Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.01 – Информатика и вычислительная техника
Профиль (направленность) образовательной программы Автоматизированные системы обработки информации и управления

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«_____»_____2021 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка мобильной игры на базе unity

Исполнитель
студент группы 753-об _____ М.А. Свитецкий
(подпись, дата)

Руководитель
доцент, канд. техн. наук _____ А. В. Бушманов
(подпись, дата)

Консультант
по безопасности и экологичности
доцент, канд. техн. наук _____ А.Б. Булгаков
(подпись, дата)

Нормоконтроль
доцент, канд.техн. наук _____ О.В. Жилиндина
(подпись, дата)

Благовещенск 2021

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет Математики и информатики

Кафедра Информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«_____»_____2021 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента Свитецкого М.А

1. Тема выпускной квалификационной работы: Разработка мобильной игры на базе unity

(утверждена приказом от 23.04.2021 № 812-уч)

2. Срок сдачи студентом законченной работы 24.06.2021

3. Содержание выпускной квалификационной работы: анализ предметной области; проектирование программного продукта; практическая реализация программного продукта; безопасность и экологичность.

4. Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов): А.Б. Булгаков, доцент, канд. техн. наук, раздел 4 «Безопасность и экологичность»

5. Дата выдачи задания 20.02.2021

Руководитель выпускной квалификационной работы: Бушманов А.В., доцент, канд. техн. наук

Задание принял к исполнению (20.02.2021): _____

(подпись студента)

РЕФЕРАТ

Дипломная (бакалаварская) работа содержит 80 с., 25 рисунков, 1 таблицу, 25 источников.

МОБИЛЬНЫЕ ВИДЕОИГРЫ, UNITY, ПРОГРАММНЫЙ ПРОДУКТ, ПРОЕКТИРОВАНИЕ, ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС.

В работе была изучена технология создания мобильных игр.

Цель работы – разработка мобильной игры.

Выполнение проекта включает ряд этапов:

– На первом этапе был проведен анализ предметной области.

– На втором этапе выполнено проектирование приложения. Для этого были проанализированы требования к программному продукту, выделены логические части программы, описано взаимодействие функциональных подсистем.

– На третьем этапе выполнена разработка программного продукта.

Результатом является прототип мобильной игры в жанре 2D платформер.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Смартфон сейчас есть почти у каждого. По данным исследования «DIGITAL 2021: GLOBAL OVERVIEW REPORT» 96,6% от всех интернет пользователей в возрасте от 15 до 64 имеют смартфон, а планшетный компьютер 34,3%. Для России эти показатели схожи – 94,9% и 34% соответственно.

Увеличение числа смартфонов стимулирует развитие отрасли мобильной разработки. Из года в год этот сегмент ИТ-рынка растет высокими темпами.

Игры долгое время были движущей силой сегмента мобильных приложений, но пандемия помогла им выйти на совершенно иной уровень: по сравнению с 2019 годом количество установок в 2020 увеличилось на 45%. Это на 40% выше, чем показатель роста прошлого года.

В настоящее время рынок мобильных игр является одним из самых быстрорастущих в мире. В отчете аналитической компании App Annie за 2020 год приводится следующая статистика:

- в прошлом году 72% трат пользователей всех мобильных приложений пришлось на игры;
- более 50% пользователей мобильных приложений играют в игры, что делает эту категорию приложений такой же популярной, как музыкальные приложения и уступает только приложениям для социальных сетей и коммуникаций с точки зрения затраченного времени;
- игры отвечают за 10% времени, которое пользователи тратят на работу с мобильными приложениями;
- из всех покупок цифрового контента 17,8 % были связаны с мобильными видеоиграми.

Увеличение доходов от мобильных игр продолжит опережать рост доходов от продажи ПК игр в следующем году, что в конечном итоге приведет к сокращению доли рынка игр для ПК. Кроме этого, доход от мобильных игр также опережает доход от игр на консолях.

По прогнозам В 2020 году компания Newzoo прогнозирует, что на игры

для ПК будет приходиться не более 20% дохода рынка, для консольных игр – не более 30%, а для мобильных устройств и планшетов от 46%, а к 2022 году вырастет до 50% всего игрового рынка.

Если сравнивать разработку компьютерных и мобильных игр то, стоит отметить что хотя высокопроизводительные мобильные процессоры, поддержка мощной графики, качественные экраны и быстрое интернет-соединение превратили смартфоны в игровые устройства, по техническим характеристикам они все же гораздо слабее стационарных компьютеров или консолей, и им не требуется такая же детальная проработка графики. Также пользователи не предъявляют к мобильным играм такие же требования по числу механик, степени проработки мира, времени прохождения, что позволит сократить как бюджет, так и время затраченное на разработку. При этом потенциальная аудитория будет больше чем у рынка ПК и консолей.

Также стоит отметить что особенности ввода для мобильных устройств, такие как сенсорный ввод и наличие акселерометра, позволяют создавать игры, дающие уникальный игровой опыт использующим эти особенности, например рисование на экране, наклон устройства для изменения гравитации в игре и т. д.

Как будет развиваться рынок мобильных игр, покажет время. Последние тенденции предполагают, что в обозримом будущем рынок продолжит генерировать все больший и больший доход. Совершенно ясно, что широко обсуждаемая революция мобильных приложений только начинается.

Эти данные говорят об актуальности разработки мобильных игр в 2021 году.

При этом создание игры это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов, включающий в себя как технические, так и творческие моменты.

Сначала необходимо сформулировать цель и провести анализ существующих средств разработки и выбрать наиболее подходящие. Творческий этап создания игры заключается в разработке игровых механик, создании уровней, сюжета, графики и звука. Процесс создания игр требует выполнения всех эта-

пов проектирования, и представляет собой итерационный процесс.

Цель исследования – разработка мобильной игры с использованием кросс-платформенной среды разработки Unity.

Задачи:

– провести сравнительный анализ и выбрать наиболее подходящую среду разработки;
– определить структуру приложения;
– спроектировать дизайн пользовательского интерфейса приложения;
– адаптировать дизайн под разные размеры экранов;
– разработать алгоритмы и механики игры;
– реализовать управление, адаптированное под мобильные устройства.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обоснование и обзор выбранного жанра игры

Видеоигра может быть исполнена как в 2D, так и в 3D, однако для каждого формата можно выделить свои особенности, которые делают его предпочтительнее для конкретного проекта.

В 2D-играх используется плоская графика, называемая спрайтами, которая не имеет трехмерной геометрии. Спрайты отображаются на экране как плоские изображения, а камера не имеет перспективы (ортогональная проекция). Пример двумерной графики можно увидеть на рисунке 1.

Рисунок 1 – Пример двумерной графики

В 3D-играх обычно используется трехмерное пространство, где материалы и текстуры отрисовываются на поверхности игровых объектов, формируя целостное окружение, персонажей и объекты игрового мира (рисунок 2) [1]. Сцена 3D-игры обычно отрисовывается в перспективе, поэтому с приближением объектов к камере они становятся больше.

Рисунок 2 – Пример трехмерной графики

В некоторых 2D-играх используются трехмерные окружения и персонажи, но игровой процесс ограничивается двумя измерениями. Например, камера демонстрирует окружение сбоку и движется слева направо, а игрок может двигаться только в двух измерениях. В таких играх эффект трехмерности играет скорее визуальную роль, нежели функциональную (рисунок 3).

Рисунок 3 – Пример изометрической графики

Кроме того, есть игры с трехмерной геометрией и глубиной, камера в которых строит изображение в ортогональной проекции, а не в перспективе. Это

популярный прием в играх, где игрок смотрит на происходящее с высоты птичьего полета, а камера зачастую строит изображение в изометрической проекции.

В разрабатываемом проекте применяется 2D графика по следующим причинам:

– упрощенный процесса создания анимаций по сравнению с 3D графикой;

– отсутствие необходимости разрабатывать 3D модели;

– меньшие требования к производительности устройства, что позволит охватить аудиторию владельцев маломощных смартфонов;

– большое число доступных для использования бесплатных материалов.

Для разработки выбран жанр 2D платформер. Это жанр компьютерных игр в которых основной чертой игрового процесса является перемещение по уровню в двумерной плоскости.

Жанр платформер достаточно популярен среди игроков, так как представляет собой простой и понятный, а так же интересный игровой процесс. Обладая очень низким порогом вхождения, такие игры доступны очень широкому кругу людей, в том числе людям, ранее не интересовавшимися мобильными играми.

Так же стоит помнить, что в индустрии мобильных игр не очень востребованы крупные проекты, требующие от игрока глубокого понимая механики игрового процесса. Для их прохождения зачастую требуется довольно большое количество времени. На их фоне проект с простым и понятным геймплеем будет смотреться более предпочтительно среди людей, которым сложно уделять достаточно много времени на игры.

Жанр платформер выбран по следующим причинам:

– данный жанр обладает простой и понятной механикой, что означает легкость в разработке [2].

– проще реализовать управление, адаптированное под мобильные устройства. При этом смартфоны и планшеты должно быть удобно держать чтобы играть одной/двумя руками.

– проще программировать взаимодействие между объектами. При обработке физики не требуется учитывать движение в третьем измерении.

1.2 Анализ аналогов

Наиболее близким аналогом разрабатываемой игры является «Namless Cat» от разработчика Kotoba Games. Это 2D игра в жанре платформер.

Игра имеет более одного миллиона установок на площадке Google Play.

Возрастной рейтинг не имеет ограничений.

Игра распространяется бесплатно, но имеет встроенную рекламу и поддерживает покупки внутри приложения. Однако покупки за настоящие деньги возможно восстановить.

Из особенностей можно отметить что игра имеет сюжет и оригинальное музыкальное сопровождение, способствующее погружению в игру.

Графика проекта реализована в стиле «Пиксель-Арт». Это форма цифрового изображения, созданного на компьютере с помощью растрового графического редактора, где изображение редактируется на уровне пикселей (точек), а разрешение изображения настолько мало, что отдельные пиксели чётко видны.

Игра переведена на несколько языков.

Требуемая версия ОС для устройств, работающих на платформе Android – 4.1 и выше, для iPhone (iOS 9.0 и выше), iPad (iPadOS 9.0 и выше) , iPod touch(iOS 9.0 и выше).

Размер приложений – 32 МБ.

Игровые данные хранятся в памяти устройства пользователя. Удаление или переустановка игры сбросят все внутриигровые данные.

Для работы требуются следующие разрешения:

– неограниченный доступ к интернету;

– просмотр сетевых подключений;

– отключение спящего режима;

– Play Install Reffer API (стандартный интерфейс для получения данных об установке приложения);

– платежная система Google Play;
– получение данных из интернета.

Для входа в игру не требуется регистрация или вод в учетные записи.

Приложение использует горизонтальную ориентацию экрана.

При запуске приложения проигрывается заставка с логотипом компании разработчика. Далее открывается окно ожидания. В нем демонстрируется логотип компании, название игры, присутствуют анимированные изображения. Текстовые надписи слегка колеблются вокруг своего положения. Также у текстовых надписей присутствует эффект мерцания. Начинает проигрываться фоновая музыка, представляющая собой циклически проигрываемый музыкальный отрывок.

Для всех текстовых надписей в игре используются шрифты, органически вписывающиеся в пиксельную графику игры.

При переходе между окнами отображается анимированная заставка, длящаяся до загрузки следующего окна, сам переход к которой происходит плавно, в течение секунды.

Во время игры при длительном бездействии экран уменьшает яркость до минимума, но устройство не переходит в спящий режим.

При нажатии на любую точку экрана происходит переход в окно выбора уровня (рисунок 4). В нем также присутствуют анимированные элементы. Окно разбито на три панели, переход к каждой может осуществляться либо с помощью стрелок в нижних углах экрана, либо с помощью свайпа (управляющий жест при работе с сенсорным экраном, при котором палец кладут на экран и проводят в каком-либо направлении) в соответствующую сторону. Переход между панелями осуществляется плавно, путем смещения предыдущей панели за пределы экрана. Если открыта крайняя панель, стрелка перехода в эту же сторону становится недоступна, а свайп влево незначительно сдвигает панель, которая по окончании нажатия на экран сразу возвращается на прежнее место. В Unity подобного эффекта можно добиться используя компонент «UI ScrollRect», который в пользовательском интерфейсе отображается как прямо-

угольник, который можно прокручивать по горизонтали или вертикали для создания маскированных (ограниченных пределами прямоугольника) пространств со скрытым содержимым.

Рисунок 4 – Окно выбора уровня

В левом верхнем углы отображается счетчик внутриигровой валюты.

На каждой панели размещаются кнопки для выбора соответствующего уровня. Для заблокированных уровней используется отдельная иконка и отключена реакция на нажатие. Также радом с иконками выбора уровня находится изображение, показывающее подобрана ли на уровне единица внутриигровой валюты. Изначально доступен только первый уровень, а остальные открываются последовательно, после прохождения предыдущего.

При нажатии на телефоне клавиши «Отмена» вызывается панель, подтверждающая выход из приложения. На панели содержится текстовая надпись и две кнопки, в виде текстовых надписей, для отмены и подтверждения выхода из игры. При этом панель затеняет содержимое остальных элементов интерфейса, в результате на контрасте повышается читаемость надписей.

Все кнопки в окне реагирует на нажатие изменением цвета иконки и небольшим смещением вниз, что указывает пользователю на возможность интерактивного взаимодействия.

В правом верхнем углу отображается кнопки для перехода в окно выбора

скина персонажа, окно достижений, и окно настроек.

В окне выбора скина (пакет данных, предназначенный для настройки внешнего вида) персонажа игрок может выбрать внешний вид персонажа. Изначально доступен только один скин. Часть скинов покупается за внутриигровую валюту, а часть за реальные деньги. Оплата совершается с использование платежной системы Google Pay.

В окне достижений размещены три панели. На первой отображаются достижения игрока (необязательные связанные с прогрессом прохождения).

Для каждого достижения указан его заголовок, условие получения и прогресс игрока в его получении.

Выполненных достижения визуально выделяются цветом фона, а вместо прогресса отображается дата их получения.

На второй панели отображаются собранные игроком коллекционные предметы.

На третьей панели показа внутриигровая статистика (общее время игры, количество смертей, количество прыжков и т.д.).

Окно настроек позволяет выбрать язык, используемый в приложении, отключить звуки и фоновую музыку, стереть данные (в результате сбросится весь прогресс прохождения, в том числе статистика), отключить рекламу (данная функция реализована в виде покупки). Также окно содержит ссылку на политику конфиденциальности приложения.

После выбора уровня осуществляется переход непосредственно в игру.

В начале некоторых уровней отображается кат-сцены (внутриигровое видео в котором игрок никак не может влиять на происходящие события), раскрывающие сюжет игры. Также в начале каждого уровня начинается воспроизведение фоновой музыки.

Пользовательский интерфейс состоит из кнопок управления, кнопки вызова паузы. Также присутствует индикатор сбора внтриигровой валюты. На каждом уровне размещается только по одной единице. Обычно её сбор связан с решением определенной логической задачи, изучением уровня и требует от иг-

рока хорошей реакции, проявления смекалки и навыков владения игровыми механиками.

Меню паузы позволяет отключить звуки и фоновую музыку, вернуться в главное меню, переиграть уровень, отключить рекламу. Также есть настройка позволяющая изменить положение кнопок управления но только в пределах установленных разработчиком. Вызов меню паузы останавливает действие игры и затеняет содержимое кадра.

Для управления служат 4 кнопки, отвечающие за следующие действия – движение влево, движение вправо, прыжок, «вселение» (основная механика игры, которая изначально недоступна но открывается по мере прохождения игры).

Управление не имеет инерции, то есть при нажатии кнопки движения персонаж сразу начинает перемещаться с фиксированной скоростью, а если отпустить кнопку движения, персонаж сразу остановиться, но при прыжке горизонтальная скорость сохраняется. Высот прыжка фиксирована и не зависит от продолжительности нажатия на кнопку.

При нажатии кнопок движения проигрывается анимация бега. При падении или прыжке проигрывается анимация падения, которая может прервать выполнение всех остальных анимаций. При приземлении проигрывается анимация «пылевого облака» и воспроизводится звуковой клип приземления.

В игре существует система диалогов. Переход в режим диалога происходит при приближении к определенному персонажу, также может потребоваться нажатие клавиши взаимодействия. При этом скрываются все элементы управления. Над персонажем, произносящим реплику появляется анимированное диалоговое окно. Текст выравнивается по центру окна, перенос слов по слогам не используется. Также текст может быть выделен цветом. Переход к новой реплике осуществляется только после нажатия на экран. Также возможно прервать диалог. К верхней и нижней границе кадра добавляются черные полосы, служащие для достижения кинематографического эффекта.

Сами уровни линейны и не предполагают вариативности прохождения.

Время, затрачиваемое на прохождения одного уровня обычно занимает не более 5 минут. Для прохождения уровня обычно требуется дойти до определенной точки. Всего в игре 45 уровней. Они разбиты на 3 группы по 15 уровней. Каждая группа имеет собственный визуальный стиль, музыкальное сопровождение, интерактивные элементы на уровнях.

Основная механика в игре это «вселение» - вид взаимодействия с определенными интерактивными объектами размещенными на уровне. Взаимодействие заключатся в следующем:

– Когда персонаж игрока находится рядом с объектом, допускающим «вселение», над спрайтом (отображаемое непосредственно на экране двумерное изображение) появляется изображение, сигнализирующее о возможности взаимодействия;

– При нажатии кнопки «вселения» спрайт персонажа исчезает, камера фокусируется на центре объекта. При этом отображается анимированный след, направленный от прошлого местоположения игрока к центру объекта. В этом состоянии игрок не получает урона. Из управления становится доступна только кнопка «вселения». При повторном нажатии персонаж игрока появится над объектом.

Существуют разные типы объектов «вселения». Они могут быть стационарные (не могут передвигаться игроком) и подвижные. При этом объекты делятся на те что сохраняют исходную скорость игрока, либо сбрасывают её. Также существуют объекты которые при «вселении» меняются местоположениями с игроком.

При получении урона или выход за установленные границы уровня наступает «смерть» персонажа. Уровень загружается заново, а положение игрока находится в последней пройденной контрольной точке.

