**РЕФЕРАТ**

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ 3D-ИГРЫ НА UNITY И ЕГО ИСПЫТАНИЕ : дипломный проект / В.С. Евтух. – Минск : БГУИР, 2021, – п.з. – 109 с., чертежей (плакатов) – 6 л. формата А1.

Ключевые слова: мобильное приложение, 3D-игра, Unity.

Целью дипломного проекта является разработка мобильного приложения для 3D-игры на Unity, предназначенное для развлекательного времяпрепровождения людей независимо от пола или возраста.

Объектом дипломного проекта является мобильное приложения для 3D-игры.

Предмет дипломного проекта – принципы проектирования мобильных игр при помощи платформы для разработки игр Unity 3D.

При разработке системы платформой для разработки была выбрана Unity 3D. Исходя из специфики работы с платформой Unity, языком программирования был выбран C#.

В ходе дипломного проектирования проведён сравнительный анализ существующих решений мобильных игр, изучены современные технологии разработки 3D-игр, проведено инженерно-психологическое проектирование приложения, спроектированы архитектура и структура мобильного приложения, протестирована работоспособность системы, выполнено технико-экономическое обоснование разработки, подготовлен комплекс мер по защите от статического электричества для обеспечения безопасных условий труда при разработке приложения.

В результате проектирования разработано мобильное приложение для 3D-игры на Unity.

**СОДЕРЖАНИЕ**

Введение ...............................................................................................................6 1 Мобильные приложения для 3D-игр на Unity...............................................8

1.1 Сравнительный анализ существующих решений

мобильных игр на Unity.............................................................................8 1.2 Современные технологии разработки мобильных 3D-игр ....................15 1.3 Выводы и постановка задач на дипломное проектирование .................22

2 Инженерно-психологическое проектирование

мобильного приложения для 3D-игры на Unity ...........................................24 2.1 Анализ функций и их распределение в проектируемой системе ..........24 2.2 Проектирование деятельности человека в системе

«человек-мобильное устройство-среда».................................................28 2.3 Разработка эргономических требований и сценария

информационного взаимодействия.........................................................32 3 Разработка мобильного приложения для 3D-игры на Unity

и его испытание..............................................................................................49 3.1 Проектирование архитектуры и структуры мобильного

приложения для 3D-игры.........................................................................49 3.2 Программная реализация мобильного приложения для 3D-игры.........54 3.3 Тестирование мобильного приложения для 3D-игры............................59

4 Технико-экономическое обоснование разработки и реализации

мобильного приложения для 3D-игры на Unity ...........................................65 4.1 Описание функций, назначения и потенциальных пользователей

мобильного приложения для 3D-игры на Unity .....................................65 4.2 Расчёт затрат на разработку мобильного приложения

для 3D-игры на Unity................................................................................67 4.3 Оценка экономического эффекта и расчет показателей

эффективности инвестиций при разработке программного

обеспечения для свободной реализации на рынке IT.............................70 5 Охрана труда. Обеспечение безопасных условий труда

программиста: выбор и обоснование комплекса мер по защите

от статического электричества......................................................................75 Заключение.........................................................................................................81 Список использованных источников................................................................83 Приложение А (обязательное) Фрагмент программного кода ........................86 Ведомость дипломного проекта......................................................................109

**ВВЕДЕНИЕ**

Сегодня индустрия компьютерных игр представляет собой быстро развивающийся сектор глобальной экономики, связанный с разработкой, продвижением и продажей игр. А сами игры теперь позиционируются и воспринимаются пользователями как качественное многожанровое интерактивное развлечение, которое постепенно становится все популярнее.

По мнению экспертов, именно игровая индустрия становится проводником идей современного творчества и новаторства. Компьютерные игры сейчас присутствуют в различных сегментах массовой культуры, а некоторые даже стали ее феноменами (например, Mario или Final Fantasy). В некоторых странах, например в США, компьютерные игры были признаны отдельным видом искусства. Кроме того, было положено начало новой субкультуре со своим языком, иконами, уникальными событиями и мероприятиями. Формируется мощная экосистема. Гейминг становится высокооплачиваемой профессией. Согласно экспертным оценкам, свыше 35 % всех геймеров в мире готовы сделать гейминг своим основным родом занятий [1].

О возрастающей роли игр в жизни современного общества также свидетельствует активное проникновение игровых практик в неигровые процессы – геймификация (gamification) – в ряд традиционных отраслей (например, ритейл, реклама, банковский сектор, образование, гражданские авиаперевозки) в целях привлечения, удержания и обучения целевой аудитории [1].

Эксперты и участники отрасли отмечают развитие ряда сильных тенденций, которые определяют развитие глобальной игровой индустрии. Так, большую роль продолжают играть новые технологии (виртуальные, мобильные, облачные и др.), наблюдается слияние виртуальных и физических сред, социализация игр.

Одним из центральных трендов индустрии является мобильный гейминг. В настоящий момент мобильные игры – игровые приложения для смартфонов и планшетов – самый быстрорастущий сегмент игрового рынка. Смартфоны являются самым популярным игровым устройством во всем мире [1].

В современной жизни часто бывает, что человек вынужден ехать в автобусе или стоять в очереди, где ему попросту нечем заняться. Для решения этой проблемы существуют игровые приложения на смартфоны, которые

6

помогут немного скрасить ожидание или просто позволят увлекательно провести время.

Целью дипломного проекта является разработка мобильного приложения для 3D-игры на Unity, предназначенное для развлекательного времяпрепровождения людей независимо от пола или возраста.

Объектом дипломного проекта является мобильное приложения для 3D-игры.

Предмет дипломного проекта – принципы разработки и шаблоны проектирования мобильных приложений.

Для достижения цели дипломного проекта необходимо решить следующие задачи:

провести сравнительный обзор аналогов разрабатываемой системы и провести анализ их достоинств и ограничений;

проанализировать технологии, подходящие для разработки мобильных приложений, и выбрать наиболее удобные и эффективные;

провести инженерно-психологическое проектирование приложения; спроектировать и реализовать приложение, используя эффективные

шаблоны проектирования и разработки мобильных приложений; провести тестирование мобильного приложения;

выполнить технико-экономическое обоснование эффективности разработки и использования приложения;

протестировать работоспособность и корректность функционирования приложения;

разработать комплекс мер по защите от статического электричества для обеспечения безопасных условий труда при разработке приложения.

Преимуществом приложения можно считать использование кроссплатформенного решения для реализации игр, что значительно сокращает сроки и затраты на разработку. Это позволит охватить как можно большую часть аудитории рынка мобильных игр.

7

**1** **МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ ДЛЯ 3D-ИГР НА UNITY**

**1.1 Сравнительный анализ существующих решений мобильных игр на Unity**

В наше время огромную роль в жизни человека играют мобильные телефоны. Людям ежедневно необходимо иметь под рукой определённое количество информации, такой как контакты, записки и т.д. Самый быстрый и практичный способ хранения и получения этой информации – мобильный телефон.

Помимо базовых ежедневных задач, люди используют смартфоны как средство развлечения в свободное время. Игры ¬ это отличный способ скоротать время в очередях, общественном транспорте или в школе на перемене.

В последнее время рынок мобильных игр растёт с невероятной скоростью. Постоянно появляются новые жанры, либо совершенствуются уже имеющиеся и проверенные временем. На этом рынке часто становятся популярны игры с простой игровой механикой, которые не требуют от игрока больших познаний в игровой индустрии. В такие игры могут играть все, аудитория мобильных игр очень широка, в нее играют и дети, и взрослые

Рынок мобильных игр является одним из наиболее востребованных и перспективных с точки зрения целевой аудитории и рентабельности в целом.

Основными факторами популярности мобильных игр являются ¬ жанр мобильной игры и ее визуальная составляющая. Именно они диктуют разработчикам логику первичного мониторинга потребительских предпочтений на рынке с тем, чтобы в дальнейшем представить качественный продукт, который будет пользоваться высоким спросом [2].

Один из популярнейших жанров – Idle. Idle-игры (они же «кликеры или инкрементальные игры) – это игры, целиком посвящённые управлению потоками доходов. Как и симуляторы, они фокусируются на принятии решений, которые увеличивают доход. Ядром игры является скорость генерации ресурсов. Но экономика с единственной валютой, которая так быстро растёт, имеет и недостатки [3].

Визуальная составляющая будет выполнена в виде 3D-графики с игровой камерой, фиксированной на главном здании. Также будет присутствовать развитие команды игрока.

Геймлей разрабатываемого проекта будет основан на облагораживания города посредством нанесения граффити на стены зданий.

8

Разрабатываемое мобильное приложение должно быть устойчиво к аппаратным и программным сбоям, быть защищено от неправильных действий пользователей и обладать максимальной восстанавливаемостью при возникновении сбоев.

Актуальность проекта обусловлена внедрением и развитием информационных технологий в повседневной жизни человека. Разработка, выпуск и последующее добавление мобильной игры в сеть интернет позволит пользователям проводить время более увлекательно, например в поездках на общественном транспорте.

Рассмотрим существующие аналоги разрабатываемой мобильной игры. Оценка данных систем будет производиться исходя из критериев:

графическая составляющая игры; адаптивная вёрстка;

юзабилити;

производительность;

количество контента в игре; кроссплатформенность;

стабильность работы; ценовая политика;

локализация. *Game Dev Tycoon.*

Игра позволяет пройти путь разработчика от самых низов до лидера игровой индустрии. Доступна на ПК, Android и iOS.

В ходе создания игры разработчик генерирует очки дизайна, технологий и исследований. Первые два типа отражают качество игры и показывают на что в ней делается упор. Очки исследований используются для открытия новых технологий и возможностей.

Готовая игра рассылается прессе, и та выставляет оценки. Отзывы журналистов представляют собой итоговый вердикт продукту – люди не станут покупать то, что не похвалили. Реакция прессы зависит от многих факторов, но решающим является количество набранных очков дизайна и технологий. Для того, чтобы продавать много копий, нужно получать высокие оценки [4].

На рисунке 1.1 представлен главный экран Game Dev Tycoon.

9



Рисунок *1.1 –* Главный экран игры [5]

В ходе игрового процесса, игроку необходимо разрабатывать игры. При разработке игр, пользователем выбирается тематика. На рисунке 1.2 представлен экран выбора тематики игры.

Рисунок *1.2 –* Экран выбора тематики разрабатываемой игры [5]

10

После завершения этапа разработки игры, пользователю отображается экран результатов, на котором представлены итоговые оценки и отзывы о разработанной игре. На рисунке 1.3 представлен экран результатов.

Рисунок *1.3 –* Экран результатов [5]

*Idle Light City.*

Игра, где каждый может стать настоящим магнатом. Город – это огромная территория и для его освоения придётся приложить большие усилия. Большое количество домов, каждый из которых нуждается в лампочках и освещении, бустеры, которые позволят производить ещё больше лампочек. В процессе игры появляется возможность построить в городе парк развлечений, который будет поглощать готовые лампочки, чтобы привлекать своим светом клиентов. Игра будет добывать их даже в офлайн-режиме, но значительно меньше, чем онлайн.

Основная задача в игре – организовать управление фабрикой лампочек. Игроку нужно произвести как можно больше лампочек. Открываются новые здания, и их нужно осветить, чтобы собрать прибыль. На заработанные деньги игрок сможет усовершенствовать свои производственные мощности и расширить город [6].

На рисунке 1.4 представлены экраны игрового процесса игры Idle Light City.

11



Рисунок *1.4* – Экраны игрового процесса [7]

*Graffiti Idle.*

Это игра, в которой нужно рисовать граффити и разблокировать новых членов команды. На старте игры, игроку доступно только одно здание и один персонаж, который будет рисовать граффити.

Рисование происходит по нажатию и удержанию пальца на экране. В игре присутствует возможность улучшать параметры персонажа, что позволит ускорять процесс рисования граффити.

Параметры, которые игрок может улучшать: уровень дохода;

скорость рисования;

количество членов команды.

В игре отсутствует музыкальное сопровождение и игра доступна только на платформе iOS. На рисунке 1.5 представлен главный экран игры Graffiti Idle.

12



Рисунок *1.5 –* Главный экран игры [8]

Проведём описание критериев для взаимного сравнения выбранных мобильных игр. В таблице 1.1 приведены критерии сравнения мобильных игр.

Таблица *1.1 –* Критерии сравнения мобильных игр

Название критерия 1

1 Графическая

составляющая игры

Описание критерия 2

Качество визуального сопровождения игры

2 Адаптивная вёрстка Правильное отображение интерфейса на различных устройствах

3 Юзабилити Способность быть понимаемым, изучаемым, используемым и привлекательным для пользователя

4 Производительность Количество кадров в секунду (FPS)

5 Количество контента в Количество полностьюготовыхи играбельных

игре

6 Кроссплатформенность

уровней

Возможность поддерживать сразу несколько

платформ

13

Продолжение таблицы *1.1* 1

7 Стабильность работы 8 Ценовая политика

9 Локализация

2 Работа без программных сбоев Стоимость игры

Присутствие мультиязычности

Проведём сравнительный анализ выбранных мобильных игр. Сравнение аналогов разрабатываемой мобильный игры приведено в таблице 1.2.

