**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc138343009)

[1 Постановка задачи 4](#_Toc138343010)

[2 Обоснование иерархии используемых классов 6](#_Toc138343011)

[3 Описание программы 9](#_Toc138343012)

[4 Инструкция пользователю 11](#_Toc138343013)

[ЗАКЛЮЧЕНИЕ 17](#_Toc138343014)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc138343015)

[ПРИЛОЖЕНИЕ А. Текст программы 19](#_Toc138343016)

**ВВЕДЕНИЕ**

Установлено, что компьютерные игры оказывают значительное влияние на мировосприятие, мироощущение, миропонимание современного человека, становятся не только объектами культурного досуга, но и новой профессиональной деятельностью.

С одной стороны, компьютерные игры являются программами, с другой стороны - видеоиграми, с третьей - жанром искусства. Полностью проанализировать данное явление достаточно сложно в силу того, что каждый день появляются новые игры и игровые сообщества. Вектор развития игровой индустрии постоянно меняется в связи с возникновением новых игровых жанров и становится сегодня точкой бифуркации для определения роли и места компьютерных игр в истории человечества [1].

Целью данной курсовой работы является разработка игры на языке программирования C++ с использованием библиотеки SDL (Simple DirectMedia Layer) версии 2.0. SDL представляет собой мощный инструментарий, предоставляющий доступ к аппаратным возможностям компьютера для создания мультимедийных приложений.

В процессе работы будут рассмотрены основные принципы объектно-ориентированного программирования (ООП), такие как наследование, полиморфизм и инкапсуляция, которые позволяют создавать структурированный и гибкий код. Будет проведен анализ требований к игре, спроектировано пользовательское меню и интерфейс, а также создана иерархия классов для управления игровыми объектами.

Успешное выполнение курсовой работы позволит приобрести практические навыки разработки игр на языке C++ с использованием библиотеки SDL 2.0, а также понимание принципов ООП и их применение в игровом программировании.

1. **Постановка задачи**

В рамках курсовой работы необходимо создать игру «Dlinnochertv». Правила игры: Управляя монстром по имени Длинночертв игрок должен собирать светящиеся предметы, такие как лампочки, свечи и люстры. За каждый собранный светящийся предмет начисляется определённое количество баллов: за лампочку 5 баллов, за свечу 10 баллов, за люстру 15 баллов. Также на карте присутствуют объекты «люди». За каждую встречу с человеком игрок теряет 5 баллов. В левом верхнем углу расположен таймер с обратным отсчётом времени. Игроку нужно набрать как можно больше баллов за 60 секунд, по окончании времени игра завершается. Баллы отображаются в правом верхнем углу экрана.

В рамках игры должно быть реализовано меню позволяющее игроку: запустить игровой процесс, просматривать 10 лучших результатов игры, которые хранятся в бинарном файле и выводятся на экран по запросу пользователя (результаты можно как стереть, так и переписать в текстовый файл), сменить имя игрока и узнать правила игры.

Требования к возможностям программы и процессу ее выполнения:

а) Игра реализуется согласно варианту задания.

б) Программа при выполнении должна иметь дружественный интерфейс, включающий заставку, меню, пояснения и удобные средства для управления программой.

1) В любой момент игры на экране должна присутствовать справочная строка с актуальной для текущего момента информацией («горячие» клавиши, подсказки и т.п.).

2) Заставка должна выводится на экран при запуске программы и давать представление об игре. На заставке должны быть отражены название игры, фамилия и номер учебной группы ее автора. По завершении заставки игрока просят ввести имя.

3) Главное меню должно в обязательном порядке включать в себя пункты «Игра», «Смена игрока», «Просмотр результатов», «Правила игры», «Выход».

4) При выборе пункта меню «Просмотр результатов» на экран должно в табличном виде выводиться содержимое двоичного файла результатов и подменю, позволяющее «вернуться в главное меню».

в) Управление движением управляемого игроком объекта осуществляется с клавиатуры.

г) Необходимо обеспечить достаточную скорость перерисовки динамически перемещаемых элементов.

д) Десять лучших результатов игры (очки, минуты и т.п.) сохраняются в двоичном файле в виде структур, содержащих непосредственно результат и имя игрока, получившего его.

