|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | МИНОБРНАУКИ РОССИИ |
|  |  |  |  | федеральное государственное бюджетное образовательное учреждение |
|  |  |  |  | высшего образования |
|  |  |  | **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф.** | |
|  |  |  |  | **Устинова»** |
|  |  |  |  | **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
|  |  |  |  | БГТУ.СМК-Ф-4.2-К5-01 |
| Факультет |  | И |  | Информационные и управляющие системы |
| Кафедра |  | шифр |  | наименование |
|  | И9 |  | Систем управления и компьютерных технологий |
| Дисциплина |  | шифр |  | наименование |
|  | Программирование на языке высокого уровня | | |



КУРСОВАЯ РАБОТА

на тему

Разработка игры «Космическая война» на языке C++ с использованием библиотеки SDL1.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выполнила студентка группы | |  |  | И967 |
|  | Королёва Ю. Ю. | | |  |
|  |  |
|  |  |  | |  |
|  |  | Фамилия И.О. | | |
|  | **РУКОВОДИТЕЛЬ** | | | |
| Васюков В. М. | |  |  |  |
|  | |  | |  |
| Фамилия И.О. | | Подпись | | |
| Оценка | |  |  |  |
| «\_\_\_\_\_» |  |  |  | 2018 г. |

САНКТ-ПЕТЕРБУРГ

2018 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ…………………………………………………………………………...3

1. ПОСТАНОВКА ЗАДАЧИ………………………………………………………6
2. ОПИСАНИЕ ИСПОЛЬЗУЕМОЙ ИЕРАРХИИ КЛАССОВ…………………..7
   1. Назначение и связь использованных иерархий классов…………………7
   2. Игровые объекты и изображения…………………………………………7
   3. Текст и кнопки……………………………………………………………..7
3. ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММЫ…………………………………...10
4. АЛГОРИТМ ПРОГРАММЫ…………………………………………………….11
   1. Обобщённый алгоритм работы программы…………………………….11
   2. Алгоритм игры…………………………………………………………….11
5. ПОДГОТОВКА ГРАФИЧЕСКОЙ СОСТАВЛЯЮЩЕЙ……………………..13
6. РЕАЛИЗАЦИЯ…………………………………………………………………...14

ЗАКЛЮЧЕНИЕ……………………………………………………………………...15

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ………………………………..16

ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ……………………………………….17

ВВЕДЕНИЕ

Объектно-ориентированное программирование (ООП) приносит с собой новый подход к решению задач, который включает в себя снижение общего уровня сложности программ, что позволяет программисту справляться с более объёмными программными продуктами. В отличие от процедурного программирования, где главное внимание уделяется алгоритмам, в ООП основное внимание направлено на данные. При использовании ООП проблему не решают с помощью процедурного подхода, заложенного в язык, а приспосабливают язык для решения этой проблемы. Идея заключается в создании таких форм данных, которые соответствовали бы основным характерным чертам проблемы. В языке С++ класс является спецификацией, описывающей такую новую форму данных, а объект - конкретной структурой данных, созданной в соответствии с этой спецификацией, класс в свою очередь определяет, какие данные будут образовывать объект и какие операции могут выполняться над этими данными [1].

Объектно-ориентированный язык характеризуется следующими свойствами [2]:

* Абстракция – выделение значимой информации и исключение из рассмотрения незначимой.
* Инкапсуляция – механизм, который связывает воедино код и данные, которые он обрабатывает, чтобы обезопасить их как от внешнего вмешательства, так и от неправильного использования.
* Наследование – процесс, с помощью которого один объект приобретает свойства другого.
* Полиморфизм – свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

Исходя из вышесказанного, можно сделать вывод о том, что при проектировании продукта на объектно-ориентированном языке, важными являются следующие этапы:

* определение основных объектов, необходимых для решения данной задачи;
* определение второстепенных объектов;
* определение иерархической системы классов, представляющих выбранные объекты.

Целью курсовой работы является проектирование и реализация программного продукта на объектно-ориентированном языке программирования С++ с использованием возможностей графической библиотеки SDL1.2.

Для достижения поставленной цели нужно было решить следующие задачи:

* Спроектировать иерархическую систему классов, удовлетворяющую требованиям к выполнению работы;
* Разработать структуру, позволяющую реализовать программный продукт на базе спроектированной системы классов;
* Подготовить медиа составляющую (движущиеся объекты-персонажи и фоновые изображения);
* Создать код программы, реализующей спроектированную иерархическую систему и разработанную структуру продукта, обеспечив связь программного исполнения и графической составляющей.

Пояснительная записка состоит из введения, шести разделов и заключения. В первом разделе приводится задание для курсовой работы и требования по её выполнению. Во втором разделе рассматривается использованная иерархия классов. Темой третьего раздела является общая структура программы. Четвёртый раздел описывает алгоритм программы. В пятом разделе описан процесс подготовки мультимедийной составляющей игры. Шестой раздел рассказывает о реализации продукта.

