

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Программирование

Отчет по выполнению проекта
Генерация кроссворда

Работу выполнил:

Курякин Д. А.

Группа: 13501/4

Преподаватель:

Вылегжанина К.Д.

Санкт-Петербург
2016

Содержание

1	Генерация кроссворда	2
1.1	Игровые принадлежности	2
1.2	Порядок игры	2
2	Проектирование приложения	2
2.1	Концепция приложения	2
2.2	Минимально работоспособный продукт	2
2.3	Прецеденты использования	2
2.4	Основные компоненты приложения	3
2.5	Используемые инструменты	3
2.5.1	Qt	3
3	Реализация приложения	3
3.1	Среда разработки	3
3.2	Реализация основных компонентов приложения	4
3.2.1	Библиотека Core	4
3.2.2	Консольное приложение	4
3.3	Тестирование	4
3.4	Статический анализ	4
3.5	Просмотр кода	5
3.6	Демонстрации	5
4	Выводы	5

1 Генерация кроссворда

1.1 Игровые принадлежности

Кроссворд — игра, состоящая в разгадывании слов по определениям. К каждому слову даётся текстовое определение, в описательной или вопросительной форме указывающее некое слово, являющееся ответом. Ответ вписывается в сетку кроссворда и, благодаря пересечениям с другими словами, облегчает нахождение ответов на другие определения.

Вместо текстовых определений могут выступать любые задачи, позволяющие дать ответ в одно слово в случае данного приложения текстовые определения - это сами слова.

1.2 Порядок игры

Игрок добавляет слова в словарь, затем нажимает на кнопку генерация. После того как кроссворд сгенерировался, игрок разгадывает кроссворд. После того как все слова разгаданы игрок получает сообщение, то что кроссворд разгадан.

2 Проектирование приложения

2.1 Концепция приложения

В ходе проектирования было разработана концепция продукта. Созданное приложение должно предполагать возможность добавление и удаление слов из базы данных - словаря, генерация сетки кроссворда и разгадывание.

2.2 Минимально работоспособный продукт

Минимальным работоспособным продуктом было признано консольное приложение, позволяющие производить генерацию кроссворда.

2.3 Прецеденты использования

На основе разработанной концепции была составлена UML диаграмма прецедентов использования (рис.1).

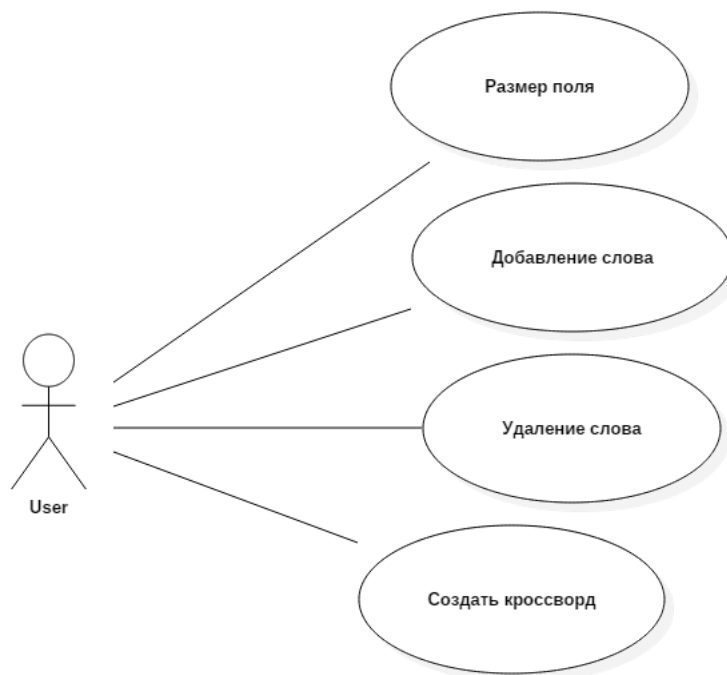


Рис. 1: Диаграмма прецедентов использования

2.4 Основные компоненты приложения

На основе анализа концепции и выделенных прецедентов использования было принято решение выделить три основных компонента, которые будут входить в состав продукта:

1. Библиотека

Включает в себя игровую модель, а также должна обеспечивать соблюдение правил при ее изменении, сообщать об игровой ситуации и определять моменты окончания игры. Кроме того, в библиотеке должны быть реализованы простой искусственный интеллект для игры в Сёги и механизм сохранения и загрузки партий. На основе этого было выделено два интерфейса: первый обеспечивает доступ к игровой модели, взаимодействие с ней, а второй позволяет использовать искусственный интеллект для игры.

2. Консольное приложение

Должно визуализировать с помощью текста игровую модель и позволять пользователю взаимодействовать с ней, а также предоставлять возможность использовать остальные функциональности, поддерживаемые библиотекой.

3. Графическое приложение

Графически визуализирует игровую модель, предоставляет пользователю графический интерфейс для взаимодействия с ней и выполнения остальных действий предусмотренных в реализации библиотеки.

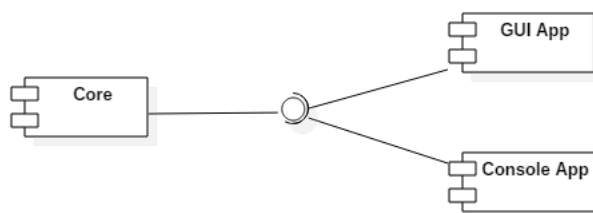


Рис. 2: Диаграмма компонентов

Так, на рис. 2 изображена UML диаграмма компонентов, описывающих взаимодействие компонентов и интерфейсов.