Контрольная точка – вид интерактивных объектов, служащий для сохранения прогресса внутри уровня. Активация контрольной точки происходит при прохождении рядом с ней. При этом проигрывается анимация, уведомляющая что прогресс сохранен. Также существуют контрольные точки, которые акти-

вируются только после завершения просмотра рекламы. Рекламные ролики проигрываются после каждых пяти «смертей персонажа». В игре отсутствуют рекламные баннеры и всплывающие окна.

Урон можно получить при столкновении со специальными объектами, расположенными на уровне. Они могут быть стационарными или двигаться по определенной траектории и имеют разное графическое оформление.

В игре существует несколько видов противников. Есть три уникальных противника «босса», каждый из которых имеет уникальную линию поведения и для прохождения которых выделены отдельные уровни.

Также есть рядовые противники которые встречаются на всех уровнях. Урон наносится при столкновении с ними. Каждый вид рядовых противников имеет свои уникальные особенности:

- противник, перемещающийся по заранее определенной траектории. Если игрок появится на линии его движения, ускоряется до столкновения с ближайшим препятствием;
- противники, перемещающиеся по заранее определенной траектории с одинаковой скоростью. Может атаковать игрока, находящегося в состоянии «вселения»;
- противники, появляющиеся в определенной точке с заданной периодичностью, и движутся по направлению к игроку. При движении не огибают препятствия и уничтожаются после столкновений с любым объектом.

Помимо перечисленных элементов в игре присутствуют переключатели – интерактивные объекты, служащие для запуска других элементов уровня. Обычно они позволяют разблокировать изначально закрытый участок пути. Эти элементы указывают игроку на очередность действий, тем самым формируя логику прохождения уровня.

В целом дизайн уровней спроектирован так, чтобы игрок не мог нарушить порядок прохождения уровня, предусмотренный разработчиком.

1.3 Обзор программных средств

Unity – это кроссплатформенный SDK для разработки игр с двухмерной и

трехмерной графикой, распространяемый по условно бесплатной модели. Использовать его можно безвозмездно, но на разработчика накладываются ограничения (доход не более $100 тыс. в год). Движок работает с API DirectX, OpenGL, для работы с физикой задействован PhysX [3]. Скрипты пишутся на C#.

Unity обладает интуитивно-понятным интерфейсом и прост в освоении. Движок использует компонентно-ориентированный подход. Он поддерживает модульность при разработке игр и упрощает подключение объектов. На движке возможна разработка игр любого жанра с графикой любого уровня.

Unity является одним из самых популярных сред для разработки мобильных игр [4].

В целом из достоинств можно выделить:

- кросплатформенность;
- высокая функциональность;
- большое сообщество разработчиков;
- удобство в работе и компиляции;
- большое количество библиотек расширяющих функционал.

Также можно отметить следующие недостатки:

- закрытый исходный код;
- условно-бесплатная модель распространения;
- Большой размер генерируемых файлов.

Перечисленные особенности делают Unity подходящей средой разработки для реализации проекта.

Unreal Engine – игровой движок компании Epic Games, ориентированный на AAA-проекты и проекты в 3D. Он условно бесплатен при некоммерческом применении, но если проект приносит больше $3 тыс. в квартал – разработчикам движка нужно платить авторские отчисления в размере 5% от выручки.

Unreal дает разработчику большой набор простых в освоении и интуитивно понятных инструментов. C++ накладывает минимум ограничений во время написания скриптов, а система визуального программирования Blueprint

облегчает прототипирование или написание скриптов. Создавать элементы игры можно наглядно, перемещая объекты, без ручного ввода кода.

В фирменном магазине доступен ассортимент готовых шаблонных решений. Они подойдут разработчикам, заинтересованным в быстром завершении игрового проекта. Среди прочих особенностей SDK – регулярные обновления, достаточно большое сообщество и поддержка разработчиков.

Поддерживает большинство известных платформ: Microsoft Windows, Linux, Mac OS и Mac OS X; консолей Xbox, Xbox 360, Xbox One, PlayStation 2, PlayStation 3, PlayStation 4, PSP, PS Vita, Wii, Dreamcast, GameCube, Nintendo Switch и т.д., в том числе iOS и Android.

В версии 4.0 присутствует мощный редактор искусственного интеллекта, редактор для создания кат-сцен. Поддерживается технология DirectX 12.

В целом Unreal Engine не очень подходит для разрабатываемого прооекта из-за его ориентированности на высокоуровневые 3D-игры.

Например файлы APK сгенерированные Unreal Engine 4 больше, чем файлы, созданные с помощью Unity, и, как правило, для работы требуются более высокие системные требования.

Также в Unreal Engine используется язык C++, который сложнее в изучении и использовании чем C#, используемый в Unity.

Unreal Engine имеет меньший, по сравнению с Unity размер сообщества.

Corona – Это бесплатный фреймворк, ориентированный под написание приложений для мобильных и стационарных платформ. Он использует широко известный язык Lua, отличающийся простотой освоения и универсализмом. Модульная конфигурация поддерживает подключение внешних API и расширений. В магазине доступно более двух сотен плагинов, отвечающих за эффекты, аналитику, мультимедиа, интеграцию рекламы и другие функции во время разработки ПО.

Встроенный эмулятор Android и iOS позволяет тестировать примененные решения и просматривать результаты в режиме реального времени. Система Live Build обеспечивает тесты разрабатываемой программы без ручной уста-

новки, так как обновление выполняется автоматически.

Плюсы Corona:

– бесплатная модель распространения;
– кроссплатформенность;
– удобство отладки и тестирования проектов.

Corona не подходит для разрабатываемого проекта по следующим причинам:

– для компиляции нужен интернет. Платформа отправляет байт-код на сервера Corona, которые компилируют его в исполняемый файл;
– нельзя добавлять сторонние плагины или библиотеки, только покупать в магазине Corona;
– Меньшее число возможностей по сравнению с Unity.

Godot - полнофункциональный игровой движок с открытым исходным кодом. Имеет встроенную физическую систему, поддерживающую рейкастнг, обнаружение столкновений, динамику твёрдых тел и соединений между ними. Также имеется собственная реализация кинематического контроллера персонажа и 3D-контроллер автотранспортных средств с упрощённой системой подвески. Есть встроенные алгоритмы поиска пути с обходом препятствий. Реализован визуальный редактор шейдеров, представлены инструменты скелетной анимации и визуальные эффекты, основанные на частицах. Есть возможность использовать скриптинг на C++/D/Rust и другие языки через систему GDNative. Редактор анимаций позволяет анимировать практически любой параметр, присутствующий в проекте. В качестве языка программирования предложен GDScript, по синтаксису похожий на язык программирования Python. Начиная с версии 3.0 появилась возможность использовать систему визуального программирования Visual Scripting.

Проект в Godot представляет собой древовидную структуру, каждый узел которой может служить самостоятельной сценой. Объекты могут общаются между собой с помощью так называемых сигналов, которые могут содержать в себе переменные.

Среда разработки Godot не подходит для разрабатываемого проекта по

следующим причинам:

– по сравнению с Unity имеет ограниченные возможности смешивания 2D и 3D элементов в одной и той же сцене;

– от пользователя требуется намного больше усилий по настройке проекта, в то время как в Unity многие вещи обрабатываются автоматически «за кулисами». С Godot пользователь должен вручную контролировать выделение памяти, освобождение ресурсов, управление шейдерами и материалами и т. д.;

– могут возникать нежелательные изменения в скорости рендеринга, включая загрузку и выгрузку материала, потому что движок рендеринга Godot и система управления памятью не так развиты и оптимизированы, как Unity. Даже наличие слишком большого количества материалов в сцене в Godot замедлит рендеринг (FPS) для сканирования;

– имеет меньшее по сравнению с Unity число встроенных компонентов и возможностей.

1.4 Обоснование выбора ПО

В качестве среды разработки был выбран игровой движок Unity по следующим причинам:

– компонентно-ориентированный подход — разработчик прописывает объекту компоненты вроде возможности управления объектом и модели поведения;

– кросплатформенность;

– интеграция сервисов;

– условно-бесплатная модель распростаренения;

– большая библиотека ассетов и плагинов, которые можно использовать для создания прототипа и готовой игры;

– надежность и стабильность движка;

– широкие возможности по оптимизации;

– подробная документация, множество обучающих материалов, развитое сообщество.

В среде Unity для написания управляющих скриптов используется C#, по-

этому именно он был выбран в качестве языка программирования.

C# – объектно-ориентированный язык программирования. Разработан в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java [5]. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Переняв многое от своих предшественников – языков C++, Pascal, Модула, Smalltalk и, в особенности, Java – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

Для написания кода был выбран текстовый редактор Visual Studio Code. Это редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для настройки: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом.

Был выбран для использования из-за быстрого запуска, малого объема занимаемой оперативной памяти, расширяемости функционала за счет легковесных плагинов, поддержки отладчика для Unity, удобной работы с системой контроля версий.

Для подготовки графических ресурсов были выбраны графические редакторы Adobe Illustrator и Adobe Photoshop.

Adobe Photoshop – мощный графический редактор для фотографий и ри-

сования, разработанный и поддерживаемый Adobe Systems [6]. Работает с растровыми изображениями, однако имеет несколько векторных инструментов. Лидер рынка в области коммерческих средств редактирования растровых изображений и наиболее известный продукт фирмы Adobe. Изначально задумывался как мощный графический редактор для растровых изображений в полиграфии, но быстро стал популярен в области Web-дизайна. Позволяет работать в большинстве популярных цветовых моделей.

Выбран по следующим причинам:

- широкий набор инструментов, кистей, эффектов для работы с графикой;
- большое число поддерживаемых форматов графических файлов;
- интеграция в экосистему Adobe.

Adobe Illustrator - инструмент для векторного рисования и редактирования художественных работ. Он помогает создавать графические проекты во всех стандартных форматах, а также предоставляет функции печати, размещения на веб-сайтах. Illustrator предоставляет множество инструментов для работы над графическими объектами, включая стандартные средства рисования, выделения, перемещения, масштабирования и трансформации объектов. При работе с текстом программа позволяет применять различные эффекты, шрифты, стили, форматирование и редактировать отдельные символы.

Платформа предлагает опции разнообразных кистей и карандашей, поиска и загрузки шрифтов, синхронизации цветов, инструменты кривизны и автоматической калибровки. Можно создавать привязку к пикселям, точкам и сетке, а также фигуры, в том числе эллипсы, многоугольники, линии и закруглённые прямоугольники. Доступна настройка пользовательских шаблонов графики и мгновенное масштабирование.

Выбран по следующим причинам:

- большой набор инструментов для работы с векторной графикой;
- возможность работы с форматами векторных и растровых изображений;
- интеграция в экосистему Adobe.

2 ПРОЕКТИРОВАНИЕ

2.1 Анализ требований к программному продукту

Данный программный продукт является мобильной видеоигрой. Исходя из анализа аналогов можно выделить следующие требования к разрабатываемому продукту:

Он должна поддерживаться на мобильных устройствах с операционной системой Android 7.0 и выше.

Должны поддерживаться устройства с разными разрешениями экрана и соотношением сторон.

Игра должна иметь систему сохранения прогресса пользователя (число пройденных уровней и прочая игровая статистика), а также функцию сброса прогресса.

Должна быть система локализации игры.

Должна быть система управления, адаптированная под мобильные устройства.

Игра должна иметь возможность регулирования громкости звуков и фоновой музыки.

Внутри уровней должна быть реализована система чекпоинтов (сохранения внутри уровня).

Должна быть реализована система загрузки уровней с отображением прогресса загрузки.

Должна быть система покупки и смены скинов.

2.2 Логические части программы

В соответствии с требованиями, предъявляемыми к программному продукту были выделены две логические части программы: главное меню и уровень.

В главном меню пользователь должен иметь следующие возможности: просмотра достижений, выбора или покупки скина, выбора уровня, настройки параметров громкости, сброса данных. Диаграмма вариантов использования

для главного меню представлена на рисунке 5.

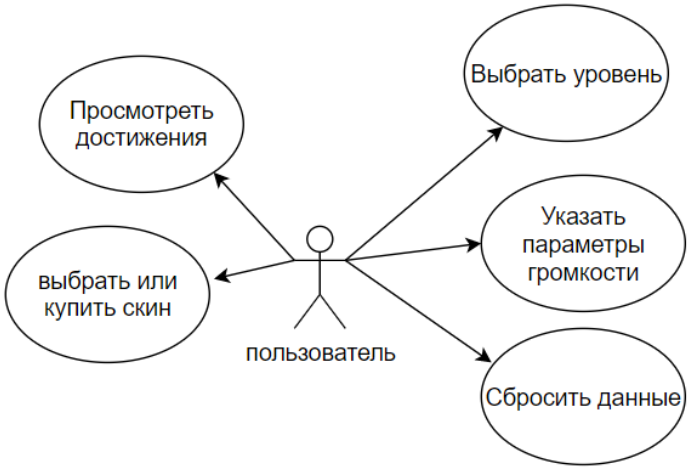

Рисунок 5 – Диаграмма вариантов использования для главного меню

Главное меню состоит из нескольких логических частей:

– Окно выбора уровня. Отвечает за выбор уровня игроком. При этом доступ на еще не открытые уровни должен быть заблокирован.

– Окно выбора скина. Содержит счетчик имеющихся монет (внутриигровой валюты), список скинов, а также кнопки для их выбора или покупки. Информация о покупках и смене скина должна сохраняться.

– Окно просмотра достижений. Содержит список достижений (необязательных заданий выдаваемых игроку). Каждый элемент списка включает описание условия получения достижения и прогресс в его получении. Данные для проверки условий получения берутся из подсистемы сохранений.

– Окно настроек. Позволяет пользователю указывать параметры громкости для звуков и фоновой музыки, выбирать язык игры. Выбранные значения должны сохраняться. Также должна присутствовать команда сброса игрового прогресса.

Выделенные логические части взаимодействуют в основном с подсистемой сохранений. Их взаимодействие показано на рисунке 6.

При этом часть сохраненной информации (связанная с прогрессом игрока) сохраняется в виде файла и может быть удалена. Другая часть (настройки звука, выбранный язык) хранится отдельно и не сбрасывается при удалении

файла сохранений.

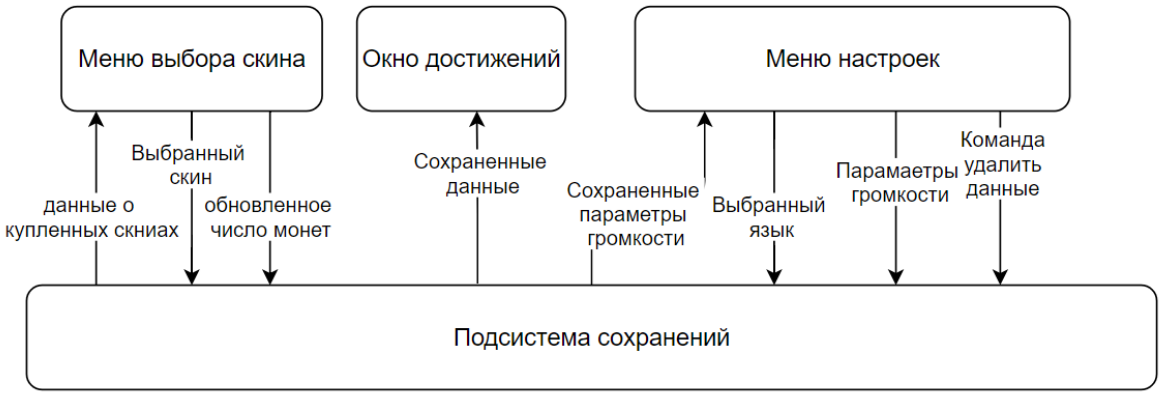

Рисунок 6 – Взаимодействие главного меню и подсистемы сохранений

На рисунке 7 изображено взаимодействие главного меню с подсистемами.

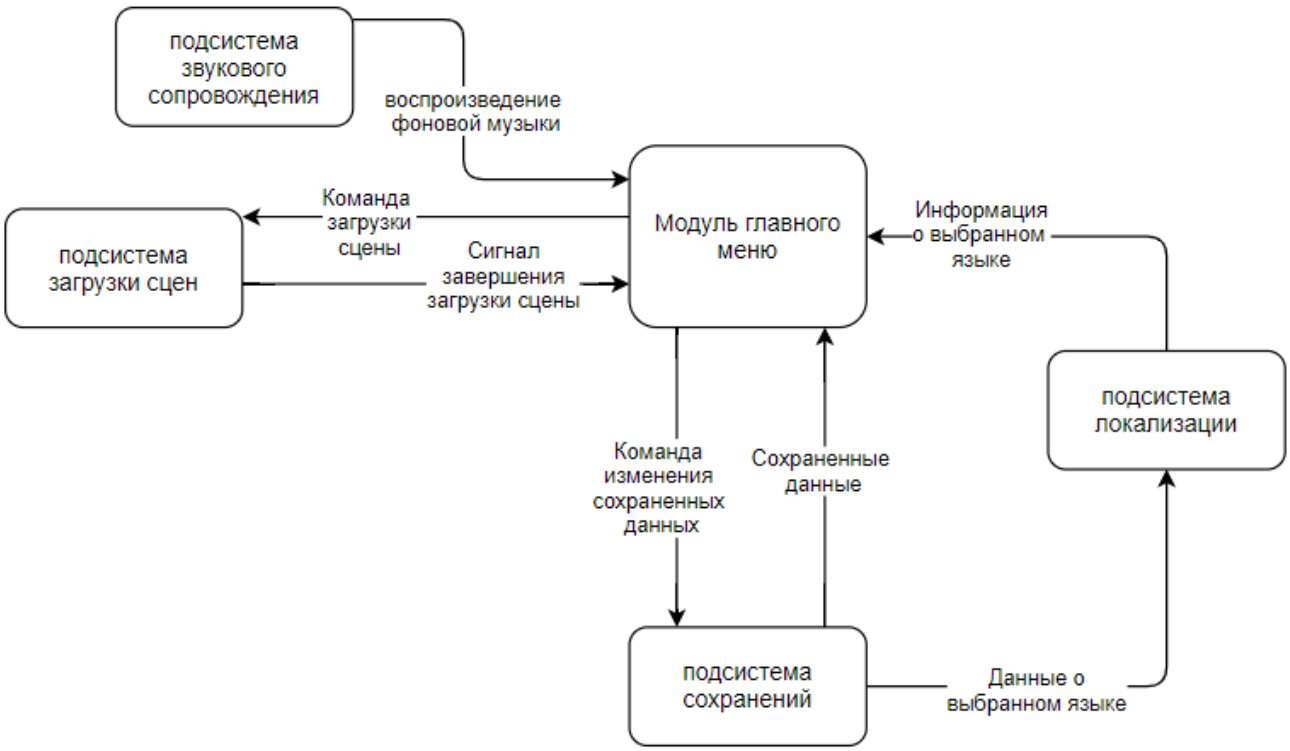

Рисунок 7 – Взаимодействие главного меню с подсистемами

Уровень был выделен как вторая логическая часть программы. Каждый уровень уникален, но при этом все они построены по одному принципу и содержат общие элементы.

