

Техническое задание курсового проекта "Судoku"

Пьянков Семен, 572 группа

Вознюк Данила, 572 группа

1. Цель проекта

Целью курсового проекта является создание игры "Судoku" - с полями размера 9×9 и 16×16 , user-friendly gui для этой игры.

2. Общая идея задачи

В таблице 9×9 (16×16) должны быть расставлены числа от 1 до 9 (16) так, чтобы ни в одном столбце, строке и области 3×3 (4×4) (центральной, боковых и угловых) не было повторяющихся чисел (см. рисунок). Программа создает кроссворды разной сложности, по заполнении пользователем есть возможность проверить правильность заполнения.

3. Основные подзадачи и их взаимосвязь

- Решение созданного судoku
- Создание заполненного поля из заранее заданного константного поля
- Создание кроссворда из заполненного поля
- Хранение решений в отдельной структуре
- Создание интерфейса приложения

4. Общие предпосылки моделирования

Модель судoku - двумерный массив, каждый элемент которого является набором возможных значений данной ячейки кроссворда. В отдельной модели хранится база решений судoku - очевидно, что для каждого кроссворда не должно быть больше одного уникального решения (отправная точка для генерации кроссворда). В начале используется sample - уже созданный решенный судoku, элементы поля перемешиваются (не меняя правила кроссворда), выбирается случайное поле и удаляется. Это действие повторяется до тех пор, пока решений не станет больше одного.

5. Детальное описание содержания подзадач

Заранее задан пример решенного судoku. Перемешиваем его, меняя местами отдельно строки и отдельно столбцы внутри каждой зоны 3×3 (4×4) и меняя местами два конкретных числа (например меняем местами все числа 2 и все числа 9).

Затем запускаем алгоритм генерации судoku из готового поля - он использует алгоритм решения кроссворда. В первую очередь заполняем для всех пустых элементов двумерного массива все возможные числа. Затем проверяем (исходя из правил судoku), какие числа не могут располагаться в каждой ячейке и удаляем их. Проверяем ячейки, в которых есть только одно возможное число, сохраняем его и повторяем эти проверки пока происходят изменения.

Если после этого все ячейки заполнены - все хорошо. Иначе делаем рекурсивный перебор с возвратом и находим все(!!!) решения.

Умея решать, начинаем генерировать поле судoku: из созданного и перемешанного поля удаляем по одному элементу и проверяем число решений после удаления. В случае сохранения единственности решения - сохраняем изменение, в обратном случае откатываем изменение. *Возможность оптимизации - удаление сразу нескольких чисел или распараллеливание.*

Созданный массив выводится в интерфейс, написанный с использованием библиотеки Qt - из интерфейса есть возможность сохранять/загружать игры, решать судoku разной сложности, проверять на правильность решение, получать подсказки.

6. Состав работ и исполнители

- Ядро: структура кроссворда, алгоритм перемешки, алгоритм решения, алгоритм генерации sudoku - *Пьянков Семен, 572*
- Ядро: база решений, сохранение решений, оптимизация алгоритма генерации - *Вознюк Даниил, 572*
- Создание и отладка интерфейса - *Совместная работа*
- **Дополнительно:** возможность сохранения игр в виде исполняемых файлов - *Вознюк Даниил, 572*
- **Дополнительно:** создание кроссплатформенного приложения, способного запускаться под любой операционной системой вне зависимости от предустановленных библиотек - *Пьянков Семен, 572*

7. Используемые программные и технические средства

Компилятор - clang, g++.

Операционные системы - MS Windows, Unix, macOS с установленной библиотекой Qt.