1. ПОСТАНОВКА ЗАДАЧИ

Требуется разработать игру, соответствующую следующему описанию:

Существует игровое поле в виде космического пространства. Из файлов загружаются картинки корабля, управляемого пользователем, и вражеских кораблей. Корабль игрока расположен внизу окна, может перемещаться вправо и влево посредством нажатия и удержания клавиш 🡨 и 🡪. Сверху на корабль игрока «сыпятся» корабли противника. Они имеют различный размер, внешний вид и скорость, все три параметра задаются компьютером случайно в процессе выполнения функции по управлению кораблями. Сделано это для усложнения процесса игры и улучшения графической составляющей. По нажатию клавиши «пробел» осуществляется выстрел из управляемого игроком корабля. Выстрел представляет собой сплошной зелёный лазерный луч, который насквозь прошибает корабли противника. При уничтожении вражеского корабля его изображение заменяется картинкой взрыва. Игра заканчивается тогда, когда один из кораблей противника достигнет того места внизу окна, где единственным способом уничтожить его станет таран с потерей своего корабля. При этом корабль игрока взорвётся и на экране появится сообщение о том, что игра окончена.

1. ОПИСАНИЕ ИСПОЛЬЗУЕМОЙ ИЕРАРХИИ КЛАССОВ

2.1 Назначение и связь использованных иерархий классов

Использованная в программе иерархия классов состоит из двух частей:

* совокупности “Игровые объекты и изображения”, предоставляющей классы для игровых объектов (корабля противника и корабля игрока), которые реализуют всю логику поведения и взаимодействия этих объектов, а также их отрисовку и перемещение.
* иерархии “Тексты и кнопки”, включающей в себя классы облегчённого осуществления вывода текстовых строк на экран, что в SDL2 сделать непросто стандартными средствами. Данная реализация позволяет существенно сократить размер кода и его читабельность, не допуская многократного дублирования фрагментов кода для отображения многочисленных текстовых строк.

2.2 Игровые объекты и изображения

Базовым классом будет “Статичное изображение”, обладающее координатами левого верхнего угла окна, а также именем загружаемого файла. Объект такого класса представляет собой простейшее статичное изображение без каких-либо элементов управления.

От него наследуют два класса – класс Me, описывающий поведение объекта-корабля игрока, и класс Enemy, описывающий движение корабля противника.

2.3 Тексты и кнопки

Есть базовый класс Text, представляющий собой тестовую строку, выводимую в заданном по координатам месте окна.

От этого класса наследует класс “Кнопка”, представляющий собой такую же строку текста, только запоминающее состояние (выбрана/не выбрана) и меняющее вид в зависимости от этого состояния. Элементы класса “Кнопка” используются в качестве управляющих элементов в меню (или, вернее, в качестве визуализации управляющих элементов).

1. ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММЫ

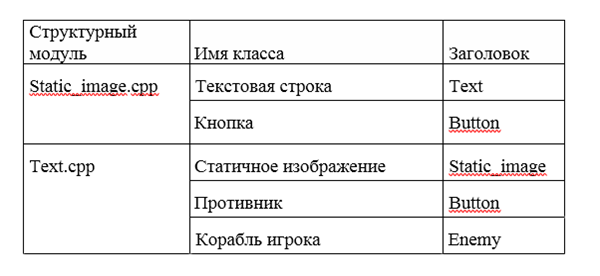
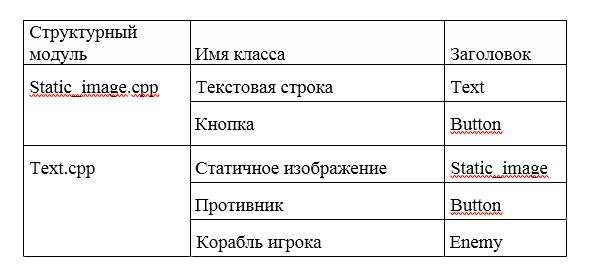
Со структурной точки зрения, программа представляет собой 3 независимо компилируемых файлов.

Наиболее важным из них является “main.cpp”, обеспечивающий загрузку ресурсов, инициализацию библиотек и собирающий события для дальнейшей обработки. Кроме того, именно в этом файле заданы алгоритмы взаимодействия классов описанных в других единицах компиляции. Остальные модули подключены с помощью директивы препроцессора #include и одноимённых заголовочных файлов, содержащих объявления описанных в этих модулях классов и функций.

* + файле “Static\_image.cpp” реализована иерархия “Игровые объекты и изображение”, описание которой можно найти в предыдущем разделе. Здесь же реализованы функции для организации движения объектов и загрузки картинок. Файл “Text.cpp” содержит описание реализации классов иерархии “Тексты и кнопки”, описанной в разделе 2.2.

