

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
" ____ " _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Компьютерная игра «Dolly»

Руководитель	_____	_____	_____	Л.И. Покидышева
	<i>подпись</i>	<i>дата</i>	<i>доцент, к. т. н.</i>	
			<i>должность, ученая степень</i>	
Выпускник	_____	_____		А.Е. Мельникова
	<i>подпись</i>	<i>дата</i>		
Нормоконтролер	_____	_____		Л.И. Покидышева
	<i>подпись</i>	<i>дата</i>		

Красноярск 2023

РЕФЕРАТ

Выпускная квалификационная работа по теме «Компьютерная игра «Dolly»» содержит 40 страниц текстового документа, 24 иллюстрации, 7 использованных источников.

КОМПЬЮТЕРНАЯ ИГРА, C#, UNITY, ХИМИЯ, 2D-ИГРА, ASEPRITE.

Цель работы: разработка компьютерной игры.

Выпускная квалификационная работа состоит из введения, основной части из трёх глав и заключения. Структура работы отражает решаемые задачи.

Во введении определяется актуальность и полезность разрабатываемого приложения.

В первой главе рассматриваются обзор аналогов и жанров приложения, описывается игровая механика, представлены графические концепты и произведен анализ сред разработки игр, а также определяются инструменты разработки.

Во второй главе рассматривается проектирование сцен и механики, определяется и описываются события, объекты, компоненты, классы и методы с демонстрацией разработанных диаграмм последовательностей, состояний и классов, а также структурной схемы.

В третьей главе демонстрируется итоговый результат графического представления персонажей, локаций и объектов, а также реализуется интерфейс и описывается работа игры.

В заключении формулируются основные итоги по выполненной работе.

СОДЕРЖАНИЕ

Введение	5
1 Анализ предметной области	7
1.1 Цели и задачи.....	7
1.2 Обзор выбранных жанров	7
1.2.1 Action (экшн).....	8
1.2.2 Action-adventure (экшн-приключение)	8
1.2.3 Metroidvania (метроидвания)	8
1.2.4 Приключение	8
1.2.5 Головоломка.....	9
1.2.6 Казуальная игра	9
1.2.7 2D-игра	9
1.2.8 Инди-игра.....	10
1.3 Обзор аналогов	10
1.3.1 Игра «The Elements».....	10
1.3.2 Игра «Molecules»	11
1.3.3 Игра «Chemik»	12
1.4 Концепт разрабатываемой игры	13
1.5 Инструменты разработки	14
1.5.1 Сравнительный анализ сред разработки игр.....	14
1.6 Концепт графического оформления	15
1.7 Вывод по главе	19
2 Проектирование приложения	20
2.1 Структура системы	20
2.2 Базовые элементы Unity	21
2.2.1 MonoBehaviour	21
2.2.2 Scene	22
2.2.3 GameObject.....	22
2.2.4 Компоненты	23
2.3 Главное меню.....	25
2.4 Меню паузы	26
2.5 Новая игра.....	26
2.6 Переход на новый уровень.....	27
2.7 Механика изучения сведений перед боем.....	27
2.8 Механики персонажа.....	28
2.9 Логика поведения противников.....	31
2.10 Окно с вопросами	32
2.11 Камера.....	33
2.12 Звуки.....	33
2.13 Переход между сценами.....	33
2.14 Вывод по главе	34
3 Реализация приложения	35
3.1 Интерфейс игры.....	35

3.2 Графическое оформление	35
3.3 Описание работы игры.....	37
3.3.1 Главное меню.....	37
3.3.2 Комната алхимика	38
3.3.3 Книга	38
3.3.4 Боевая арена.....	38
3.3.5 Меню паузы	39
3.4 Вывод по главе	39
Заключение	40
Список использованных источников	41

ВВЕДЕНИЕ

В настоящее время создание игр стало одной из наиболее крупных частей индустрии развлечений. Игровая индустрия по масштабу сопоставима, например, с киноиндустрией и даже опережала ее по скорости роста за последние несколько лет. По степени влияния на потребителя и его вовлеченности в интерактивное окружение, которое предлагают игры, этот сегмент также давно выделяется среди других видов развлечений. Игры стали неотъемлемой частью в жизни людей. Они помогают расслабиться или же просто приятно провести досуг. Игры подразделяются на множество жанров, так что каждый с легкостью сможет найти что-то свое. Структура современной игровой индустрии выделяет несколько уровней: платформы (компьютеры, консоли, мобильные устройства, игровые автоматы и тому подобное), игровые движки (программные системы, которые обеспечивают различные инструменты для разработки игр), разработка игр, издание и оперирование, популяризация, потребление.

Сейчас различные обучающие игры не менее популярны многих других видов. Они используются как для обучения, так и для развлечения. В процессе таких игр происходит развитие или усовершенствование каких-либо навыков, тренируется память, мышление, внимание и другое. Обучающие игры подходят и взрослым, и детям.

Обучающих игр, как в целом и всех игр, существует большое множество на данный момент. Благодаря таким играм люди могут с интересом и простотой изучить различные науки такие как химия, физика, математика, биология и другие. Над их популяризацией в настоящее время трудятся многие разработчики игр.

Химия – одна из увлекательных наук, изучающая вещества, их состав, строение и свойства. На первый взгляд многим она кажется непонятной и ужасающей. Если разнообразить изучение химии с помощью красивой и интуитивно-понятной игры, то наука перестанет казаться недоступной и

скучной. Поэтому разработка компьютерной игры, которая позволяет закрепить и определить уровень знаний по химии, будет актуальной и полезной.

1 Анализ предметной области

1.1 Цели и задачи

В результате выполнения данного проекта должна быть разработана компьютерная игра с определенными жанрами (экшн, экшн-приключение, метроидвания, приключение, головоломка, казуальная игра, 2D-игра, инди-игра). Игра должна быть реализована в виде приложения для персональных компьютеров на базе операционных систем Windows с помощью среды разработки Unity.

Цель текущего проекта: разработка обучающей компьютерной игры.

Задачи проекта:

- обзор жанров разрабатываемого приложения;
- анализ аналогов разрабатываемого приложения;
- создание и отрисовка персонажей, локаций, игровых предметов;
- реализация передвижения персонажей в пространстве и взаимодействия с интерактивным окружением;
- написание справочного материала для успешного прохождения игры;
- составление списка задач для уровней;
- составление логики и реализация выпадения предметов в случае успешного прохождения уровня;
- создание музыкального сопровождения;
- тестирование игры.

1.2 Обзор выбранных жанров

В современном мире существует огромное количество разных жанров видеоигр, которые придется по вкусу каждому человеку. Игра необязательно должна основываться только на одном жанре, она может включать в себе несколько жанров или поджанров.

1.2.1 Action (экшн)

Слово «action» в переводе с английского — действие, воздействие. В этом и заключается сущность жанра, то есть все действия сосредоточены вокруг игрока, он находится в самом их центре и управляет ими. Игры данного жанра были и остаются в наше время популярными видеоиграми. Жанр классифицируется по многим поджанрам [1].

1.2.2 Action-adventure (экшн-приключение)

Action-adventure или же экшн-приключение сочетает в себе: квесты или испытания, с которыми необходимо справляться, используя собранные во время прохождения игры предметы; элементы экшена, где применяются различные инструменты (оружие, снаряжение и другие предметы) [1].

1.2.3 Metroidvania (метроидвания)

Этот жанр обязан своим названием двум вдохновившим его играм — Metroid и Castlevania. Метроидвании похожи на обычные экшн-приключения, но они не линейны и часто требуют от игрока возвращаться назад или ограничивают его развитие, пока он не найдет конкретный предмет или особый инструмент. Усовершенствование персонажа новым оружием, способностями и другими усилениями открывают доступ в различные места игры. Сражения с боссами — отличительная черта жанра метроидвании [1].

1.2.4 Приключение

Приключение предполагает решение игроком различных головоломок, взаимодействуя с окружающей средой и предметами. Таким образом игрок продвигается по сюжету [1].

1.2.5 Головоломка

Головоломки, или логические игры, требуют от игрока решения каких-либо интеллектуальных задач или прохождения испытаний для ума, например, лабиринты. Логические игры могут быть как простыми адаптациями реальных головоломок, например как судоку, так и полномасштабными играми для ума. Они проверяют навыки игрока в решении проблем, включая логику, распознавание образов, решений последовательностей и другое. Игра-головоломка фокусируется на решении головоломок в качестве основного игрового действия. Данный жанр, являясь одним из самым старым видом игр, продолжает находить миллионы поклонников [1].

1.2.6 Казуальная игра

Казуальный жанр стал одним из популярных и прибыльных жанров. Казуальные игры — это игры, предназначенные для широкого круга пользователей. В данном жанре обычно суть игры понятна без пояснений, а правила не так сложны. От игрока не требуется больших затрат на обучение и каких-либо особых навыков [1].

