

Введение

На курсовое проектирование была поставлена задача «Разработка игрового приложения «Валли».

Задача заключается в разработке 2D-платформера на языке C++, основная цель проекта – создание игрового продукта, где игрок управляет персонажем, преодолевая уровни с неожиданными препятствиями.

Разрабатываемая игра ориентирована на любителей аркадных платформеров всех возрастов – от школьников до взрослых, желающих позабавиться и проверить свои навыки. Игровая система интуитивно понятна, не требует специальных знаний, но предоставляет достаточно вызовов и стимул для повторных прохождений и достижения лучших результатов.

Пояснительная записка состоит из следующих разделов:

- «Анализ задачи»: в первом разделе анализируется предметная область проекта, анализируются существующие платформер-игры аналогичного жанра, определяются ключевые требования к игровым механикам и функциональным модулям. Описываются входные данные (управление, параметры уровней) и ожидаемые выходные данные (результаты прохождения, статистика игрока);

- «Проектирование задачи»: в данном разделе детально описывается архитектура приложения: организация уровней, сценарии взаимодействия игрока с объектами (шипами, подвижными блоками, проваливающимся полом), модель начисления и расходования очков, структура базы данных для сохранения прогресса;

- «Реализация»: в третьем разделе приводятся описания основных игровых компонентов и их реализация в выбранной среде разработки. Представлены структуры данных, используемые алгоритмы обработки физики и коллизий, интерфейсные элементы (экран загрузки, HUD, магазин);

- «Тестирование»: описывается план функционального и нагрузочного тестирования: проверка всех игровых механик на разных уровнях сложности, тестирование системы сохранений и корректность начисления/списания очков, устойчивость приложения при многократных запусках.

- «Применение»: рассматриваются целевое назначение игры, среда её функционирования (платформы, на которых она может запускаться), описываются процесс использования игры (руководство пользователя) и рекомендации по её дальнейшему развитию и поддержке.

- «Заключение»: содержит итоговую формулировку задачи, основные результаты проделанной работы, описание применённых технологий и оценку степени автоматизации разработки;

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

- «Список использованных источников»: в данном разделе приводится перечень всех источников, которые были использованы при разработке проекта.
- «Листинг программы будет представлен в приложении»: в приложении приводятся ключевые фрагменты исходного кода игры;
- в графической части будет представлена структура проекта.

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ задачи

1.1 Постановка задачи

1. Наименование задачи: «Разработка игрового приложения «Валли».

2. Описание и исследование предметной области.

Жанр: 2D платформер.

Игровой процесс: игрок управляет персонажем, перемещаясь по уровням, преодолевая препятствия (шипы, движущиеся платформы, проваливающийся пол) и собирая монеты. Основная цель – добраться до финиша уровня. Смерть от препятствий или выхода из за границ карты приводит к перезапуску уровня

Ключевые механики:

- шипы: статичные объекты, при контакте с которыми персонаж погибает;
- движущиеся блоки/платформы: объекты, перемещающиеся по заданным траекториям (горизонтально, вертикально), могут перемещать игрока;
- проваливающийся пол: участки пола, проваливающееся пол по определенным координатам , создавая ловушки или открывая новые пути;
- монеты: коллекционные объекты, дающие игроку очки +10 за монету и за прохождение. Служат внутриигровой валютой;
- система очков и магазин: накопленные очки тратятся в магазине на скины (визуальные изменения персонажа), которые влияют только на внешний вид.

Периодичность использования программы: игра предназначена для эпизодического использования пользователем (сессии по прохождению уровней). Периодичность зависит от интереса игрока.

Как новая задача упрощает решение: данный проект автоматизирует создание игрового опыта с конкретными механиками (шипы, движущиеся блоки, монеты, магазин скинов), предоставляя пользователю готовую для игры программу. Вместо ручного создания уровней и программирования физики в каждом проекте, здесь это сделано один раз в коде.

Описание существующих аналогов:

- Geometry Dash – это ритмичный платформер с автоматическим бегом и акцентом на препятствия;
- Level Devil – это хардкорный платформер, известный своей высокой сложностью и ловушками, которые могут появиться в любой момент.

Отличия данного проекта. Фокус на конкретном наборе механик (особенно взаимодействие с движущимися блоками), реализация магазина скинов за внутриигровую валюту (монеты), разработка на языке программирования C++ с использованием библиотеки SFML , без использования игровых движков.

3. Выходная информация:

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

Перечень используемых способов вывода данных: графический вывод через SFML (рендеринг спрайтов, текстур, примитивов, текста). Аудиовыход через SFML (воспроизведение звуковых эффектов и музыки).

Характеристики выходных наборов данных: Визуальное представление игрового мира (уровень, персонаж, препятствия, монеты, интерфейс). Звуковое сопровождение действий (прыжок, сбор монеты, смерть, фоновая музыка).

Описание выходных экранных форм.

Главное меню: Кнопки «Новая игра», «Магазин», «Настройки», «Выход», «Продолжить», «Об авторе», «Об игре».

Экран уровня – основной игровой экран. Отображает игровой мир (платформы, шипы, движущиеся блоки, монеты), персонажа, счетчик общих очков, кнопку паузы (ESC), портал (в следующий уровень).

Магазин – сетка или список доступных скинов с изображениями (превью), названиями, ценами в очках. Отображение текущего баланса игрока (очков). Кнопка «Купить» / «Установить» / «Установлено» для выбранного скина. Для возврата назад необходимо нажать ESC.

Настройки – ползунок для регулировки громкости музыки и звуков. Для возврата назад необходимо нажать ESC.

4. Входная информация:

Описание способов ввода данных. Управление с клавиатуры (клавиши стрелок/WAD для перемещения, ESC для паузы/меню). Управление мышью в меню и магазине (выбор кнопок, скинов). Загрузка данных при старте игры (уровни, текстуры, звуки, настройки).

5. Постоянная информация (нормативно-справочная информация):

Данные, не изменяемые в процессе игры, но необходимые для ее работы:

- физические константы (гравитация, сила прыжка, скорости движения блоков);
- стандартные параметры объектов (размер хит бокса шипа, урон от шипа (смерть), значение монеты (+10 очков));
- спрайт-листы (текстуры) персонажа, блоков, шипов, монет, фонов, элементов интерфейса;
- звуковые эффекты и фоновая музыка;
- шрифты для интерфейса.

Базовая структура уровней – до их загрузки из файлов.

Список и параметры скинов (до их загрузки из файла магазина).

Описание способа хранения: Хранится в виде файлов ресурсов (текстуры, звуки, шрифты) в папках проекта

6. Описание требований к составу выполняемых функций.

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Загрузка и инициализация ресурсов – текстуры, звуки, шрифты, уровни, настройки, данные магазина.

Управление игровыми состояниями: главное меню, игра (уровень), пауза, магазин, настройки, экран смерти.

Обработка пользовательского ввода – клавиатура, мышь.

Обновление игровой логики: расчет физики персонажа – движение, прыжки, гравитация, коллизии с платформами, блоками, полом.

Обновление состояния движущихся блоков – перемещение по траекториям.

Проверка коллизий: персонаж-шипы (смерть), персонаж-монеты (сбор, +очки), персонаж-движущийся блок, проваливающийся пол (падение).

Обновление таймеров – уровня, анимаций.

Отрисовка игрового состояния:

- рендеринг игровых объектов – персонаж с текущим скином, шипы, движущиеся блоки, монеты;

- рендеринг интерфейса – очки, монеты, таймер, меню, магазин.

Управление звуком – воспроизведение эффектов, музыки.

Сохранение и загрузка прогресса – текущий уровень, количество очков, купленные скины, активный скин, настройки, количество смертей, громкость звука.

Функционал магазина – просмотр скинов, покупка за очки. Применение – смена текстуры персонажа.

Реализация системы уровней – загрузка, завершение, переход к следующему.

1.2 Инструменты разработки

1. Выбор и обоснование среды разработки приложения:

Основной язык и библиотека: C++ и SFML (Simple and Fast Multimedia Library) 2.6.

C++ – мощный язык программирования, который используется для создания игр, системного ПО и высокопроизводительных приложений. Он поддерживает объектно-ориентированное, процедурное и обобщённое программирование, а его строгая типизация и прямой доступ к памяти делают его гибким и эффективным.

SFML 2.6 – удобная библиотека для работы с графикой, окнами, звуком и сетью. Она предоставляет простые API для рендеринга изображений, обработки событий и воспроизведения аудио

Visual Studio 2022 – мощная среда разработки с поддержкой C++20 и экспериментальными возможностями C++23. Она включает авто дополнение кода, удобный отладчик и инструменты для совместной работы.

2. Требования к аппаратным и операционным ресурсам.

Минимальная конфигурация:

- процессор: 1.5 GHz Dual-Core (Intel/AMD);
- оперативная память (RAM): 512 MB;
- видеокарта: любая с поддержкой OpenGL 3.3 (или выше) или DirectX 9.0c (через реализацию SFML для Windows). Интегрированной графики (Intel HD, AMD Radeon HD серии) обычно достаточно для простой 2D графики SFML;
- свободное место на диске: 1GB (зависит от количества уровней, текстур и звуков);
- операционная система: Windows 10 (или новее);
- оптимальная конфигурация (для комфортной игры);
- процессор: 2.0 GHz Dual-Core (Intel/AMD) или лучше;
- оперативная память (RAM): 1 GB или больше.

