

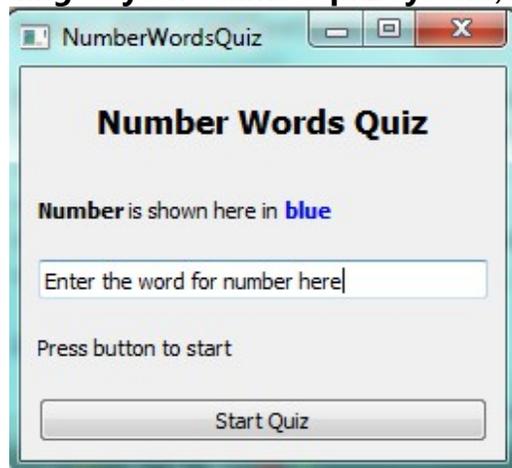
SFDV4001 OOP with C++ and UI

Part 2 of the Quiz System project – implementing the user interface

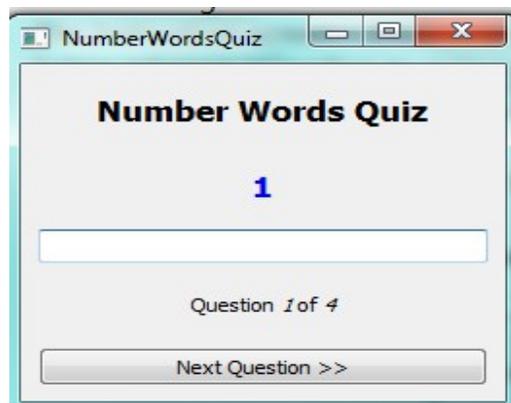
In this part of the project you will use QT to build the GUI for the project you have done in part 1. Instead of STL containers you will use QT containers – *QMap*, *QList* and others as needed.

The following features are needed for the GUI.

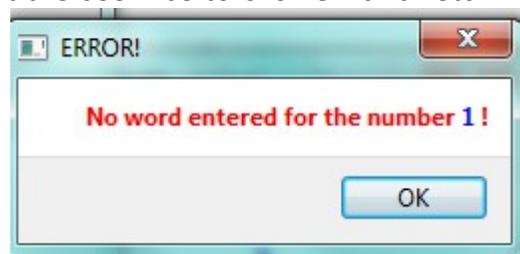
1. The Instructions screen – this screen appears before the quiz starts and should look as shown below (**change according to your team's quiz system**)



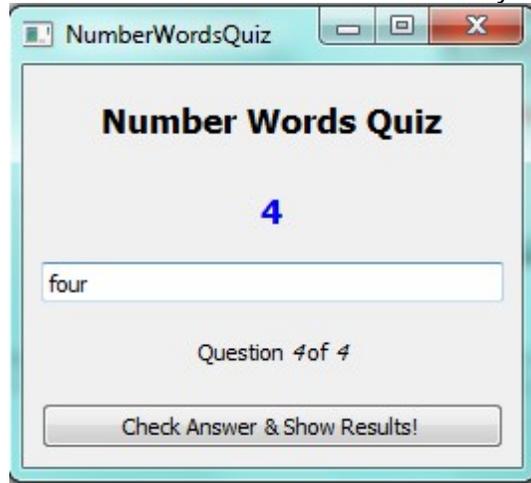
2. The quiz screen showing the question, text field (*QLineEdit* in QT) for the answer and the Next Question button



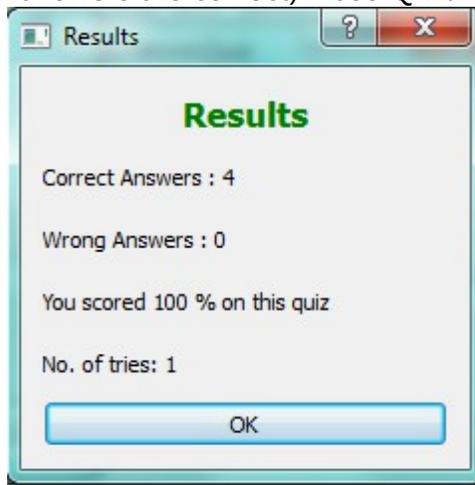
3. Show an error message using a *QMessageBox* if the user does not attempt a question. This should be modal so that the user has to click OK and return to the quiz screen.