1.2.7 2D-игра

2D-игры представляются с видом сверху, сбоку или от третьего лица, то есть, когда игровой процесс идет в виде панорамы. Игра в стиле 2D влечет (не всегда) за собой пиксельную графику, а она в свою очередь — простоту в использовании и дешевизну. С такой графикой разработчику легче создать игру. Даже без использования новейших графических технологий можно создать настоящий визуальный шедевр [1].

1.2.8 Инди-игра

Инди-игры — это игры, которые создаются независимыми авторами, то есть людьми, работающими на себя. Независимые авторы редко берутся за масштабные проекты, это позволяет сфокусироваться на особенностях игры — интересный сюжет, необычный геймплей, красивая графика или душевное звуковое сопровождение — и довести их до идеала. Независимые разработчики создают те проекты, которые хотят именно они и не боятся преследовать безумные идеи [1].

1.3 Обзор аналогов

Химия – одна из увлекательных наук, изучающая вещества, их состав, строение и свойства. На первый взгляд многим она кажется непонятной и ужасающей. Если разнообразить изучение химии с помощью красивой и интуитивно-понятной игры, то наука перестанет казаться недоступной и скучной.

Стоит рассмотреть несколько уже существующих аналогов.

1.3.1 Игра «The Elements»

Игра предназначена для изучения химических элементов. «The Elements» представляет собой таблицу Менделеева со всеми химическими элементами, которые реализованы в виде красивых и четких 3D-моделей [2] (Рисунок 1).



Рисунок 1 – Интерфейс игры «The Elements»

В таблице у всех элементов есть свой собственный образец в виде вращающейся 3D-модели. Например, для серебра – серебряная монета. Игрок может выбрать любую модель из таблицы и вращать или изменять ее масштаб, а также узнать различные данные об элементе такие, как атомная масса, температура плавления и другие интересные факты.

Данная игра была признана одной из лучших программ для изучения химии.

1.3.2 Игра «Molecules»

Исходя из названия можно понять, что игра направлена на изучение молекул. В ней представлены трехмерные модели различных веществ, которые можно вращать, также изменять размер молекулы и другое благодаря нескольким режимам визуализации [2] (Рисунок 2).

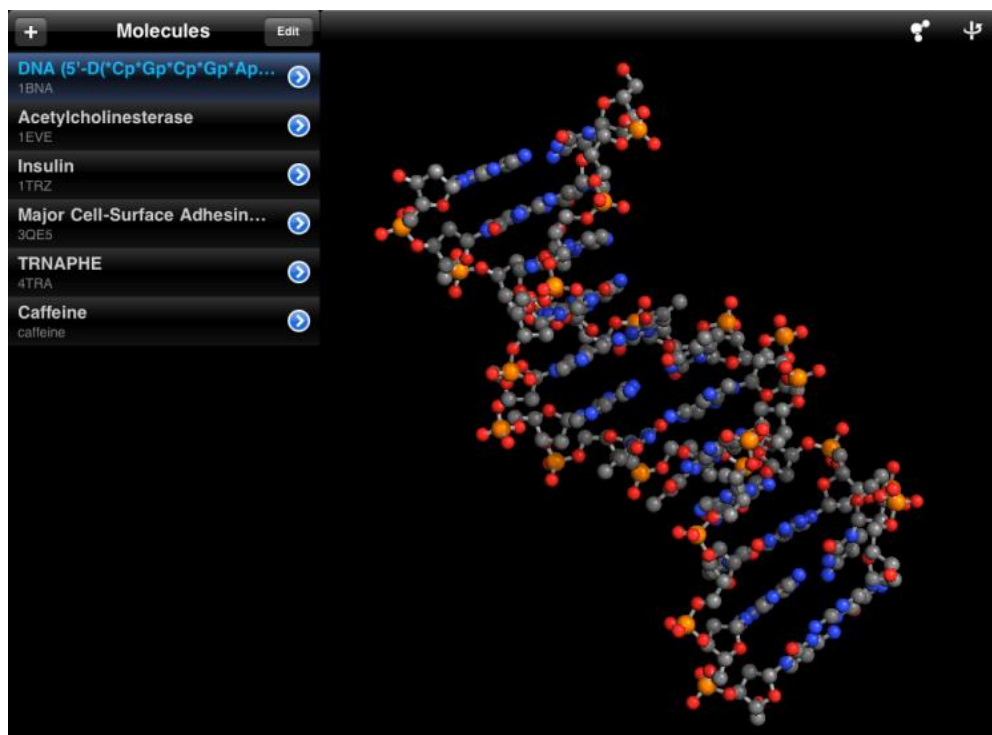


Рисунок 2 – Молекулярная модель ДНК в игре «Molecules»

Приложение наполнено огромной базой данных молекулярных моделей, при этом игроку предоставляется возможность загружать примеры со специализированных сайтов международных хранилищ биологических молекул и их трехмерных моделей. В игре имеется полная информация о каждой молекуле и каждом молекулярном соединении. Например, полное название молекулярной структуры, количество и виды аминокислот в случае белков, нуклеотидные последовательности для ДНК и РНК, имена исследователей соединения и многое другое [2].

1.3.3 Игра «Chemik»

Игра позволяет ознакомиться с активностью элементов, изучать окислительно-восстановительные реакции, получать конечные продукты реакции и уравнивать коэффициенты. Имеется описание реакций большого множества химических соединений. Интерфейс и взаимодействия в игре

достаточно просты и понятны: например, для реакции необходимо выбрать из таблицы нужные элементы и просто соединить их (Рисунок 3) [2].

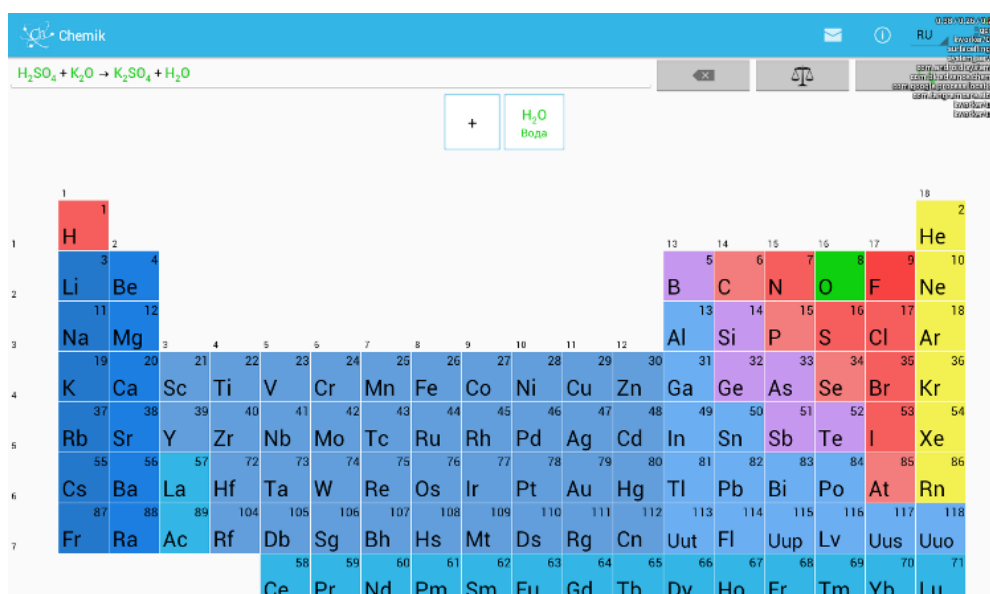


Рисунок 3 – Интерфейс игры «Chemik»

1.4 Концепт разрабатываемой игры

Игровой процесс заключен в стенах комнаты алхимика (неигрового персонажа). Алхимик обучает главного героя (игрока) химии. Задачей игрока является борьба с монстрами (противниками), то есть решение различных химических задач. Химическая задача представляет из себя вопрос и предложенные на него варианты ответа. В игре существует 5 уровней, в каждом уровне по 5 химических задач. Сложность уровней, а соответственно и вопросов, возрастает по мере прохождения игры. Под конец игры главному герою предстоит финальный уровень — бой с боссом (заключительным противником).

В комнате алхимика находятся магический портал и книга-справочник. В книге-справочнике представлена вся необходимая информация для решения химических задач. Для того, чтобы вступить в бой, необходимо зайти в магический портал. После чего игрока переносит на локацию боевой

арены, и задается вопрос по пройденному материалу. В случае верного ответа монстру наносится определенный урон. В случае неверного ответа игрок получает фиксированный урон, и его очки здоровья снижаются. Если игрок отвечает верно на все вопросы, то он побеждает монстра, следовательно успешно проходит уровень и получает уникальный игровой предмет. Если игрок допустил множество ошибок, то есть неверно отвечал на представленные вопросы, и его очки здоровья снизились до нуля, тогда уровень необходимо проходить заново.

1.5 Инструменты разработки

Для создания игрового процесса используется IDE Visual Studio 2022 для работы с языком C#. Для визуальной составляющей используется графический редактор Aseprite.

Чтобы упростить разработку новых игр, а также удешевить их производство, можно использовать уже готовый игровой движок. Современные игровые движки способны работать с несколькими аппаратными платформами или операционными системами. Существует множество продвинутых движков, например, Unity, Unreal Engine и другие.