1.3 Требования к приложению

1. Накладываемые ограничения:

Платформа: настольные ОС. Отсутствие поддержки мобильных платформ и веб-браузеров.

Графика: исключительно 2D-графика. Отсутствие поддержки 3D-рендеринга.

Сетевая игра: отсутствие многопользовательского (сетевого) режима. Игра рассчитана на одного игрока локально.

Сложность уровней: уровни проектируются вручную. Отсутствие процедурной генерации уровней.

Объем контента: ограниченное количество изначально созданных уровней

15. Магазин скинов содержит конечный набор предметов.

Физика: упрощенная физическая модель (платформенная физика с гравитацией, прыжками).

2. Требования к интерфейсу:

Четкость и понятность: все элементы интерфейса (меню, кнопки, текст в HUD) должны быть легко читаемы и интуитивно понятны.

Минимализм (HUD): во время игры HUD должен отображать только критически важную информацию (счет очков, количество монет, счет смертей) без перегрузки экрана.

Единообразие: стиль интерфейса (шрифты, цвета кнопок, оформление окон) должен быть единым во всех разделах игры (меню, магазин, настройки, игровой экран).

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

Отзывчивость: кнопки и элементы интерфейса должны визуально реагировать на наведение курсора мыши и нажатие (изменение цвета, размера).

Управление: управление в игре должно быть точным и отзывчивым.

Назначение клавиш (Стрелки/WAD).

Игровой интерфейс:

Камера: определена изначально по центру экрана, обеспечивая хороший обзор необходимой области уровня. Возможность «предпросмотра» вверх/вниз (опционально).

Визуальная обратная связь: четкие визуальные сигналы о событиях: мигание персонажа при получении урона (смерти), анимация сбора монеты, визуальное выделение активного скина в магазине, анимация движения блоков и исчезновения пола.

3. Средства защиты:

Защита файлов данных:

Чтение/запись: обеспечить корректное чтение и запись файлов уровней, настроек и данных магазина. Проверка целостности файлов при загрузке (базовая валидация структуры).

Предотвращение коррупции: обработка ошибок ввода/вывода при работе с файлами, чтобы избежать краша игры или потери данных при сбоях (например: не правильное загрузка изображения)

Простые меры (опционально, для учебного проекта): сохранение данных магазина (баланс, купленные скины) в файле с проектом и использование простого шифрования/кодирования (Бинарное) содержимого файла сохранений, чтобы затруднить прямое редактирование игроком. Замечание: Полной защиты от читерства это не обеспечивает, но усложняет его.

Устойчивость к некорректному вводу: обработка всех возможных действий пользователя (нажатие не тех клавиш, клики мимо кнопок) без падения приложения. Корректное восстановление после ошибок (например, невозможность загрузить уровень - вывод сообщения и возврат в меню).

2 Проектирование

2.1 Организация данных

В рамках реализации игрового приложения «Валли» данные можно разделить на три категории:

Входная информация – данные, необходимые для инициализации уровня и игрового процесса (спрайты, макет уровня, параметры объектов).

Условно-постоянная информация – игровые константы и настройки (скорость передвижения, стоимость скинов, и т.д.).

Выходная информация – сохраняемые данные прогресса пользователя (уровень, накопленные очки, купленные скины).

Логическая структура.

Список объектов:

- шипы (позиция, активен/неактивен);
- блок для движения (путь движения, скорость);
- проваливающейся пол (координаты «открытого» пола);
- монеты (координаты, собран/не собран).

Персонаж :

- текущие координаты и скорость;
- набранные очки;
- активный скин.

Магазин скинов: список доступных скинов: идентификатор, стоимость.

Сохранение:

- текущий уровень;
- общие очки;
- количество смертей;
- список купленных скинов;
- купленные скина.

Физическая структура: все сохранения будут храниться в файле с проектом в бинарной кодировке.

Динамические структуры:

- во время игры объекты создаются и удаляются по мере необходимости;
- при рестарте уровня – пересоздаются все игровые объекты;
- монеты удаляются из списка при сборе.

2.2 Процессы

Инициализация:

- загрузка сохранения с файла savegame.dat;

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

- загрузка ресурсов уровня (lvl1-15.tmx);
- установка игрока в стартовую точку.

Обработка ввода персонажа: функция void control() реагирует на клавиши;

Обновление состояния мира:

- движение персонажа;
- применение физики – гравитация, трение;
- коллизии с тайлами;
- движение блоков;
- перемещение вдоль заданного пути;
- проверка столкновений;
- шипы – если пересечение границ;
- монеты – при пересечении сбор монеты.

Рендеринг:

- отрисовка тайлов , объектов, интерфейса;
- магазин скинов;
- главном меню или после уровня вызывается;
- отображение списка доступных скинов;
- при покупке проверка хватает ли монет на покупку, списание очков.

Сохранение прогресса:

- при выходе из игры сохраняем прогресс и скины в файл;

Переход на следующий уровень:

- при достижении финиша;
- увеличение текущего уровня;
- загрузка следующего уровня.

2.3 Описание внешнего пользовательского интерфейса

Структура навигации по проекту представлена на рисунке 2.1.

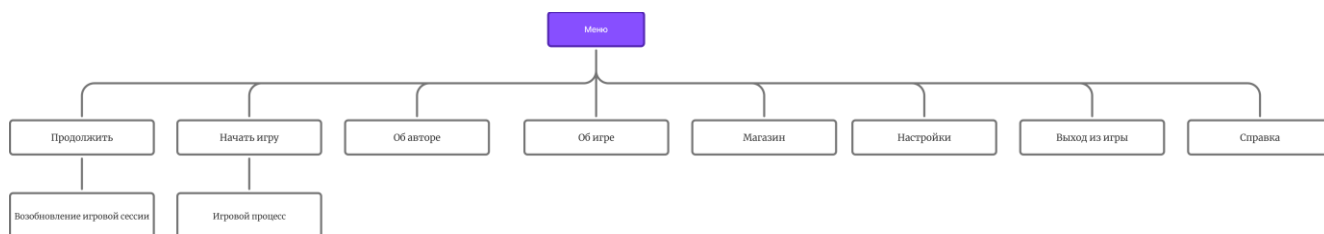


Рисунок 2.1 – Структура навигации по проекту

1.Загрузочный экран (рисунок 2.2):

- светло фиолетовый фон ;

- загрузочная полоса пурпурного цвета ,идущая от левого к правому краю;
- логотип: название «Валли» крупным шрифтом, полупрозрачным тёмно-фиолетовым цветом;
- спрайт: спрайт бежит по полосе.

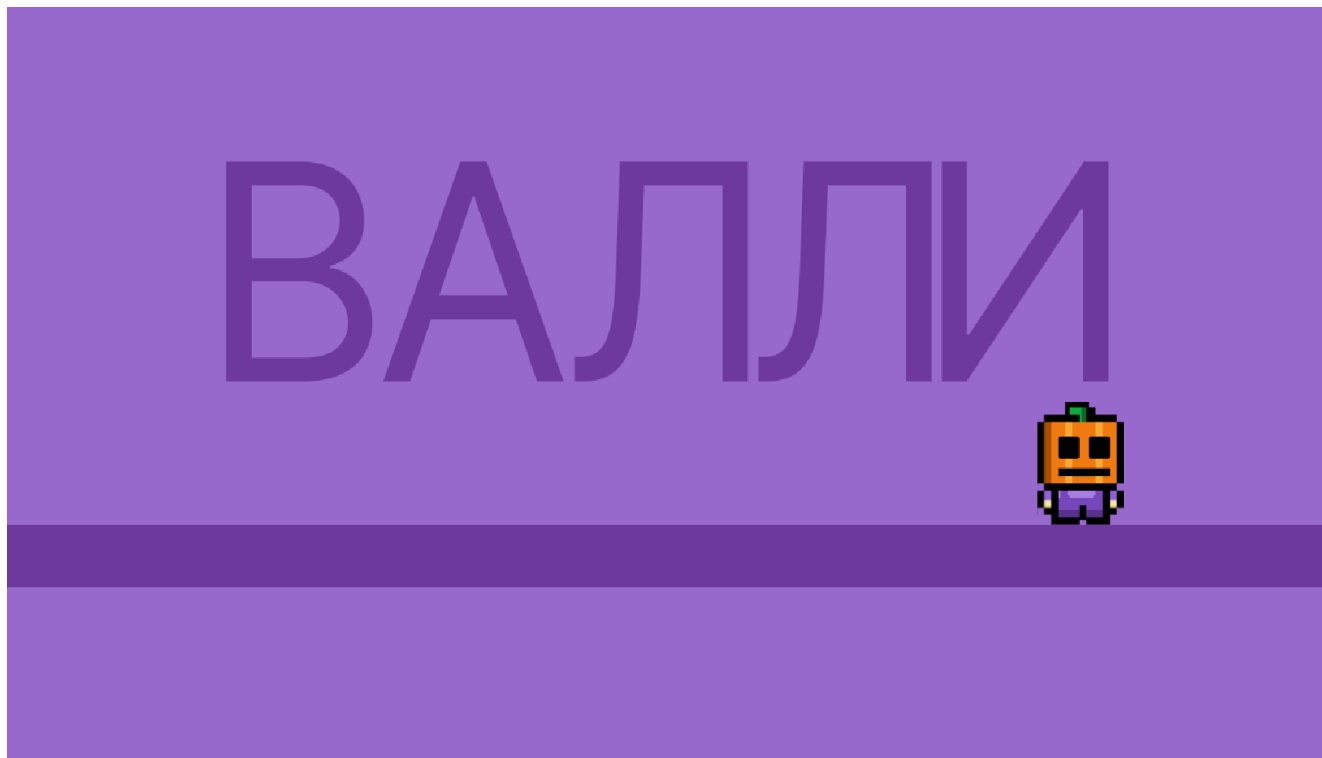


Рисунок 2.2 – Загрузочный экран

2. Меню (рисунок 2.3):