Таблица *1.2 –* Сравнительный анализ мобильных игр

Критерий

1 Графическая составляющая игры

Адаптивная вёрстка

Юзабилити

Производительность

Game Dev Tycoon 2

Визуальное сопровож-дение выпол-нено в 2D-изометричес-ком стиле. Дизайн устарел.

Все интер-фейсы отоб-ражаются правильно.

Сложна для понимания.

60 FPS

Idle Light City

3 Визуальное сопровождение выполнено в 2D-изометрическом стиле. Красивый и насыщенный дизайн.

Все интерфейсы отображаются правильно.

В игре присутствует обучение, что позволяет легко разобраться в игре.

60 FPS

Graffiti Idle

4 Визуальное сопровождение выполнено в 3D-виде сбоку. Цветовая палитра подобрана не гармонично.

Все интерфейсы отображаются правильно.

В игре разобраться просто из-за небольшого функционала.

60 FPS

14

Продолжение таблицы *1.2*

1 Количество контента в игре

Кроссплатформенность

Стабильность работы

Ценовая политика

Локализация

2 Уровней хватает на 8 часа непрерывной игры. Windows, iOS, Android Программных сбоев найдено не было.

Игра платная. Реклама отсутствует.

Локализация присутствует на 16 языках.

3 Уровней хватает на 2 часа непрерывной игры.

iOS, Android

На некоторых устройствах прогресс игры не сохраняется. Игра бесплатная. Присутствует реклама. Локализация

отсутствует.

4 Уровней хватает на 5 минут непрерывной игры.

iOS

Программных сбоев найдено не было.

Игра бесплатная. Реклама отсутствует. Локализация

отсутствует.

Исходя из проведённого сравнительного анализа аналогов, были подготовлены требования для разрабатываемой мобильной 3D-игры на Unity:

использование смешанных визуальных стилей; грамотно подобранная цветовая палитра;

адаптация интерфейсов к различным разрешениям экранов; количество кадров в секунду – не менее 60;

количество контента – не менее 50 часов игры; поддержка на iOS и Android платформах;

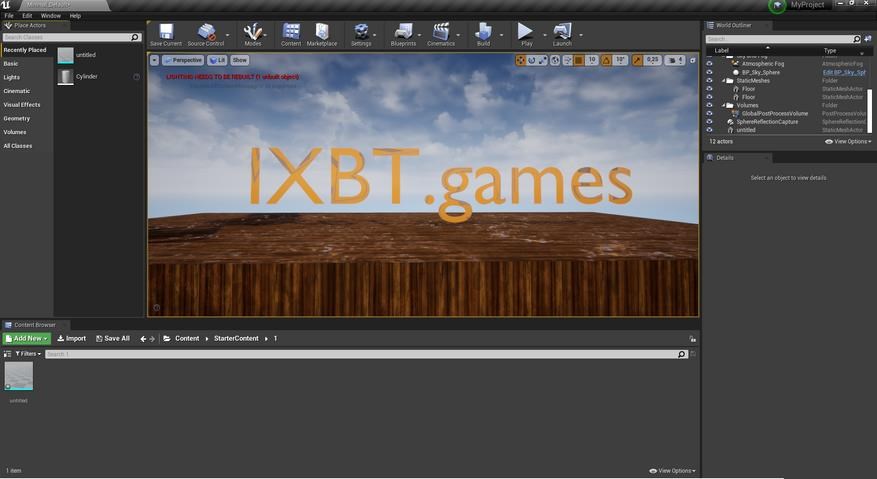
отсутствие программных сбоев;

игра должна быть бесплатной со встроенной рекламой.

**1.2 Современные технологии разработки мобильных 3D-игр**

Игровой движок является базовым программным обеспечением, на котором разрабатывается и действует игра, общее название для совокупности нескольких подсистем (звуковая, графическая, физическая и прочие). В

15

современных игровых движках эти подсистемы – модульные, и качество проработки у них различается от движка к движку [9].

Разберем три самых популярных открытых игровых движка: Unreal Engine 4, Cryengine 5 и Unity и оценим их по следующим пунктам:

доступность;

поддерживаемые платформы; поддержка пользователя;

доступность игровых ассетов и инструкций; личное мнение.

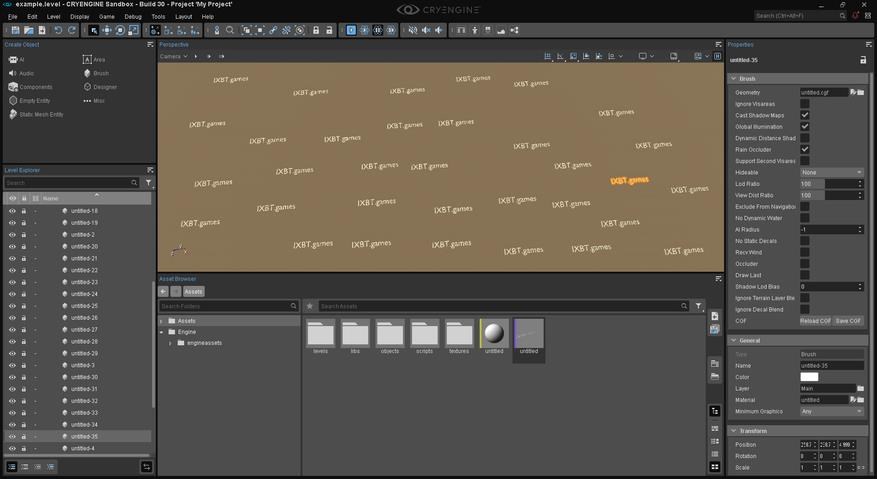
*Unreal Engine.*

Интерфейс движка интуитивно понятен и не требует долгого привыкания. Это является достоинством, потому что позволяет быстро перейти от изучения к творческому процессу. Гибкость в создании проекта позволяет охватить почти любые платформы и механики, а технология Blueprints (построение игровой логики с помощью наглядных схем) позволяет создавать игры без знания языков программирования. На рисунке 1.6 представлен интерфейс Unreal Engine [9].

Рисунок *1.6 –* Интерфейс игрового движка Unreal Engine [9]

В таблице 1.3 приведены достоинства и ограничения Unreal Engine.

16

Таблица *1.3 –* Достоинства и ограничения Unreal Engine Достоинства Ограничения

– бесплатен для некоммерческого – большой размер, более чем в

использования. отчислять 5 % при прибыли в 100 000 $;

Необходимо достижении

20 гигабайт;

– загрузка Windows-версии привязана к магазину.

– возможность работы на Windows и Linux-подобных системах;

– игровые ресурсы и библиотеки доступны в огромном количестве;

– практически полное отсутствие ошибок.

*Cryengine.*

Интерфейс скучен и невыразителен, что больше подходит закрытым движкам внутренних студий, а не массовому продукту. Однако Cryengine является логическим продуктом, что позволяет освоить движок за несколько месяцев. Несмотря на позиционирование движка как универсального, остаётся лучшим средством для создания шутеров, для иных целей пригоден мало. Знание языка программирования С++ необходимо. На рисунке 1.7 представлен интерфейс Cryengine [*9*].

Рисунок *1.7 –* Интерфейс игрового движка Cryengine [9]

17



В таблице 1.4 приведены достоинства и ограничения Unreal Engine.

Таблица *1.4 –* Достоинства и ограничения Cryengine Достоинства Ограничения

– бесплатен для некоммерческого – плохая служба поддержки;

использования, просит отчисление в 5 % от прибыли более 5 000 $;

– отсутствие нормальной документации;

– лучшая графика на рынке; – нераспространённость игровых – малый размер. ассетов и полноценных гайдов от

комьюнити;

– малопригоден для чего-то, кроме шутеров.

*Unity 3D.*

Несмотря на серость, интерфейс очень прост и понятен. Разработка на движке не требует знаний в программировании: для Unity существуют дополнения, позволяющие задействовать аналоги Blueprints (построение игровой логики с помощью наглядных схем). На рисунке 1.8 представлен интерфейс Unity [9].

Рисунок *1.8 –* Интерфейс игрового движка Unity 3D [9]

18

В таблице 1.5 приведены достоинства и ограничения Unity 3D.

Таблица *1.5 –* Достоинства и ограничения Unity 3D

Достоинства Ограничения

– бесплатен для некоммерческого – плохая производительность использования; готовых проектов;

– простота и удобство; – низкий уровень графики по – много материалов и ассетов; сравнению с конкурентами.

– отличная служба поддержки;

– кроссплатформенность.

Исходя из анализа вышеперечисленных достоинств и ограничений, в качестве игрового движка был выбран Unity3D.

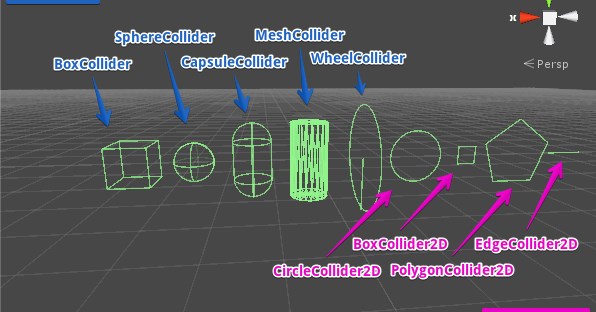
Созданные с помощью Unity приложения работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровыхприставках Wii, PlayStation3, PlayStation4, PlayStation5, Xbox360, Xbox One и MotionParallax, 3D-дисплеях (устройства для воспроизведения виртуальных голограмм), например, Nettlebox [10].

Приложения, созданные с помощью Unity, поддерживают DirectX и OpenGL. Активно движок используется как крупными разработчиками (Blizzard, EA, QuartSoft, Ubisoft), так и разработчиками Indie-игр (например, ремейк Мор. Утопия (Pathologic), Kerbal Space Program, Slender: The Eight Pages, Slender: The Arrival, Surgeon Simulator 2013, Baeklyse Apps: Guess the actor и т.д.) в силу наличия бесплатной версии, удобного интерфейса и простоты работы с движком.

Редактор Unity имеет простой Drag&Drop интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два сценарных языка: C#, JavaScript (модификация). Расчёты физики производит физический движок PhysX от NVIDIA [10].

Проект в Unity делится на сцены (уровни) – отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты – объекты, которые не имеют модели («пустышки»). Также у объектов есть название (в Unity допускается наличие двух и более объектов с одинаковыми названиями), может быть тег (метка) и

19

слой, на котором он должен отображаться. Так, у любого объекта на сцене обязательно присутствует компонент Transform– он хранит в себе координаты местоположения, поворота и размеров объекта по всем трём осям. У объектов с видимой геометрией также по умолчанию присутствует компонент Mesh Renderer, делающий модель объекта видимой [10].

К объектам можно применять коллизии (в Unity так называемые коллайдеры – collider). Существует несколько типов коллайдеров:

character controller – вид физической модели, созданный специально под использование его для игровых персонажей;

box collider (физическая модель образует куб, в который попадает вся модель объекта);

sphere collider (физическая модель образует сферу, в которую попадает вся модель объекта);

capsule collider (физическая модель образует капсулу, в которую попадает модель объекта. В отличие от предыдущего типа размеры можно менять и по одной, и по трем осям сразу);

mesh collider (физическая модель полностью повторяет реальную геометрию объекта);

wheel collider (физическая модель колеса); Типы коллайдеров представлены на рисунке 1.6.

Рисунок *1.6 –* Типы коллайдеров [11]

20

Также Unity поддерживает физику твёрдых тел и ткани, а также физику типа Ragdoll (тряпичная кукла). В редакторе имеется система наследования объектов; дочерние объекты будут повторять все изменения позиции, поворота и масштаба родительского объекта. Скрипты в редакторе прикрепляются к объектам в виде отдельных компонентов [10].

При импорте текстуры в Unity можно сгенерировать alpha-канал, mip-уровни, normal-map, light-map, карту отражений, однако непосредственно на модель текстуру прикрепить нельзя – будет создан материал, которому будет назначен шейдер, и затем материал прикрепится к модели. Редактор Unity поддерживает написание и редактирование шейдеров. Редактор Unity имеет компонент для создания анимации, но также анимацию можно создать предварительно в 3D-редакторе и импортировать вместе с моделью, а затем разбивать на файлы.

Unity 3D поддерживает систему Level Of Detail (сокр. LOD), суть которой заключается в том, что на дальнем расстоянии от игрока высокодетализированные модели заменяются на менее детализированные, и наоборот, а также систему Occlusion culling, суть которой в том, что у объектов, не попадающихв поле зрения камеры не визуализируется геометрия и коллизия, что снижает нагрузку на центральный процессор и позволяет оптимизировать проект [10].

