

CMPE 230 SYSTEMS PROGRAMMING

CARD MATCH GAME PROJECT DOCUMENTATION

This documentation expresses my approach to implement card matching game, using QT Creator.

General Structure / Approach

I implemented different elements in the game using different classes for clarity purposes. Most of the classes inherit a class of QT. I used Qt Creator 4.12.2 which is Based on **Qt 5.14.2** (Clang 10.0 (Apple), 64 bit).

The Card Class

The class inherits from QPushButton. I implemented fields of the class as:

QString cardText (to hold the text written on the card -every distinct word is written on exactly 2 cards-),

bool paired (to hold whether the card has been matched with its other pair),

bool open (to hold whether the card has been opened by the user at a specific time).

Slots belong to this class are as following:

void openCard(): Changes status as opened, makes the text visible to the player, disables clicking, changes color for a better graphical user interface. **newCardOpened()** signal is given.

void closeCard(): Does reversed operations of openCard() slot

void enableCard(): connects clicked() signal of the card to the openCard() slot

The myGrid Class

The class inherits from QGridLayout. I implemented fields of the class as:

QTimer timer* (to hold a pointer to the timer object),

int matchedpairs (to hold current the score),

int opencards (to hold number of cards opened unpaired by the user at a specific time).

Slots belong to this class are as following:

void endgame(): when timer exceeds the limit, stops the timer and disables the cards, gives a message

void checkPaired(): check for pairs in the game layout.

void createNew(): construct a new game layout with different positioning of cards, return all states to initial.

The scorer Class

The class inherits from QObject. I implemented fields of the class as:

*QLabel *label* (to show the current score),

int counter (to hold current the score),

Slots belong to this class are as following:

void increaseScore(): increases the score by 1 when the user finds a pair

void reset(): resets the score when a new game starts.

The MyTimer Class

The class inherits from QObject. I implemented fields of the class as:

*QTimer *timer* (to do the timer actions),

*QLabel *label* (to show the current time),

int counter (to hold current the time),

Slots belong to this class are as following:

void MyTimerSlot(): increases the time by 1 and stops the game if time limit is exceeded.

void reset(): resets the timer when a new game starts.

The mainWindow

All the explained different classes are gathered together in this cpp file. Necessary connections of slots/signals are handled. New Game and Quit buttons are created. 2 buttons, score and time labels and game grid are gathered together in a grid layout. Design choices are specified.

Designer's Choices

Lots of points can be considered as designer's choice in this project. These are specified below:

- 1) The words on the 30 cards are 15 characters of the TV show The Office.
- 2) For a better user experience, different card states have different colors. i.e:
dark red= closed cards with "?" text,
dark gray = opened cards with the actual text visible,
dark green = matched cards with actual text visible.
- 3) A card that is opened cannot be clicked again to close it, which means the player has to open 2 distinct cards at every attempt. I believe opening and closing a card right after is against the logic of the "memory" game.
- 4) When 2 cards are opened, if the cards do not match, both of the cards are kept open for 1 second for the user to be able to memorize. During this time period, opening another card is prohibited by disconnecting the clicked fields of the cards and openCard() slot. If the cards do match, the player can open another card right 0.1 second after (to avoid any possible errors).
- 5) The main window of the game has been designed to fit on a grid layout. The size of the main window is restricted with minimum/maximum limits. This is done to keep the game in a stable appearance.

Conclusion

New Game and Quit buttons work properly. Cards are shuffled at each new game. The user wins if he/she is able to match all 15 pairs within 180 seconds. Win/lose messages are shown in corresponding scenarios. Detailed information about the procedures of classes and functions can be found in the comments of the code. In conclusion, I have properly implemented a card matching game, using QT Creator.