- светло фиолетовый фон ;
- логотип: название «Валли» крупным шрифтом, полупрозрачным тёмно-фиолетовым цветом;
- главные кнопки: расположены по центру вертикальным списком. Каждая кнопка представляет скруглённый прямоугольник с белой обводкой и светлым шрифтом: «Продолжить», «Новая Игра», «Об Авторе», «Об Игре», «Магазин», «Настройки», «Выход», «Справка»;
- декоративные спрайты: по углам экрана – пиксельные персонажи (зеленый, тыквенный, арбузный) и центральный портал.

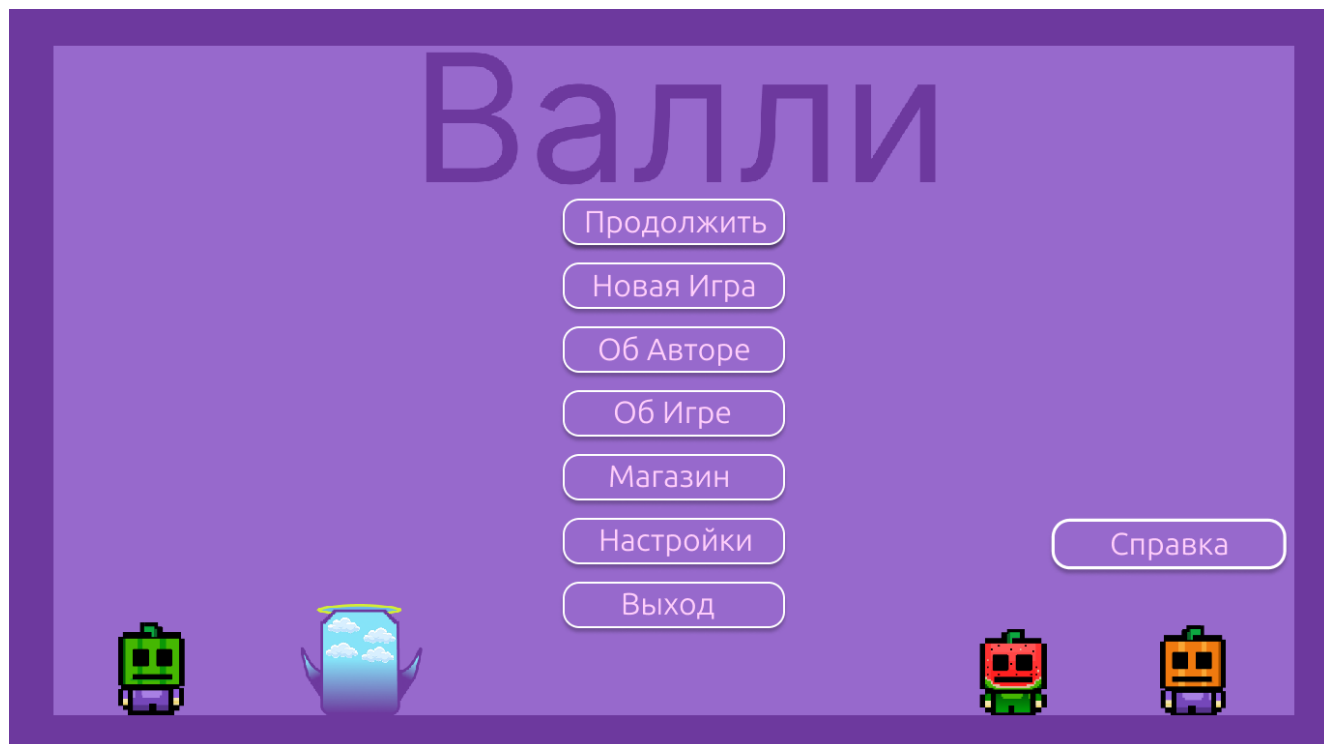


Рисунок 2.3 – Меню

3. Окно «Магазин» (рисунок 2.4):

- заголовок: крупно вверху – «Магазин»;
- счётчик очков: в правом углу – «POINTS: XX»;
- область товаров: большой полупрозрачный прямоугольник закруглёнными углами, внутри – карточки скинов.

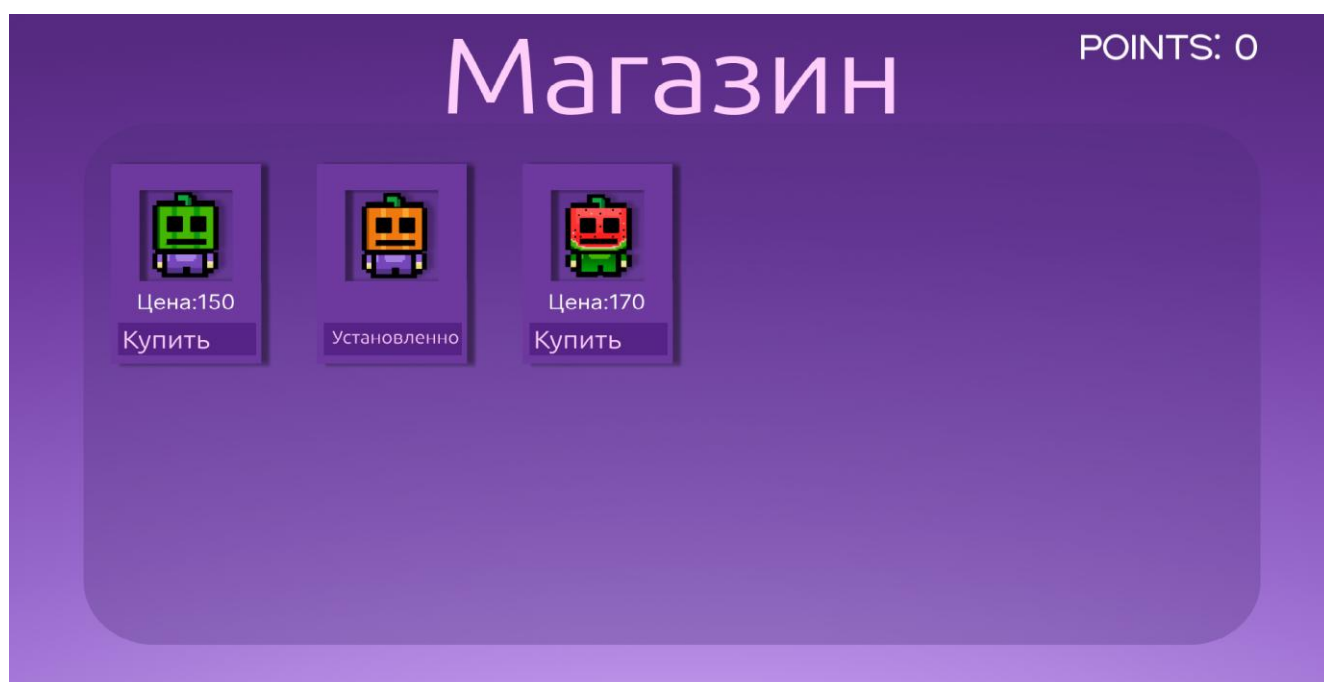


Рисунок 2.4 – Магазин

4. Окно «Об Игре» (рисунок 2.5):

- заголовок: «Об Игре» крупным светло-розовым шрифтом;
- клавиши управления: стрелки или клавиши A, W, D ;
- описание валюты : пиктограмма золотой монеты и текст.