Движок поддерживает множество популярных форматов, таких как: 3ds, .max, .obj, .fbx, .dae, .ma, .mb, .blend для трёхмерных моделей; .mp3, .ogg, .aiff, .wav, .mod, .it, .sm3 для звуковых файлов;

.psd, .jpg, .png, .gif, .bmp, .tga, .tiff, .iff, .pict, .dds для изображений; .mov, .avi, .asf, .mpg, .mpeg, .mp4 для видеофайлов;

.txt, .htm, .html, .xml, .bytes для текста.

Модели, звуки, текстуры, материалы, скрипты можно запаковывать в формат. unityassets и передавать другим разработчикам, или выкладывать в свободный доступ. Этот же формат используется во внутреннем магазине Unity Asset Store, в котором разработчики могут бесплатно и за деньги выкладывать в общий доступ различные элементы, нужные при создании игр. Чтобы использовать Unity Asset Store, необходимо иметь аккаунт разработчика Unity. Unity имеет все нужные компоненты для создания мультиплеера [10].

Исходя из того, что в вышеизложенном пункте выбрана среда разработки Unity 3D, языком программирования был выбран C#.

C# – объектно-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее

21

близок к C++ и Java. 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 [12].

**1.3 Выводы и постановка задач на дипломное проектирование**

Целью дипломного проекта является разработка мобильного приложения для 3D-игры на Unity, предназначенное для развлекательного времяпрепровождения людей независимо от пола или возраста.

Визуальная составляющая будет выполнена при помощи 2D- и 3D-графики с игровой камерой, динамически перемещающейся по игровому пространству. Будет присутствовать развитие игрока и другие геймплейные особенности.

Геймлей разрабатываемого проектабудетоснован на увеличении дохода от города при помощи покупки новых домов.

На основании рассмотренных выше аналогов следует, что разработка программного продукта является оптимальной т.к. разрабатываемый продукт будет обходить по показателям конкурирующие системы. На этом же основании можно сделать выводы о том, на что именно стоит обратить внимание при создании мобильного приложения компьютерной игры.

Проведя сравнительный анализ трёх игровых проектов, можно составить основные требования, которые должны быть реализованы в разрабатываемом мобильном приложении компьютерной игры:

использовать смешанные визуальные стили. Игра должна комбинировать различные типы графики;

грамотно подобрать цветовую палитру;

разработать простой и понятный интерфейс;

адаптировать интерфейсы к различным разрешениям экранов;

оптимизироватьигруктакимпоказателям, которыепозволятохватить максимально возможный список устройств;

добавить контента на 50 часов игрового процесса, что позволит добиться высокого удержания пользователей;

сделать возможным запуск игры на iOS и Android платформах; протестировать и исправить найденные дефекты;

внедрить в игру рекламу.

22

В дипломном проекте уделяется внимание вопросу об обеспечении безопасных условий труда разработчика информационных систем по средствам выбора и обоснования комплекса мер по защите от статического электричества.

В процессе дипломного проектирования требуется разработать мобильное приложение для 3D-игры.

Программное средство должно быть кроссплатформенным, функционировать на iOS и Android платформах.

Исходя из требования о кроссплатформенности, в качестве среды разработки было решено выбрать игровой движок Unity 3D, который на данный момент является одним из самых популярных игровых движков. В Unity присутствуют визуальная среда разработки, межплатформенная поддержка, модульная система компонентов и возможность создания анимации.

Из-за специфики работы в игровом движке Unity 3D, в качестве языка программирования был выбран C#.

Разрабатываемое мобильное приложение должно быть устойчиво к аппаратным и программным сбоям, быть защищено от неправильных действий пользователей.

Преимуществом приложения можно считать использование кроссплатформенного решения для реализации игр, что значительно сокращает сроки и затраты на разработку. Это позволит охватить как можно большую часть аудитории рынка мобильных игр.

23

**2 ИЖЕНЕРНО-ПСИХОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ 3D-ИГРЫ НА UNITY**

В данном разделе дипломного проекта будет выполнятся эргономическое проектирование одного из видов современных систем – системы «человек – мобильное устройство – среда» (СЧМС).

**2.1 Анализ функций и их распределение в проектируемой системе**

Проектируемая система должна обеспечивать выполнение следующих функций:

1 Запуск игры.

2 Просмотр уровня команды. 3 Получение игровой валюты.

4 Просмотр полученной игровой валюты. 5 Включение и выключение звуков в игре. 6 Улучшение команды игрока.

7 Улучшение ёмкости баллончика с краской. 8 Улучшение скорости бега художников.

9 Улучшения скорости покраски граффити. 10 Просмотр нарисованных граффити.

11 Просмотр карты города.

12 Покупка заблокированных домов. 13 Улучшение домов города.

14 Улучшение скорости получения дохода. 15 Улучшение дохода от дома.

16 Сохранение прогресса игры. 17 Загрузка прогресса игры.

18 Выход из приложения.

Содержание функций СЧМС, указанных в пунктах 1–18, можно описать следующим образом:

1 Запуск игры осуществляется путем нажатия на иконку игры на рабочем столе.

2 Текущий уровень команды игрока отображается на верхнее панели меню.

3 Каждый нарисованный рисунок и каждый купленный дом приносят игроку внутриигровую валюту.

24

4 Текущее количество игровой валюты игрока отображается на верхней панели меню.

5 Включение и выключение звуков в игре осуществляется в меню настроек.

6 Улучшение команды игрока осуществляется путём увеличения количества художников.

7 Ёмкость баллончика с краской увеличивается за внутриигровую валюту в меню изменения параметров художника.

8 Скорость бега художников увеличивается за внутриигровую валюту валюту в меню изменения параметров художника

9 Скорость покраски граффити увеличивается за внутриигровую валюту валюту в меню изменения параметров художника.

10 Просмотр нарисованных граффити осуществляется при помощи перехода на экран дома через специальное меню.

11 Просмотр карты города осуществляется на главном экране игры.

12 Покупка заблокированного дома осуществляется в меню дома за внутриигровую валюту.

13 Параметры дома изменяются в специальном меню за внутриигровую валюту.

14 Скорость получения дохода в игре увеличивается при помощи изменения параметра в специальном меню.

15 Доход в игре увеличивается при помощи изменения параметра в специальном меню.

16 Сохранение прогресса игры осуществляется автоматически в процессе выхода из игры.

17 Загрузка прогресса игры осуществляется во время загрузки игры. 18 Выход из приложения осуществляется в главном меню игры.

Создание эффективного мобильного приложения компьютерной игры заключается в поиске оптимального сочетания возможностей машины и человека.

На человека следует возлагать выполнение функций по:

1 Распознаванию ситуации в целом по её многим сложно связанным характеристикам, а также при неполной информации о ней.

2 Осуществлению функций индуктивного вывода, т.е. обобщению отдельных фактов в единую систему.

3 Решению задач, в которых отсутствует единый алгоритм или нет чётко определённых правил обработки информации.

25

4 Решению задач, в которых требуется гибкость и приспособляемость к изменяющимся условиям, особенно задач, появление которых заранее трудно предвидеть.

5 Решению задач с высокой ответственностью в случае возникновения ошибки [13].

Машине следует поручать:

1 Выполнение всех видов математических расчётов.

2 Выполнение однообразных, постоянно повторяющихся операций, реализуемых по заданному алгоритму.

3 Хранение и динамическое представление больших объёмов однородной информации.

4 Решение задач, требующих дедуктивного вывода, т.е. получения на основе общих правил решений для частных случаев.

5 Выполнение действий, требующих высокой скорости реакции на команду [13].

Учитывая вышеперечисленные пункты, проведём анализ функций разрабатываемого мобильного приложения компьютерной игры с целью распределения ихмеждучеловеком и компьютером. Результаты представлены в таблице 2.1.

Таблица *2.1* – Распределение функций между пользователями и мобильным устройством

Название функции

1

1 Запуск игры

2 Просмотр уровня команды

3 Получение игровой валюты 4 Просмотр

полученой игровой валюты

5 Включение и выключение звуков в игре

Исполнитель функции в СЧМС 2 Человек

(пользователь)

Человек (пользователь)

Человек (пользователь) Человек (пользователь)

Человек (пользователь)

Обоснование распределения функций

3

Информацию вводит в систему человек, компьютер хранит введённые данные

Информацию вводит в систему человек, компьютер хранит введённые данные

Функция остаётся за человеком в соответствии с назначением системы Информацию вводит в систему человек, компьютер хранит введённые данные

Функция остаётся за человеком в соответствии с назначением системы

26

Продолжение таблицы *2.1*

1 2

6 Улучшение Человек команды игрока (пользователь) 7 Улучшение Человек емкости (пользователь) баллончика с

краской

8 Улучшение Человек скорости бега (пользователь) художнмков

9 Улучшения Человек скорости покраски (пользователь) граффити

10 Просмотр Человек нарисованных (пользователь) граффити

3

Функция остаётся за человеком в соответствии с назначением системы Функция остаётся за человеком в соответствии с назначением системы

Функция остаётся за человеком в соответствии с назначением системы

Функция остаётся за человеком в соответствии с назначением системы

Информацию вводит в систему человек, компьютер хранит введённые данные

11 Просмотр карты города

Человек (пользователь)

Информацию вводит в человек, компьютер введённые данные

систему хранит

12 Покупка заблокированных домов

13 Улучшение домов города

14 Улучшение скорости получения дохода

15 Улучшение дохода от дома

16 Сохранение прогресса игры

17 Загрузка прогресса игры 18 Выход из приложения

Человек (пользователь)

Человек (пользователь)

Человек (пользователь)

Человек (пользователь) МУ

МУ

Человек (пользователь)

Информацию вводит в систему человек, компьютер хранит введённые данные

Информацию вводит в систему человек, компьютер хранит введённые данные

Функция остаётся за человеком в соответствии с назначением системы

Функция остаётся за человеком в соответствии с назначением системы МУ это делает надёжнее и может сохранять результат в течении неограниченного времени

МУ это делает надёжнее и быстрее

Информацию вводит в систему человек, компьютер обрабатывает данные

В результате проведённого анализа и распределения функций в проектируемой СЧМС можно сделать следующий вывод: проектируемая

27

система должна состоять из одной подсистемы, которая является моносистемой «пользователь – мобильное устройство – среда», т.е. в ее сосав входит только один пользователь.

**2.2 Проектирование деятельности человека в системе «человек-мобильное устройство-среда»**

Система «человек – мобильное устройство – среда» состоит из трёх основных частей:

1 Человек-оператор, который может кем угодно, так как для разрабатываемой мобильный игры нет определённой целевой аудитории. Человек, воздействуя на техническое звено системы, управляет игровым процессом. На работу человека-оператора влияют его знания, опыт, психические и физиологические особенности.

2 Техническое звено системы включает программное и аппаратное обеспечение. К аппаратному обеспечению относится мобильное устройство. Программное обеспечение представляет собой совокупность операционной системы и прикладной программы.

3 Рабочая среда рабочего места человека-оператора включает такие факторы, как освещённость, аэроионный состав воздуха, вибрация, электромагнитное излучение, диапазон станции оператора мобильной связи. Если не ставится цель изучить работу человека-оператора в экстремальных условиях, то все факторы необходимо привести к оптимальному значению. В иных случаях можно варьировать значения необходимых параметров, не допуская при этом причинения ущерба здоровью работающего [14].

Если хотя бы один из компонентов СЧТС находится в состоянии, не удовлетворяющем условиям техники безопасности, эксперимент должен быть отложен до устранения негативно влияющих факторов.

Программа должна являться легко расширяемой в целях добавления новых функций, написана на языке, обеспечивающем достаточное быстродействие для эксплуатации пользователем. Она не должна содержать грамматических ошибок, неправильных функциональных переходов, а также иметь защиту от неправильных действий пользователя, таких, как например ввод некорректной информации или не преднамеренное удаление данных.

Для данной системы необходимо разработать алгоритмы работы человека в системе «человек – мобильное устройство – среда», представленные в таблицах 2.2, 2.3, 2.4 и 2.5. В процессе реализации названного алгоритма некоторые операции могут выполняться в другой последовательности или пропускаться.

28

Деятельность оператора в системе зависит от прав доступа, которыми он наделён (администратор, пользователь) и вводом пароля. Администратор вводит свои данные в форму авторизации и перенаправляется в режим работы с системой и её модулями. Пользователь вводит свои данные в форму авторизации и перенаправляется на страницу со списком доступных для покупки товаров. В личном профиле он имеет возможность просматривать всю информацию относительно себя самого, хранящуюся в базе данных и изменять её.

В разрабатываемой системе только один пользователь. Алгоритмы системы «человек – мобильное устройство – среда» представлены в таблице 2.2, 2.3, 2.4 и 2.5. В процессе реализации названных алгоритмов некоторые операции могут выполняться в другой последовательности или пропускаться.