4. Screen for the last question – inform the user about results by changing the button text.



5. Results dialog box (when all answers are correct) – use *QDialog* with layout

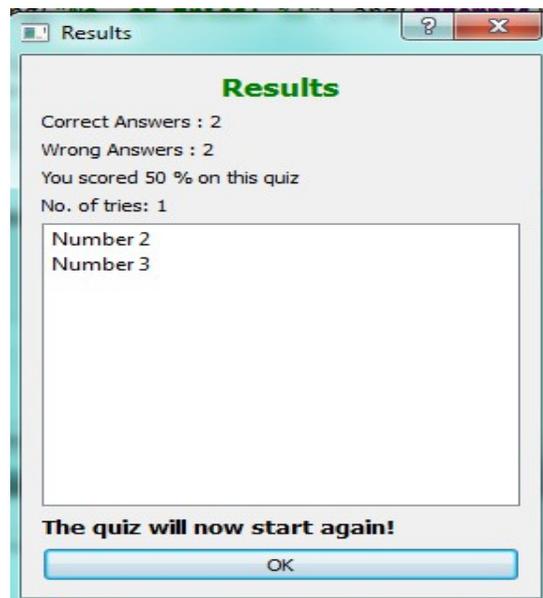


6. When the user clicks OK both the results dialog and the main window (showing the questions) should be closed since the quiz will not repeat.

7. When the user has at least one wrong answer:

Shows results as in screenshot and on the right and

8. restart the quiz with questions now appearing in some random order – use *QTime* together with *qsrand()* and *qrand()* functions to achieve this



Total marks for the Part 2 :10% of the course work

What will make you loose marks:

1. Not using classes with MVC type implementation - you should have at least 2-3 classes. You must separate the quiz model from the *MainWindow* class which is the view and the controller as well. The quiz model class supplies questions and the answers. See page 5 for the headers of the *QuizModel* class and *MainWindow* class.
2. Your program does not work at all
3. Your program works incorrectly
4. Does not count correct and incorrect answers properly
5. Questions which were answered incorrectly are not shown
6. Report at the end of each quiz is not shown exactly as above
7. Number of tries are not calculated correctly
8. Quiz does not repeat in a different randomized order if there are incorrect answers
9. Code must be indented use Ctrl + A + ctrl +i in QtCreator to indented
10. Header files are need to for the quiz class and the MainWindow class

How to submit:

Send the QT project folder as a zip file.

TBA

Last date for submission:

24 Dec 2013

Marks are cut for late submission at 20% of the assignment marks per day.

Groups:

Same as for part 1

Copying:

Copying from Internet or otherwise will strictly not be tolerated leading to loss of marks significantly.

Headers for the Quiz model and the MainWindow:

The Quiz model header (notice no UI)

```
#ifndef NUMBERWORDSQUIZ_H
#define NUMBERWORDSQUIZ_H

#include <QMap>
#include <QList>
#include <QString>

class NumberWordsQuiz {

public:
    NumberWordsQuiz ();
    int getNextNumberForQuestion ();
    QString getAnswerForNumber (int);
    int getSize ();

private:
    void initNumberWordsMap ();
    void initQuestionNumbersList ();
    QMap<int, QString> *numberWordsMap;
    int nextNumber;
    QList<int> *questionNumbersList;
};

#endif // NUMBERWORDSQUIZ_H
```

The MainWindow header

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QtGui/MainWindow>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QVBoxLayout>
#include <QWidget>

#include "NumberWordsQuiz.h"

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow();
    ~MainWindow();

public slots:
    void checkAnswer();

private:
    void showNextQuestion();
    int showResultsDialog();
    void showErrorMessageBox();

    //GUI widgets
    QLabel *headerLabel;
    QLabel *questionLabel;
    QLineEdit *answerLineEdit;
    QLabel *questionTrackerLabel;
    QPushButton *nextButton;
    QVBoxLayout *vBoxLayout;
    QWidget *window;

    NumberWordsQuiz *quiz;
    int questionsAsked;
    int currentNumber;
    int right;
    int wrong;
    double percentage;
    int attempts;
    bool showInstructions;
    QList<int> *numbersWithWrongAnswers;
};

#endif // MAINWINDOW_H
```