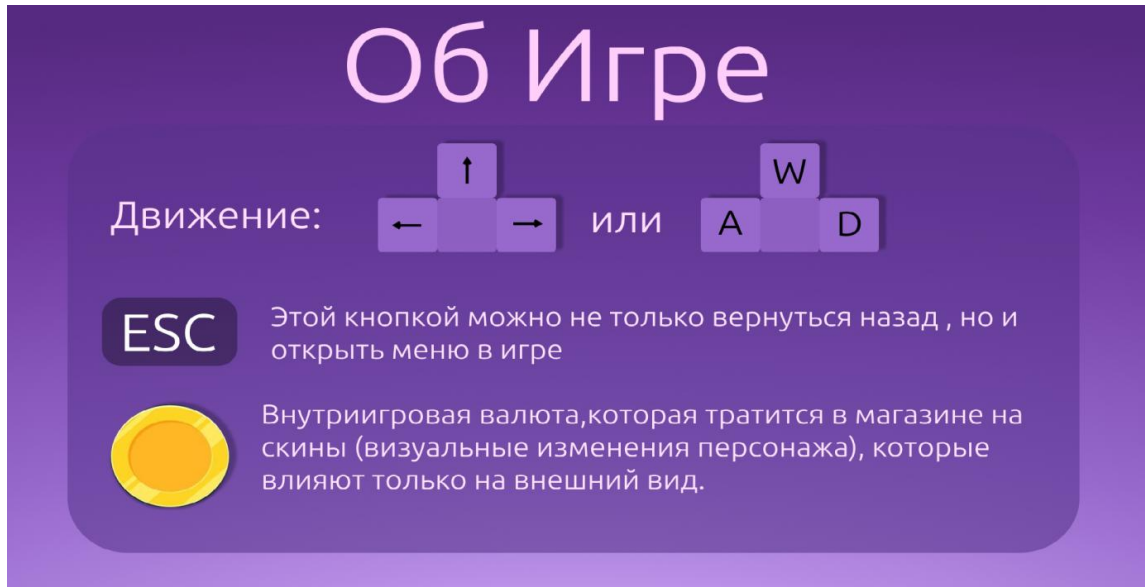


Рисунок 2.5 – Об Игре

5. Окно «об Авторе» (рисунок 2.6):

- светло-фиолетовый фон;
- текст: информация об авторе и шутка о багах, оформленная стандартным шрифтом.

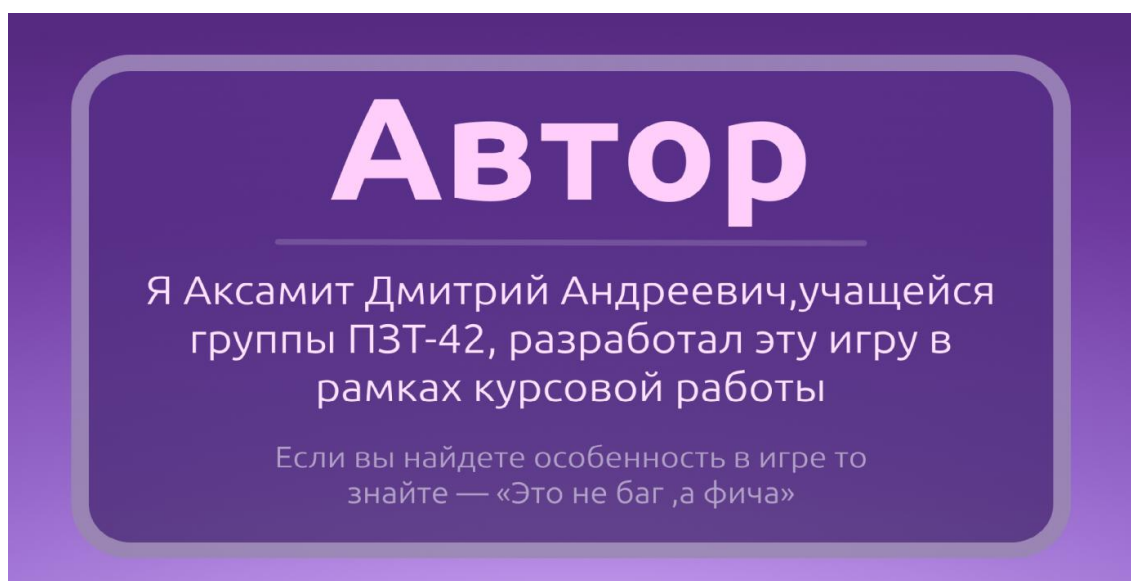


Рисунок 2.6 – Об Авторе

6. Окно «Настройки» (рисунок 2.7):

- громкости: музыка и звуки;

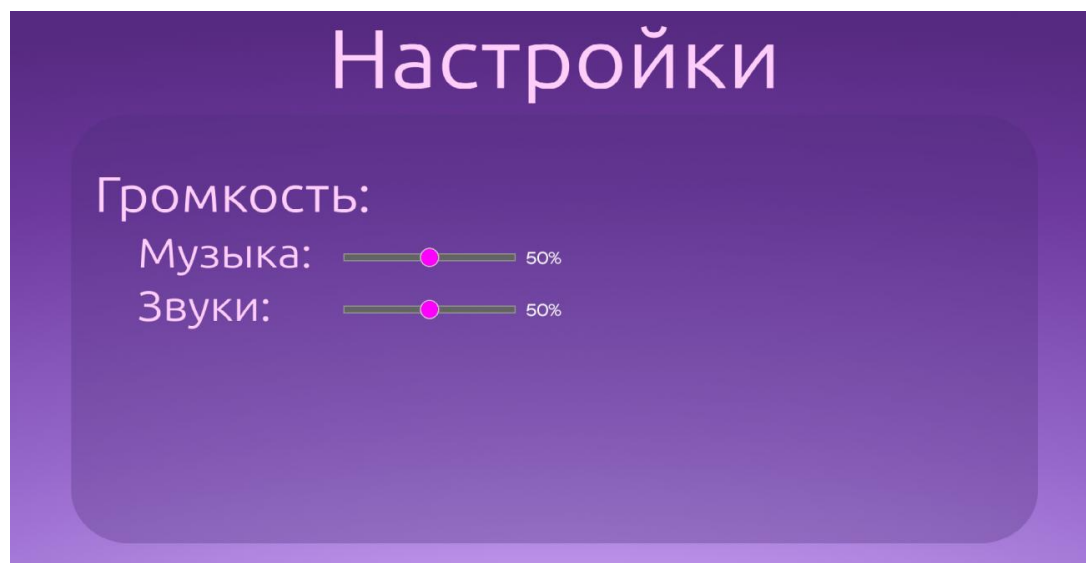


Рисунок 2.7 – Настройки

6. Окно «ПОБЕДА!!!» (рисунок 2.8):

- фон: светло-фиолетовый;
- текст: поздравление с победой, информация об авторе и шутка о нервах игрока.

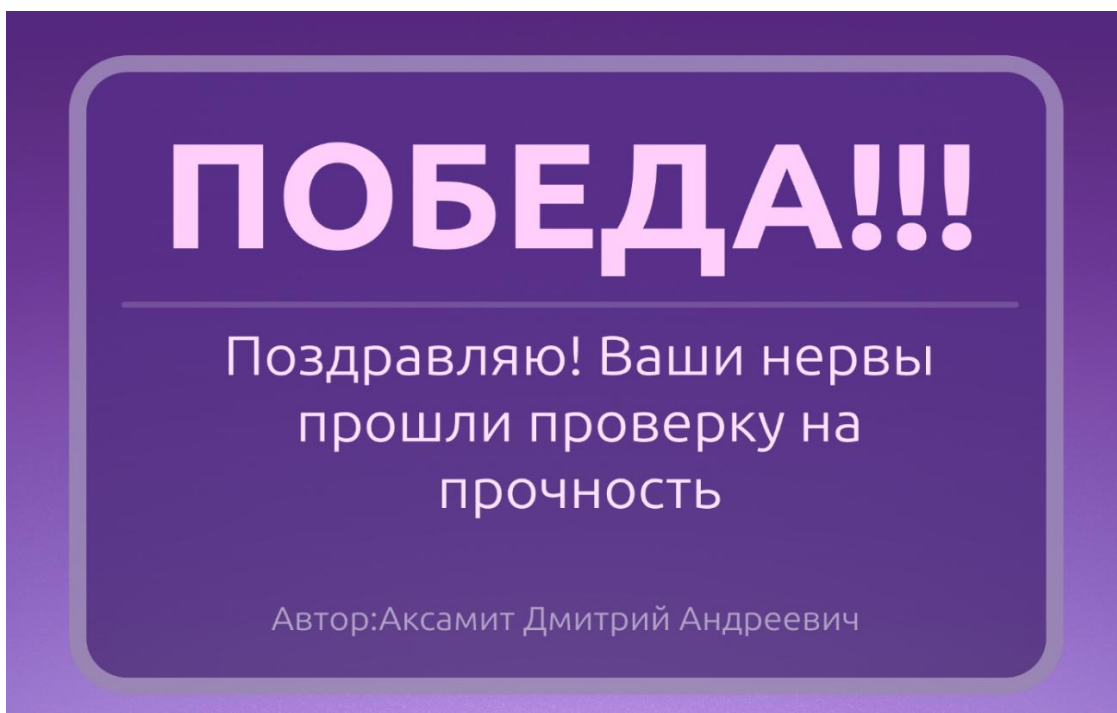


Рисунок 2.8 – Победа

3. Реализация

3.1 Структура программы

Данный курсовой проект содержит 9 модулей. Далее рассмотрим назначение каждого модуля:

1. main - главный модуль, точка входа, управление игровым циклом
2. Entity - базовый класс для всех игровых объектов
3. Player- класс игрока (наследник Entity)
4. Environment – игровые объекты окружения (наследники Entity): MovingPlatform, Portal, FailurePlatform, HardPlatform, Spike, Coin.
5. Level - загрузка и управление игровыми уровнями
6. View- управление игровой камерой
7. UI- пользовательский интерфейс: Menu, LoadingScreen, PauseMenu, Shop, Settings.
8. Audio - управление звуками и музыкой
9. SaveSystem - система сохранения/загрузки игры

3.1.1 Структура и описание разработанных процедур и функций

Описание разработанных процедур и функций приводится в таблице 1.