Таблица *2.2* – Алгоритм прохождения уровня игры в подсистеме «человек – мобильное устройство – среда»

Содержание операции 1

1 Включение мобильного устройства

2 Запуск приложения

Обращение к средствам отображения информации 2

Приветственное сообщение или отображение экрана блокировки устройства Ярлык приложения на рабочем столе или в меню программ

Обращение к органам управления 3

Нажатие кнопки питания на корпусе мобильного устройства

Нажатие пальца на ярлык

3 Увеличение ёмкости баллончика

4 Увеличение скорости бега художника

Кнопка улучшения емкости баллончика на нижней панели экрана

Кнопка увеличения скорости бега на нижней панели экрана

Нажатие кнопку

Нажатие кнопку

пальца на

пальца на

5 Увеличение Кнопка увеличения Нажатие пальца на скорости рисования скорости рисования на кнопку

художника нижней панели экрана

6 Увеличение Кнопка улучшения Нажатие пальца на количества количества художников на кнопку

художников верхней панели экрана

*Примечание* *–* Операции 3–6 могут выполняться в другой последовательности или пропускаться.

7 Выход из Кнопка «Выйти» на экране Нажатие пальца на приложения дисплея кнопку «Выйти»

29

Таблица *2.3* – Алгоритм просмотра пройденных уровней в подсистеме «человек – мобильное устройство – среда»

Содержание операции 1

1 Включение мобильного устройства

2 Запуск

приложения

Обращение к средствам отображения информации 2

Приветственное сообщение или отображение экрана блокировки устройства Ярлык приложения на рабочем столе или в меню

программ

Обращение к органам управления 3

Нажатие кнопки питания на корпусе мобильного устройства

Нажатие пальца на ярлык

3 Выход на главный экран игры

Кнопка выхода на главный экран игры расположенная в

левой части экрана

Нажатие пальцем на кнопку

4 Просмотр Элементы уровней Нажатие пальцем на пройденного расположенный в 3D элемент уровня

уровня пространстве на главном экране игры

5 Выход из

приложения

Кнопка «Выйти» на экране

дисплея

Нажатие пальца на

кнопку «Выйти»

Таблица *2.4* – Алгоритм покупки нового дома в подсистеме «человек – мобильное устройство – среда»

Содержание операции 1

1 Включение мобильного устройства

2 Запуск

приложения

Обращение к средствам отображения информации 2

Приветственное сообщение или отображение экрана блокировки устройства Ярлык приложения на рабочем столе или в меню

программ.

Обращение к органам управления 3

Нажатие кнопки питания на корпусе мобильного устройства

Нажатие пальца на ярлык

3 Открытие меню Кнопка открытия меню Нажатие пальцем на дома дома расположена в 3D кнопку

пространстве игре.

30

Продолжение таблицы *2.4*

1 2 3

4 Покупка дома Кнопка покупки дома Нажатие пальцем на расположена в кнопку.

открывшемся меню дома.

5 Назначение Ползунок для назначения Перетягивание пальцем

художников для покраски дома

художников расположен в ползунка. меню дома.

6 Переход к просмотру дома

Кнопка просмотру

перехода к дома

Нажатие пальцем на кнопку.

расположена в меню игры.

Таблица *2.5* – Алгоритм увеличения параметров художника в подсистеме «человек – мобильное устройство – среда»

Содержание операции 1

1 Включение мобильного устройства

2 Запуск приложения

3 Переход в меню художников

4 Отображение меню художника

4 Увеличение параметра

5 Изменение значения игровой валюты

Обращение к средствам отображения информации 2

Приветственное сообщение или отображение экрана блокировки устройства Ярлык приложения на рабочем столе или в меню программ.

Кнопка по середине в нижней части экрана Отображение меню художника на экране устройства.

Параметры художников расположены на всплывающем меню Изменение значения игровой валюты, отображаемой на экране

устройства

Обращение к органам управления 3

Нажатие кнопки питания на корпусе мобильного устройства

Нажатие пальцем на ярлык

Нажатие пальцем на кнопку

Обращение к органам управления не происходит, так как операция выполняется автоматически Нажатие пальцем на кнопку увеличения параметра

Обращение к органам управления не происходит, так как операция выполняется

автоматически

31

**2.3 Разработка эргономических требований и сценария информационного взаимодействия**

Эффективность функционирования проектируемой системы определяется взаимовлияющими показателями работы технического звена и работоспособности человека-оператора, а также эффективностью взаимодействия человека и мобильного устройства в процессе функционирования системы [14].

Факторы, определяющие эффективность функционирования системы «человек – мобильное устройство – среда» можно разделить на несколько групп:

1 Антропометрические, которые обусловлены анатомическими, морфологическими и биомеханическими характеристиками конкретного пользователя: размером, формой и весом человеческого тела и его частей (рук, ног, головы, туловища) в статике и динамике, углами поворота рук, ног и туловища, амплитудами различных движений, траекториями движений и т.п.. 2 Психофизиологические, которые обусловлены возможностями и особенностями органов чувств (сенсорных систем) человека. Они представляют собой значения порогов чувствительности различных анализаторов человека: зрения, слуха, осязания, кожной чувствительности и других, а также учёт особенностей различных явлений и эффектов, связанных с функционированием анализаторов человека: синестезии, адаптации,

взаимодействия ощущений и др.

3 Физиологические, которые учитывают энергетические возможности организма человека по реализации физических качеств силы, скорости, выносливости движений при эксплуатации техники. Многие из требований этой группы составлены на основе принципа экономии движений, т. е. их оптимального характера, последовательности, темпа и ритма рабочих движений.

4 Психологические, которые определяют соответствие СЧТС и её элементов психологическим особенностям человека. К ним относятся особенности восприятия, памяти, мышления, психомоторики человека. Психологические требования учитывают возможности человека в его информационном взаимодействии с техникой, определяющие процессы приёма и переработки информации, выполнение управляющих действий в СЧТС. Кроме этого, они учитывают влияние характеристик технического звена системы на лёгкость и быстроту формирования навыков человека,

32

закрепление у него вновь приобретённых навыков, их соответствие существующим стереотипам [14].

5 Гигиенические факторы, которые определяют безвредные и безопасные условия жизнедеятельности человека, обусловливают роль среды в СЧМС. Эти требования составляют на основе санитарно-гигиенических рекомендаций и нормативов к параметрам рабочей среды. Они обеспечивают соблюдение норм микроклимата, шума, освещённости и ограничивают воздействие других вредных и опасных факторов производственной среды на человека-оператора [15].

В случае разрабатываемой системы, наибольшее значение имеют психологические, психофизиологические и антропометрические факторы, т.к. пользователю предстоит работать с большим объёмом информации и возможно, вне оборудованного рабочего места.

Методы предъявления данной информации должны наилучшим образом соответствовать предъявляемым требованиям, иначе существует вероятность утомления зрительной системы, совершения ошибок и даже причинение вреда пользователю.

Номенклатура указанных выше факторов приведена в таблице 2.6. Следующим этапом является обоснование и разработка эргономических требований к проектируемой СЧМС. Эргономические требования должны предъявляться как к свойствам технической системы и к различным её элементам, так и к человеку как оператору, включённому в эргономическую систему, создаваемую на основе исходной технической системы.

Таблица *2.6* – Номенклатура эргономических требований (ЭТ) определяющих эффективность функционирования СЧМС [14]

Группа ЭТ 1

Психофизиологические

Номенклатура ЭТ группы 2

соответствие размеров знаков на экране дисплея оперативному порогу зрения человека;

соответствие контраста знаков и фона оптимальным условиям восприятия;

соответствие вида контраста знаков и фона

уровню освещённости рабочего места.

33

Продолжение таблицы *2.6*

1 2

Психофизиологические

Физиологические

Психологические

соответствие расположения надписей условиям их оптимального считывания;

отсутствие требований, несоответствующих возможностям органов чувств человека.

соответствие компоновки ОУ принципам экономии рабочих движений;

соответствие усилий на ОУ силовым возможностям человека;

соответствие требований к скорости движений возможностям человека;

отсутствие требований, связанных с обслуживанием системы, несоответствующих силовым и скоростным возможностям человека;

отсутствие требований, связанных с освоением системы, несоответствующих силовым и скоростным возможностям человека.

соответствие цветов надписей и знаков стереотипам восприятия;

соответствие объёмов информации, требующей запоминания, возможностям памяти человека;

отсутствие неоднозначного толкования требований, инструкций и команд;

соответствие компоновки ОУ и СОИ стереотипам восприятия;

соответствие индикации срабатывания ОУ сформированным навыкам, наличие индикации

хода выполнения функции;

34

Продолжение таблицы *2.6*

1 2

Психологические

Гигиенические

Антропометрические

соответствие количества одновременно предъявляемых сигналов возможностям внимания человека.

соответствие сложности инструкций, времени, отводимому на их усвоение;

один и тот же характер команд на протяжении всего периода работы в системе в схожих ситуациях;

наличие указаний на проблемы, возникающие в процессе обслуживания системы;

наличие подсказок о следующих шагах работы в системе;

наличие предупреждений о нежелательных последствиях некоторых действий.

соответствие параметров рабочей среды гигиеническим нормативам [15].

соответствие размеров зон управления и обслуживания антропометрическим характеристикам человека;

соответствие размеров рабочего стола антропометрическим характеристикам человека;

соответствие размеров рабочего кресла антропометрическим характеристикам человека;

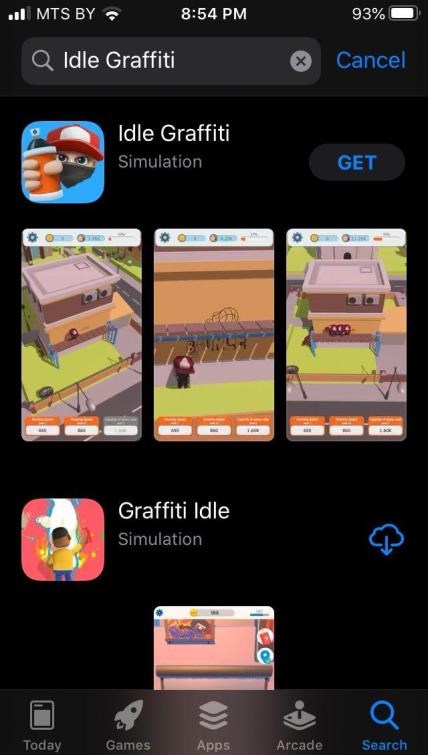
соответствие размеров зон обслуживания антропометрическим характеристикам

человека.

Рассмотренные эргономические требования предъявляют к различным элементам СЧМС: рабочим местам операторов, пультам управления, органам управления и индикации, системам отображения и ввода информации, эксплуатационной документации.

Эргономические требования к СЧМС – это требования к системе в целом, её отдельным подсистемам, оборудованию, рабочей среде, определяемые

35

свойствами человека и устанавливаемые для обеспечения его эффективной и безопасной деятельности. Они формируются на основании экспериментальных исследований и опыта эксплуатации СЧМС, требований эргономических стандартов [14].

В результате эргономического проектирования был разработан пользовательский интерфейс, который удовлетворяет условиям технического задания.

Работа начинается с установки мобильного приложения компьютерной игры. Для установки программного средства необходимо выполнить следующие действия:

1 Скачать клиентское приложение с репозитория Apple Store (рисунок 2.1).

2 Установка приложения начинается автоматически после загрузки программного средства на устройство под управлением ОС iOS.

После завершения установки приложения, на рабочем столе устройства появится значок игры (рисунок 2.2).

Рисунок *2.1* – Отображение игры в репозитории Apple Store

36



Рисунок *2.2* – Отображение игры на рабочем столе мобильного устройства

При открытии мобильной игры пользователь попадает на главный экран игры. На этом экране пользователь может управлять городом. Главный экран игры отображён на рисунке 2.3.

Рисунок *2.3* – Отображение игры на рабочем столе мобильного устройства

37

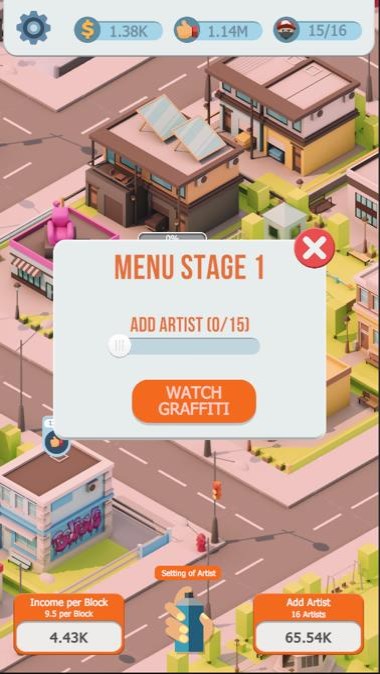
На главном экране игры происходит управление городом. Пользователь может покупать новые дома, улучшать параметры уже купленных домов и переходить к просмотру дома. На рисунке 2.4 отображён процесс покупки дома.

Рисунок *2.4* – Покупка заблокированного дома

После покупки дома, пользователю необходимо назначить художников, которые будут рисовать граффити на этом доме. Процесс назначения художников отображён на рисунке 2.5.