Исходя из требований программного продукта для уровня также была разработана диаграмма вариантов использования (рисунок 8).

Во время прохождения уровня пользователю должна быть доступна вся

необходимая информация, влияющая на игру [7]. Например это может быть информация о числе монет. Также должна быть создано меню уровня из которого можно выйти в главное меню, перезагрузить уровень, изменить параметры звука. При этом когда меню уровня открыто игра должна быть приостановлена.

Рисунок 8 – Диаграмма вариантов использования для уровня

Взаимодействие подсистем на уровне показано на рисунке 9.

Рисунок 9 – Взаимодействие подсистем на уровне

2.3 Описание подсистем

Подсистема персонажа отвечает за управление персонажем, прием звуковых сигналов, реакцию на взаимодействие с интерактивными объектами. Ввод данных для управления осуществляется с помощью джойстика, отображаемого

на экране и управляемого сенсорным вводом.

Подсистема звукового сопровождения отвечает за воспроизведения звуков и фоновой музыки в игре.

Подсистема сохранения данных отвечает за сохранения прогресса игрока и внутриигровой статистики.

Подсистема инициализации уровня отвечает за действия, выполняемые во время загрузки уровня и выхода из него.

Подсистема чекпоинтов отвечает за сохранение прогресса игрока внутри уровня.

Подсистема UI уровня отвечает за вывод на экран данных, требуемых пользователю во время игры. Также включает меню паузы.

Подсистема загрузки сцен отвечает за плавный переход между сценами и отображения прогресса загрузки.

Подсистема интерактивных объектов включает совокупность объектов на уровне, с которыми может взаимодействовать персонаж, и реализует их реакцию на это взаимодействие.

3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

3.1 Создание главного меню

Главное меню представляет собой отдельную сцену. В настройках проекта сцена указана первой в иерархии, поэтому при загрузке игры загружается именно она. Сцена состоит из двух частей: пользовательского интерфейса и фона. Также в сцене находится объект, отвечающий за воспроизведение фоновой музыки.

3.1.1 Создание фона главного меню

Фон главного меню состоит из нескольких статичных изображений гор и неба, которым присвоен определенный порядок сортировки.

Также на фоне присутствуют изображения облаков, которые перемещаются по экрану, чтобы создать эффект динамики. Для этого был написан скрипт, общий алгоритм работы которого показан на рисунке 10.

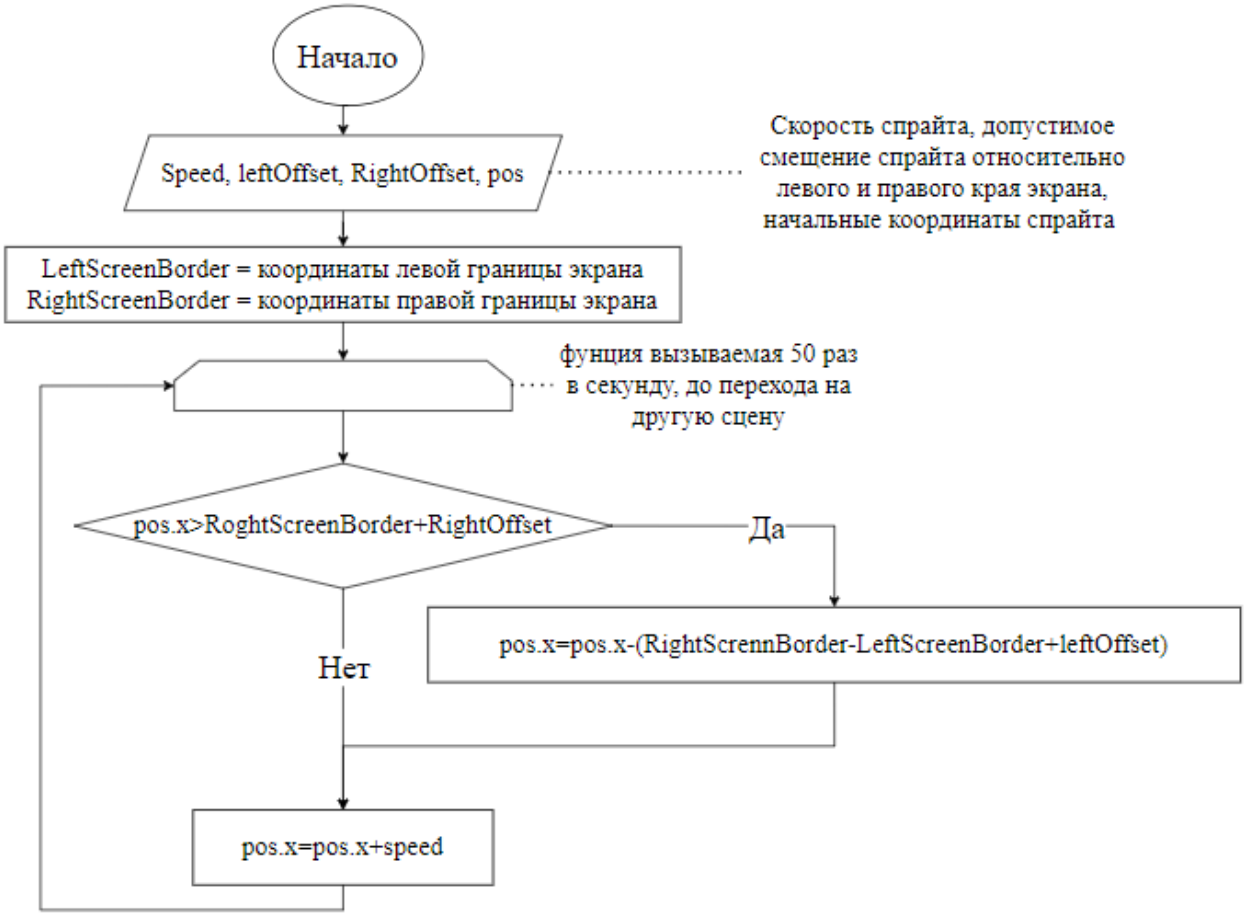

Рисунок 10 – Алгоритм движения спрайта облака

Скрипт функционирует следующим образом:

Вначале с помощью методов, определенных в компоненте камеры получаются позиция левой и правой сторон экрана в мировых координатах. Эти значения записываются в статические переменные доступные всем экземплярам класса облаков на сцене.

Также у класса определена публичная переменная скорости. Это позволяет назначить каждому облаку собственную скорость, в зависимости от его размеров. Если назначить более крупным и детальнее прорисованным облакам скорость больше, чем маленьким, то удастся добиться эффекта параллакса (эффект смещения близкого объекта относительно далекого при изменении угла зрения), что создаст иллюзию объемного пространства и движения.

Помимо скорости и координат границ экрана в классе облаков определены две переменные определяющие величину смещения спрайта за левую и правую границы экрана. Эти переменные необходимы, так как координаты всего спрайта определяются положением его точкой вращения. При этом левая и правая границы спрайта могут находиться от этой точки на произвольном расстоянии. Вводимые переменные индивидуальны для каждого спрайта и косвенно характеризуют его размеры. Их величины подбираются так, чтобы при сложении их с координатами границы экрана спрайт полностью находился за этой границей.

Положение каждого облака изменяется в не зависящей от частоты кадров функции, вызываемой 50 раз в секунду.

При каждом вызове функции проверяются координаты спрайта каждого облака. Если они больше чем сумма координат экрана и значение смещения спрайта за правую границу, то спрайт перемещается за левую границу экрана. Для этого спрайт перемещается влево на значение равное сумме ширины экрана, вычисленной как разница координат его левой и правой границ, и смещения спрайта за левую границу. В противном случае спрайт смещается вправо на значение скорости. Таким образом удается реализовать зацикленное движение спрайтов облаков.

3.1.2 Создание экрана главного меню

Экран главного меню состоит из элементов пользовательского интерфейса (UI). Пользовательские интерфейсы отображаются прямо на экране как простые элементы. Это значит, что они не имеют понятия 3D пространства, отображаемого камерой и размещены в экранном пространстве [8].

Для создания экрана главного меню сначала в сцене был создан пустой объект, к которому был прикреплен компонент canvas. Этот компонент представляет собой абстрактное пространство, в котором производится настройка и отрисовка элементов пользовательского интерфейса.

Экран главного меню состоит из следующих объектов:

- Надпись приветствия;
- Панель;
- Меню выбора скина;
- Меню просмотра достижений;
- Меню выбора настроек;
- Меню выбора уровня;
- Кнопки, отвечающие за вызов меню достижений, настроек, выбора скина.

Надпись приветствия нужна чтобы дать понять пользователю об окончании загрузки игры и обеспечить плавный визуальный переход от заставки, проигрываемой во время загрузки до отображения главного меню игры.

Она выводит на экран текстовое сообщение, призывающее пользователя нажать на экран. Надпись представляет собой текстовый элемент TextMeshPro. Положение якоря привязано ко всем сторонам экрана. К текстовому элементу прикреплен компонент Localize string event, отвечающий за перевод текста в зависимости от выбранного пользователем языка. Чтобы зарегистрировать нажатие на экран и вызвать функцию, отвечающую за отображение содержимого главного меню, используется компонент EventTrigger.

Панель – это элемент пользовательского интерфейса с прикрепленным компонентом изображения. Размер указан так, чтобы панель занимала весь экран, что позволяет перекрыть все содержимое сцены. В настройках панели прозрачность изображения указана так, чтобы содержимое сцены оставалось

видимым, но при этом текстовые надписи была хорошо видны на ярком фоне.

3.1.3 Разработка меню выбора скина

Это меню позволяет игроку выбрать скин (пакет данных, предназначенный для настройки внешнего вида) управляемого персонажа. Меню вызывается по нажатию соответствующей кнопки. Итоговый вид меню показан на рисунке 10.

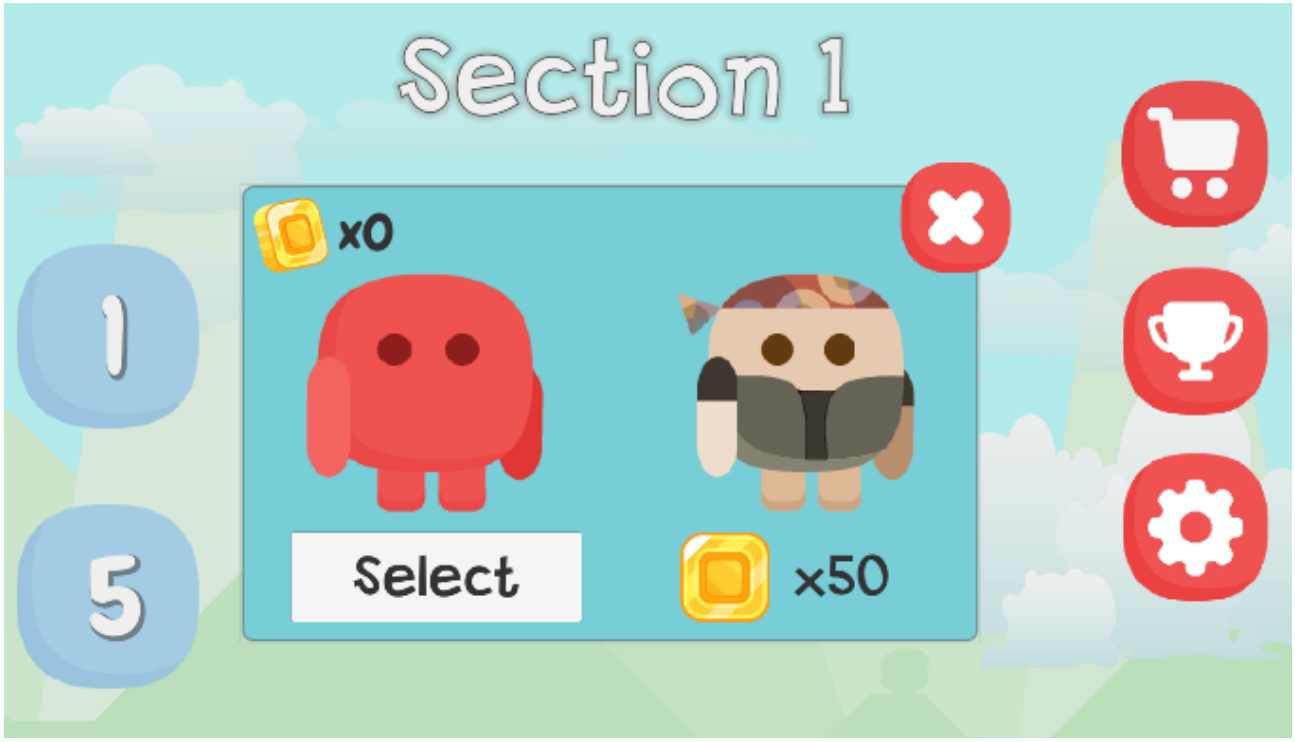

Рисунок 10 – Итоговый вид меню выбора скина персонажа

Меню выбора скина состоит из следующих элементов:

– Панель, которая служит для визуального выделения меню на фоне остальных элементов интерфейса. В иерархии графических элементов расположена выше окна выбора уровня, но ниже кнопок вызова меню, поэтому перекрывает содержимое окна, но не перекрывает кнопки. Также панель блокирует нажатия пользователя на перекрытой области, что позволяет предотвратить случайный запуск уровня.

– Кнопка выхода, позволяющая закрыть меню. При нажатии вызывает в управляющем скрипте функцию закрытия панели. Также закрыть панель можно щелкнув на кнопку вызова меню.

– Счетчик монет. Состоит из изображения и текстового элемента. При открытии окна управляющий скрипт обновляет значение текстового поля. Также обновление происходит при покупке скина.

– Кнопка выбора скина. В качестве изображения кнопки используется спрайт, представляющий внешний вид скина. Текстовый элемент кнопки удален за ненадобностью. При нажатии на кнопку вызывается функция управляющего скрипта, отвечающая за выбор скина. В качестве аргумента в функцию передается идентификатор выбранного скина. Также созданы две дополнительные кнопки: «выбор» и «покупка», которые являются дочерними по отношению к кнопке выбора спрайта.

Кнопка «выбор» показывает пользователю, что скин куплен и доступен для выбора. Кнопка «покупка» отображает его стоимость. Обе кнопки при нажатии вызывают функция выбора скина в управляющем скрипте.

На данный момент создано два скина персонажа. Каждому назначены копия кнопки выбора, соответствующий спрайт и указан передаваемый идентификатор.

Также для данного меню был разработан управляющий скрипт, алгоритм работы которого показан на рисунке 11.

При открытии меню скрипт обращается к массиву, определенному в скрипте, отвечающем за сохранения и пытается загрузить данные о купленных скинах. Данные хранятся в виде массива, где индекс элемента соответствует идентификатору скина, а значение – стоимости скина. Считается что данные отсутствуют, если ссылка на массив равно нулю. Также управляющий скрипт содержит массив, в котором хранится стоимость скинов по умолчанию. Он используется если данные не удалось загрузить. Организация данных в виде массива позволяет добавлять новые скины без написания нового кода. От разработчика потребуется лишь изменить пользовательский интерфейс, чтобы обеспечить возможность просмотра всех вариантов и задать их стоимость по умолчанию в окне инспектора.

Далее каждому скину назначается кнопка «выбрать» или «купить», в за-

висимости от того куплен ли он. Скин считается купленным если соответствующее ему значение стоимости равно нулю.

Затем скрипт обращается к полю класса сохранений, отвечающему за хранение числа монет, собранных игроком, конвертирует его значение в строку и присваивает её текстовому элементу счетчика монет.

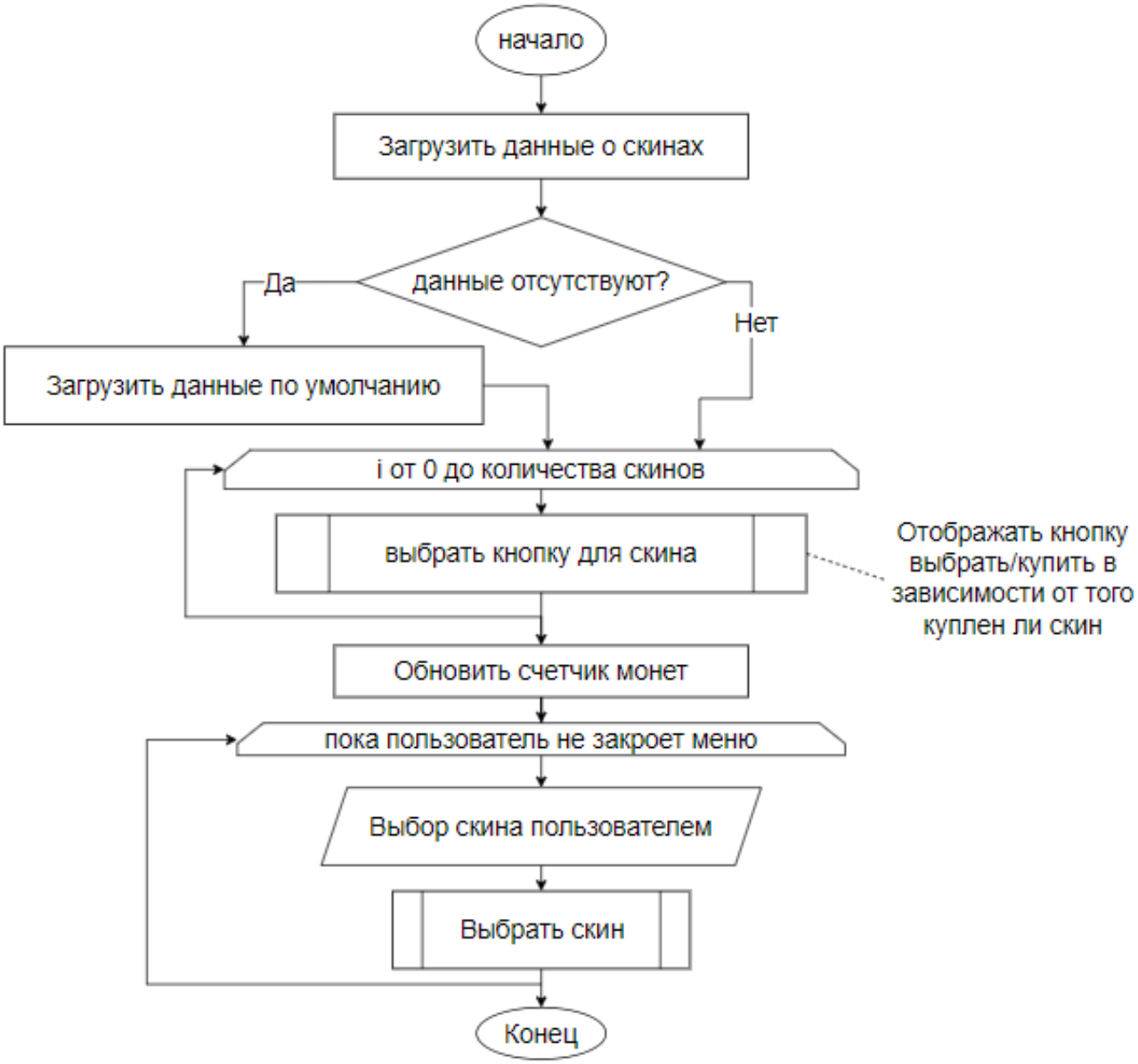

Рисунок 11 – Алгоритм работы управляющего скрипта меню выбора скина

После этого начинается ожидание выбора скина пользователем. При выборе вызывается функция, алгоритм выполнения которой представлен на рисунке 12.