1.5.1 Сравнительный анализ сред разработки игр

Unity — межплатформенная среда разработки игр. Она появилась в 2005 году и в настоящее время остается одной из самых популярных платформ. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства и другие. Unity преимущественен тем, что предполагает собой визуальную среду разработки, а также межплатформенную поддержку, которая позволяет адаптировать проект для любой платформы путем внесения минимальных изменений. Графический

редактор с большим набором функций предусматривает создание моделей, локаций, расстановку объектов сцены с использованием тестирования в реальном времени. Игровой движок Unity популярен как среди крупных разработчиков, так и независимых студий. На нем написана не одна тысяча игр и приложений, охватываемых большей частью платформ и жанров. Движок использует для написания скриптов язык программирования C# [3].

Unreal Engine — также межплатформенная среда, больше направленная на 3D проекты со встроенной системой визуального программирования Blueprint. Набор функций лучше подходит для трехмерных проектов. Система Blueprint является быстрым способом создания прототипов игр. Благодаря ей можно делать все визуально, вместо построчного написания кода. Unreal Engine зачастую используется среди крупных и опытных разработчиков для создания больших и дорогостоящих проектов. Для разработки используется язык программирования C++ [4].

1.6 Концепт графического оформления

С помощью редактора Aseprite нарисованы различные персонажи, локации и прочее необходимое для разрабатываемой игры.

Графическое оформление главного героя (Рисунок 4).

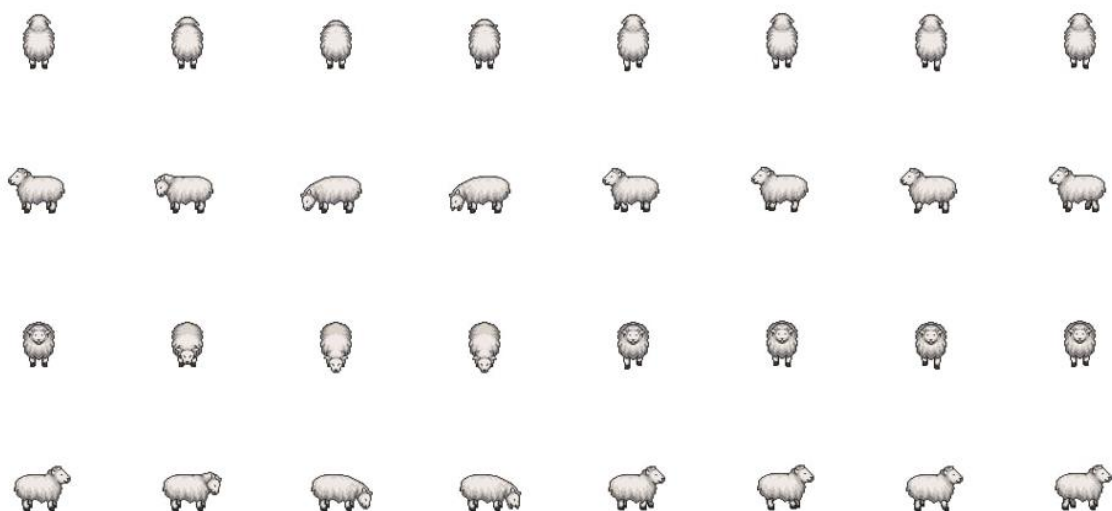


Рисунок 4 – Концепт изображения главного героя

Графическое оформление алхимика (Рисунок 5).

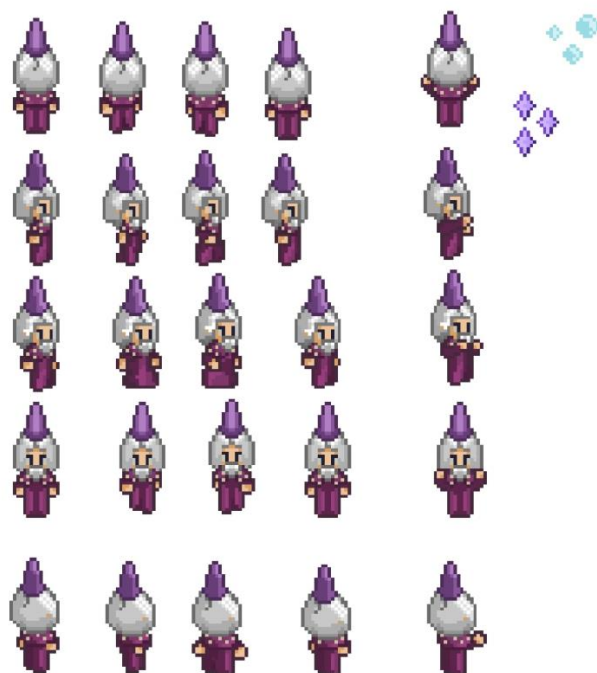


Рисунок 5 – Концепт изображения алхимика

Графическое оформление комнаты алхимика (Рисунок 6).

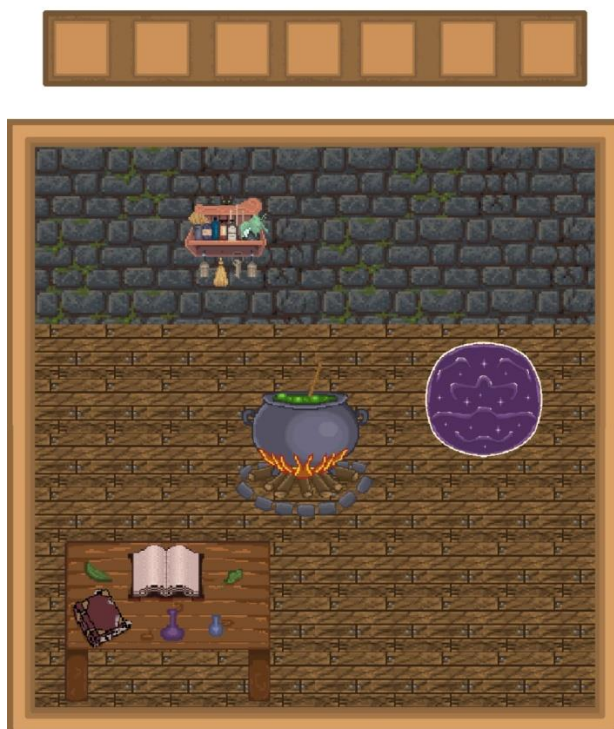


Рисунок 6 – Концепт изображения комнаты алхимика

Графическое оформление боевой арены (Рисунок 7).

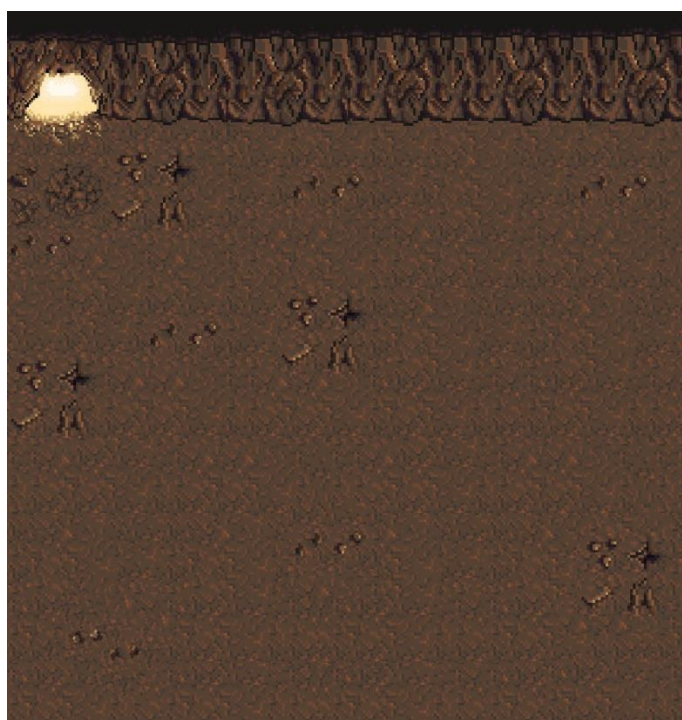


Рисунок 7 – Концепт изображения боевой арены

Графическое оформление иконки игры (Рисунок 8).



Рисунок 8 – Концепт изображения иконки игры

Текстура эффектов взрыва (Рисунок 9).

