

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический
университет Петра Великого»

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Направление: 02.03.01 «Математика и компьютерные науки»

Отчет по практической работе
Создание игры Крестики-нолики с использованием
фреймворка Qt

Студент,

группы 3530201/20001

_____ Якунин Д. Д.

Преподаватель

_____ Глазунов В. В.

«_____» _____ 2023г.

Санкт-Петербург, 2023

Содержание

Введение	3
1 Постановка задач	4
2 Реализация	5
2.1 Реализация логики игры	5
2.2 Программная реализация игры	5
2.3 Реализация графического интерфейса	5
2.3.1 Главное меню	5
2.3.2 Настройки	6
2.3.3 Игровое окно	6
3 Тестирование	7
Заключение	10
Список литературы	11

Введение

Работа выполнена на языке C++. Для работы с графическим интерфейсом был использован Qt - фреймворк для разработки кроссплатформенного программного обеспечения. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования[1].

В ходе практической работы была создана игра Крестики-нолики. Суть игры заключается в поочередном выставлении игроками на поле символов 'X' и 'O'. Первый игрок, установивший в линию 5 символов своего вида выигрывает. Устанавливать символы в ряд можно по горизонтали, вертикали и диагоналям. Размер поля не ограничен и задается при запуске игры.

1 Постановка задач

Необходимо:

- Написать игру Крестики-нолики с графическим интерфейсом. Интерфейс должен быть создан с использованием фреймворка Qt.
- Создать главное меню, которое должно включать в себя три кнопки: «Play», «Settings» и «Exit». Кнопка «Play» должна вызывать игровое окно.
- Создать игровое окно, в котором будут расположены кнопки управления и ячейки игрового поля, в которые игроки будут ходить.
- Реализовать проверку победы какой-либо из сторон и соответствующе изменить интерфейс.

2 Реализация

2.1 Реализация логики игры

Игровое поле представляет собой квадратную матрицу произвольного размера. При начале игры создается такая матрица с пустыми элементами. Во время игры игроки нажимают на клетки поля и матрица заполняется соответствующими значениями.

После каждого хода идет проверка полей матрицы на наличие выигрышной последовательности. Если такая последовательность находится, то игра завершается победой игрока, собравшего такую последовательность. Если свободных полей в матрице не осталось, то игра завершается ничьей.

2.2 Программная реализация игры

Для проведения игры создан пользовательский класс `game`. Класс имеет следующие поля:

- 1) `State` - перечисление `enum`, содержащее текущее состояние игры.
- 2) `Map` - двумерная матрица с элементами типа `char`.

В конструкторе класса формируется поле размера, задаваемого в настройках, и оно заполняется пробелами.

Для совершения хода используется метод `makeMove()`. Он принимает координаты хода, заполняет соответствующую ячейку нужным символом, изменяет состояние игры и возвращает символ, который нужно написать на нажатой кнопке `CellButton` в игровом окне.

После каждого хода вызывается метод `checkGameOver()`, который проверяет все линии поля на наличие последовательности из 5 крестиков или ноликов, либо на отсутствие пустых клеток.

2.3 Реализация графического интерфейса

2.3.1 Главное меню

Главное меню создается с помощью класса `MainWindow`, который наследуется от `QMainWindow`. В главном меню расположены 3 кнопки типа `QPushButton` сгруппированных по вертикали с помощью дизайнера `Qt Creator`. Стил кнопки устанавливается редактированием поля кнопок `StyleSheet`.

Окна с настройками и игрой создаются в конструкторе главного меню, что позволяет пользоваться слотами этих окон без привязки.

- Кнопка «Play» открывает игровое окно и скрывает главное меню, используя базовые слоты `show()` для игрового окна и `hide()` для главного меню.
- Кнопка «Settings» открывает дополнительное окно настроек поверх основного, используя его слот `show()`.
- Кнопка «Exit» открывает стандартное окно `QMessageBox::question`, в котором у пользователя уточняют, точно ли он хочет закрыть приложение. Если да, то вызывается слот `QApplication::quit()` и приложение закрывается, иначе закрывается окно с вопросом.

2.3.2 Настройки

Окно с настройками является наследником класса `QDialog`. В нем есть окно выбора размера поля класса `QSpinBox` и две кнопки.

При нажатии на кнопку «ОК» значение записывается в глобальную переменную `MapSize` и вызывается сигнал `resizeGame()`, создающий новую игру с новым размером.

При нажатии на кнопку «Cancel» окно с настройками закрывается.

2.3.3 Игровое окно

Игровое окно описывается классом `gamewindow`, который наследуется от `QWidget`. В игровом окне присутствуют две кнопки типа `QToolButton`: "New game" и "Main menu"; и две надписи типа `QLabel`: одна статичная, а вторая отображает состояние игры.