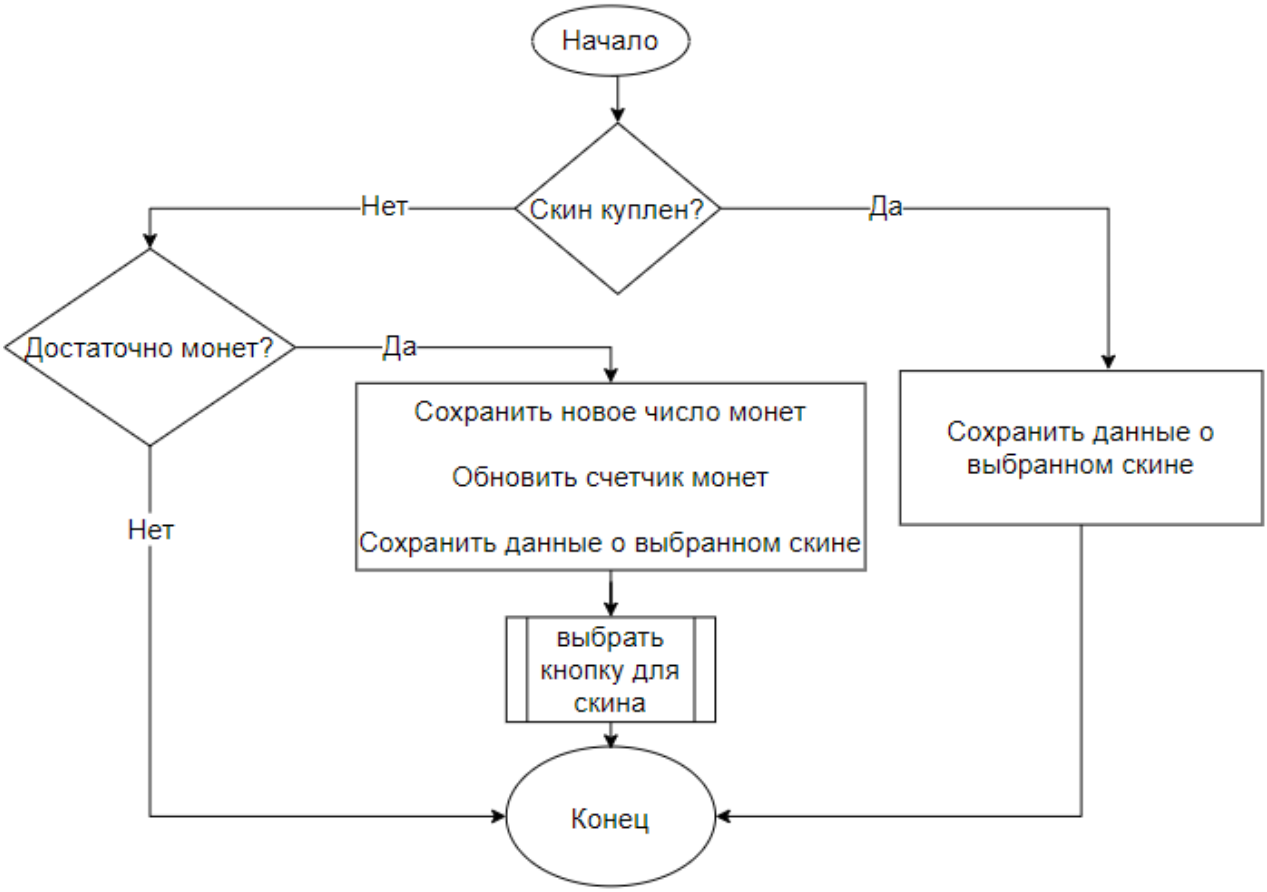

Рисунок 12 – Алгоритм выполнения функции выбора скина

При вызове в функцию в качестве аргумента передаются идентификатор выбранного скина. Сначала скрипт определяет был ли скин уже куплен. Скин считается купленным если значение соответствующего элемента массива, хранящего данные о купленных скинах, равно нулю. Если он куплен, то полю класса сохранений, хранящему идентификатор выбранного спрайта, присваивается значение аргумента функции. Если же нет, то проверяется, достаточно ли у игрока монет для покупки. В случае когда монет достаточно, полю в классе сохранений, хранящему число монет, присваивается разница его старого значения и стоимости скина. В соответствии с новым значением также обновляется счетчик монет. Происходит сохранение идентификатора выбранного скина. Также соответствующему элементу массива данных о купленных скинах присваивается значение ноль. Если монет недостаточно, то функция завершает свое выполнение.

Если произошла покупка скина, то при закрытии окна произойдет вызов

функции, отвечающей за запись сохраненных данных в файл.

3.1.4 Разработка меню достижений

Внешний вид меню достижений представлен на рисунке 13.

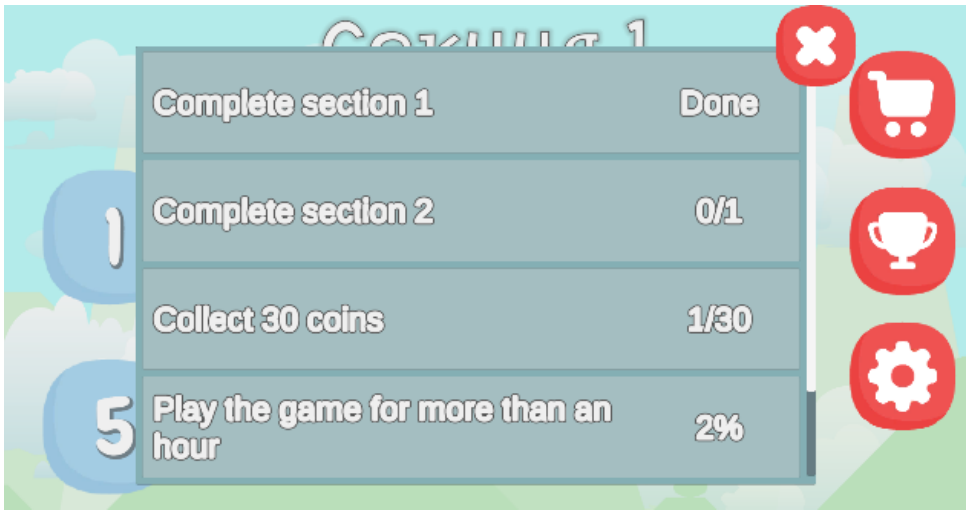

Рисунок 13 – Итоговый вид меню достижений

Меню состоит из следующих элементов:

– окно просмотра достижений. Это объект с прикрепленным компонентом SrollRect, который позволяет прокручивать содержимое окна (контента).

– объект Viewport, определяющий видумую часть содержимого окна. Содержит компонент Mask, который позволяет скрыть часть контента, если он выходит за границы окна Viewport.

– контент. Этот объект содержит список достижений. Имеет компонент Vertical Layout Group, который позволяет разместить достижения в виде вертикального списка, с настраиваемым отступом каждого элемента.

– достижение. Это объект состоит из панели, позволяющей визуально отличать отдельные достижения, и двух текстовых элементов (объектов, содержащих компоненты TextMeshPro и Localize String Event). Первый текстовый элемент описывает условие получения достижения, второй отображает текущий прогресс;

– полоса прокрутки. Объект содержит компоненты Scrollbar(позволяет прокручивать содержимое окна) и Image (отвечает за внешний вид). Настройки компонента Scrollbar выполнены так чтобы полоса прокрутки отображала по-

ложение контента относительно окна просмотра.

– кнопка выхода. При нажатии меню закрывается.

Также был написан скрипт, управляющий поведением окна достижений. Алгоритм его работы представлен на рисунке 14.

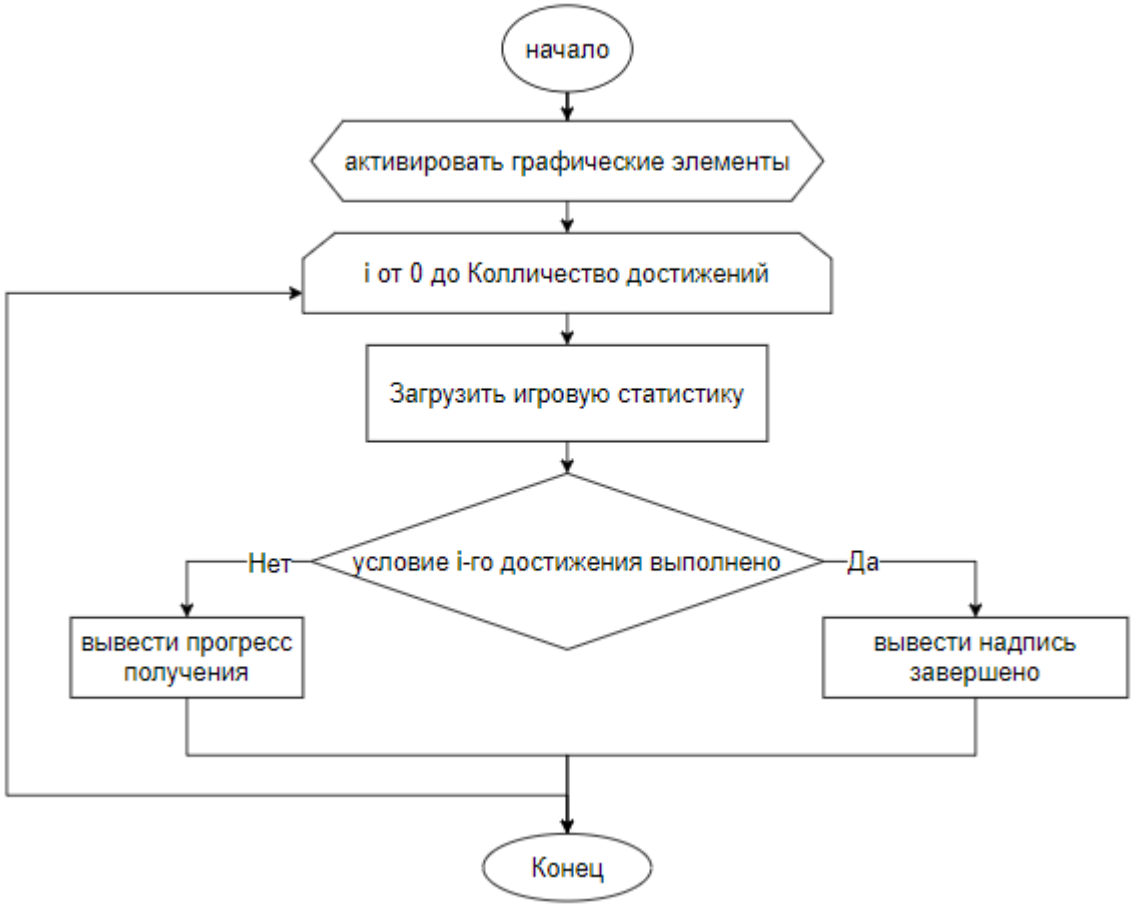

Рисунок 14 – Алгоритм работы скрипта меню достижений

При открытии меню достижений происходит активация вложенных графических объектов. В скрипте для всех достижений хранятся ссылки на текстовые элементы прогресса. Текстовым полям присваивается либо строка выполнено, либо строка, отображающая прогресс, в зависимости от того выполнено ли условие конкретного достижения. Проверка условия осуществляется на основе данных, хранящихся в специальных полях класса сохранений.

3.1.5 Разработка меню настроек

Меню состоит из следующих элементов:

– ползунки регулировки громкости. Выставленные настройки используется

во всей игре. Также рядом с ползунками размещены изображения, информирующие о назначении ползунка;

– кнопка выбора языка. Позволяет пользователю выбрать язык, на котором будут отображаться надписи в игре. Переключение языка происходит при нажатии кнопки;

– кнопка сброса данных. Удаляет все сохраненные внутриигровые данные, кроме настроек языка и звука.

Итоговый вид меню показан на рисунке 16.

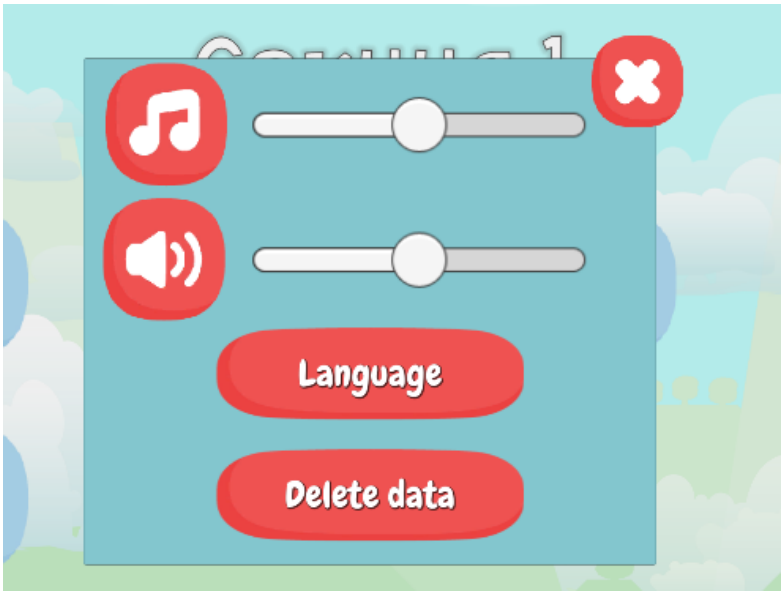

Рисунок 15 – Итоговый вид меню настроек

Для удобного управления звуком был создан объект AudioMixer. Он позволяет микшировать различные источники звука, применять к ним эффекты и выполнять мастеринг [9]. Полезной функцией AudioMixer является возможность объеденять разные источники звука в группы.

Audio Mixer позволяет регулировать для каждой конкретной группы громкость, корректировать высоту тона, создавать звуковые эффекты.

Для того чтобы можно было регулировать воспроизведение звуков отдельно от фоновой музыки, в объекте Audio Mixer были созданы группы Music и Sounds. Каждому источнику звука в сцене была присвоена соответствующая группа.

Также был написан управляющий скрипт для управления воспроизведения звуков. Он функционирует следующим образом:

При инициализации, скрипт устанавливает значения ползунков громкости в соответствии с сохраненными настройками. Далее он ожидает изменения положения ползунков регулирования громкости. Если ползунок меняет значение, то в скрипте вызывается соответствующая функция, куда в качестве аргумента передается новое положение ползунка. В зависимости от этого аргумента, параметру громкости соответствующей группы Audio Mixer присваивается новое значение. Также выполняется сохранение значения громкости, которое будет использоваться в игре. Когда громкость равна нулю, изображение рядом с ползунком меняется, информируя что звук в этой группе отключен. Для этого в скрипте определены массивы, один из которых хранит ссылки на сами изображения, а другой на спрайты, которые требуется подставлять.

Сохранение звука выполняется с помощью методов класса PlayerPrefs, предоставляемого средой разработки. В этом классе хранятся предпочтения игрока между игровыми сессиями [10]. Он может сохранять строковые, плавающие и целочисленные значения в реестре платформы пользователя.

Кнопка выбора языка вызывает функцию, меняющую текущую локаль (используемый в игре язык) игровой сессии. Для этого написан специальный управляющий скрипт, который работает следующим образом:

Сначала скрипт ожидает пока инициализируется система перевода. После этого в глобальных настройках системы перевода указывается последняя сохраненная локаль. При нажатии кнопки происходит смена локали и сохраняется её идентификатор. Сохранение происходит с помощью методов класса PlayerPrefs. В таком случае идентификатор локали хранится отдельно от файла внутриигровых сохранений и не будет удален при сбросе статистики. Язык по умолчанию определяется в настройках системы перевода. В данном проекте им является английский. Именно он будет использоваться, если не удастся найти сохраненный идентификатор.

Кнопка сброса данных при нажатии вызывает специальный метод, опре-

деленный в классе сохранений, который удаляет файл, содержащий сохранения.

3.1.6 Разработка меню выбора уровня

Меню выбора уровней состоит из следующих элементов:

– объект LevelSelect. Он содержит компонент Scroll Rect, позволяющий прокручивать содержимое окна. Также к этому объекту прикреплен компонент EventTrigger вызывающий специальное событие, определенное в управляющем скрипте меню, когда пользователь перестает прокручивать содержимое окна;

– объект Pages является дочерним по отношению к LevelSelect. Он содержит скрипт, отвечающий за плавное прокручивание страницы.