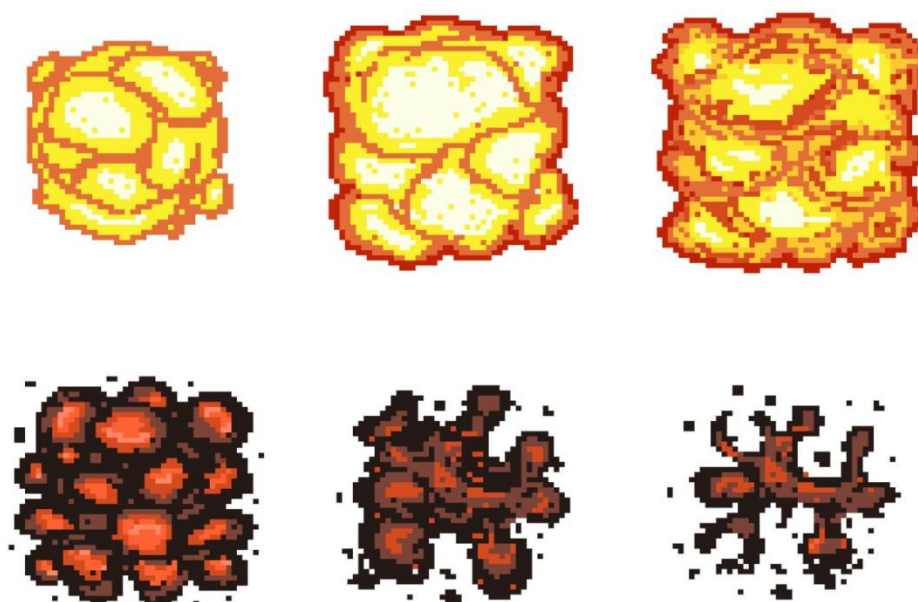


Рисунок 9 – Концепт эффектов взрыва

Графическое оформление монстра (Рисунок 10).



Рисунок 10 – Концепт изображения монстра

1.7 Вывод по главе

В данной главе поставлены цели и задачи, произведен обзор различных жанров и аналогов, описан концепт игровой механики, представлены графические концепты, а также произведен анализ существующих наиболее удобных сред разработки игр.

Обзор аналогов позволил сделать вывод, что разрабатываемый в работе продукт актуален и уникален.

Исходя из требований приложения, в результате сравнения сред разработки игр Unity и Unreal Engine, были выявлены преимущества и недостатки для разных задач. Unity подходит для новичков и менее крупных разработчиков, в свою очередь Unreal Engine — для профессиональных студий. Таким образом, в качестве среды разработки игры был выбран Unity.

2 Проектирование приложения

2.1 Структура системы

При работе над приложением была разработана структура программной системы (Рисунок 11).

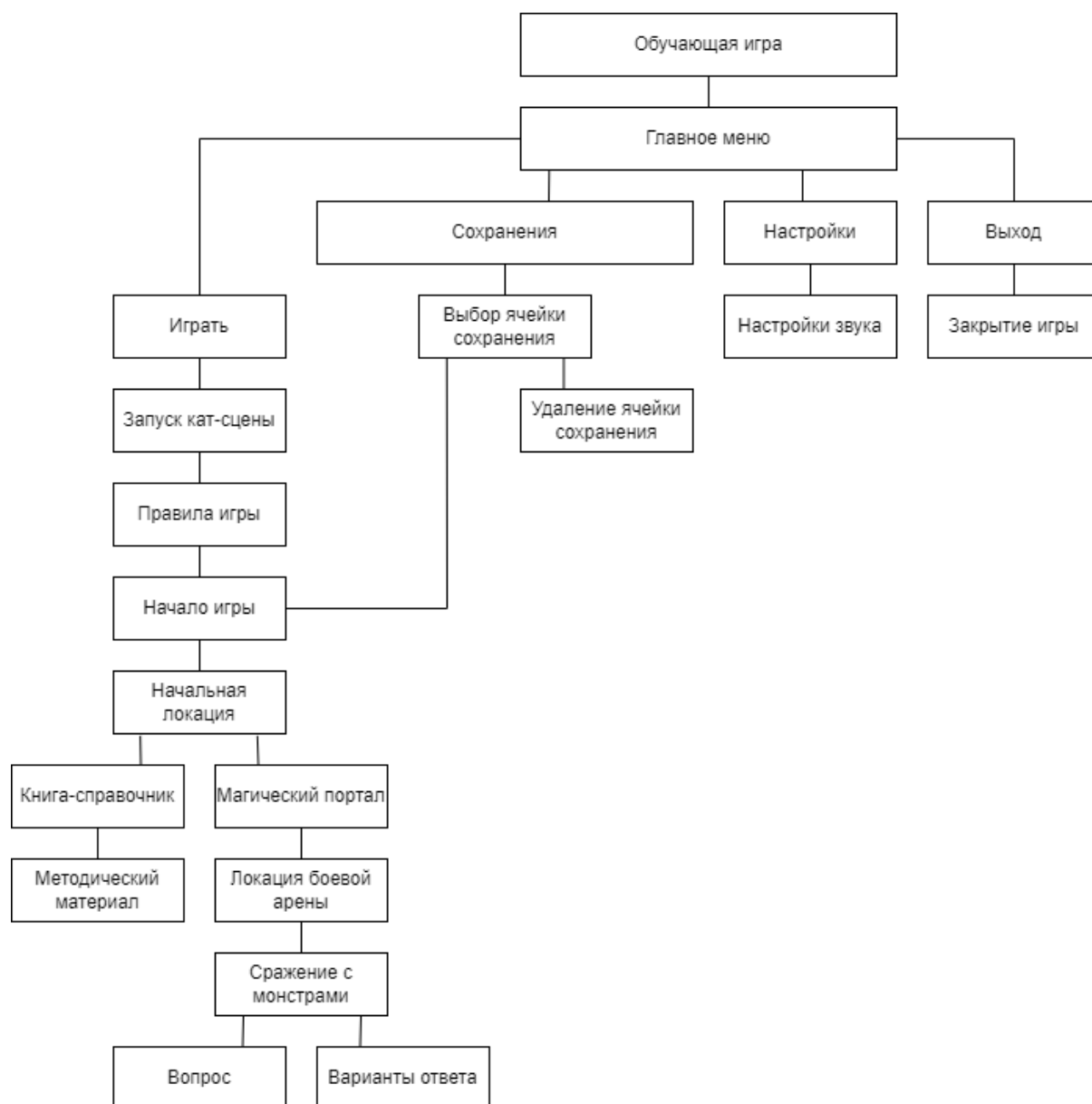


Рисунок 11 – Структура разрабатываемого приложения

2.2 Базовые элементы Unity

2.2.1 MonoBehaviour

Класс **MonoBehaviour** – это базовый класс Unity, от которого наследуются все скрипты. При создании сценария Visual Studio из окна проекта Unity, он автоматически наследуется от MonoBehaviour и предоставляет шаблон сценарий (Рисунок 12). Также класс предоставляет структуру, которая позволяет прикрепить сценарий к GameObject в редакторе [5].