Таблица 1 – Структура и описание разработанных процедур и функций

Имя процедуры (функции)	В каком модуле находится	За каким элементом управления закреплена	Назначение
1	2	3	4
main()	main	Точка входа	Инициализация приложения
showLoadingScreen()	UI	LoadingScreen	Отображение анимированного загрузочного экрана
menu()	UI	MainMenu	Отрисовка главного меню и управление
startGame()	Game	GameCore	Запуск и управление игровым процессом
Player::control()	Player	Player	Обработка управления игроком
Player::update()	Player	Player	Обновление состояния игрока
Player::die()	Player	Player	Обработка смерти игрока

Продолжение таблицы 1

1	2	3	4
Player::AddScore()	Player	Player	Добавление очков игроку
Player::completeLevel()	Player	Player	Завершение уровня
Coin::update()	Environment	Coin	Анимация вращения монетки
changeLevel()	Level	LevelManager	Загрузка и переключение уровней
saveGame()	SaveSystem	SaveManager	Сохранение прогресса (уровень, очки, монеты)
loadGame()	SaveSystem	SaveManager	Загрузка сохраненной игры
Slider::handleEvent()	UI	Slider	Обработка взаимодействия с ползунком
Slider::updateThumbPosition()	UI	Slider	Позиционирование ползунка слайдера
Slider::setPosition()	UI	Slider	Установка позиции слайдера
Slider::setWidth()	UI	Slider	Установка ширины слайдера
Portal::checkActivation()	Environment	Portal	Активация портала при приближении игрока
HardPlatform::activate()	Environment	HardPlatform	Активация движения сложных платформ
Spike::activate()	Environment	Spike	Активация шипов при приближении игрока
Spike::getActivationXStart()	Environment	Spike	Получение границы зоны активации (X start)
Spike::getActivationXEnd()	Environment	Spike	Получение границы зоны активации (X end)
Enemy::update()	Enemy	Enemy	Обновление состояния врага
Enemy::checkCollisionWithMap()	Enemy	Enemy	Проверка столкновений врага с картой
MovingPlatform::update()	Environment	MovingPlatform	Обновление позиции движущейся платформы
failure::activate()	Environment	failure	Активация падения платформы
failure::activate()	Environment	failure	Активация падения платформы
gameRunning()	main	GameCore	Основной цикл выполнения игры

3.1.2 Описание использованных компонентов

Описание использованных при разработке приложения компонентов приводиться в таблице 2

Таблица 2 – Описание использованных компонентов

Компонент	На какой форме расположено	Назначение
Entity	Базовый класс для всех игровых объектов (персонажи, платформы, предметы)	Все игровые уровни
Player	Управляемый игровой персонаж с физикой, анимацией и системой здоровья	Игровые уровни
MovingPlatform	Движущиеся платформы с циклическим перемещением	Игровые уровни
Portal	Телепорты для перехода между уровнями	Игровые уровни
failure	Падающие платформы, активируемые игроком	Игровые уровни
HardPlatform	Платформы со сложной траекторией движения	Игровые уровни
Spike	Шипы, наносящие урон игроку	Игровые уровни
Coin	Собираемые монеты с анимацией	Игровые уровни
Slider	Элемент UI для регулировки громкости	Меню настроек
Text	Отображение текстовой информации (счет, уровень, смерти)	Все экраны интерфейса
Компонент	На какой форме расположено	Назначение
Music	Фоновая музыка	Главное меню и игровые уровни
Clock	Управление временем для анимации, движения и таймеров	Все системы
View	Управление игровой камерой и областью видимости	Игровые уровни
RenderWindow	Основное окно рендеринга игры	Весь проект
ProgressBar	Визуализация прогресса загрузки	Экран загрузки
Button (Sprite)	Интерактивные элементы интерфейса	Все меню (главное, магазин, настройки)
Sprite	Визуальное представление игровых объектов	Все экраны
Sound/SoundBuffer	Воспроизведение звуковых эффектов (прыжки, смерть, сбор монет)	Игровые уровни и меню

3.2 Спецификация программы

Таблица 3 – Описание использованных файлов

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

Имя файла	Назначение
Bykva.wav	Звук появления букв в загрузочном экране
button.ogg	Звук при наведении на кнопки в меню
na_fone.wav	Фоновая музыка игры
jump.wav	Звук прыжка игрока
dead.wav	Звук смерти игрока
hard.wav	Звук активации платформ
fall.wav	Звук падения платформы
coin.wav	Звук сбора монеты
Zaryzka.png	Фон загрузочного экрана
polosa.png	Изображение полосы загрузки
zagryzakapers.png	Анимированный персонаж на загрузочном экране
B.png	Буква «В» для слова «ВАЛЛИ»
A.png	Буква «А» для слова «ВАЛЛИ»
L.png	Буква «Л» для слова «ВАЛЛИ»
I.png	Буква «И» для слова «ВАЛЛИ»
StartGame.png	Кнопка «Начать игру»
StartGameh.png	Кнопка «Начать игру» (состояние при наведении)
Avtor.png	Кнопка «Автор»
Avtorh.png	Кнопка «Автор» (состояние при наведении)
Ob Game.png	Кнопка «Об игре»
Ob Gameh.png	Кнопка «Об игре» (состояние при наведении)
Exit.png	Кнопка «Выход»
Exith.png	Кнопка «Выход» (состояние при наведении)
Continue.png	Кнопка «Продолжить»
Continueh.png	Кнопка «Продолжить» (состояние при наведении)
shop.png	Кнопка «Магазин»
shoph.png	Кнопка «Магазин» (состояние при наведении)
settings.png	Кнопка «Настройки»
settingsh.png	Кнопка «Настройки» (состояние при наведении)
about.png	Экран информации об авторе
aboutGame.png	Экран информации об игре
menu.png	Фон главного меню
skins.png	Фон магазина скинов
installed.png	Кнопка «Установлено» в магазине
by.png	Кнопка «Купить» в магазине
pers.png	Скин персонажа по умолчанию
pers2.png	Альтернативный скин персонажа
pers3.png	Альтернативный скин персонажа
shamaich.png	Изображение врага
spike.png	Изображение шипов
portalmy2.png	Изображение портала
failure.png	Изображение падающей платформы
coin.png	Изображение монеты (анимированное)
Menubtn.png	Кнопка «Меню» в паузе
Menubtnh.png	Кнопка «Меню» в паузе (состояние при наведении)
settingsmenu.png	Фон меню настроек
lvl1.tmx - lvl15.tmx	Файлы уровней игры (15 уровней)

Продолжение таблицы 3

1	2
Text.ttf	Основной шрифт для текста в интерфейсе
savegame.dat	Файл сохранения игры (прогресс, скин, настройки)
view.h	Управление камерой и видами
level.h	Работа с уровнями и загрузкой карт
tinysql.h	Библиотека для работы с XML-файлами

4 Тестирование

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в таблице 4.

Таблица 4 – Отчёт результатах тестирования

№	Тест	Ожидаемый результат	Физический результат	Результат тестирования
1	Проверка кнопки «Новая игра»	Сброс прогресса, начало с 1 уровня	Сброс прогресса, начало с 1 уровня	Выполнено
2	Проверка кнопки «Продолжить»	Загрузка сохраненной игры	Загрузка сохраненной игры	Выполнено
3	Проверка кнопки «Магазин»	Открытие экрана магазина скинов	Открытие экрана магазина скинов	Выполнено
4	Проверка кнопки «Настройки»	Открытие меню настроек громкости	Открытие меню настроек громкости	Выполнено
5	Проверка кнопки «Об авторе»	Открытие информации об авторе	Открытие информации об авторе	Выполнено
6	Проверка кнопки «Об игре»	Открытие информации об игре	Открытие информации об игре	Выполнено
7	Проверка кнопки «Выход»	Заккрытие приложения	Заккрытие приложения	Выполнено
8	Проверка перехода между уровнями (1-15)	Корректная загрузка каждого уровня	Корректная загрузка каждого уровня	Выполнено
9	Проверка шипов	Смерть игрока при касании	Смерть игрока при касании	Выполнено
10	Проверка порталов	Переход на следующий уровень	Переход на следующий уровень	Выполнено
11	Проверка кнопки «Справка»	Открытие справки	Открытие справки	Выполнено

Продолжение таблицы 4

1	2	3	4	5
12	Проверка падающих платформ	Падение платформы после активации	Падение платформы после активации	Выполнено
13	Проверка сложных платформ	Движение по заданной траектории	Движение по заданной траектории	Выполнено
14	Проверка сбора монет	Увеличение счета, исчезновение монеты	Увеличение счета, исчезновение монеты	Выполнено
15	Проверка фоновой музыки	Воспроизведение музыки только на уровнях	Воспроизведение музыки только на уровнях	Выполнено
16	Проверка системы сохранений	Сохранение прогресса между сессиями	Сохранение прогресса между сессиями	Выполнено
17	Проверка инвертированного управления	Инверсия управления на уровнях 9-10	Инверсия управления на уровнях 9-10	Выполнено
18	Проверка загрузочного экрана	Корректное отображение анимации	Корректное отображение анимации	Выполнено
19	Проверка экрана паузы	Отображение меню паузы при нажатии ESC	Отображение меню паузы при нажатии ESC	Выполнено
20	Проверка звуковых эффектов	Воспроизведение звуков взаимодействий	Воспроизведение звуков взаимодействий	Выполнено

При разработке программного продукта было решено множество проблем, например, неправильная задача координат активации – платформ , падающего пола, шипов , портала.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

5 Применение

5.1 Общие сведения о программном продукте

Цель проекта – создание игрового продукта, где игрок управляет персонажем, преодолевая уровни с неожиданными препятствиями.