Требования к реализации программы:

а) Выполнение курсовой работы должно быть основано на принципах объектно-ориентированного программирования. В работе должна быть построена и обоснована иерархия классов, описывающих предметную область конкретной задачи.

б) Программа должна являться графическим приложением, написанным на языке C++ с использованием библиотеки SDL2.

в) Описание классов одной иерархии должно располагаться в отдельном файле.

г) Разнотипные объекты одной иерархии должны быть объединены в общий массив или список. В простых случаях допускается использовать одну простую или ссылочную переменную для работы с разными объектами.

1. **Обоснование иерархии используемых классов**

В контексте данной задачи, игра содержит несколько основных объектов: фон, светящиеся предметы, люди, счетчик времени, счетчик баллов и монстр. Чтобы облегчить реализацию, эти объекты могут быть разделены на группы, исходя из их сходных характеристик. Такой подход позволяет структурировать программу, работать с каждым элементом отдельно и использовать принципы наследования, что устраняет необходимость дублирования полей и методов для схожих объектов [2]. Например, счетчик времени и счетчик баллов выполняют определенные подсчеты и могут быть описаны на основе абстрактного класса "Counter". Аналогично, объекты, которые позволяют набирать и терять баллы, могут быть сгруппированы и основаны на абстрактном классе "ScoreObjects". Таким образом, объекты разделены на две группы, каждая из которых представляет иерархию классов.

Объекты, которые ведут подсчет в ходе программы, объединяют следующие характеристики: шрифт, цвет шрифта, структура для определения размеров и позиции. Методом класса является рендеринг нового состояния счётчика, актуального на данный момент игры.

На основе «Counter» написан класс «ScoreCounter». В нем полем класса является переменная типа int предназначенная для хранения актуального количества набранных баллов. Методом является функция обновления состояния счетчика баллов.

Ещё одним производным классом от класса «Counter» является класс «TimeCounter». Полями класса являются переменные типа int предназначенные для обратного отсчёта времени игрового процесса. Методом является функция для обновления состояния таймера актуального на данную секунду игры.

Полная иерархия объектов ведущих подсчет по ходу игрового процесса, представлена на рисунке 1.

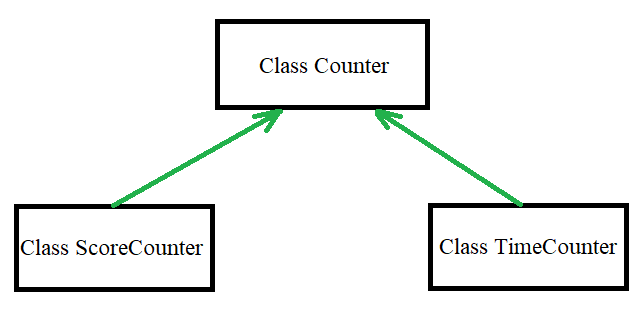


Рисунок 1 – Изображение иерархии объектов, ведущих подсчет

Объекты (балльные предметы), позволяющие набирать и терять баллы имеют несколько общих характеристик: переменная для определения ценности объекта, координаты по x и y. Методами являются функции: определения случайной позиции на карте, обновление позиции на карте, рендеринг позиции на карте и обработка взаимодействия игрока с объектами. Всё это является содержимым класса ScoreObjects.

На основе базового класса описаны классы для объектов с положительной балльной ценностью: «Lamp», «Candle», «Chandelier». И отрицательной балльной ценностью: «Red\_man», «Violet\_man», «Orange\_man». Для каждого из этих классов определена своя текстура отображающаяся во время игры на карте.

Базовый класс ScoreObjects определяет общий интерфейс и функционал для всех классов-потомков, абстрактные методы, которые должны быть реализованы в каждом классе-потомке.

Использование наследования позволяет сократить повторяющийся код и структурировать объекты игры в соответствии с их общими характеристиками и функционалом. Кроме того, благодаря базовому классу ScoreObjects, можно работать с объектами-потомками через указатель на базовый класс, что упрощает обработку и взаимодействие с ними в общем контексте игры.

Общая схема иерархии балльных предметов представлена на рисунке 2.