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class qwe : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10         }
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16         }
17     }
18 }
```

Рисунок 12 – Шаблон сценарий

Класс **MonoBehaviour** предоставляет доступ к большому набору методов, например, таких как:

- Start() вызывается, когда GameObject начинает существовать (либо при загрузке сцены, или создается экземпляр GameObject);
- Update() вызывается каждый кадр;
- FixedUpdate() вызывается каждый временной шаг;

- OnCollision() вызывается при столкновениях физики;
- OnTrigger() вызывается при возникновении триггеров.

2.2.2 Scene

Сцена – это самый крупный блок, который описывает организацию объектов в памяти. В сценах содержатся объекты, из которых состоит игра (Рисунок 13).

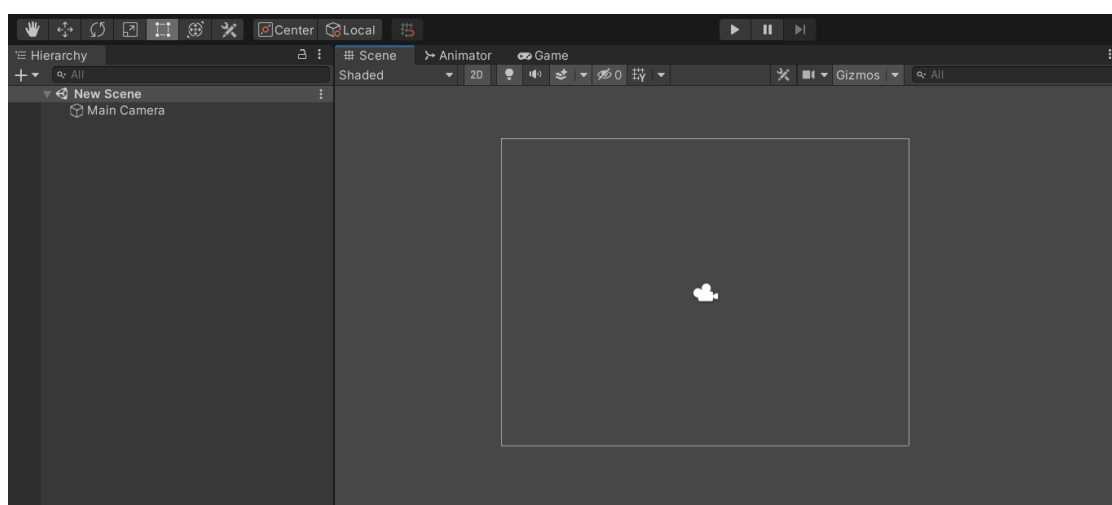


Рисунок 13 – Новая сцена

Сцена представляет собой отдельно взятый уровень игры, где в любой конкретный момент времени загружена одна сцена. Одновременно могут быть активны две или более сцен. В пустых сценах Unity2D по умолчанию содержится объект Main Camera (камера). Каждый игровой объект в Unity должен находиться в сцене [6].

2.2.3 GameObject

GameObject – это базовый игровой объект, из которого строится игра.

В виде игровых объектов можно представлять как физические сущности (например: персонаж, стол, ландшафт, выстрел, взрыв и другие), так и

метафизические (например: менеджер снаряжения, контроллер мультиплеерного режима и другие).

У каждого игрового объекта есть значения положения. Для метафизических объектов они не имеют значения. Объект позиционируется относительно координат [6].

2.2.4 Компоненты

Каждый игровой объект состоит из компонент (Рисунок 14). Компонент реализует определенный набор поведений, необходимых для выполнения GameObject.

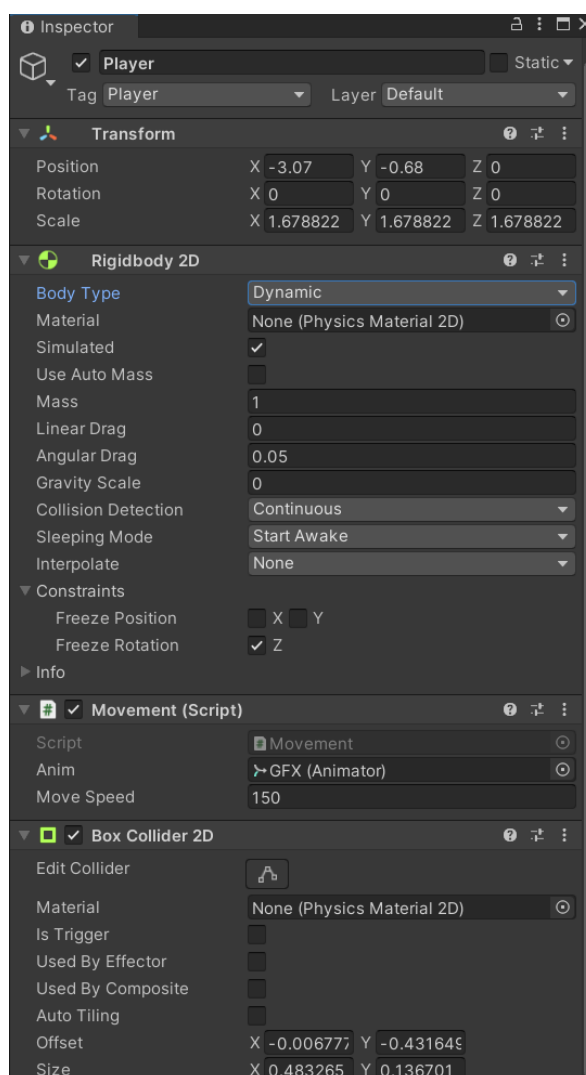


Рисунок 14 – Компоненты объекта Player

Компонент Rigidbody 2D позволяет взаимодействовать с игровыми объектами с помощью физики. Для реалистичного перемещения объектов на них воздействуют масса, гравитация и другое.

Любой игровой объект должен содержать в себе компонент **Rigidbody**, чтобы быть подверженным гравитации, действовать согласно назначенным путём скриптинга силам, или взаимодействовать с другими объектами [7].

Компонент Box Collider 2D – это коллайдер, который взаимодействует с системой 2D-физики – прямоугольник в форме с определенным положением, шириной и высотой в локальном пространстве координат [5].

Компонент содержит в себе большой набор редактируемых свойств и функций, например:

- **Material** – физический материал, который определяет свойства коллизий, например трение и отскок;

- **Is Trigger** – чтобы **Box Collider 2D** вел себя как триггер, необходимо установить этот флаг;

- Во вкладке **Size** можно задать размер.

Компонент Animator нужен для назначения анимации **GameObject**. Компонент также содержит набор различных свойств и функций, например, – **Controller** – контроллер аниматора, прикрепленный к персонажу.

Компонент Audio Source – это источник звука. Именно этот компонент воспроизводит звуки в сценах.

Компонент Canvas – это область для элементов пользовательского интерфейса (кнопки, текст, изображения и другое). Она предоставляет возможность создавать элементы пользовательского интерфейса и размещать их на экране.

UI (User Interface) – интерфейс пользователя. Модуль пользовательского интерфейса реализует основные компоненты, необходимые для системы пользовательского интерфейса Unity.

2.3 Главное меню

Начальная сцена после загрузки игры представляет собой главное меню (Рисунок 15).

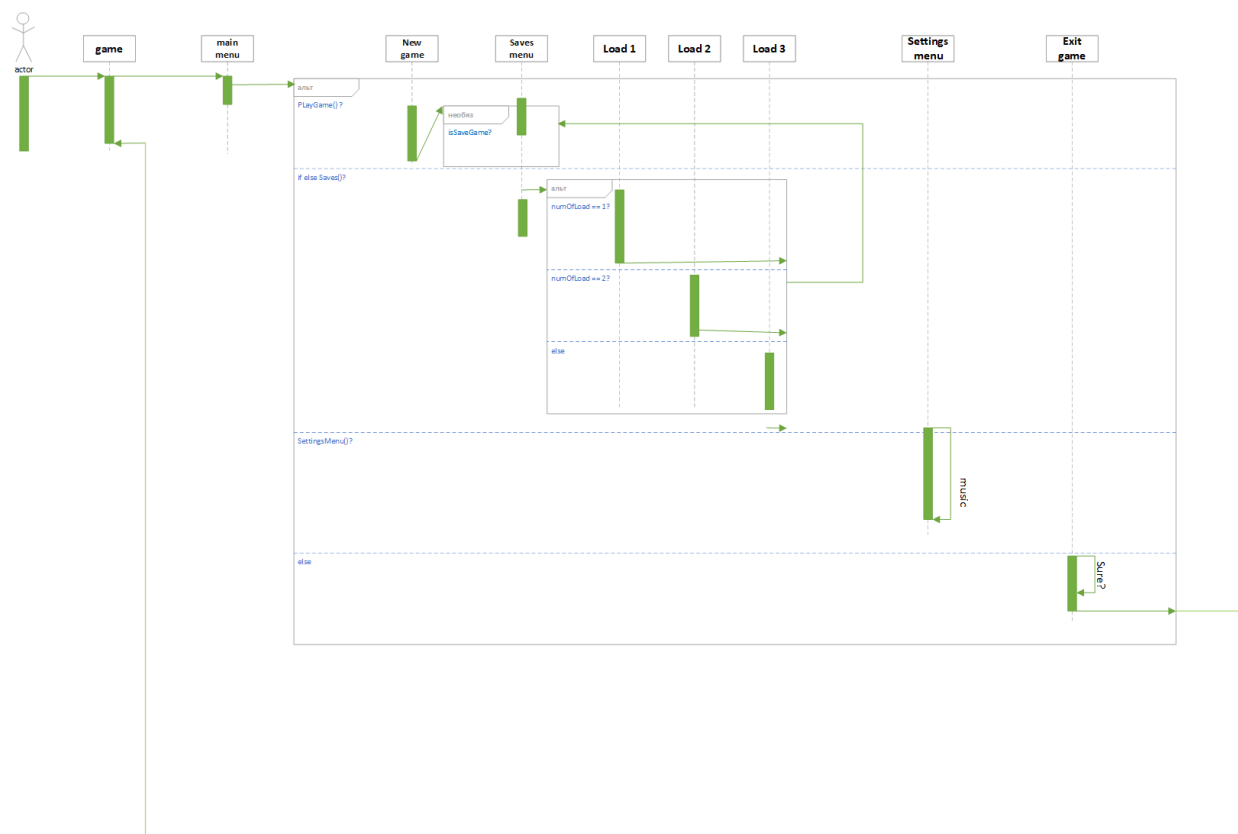


Рисунок 15 – Диаграмма последовательностей главного меню

Главное меню состоит из следующих кнопок: «Играть», «Сохранения», «Настройки» и «Выход». Начальной сценой и объектами управляет класс **Navigation**, который содержит в себе следующие методы:

- PlayGame() запускает начальный уровень игры при нажатии кнопки «Играть»;
- Saves() запускает выбранное сохранение;
- ExitGame() закрывает игру.

При срабатывании метода Saves() вызывается метод стандартного класса Unity SceneManager – LoadScene(), который запускает выбранную

сцену (сохранения игрового прогресса). Метод `Reset()` удаляет сохранение. Для того, чтобы удалить сохранение, нужно нажать на него правой кнопкой мыши, после чего срабатывает разработанный класс **DropDownMenu**, и вызывается метод `DeleteSave()` удаления сохраненного состояния игры (Появляется кнопка «Удалить», при ее нажатии всплывает предупреждение об удалении. Если пользователь нажал на кнопку согласия удаления, срабатывает метод `Reset()`, и сохранение удаляется навсегда).