Скрипт необходим так как окно выбора уровня содержит две секции. У компонента Scroll Rect нет функционала, позволяющего фокусироваться на отдельном участке прокручиваемого контента. Когда пользователь заканчивает прокручивание, скрипт определяет ближайшую секцию и перемещает контент так, чтобы участок контента, на котором размещена секция, был полностью виден. Для этого в скрипте используется массив, хранящий точки привязки каждого участка контента к окну просмотра. Это позволяет добавлять новые секции без необходимости написания нового кода. На текущий момент реализовано две секции.

– Объекты Page. Каждый объект представляет собой отдельную секцию, содержащую набор кнопок для выбора уровней. Page содержит компонент Grid Layout Group, который служит для автоматического позиционирования кнопок.

– Кнопки выбора уровня. Каждая кнопка вызывает специальную функцию в скрипте, отвечающем за загрузку сцен и в качестве аргумента передает номер уровня.

3.2 Настройка интерфейса (UI) под разные разрешения экрана

При разработке интерфейсов необходимо было учитывать проблему разных разрешений экранов мобильных устройств. Система создания интерфейсов в Unity снабжена рядом различных инструментов для реализации этих возможностей, которые также можно комбинировать между собой множеством раз-

личных способов.

Для позиционирования элементов интерфейсов использовались следующие способы:

– Настройка якорей элементов для адаптации к разным соотношениям сторон:

В Unity элементы интерфейса позиционируются относительно друг друга. По умолчанию каждый создаваемый элемент привязан к центру прямоугольника родительского элемента. Это означает что при изменении положения родительского элемента дочерние перемещаются вместе с ним, сохраняя постоянное смещение относительно центра.

Если с данной настройкой разрешение было изменено под другое соотношение сторон, элементы могут выпасть из своих прямоугольных областей, где они изначально должны были быть расположены.

Одним из способов сохранить расположение элементов в области экрана является изменение компоновки таким образом, чтобы места их расположения были связаны с их соответствующими углами на экране либо другими подходящими точками. Привязка элемента может быть установлена при использовании в инспекторе выпадающего списка Anchors Preset (наборы привязок), или путём перетаскивания треугольных ручек привязок (якорей) в видовом окне сцены (Scene View). Компановку элементов следует выполнять пока текущее разрешение экрана, установленное в игровом режиме (Game View) является тем разрешением, для которого изначально был разработан дизайн интерфейса и где места расположения элементов были специально подобраны [11].

Если элементы будут привязаны к соответствующим точкам привязки, то при дальнейших изменениях разрешения экрана и соотношений сторон они будут сохранять свои позиции относительно этих точек.

Компоновку можно выполнять не только относительно отдельных точек, но и определенных участков экрана. В таком случае при изменении разрешения экрана элемент будет сохранять смещение относительно всего указанного участка.

– Масштабирование с компонентом Screen Size:

После настройки якорей, когда разрешение экрана изменяется на большее и меньшее, элементы должны по прежнему сохранять своё изначальное расположение относительно точек, к которым они привязаны. Однако, сохраняя своё оригинальное разрешение, заданное в пикселях, они могут становиться как больше так и меньше, соответствуя пропорциям текущего разрешения экрана, что может быть не желательным эффектом.

Также на устройстве с одинаковыми размерами экрана, может быть абсолютно разное разрешение, которое зависит от плотности пикселей на 1 дюйм самого экрана. Но при этом на экранах с меньшей плотностью пикселей кнопки не должны отображаться крупнее чем на экранах устройств с большей плотностью – они должны быть точно такого же размера. Это означает что кнопки должны становиться меньше настолько же, насколько в процентном соотношении становиться меньше сам экран. Другими словами, масштаб кнопок должен быть привязан к масштабу размеров экрана.

Для того чтобы избежать указанных проблем используется компонент Canvas Scaler. Он предназначен для управления общим масштабом и плотностью пикселей элементов пользовательского интерфейса. Данный компонент выполняет масштабирование всего пользовательского интерфейса, отображаемого с помощью компонента Canvas, включая размеры шрифтов и границы изображений.

Компонент Canvas Scaler может быть добавлен в корень Canvas, все интерфейсные элементы которого являются его потомками. Он также создаётся по умолчанию во время создания нового компонента Canvas через меню GameObject.

В компоненте Canvas Scaler можно установить режим масштабирования UI Scale Mode в Scale With Screen Size. В данном режиме масштабирования можно определить какое разрешение использовать в качестве базового. Если текущее разрешение больше или меньше базового, фактор масштабирования компонента Canvas устанавливается соответственно так, чтобы

все элементы интерфейса масштабировались в большую или меньшую сторону вместе с разрешением экрана. В качестве базового было выбрано разрешение 1480x720.

3.3 Разработка системы сохранений

Скрипт, реализующий систему сохранений содержит класс SaveData, предоставляющий методы для сохранения данных, и сериализуемую структуру JsonController, которая служит оболочкой для записи данных.

Класс SaveData является статическим и доступен из любой сцены в игре. Соответственно все поля и методы в классе также являются статическими. Это позволяет любому скрипту в игре обращаться к элементам SaveData по имени класса [12]. В таком случае в вызывающих скриптах не требуется хранить ссылки на класс сохранений.

На данный момент в классе определены поля для хранения следующих значений:

- Количества пройденных уровней;
- Количество имеющихся монет (внутриигровая валюта);
- Количество монет, собранных за всю игру;
- Время проведенное в игре (в секундах). Хранится в переменной типа float. Данный тип выбран поскольку он имеет достаточный диапазон значений для представления времени, которое пользователь может провести в игре. При этом значение в основном используется в выражениях, оперирующих типом float, поэтому преобразование из других типов было бы не очень эффективным [13];
- Идентификатор выбранного скина;
- Массив, хранящий сведения о купленных скинах.

Также класс содержит публичный метод для сохранения значений данных полей в файл. Он работает следующим образом:

Создается экземпляр структуры, в которую заносятся значения всех полей класса. Далее с помощью методов класса JsonUtility, предоставляемого средой разработки, структура преобразуется в строку формата JSON. Создание эк-

земпляра структуры было необходимо, поскольку JsonUtility не поддерживает сериализацию статических полей [14]. Далее полученная строка записывается в файл по месту размещения проекта с помощью класса BinaryWriter. Этот класс поддерживает запись только примитивных типов, поэтому требуется промежуточное преобразование в JSON. Ссылочные типы могли бы быть сериализованы с помощью класса BinaryFormatter, однако на сегодняшний день он объявлен устаревшим и не рекомендован к использованию.

Метод загрузки данных работает по обратному принципу. Сначала проверяется наличие файла сохранений. Если он отсутствует, то функция завершает работу. Поля класса SaveData будут иметь значения, присвоенные им по умолчанию. Если файл найден то сначала создается экземпляр структуры. Далее выполняется чтение строки, записанной в файл с помощью класса BinaryReader. Затем с помощью метода JsonUtility из строки извлекаются сохраненные значения и присваиваются созданному экземпляру структуры. В конце значения структуру присваиваются соответствующим полям класса SaveData.

Также определен метод, удаляющий файлы сохранения. В таком случае прогресс пользователя в игре будет сброшен. Метод удаляет файл сохранений и присваивает полям класса значения по умолчанию.

3.4 Разработка системы загрузки уровней.

Для того чтобы сосредоточить логику перехода между сценами в одном месте, был разработан специальный класс Manager [15].

Также для осуществления плавного перехода между сценами было создано окно перехода, способное выводить прогресс загрузки. Оно имеет две анимации одна из которых проигрывается при закрытии, а другая при открытии сцены. Также окно перехода имеет текстовый элемент, которому присваивается значение прогресса загрузки.

Класс Manager содержит несколько статических методов:

– загрузка произвольной сцены. Метод принимает в качестве аргумента номер сцены, которую требуется загрузить. При вызове определяется, происходит ли загрузка другой сцены. Для этого проверяется значение специальной пе-

ременной типа bool, определенной в классе Manager. Она принимает значение true при вызове метода загрузки сцены. Далее сохраняется время, проведенное в игре и вызывает окно перехода между сценами;

– загрузка следующей сцены. Этот метод обычно вызывается для перехода между уровнями игры. Он определяет номер текущей сцены, прибавляет к нему единицу и вызывает метод загрузки произвольной сцены;

– перезагрузка текущей сцены. Вызывает метод загрузки произвольной сцены и передает в качестве аргумента номер текущей сцены.

Окно переходов содержит скрипт, отвечающий за непосредственную загрузку сцен. Сцены загружаются асинхронно. Новая сцена не может быть открыта до того как, как закончится анимация закрытия предыдущей. Если анимация закрытия закончилась, а сцена еще не успела загрузиться, то прогресс (в процентах) отображается в текстовом поле окна переходов.

3.5 Реализация перевода игры

Для перевода игры на другие языки используется пакет Localization. Данный пакет предоставляет инструменты для добавления поддержки нескольких языков в приложение (например, поддержку текста на нескольких языках или ассетов, зависящих от языка и региональных параметров, таких как аудио или текстуры) [16].

Инсталляция пакета производилась с помощью инструмента PackageManager.

Перед использованием инструментов пакета необходимо было создать ассет настроек локализации. Генерация ассета и его активация происходит автоматически, по нажатию кнопки в настройках локализации проекта. Далее с помощью инструмента Locale Generator были созданы настройки локалей (языков), которые будет поддерживать игра (на текущий момент это русский и английский). Далее в настройках локализации проекта указан язык, используемый по умолчанию (в данном проекте это английский).

Затем необходимо было создать таблицы строк, содержащие переводы текста на все поддерживаемые языки. Предполагается, что в каждой таблице

содержатся переводы строк, относящихся к какой-то определенной части проекта. На текущий момент в проекте текстовые надписи содержатся только в элементах пользовательского интерфейса, поэтому была создана одна таблица строк с названием UI.

Далее в таблицу были добавлены переводы текстовых фраз, встречающихся в игре. Для каждой фразы был указан специальный идентификатор, чтобы компоненты, отвечающий за перевод, могли определить какую запись таблицы использовать.

Далее ко всем объектам содержащим текстовые элементы был добавлен компонент перевода, указаны необходимые настройки (идентификаторы таблицы, в которой следует искать перевод, и конкретной фразы).

В результате все текстовые элементы, содержащие компонент перевода, будут отображать фразу на языке соответствующем выбранной локали (выбор осуществляется в главном меню).

3.6 Подготовка текстовых ассетов

Unity имеет стандартную поддержку файлов шрифтов .ttf и .otf , но компонент TextMeshPro не работает с файлами этих форматов напрямую, поэтому их нужно преобразовать в элемент Font Asset. Это было сделано с помощью инструмента Font Asset Creator, который создает ресурс TextMesh Pro из файла шрифта.

Для этого в окне сначала в окне Font Asset Creator был указан исходный файл шрифта. Далее указаны необходимые настройки (рисунок 16).

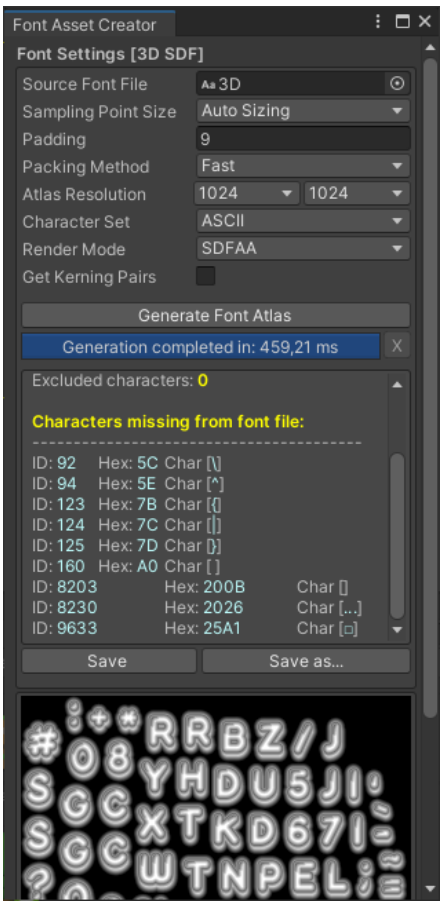

Рисунок 16 – Окно инструмента Font Asset Creator с принятыми настройками

Настройка Sampling Point Size определяет размер для отдельного символа в файле шрифта, что влияет на точность выборки при распознавании символов. Для того чтобы ввести размеры вручную можно выбрать параметр CustomSize. При значении AutoSizing, принятому по умолчанию, выполняется попытка заполнить атлас (изображение, содержащее набор под-изображений, каждое из которых является текстурой для некоторого 2D или 3D объекта) всеми символами шрифта. При автоматическом определении размеров распознавание было выполнено достаточно точно, поэтому ручной ввод размеров не потребовался.

Настройка Padding тесно связана с параметром Sampling Point Size. Она определяет ряд визуальных эффектов текста: интервал между символами в тексте, смещение тени или сияния символов, толщину контура символов и т. д. Обычно рекомендуется выбирать значение Padding, лежащее в пределах 7–10 % от Sampling Point Size. Так как Sampling Point Size =89 было выбрано значение Padding = 9.

Настройка Packing Method определяет метод заполнения атласа. При параметре Fast генерация происходит быстрее, что хорошо для быстрого предварительного просмотра полученного текстового ассета. Был выбран параметр Optimum поскольку он больше подходит для генерации окончательного варианта текстового ассета.

Надстройка Atlas Resolution определяет размер текстуры атласа. Это определяется размером и заполнением точки выборки. Для проектов ориентированных на мобильные устройства рекомендуется ограничивать разрешение атласа размером 2048x2048, поэтому было выбрано значение 1024x1024.

Настройка Character Set определяет диапазон символов, используемых в шрифте. Был выбран диапозон Extended ASCII, однако не все символы из этого диапазона определены в используемом шрифте, поэтому часть символов распознана не будет.

Настройка RenderMode определяет режим отрисовки шрифта. Был выбран режим SDFAA поскольку в этом режиме шрифт визуально выглядел лучше.

Настройка Get Kerning Pairs определяет, будет ли использоваться встроенная в шрифт информация о кернинге (избирательное изменение интервала между буквами в зависимости от их формы) если таковой имеется.

После указания всех настроек была выполнена генерация текстового ассета, который может использоваться в компоненте TextMeshPro.

Также была импортирован кирилиический шрифт с похожим начертанием, который будет использоваться в качестве дополнительного. Это требуется, чтобы при переключении на другой язык надписи в игре корректно отображали все символы и при этом не меняли визуальный стиль.

Если создавать отдельный Font Asset для каждого языка и копировать в него все символы соответствующего этому языку шрифта, то увеличится объем сборки и снизится быстродействие игры, поэтому процесс требует оптимизации. Для этого можно объединить языки в элементы типа Font Asset по системе письменности (латиница, кириллица и т. п.) и в дальнейшем комбинировать эти

группы.

Алгоритм, по которому игра ищет нужный символ в наборах (например, при пользовательском вводе), показан на рисунке 17.

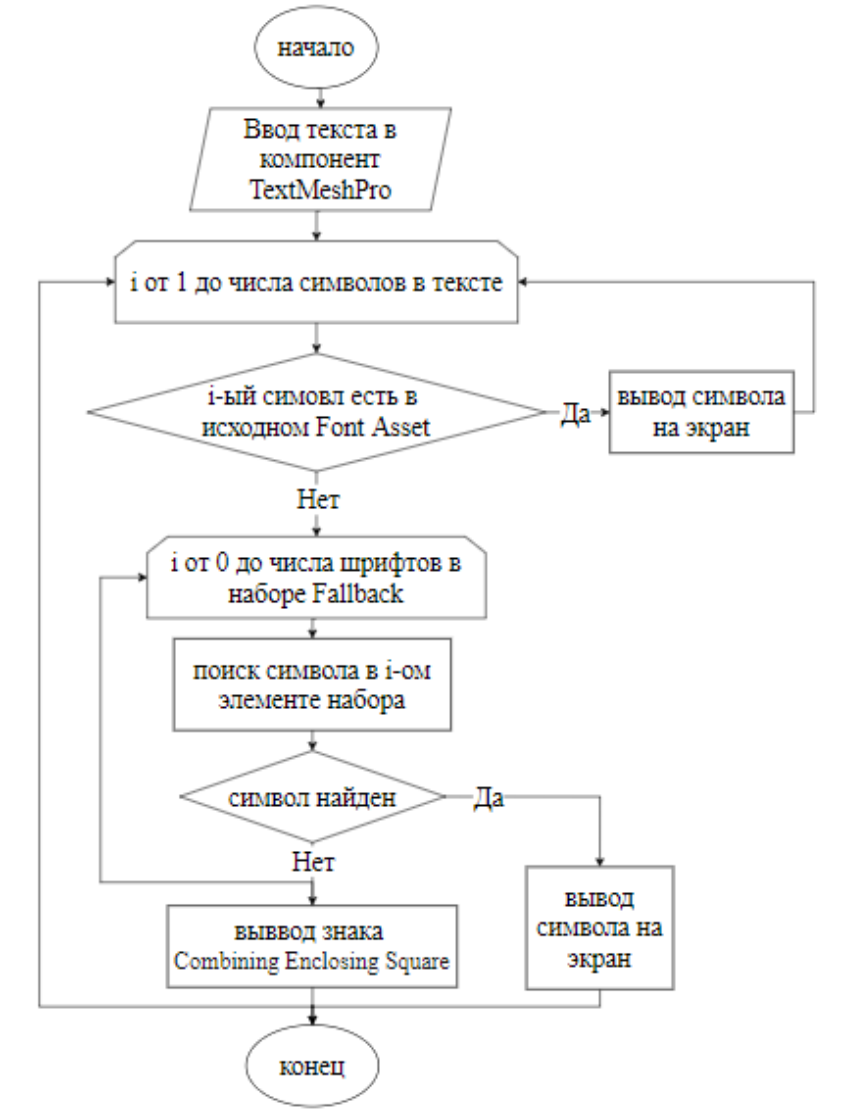

Рисунок 17 – Алгоритм поиска игрой нужного символа

Исходный латинский Font Asset включает символы расширенной таблицы ASCII. Также был создан еще один Font Asset с кириллицей. Процесс создания аналогичен созданию латинского элемента Font Asset, однако дополнительно потребовалось указать кириллические символы, отсутствующие в таблице ASCII, чтобы инструмент Font Asset Creator смог их распознать. Для этого в настройке Character Set указан параметр Custom Characters, что позволило вручную указать набор символов, содержащихся в шрифте.