Рисунок *2.5* – Назначение художников для рисования граффити

38

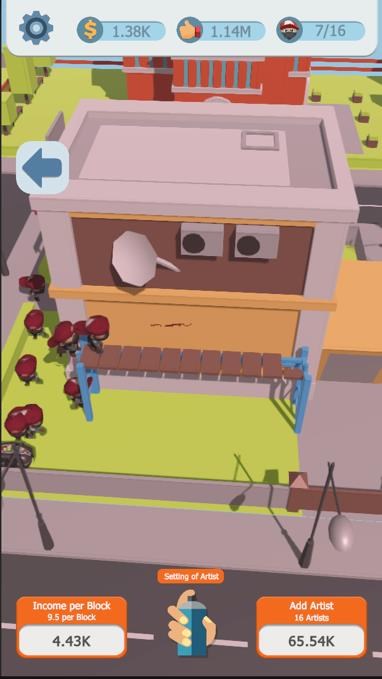
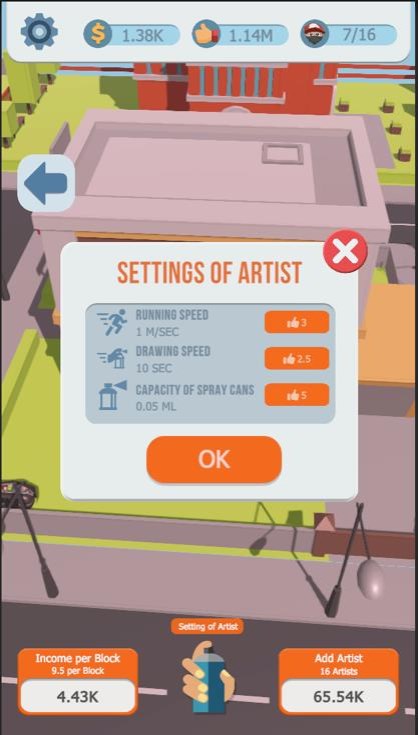
После того как пользователь назначит художников, на экране отобразится процесс рисования граффити. Процесс рисования граффити отображён на рисунке 2.6.

Рисунок *2.6* – Процесс рисования граффити

Улучшая характеристики художников, процесс рисования граффити происходит быстрее. Игрок может улучшить параметры такие как скорость бега, ёмкость баллончика и скорость рисования граффити. Также игрок может нанять нового художника в команду. На рисунке 2.7 отображён процесс улучшения команды художников.

Рисунок *2.7* – Процесс улучшения команды художников

39

Время рисования граффити зависит от уровня команды и количества художников участвующих в рисовании. После завершения рисования граффити, игроку отображается итоговый результат после чего он может перейти на новый уровень. Результат нарисованного граффити отображён на рисунке 2.8.

Рисунок *2.8* – Результат нарисованного граффити

После того как пользователь разблокирует дом, появляется возможность улучшения параметров домов. Пользователь может улучшать скорость дохода и количество дохода за определённое время.

Улучшая параметры дома, пользователь увеличивает доход от дома. Процесс улучшения параметров дома отображён на рисунке 2.9.

Рисунок *2.9* – Процесс улучшения параметров дома

40

Соответствиесистемыкаждомуотдельномуэргономическомутребованию определяет единичный эргономический показатель её качества, соответствие множеству требований той или иной группы определяет групповой эргономический показатель качества СЧМС (гигиенический, антропометрический, физиологический, психофизиологический, психологический, социально-психологический).

Групповые эргономические показатели определяют комплексные эргономические показатели, которые в совокупности обуславливают эргономические свойства системы, в частности речь идёт о следующих:

управляемость, то есть обеспечение возможности оптимальных условий, для высокого уровня контроля над системой конкретным пользователем;

обслуживаемость, тоестьобеспечениеоптимальнойструктурысистемы, сцельюпредоставленияконкретномупользователювозможности качественного обслуживания, эксплуатации и ремонта системы;

усвояемость, то есть обеспечение оптимальной структуры системы для предоставления возможности быстрого её освоения, высокого уровня обучаемости функциям системы и приобретении навыков и умений, связанных с ней;

обитаемость, то есть обеспечение оптимальной структуры системы для отсутствия вредоносных факторов, связанных с работой системы, способных привести к нежелательным воздействиям на здоровье пользователя [14].

Дальнейший анализ представляет собой выделение единичных эргономических показателей, имеющих непосредственное отношение к каждой из названных выше групп показателей. Результаты анализа представлены в таблице 2.7.

Таблица *2.7* – Единичные эргономические показатели проектируемой СЧМС

Группа Эргономические требования

1 2

ПФ-1. Соответствие размеров знаков на Психо- экране дисплея оперативному порогу физио- зрения человека

логи- ПФ-2. Соответствие контраста знаков и ческие фона оптимальным условиям

восприятия

Единичные эргономические показатели 3

Размеры шрифта текста и знаков

Величина контраста знаков и фона

41

Продолжение таблицы *2.7*

1 2

ПФ-3.Соответствие вида контраста знаков и фона уровню освещенности рабочего места

ПФ-4. Скрывание недоступные пунктов Психо- меню

физио- ПФ-5. Соответствие расположения логи- надписей условиям их оптимального

ческие считывания

ПФ-6. Использование пролистываемых и раскрывающихся списков в целях экономии экранного пространства

П-1. Соответствие сложности инструкций, времени, отводимому на их восприятие

П-2. Один и тот же характер команд на протяжении всего периода работы в системе в схожих ситуациях

П-3. Наличие указаний на проблемы, возникающие в процессе обслуживания системы

П-4. Наличие подсказок о следующих шагах работы в системе

П-5. Наличие предупреждений о нежелательных последствиях

Психоло-

гические некоторых действий

П-6. Соответствие цветов знаков и надписей сформированным стереотипам восприятия цвета

П-7. Соответствие формы и расположения знаков сформированным стереотипам восприятия

П-8. Выделение в текстовых инструкциях смысловых фрагментов

П-9. Отсутствие в текстовых сообщениях аббревиатур, непонятных слов и сокращений

3

Вид контраста знаков и фона

Содержание меню

Расположение и ориентация надписей на экране дисплея Наличие и типы пролистываемых и раскрывающихся списков

Длина инструкции и время ее экспозиции

Тип ОУ и их обозначение

Сообщения об ошибочных действиях пользователей Сообщения о следующих действиях пользователей Предупреждения о возможных нежелательных действиях

Цвета знаков, кнопок, надписей

Форма и ориентация знаков

Компоновка текста инструкции (наличие абзацев)

Словарный состав текстовых инструкций

42

Продолжение таблицы *2.7*

1 2

П-10. Привлечение внимания пользователей к важным сообщениям

Психоло-

гические П-11. Наличие индикатора степени выполнения заданий (операций)

П-12. Наличие кратких и понятных заголовков окон

П-13. Использование для названий пунктов меню одного слова (глагола для действий, существительного для объектов)

3 Используемые средства привлечения внимания пользователя (цвет, мигание, звуковые сигналы)

Наличие и вид индикатора выполнения Наличие и вид заголовков окон Названия пунктов меню

Психоло-гические

П-14. Применение в названиях пунктов меню норм использования заглавных букв, принятых в языке

П-15. Соответствие опций элементов интерфейса установленным, привычным нормам (например, использование клавиши Enter)

П-16. Отсутствие у пользователей сложностей в поиске необходимых директив (элементов интерфейса) для управления процессом решения поставленной задачи

П-17. Сообщение об ошибке должно отвечать всего на три вопроса:

в чем заключается проблема?

как исправить эту проблему сейчас?

как сделать так, чтобы проблема не повторилась?

П-18. Вежливое и понятное пользователю сообщение об ошибках П-19. К строкам ввода там, где это возможно, с целью разгрузки памяти целесообразно присоединять выпадающий список допустимых значений

Названия пунктов меню

Соответствие привычным нормам

Естественность взаимодействия

Содержание сообщений об ошибках

Содержание сообщений об ошибках

Наличие выпадающих списков допустимых значений в строках ввода

43

Продолжение таблицы *2.7*

1 2

Психоло-

гические

П-20. Целесообразно использовать в рамках одного приложения окна, построенные по одному шаблону, в которых одинаковые элементы расположены одинаково

П-21. Интерфейсные элементы должны иметь не только согласованные изображения, но и согласованное управление. Например, активизация всех пиктограмм – двойным щелчком мыши

П-22. Следует учитывать при проектировании меню и диалоговых

Психоло- окон стереотипную логическую гические последовательность чтения текста

справа налево и сверху вниз. В левом верхнем углу следует располагать элемент, с которым пользователь должен работать в первую очередь, а в правом нижнем углу – тот, который используется в конце. Не следует первым элементом меню ставить опцию «Выход»

Ф-1. Соответствие размеров элементов

физиологическим возможностям движений

Ф-2. Использование в группе радиокнопок не менее одной с

режимом по умолчанию

3

Окна интерфейса в программе

Средства управления элементами интерфейса

Компоновка опций меню и диалоговых окон

Размеры меню, списков, кнопок на экране дисплея Наличие в группе радиокнопок не менее одной с режимом по

умолчанию

Физиоло-гические

Ф-3. Использование командных кнопок для ввода явных действий

Ф-4. Использование чекбоксов и радиокнопок для ввода параметров запускаемого впоследствии действия

Ф-5. Соответствие времени экспозиции списков, меню, кнопок скоростным

Наличие командных кнопок для ввода явных действий

Наличие чекбоксов для ввода параметров запускаемого впоследствии действия Длительность

экспозиции

44

Продолжение таблицы *2.7*

1 2 3

Физиоло-гические

Гигие-нические

Ф-6. Использование крутилок для ввода числовых значений

Ф-7. Использование значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации.

Ф-8. Отсутствие требований к пользователям вводить информацию, которая была предварительно введена или которая может быть автоматически получена из системы

Г-1. Соответствие параметров изображения на экране дисплея условиям комфорта зрительной работы пользователей (отсутствие мельканий, слепящих яркостей и т.п.) [15]

Наличие крутилок для ввода числовых значений Используемые значения по умолчанию

Отсутствие необходимости вводить информацию, которая может быть автоматически получена из системы Энергетические и временные параметры изображения на экране

дисплея

После построения таблицы проведём оценку значений единичных эргономических показателей. Указанные показатели оцениваются по бинарной шкале с числовым диапазоном от 0 до 1. Показатель принимает значение, равное 1, если фактическое его значение соответствует рекомендуемому. Если соответствия нет (т.е. требование фактически не выполняется), то показатель принимается равным 0.

Групповой эргономический показатель ЭПгр рассчитывается как общая оценка по группе единичных показателей:

ЭПгр = ∑1/(∑1 + ∑0) , (2.1)

где ∑ 1 – суммарное число случаев, когда единичный показатель соответствует эргономическому требованию;

∑ 0 – суммарное число случаев, когда соответствия нет.

Результаты оценки значений единичных и групповых эргономических показателей приведены в таблице 2.8.

45

Таблица *2.8* – Значения единичных и групповых эргономических показателей проектируемой системы

Группа ЭП

Психофизиологические

Психологические

Физиологические

Антропометрические

Значение единичных ЭП

ПФ-1, ПФ-2, ПФ-3, ПФ-5, ПФ-6 = 1 ПФ-4=0

П-1, П-3, П-4, П-5, П-6, П-7 = 1; П-2 = 0

Ф-1, Ф-2, Ф-3= 1

А-1 = 1

Значение групповых ЭП

5 / (5 + 1) = 0,83

6 / (6 + 1) = 0,86

3 / (3 + 0) = 1

1 / (1 + 0) = 1

Проведём оценку эргономических свойств проектируемой системе. Эргономические свойства системы определяются как некоторая совокупность групповых эргономических показателей, при этом чаще всего применяется аддитивная функция:

ЭСВ = ∑𝑎н𝑖 ∙ ЭПгр𝑖, (2.2)

где 𝛼Hj – нормированные весовые коэффициенты, сумма которых должна быть равна 1, т. е. ( ∑ 𝛼Hj = 1).

Для оцениваемого эргономического свойства выбираем величины весовых коэффициентов.

Результаты распределения представлены в таблице 2.9.

Таблица *2.9* – Значения весовых коэффициентов групп эргономических требований и эргономических свойств данной СЧМС

Группа эргономических требований Психофизиологическая Психологическая

Физиологическая

Антропометрическая

Значение весового коэффициента 0,2 0,6 0,1

0,1

На основе таблиц 2.8 и 2.9, используя формулу 2.2, определим количественное значение указанных выше эргономических свойств.

Получим:

ЭСВуправляемость = 0,2 ∙ 0,83 + 0,6 ∙ 0,86 + 0,1 ∙ 1 + 0,1 ∙ 1= 0,88

46

Показатель эргономичности разрабатываемой системы 0,88. Полученное значение группового эргономического показателя оценивается с учётом следующей градации:

0,8–1,0 – «отлично» – эргономические характеристики изделия соответствуют базовым значениям;

0,5–0,8 – «хорошо» – приближается к базовым, но требуется совершенствование изделия;

0,2–0,5 – «удовлетворительно» – далеки от базовых, требуется значительное улучшение изделия;

0–0,2 – «неудовлетворительно» – практически не обеспечивается необходимая производительность, удобство и безопасность труда человека – оператора.