2.4 Меню паузы

Кнопка паузы постоянно находится как UI-элемент на экране игрока в процессе игры в правом верхнем углу. При нажатии на нее срабатывает скрипт с классом **PauseMenu**, в котором содержатся методы:

- `SaveThisGame()` сохраняет данный игровой процесс;
- `Settings()` отвечает за настройки звука;
- `BackToMainMenu()` возвращает игрока в главное меню.

2.5 Новая игра

При нажатии пользователем на кнопку «Играть» срабатывает метод класса **Navigation** – `PlayGame()`, который вызывает сцену начала игры.

При запуске новой игры срабатывает класс **TimeLineManager**, в нем класс **OnEnable**, который передает в объект `Unity – Animator` полный контроль над персонажами и игровыми объектами. Метод `Update()` проверяет закончилась ли анимация и сообщает результат проверки `Unity`. Это позволяет проиграть кат-сцену (внутриигровое видео), и ввести игрока в суть игры, а также объяснить правила.

2.6 Переход на новый уровень

Диаграмма последовательностей перехода игрока на следующий уровень (Рисунок 16).

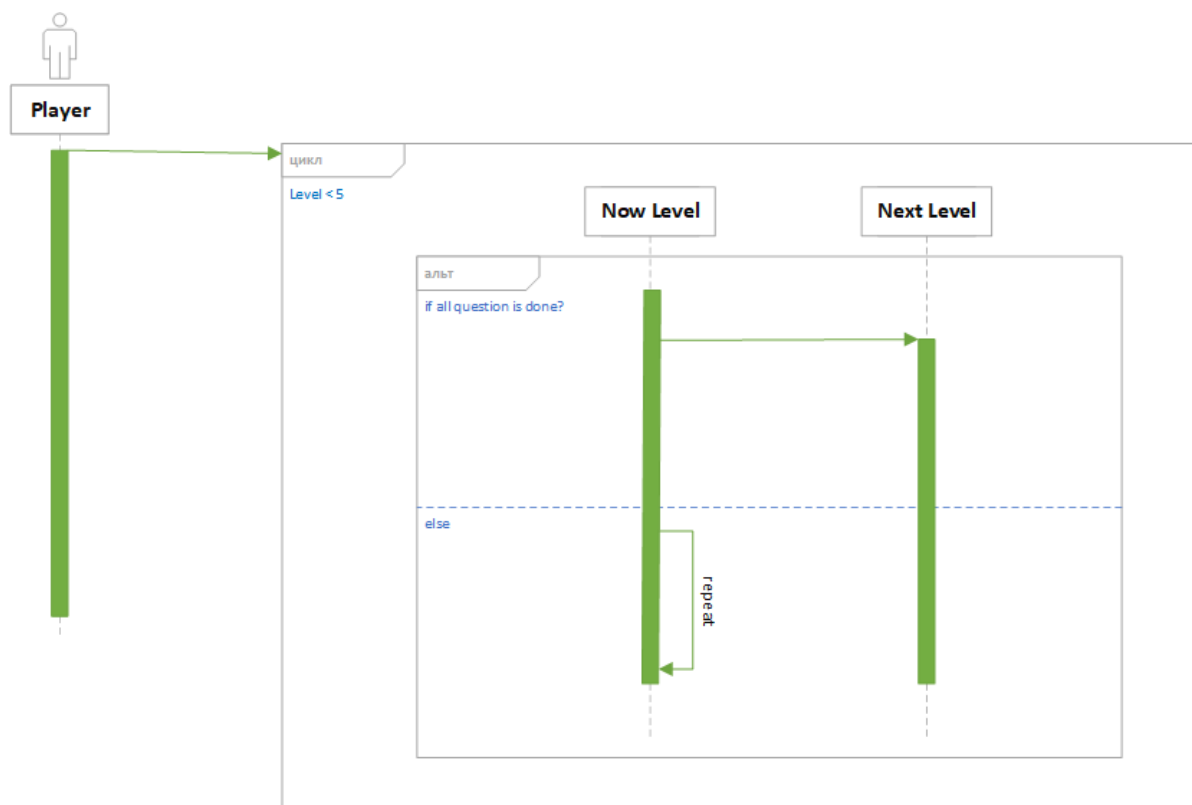


Рисунок 16 – Диаграмма последовательностей перехода на новый уровень

2.7 Механика изучения сведений перед боем

На игровом поле находится игровой объект **Book**. К нему прикреплен скрипт с классом **OpenBook**. Если игрок подойдет достаточно близко и нажмет клавишу «E» (английская раскладка), сработает метод **OnTriggerStay2D()**, который вызывает метод **Open()**, отвечающий за загрузку сцены с методическим материалом.

Класс **BookButtonsController** отвечает за навигацию внутри сцены с методическим материалом и содержит в себе приватные переменные с

игровым контентом (текст методического материала и таблица Менделеева) и следующие методы:

- Start() инициализирует объекты Unity – Text и Image, которые находятся на страницах книги;
- Return() позволяет игроку вернуться на предыдущую сцену (комната алхимика);
- NextPage() отвечает за переворачивание страниц вперед путем добавления контента в объекты Text и Image;
- BackPage() отвечает за переворачивание страниц назад путем добавления контента в объекты Text и Image.

2.8 Механики персонажа

UML-диаграмма классов боевой арены (Рисунок 17).

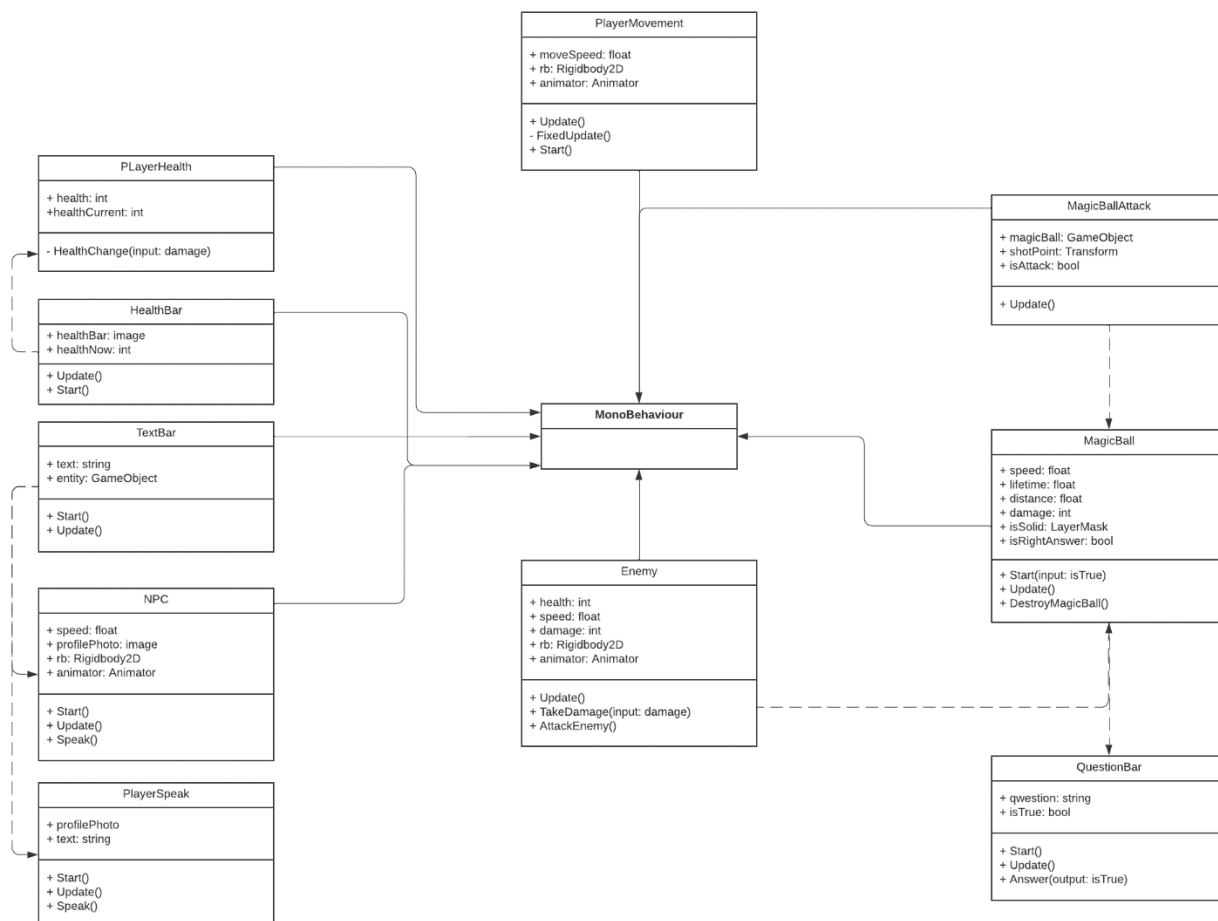


Рисунок 17 – Диаграмма классов боевой арены

Управление персонажем реализовано в классе **Movement**, который содержит следующие методы:

- Start() вызывается при старте игры и инициализирует Rigidbody2D;
- В Update() вызывается каждый кадр игры. Метод отслеживает перемещение игрока по игровому полю с помощью переменных (x, y), куда сохраняется положение игрока по горизонтали (Input.GetAxisRaw("Horizontal")) и вертикали (Input.GetAxisRaw("Vertical")). Затем проходит проверка движется ли игрок и запускается соответствующая анимация. В переменную moveDir сохраняется нормализованное положение игрока с помощью объекта класса Vector3. Update() также проверяет уровень здоровья игрока, если он меньше либо равен нулю, тогда вызывает анимацию UI-компонента текст, обозначающую проигрыш;

– `FixedUpdate()` вызывается в каждом фиксированном кадре игры. В данном методе у объекта `Rigidbody2D` в атрибут `velocity` передается нынешняя позиция объекта `Rigidbody2D`, умноженная на его скорость и на нормализующее значение (`moveDir * moveSpeed * Time.deltaTime`).

`FixedUpdate()` также отвечает за отображение жизней игрока;

– `StopMoving()` задает объекту `Rigidbody2D` в атрибут `velocity` нулевое значение `Vector3`, что останавливает персонажа;

– `ChangeHealth()` меняет значение жизней игрока.

Класс **`shotPointMagicBall`** отвечает за реализацию атаки игрока, содержит в себе следующие методы:

– `Start()` инициализирует игрока и определяет начальное состояние возможности атаки в `true`;

– `Update()` отслеживает положение курсора и стреляет в его направлении, также регулирует перезарядку;

– `Shoot()` проверяет может ли игрок стрелять в данный момент и создает объект снаряда.

Класс **`MagicBall`** отвечает за объект снаряда, содержит в себе следующие методы:

– `Start()` удаляет снаряды по истечении определенного времени;

– `Update()` проверяет столкнулся ли снаряд с каким-либо объектом, также перемещает его в пространстве;

– `DestroyMagicBall()` удаляет снаряд по истечении определенного времени с заданным эффектом.

Диаграмма состояний игрового персонажа при входе в боевой режим (Рисунок 18).

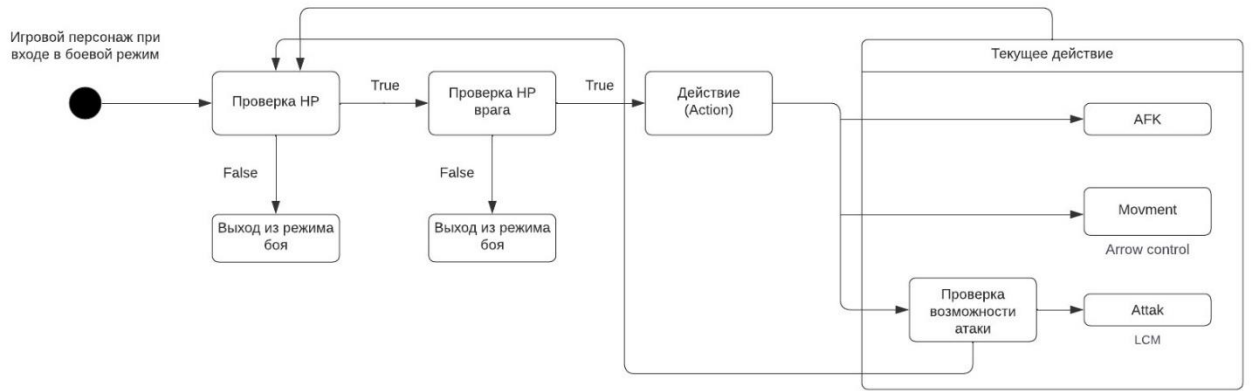