Также во время игры окно заполняется клетками поля. Клетки поля реализованы с помощью пользовательского класса `CellButton`, который наследуется от `QPushButton`. `CellButton` имеет дополнительные поля `row` и `col` для задания координат, а так же указатель на игру для передачи информации о совершенном ходе. В классе есть слот `slotCellClicked()`, вызываемый при нажатии на кнопку: информация о ходе передается в игру и кнопка изменяет свой внешний вид.

- Кнопка «New game» перезапускает игру, используя `connect(settingswindow, &settings::resizeGame, gamewin, &gamewindow::newGame);`.
- Кнопка «Main menu» возвращает пользователя в главное меню, используя `connect(gamewin, &gamewindow::firstWindow, this, &MainWindow::show)`.
- Ячейки поля создаются при помощи слота `updateGameButtons()`. После их создания они связываются с игровым окном при помощи `connect(cells[i][j], SIGNAL(clicked()), this, SLOT(updateWindow()))`, а с самой игрой при помощи `connect(cells[i][j], SIGNAL(clicked()), cells[i][j], SLOT(slotCellClicked()))`.
- Слот `updateWindow()` нужен для изменения надписи с состоянием игры и вызова метода `endgame()` в случае окончания игры.
- Метод `endgame()` делает все кнопки на поле некликабельными.

3 Тестирование

При запуске программы открывается окно с главным меню. По центру расположены 3 кнопки. (Рис. 1.)

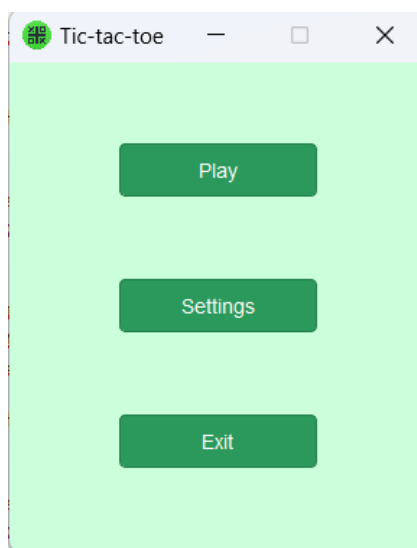


Рис. 1. Окно главного меню

Все кнопки выполняют свои задачи. Никаких проблем со взаимодействием нет, программа на данном этапе работает стабильно. Если нажать на кнопку «Settings», то откроется окно настроек. (Рис. 2.)

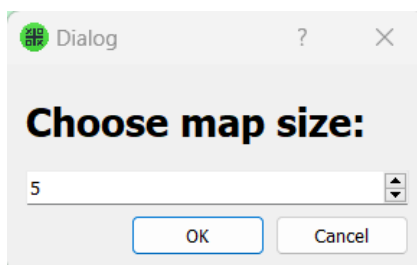


Рис. 2. Окно настроек

Если нажать на кнопку «Exit», то откроется окно выхода. (Рис. 3.)

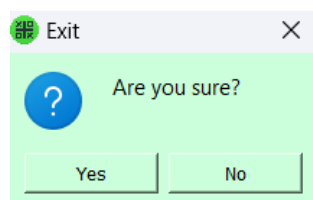


Рис. 3. Окно выхода

При нажатии на кнопку «Play» открывается игровое окно с пустым полем. (Рис. 4.)

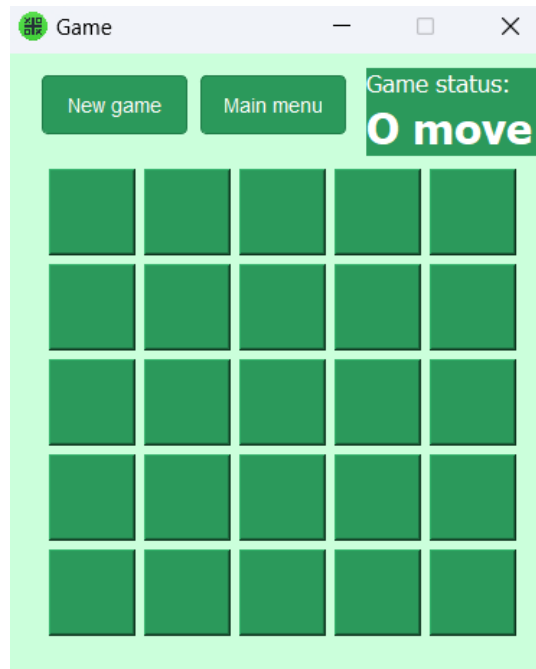


Рис. 4. Игровое окно

Во время игры клетки будут отмечаться соответствующими символами и сторона, которая сейчас ходит, будет указана в правом верхнем углу. (Рис. 5.)