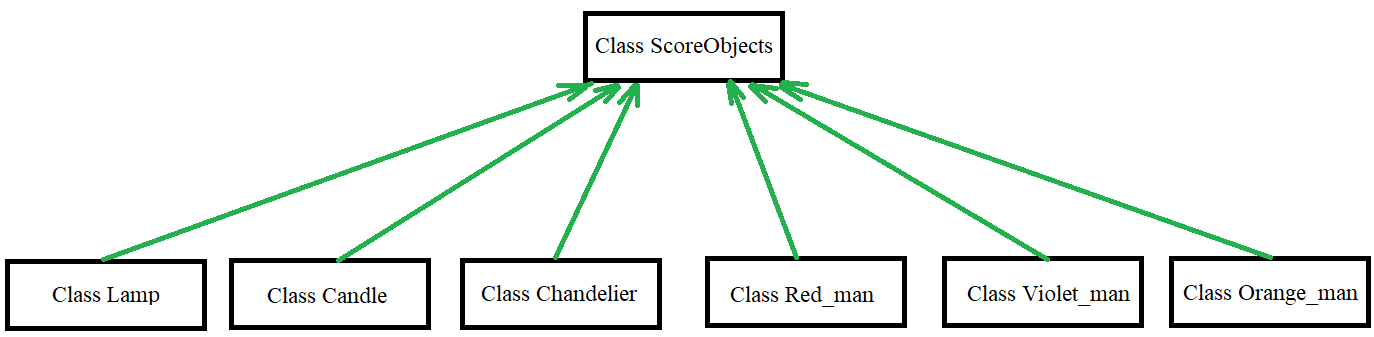


Рисунок 2 – Изображение иерархии балльных предметов

1. **Описание программы**

В ходе разработки данной программы на языке C++ с применением графической библиотеки SDL версии 2.0 была использована соответствующая справочная литература [3,4].

Игра, написанная с использованием библиотеки SDL версии 2.0, может быть импортирована и работать на различных платформах, таких как Windows, Linux, MacOS. SDL2 является кроссплатформенной библиотекой, которая предоставляет абстракцию для работы с графикой, аудио, вводом и другими аспектами игрового программирования [5].

Для обеспечения правильной работы приложения следует разместить файлы "SDL\_ttf.dll", "zlib1.dll", "SDL.dll", "libfreetype-6.dll" и "Samson.ttf" в одном каталоге. Файлы с расширением ".dll" (Dynamic Link Library) являются исполняемыми библиотеками в операционных системах Windows, они используются программой, а файл "Samson.ttf" содержит данные о шрифте, используемом для вывода текста.

В процессе разработки программы, различные иерархии классов были разделены и организованы в отдельные файлы в соответствии с требованиями курсовой работы. Файл "Counter.hpp" содержит описание классов, связанных с иерархией счетчиков, в то время как файл "ScoreObjects.hpp" содержит описание классов, относящихся к иерархии балльных предметов. Оба этих файла были включены в основную программу с использованием директивы #include. Аналогичным образом создан файл «initWindow.hpp», содержащий описание методов для инициализации окна игры и отрисовки фона, в файле «ConstantsSize.h» определены константные значения для ширины и высоты игрового окна, этот заголовочный файл предоставляет удобный способ задания и использования фиксированных размеров для окна игры, он позволяет избежать повторения числовых значений в коде и делает их более читаемыми и легко изменяемыми. Также был создан файл «Game.hpp», в котором содержится описание методов содержащих логику игрового процесса игры, в файле «GameObject.hpp» находится описание класса реализующего управление главным игровым персонажем, а в файле «ObjectManager.hpp» располагается описание менеджера балльных предметов, в нём имеются описание методов добавления нужного количества объектов в вектор. Таким же образом был создан файл «Menu.hpp», в котором описываются методы реализующие логику взаимодействия пользователя/игрока с игрой, в файле «filesForRecords.hpp» представлено описание класса, который реализует логику сохранения, считывания, подготовки для копирования и удаления бинарных записей рекордных результатов. Файл «TextRender.hpp» содержит в себе описание методов для размещения текста на экране. Также в проекте присутствует файл «TextureManager.hpp», менеджер текстур предоставляет функциональность загрузки изображений и создания текстур для дальнейшего использования в игровом окне, одним из преимуществ такого менеджера текстур является возможность централизованного управления и загрузки текстур в игре.