Рисунок 18 – Диаграмма состояний игрока

Диаграмма последовательностей нанесения урона противнику (Рисунок 19).



Рисунок 19 – Диаграмма последовательностей нанесения урона противнику

2.9 Логика поведения противников

Класс **Enemy** отвечает за поведение противника, содержит в себе следующие методы:

- Start() срабатывает при первом появлении противника на уровне боевой арены. Определяет игрока и базовые анимации врага;
- В Update() вызывается каждый кадр игры. Проверяет уровень жизней противника, если их не осталось, тогда удаляет объект противника и вызывает UI-компонент текст, сигнализирующий о победе. Также отвечает за передвижение врага и его анимации.
- TakeDamage() отнимает здоровье у противника;
- OnCollisionEnter2D() проверяет вступил ли объект MagicBall с врагом, если вступил, то вызывает анимацию с текущим вопросом, замедляет скорость противника, его анимацию, а также скорость игрока и блокирует возможность атаки игроком.
- QuesCompl() определяет номер текущего вопроса, меняет его на следующий номер вопроса и возвращает значение скорости и другие параметры, измененные в предыдущем методе.

Диаграмма состояний противника при входе в боевой режим (Рисунок 20).

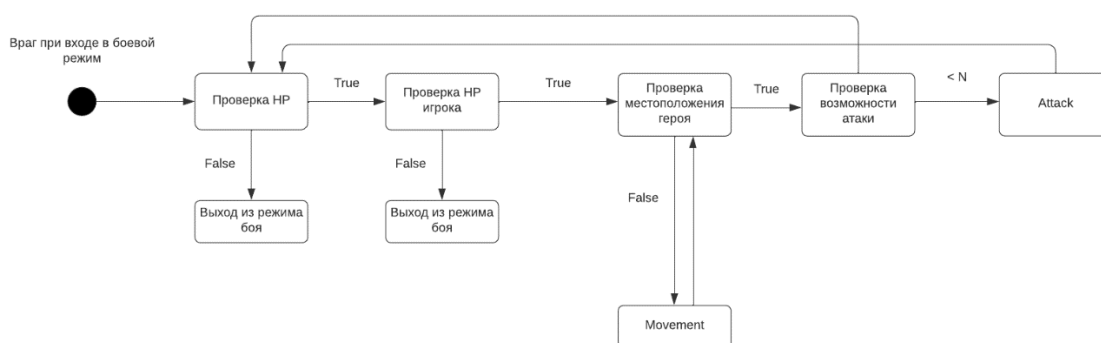


Рисунок 20 – Диаграмма состояний противника

2.10 Окно с вопросами

Класс **DialogueManager** отвечает за окна с вопросами, содержит в себе методы:

- Start() инициализирует первый номер вопроса;
- QuestionUpdate() увеличивает номер вопроса;
- TrueAnswer() отвечает за правильно выбранный ответ, вызывает у класса Enemy метод TakeDamage(), передавая в него силу удара;
- FalseAnswer() отвечает за неправильно выбранный ответ, вызывает у класса Movement метод ChangeHealth() и передает в него силу удара врага.

2.11 Камера

Класс **CameraScript** предназначен для следования камерой за игроком в процессе его перемещения по игровому полю, содержит методы:

- Start() инициализирует игрока;
- Update() проверяет перемещение игрока и задает координаты камеры.

2.12 Звуки

Класс **SoundsManager** предназначен для музыкального сопровождения игры и различных звуковых эффектов. Он содержит публичные переменные с музыкой и звуками. В классе находится метод ButtonMusic(), который назначает музыку в атрибут AudioClip и у AudioSource вызывается source.Play().

2.13 Переход между сценами

Класс **ExitToMainMenu** имеет единственный метод Open(), который вызывается внешними методами и загружает сцену главного меню.

Класс **returnToMainRoom** имеет единственный метод Open(), который вызывается внешними методами и загружает сцену комнаты алхимика.

Класс FitingRoom имеет два метода:

– OnTriggerStay2D() проверяет зашел ли игрок в зону действия коллайдера и была ли нажата кнопка E, в этом случае он вызывает метод Open();

– Open() загружает сцену боевой арены FightingRoom.

2.14 Вывод по главе

В данной главе спроектированы различные сцены и механики, а также определены классы и методы.

3 Реализация приложения

3.1 Интерфейс игры

Интерфейс игры реализован с помощью различных UI-элементов. Область Canvas представляет собой пространство, в котором находятся все элементы UI (пользовательского интерфейса). Область видимости представляет собой три слоя различных Tilemap. Самый нижний CollisionTilemap не отрисовывает интерфейс, а представляет собой физику игровых объектов. Два других слоя отвечают за визуальный интерфейс. Для текста используется TextMeshPro.

3.2 Графическое оформление

Графическое оформление главного меню при запуске игры (Рисунок 21).



Рисунок 21 – Главное меню

Графическое представление комнаты алхимика (Рисунок 22).



Рисунок 22 – Комната алхимика

Графическое представление книги-справочника (Рисунок 23).

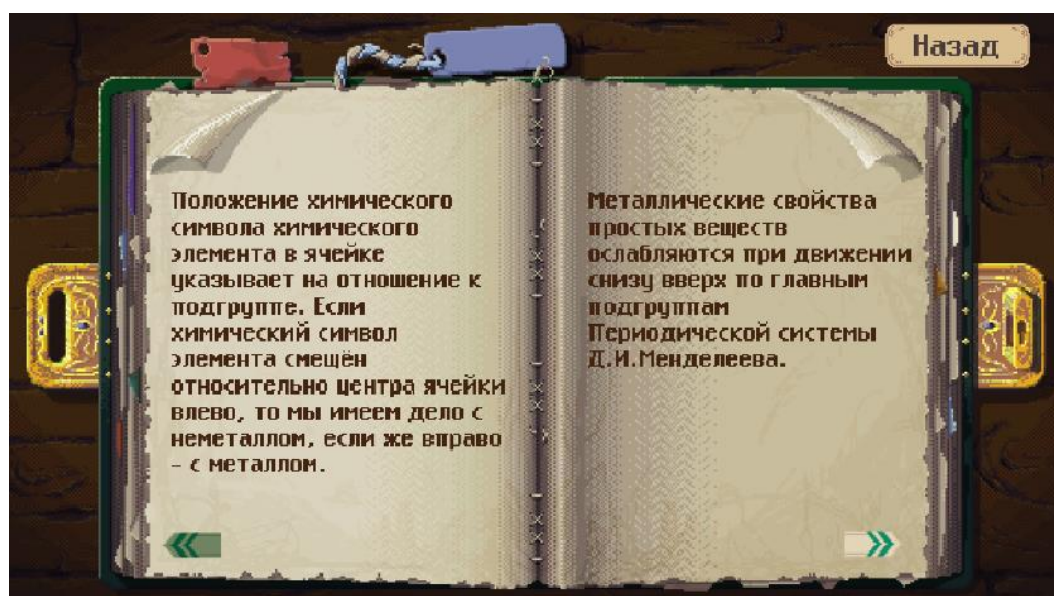


Рисунок 23 – Книга-справочник

Графическое представление локации боевой арены (Рисунок 24).

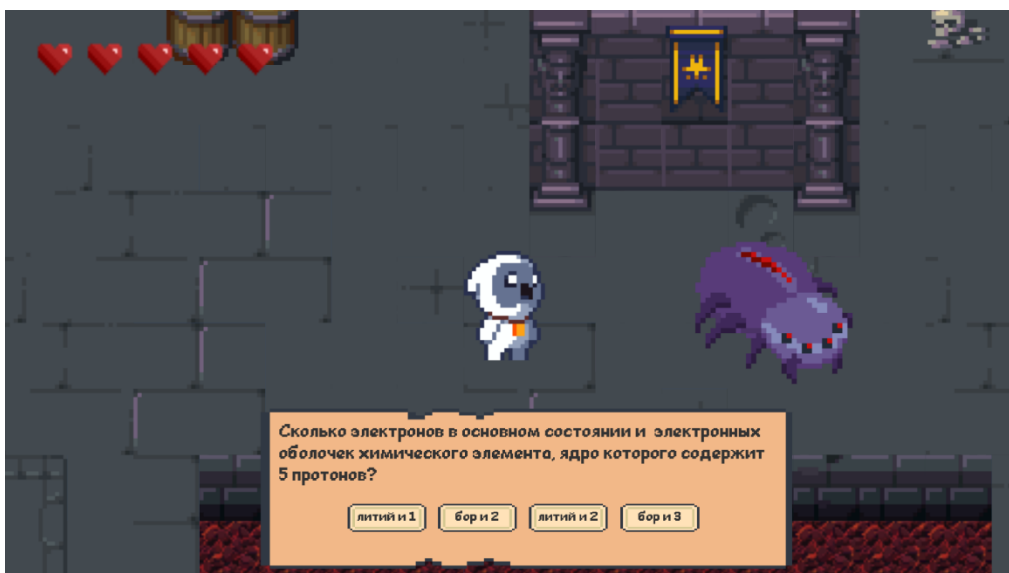


Рисунок 24 – Боевая арена

3.3 Описание работы игры

3.3.1 Главное меню

При запуске игры пользователь видит главное меню, на котором присутствуют кнопки:

- «Играть» – при нажатии на нее начинается новая игра;
- «Сохранения» – при нажатии на нее открывается подменю с тремя ячейками сохранения и одной кнопкой для удаления ячейки сохранения. Если нажать на сохранение два раза, то оно откроется и сохраненный игровой процесс запустится. Если нажать на сохранение один раз, а затем на кнопку удаления, появится предупреждение («Вы действительно хотите удалить?») и две кнопки – «Удалить» и «Отмена», которые выполняют соответствующие действия с сохранением.