Распределение классов по структурным модулям показано в таблице 1.

Таблица 1 - Распределение классов по структурным модулям



4. РАЗРАБОТАННЫЕ АЛГОРИТМЫ

4.1 Обобщённый алгоритм работы программы

При запуске программы происходит инициализация графической библиотеки SDL2 и её расширения SDL2\_ttf. В случае успешной инициализации создается окно, которое и будет формировать вместе с наложенными на него текстурами изображение на экране. Появляется заставка (draw\_beginning()) с предложением ввести имя игрока (input\_name()). Затем на экран выводится меню, с пунктами “Play”, “Change Player”, “Score”, “Rules”, “Exit”. Далее, пока в меню не будет выбран пункт “Exit” или не будет нажата клавиша Escape, происходит опрос событий мыши и их обработка, по результатам которой вызываются функции “Play” (play()), “Change Player” (draw\_change\_player()), “Score” (draw\_results()), “Rules” (draw\_rules()). После нажатия клавиши Escape или выбора пункта “Exit” в меню цикл прерывается. Затем происходит очистка памяти, закрытие графических окон и деинициализация библиотек SDL2, SDL2\_ttf.

4.2 Алгоритм игры

При запуске игровой функции инициализируются объекты требуемых классов. Когда инициализация объектов завершена, запускается основной игровой цикл. На каждом шаге основного цикла первым делом производится опрос клавиатуры и обработка событий. Если нажата клавиша 🡨, изображение корабля героя продвигается влево, если нажата клавиша 🡪, изображение корабля героя движется вправо. При нажатии клавиши пробел происходит выстрел. Клавиша “Escape” вызывает сохранение игры и возврат в главное меню. Функция движения корабля противника запускается в отдельном потоке. При каждом уничтожении корабля противника меняются показатели количества очков в правом верхнем углу. Также в любой момент игрового цикла на экране высвечивается информация со справочной строки с подсказками по поведению и управлению персонажем, актуальной на данный момент времени. При достижении кораблём противника расчётной границы игра завершается, и на экране появляется сообщение о её окончании, затем выполняется возврат в главное меню.

5. ПОДГОТОВКА ГРАФИЧЕСКОЙ СОСТАВЛЯЮЩЕЙ

Для подготовки графической составляющей использовался графический редактор Microsoft Paint версии 6.3. Задние планы и картинки героев были взяты со следующих сайтов:

* https://ru.pinterest.com/
* http://pikabu.ru/story/kosmicheskie\_korabli\_iz\_fantasticheskikh\_filmov\_326082

Изображения кропотливо приводились в соответствие общему художественному стилю игры.

6. РЕАЛИЗАЦИЯ

Программный продукт написан на языке С++ в среде разработки Code::Blocks версии 13.12 (Набор средств разработки версии 1.19, компилятор MinGW32-g++) с использованием графической библиотеки SDL (Simple DirectMedia Layer) и библиотеки для её расширения SDL\_ttf. Целевая платформа - 32 битные операционные системы семейства Microsoft Windows версии XP и выше.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была спроектирована иерархия классов, описывающих элементы интерфейса и объекты реализующие игровую логику. Затем была разработана структура программы, поддерживающая созданную иерархию и подготовлена графическая составляющая продукта, в частности, текстуры для карты и бойцов. Результатом работы стала реализация программного продукта, который соответствовал критериям технического задания и имел законченный вид.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* Прата. С. [Stephen Prata] Язык программирования C++. Лекции и упражнения / С. Прата. – 6-ое издание; – М.: Вильямс, 2015. – 1248 с. — ISBN 978-5-8459-2048-5
* Шилдт. Г. [Herbert Schildt] С++: Базовый курс / Г. Шилдт. – 3-е издание;

– М.: Вильямс, 2008. – 624 с. — ISBN 978-5-8459-0768-4

* Lazy Foo' Productions, Texture Loading and Rendering [Электронный ресурс]:URL:http://lazyfoo.net/tutorials/SDL/07\_texture\_loading\_and\_rendering/inde x.php.
* Lazy Foo' Productions, Color Keying [Электронный ресурс]: URL:

http://lazyfoo.net/tutorials/SDL/10\_color\_keying/index.php.

* Lazy Foo' Productions, Timing [Электронный ресурс]: URL: http://lazyfoo.net/tutorials/SDL/22\_timing/index.php.
* О. В. Арипова, А. Н. Гущин, О.А. Палехова Программирование на языке высокого уровня; - Спб., 2014 – 164 с. – ISBN 978-5-85546-841-0
* Гущин А. Н. Применение библиотеки SDL для разработки программ на языке С: учебное пособие – СПб, 2014 – 118 стр. – ISBN 978-5-885546-833-5

ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ

Исходный текст программы и все необходимые для её работы файлы предоставляется на оптическом носителе информации - DVD-RW диске.