Разработанная игра ориентирована на любителей аркадных платформеров всех возрастов – от школьников до взрослых, желающих позабавиться и проверить свои навыки. Игровая система интуитивно понятна, не требует специальных знаний, но предоставляет достаточно вызовов и стимул для повторных прохождений и достижения лучших результатов.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на разных классах ЭВМ и работать с данной программой было комфортно. Для корректной работы игрового приложения «Валли» рекомендовано использовать ПК со следующими характеристиками:

- процессор 12th Gen Intel(R) Core(TM) i5-12450H;
- объем ОЗУ 32Гб;
- операционная система Windows 10;
- видеокарта GeForce RTX 3050 6 GB.

Также тестирование проходило на ПК со следующими характеристиками:

- процессор AMD Ryzen 7 4700U with Radeon Graphics, 2000 МГц, ядер: 8, логических процессоров: 8;
- объем ОЗУ 16Гб;
- графический адаптер AMD Radeon Graphics Processor (0x1636), Advanced Micro Devices, Inc.-совместимый;
- операционная система Windows 10.

5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл Валли Setup.exe. Появится окно установки приложения «Валли». Затем достаточно следовать приведенной инструкции установки приложения.

Подробная инструкция:

- запустите установщик;
- выберите язык установки;
- согласитесь с лицензией;
- выберете путь установки программы;
- поставьте галочку если хотите создать ярлык на рабочем столе;
- нажмите установить;
- при завершении установки нажмите завершить.

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		23

5.3 Выполнение программы

5.3.1 Запуск программы

Данную программу можно запустить различными способами.

Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием «Валли» .

Вторым способом является запуск из каталога, в который пользователь установил приложение (по умолчанию C:\Program Files\Валли) .

5.3.2 Инструкции по работе с программой

После запуска приложения на экране нас встречает загрузочный экран . (рисунок 5.1).

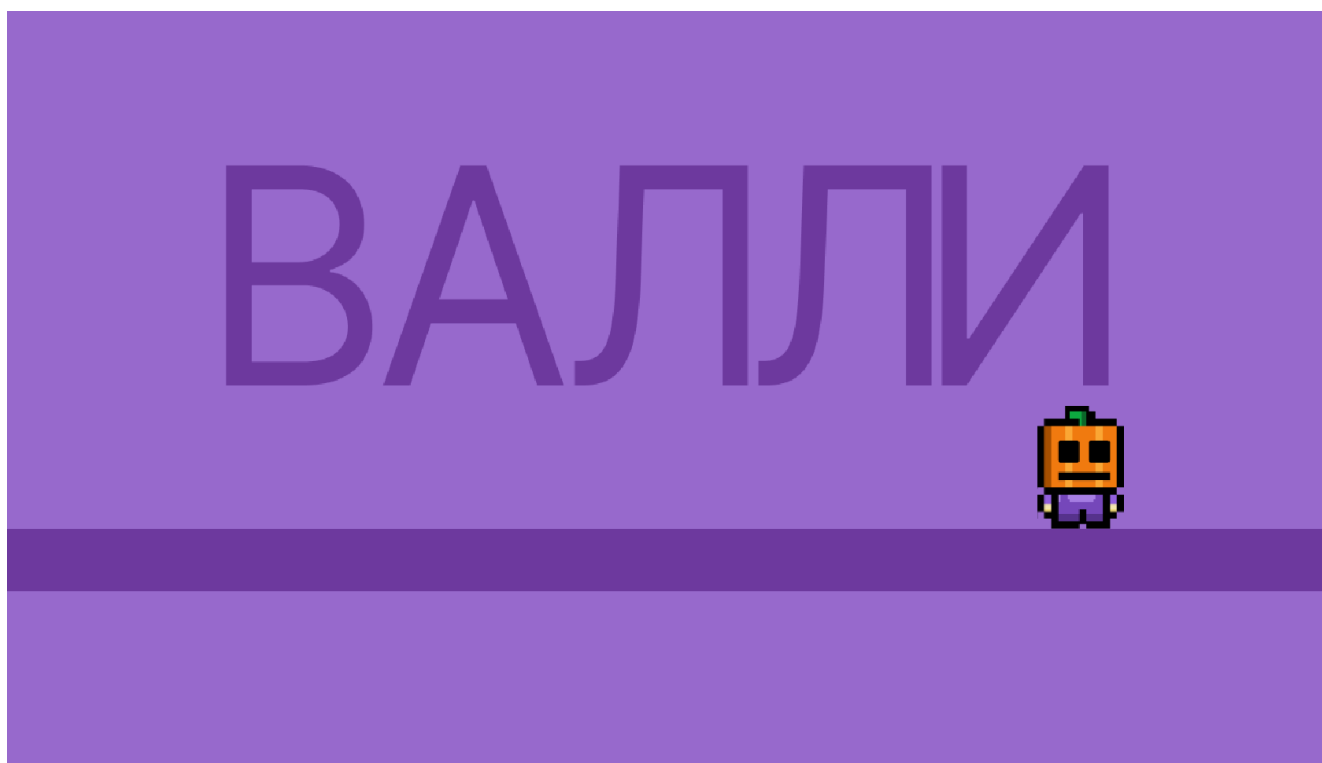


Рисунок 5.1 – Загрузочный экран

После загрузочного экрана нас встречает главное меню (рисунок 5.2).

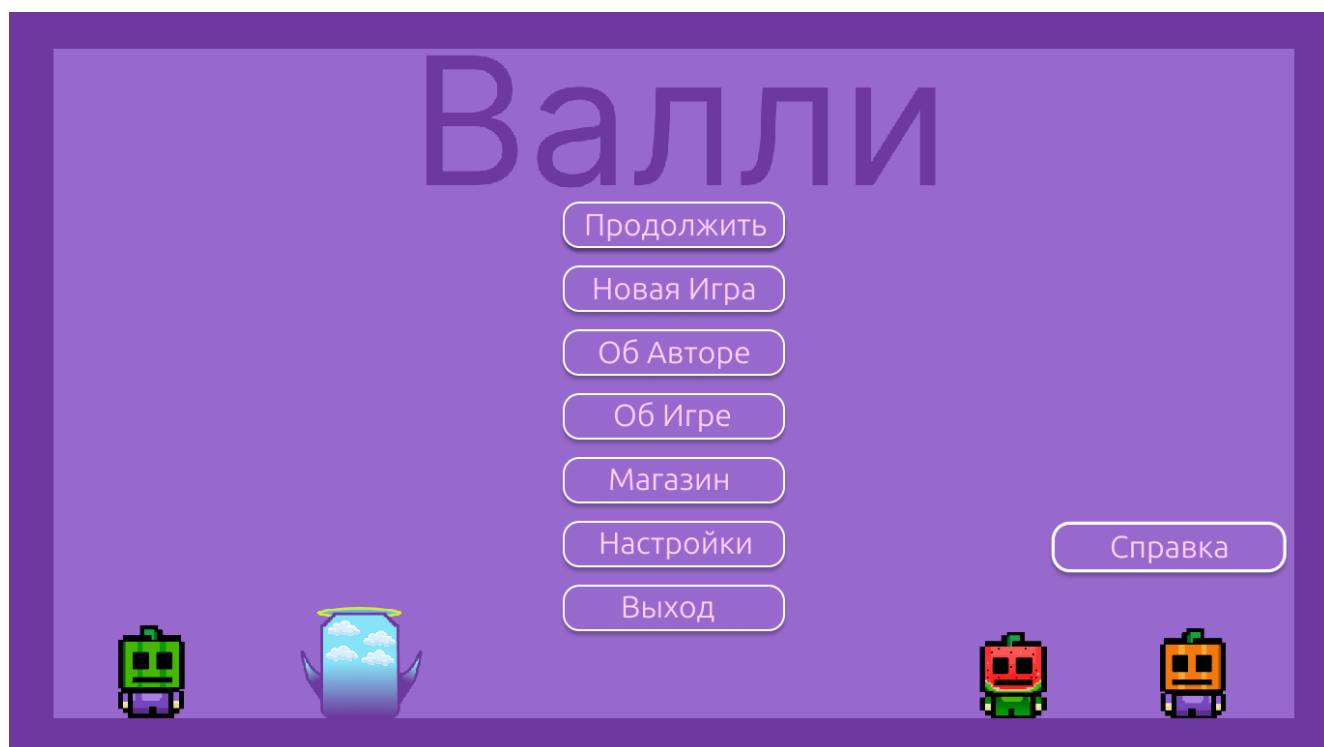


Рисунок 5.2 – Главное меню

На главном меню 8 кнопок – «Новая игра», «Магазин», «Настройки», «Выход», «Продолжить», «Об авторе», «Об игре», «Справка».