Также в процессе игры создаются файлы «record.bin» и «file\_for\_print.txt». В «record.bin» хранятся 10 лучших результатов, а в «file\_for\_print.txt» записываются результаты из бинарного файла с рекордами при запросе пользователя.

1. **Инструкция пользователю**

При запуске приложения пользователю демонстрируется заставка (Рисунок 3), которая предусмотрена требованиями к выполнению курсовой работы.



Рисунок 3 – Заставка игры

После этого пользователь должен ввести имя с помощью клавиатуры (Рисунок 4). Это имя будет сохранено вместе с набранными в игре баллами в списке рекордов.



Рисунок 4 – Ввод имени игрока

Затем пользователь попадает в главное меню (Рисунок 5). Меню поддерживает взаимодействие как через нажатие клавиш-стрелок вверх/вниз, так и через использование мыши. В меню пользователю предоставляется возможность выбора одного из пяти пунктов: «Run the game», «Guide», «Player change», «Records», «Exit».



Рисунок 5 – Меню игры

Если пользователь выберет первый пункт меню, то есть «Run the game», то начнется игровой процесс (Рисунок 6). Суть игры: накопить как можно больше баллов за 60 секунд собирая такие предметы как лампочки (5 баллов), свечи (10 баллов) и люстры (15 баллов) при этом стараться избегать встречи с людьми, потому что за каждую встречу с человеком игрок теряет 5 баллов. Движение вверх, вниз, вправо, влево осуществляется посредством нажатия на соответствующие клавиши-стрелки. В левом верхнем углу расположен таймер ведущий обратный отсчет, нужно собирать баллы пока таймер не дошел до нуля. В левом верхнем углу располагается счетчик очков, по центру сверху отображается имя текущего игрока:



Рисунок 6 – Игровой процесс

Нажав на клавишу ESC можно поставить игру на паузу (Рисунок 7), для возобновления игры нужно заново нажать на клавишу ESC, если пользователь в состоянии паузы нажмёт на BACKSPACE, то он вернётся в меню:



Рисунок 7 - Пауза

По окончании игрового процесса появляется окно (Рисунок 8) демонстрирующее результат за прошедшую игру:



Рисунок 8 – Демонстрация результатов игры

Если выбрать пункт меню «Guide», то отобразится текст (Рисунок 9), касающийся правил игры и небольшой информации по лору персонажа:

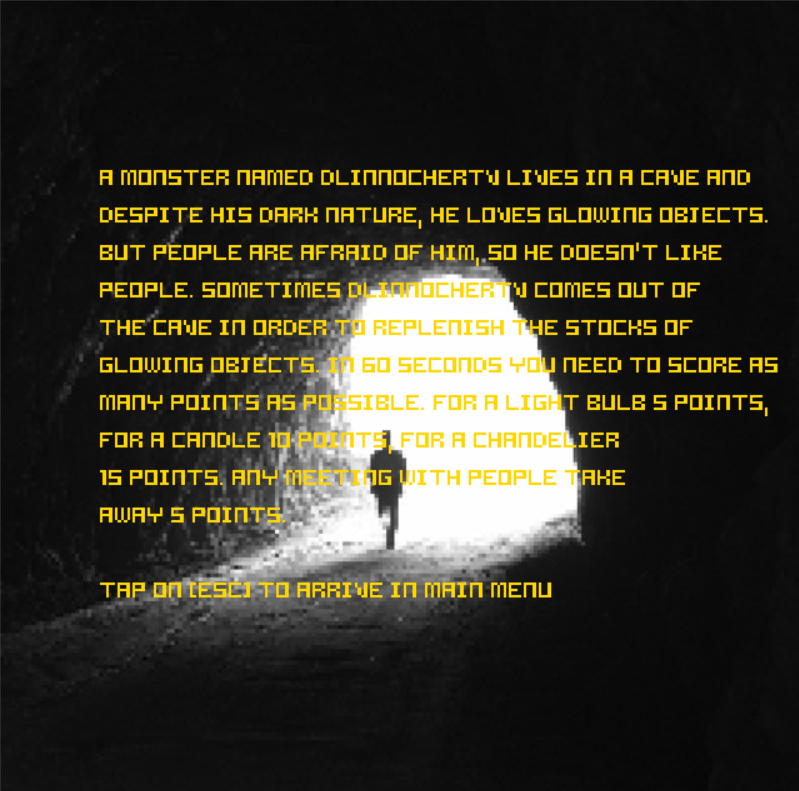


Рисунок 9 – Справочная информация

Если пользователь решит нажать на пункт меню «Records», то перед ним отобразится таблица лучших результатов игры (Рисунок 10), нажав на кнопку ESC можно покинуть данный пункт меню, при нажатии на E будет подготовлен файл для печати, а при нажатии на Q список рекордов удалится.

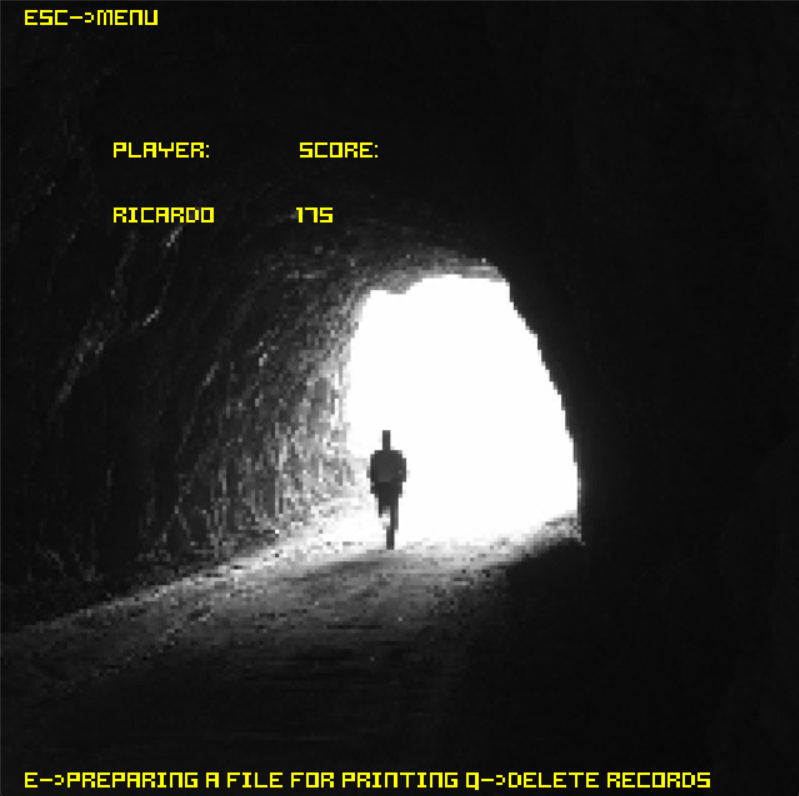


Рисунок 10 – Список рекордов

Удалив результат или подготовив файл к печати выводится текст сигнализирующий о том, что операция выполнена (Рисунок 11):



Рисунок 11 – Сигнализация об успешности удаления или подготовки к печати

Зайдя повторно в пункт «Records» после удаления рекордов, появляется текст сигнализирующий о том, что список рекордов пуст (Рисунок 12):



Рисунок 12 – Список рекордов пуст

При нажатии пункта меню «Player change» вновь появляется окно (Рисунок 4), в котором будет предложено сменить имя игрока.

При нажатии на «Exit» приложение будет закрыто.

**ЗАКЛЮЧЕНИЕ**

В рамках выполнения курсовой работы по дисциплине "Информационные технологии и программирование" была успешно разработана игра под названием "Dlinnochertv". Одним из важных достижений работы является создание дружественного пользовательского интерфейса, который обеспечивает комфортное взаимодействие пользователя с игрой. Кроме того, были разработаны обоснованные иерархии классов, которые соответствуют принципам объектно-ориентированного программирования.

Разработка игры "Dlinnochertv" стала ценным опытом, который позволил погрузиться в мир программирования и расширить свои знания в области информационных технологий. Благодаря этой работе я приобрел не только навыки разработки игровых приложений, но и глубже понял принципы объектно-ориентированного программирования и эффективного проектирования пользовательского интерфейса.

Все задачи выполнены, поставленная цель данной курсовой работы достигнута.