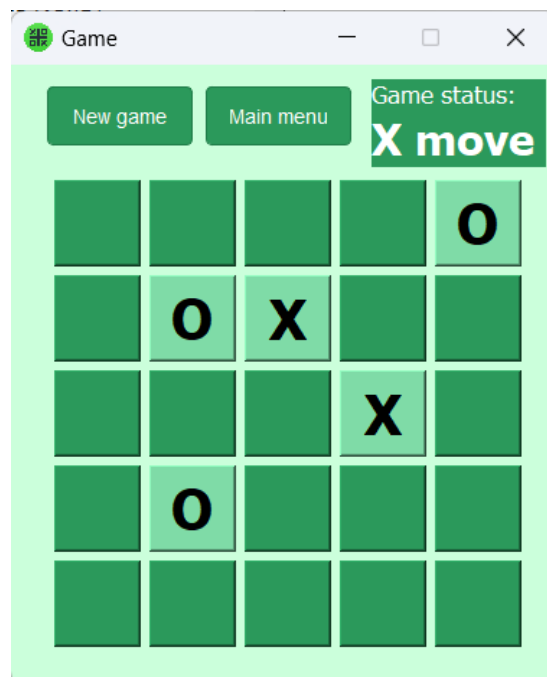


Рис. 5. Игровое окно во время игры.

По окончании игры статус игры будет написан в правом верхнем углу. Например, если выиграли нолики, то будет написано "O won!". (Рис. 6.)



Рис. 6. Игровое окно. Победа ноликов.

Если никто не победил, а свободные клетки закончились, то будет написано "Draw!". (Рис. 7.)

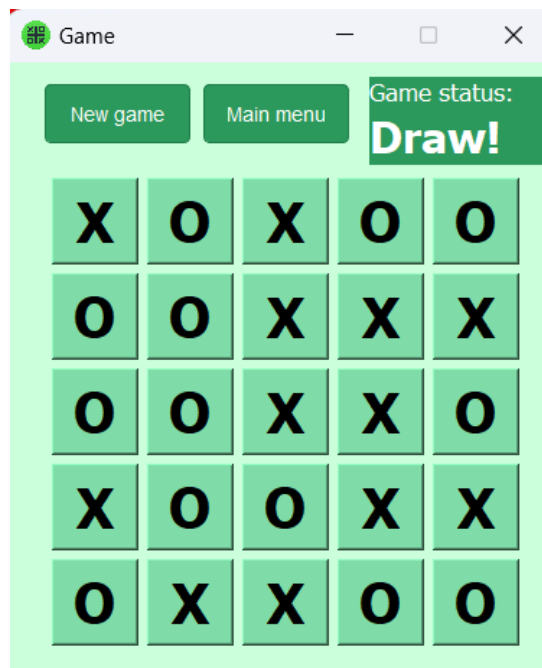


Рис. 7. Игровое окно. Ничья.

Заключение

В ходе выполнения работы были выполнены поставленные задачи:

- Написана игра Крестики-нолики с графическим интерфейсом. Для этого был использован фреймворк Qt.
- Создано главное меню, включающее в себя три кнопки: «Play», «Settings» и «Exit». Окно реализовано через QMainWindow, а кнопки через QPushButton.
- Сделано игровое окно, в котором расположены кнопки управления и ячейки игрового поля, в которые игроки будут ходить. Ячейки были реализованы через пользовательский тип CellButton.
- Реализована проверка событий, таких как победа какой-либо из сторон или ничьей. При таких событиях изменяется интерфейс.

В ходе выполнения работы были реализованы структуры данных, отвечающих за программную и графическую реализации игры. Также для графической реализации был использован дизайнер Qt Studio.

Была достигнута основная поставленная задача - написана игра Крестики-нолики, в приложении имеются главное меню, настройки и окно с самой игрой.

Для написания игры был использован язык C++, стандарт ISO C++ 11. Для работы была использована среда Qt Creator 5.0.2 с компилятором MinGW 7.3.0 64-bit.

Дополнительная информация для написания кода бралась из программы Assistant 5.12.12 и источников, указанных в списке литературы.

Был получен опыт разработки полноценного приложения с использованием объектно-ориентированного программирования на примере фреймворка Qt. Полученные знания могут быть использованы в будущем в учебе и работе.

Список литературы

- [1] [Qt](https://ru.wikipedia.org/wiki/Qt). Wikipedia : интернет-энциклопедия : сайт. – URL: <https://ru.wikipedia.org/wiki/Qt> (дата обращения: 28.06.2023)
- [2] [Signals & Slots](https://doc.qt.io/qt-6/signalsandslots.html). Документация Qt : сайт. – URL: <https://doc.qt.io/qt-6/signalsandslots.html> (дата обращения: 28.06.2023)
- [3] [Реализация похожей задачи](https://github.com/levelp/Qt_XO) : сайт. – URL: https://github.com/levelp/Qt_XO