SEVDE SARIKAYA - 2017400081

QT GAME DESCRIPTON

PROJECT CLASSES

1. colorbutton.cpp / colorbutton.h
2. mygridlayout.cpp / mygridlayout.h
3. mytimer.cpp / mytimer.h
4. newgame.cpp / newgame.h
5. MainW.cpp

Problem: Implementing a Card Match Game by using QT.

QT Creator 4.12.3(Community) is used to implement and run the code.

Creating project steps:

1-Other Project

2-Empty qmake Project

3-Desktop Qt 5.14.2 MinGW 64-bit kit

(This kit is used to implement)

Mytimer.cpp:

It is a basic timer class, it creates a timer and measures the time. By MyTimerSlot, it checks if it takes 180 secs or not. If it is, it stops the timer.

#include <QTimer> // for timer object

#include <QLabel> //for timer label

#include <QMessageBox> //to give message box when time out

Colorbutton.cpp

It creates a button, colored with “cyan”. It takes a word which is an animal name. We can also call this buttons as cards. It has a show() slot to show the name written in the card.

#include <QPushButton> //to create push button

#include <QPalette> // to give color to the button

Mygridlayout.cpp

It takes a QTimer object to measure the time and set a label on it. It also has a “score” label which keeps the total matched cards. The timer is connected with dis() slot in mygridlayout to disable the buttons after time out.

It has 2 other slots, too. Match() slot checks if the open cards have the same animal name or not. If they do, sets their name as empty. Otherwise, sets their names as question mark.

Check() slot checks if the game is finished. Travels all layout and look for buttons if they matched or not.

If there is a matched card, it is disabled here. If all cards are matched, it gives a QMessageBox as “You won!”

#include <QGridLayout> //to create grid layout

#include <QTimer> //for timer object

#include <QLabel> //for score label

#include <QMessageBox> //to give message when the user wins

#include <QEventLoop> //to be able to wait 500 ms after 2 cards are opened.

#include "colorbutton.h" //for buttons

The game is created inside the newgame.cpp:

it creates a push button called as “New Game” and calls newgame() slot to start the game. If new game button is clicked, it creates a new game.

Restart() slot is called when the object is created and also when New Game button is clicked. It creates widget, vertical box layout, horizontal box layout, timer and grid layout objects.

Horizontal box layout includes timer and score labels, New Game and Quit buttons.

Vertical box layout includes the horizontal layout we mentioned before and the cards (colorbutton objects)

We add this objects as widgets.

My vertical box layout includes 30 card objects. Randomly chooses an animal name and places it on the card. Then connects this buttons with show(), match(),check() slots (we mentioned before)

Then all labels and layouts are combined.

The class includes:

#include <QApplication> //for app

#include <QWidget> //for widgets

#include <QVBoxLayout> //for layout

#include <QSpacerItem> //for spacer item

#include <QList> //for list

#include <QDateTime> //to set random

#include "mytimer.h"

#include "colorbutton.h"

#include "mygridlayout.h"

In Main, it basically creates a newgame object(it runs the game) and executes the application.