При нажатии на кнопку – «Новая игра» вы запустите игру.

При нажатии на кнопку – «Продолжить» вы продолжите игру.

При нажатии на кнопку – «Выход» вы выйдете из игры.

При нажатии на кнопку – «Об Авторе» вы перейдете в окно Об Авторе.
(рисунок 5.3).

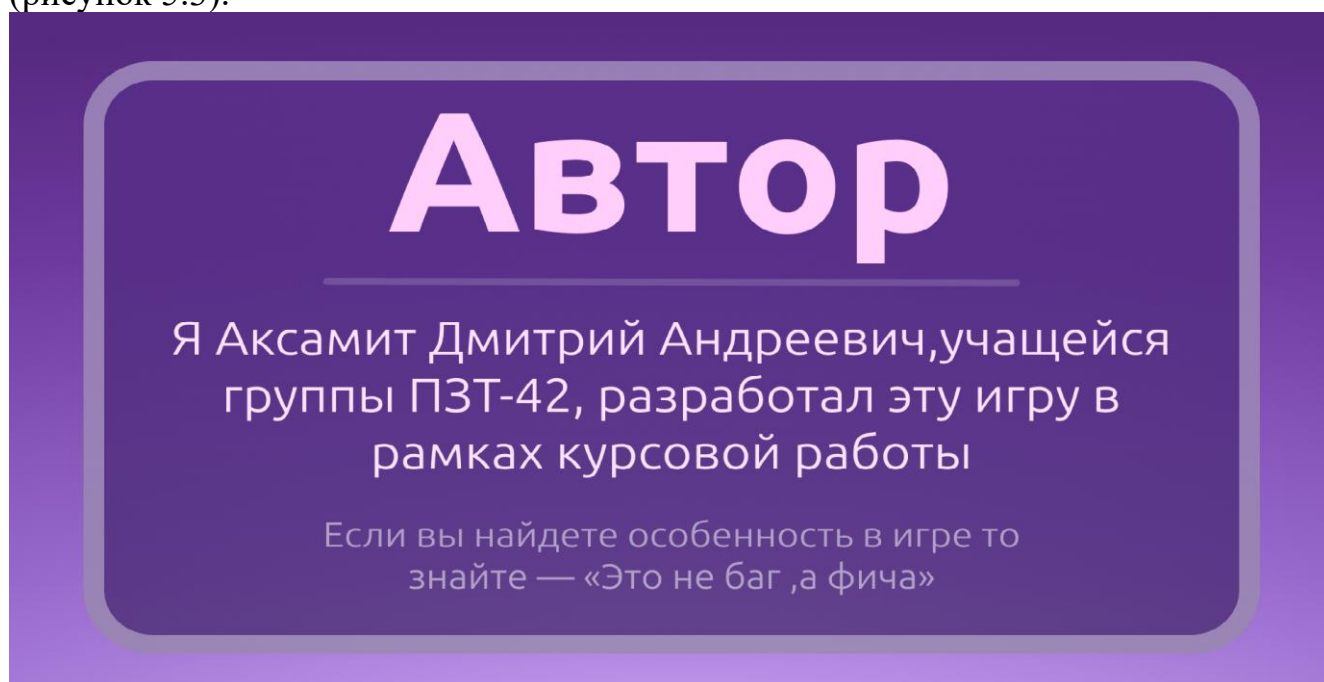


Рисунок 5.3 – Об Авторе

При нажатии на кнопку – «Об Игре» вы перейдете в окно Об Игре. (рисунок 5.4).

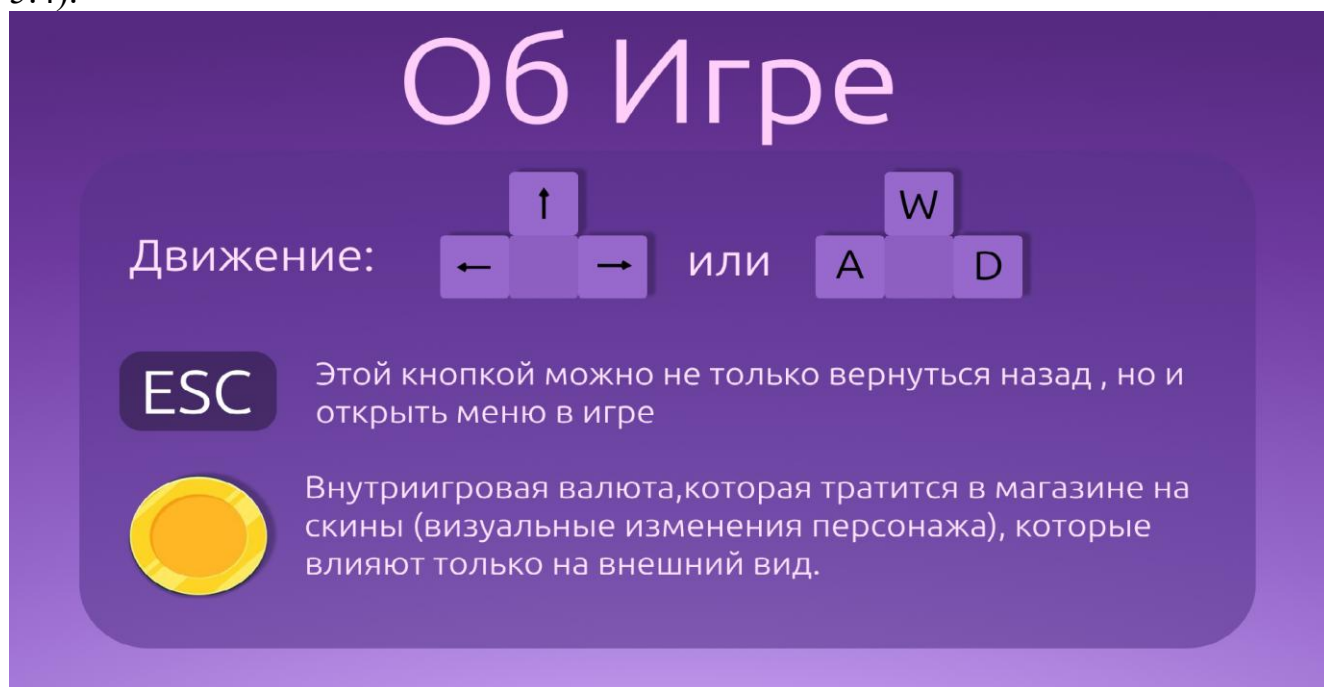


Рисунок 5.4 – Об Игре

При нажатии на кнопку – «Магазин» вы перейдете в окно Магазин. (рисунок 5.5).

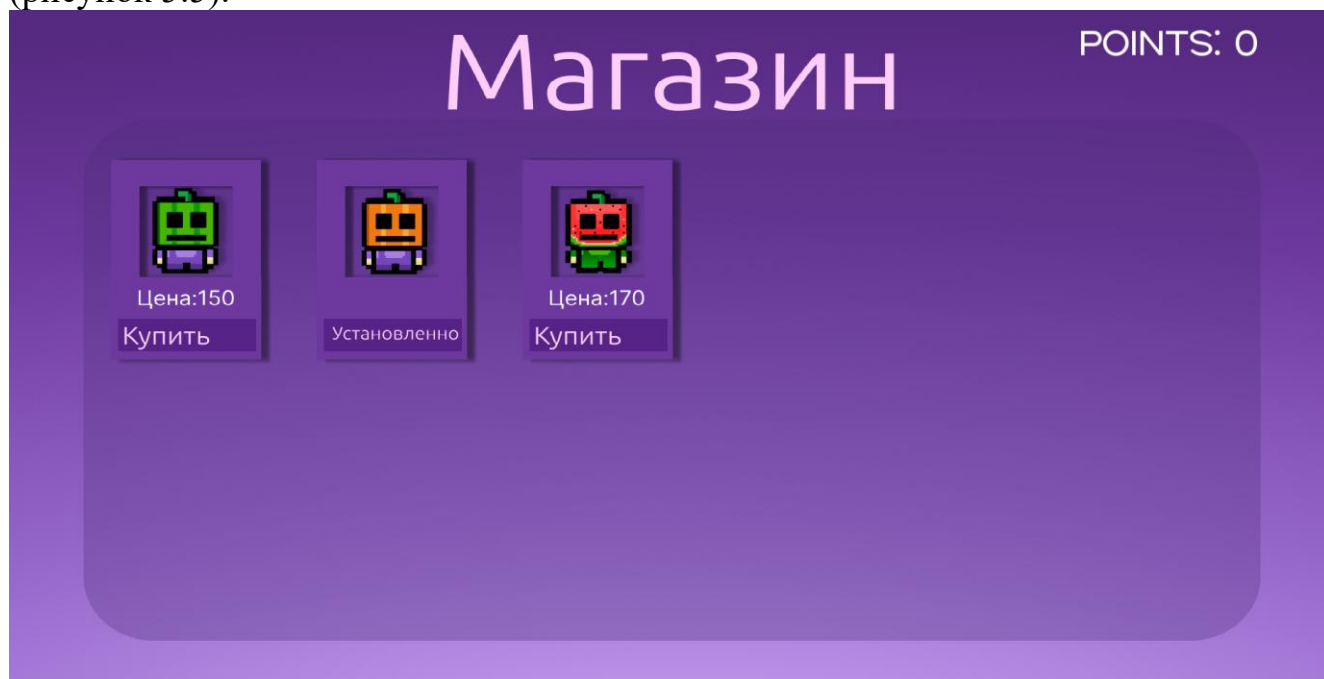


Рисунок 5.5 – Магазин

При нажатии на кнопку – «Настройки » вы перейдете в окно Настройки. (рисунок 5.6)