Согласно расписанной выше системе оценивания, уровень эргономичности данной системы оценивается отметкой «отлично», что свидетельствует о тщательной разработке системы, комплексном подходе в ее создании и детально проработанного взаимодействия пользователя с приложением.

Однако так как показатель эргономичности не равен 1, нужно провести анализ единичных показателей, которые не соответствуют предъявляемым требованиям.

Результаты анализа представлены в таблице 2.10.

Таблица *2.10* – Рекомендации по улучшению эргономичности, проектируемой

системы Невыполненное эргономическое

требование

Предложение по улучшению эргономичности

ПФ-4. Соответствие количества Вместо аббревиатур и сокращений одновременно предъявляемых использовать полные версии слов и стимулов объёму восприятия названий

человека

П-2. Предъявление сообщений об Проработать и ввести в систему ошибочных действиях пользователя сообщения о возможных ошибках

пользователя

В результате эргономического проектирования был разработан пользовательский интерфейс, который удовлетворяет условиям технического задания.

47

При разработке дипломного проекта было проведено инженерно-психологическое проектирование мобильного приложения для 3D-игры в жанре «Idle»для операционной системы iOS на платформе Unity. Назначением разработки является обеспечение увлекательного времяпрепровождения для пользователя.

При проведении инженерно-психологического проектирования были решены следующие задачи:

1 Выполнен анализ и определено содержание функций.

2 Распределены функции между человеком и техникой системы.

3 Разработана структура СЧМС и структура программы, обеспечивающей функционирование технического звена системы.

4 Спроектирована деятельность человека в системе «человек-мобильное устройство-среда».

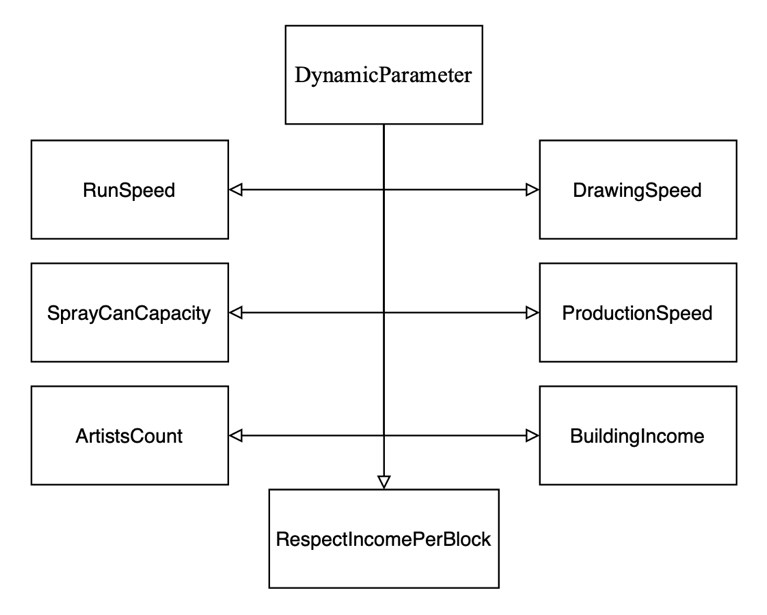
5 Проанализированы факторы, определяющие эффективность функционирования СЧМС.

6 Выполнено обоснование эргономических требований к системе.

7 Разработан сценарий информационного взаимодействия пользователя и мобильного устройства.

На основании полученной информации можно в дальнейшем приступать к программной и аппаратной реализации системы.

48

**3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ 3D-ИГРЫ НА UNITY И ЕГО ИСПЫТАНИЕ**

**3.1 Проектирование архитектуры и структуры мобильного приложения для 3D-игры**

Мобильное приложение для 3D-игры на Unity представляет собой мультиплатформенное приложение. Перед разработкой системы необходимо определить архитектуру программного продукта и структуру проекта.

Определим основные модули проекта: модуль динамических параметров;

модуль подготовки и создания игрового уровня; модуль получения дохода;

модуль игровых валют;

модуль домов, производящих доход; модуль художников;

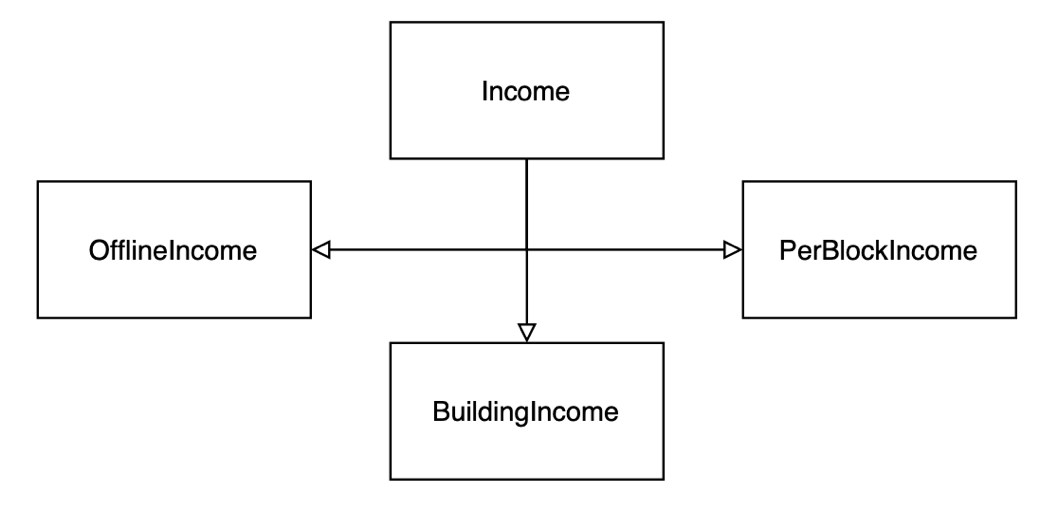
модуль пользовательского интерфейса.

Модуль динамических параметров предназначена для определения параметров, которые будут изменяться и улучшаться в процессе игры. Система будет использоваться для увеличения параметров художника и параметров дома.

Для обеспечения расширяемости, необходимо определить базовую сущность, которая будет описывать основную логику системы и быть функционально расширяемой. На рисунке 3.1 представлена архитектура модуля динамических параметров.

Рисунок *3.1* – Архитектура модуля динамических параметров

49

Модуль подготовки и создания игрового уровня предназначена для генерации уровней с разными режимами игры. В игре будет возможность генерировать уровень в режимах рисования граффити и просмотра нарисованных граффити.

Для обеспечения расширяемости было принято использовать паттерн «Абстрактная фабрика». На рисунке 3.2 представлена архитектура модуля подготовки и создания игрового уровня.

Рисунок *3.2* – Архитектура модуля подготовки и создания игровых уровней

Модуль получения дохода предназначена для начисления дохода игроку. Игрок может получить доход разными способами.

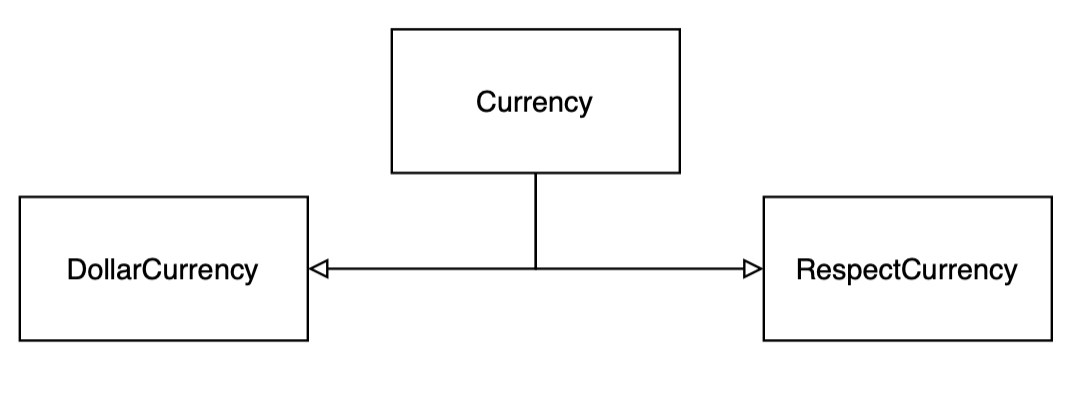
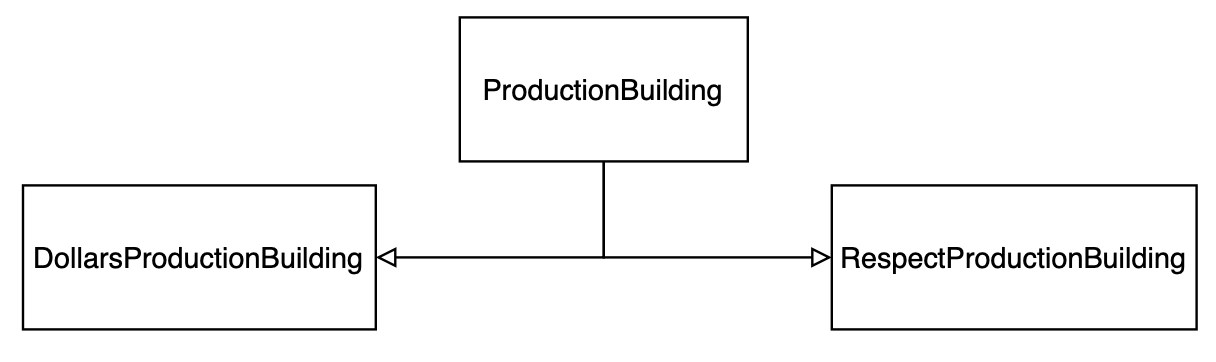
Определим следующие способы получения дохода: оффлайн-доход;

доход за закрашенный блок граффити; доход, генерирующийся домами.

На рисунке 3.3 представлена архитектура модуля получения дохода.

Рисунок *3.3* – Архитектура модуля получения дохода

50

Модуль игровых валют предназначена для хранения значения игровых валют игрока. Определим две игровые валюты: доллары и респект.

Доллары – это редкая игровая валюта, которая будет начисляться в специальных домах и за просмотр рекламы. За доллары можно открывать специальные дома и увеличивать их параметры.

Респект – это основная игровая валюта, которая начисляется за рисование граффити в специальных домах, и за просмотр рекламы. За респект можно увеличивать параметры художников, расширять команду игрока, покупать дома и увеличивать их параметры.

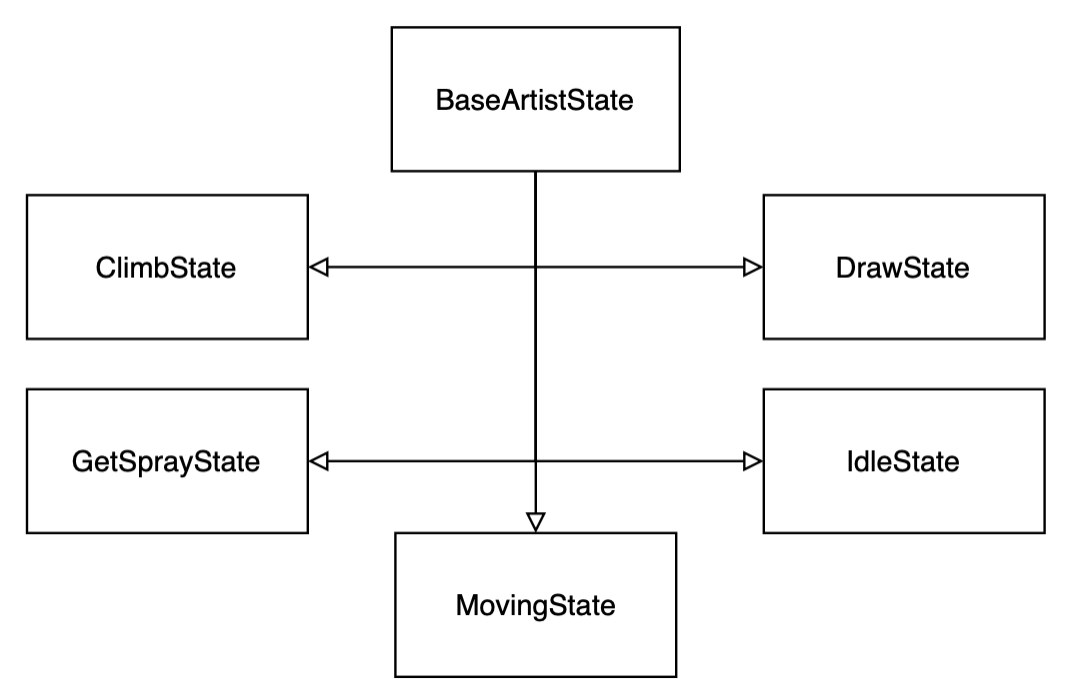
На рисунке 3.4 представлена архитектура модуля игровых валют.

Рисунок *3.4* – Архитектура модуля игровых валют

Модуль домов, производящих доход предназначена для генерации и начисления дохода игроку. Дома могут генерировать две игровые валюты: доллары и респект. На рисунке 3.5 представлена архитектура модуля игровых домов, производящих доход.