- «Настройки» – при нажатии на эту кнопку открывается подменю с настройками звука в игре: настройка громкости звуков интерфейса, настройка громкости фоновой музыки.

– «Выход» – при нажатии на нее появляется предупреждение («Вы действительно хотите выйти из игры?»). При подтверждении приложение закрывается.

3.3.2 Комната алхимика

В данной локации есть два игровых предмета – магический портал и книга-справочник.

Книга-справочник – если игрок находится достаточно близко и нажимает клавишу E (английская раскладка), то открывается книга с методическими материалами.

Магический портал – если игрок находится достаточно близко и нажимает клавишу E (английская раскладка), персонаж переносится на боевую арену.

3.3.3 Книга

После открытия книги с методическими материалами игрок видит текст, таблицу Менделеева, две кнопки для перелистывания страниц, а сверху находится кнопка возврата «Назад» в комнату алхимика.

3.3.4 Боевая арена

На этой локации можно взаимодействовать только с противником. Для того чтобы нанести урон необходимо произвести атаку магическим шаром (нажатие левой кнопкой мыши по врагу), после чего игроку нужно ответить на вопрос. В случае верного ответа враг получит урон, в ином случае урон получит уже игрок. После того как игрок ответит на все вопросы, он перемещается в главную комнату, с результатом своих ответов – проигрыш или выигрыш.

3.3.5 Меню паузы

Открывается при нажатии на кнопку в правом верхнем углу. На этой сцене есть кнопки «Главное меню», «Настройки музыки», а также кнопка «Сохранение».

«Главное меню» – при нажатии на эту кнопку игрок попадает на сцену с главным меню.

«Настройка музыки» – отвечает за громкость музыки в игре (громкость звуков интерфейса, громкость фоновой музыки).

«Сохранение» – данная кнопка сохраняет текущий игровой прогресс.

3.4 Вывод по главе

В данной главе созданы персонажи, локации, эффекты; реализован интерфейс; описана работа игры.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы была разработана компьютерная игра под названием «Dolly» и выполнены следующие задачи:

- проанализированы аналоги разработанного приложения;
- произведен обзор жанров разработанного приложения;
- созданы и отрисованы персонажи, локации, игровые предметы;
- реализованы передвижения персонажей в пространстве и взаимодействия с интерактивным окружением;
- написан справочный материал для успешного прохождения игры;
- составлен список задач для уровней;
- составлена логика и реализовано выпадение предметов в случае успешного прохождения уровня;
- создано музыкальное сопровождение.

В дальнейшем можно расширить перечень справочного материала и вопросов, а также использовать приложение для закрепления школьной программы по химии.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Жанры компьютерных игр // Cubiq : русскоязычный интернет-ресурс о компьютерных играх. — URL: <https://cubiq.ru/zhanry-kompyuternyh-igr> (дата обращения: 25.11.2022).

2. 13 игр и приложений для изучения химии // Newtonew : русскоязычный интернет-ресурс в сфере образования. — URL: <https://newtonew.com/app/13-igr-i-prilozhenij-dlja-izuchenija-himii> (дата обращения: 25.11.2022).

3. Игровой движок Unity : официальный сайт. — URL: <https://unity.com/ru> (дата обращения: 26.11.2022).

4. 10 лучших движков для создания игр // Комьюнити : русскоязычный интернет-ресурс в сфере IT. — URL: <https://timeweb.com/ru/community/articles/10-luchshih-dvizhkov-dlya-sozdaniya-igr> (дата обращения: 27.11.2022).


5. Сообщество разработчиков // UnityHub : русскоязычный интернет-ресурс в сфере разработки игр. — URL: <https://unityhub.ru> (дата обращения 10.05.2023).

6. Базовые концепции Unity для программистов // Хабр : русскоязычный интернет-ресурс в сфере IT. — URL: <https://habr.com/ru/companies/piter/articles/529648> (дата обращения 10.05.2023).

7. Руководство пользователя Unity // UnityDocumentacion : русскоязычный интернет-ресурс по руководству в Unity. — URL: <https://docs.unity3d.com/ru/2019.4/Manual/index.html> (дата обращения 10.05.2023).

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

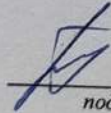
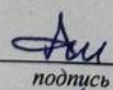
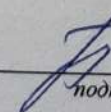
Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой

О.В. Непомнящий
"22" 06 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Компьютерная игра «Dolly»

Руководитель	 подпись	19.06.23 дата	доцент, к. т. н. должность, ученая степень	Л.И. Покидышева
Выпускник	 подпись	18.06.23 дата		А.Е. Мельникова
Нормоконтролер	 подпись	19.06.23 дата		Л.И. Покидышева

Красноярск 2023