2.5 Используемые инструменты

Было принято решение, для реализации приложения соответствующего концепции использовать сторонние инструменты.

2.5.1 Qt

Qt - кроссплатформенный инструментарий разработки ПО, помимо всего, включающий средства для тестирования (Qt Test), разработки графического интерфейса (Qt Widget) и локализации (Qt Translation), которые необходимы для реализации проекта. Также было принято решение использовать файлы ресурсов, позволяющие хранить данные, необходимые для работы приложения прямо в исполняемом файле. Именно наличие этих инструментов стало основополагающим фактором при выборе Qt. По очевидным причинам использовалась последняя версия Qt 5.6.

3 Реализация приложения

3.1 Среда разработки

- Операционная система: Windows 10
- Система автоматической сборки: Cmake 3.5.1
- Компилятор: MinGW 4.9.2 32bit

3.2 Реализация основных компонентов приложения

3.2.1 Библиотека Core

Для реализации всех запланированных функциональностей было принято решение, двумя классами:

1. Vocabulary

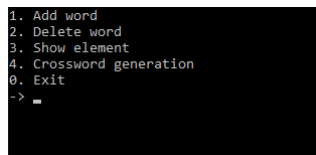
Данный класс предоставляет возможность посмотреть, добавить и удалить слова из словаря. Для выполнения данных действий в классе существуют методы **Output**, **AddWord** и **Remove**.

2. Field

Данный класс предоставляет возможность генерации сетки кроссворда и вписывании самих слов в сетку.

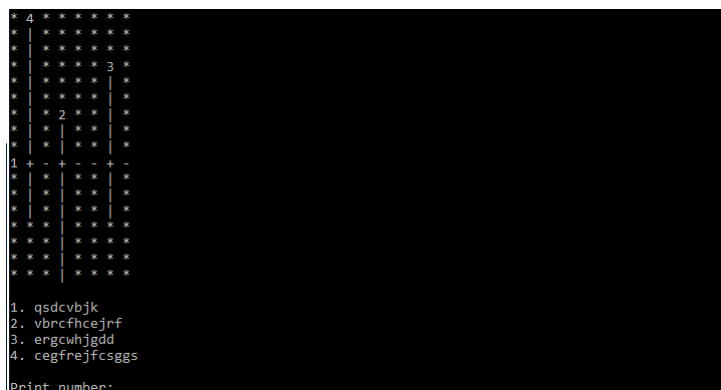
3.2.2 Консольное приложение

Консольное приложение было условно поделено на сцены: добавление слова, удаление слова, показать слова и генерация. Был создан классы **ConsoleApp**, **ConsolePlaingField** в которых была, реализована логика переключения между сценами.



```
1. Add word
2. Delete word
3. Show element
4. Crossword generation
0. Exit
-> _
```

Рис. 3: Основное меню консольного приложения



```

  4 * * * * *
x | * * * * *
x | * * * * *
x | * * * * 3 *
x | * * * * *
x | * * * * 2 *
x | * * * * *
x | * * * * *
1 + - + - + -
x | * * * * *
x | * * * * *
x | * * * * *
x | * * * * *
x | * * * * *
x | * * * * *
x | * * * * *
x | * * * * *
1. qsdcvbjk
2. vbrcfhcejrf
3. engcwhjgdd
4. cegfrejfcsggs
Print number: _
```

Рис. 4: Игровой процесс в консольном приложении

3.3 Тестирование

В ходе разработки проекта регулярно проводилось ручное тестирование.

Тестирование позволило обеспечить работоспособность продукта в ходе всего процесса разработки.

3.4 Статический анализ

Для улучшения качества и надежности создаваемого продукта использовались средства для статического и динамического анализа кода.

Для статического анализа использовалась программа Cppcheck v1.72. Данный инструмент производит анализ кода до момента компиляции и способен обнаруживать ряд ошибок связанных со стилем кода, его переносимостью и производительностью конечного продукта. Использование данной программы позволило создавать качественный код с высокой скоростью работы.

3.5 Просмотр кода

Применялась практика так называемого "code review" суть которого заключается в том, что достаточно квалифицированные люди просматривали код, в создании которого они не участвовали, и высказывали свои замечания и предложения. Так в ходе разработки было проведено всего лишь один просмотр кода, нацеленный на выявление ошибок и недоработок, связанных непосредственно с кодом и его стилем. В ходе данных проверок было получено более 50 замечаний, большая часть замечаний была исправлена, что позволило значительно повысить качество и надежность кода.

3.6 Демонстрации

Во время создания приложения было проведено 1 демонстрация, на которых группой людей, представляющих собой потенциальных пользователей разрабатываемого приложения, были сделаны различные замечания и высказаны множество предложений и пожеланий, основанных на внешнем виде продукта и стандартном цикле работы с ним. Анализ полученной информации позволял обнаруживать недочеты присутствующие в продукте на том, или ином этапе разработки, а также определять дальнейшие направления улучшения и расширения проекта, что, безусловно, положительно сказалось на конечном результате.

4 Выводы

В ходе выполнения данного проекта было получено множество новых знаний. Во-первых удалось улучшить владение языком C++, больше узнать о его строении и принципах работы, глубже познакомиться с STL, использовать многие сильные стороны языка в ходе разработки. Во-вторых был получен опыт, связанный с процессом разработки программного продукта.

Созданный в ходе работы продукт оказался вполне работоспособным. Планируется продолжить работу по совершенствованию библиотеки и создания графического приложения.