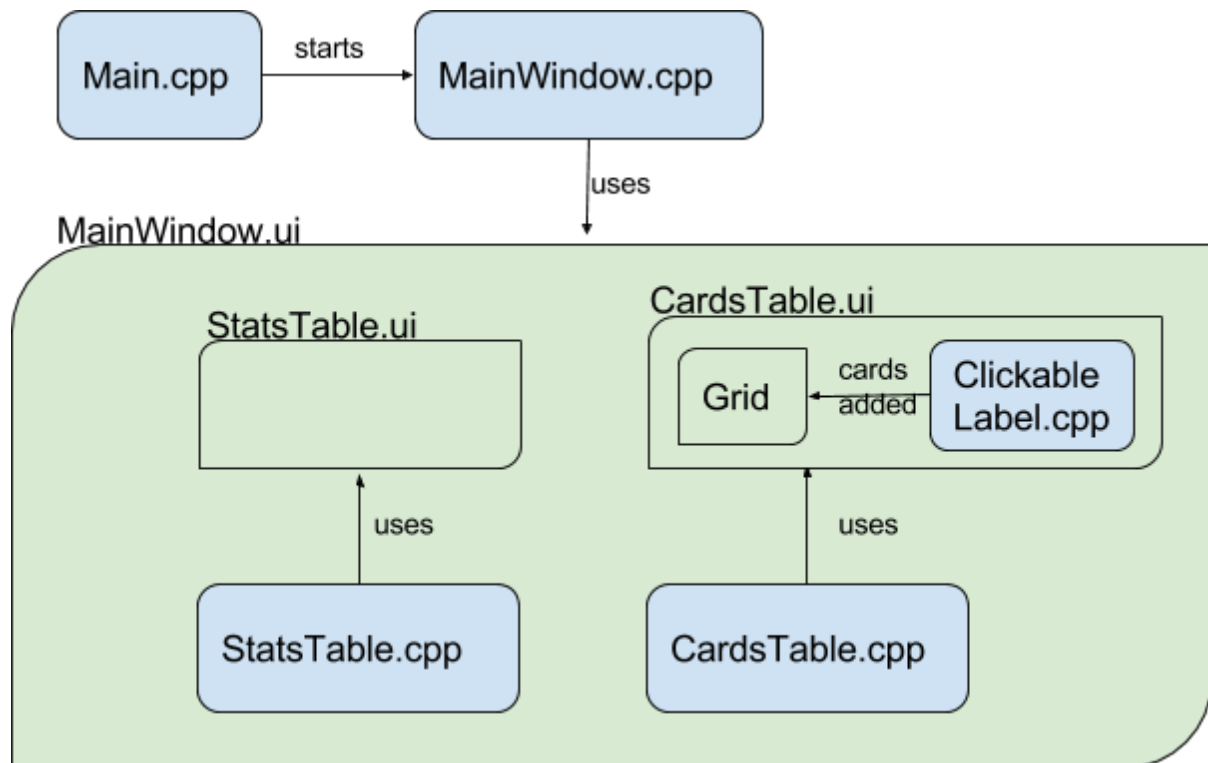
QT PROJECT REPORT

Student Names: Irmak Kavasoglu, Ali Kaan Gündüz

Student IDs: 2013400090, 2013400060

The main purpose of this project was to implement a matching cards game for two players. We have used Qt Creator to implement our project.

General Structure of the Project



In the illustration above, blue color represents .cpp files and green color represents the .ui files attached to them. The Main.cpp file is responsible for starting the MainWindow, which has all the game components in it. MainWindow contains the general ui of the game, and has slots for the components which will change from game to game such as game scores and cards. StatsTable has 4 QTextBrowsers; score and time of each player. CardsTable has a QGridLayout in it, which is later programmatically filled by the CardsTable.cpp class with ClickableLabel objects.

The following pages explain each class in detail.