Далее кириллический Font Asset был подключен к латинскому как

Fallback (т. е. дополнительный набор, к которому система вывода будет обращается при отсутствии нужного символа в основном наборе). Можно подключить неограниченное количество элементов Fallback, что может быть полезно при переводе проекта на другие языки.

3.7 Разработка системы инициализации уровня

Каждый уровень содержит управляющий скрипт. Он отвечает за действия, выполняемые во время загрузки уровня и выхода из него. Скрипт прикреплен к объекту в сцене, поэтому он будет выполняться при каждом запуске уровня заново.

При загрузке уровня выполняется начальное позиционирование персонажа. Для этого служит система чекпоинтов (сохранений внутри уровня). Внутри скрипта уровня определен статический класс LastCheckpointData, содержащий данные о последнем активированном чекпоинте (ссылку, название объекта, позицию чекпоинта) и метод, позволяющий добавить новый чекпоинт. Ссылка нужна, поскольку при активации чекпоинта, предыдущий активированный чекпоинт должен менять спрайт. Доступ по ссылке осуществляется быстрее чем поиск по имени объекта. Но после перезагрузки сцены ссылка будет указывать на несуществующий объект, поэтому также хранится имя чекпоинта, по которому его можно найти в иерархии элементов.

При загрузке уровня управляющий скрипт ищет в сцене сохраненный чекпоинт по названию объекта и активирует его. Также координаты персонажа выставляются равными координатам чекпоинта. Используемый скин персонажа определяется в соответствии с сохраненным значением (выбор скина осуществляется в главном меню).

Скрипт уровня содержит метод Death, вызываемый при «смерти» персонажа. Он обнуляет ссылку на последний чекпоинт и вызывает метод перезапуска уровня, определенный в менеджере загрузки.

Также в скрипте уровня определен метод, вызываемый при загрузке новой сцены. Он выставляет значение параметра TimeScale, от которого зависит переменная времени, равным 1 (соответствует нормальном течению времени),

на случай если этот параметр был изменен. После этого начинается загрузка другой сцены.

3.8 Создание управляемого персонажа

На первом этапе в программе Adobe Illustrator CC 2018 был создан скин персонажа. В качестве прототипа было взято векторное изображение из бесплатного для распространения ассета «Free Platform Game Assets» от автора Bayat Games (рисунок 18).

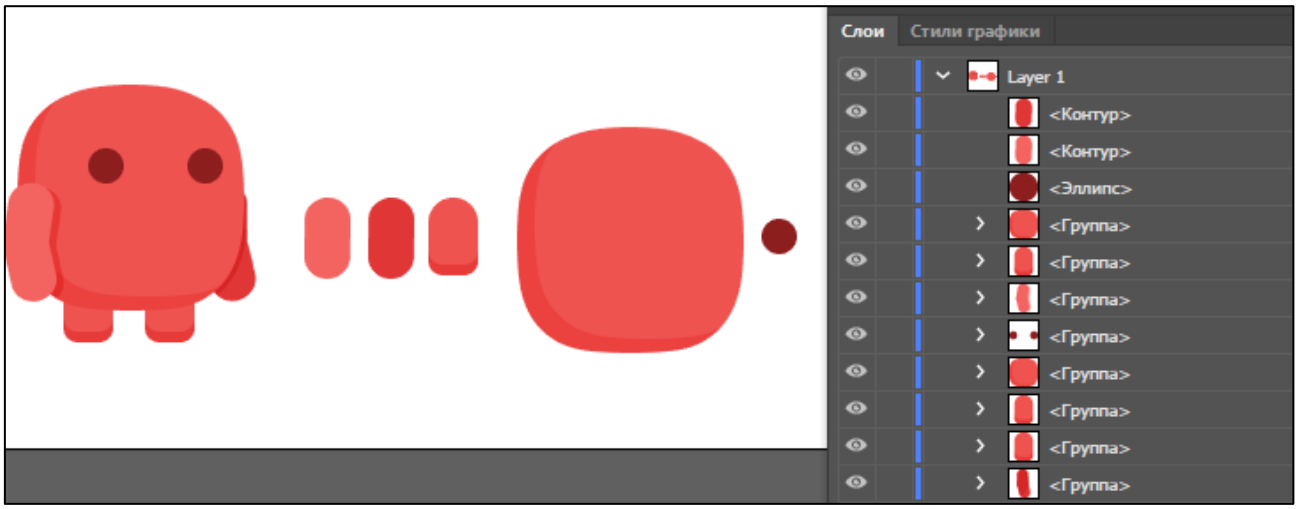

Рисунок 18 – Исходное изображение персонажа

Исходное изображение представляло собой файл с расширением ai, содержащий прототип персонажа и изображения отдельных частей тела вынесенных на разные слои. Для того чтобы можно было загрузить и правильно настроить изображение в IDE «Unity», потребовалось перераспределить изображения по слоям так, чтобы на каждом отдельном слое, либо в каждой вкладке «Группа» находилась отдельная часть тела и распределить слои в порядке их отрисовки в будущем. Это требуется для того чтобы позже PSD importer – пакет используемый в «Unity» для импорта изображений из формата PSB, смог правильно интерпретировать изображение, что будет необходимо при создании скелетной анимации.

В исходном прототипе было несколько изменено положение частей тела, так чтобы в процессе создания анимации было легче имитировать 3D перспективу. Помимо этого было создано три дополнительных скина глаз, которые мо-

гут использоваться для кастомизации (выбора игроком внешности персонажа) или как художественный элемент игры (например, для выражения эмоций персонажа).

Также по исходному шаблона персонажа с соблюдением пропорций был создан дополнительный скин персонажа такого же размера, который также может использоваться в игре для кастомизации.

В «Unity» создание скелетной 2D анимации возможно только для файлов формата PSB, но Adobe Illustrator не поддерживает прямой экспорт в этот формат, поэтому сначала изображения были сохранены в формат PSD. Далее файлы был открыт в графическом редакторе Adobe Photoshop CC 2020. На этом этапе необходимо было убедиться, что все части тела разнесены по слоям правильно, так как элементы, используемые в Adobe Illustrator для построения изображений (группы контуров, дуг, примитивных фигур и другие элементы) могут быть расценены как отдельная часть тела, из за чего изображение в «Unity» будет интерпретировано неправильно, что затруднит создание скелетной анимации.

Далее изображения были сохранены в формате PSB. Подготовленные для импорта изображения персонажа представлены на рисунке 19.

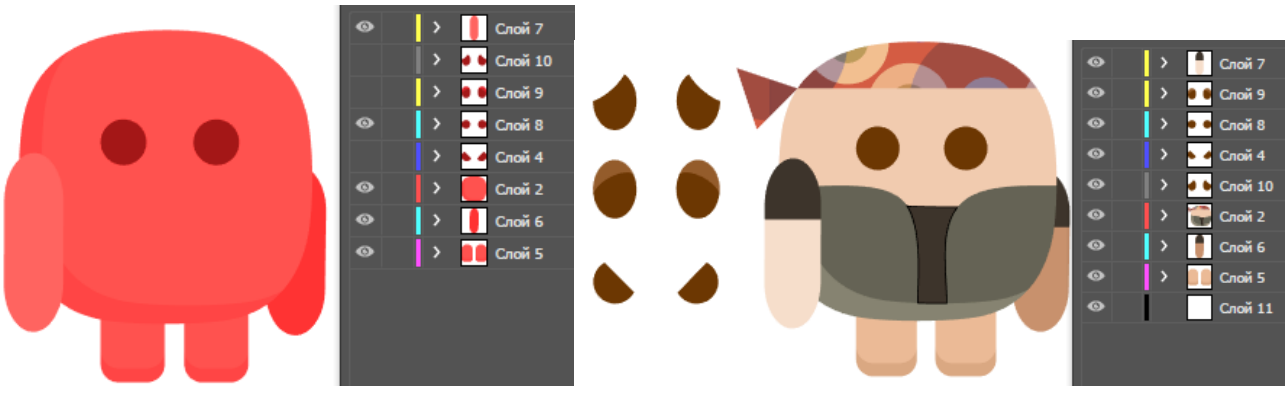

Рисунок 19 – Подготовленные для импорта изображения персонажа

После этого изображения были импортированы в «Unity». Для интерпретации файлов PSB потребовалось дополнительно установить пакет (модуль, расширяющий функции среды разработки) «PSDimporter».

Также потребовалась установка пакета «2D Animation Package», который позволяет создавать скелетную анимацию для спрайтов, а именно: поддерживает создание костей для спрайтов, автоматическую генерацию вертексов и весов костей, их ручное редактирование, содержит инструменты для создания инверсной кинематики [17].

После импорта PSB файла графический элементы из каждого слоя автоматически преобразуются в отдельный спрайт. При этом название каждого спрайта соответствует названию слоя, с которого он был импортирован.

Создание скелета происходит в окне редактора спрайтов, который используется для извлечения графических элементов из одного файла (рисунок 20). После импорта пакета для 2D анимации в редакторе становится доступен инструмент «Skinning Editor».

Сначала необходимо создать корневую кость, которая служит опорной для всего скелета. Для этого использовался инструмент «Create Bone».

Далее необходимо разместить остальные кости, учитывая их будущее влияние на персонажа. Кости ног, рук и глаз не прикреплены непосредственно к цепочке костей, идущей из корневой кости, чтобы позволит анимировать их независимо от тела. Тело состоит из двух костей, что позволит менять положение и наклон верхней части тела, учитывая положение нижней.

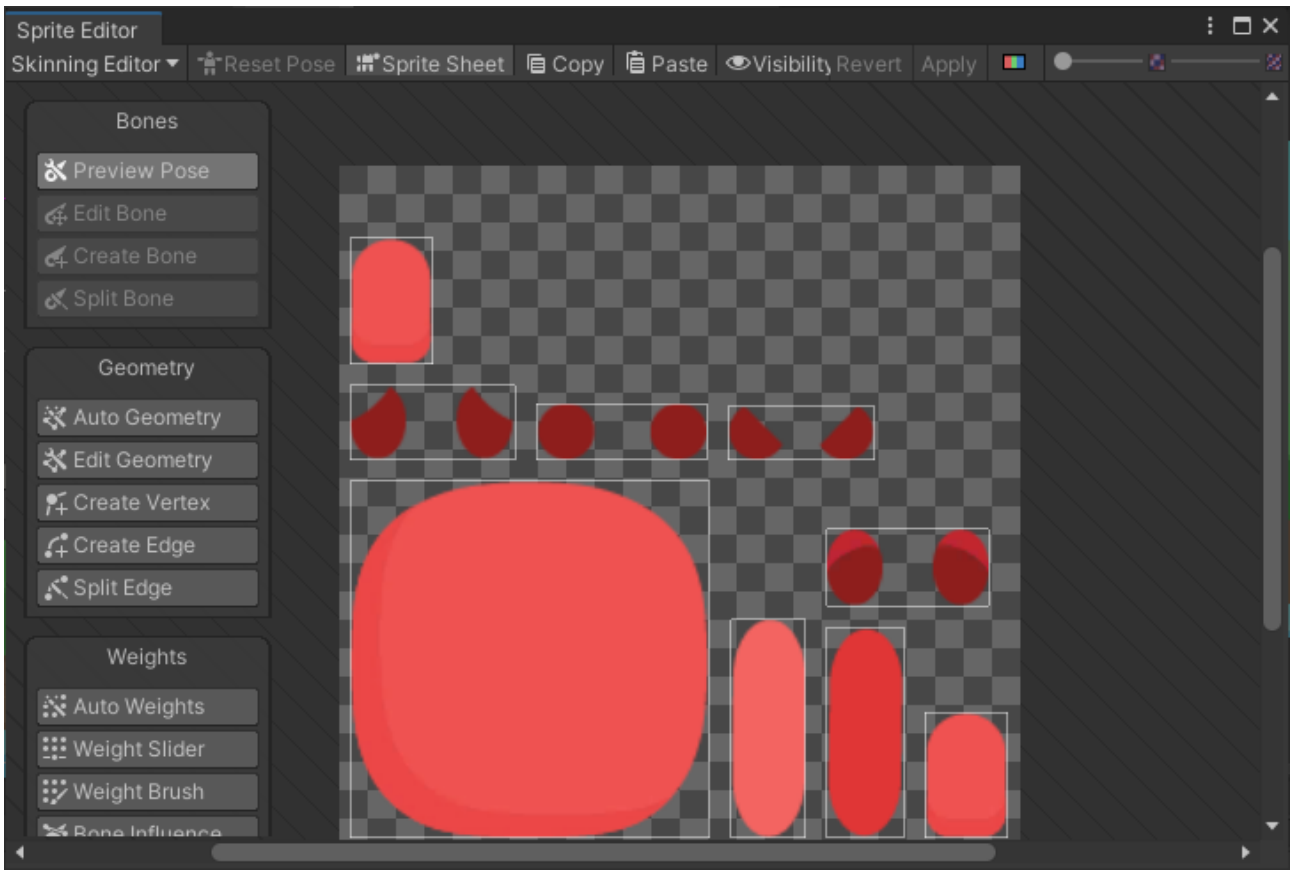

Рисунок 20 – Изображение персонажа в редакторе спрайтов

После создания костей необходимо сгенерировать меш (сетку, определяющую деформацию изображения) для каждого спрайта. Это можно сделать автоматически, используя инструмент «Auto Geometry». Параметры автоматической генерации можно указать в окне «Geometry». Автоматическая генерация меша может быть не совсем точной, поэтому необходимо проверить сгенерированный меш каждого спрайта. Неправильные границы каждого меша были отредактированы. Редактировать границы мешей можно следующим образом:

Если граница меша проходит неправильно, можно отредактировать (переместить, удалить) вершину границы с помощью инструмента «Edit Geometry» или добавить вершину с помощью инструмента «Create Vertex», либо целую границу с использованием «Create Edge». Инструмент «Split Edge» также позволяет редактировать границы меша, разбивая их на 2 части.

Процесс генерации мешей автоматически соотносит кости со спрайтами, на основе их расположения и генерирует веса влияния костей. Однако кость

может перекрывается со спрайтом, к которому не относится и в таком случае может быть ошибочно к нему отнесена. Для того чтобы вручную указать соответствие между костями и спрайтами использовался инструмент «Bone Influence».

Далее была выполнена проверка полученного скелета, на возможность создания требуемых анимаций. Для этого с помощью инструмента «Preview Pose» путем расстановки и поворота костей персонаж ставился в позы которые будут использоваться в анимации. На этом этапе были уточнены значения весов влияний каждой кости на спрайт. Это делалось с помощью инструментов «Weight Slider» для настройки влияния кости на спрайт целиком и «Weight Brush» для локальной точечной настройки весов в определенных местах спрайта.

На последнем этапе создания скелета для каждой кости была указана «глубина», определяющая порядок её отрисовки. Результат представлен на рисунке 21.

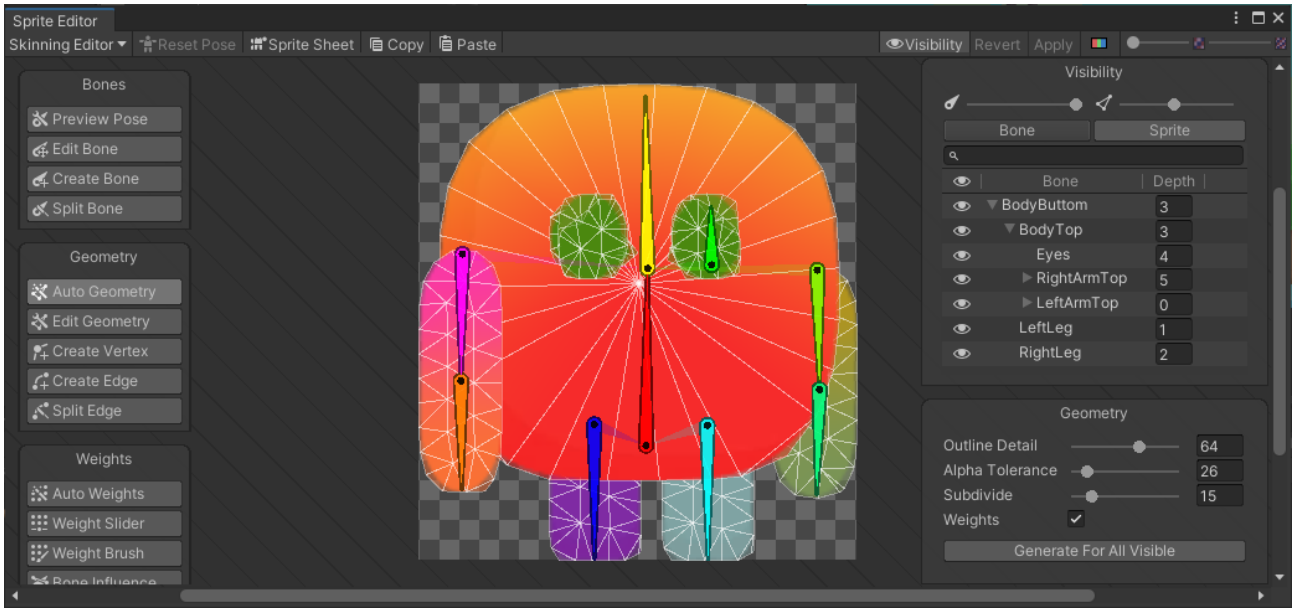

Рисунок 21 – Полученный скелет персонажа

После этого полученный скелет был скопирован во второй скин. Так как оба скина используют один и тот же скелет, для них становится возможным использование одних и тех же анимаций.

В результате для изображения персонажа были автоматически сгенерированы настройки импорта, включающие исходные изображения частей тела и скелет персонажа. Теперь при переносе спрайта в сцену поверх него отображается скелет. Меняя положение костей скелета можно анимировать персонажа.

Для удобства анимирования рук персонажа была настроена инверсная кинематика. Пакет «2D Inverse Kinematics» (IK) который входит в состав «2D Animation» позволяет применять инверсную кинематику к костям при преобразованиям скелета персонажа. Пакет 2D IK автоматически рассчитывает положения и повороты цепочки костей, движущихся к заданной позиции. Это упрощает процесс создания анимации для конечностей персонажа, так как вручную создавать ключевые кадры для цепочки костей не требуется.