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

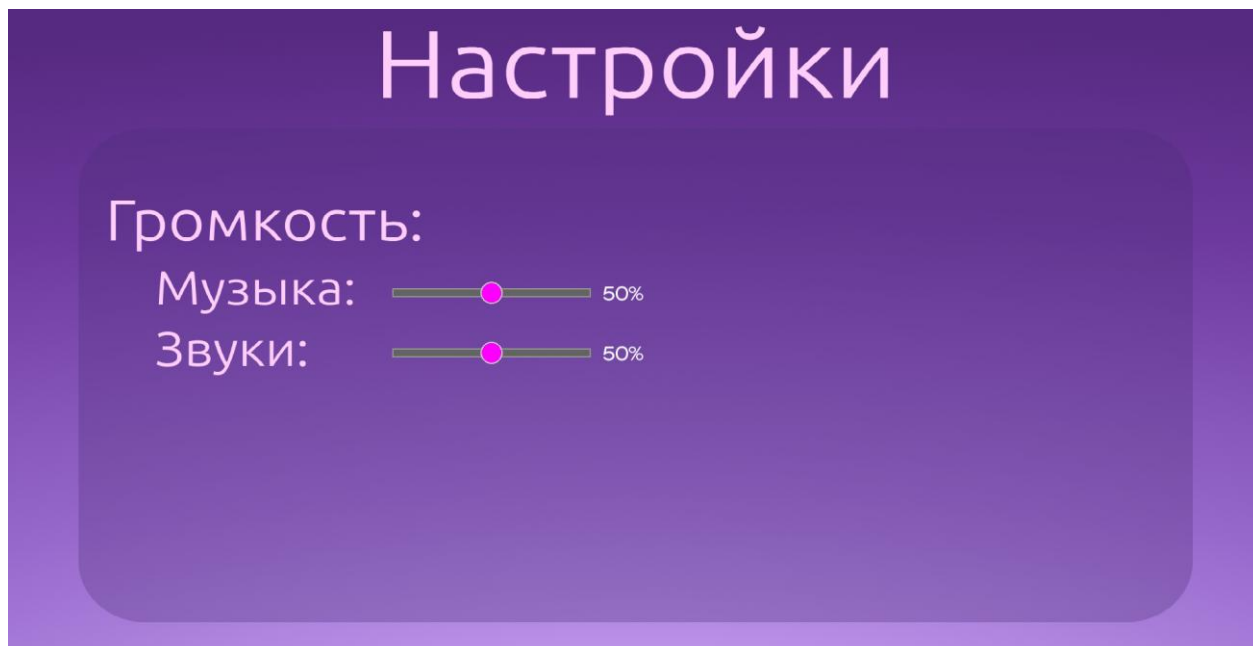


Рисунок 5.6 – Настройки

При нажатии на кнопку – «Справка» откроется модальное окно справка. (рисунок 5.7)

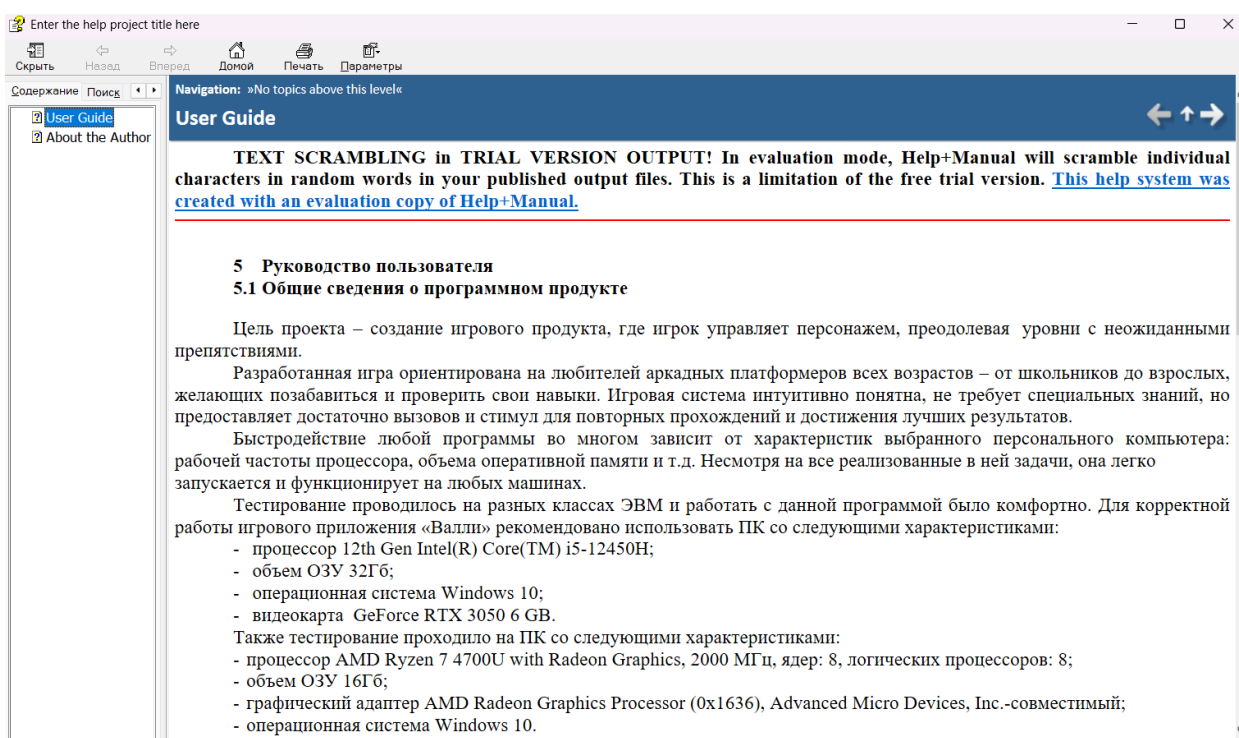


Рисунок 5.7 – Справка

5.3.3 Завершение работы с программой

Завершить работу с программой можно несколькими способами:

1. Если вы в меню нажмете кнопку «Выход».

2. Также можно закрыть игровое приложение в игре нажав кнопку ESC, а далее нажав «Выход». (рисунок 5.8).

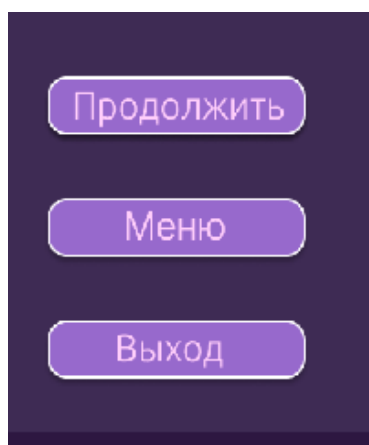


Рисунок 8 – Меню в игре

Заключение

В рамках курсового проекта было разработано игровое приложение «Валли», которое относится к играм 2D-платформер. Основная задача заключалась в создании полнофункционального игрового продукта, где игрок управляет персонажем, преодолевая тщательно спроектированные уровни, наполненные разнообразными препятствиями и системой сбора монет для внутриигровой экономики.

Разработанная игра рассчитана на широкую аудиторию любителей аркадных платформеров – от школьников до взрослых игроков, ищущих вызов. Интуитивно понятное управление и визуальная обратная связь делают игру доступной для новичков, в то время как сложность уровней и необходимость точного контроля обеспечивают глубину и реиграбельность для опытных игроков. Система скинов в магазине добавляет элемент кастомизации и долгосрочной мотивации.

В качестве дальнейшего развития могут быть добавлены новые уровни с новыми механиками.

При разработке игрового приложения «Валли» был применен объектно-ориентированный подход, наиболее эффективный для моделирования сложных взаимодействий игровых сущностей (персонажа, препятствий, объектов). Программная реализация выполнена на языке C++ с применением мультимедийной библиотеки SFML 2.6 в среде разработки Microsoft Visual Studio 2022.

В заключение можно констатировать, что программный продукт «Валли» представляет собой законченное, полнофункциональное игровое приложение, успешно реализующее поставленные задачи.

Список использованных источников

1. Официальная документация SFML 2.6 <https://www.sfml-dev.org/documentation/2.6.0/> — Дата доступа: 10.05.2025-24.06.2025
2. Канал «Павел Букреев» <http://youtube.com/@kychkaaa> — Дата доступа: 10.05.2025-24.06.2025
3. Официальная сайт Figma <https://www.figma.com/> — Дата доступа: 10.05.2025-24.06.2025
4. Канал «SimpleCode» <https://www.youtube.com/@SimpleCodeIT> — Дата доступа: 20.01.2025-24.06.2025.

					КП 5-04-0612-02.42.01.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30