Рисунок *3.5 –* Архитектура модуля домов, производящих доход

51

Модуль художников предназначена для определения и реализации поведения художников на этапе рисования граффити.

Так как художник может иметь разное поведение, было решено использовать паттерн «State». Определим варианты поведения художников:

состояние лазанья по лестнице; состояние рисования граффити;

состояние пополнения баллончика с краской; состояние ожидания;

состояние перемещения.

На рисунке 3.6 представлена архитектура модуля художников.

Рисунок *3.6 –* Архитектура модуля художников

Модуль пользовательского интерфейса предназначена для определения логики отображения интерфейсных окон. Определим типы интерфейсных окон:

окно настроек художника;

окно дома, производящего доход; окно получения оффлайн дохода;

окно завершения рисования граффити.

52

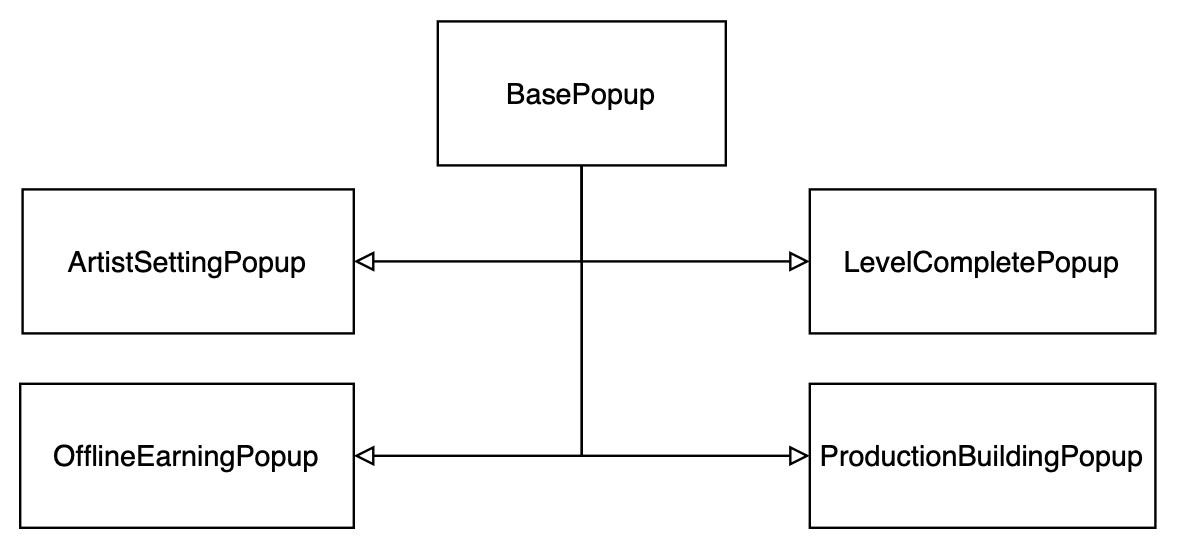
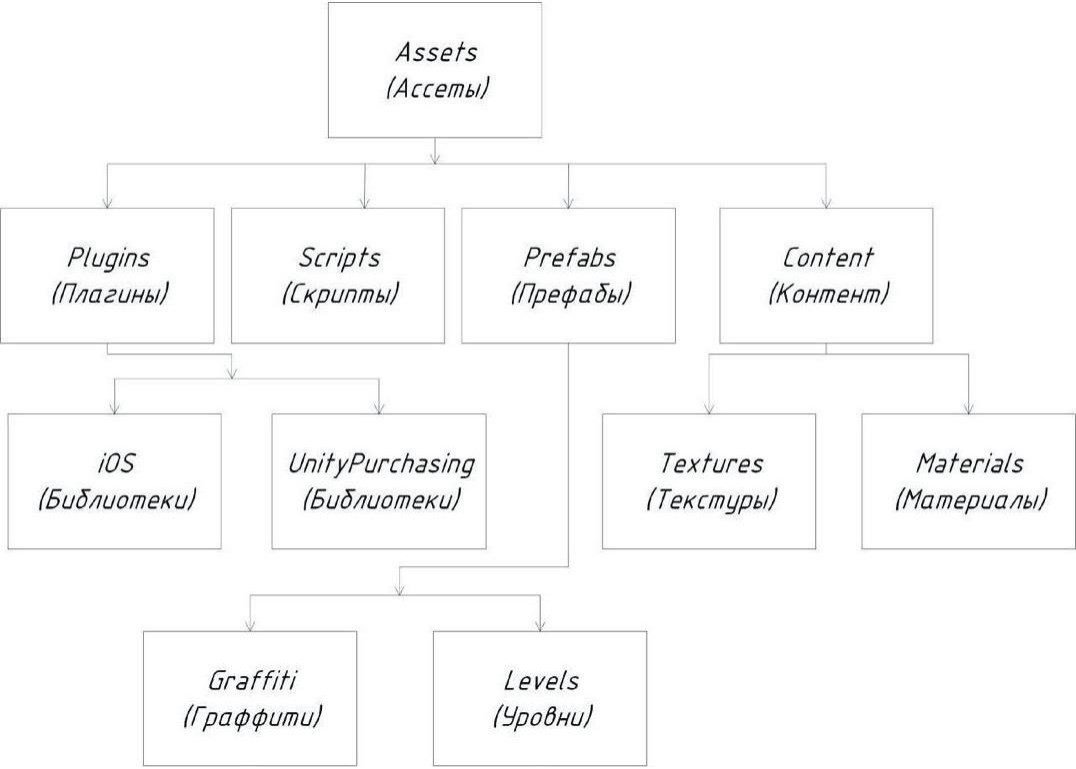
На рисунке 3.7 представлена архитектура модуля пользовательского интерфейса.

Рисунок *3.7 –* Архитектура модуля пользовательского интерфейса.

Для удобной работы с проектом, необходимо определить структуру проекта. На рисунке 3.8 представлена структура проекта.

Рисунок *3.8 –* Структура проекта

53

Папка Assets – основная папка проекта. Все необходимые файлы, предназначенные для разработки проекта, должны быть помещены в эту папку.

Папка Plugins – предназначена для хранения сторонних библиотек, используемых в проекте.

Папка iOS – хранит библиотеки, предназначенные только для платформы iOS.

Папка UnityPurchasing – хранит библиотеку для интеграции логики внутриигровых покупок.

Папка Scripts – хранит весь программный код приложения.

Папка Prefabs – предназначена для хранения префабов, используемых в игре.

Папка Graffiti – хранит префабы граффити, заготовленных по расположению и размерам.

Папка Levels – хранит префабы подготовленныхи настроенныхигровых уровней.

Папка Content – предназначена для хранения игрового контента. Игровым контентом могут быть: текстуры, спрайты, модели, материалы и аудиофайлы.

Папка Textures – предназначена для хранения всех текстур, используемых в игре.

Папка Materials – предназначена для хранения всех материалов, используемых в игре.

**3.2 Программная реализация мобильного приложения для 3D-игры**

Для реализации мобильного приложения был использован высокоуровневый язык программирования C# и движок для реализации игр Unity.

При разработке игр на Unity необходимо придерживаться модели Entity-Controller, т.е. имеется чёткое разграничение на сущности и котроллеры.

В качестве сущностей в Unity выступают любые объекты, расположенные на игровой сцене.

В качестве контроллеров выступают любые скрипты, которые прикреплены к сущностям.

54

Также в Unity есть возможность использовать логику, которая не будет прикреплена к сущностям.

Рассмотрим сущности основных модулей, описанных в подразделе проектирования архитектуры.

Сущность «DynamicParameter» базовый класс, реализующий основную логику изменения динамических параметров в игре. В таблице 3.1 представлены основные методы сущности «DynamicParemeter».

Таблица *3.1* – Основные методы сущности DynamicParameter

Имя

SetCalculators

Входные параметры

Calculator parameter, Calculator price

Выходные параметры

–

Назначение

Назначение логики расчёта параметров

Upgrade – –

Увеличения значения параметр

Принудительная SetLevel int level – установка значения

параметра

GetValue – float

Возвращение значения параметра

Возвращение

GetPrice – float значения стоимости увеличения параметра

Сущность «BaseGameFactory» предназначена для подготовки и генерации уровней с различыми режимами игры. В таблице 3.2 представлены основные методы сущности «BaseGameFactory».

Таблица *3.2* – Основные методы сущности BaseGameFactory

Имя

CreateGame

Входные параметры

–

Выходные параметры

–

Назначение

Подготавливает и инициализирует необходимые компоненты для

старта уровня

55

Сущность «DefaultGameFactory» предназначена для подготовки и генерации уровня в режиме рисования граффити. В таблице 3.3 представлены основные методы сущности «DefaultGameFactory».

Таблица *3.3* – Основные методы сущности DefaultGameFactory

Имя

InitializeGraffitiDrawer

HandleOfflineDrawer

CreateDefaultGraffiti

Drawer

Входные параметры

–

–

–

Выходные параметры

–

–

–

Назначение

Инициализирует логику рисования граффити

Инициализирует логику нарисованного граффити в режиме оффлайн

Инициализирует логику рисования граффити в режиме

реального времени

Сущность «ViewGameFactory» предназначена для подготовки и генерации уровня в режиме просмотра нарисованного граффити. Втаблице 3.4 представлены основные методы сущности «ViewGameFactory».

Таблица *3.4* – Основные методы сущности ViewGameFactory

Имя

InitializeGraffitiDrawer

Входные параметры

–

Выходные параметры

–

Назначение

Инициализирует логику рисования граффити

Сущность «Income» предназначена для начисления внутриигрового дохода игроку. В таблице 3.5 представлены основные методы сущности «Income».

56

Таблица *3.5* – Основные методы сущности Income

Имя

Give

Входные параметры

float multiplier

Выходные параметры

–

Назначение

Начисляет игроку внутриигровой доход.

Сущность «OfflineIncome» предназначена для начисления внутриигрового дохода игроку за отсутствие в игре. В таблице 3.6 представлены основные методы сущности «OfflineIncome».

Таблица *3.6* – Основные методы сущности Income

Имя

Give

Входные параметры

float multiplier

Выходные параметры

–

Назначение

Начисляет игроку внутриигровой доход

GetIncome

GetIncomeForSecond

CurrencyType type

int currencyType

float value

–

Возвращает доход для определённой валюты Возвращает доход, полученный за

одну секунду

Сущность «Currency» предназначена для хранения внутриигровых валют игрока. В таблице 3.7 представлены основные методы сущности «Currency».

Таблица *3.7* – Основные методы сущности Income

Имя

Входные параметры

Выходные параметры

Назначение

GetValue – float value

Возвращает

значение валюты

Сущность «ProductionBuilding» предназначена для генерации и начисления дохода игроку. В таблице 3.8 представлены основные методы сущности «ProductionBuilding».

57

Таблица *3.8* – Основные методы сущности ProductionBuilding

Имя

1

OnSelected

OnUnSelected

CollectIncome

UnlockBuilding

UnlockAutoCollecting

UpgradeIncome

UpgradeProduction Speed

Initialize

Входные параметры 2

–

–

–

–

–

–

–

–

Выходные параметры 3

–

–

–

–

–

–

–

–

Назначение

4 Отслеживает и обрабатывает факт выбора дома Отслеживает и обрабатывает факт отмены выбора дома

Начисляет игроку сгенерированный доход

Происходит разблокировка дома

Происходит разблокировка автоматического сбора дохода

Увеличение дохода

Увеличение скорости сбора дохода Инициализирует логику дома, производящего

доход

Сущность «Artist» предназначена для отслеживания и изменения состояний художника. В таблице 3.9 представлены основные методы сущности «Artist».

58

Таблица *3.9* – Основные методы сущности Artist

Имя

HandleState

HandleSpeedBonus

ChangeState

Входные параметры

–

–

IArtistState state

Выходные параметры

–

–

–

Назначение

Отслеживает логику текущего состояния Отслеживает активность бонуса ускорения художника Изменяет текущее

состояние на новое

Сущность «BaseArtistState» предназначена для базового описания поведения художника. В таблице 3.10 представлены основные методы сущности «BaseArtistState».

Таблица *3.10* – Основные методы сущности BaseArtistState

Имя

OnStateEnter

OnStateExit

HandleState

Входные параметры

IArtist artist

–

–

Выходные параметры

–

–

IArtistState State

Назначение

Описывает логику при включении состояния Описывает логику при выключении состояния Обрабатывает

логику состояния

Листинг кода сущностей представлен в приложении А.

**3.3 Тестирование мобильного приложения для 3D-игры**

Тестирование проводится для того, чтобы найти ошибки в программе и тем самым повысить её надёжность. Делается это путём исправления ошибок, внесённых в процессе разработки.

Под отладкой следует понимать процесс, позволяющий получить программу, функционирующую с заданными характеристиками в заданной

59

области входных данных. Основным методом отладки является тестирование, которое обеспечивает последовательность исходных данных, подаваемых на вход изделия и соответствующие им наборы эталонных результирующих данных [16].

