

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата
01.03.02 — Прикладная математика и информатика

Отчет по практике

РАЗРАБОТКА ПРИЛОЖЕНИЯ
«СУДОКУ»

выполнил: Болдык Сергей и Афолина Дарья

Петрозаводск

2021

Содержание

Введение	3
1 Требование к приложению	4
2 Проектирование приложения	5
3 Реализация приложения	6
Заключение	7

Введение

Цель проекта:

Получить опыт разработки приложения на платформе Qt с использованием QtQuick и C++, реализовав головоломку «Судоку».

Задачи проекта:

1. Разработка логики игры

- Генерация игрового поля
- Обработка нажатий на клетки без значений.

2. Реализация модулей:

- Интерфейса головоломки
- Логики игры:
 - Генерация поля случайным образом
 - Обработка нажатия на клетку без значения
 - Таймер
 - Обработка победы или поражения

3. Оценка реализации приложения

4. Анализ возможных улучшений программы в будущем

1 Требование к приложению

- Простой интерфейс
- Возможность бесконечных переигрываний После запуска приложения пользователь видит перед собой окно с клетками, некоторые из которых заполнены. Для удобства поле разделено на квадраты 3x3, различающиеся по цвету (Рис. 1).

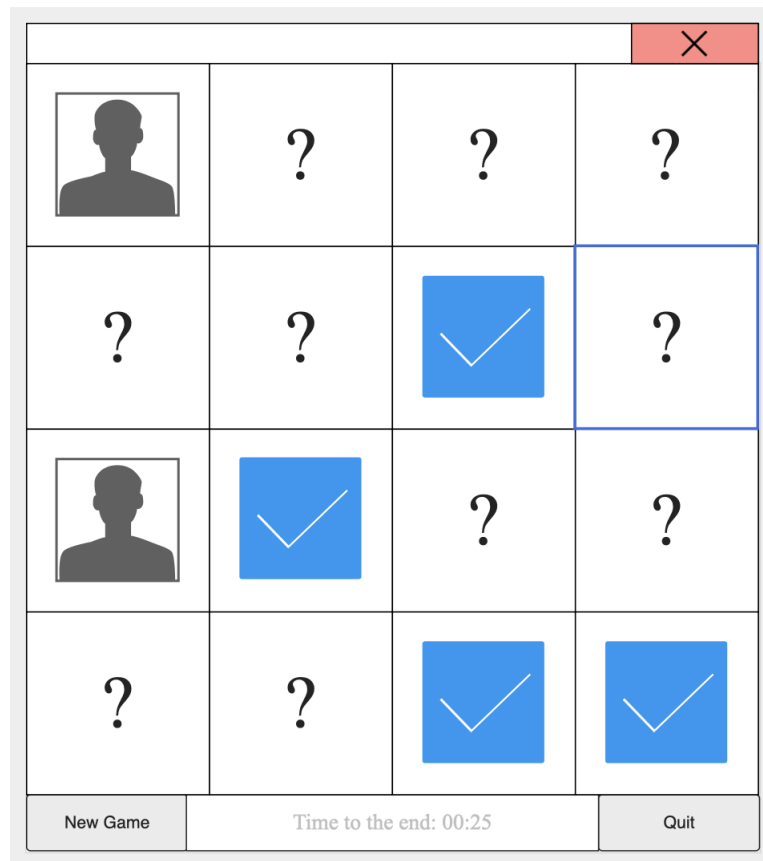


Рис. 1: Интерфейс приложения

2 Проектирование приложения

В приложении 2 основных модуля – qml сцены и C++ код:

1. main.qml - отвечает за все изменения в игре и содержит ряд элементов, отвечающих за интерфейс приложения.
2. C++ класс Helper, вспомогательный класс:
 - void QVector<int> newGame() - инициализирует игровое поле.
 - void changeState(int index) - изменяет значение в клетке, на которую нажали.
 - void swapLines(int line1, int line2) - меняет местами две строки значений в векторе состояния поля.
 - void swapRows(int row1, int row2) - меняет местами две колонки значений в векторе состояния поля.
 - bool checkSolution(void) - после каждого изменения в текущем состоянии игрового поля проверяет на достижение победы.
 - void hideSolution(void) - "прячет" случайные значения в векторе состояния игрового поля.
 - bool editable(int index) - определяет, можно ли пользователю изменять значение клетки.
3. Toolbar.qml - отвечает за игровое меню
 - stopTime() - остановка таймера
 - timeToSet(time) - вывод времени до конца игры в формате мм:сс
 - setTimer(seconds) - установка таймера
 - QString color(int index) - определение цвета для сектора, которому принадлежит клетка.
4. Gameplay.qml - игровое поле.
5. Cell.qml - элемент игрового поля.
6. EndScreen.qml - диалоговое окно, отображающееся по окончании времени на таймере или в случае победы.

3 Реализация приложения

В качестве языков программирования были использованы QML (для реализации пользовательского интерфейса в частности) и C++ (для частичной реализации логики, связанной с вычислениями), использовались библиотеки Qt.

1. Используемые библиотеки:

- QtQuick 2.0
- QtQuick 2.12
- QtQuick.Window 2.12
- QtQuick.Controls 2.0
- QtQuick.Layouts 1.12
- QtQuick.Controls 2.12
- QRandomGenerator
- QString

Итоговая сводка по объёму проделанной работы:

- Число файлов: 9
- Число функций: 10
- Число строк кода: 565

Возможности дальнейшего развития проекта:

1. Добавление различных уровней сложности головоломки.
2. Добавление возможности взять подсказку.

Заключение

В результате работы над приложением были выполнены все поставленные цели, получен опыт разработки приложений.

Реализовано приложение «Судoku» с простым управлением, понятным интерфейсом, возможностью взаимодействовать с игровым полем. Игра не ограничена одним уровнем, так как в результате выигрыша/проигрыша поле генерируется заново случайным образом.