Detailed Explanation: MainWindow

Ui:



As mentioned before, MainWindow class controls the general ui of the game. Above, there is a screenshot of the MainWindow.ui. This class has:

- All the static texts needed for game, as QTextBrowser objects.
- Variables which are not dependent to singular games but the whole program, again as QTextBrowser objects.
- New Game, Prev/Next Game buttons which switch between games, as QPushButton.
- Slots for game specific ui widgets, which are stats and cards. These slots are QStackedWidget objects, they can store all the games simultaneously.

Cpp:

The MainWindow class has the following class variables:

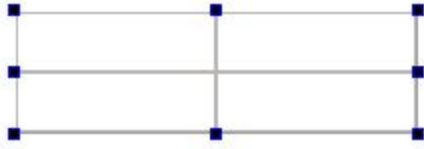
- *ui*: Holds the ui object.
- *gameNo*: Hold the id of the current game.
- *cardImages*: Holds a QPixmap array for the card images.
- *statsTableVector*: Holds the StatsTable objects for future use.
- *cardsTableVector*: Holds the CardsTable objects for future use.

And the following functions:

- *initialize*: Initializes the ui variables (gameNo etc) and loads the images to cardImages array.
- *newGame*: Starts a new game by creating a StatsTable and CardsTable, adding them to the table vectors, setting the widget stacks to the current ones and updating ui variables. This function also handles the shuffling of the cards each round. It is associated with the new game button in the ui.
- *prevGame/nextGame*: They both update gameNo and set the widget stacks of stats and cards to the desired game. They are associated with the prev/next buttons in the ui.

Detailed Explanation: StatsTable

Ui:



The ui of the StatsTable is quite simple. It holds 4 QTextBrowser objects; scores and times of the players.

Cpp:

StatsTable class has the following class variables:

- p1Time, p2Time, p1Score, p2Score integers.
- ui object.
- highest score, received pointer from the MainWindow.

And the following functions:

- *tickP1Time, tickP2Time*: increases the time of the given player by one and updates the ui.
- *increaseScore*: increases the score of the given player by one and checks if it is now bigger than the highest score. If so, updates the highscore as well.

Detailed Explanation: ClickableLabel

Cpp:

This class is a very small and simple class. Normally, QLabel's do not have a clicked() signal and if we want to use clickable images, we had to solve this problem somehow. This class inherits QLabel class, and emits a clicked() signal whenever a mousePressEvent occurs on this object.

Detailed Explanation: CardsTable

Ui:

The ui only has one QGridLayout in it.

Cpp:

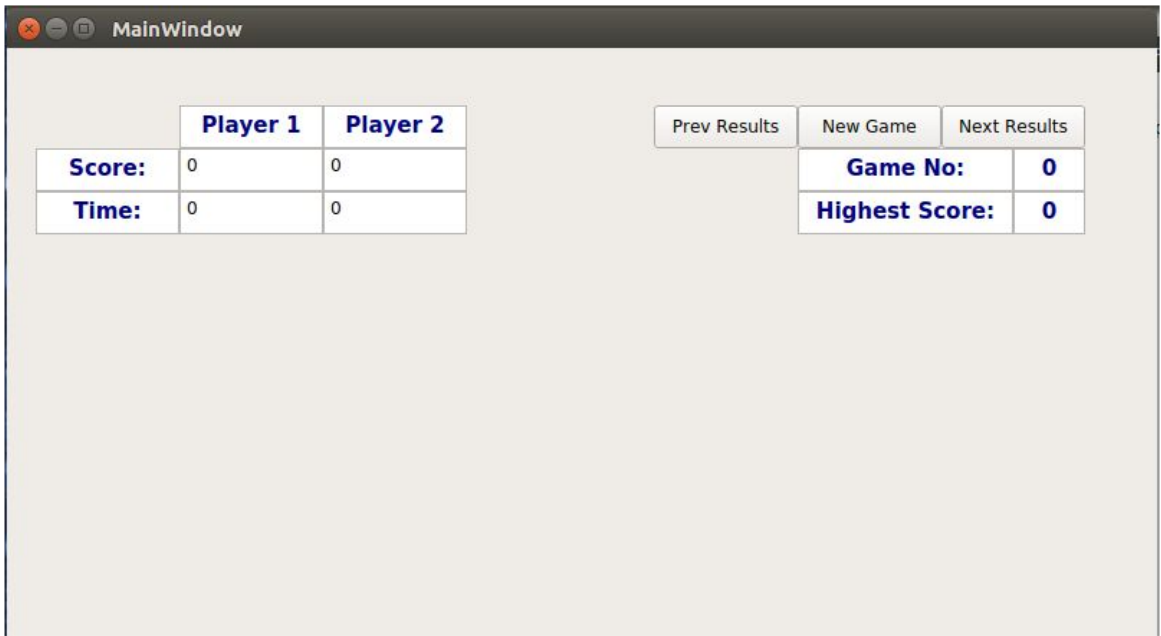
CardsTable class has the following class variables:

- *ui*: ui object
- *cardsBelow*: a pixmap array to keep the images given by MainWindow. When a card is opened, it's pixmap is replaced by these values.
- *cardsArray*: a clickable label array. It holds the references to the ui objects of the card. The card pixmap's are changed by using these references.
- *turn*: an integer, keeping track of who is the current player.
- *timer*: a QTimer object, started when the game is created and ended when all the cards are opened.
- *openCards*: a vector keeping references of actively open cards.
- *pairedCards*: a vector keeping references of paired cards.

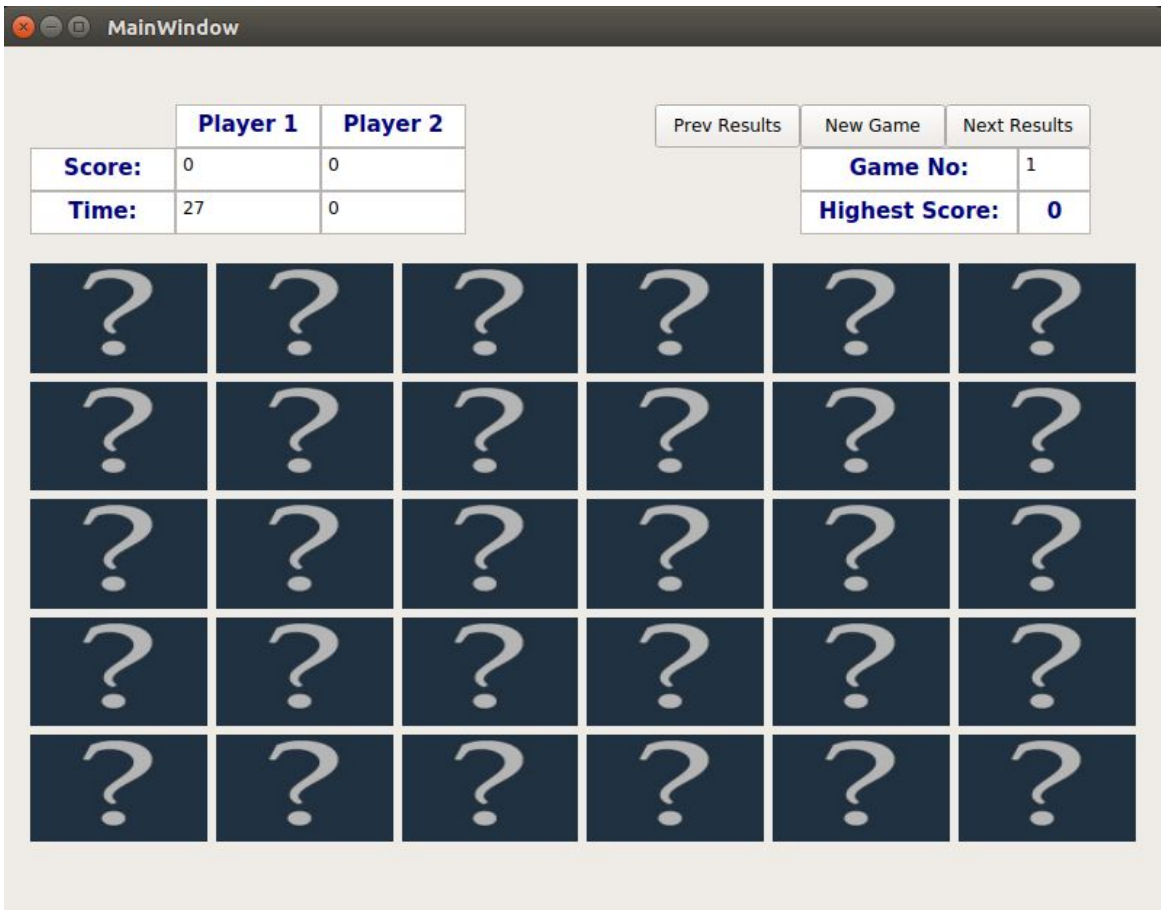
And the following functions:

- *initialize*: adds ClickableLabel objects to the card grid. Sets all their pixmaps to question mark images. Gives the first turn to player 1 and starts the timer.
- *setCardsBelow*: a public function for the MainWindow to set randomly arranged card images for the cards. It fills cardsBelow array with given pixmaps.
- *timerTick*: associated with the QTimer and triggered every 1000 ms. It calls tickPxTime function of the StatsTable, where x is the turn.
- *closeAllCards*: it sets the pixmaps of the OpenCards vector members to question mark images and removes them from the vector.
- *imageClicked*: associated with all the ClickableLabel objects. When a card is clicked, this function is called. It first checks if there are 2 or more open cards, if so, closes them all. Next, it checks if the clicked card is already opened or already paired; if so, does nothing, if not opens the card. Next, it checks if the newly opened card has any matches. If there is a match, it 'erases' the cards by setting their pixmaps to empty background images and adding them to pairedCards vector. It then calls the increaseScore function for the current player. Next, it checks if all the pairs are opened; if yes, it stops the timer. If there is no match, it gives the turn to the other player, and returns.

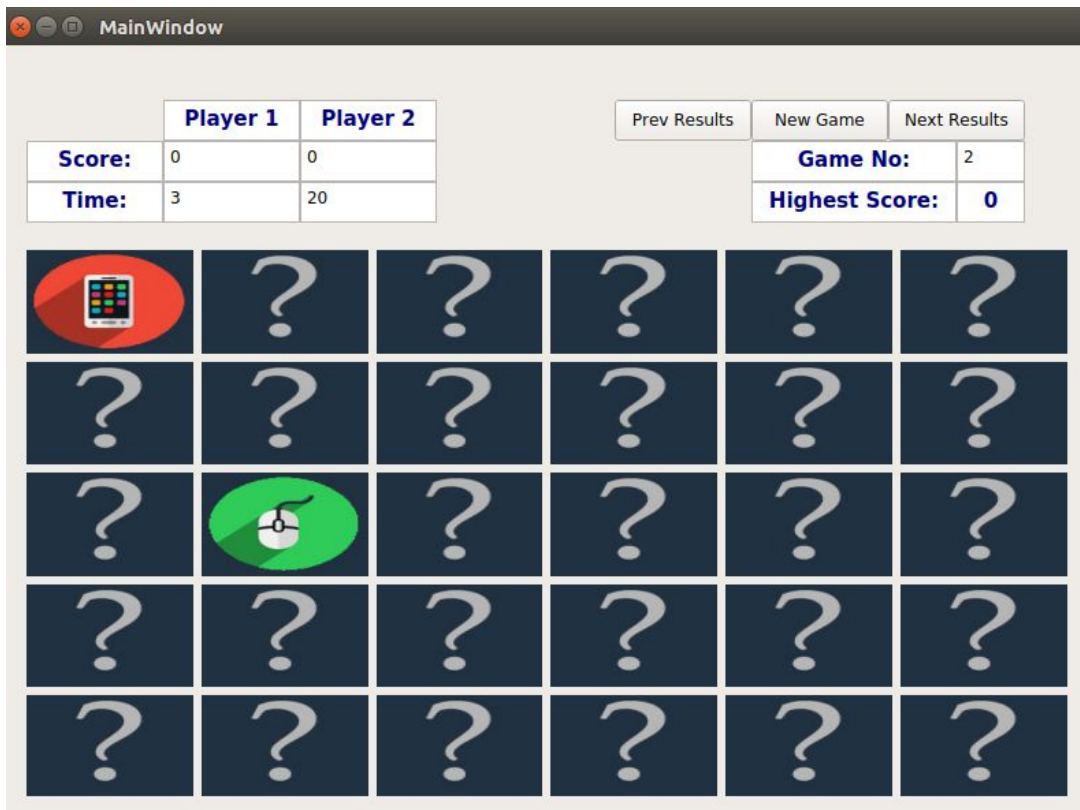
Final Look and ScreenShots:



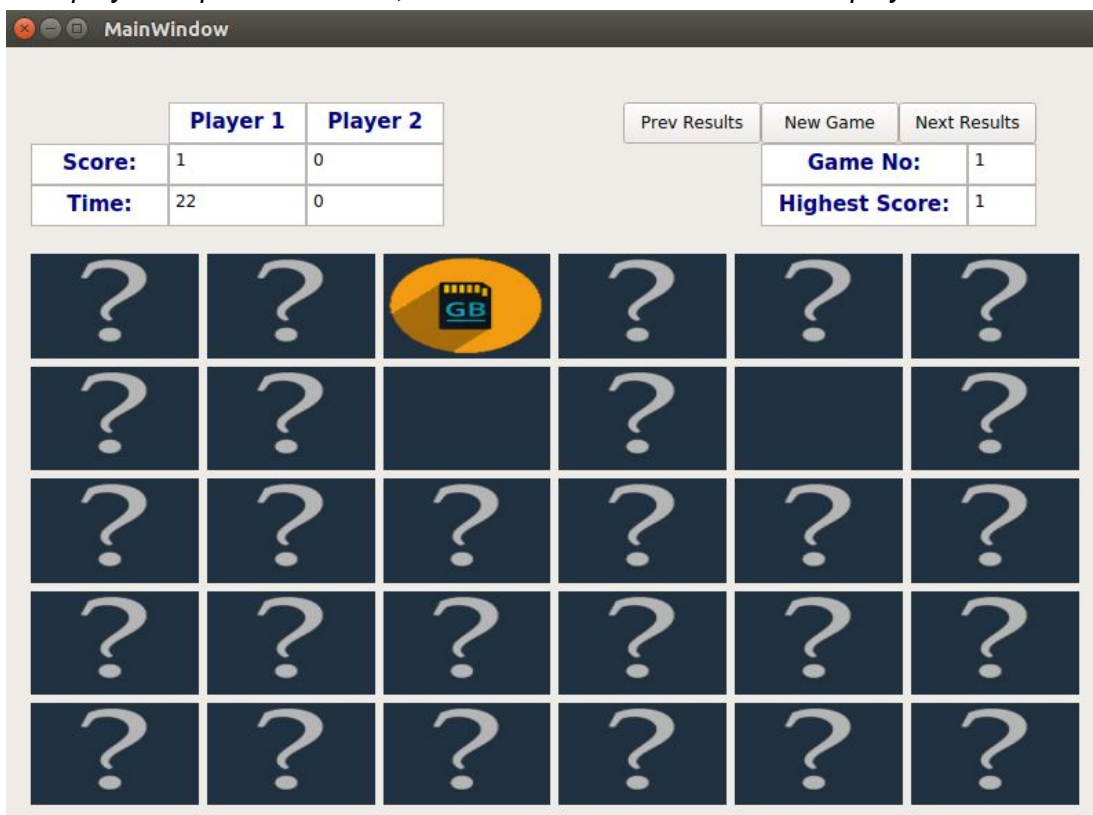
Initially, all the scores and times are set to zero. The timer waits for a new game to be started.



When a new game is started, time ticks for the player 1.



After player 1 opens two cards, the time then ticks for the second player.



If a player finds a match, the match is removed from card grid but the turn is still player 1's. High score is also updated after every pair match.