Для этого к объекту игрока в сцене был добавлен компонент IK Manager предоставляемый IK пакетом. Далее к объекту также были добавлены 2 компонента «IK Solver» с используемым алгоритмом «Limb», для левой и правой руки. IK Solver вычисляет положение и поворот, которые должен принять эффектор (якорь, смещение которого влияет на остальные кости в спрайте) и связанные с ним кости, чтобы достичь своей целевой позиции.

Решатель «Limb» это стандартный решатель для двух костей, который хорошо подходит для позиционирования суставов, таких как руки и ноги [18]. Цепочка каждого решателя состоит из эффекторной кости, и кости нижней части руки.

Далее были созданы следующие анимации персонажа:

Бег (рисунок 23), прыжок, спокойное состояние (рисунок 22), состояние бездействия, прыжок, падение. Для создания анимаций использовалась встроенная в «Unity» программа «Animation», возможности которой включают перенастраиваемую анимацию, полный контроль над весами анимации во время выполнения, вызов событий из воспроизведения, иерархии и переходы анимаций, смешивание анимаций и т.д.

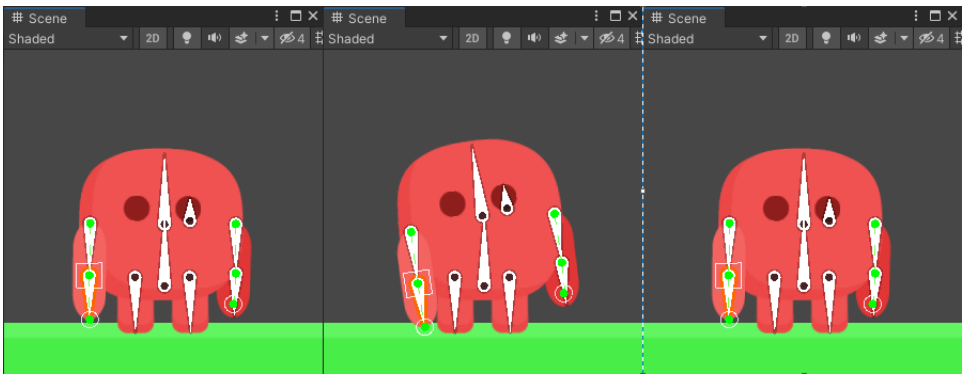

Рисунок 22 – Ключевые кадры анимации спокойного состояния

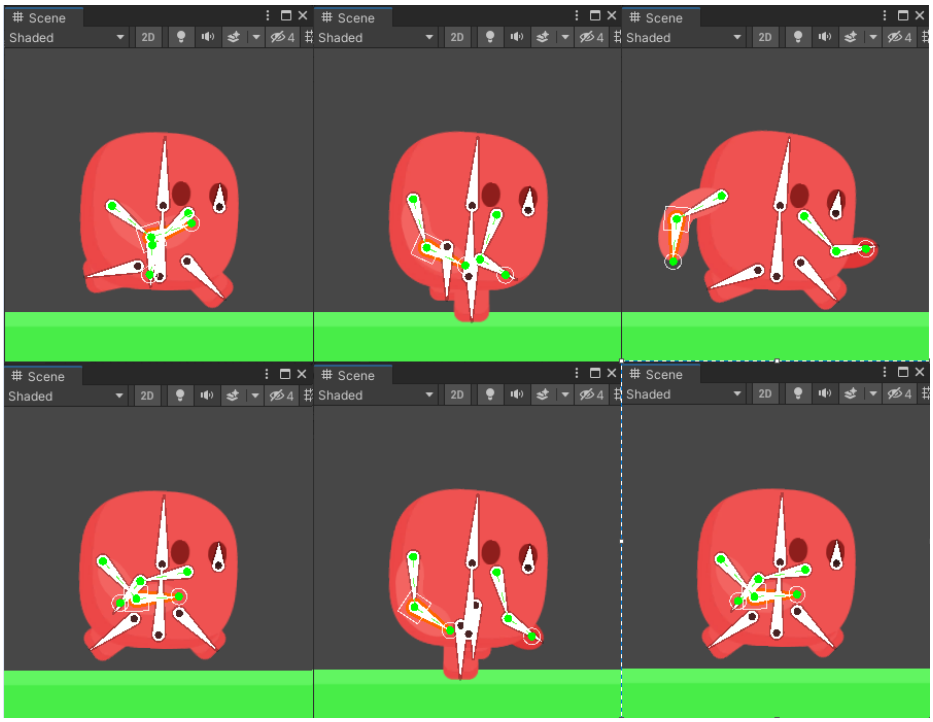

Рисунок 23 – Ключевые кадры анимации бега

Далее было разработано управление, адаптированное под мобильные устройства. В качестве элемента управления используется бесплатный ассет джойстика от разработчика Fenerax Studios. Также написан скрипт, отвечающий за управление персонажем. В нем определены функции, ставящие движения персонажа в соответствии с положением якоря джойстика.

Движение камеры за игроком реализовано путем настройки модуля Cinemachine. Он позволяет задавать поведение камер без написания кода [19]. Модуль настроен так, чтобы игрок мог перемещаться в рамках заданной обла-

сти без движения камеры. При этом чем дальше игрок удаляется от камеры, тем быстрее она начинает двигаться и к границе заданной области имеет такую же скорость как и у игрока.

Для персонажа был создан автомат переходов состояний (рисунок 24), связанный с контроллером управления, и настроены параметры смешивания анимаций, для их плавного переключения.

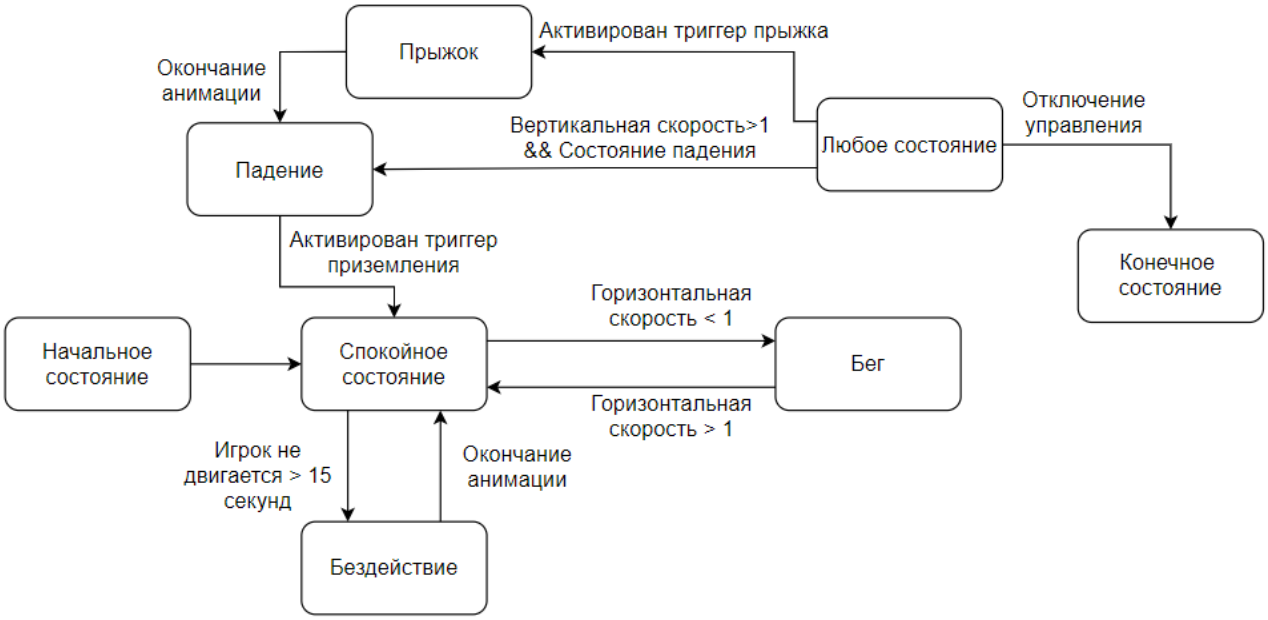

Рисунок 24 – Автомат состояний персонажа

Также с помощью компонентов Hinge Joint 2D и RigidBody 2D реализованы шарнирные соединения в местах крепления конечностей персонажа к телу (рисунок 25). Это позволяет ограничить их вращение, при обработке движения физическим движком и обеспечит их реалистичное поведение [20].

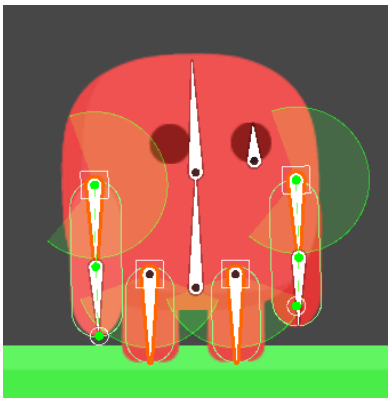

Рисунок 25 – Hinge Joint 2D в местах крепления конечностей персонажа

3.9 Разработка интерактивных объектов

На данный момент реализованы следующие интерактивные элементы:

– Пружина. Содержит компонент коллайдера с типом триггер. При коллизии с игроком вызывает метод Push, определенный в классе игрока. Он толкает игрока в направлении совпадающим с осью Y объекта пружины с силой определенной для каждого экземпляра пружины.

– Мина. Функционирует так же как и пружина, но при этом вызывается метод Regdoll, определенный в классе игрока. Он отключает присоединённый к объекту игрока компонент аниматор (поскольку при анимации положение спрайтов контролируется аниматором), и передаёт управление объектом игрока физическом движку.

– Смертельная зона. При коллизии с объектом игрока вызывает функцию Death в скрипте уровня, Она вызывает метод Regdoll класса игрока и метод Death, определенный в скрипте уровня.

– чекпоинт. При коллизии с объектом игрока меняет отображаемый спрайт и вызывает в скрипте уровня функцию, отвечающую за смену чекпоинта;

– монета. При коллизии с игроком объект монеты уничтожается, а сохраненное значение монет увеличивается на один;

– на данный момент реализован один вид противников. Объект противника содержит управляющий скрипт, два коллайдера. Первый коллайдер размещается над спрайтом врага и имеет тип триггер. При столкновеннии с ним объекта игрока в скрипте врага вызывается функция получения урона. При этом объект врага уничтожается, а в сцене на его месте создается источник звука. Это нужно поскольку при уничтожении объекта врага присоединённый компонент источника звука так же был бы уничтожен. Другой коллайдер соответствует размерам спрайта врага. При столкновении с ним вызывается функция получения урона игроком.

К каждому интерактивному элементу прикреплен трехмерный (громкость зависит от расстояния до игрока) источник звука. Также каждый источник добавлен в группу Sounds объекта AudioMixer (создан на этапе разработки глав-

ного меню), что позволит регулировать их громкость с помощью ползунков громкости. Для каждого интерактивного объекта был создан соответствующий префаб. Это особый тип ассетов, позволяющий хранить весь игровой объект со всеми компонентами и значениями свойств [21].

3.10 Пользовательский интерфейс уровня

Пользовательский интерфейс состоит из:

– Меню паузы. При активации время в игре останавливается. Это достигается путем изменения значения параметра TimeScale на ноль, что соответствует полной остановке времени. В меню находятся кнопки перехода на главную сцену и загрузки уровня. При нажатии вызывают методы, определенные в классе переходов между сценами. Также есть ползунки регулирования громкости, которые работают так же как и в главной сцене;

– Интерфейс уровня. Содержит счетчик монеток. Текстовый элемент счетчика выводит число монеток, сохраненное в классе сохранений и обновляется при подборе новой монеты. Также содержит джойстик, позволяющий управлять персонажем.

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

4.1 Безопасность

Исходным элементом любого предприятия является рабочее место, в пределах которого происходит целенаправленная деятельность конкретного работника. Рабочее место — это зона нахождения работника и средств приложения его труда, которая определяется на основе технических и эргономических нормативов и оснащается техническими и прочими средствами, необходимыми для исполнения работником поставленной перед ним конкретной задачи.

Можно выделить следующие требования к помещениям для работы с ПЭВМ:

Окраска помещений и мебели должна быть выполнена в нейтральных цветах, чтобы не вызывать зрительного утомления.

Рабочие столы следует размещать таким образом, чтобы мониторы были направлены боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева. Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться равномерно. В случаях непосредственной работы с документами, следует использовать системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк [22]. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Необходимо контролировать прямую блесткость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м2. Необходимо контролировать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусствен-

ного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м2 и яркость потолка не должна превышать 200 кд/м2. Светильники должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

Кроме того, все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации. Не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ. Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки провода электропитания, в отсутствии видимых повреждений и наличии заземления приэкранного фильтра.

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должны размещаться вне помещений с ПЭВМ. Уровень шума на рабочем месте не должен превышать значений, указанных в таблице 1 [23].

Таблица 1 – Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровень звука и эквивалентный уровень звука, дБА
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

Для снижения шума создаваемого на рабочих местах внутренними источниками, а также шума, проникающего из вне следует:

– ослабить шум самих источников (применение экранов, звукоизолирующих кожухов);

– снизить эффект суммарного воздействия отраженных звуковых волн (звукопоглощающие поверхности конструкций);

– применять рациональное расположение оборудования;

– использовать архитектурно-планировочные и технологические решения изоляций источников шума.

При выполнении работ с использованием ПЭВМ в производственных помещениях уровень вибрации не должен превышать допустимых значений вибрации для рабочих мест (категория 3, тип «в») в соответствии с действующими санитарно-эпидемиологическими нормативами. Показатель вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы.

Также следует учитывать требования к микроклимату помещения:

– На рабочем месте оператора должны обеспечиваться оптимальные микроклиматические условия в холодный и теплый периоды года.

– Температура воздуха на рабочем месте в холодный период года должна быть от 22 до 24 °С, в теплый период года - от 23 до 25 °С.

– Разница температуры на уровне пола и уровне головы оператора в положении сидя не должна превышать 3 °С.

– Относительная влажность воздуха на рабочем месте оператора должна составлять 40-60%.

– Скорость движения воздуха на рабочем месте оператора должна быть 0,1 м/с.

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с мониторами (в направлении тыла поверхности одного монитора и экрана другого монитора) должно быть не менее двух метров, а расстояние между боковыми поверхностями мониторами – не менее 1,2 м. Экран монитора должен находиться от глаз на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров шрифтов букв, знаков и символов.

Компьютер должен быть установлен так, чтобы, подняв глаза от экрана, можно было увидеть самый удаленный предмет в комнате. Удачным является расположение рабочего места, когда лицо оператора обращено к входному проему. Возможность перевести взгляд на дальнее расстояние - один из самых эффективных способов разгрузки зрительной системы во время работы с компьютером. Следует избегать расположения рабочего места в углах комнаты или лицом к стене - расстояние от компьютера до стены не менее 1м, экраном к окну, а также лицом к окну - свет из окна является нежелательной нагрузкой на глаза.

Характерной особенностью работой за ПК является статический режим: большой объем работы приходится выполнять в сидячем положении. При этом большинство групп мышц находятся в напряжении, что приводит к быстрой утомляемости, способствует развитию профессиональных патологических изгибов позвоночника. Неправильное расположение дисплеев по высоте - слишком низкое, под неправильным углом является основной причиной сутулости; слишком высокое, приводит к длительному напряжению шейного отдела - может привести к развитию остеохондроза [24].

Поэтому рабочее место с дисплеем должно обеспечивать оператору возможность удобного выполнения работ в положении сидя и не создавать перегрузки костно-мышечной системы.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм.

Основным рабочим положением является положение сидя. Рабочая поза сидя практически не утомляет пользователя. Лучше всего планировать рабочее место таким образом, чтобы документация и различное оборудование располагались на одном месте на постоянной основе.

Дисплей на рабочем месте оператора должен располагаться так, чтобы изображение в любой его части было различимо без необходимости поднять или опустить голову. Также он должен быть установлен ниже уровня глаз оператора. Угол наблюдения экрана оператором относительно горизонтальной линии взгляда не должен превышать 60°.

Конструкция рабочего кресла должна обеспечивать поддержание оптимальной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения постоянного напряжения мышц шеи и плеча, а также спины для предотвращения развития утомления. Тип рабочего кресла следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ. Рабочее кресло должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию. Поверхность сиденья, спинки и других элементов кресла должна быть мягкой, с нескользящим и не электризующимся воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Клавиатура на рабочем месте оператора должна располагаться так, чтобы обеспечивалась оптимальная видимость экрана и иметь возможность свободного перемещения.

Также стоит рассмотреть аспект безопасности и удобства использования разрабатываемого приложения для пользователя.

Интерфейс приложения выполнен в единой цветовой гамме. Отсутствуют мерцающие и резкие анимации, все переходы между окнами осуществляются плавно. Отсутствуют анимации, отвлекающие пользователя или рассеивающие его внимание. В качестве основной цветовой гаммы выбраны голубой, белый, светло-зеленый цвета. Не используются яркие цвета, вызывающие напряжение глаз пользователя.

Интерфейс интуитивно понятен. Функционал приложения сгруппирован по назначению и вынесен в отдельные окна. Все окна выполнены в едином

графическом стиле, с одинаковымм расположением основных элементов управления и навигации. Для обозначения сходных операций используются сходные графические значки. Внешнее поведение сходных элементов интерфейса реализовано одинаково К неинтерактивным элементам применяется эффект прозрачности, поэтому в любой момент времени видно, с какими элементами можно взаимодействовать.

Навигация по приложению осуществляется с помощью характерных для сенсорного ввода жестов, что делает её интуитивно понятной.

Число элементов управления сведено к минимуму, поэтому пользователям не требуется проходить обучение для работы с приложением.

Для всех кнопок выбраны иконки отражающие их функциональное назначение. Размеры кнопок достаточно большие, чтобы иконки были хорошо видны и нажатие не вызывало проблем.

Так как при работе приложения предполагается альбомная ориентация устройства, элементы управления сосредоточены по центру и в верхних углах экрана, чтобы пользователь не перекрывал их руками.

Шрифт всех текстовых элементов достаточно крупный и разборчивый, и легко читается с экрана мобильного устройства. Также в некоторых надписях вокруг букв используется черная обводка, повышающая удобочитаемость текста.

4.2 Экологичность

Электронные отходы — один из значимых источников загрязнения окружающей среды, но с другой стороны, они также могут рассматриваться в качестве вторичного ресурса, так как они содержат ценные компоненты. Таким образом, переработка электронных отходов не только является важным элементом в общей структуре управления отходами, но и представляет интерес с точки зрения материального и ресурсного потенциала [25].

Крупные сети по продаже электронной и электробытовой техники периодически проводят акции продажи оборудования с заменой старого на новое в ходе, которых удается собирать достаточно большие объемы отходов элек-

тронного и электрического оборудования. Производители товаров этой категории в организации переработки отходов не участвуют. Переработкой отходов этого оборудования занимается несколько десятков небольших компаний, расположенных в разных регионах страны.

Основными технологиями утилизации и обезвреживания электрического и электронного оборудования, утратившего потребительские свойства, являются их разборка с извлечением компонентов, представляющих ресурсную ценность, с последующей передачей их на специализированные предприятия, осуществляющие их переработку в качестве вторичного сырья.

Комплектующие компьютерной техники сортируют по своей ценности: материнские платы, процессоры, блоки питания, провода. Самое ценное в компьютере — материнская плата. Они разбираются вручную и сортируются по своей ценности, так как каждая содержит разные драгметаллы.

По своим физическим и химическим свойствам многокомпонентный электронный лом не может направляться в металлургическую плавку без механической разделки с целью выделения отдельных компонентов или групп. Технологические процессы современной переработки радиоэлектронного лома, как правило, включают в себя ручную дифференцированную разделку, механические измельчение (дробление), обогащение полученных концентратов и последующие виды переработки.

Процесс переработки мониторов начинается с ручного демонтажа составных частей. Демонтированные компоненты, как правило, сортируются на пластик, металл, печатные платы, провода, люминесцентные лампы, ЖК-дисплеи для дальнейшей переработки. Основную часть электронной техники составляют металл и пластик. Особую опасность для окружающей среды представляют ЖК-дисплеи с CCFL-подсветкой (люминесцентная лампа с холодным катодом).

Имеющиеся технологии утилизации ЖК-дисплеев в основном направлены на извлечение и повторное использование основного составляющего компонента стекла. Одной из составных частей ЖК-дисплеев с CCFL-подсветкой является ртутьсодержащая люминесцентная лампа. Основную опасность пред-

ставляют разбитые лампы, так как может происходить испарение ртути.

Основные экологические проблемы, возникающие при обезвреживании и утилизации электрического и электронного оборудования связаны с наличием в них токсичных веществ. При утилизации дисплеев важно знать, что в состав некоторых из них входят люминесцентные лампы, содержащие ртуть. Поэтому при демонтаже необходимо применять мероприятия по предотвращению попадания ртути в природные объекты.

При современных методах утилизации доля извлекаемого вторичного ресурса невелика, значительное количество вторичных отходов направляется на захоронение.

Так же следует рассмотреть порядок утилизации мобильных устройств, так как именно для них предназначен разрабатываемый программный продукт.

Утилизация мобильных телефонов проводится с целью извлечения вторичных материальных ресурсов для дальнейшего использования. Утилизация мобильного телефона начинается с сортировки телефонов. Если компания специализируется на переработке аппаратов определенного производителя, то сортировки не происходит. Если же компания «многопрофильная», то нередко аппараты сортируются не только по производителю, но и по другим критериям. Второй этап — разборка телефонов на составные части. Далее три самые важные составляющие — аккумулятор, пластиковый корпус и печатная плата — идут на переработку своим путем.

Аккумуляторы из-за большого содержания токсичных веществ (мышьяк, свинец, ртуть) поступают, как правило, на специальные заводы по переработке АКБ. Печатная плата отсоединяется от элементов корпуса и идет на получение драгоценных металлов. Далее корпусы и печатные платы снова дробятся и в достаточно измельченном виде — практически в состоянии пыли — поступают в сортировочную камеру, где посредством химических реакций или механических действий происходит окончательная сортировка сырья. Дорогостоящие металлы экстрагируются, обрабатываются, дополнительно очищаются и поступают на соответствующие предприятия. Пластик и резина чаще всего поступа-

ют на дорожно-строительные заводы и добавляются в дорожное покрытие. Эмиссии загрязняющих веществ в окружающую среду могут быть связаны с проливами (и возможным последующим испарением) реагентов на этапах экстракции и обогащения драгоценных металлов.

4.3 Чрезвычайные ситуации

Наиболее распространенными источниками возникновения чрезвычайных ситуаций техногенного характера являются пожары.

Пожар – это вышедший из-под контроля процесс горения, уничтожающий материальные ценности и создающий угрозу жизни и здоровью людей. Основные причины пожаров на предприятиях - неосторожное обращение с огнем, оставленные без присмотра электроприборы, проведение с нарушениями требований правил пожарной безопасности огневых, строительных и других пожароопасных работ, курение в не установленных местах, использование легко-воспламеняемых веществ.

Основными опасными факторами пожара являются тепловое излучение, высокая температура, отравляющее действие дыма (продуктов сгорания: окиси углерода и др.) и снижение видимости при задымлении. Критическими значениями параметров для человека, при длительном воздействии указанных значений опасных факторов пожара, являются:

- температура – 70 С°;
- плотность теплового излучения – 1,26 кВт/м2;
- концентрация окиси углерода – 0,1% объема;
- видимость в зоне задымления – 6-12 м.

В число предупредительных мероприятий могут быть включены мероприятия, направленные на устранение причин, которые могут вызвать пожар, на ограничение (локализацию) распространения пожаров, создание условий для эвакуации людей и имущества при пожаре, своевременное обнаружение пожара и оповещение о нем, тушение пожара, поддержание сил ликвидации пожаров в постоянной готовности.

Согласно НПБ 105-03 помещения с ЭВМ относятся к пожароопасным

помещениям категорий В1-В4 (т. к. содержат материалы способные при взаимодействии с водой или друг с другом гореть). Мебель и другие бытовые предметы не должны препятствовать эвакуации, а также все провода должны быть спрятаны в стену или кабель-каналы.

Для предотвращения возможного пожара необходимо соблюдать следующие правила:

– не хранить и не применять горючие жидкости, взрывчатые вещества, баллоны с газами и рядом с ЭВМ;

– не использовать электронагревательные приборы;

– не эксплуатировать провода электроприборов с поврежденной изоляцией;

– не пользоваться поврежденными розетками, и прочим электрооборудованием;

– не накрывать светильники, бытовые приборы бумагой, тканью и другими горючими материалами;

– не курить в помещении;

– оставлять без наблюдения включенную в сеть радиоэлектронную ПЭВМ;

– не пользоваться неисправной аппаратурой;

– не разрешается ремонтировать блоки ЭВМ непосредственно в помещениях, где они располагаются;

– не нарушать правила эксплуатации ПЭВМ;

– раз в 3 месяца необходимо проводить санитарную очистку.

По окончании работы необходимо обесточить все электроприборы и осмотреть помещения на наличие признаков возгорания, а также необходимо выключить автомат питания в распределительном щите, если такой имеется.

Таким образом соблюдение технологических режимов производства, содержание оборудования, особенно энергетических сетей, в исправном состоянии позволяет, в большинстве случаев, исключить причину возгорания.

Своевременное обнаружение пожара может достигаться оснащением производственных и бытовых помещений системами автоматической пожарной сигнализации или, в отдельных случаях, с помощью организационных мер.

Для тушения пожара на начальной стадии применяются огнетушители. Так как в помещениях с ПЭВМ наиболее вероятные классы пожаров– «А» и «Е» (т.е. могут гореть в основном твердые вещества, горение которых сопровождается тлением – класс А; или возможны пожары, вызванные возгоранием электроустановок – класс Е), то использовать нужно углекислотный и порошковый огнетушители. Огнетушитель углекислотный ОУ-5 предназначен для тушения любых материалов, предметов и веществ, а также электроустановок, находящихся под напряжением до 1 кВ, применяется для тушения ПЭВМ и оргтехники. При пожаре поднести огнетушитель как можно ближе к огню, направить раструб в очаг пожара, сорвать пломбу (выдернуть чеку), открыть вентиль, нажать на пусковой рычаг, направить струю выходящего газа на огонь. Во время работы раструб нельзя держать рукой, т. к. он имеет очень низкую температуру. Огнетушитель порошковый ОП-5 предназначен для тушения твердых, жидких, газообразных веществ и электроустановок, находящихся под напряжением до 1 кВ, применяется для тушения ПЭВМ и оргтехники. При пожаре поднести огнетушитель к очагу загорания, выдернуть чеку, нажать на рычаг, направить шланг с распылителем на огонь. Расстояние от возможного очага возгорания до места размещения огнетушителя не должно превышать 20 м, если ПЭВМ установлены в общественных зданиях и сооружениях. В замкнутых помещениях объемом до 50 м3 вместо переносных огнетушителей (или в дополнение к ним) можно использовать подвесные само-срабатывающие порошковые огнетушители (ОСП и другие).

При обнаружении возгорания нужно реагировать на пожар быстро, используя все доступные способы для тушения огня (песок, воду, огнетушители и т.д.). Если потушить огонь в кратчайшее время невозможно, необходимо позвонить в пожарную службу, сообщить всю необходимую информацию.

При эвакуации горящие помещения и задымленные места следует проходить быстро, задержав дыхание, защитив нос и рот влажной плотной тканью.

В сильно задымленном помещении передвигаться нужно ползком или пригнувшись – в прилегающем к полу пространстве чистый воздух сохраняется

дольше. Если нет возможности покинуть здание, то необходимо закрыться в менее задымлённой комнате, не дать дыму попадать в комнате любыми подручными средствами и открыв все окна ожидать помощи спасательной бригады.

ЗАКЛЮЧЕНИЕ

Целью данной работы было создание мобильной игры. Для этого в ходе выполнения выпускной квалификационной работы были решены следующие задачи:

– Рассмотрены основные жанры видеоигр и обоснован выбор жанра разрабатываемой игры;

– Проведен подробный анализ наиболее близкого к разрабатываемому проекту аналога;

– Произведен сравнительный обзор программных средств, используемых для разработки видеоигр, из которых были выбраны наиболее подходящие.

– Выполнен анализ требований к разрабатываемому программному продукту.

– Проведены этапы проектирования и разработки программного продукта.

В результате был получен прототип мобильной видеоигры, который без изменений программного кода может быть портирован на платформы iOS и Android средствами Unity. Выполнена локализация приложения на русский и английский языки. Создан пользовательский интерфейс, адаптированный под разные разрешения экрана, включая интерфейс уровня и главное меню приложения. Разработано управление, адаптированное под мобильные устройства. Добавлена возможность сохранения прогресса пользователя. Разработана система смены скинов. Создан управляемый персонаж, выполнена его скелетная анимация. Реализован прототип уровня, включающий нескольких интерактивных объектов. При этом некоторые из разработанных функциональных модулей могут быть использованы для других проектов.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1 Разница между 2D- и 3D-играми в Unity [Электронный ресурс]. URL: https://unity.com/ru/how-to/difference-between-2D-and-3D-games (дата обращения: 20.06.2021).

2 Гибсон, Дж. Б. Unity и C#. Геймдев от идеи до реализации / Дж. Б. Гибсон – СпБ: Питер, 2019. – 928 с.

3 Джейсон, Г. Игровой движок. Программирование и внутреннее устройство / Г. Джейсон – СпБ: Питер, 2018. – 1136 с.

4 Цехнер М. Программирование игр под Android. / М.. Цехнер - СпБ: Питер, 2017. – 352 с.

5 Тюкачев, Н. А. C#. Алгоритмы и структуры данных : учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 232 с. — ISBN 978-5-8114-2566-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/169153 (дата обращения: 20.06.2021).

6 Тучкевич, Е.И. Adobe Photoshop CC 2019. Мастер-класс Евгении Тучкевич / Е.И. Тучкевич – СпБ: БХВ, 2020. – 496 с.

7 Хокинг, Дж. Unity в действии. Мультиплатформенная разработка на C# учебное пособие/ Дж. Хокинг - СпБ: Питер, 2016. - 336с.

8 Ламот, А. Основы разработки игр на Unity 3D / А. Ламот , Д. Ратклифф, М. Семинаторе, Д. Тайлер – СпБ: Питер, 2016. – 716 с.

9 Audio tutorial for Unity: the Audio Mixer [Электронный ресурс]. URL: https://www.raywenderlich.com/532-audio-tutorial-for-unity-the-audio-mixer (дата обращения: 20.06.2021).

10 Unity Scripting Reference [Электронный ресурс]. URL: https://docs.unity3d.com/ScriptReference/ (дата обращения: 20.06.2021).

11 Unity UI: Unity User Interface [Электронный ресурс]. URL: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/index.html (дата обращения: 20.06.2021).

12 Торн, А. Искусство создания сценариев в Unity. / А. Торн - М: ДМК, 2016. – 360с.

13 Дикинсок, К. Оптимизация игр в Unity 5. / К. Дикинсон – М: ДМК, 2017. – 305 с.

14 JSON Serialization [Электронный ресурс]. URL: https://docs.unity3d.com/ Manual/ JSONSerialization.html (дата обращения: 20.06.2021).

15 Томас К. Алгоритмы: построение и анализ. / Томас Кормен, Чарльз Лейзерсон: Альфа-Книга, 2018. – 302 с.

16 Localizing Unity Games with the Official Localization Package [Электронный ресурс]. URL: https://phrase.com/blog/posts/localizing-unity-games-official-localization-package/ (дата обращения: 20.06.2021).

17 Rigging a Sprite with the 2D Animation Package [Электронный ресурс]. URL: https://learn.unity.com/tutorial/rigging-a-sprite-with-the-2d-animation-package#6017535aedbc2a69ae9b3b9b (дата обращения: 20.06.2021).

18 Торн, А. Основы анимации в Unity: учебное пособие / А. Торн - М: ДМК, 2016. - 176с.

19 Cinemachine [Электронный ресурс]. URL: https://learn.unity.com/tutorial/cinemachine# (дата обращения: 20.06.2021).

20 Мэннинг Д. Unity для разработчика. Мобильные мультиплатформенные игры. / Д. Мэннинг, П. Батфилд-Эддисон - СпБ: Питер, 2018. – 352 с.

21 Goldstone, W. Unity Game Development Essentials/ W. Goldstone – UK, Birmingham: Packt, 2017. – 266 с.

22 СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* (с Изменением N 1).

23 ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения.

24 Симакова Н.Н. Организация рабочих мест с персональными электронно-вычислительными машинами (ПЭВМ) : учебное пособие / Симакова Н.Н.. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2013. — 78 с. — Текст : электронный // Электронно-

библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/55486.html (дата обращения: 20.06.2021).

25 ИТС 15-2016 Утилизация и обезвреживание отходов (кроме обезвреживания термическим способом (сжигание отходов).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* (с Изменением N 1).

2 ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения.

3 ИТС 15-2016 Утилизация и обезвреживание отходов (кроме обезвреживания термическим способом (сжигание отходов).

4 Audio tutorial for Unity: the Audio Mixer [Электронный ресурс]. URL: https://www.raywenderlich.com/532-audio-tutorial-for-unity-the-audio-mixer (дата обращения: 20.06.2021).

5 Cinemachine [Электронный ресурс]. URL: https://learn.unity.com/tutorial/cinemachine# (дата обращения: 20.06.2021).

6 Goldstone, W. Unity Game Development Essentials/ W. Goldstone – UK, Birmingham: Packt, 2017. – 266 с.

7 JSON Serialization [Электронный ресурс]. URL: https://docs.unity3d.com/ Manual/ JSONSerialization.html (дата обращения: 20.06.2021).

8 Localizing Unity Games with the Official Localization Package [Электронный ресурс]. URL: https://phrase.com/blog/posts/localizing-unity-games-official-localization-package/ (дата обращения: 20.06.2021).

9 Rigging a Sprite with the 2D Animation Package [Электронный ресурс]. URL: https://learn.unity.com/tutorial/rigging-a-sprite-with-the-2d-animation-package#6017535aedbc2a69ae9b3b9b (дата обращения: 20.06.2021).

10 Unity Scripting Reference [Электронный ресурс]. URL: https://docs.unity3d.com/ScriptReference/ (дата обращения: 20.06.2021).

11 Unity UI: Unity User Interface [Электронный ресурс]. URL: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/index.html (дата обращения: 20.06.2021).

12 Гибсон, Дж. Б. Unity и C#. Геймдев от идеи до реализации / Дж. Б. Гибсон – СпБ: Питер, 2019. – 928 с.

13 Джейсон, Г. Игровой движок. Программирование и внутреннее устройство / Г. Джейсон – СпБ: Питер, 2018. – 1136 с.

14 Дикинсок, К. Оптимизация игр в Unity 5. / К. Дикинсон – М: ДМК, 2017. – 305 с.

15 Ламот, А. Основы разработки игр на Unity 3D / А. Ламот , Д. Ратклифф, М. Семинаторе, Д. Тайлер – СпБ: Питер, 2016. – 716 с.

16 Мэннинг Д. Unity для разработчика. Мобильные мультиплатформенные игры. / Д. Мэннинг, П. Батфилд-Эддисон - СпБ: Питер, 2018. – 352 с.

17 Разница между 2D- и 3D-играми в Unity [Электронный ресурс]. URL: https://unity.com/ru/how-to/difference-between-2D-and-3D-games (дата обращения: 20.06.2021).

18 Симакова Н.Н. Организация рабочих мест с персональными электронно-вычислительными машинами (ПЭВМ) : учебное пособие / Симакова Н.Н.. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2013. — 78 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/55486.html (дата обращения: 20.06.2021).

19 Томас К. Алгоритмы: построение и анализ. / Томас Кормен, Чарльз Лейзерсон: Альфа-Книга, 2018. – 302 с.

20 Торн, А. Искусство создания сценариев в Unity. / А. Торн - М: ДМК, 2016. – 360с.

21 Торн, А. Основы анимации в Unity: учебное пособие / А. Торн - М: ДМК, 2016. - 176с.

22 Тучкевич, Е.И. Adobe Photoshop CC 2019. Мастер-класс Евгении Тучкевич / Е.И. Тучкевич – СпБ: БХВ, 2020. – 496 с.

23 Тюкачев, Н. А. C#. Алгоритмы и структуры данных : учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 232 с. — ISBN 978-5-8114-2566-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/169153 (дата обращения: 20.06.2021).

24 Хокинг, Дж. Unity в действии. Мультиплатформенная разработка на C# учебное пособие/ Дж. Хокинг - СпБ: Питер, 2016. - 336с.

25 Цехнер М. Программирование игр под Android. / М.. Цехнер - СпБ: Питер, 2017. – 352 с.