Можно выделить три основных стадии тестирования:

для обнаружения ошибок в программе (на этой стадии выявляются все отклонения результатов функционирования от эталонных);

для диагностики и локализации причин (на этой стадии необходимо точно определить место, в котором произошло искажение программы или данных);

для контроля выполнения корректировок (после локализации и устранения ошибок выполняется контрольное тестирование, подтверждающее правильность выполненной корректировки и подтверждающее, что в результате корректировки не возникли вторичные ошибки).

Эффективность тестирования определяет стоимость и длительность разработки. По завершении отладки в течении некоторого времени интенсивность обнаружения ошибок при самом активном тестировании снижается настолько, что разработчик попадает в зону нечувствительности к отказам и ошибкам. В этой ситуации уже трудно прогнозировать, когда будет обнаружена следующая ошибка и будет ли она обнаружена вообще [17].

Типы тестов, которые проводились над программным продуктом:

функциональное тестирование. Данный тип теста показал, что программное средство выполняет все функции, которые содержатся в техническом задании, а также подтвердил устойчивость к различным экстренным ситуациям: перетаскивание предметов за пределы окна инвентаря, прохождение сквозь модели, реализацию искусственного интеллекта, обработку информации, передаваемой обработчиком;

UI-тестирование. Данный тип теста позволил устранить ошибки графического интерфейса: орфографические и синтаксические ошибки, выравнивание, отображение графических объектов, визуализацию всплывающей информации, визуализацию всплывающей информации, переданной обработчиком, обработку нажатий на элементы управления, отображения состояния активности и правильное отображение и анимации.

Так как тестирование мобильных игр автоматизировать практически невозможно, было проведено мануальное тестирование. В таблице 3.11 представлены результаты тестирования программного продукта.

60

Таблица *3.11* – Тестирование программного продукта

Описание 1

Запуск приложения

Нажатие в пустую область экрана Нажатие на заблокированный дом

Нажатие на кнопку покупки дома

Нажатие на разблокированный дом в процессе покраски

Изменение ползунка

количества художников

Ожидаемый результат 2

Отображение загрузочного окна. Переход в главное меню игры после завершения загрузки. Появление на экране эффекта волны. Появляется всплывающее окошко с отображение кнопки для покупки дома и указанием стоимости покупки.

Происходит уменьшение значения валюты на то, которое указано в цене. Открывается меню назначения художников. Появляется меню назначения художников для покраски дома. В ползунке для изменения количества художников отображается то количество, которое возможно установить. Происходит изменение количества художников в соответствии со

значением на ползунке.

Результат тестирования 3

Ошибок не найдено

Ошибок не найдено

Отображалась неверная цена для покупки дома

Значение валюты не изменялось

Ошибок не найдено

Количество художников не изменялось на экране

рисования

61

Продолжение таблицы *3.11*

1 Нажатие на разблокированный с нарисованным граффити

Отображения состояния кнопок динамических параметров

Нажатие на заблокированную кнопку изменения динамического параметра Нажатие на разблокированную кнопку изменения динамического параметра

Проверка состояния кнопки при достижении максимального значения параметра

Проверка нажатия на кнопку, достигшую максимального

значения

2 Появления меню изменения параметров дома.

Кнопка отображается активной, в случае если у игрока достаточно игровой валюты, иначе кнопка отображается не активной. Отображение корректного значения стоимости улучшения. Кнопка нажиматься не будет, анимация нажатия проигрываться не будет.

Происходит изменения значение валюты в соответствии с ценой. На кнопке отображается новое значение цены.

При достижении максимального значения параметра кнопка отображается активной, вместо цены отображается «MAX». Игровая валюты списываться не будет, параметр значение не изменяет, анимация нажатия не

проигрывается.

3 Ошибок не найдено

При недостаточном количестве денег, кнопка отображается активной

Проигралась анимация нажатия

Цена подсчитывалась неправильно

Отображалась цена за предыдущее улучшение

Ошибок не найдено

62

Продолжение таблицы *3.11*

1 Разблокировка параметра автоматического получения дохода

Закрытие интерфейсного окна

Переход в меню просмотра дома в процессе покраски

Переход в меню просмотра дома после покраски

Выход из меню дома

2 Значение валюты

списывается исходя из цены. Доход из дома собирается автоматически. Кнопка выглядит не активной. При нажатии на кнопку закрытия или в пустую область вокруг окна, интерфейсное окно закрывается. Отображается анимация нажатия на кнопку, появляется меню загрузки, после окончания загрузки открывается меню просмотра рисования граффити. Происходит генерация художников. Отображается анимация нажатия на кнопку, появляется меню загрузки, после окончания загрузки открывается меню просмотра граффити. Художники не генерируются. Происходит анимация нажатия, появляется загрузочный экран, главный экран отображается после

загрузки.

3 Автоматический доход разблокировался для всех домов

Ошибок не найдено

Ошибок не найдено

Происходит генерация художников

Ошибок не найдено

63

Продолжение таблицы *3.11*

1 Проверка логики

поведения художников

2 Художник бежит за баллончиком с краской

после генерации. После получения баллончика, художник бежит к строительным лесам. В случае если области рисования находится выше, художник лезет на нужный этаж. В случае если область рисования находится на одном и том же уровне, художник бежит к блоку. Если в процессе рисования закончилась краска, художник бежит за новым баллончиком. После завершения рисования художник бежит к

выходу.

3

В случае если блок находился выше, художник оставался на месте до момента

завершения уровня

В процессе тестирования были найдены и устранены ошибки, которые приводили к неправильной работе всей системы. Также был проверен и исправлен пользовательский интерфейс приложения.

После выполнения тестирования, программный продукт готов к использованию в качестве бета-версии продукта, так как некоторые недочёты будут замечены только после длительного использования на разных устройствах.

64

**4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И РЕАЛИЗАЦИИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ 3D-ИГРЫ НА UNITY**

**4.1 Описание функций, назначения и потенциальных пользователей мобильного приложения для 3D-игры на Unity**

Разработанное мобильное приложение для 3D-игры на Unity выполнено в жанре Idle. Idle-игры (они же фермы или инкрементальные игры) – это игры, целиком посвящённые управлению потоками доходов. Как и симуляторы, они фокусируются на принятии решений, которые увеличивают доход. Ядром игры является скорость генерации ресурсов. Но экономика с единственной валютой, которая так быстро растёт, имеет и недостатки [19].

Визуальная составляющая выполнена в виде 3D-графики с игровой камерой, фиксированной на главном здании. Также будет присутствовать развитие команды игрока.

Геймлей разрабатываемого проекта основан на облагораживания города посредством нанесения граффити на стены зданий.

Разработанная система обеспечивает выполнение следующих функций: 1 Запуск игры.

2 Просмотр уровня команды. 3 Получение игровой валюты.

4 Просмотр полученной игровой валюты. 5 Включение и выключение звуков в игре. 6 Улучшение команды игрока.

7 Улучшение ёмкости баллончика с краской. 8 Улучшение скорости бега художников.

9 Улучшения скорости покраски граффити. 10 Просмотр нарисованных граффити.

11 Просмотр карты города.

12 Покупка заблокированных домов. 13 Улучшение домов города.

14 Улучшение скорости получения дохода. 15 Улучшение дохода от дома.

16 Сохранение прогресса игры. 17 Загрузка прогресса игры.

18 Выход из приложения.

65

С расцветом мобильного рынка значительно выросли масштабы игровой аудитории, что привело к ещё большему разделению игроков на категории. Чаще всего игроков разделяют на казуальных, мидкорных и хардкорных [18]. На рисунке 4.1 представлены параметры, по которым разделяют игроков

по категориям.

Рисунок *4.1 –* Категории игроков по параметрам [18]

Исходя из жанра игры, можно определить, что потенциальные пользователи относятся к мидкорной категории. Таким образом портрет потенциального пользователя будет следующим:

мужчины и женщины, в возрасте от 6 до 28 лет;

используют смартфоны платформы iOS или Android; ежедневно играют от 30 до 120 минут;

цель игры: получение удовольствия.

Оценка аналитиков DFC Intelligence сильно превышает недавний прогноз Newzoo, ожидающих, что к концу этого года общее число геймеров во всем мире достигнет 2,7 млрд человек, а трехмиллиардный рубеж будет пройден только к 2023-му году. В DFC Intelligence к геймерам относят всех, кто периодически играет в какие-либо игры вне зависимости от платформы.

Из отчёта следует, что почти половинуиз озвученного значения в 3 млрд составляют те, кто играет только на смартфонах или мобильных устройствах, причём мобильный сегмент демонстрирует самые высокие темпы роста среди всех остальных [19].

На рисунке 4.2 представлено процентное соотношение игроков на смартфонах по отношению к разным регионам.

66

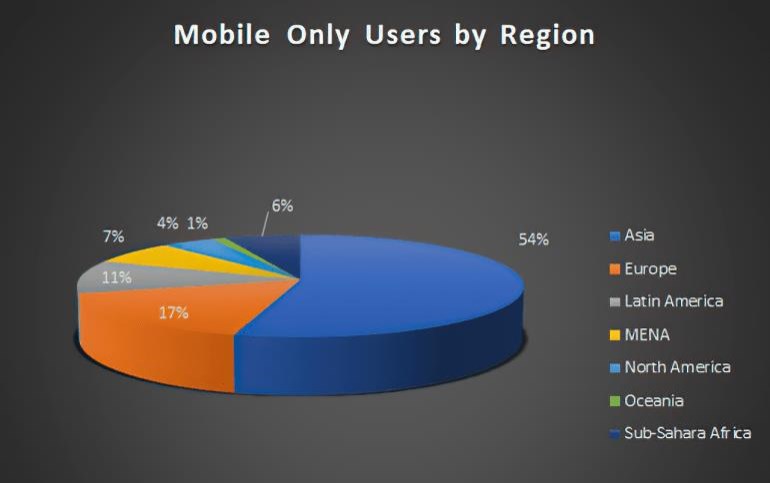


Рисунок *4.2 –* Соотношение игроков относительно разных регионов [19]

Исходя из количества игроков на мобильных платформах и довольно высоком охвате аудитории, следует, что спрос на данный продукт высокий.

Мобильное приложение разрабатывалось для получения экономического эффекта при помощи показа рекламы пользователям. Исходя из полученных данных, на этапе маркетингового тестирования, данный эффект легко поддаётся стоимостной оценке.

**4.2 Расчёт затрат на разработку мобильного приложения для 3D-игры на Unity**

Упрощённый расчёт затрат на разработку ПО следует делать в разрезе следующих статей:

затраты на основную заработную плату разработчиков;

затраты на дополнительную заработную плату разработчиков; отчисления на социальные нужды;

прочие затраты (амортизационные отчисления, расходы на электроэнергию, командировочные расходы, арендная плата за офисные помещения и оборудование, расходы на управление и реализацию и т.п.).

67

Затраты на основную заработную плату команды разработчиков рассчитываются по формуле [20]:

Зо = Кпр ∑𝑖=1 Зч.𝑖𝑡𝑖 , (4.1)

𝑛

где *n* – количество исполнителей, занятых разработкой конкретного ПО;

Кпр – коэффициент, учитывающий процент премий; Зч,*i* – часовая заработная плата i-го исполнителя, р.;

*ti* – трудоёмкость работ, выполняемых i-м исполнителем, ч.

Часовая заработная плата определяется путём деления месячной заработной платы (оклад плюс надбавки) на количество рабочих часов в месяце.

При этом месячная заработная плата определяется по фактическим данным предприятия ООО «Гисмарт», на котором проходилась преддипломная практика.

Трудоёмкость определяется исходя из сложности разработки программного обеспечения и объёма выполняемых им функций.

Размер премии может определяться исходя из практики, сложившейся в организации ООО «Гисмарт», или браться условно в диапазоне 50–100 % от размера основной заработной платы.

Расчёт затрат на основную заработную плату представлен в таблице 4.1.

Таблица *4.1 –* Расчёт затрат на основную заработную плату разработчиков Наимено- Вид выпол- Месячная Часовая Трудоём- Зарплата

вание по

должности тарифу,

няемой заработная заработная кость разработчика работы плата, р. плата, р. работ, ч р.

1 2 3 4 5 6 1 Инженер- Разработка

программист и програм- 2000 11,91 120 1429,20

мирование

ПО

2 Руководи- Разработка

тель проекта и контроль 2500 14,88 60 892,80

за планом

работ

68

Продолжение таблицы *4.1*

Р = ,

1 2 3 4 3 Специалист Разработка

по общего

компьютерной визуального 1500 8,93 графике стиля

проекта Премия (50 %)

Всего основная заработная плата разработчиков

5 6

60 535,80

1428,90 4286,70

Затраты на дополнительную заработную плату разработчиков включают выплаты, предусмотренные законодательством о труде (оплата трудовых отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по формуле [20]:

Зд = 100 % , (4.2)

З ∙ Н

о д

где *З*о – затраты на основную заработную плату, р.;

Нд – норматив дополнительной заработной платы, равный 15 %.

Вычислим затраты на дополнительную заработную плату команды

Зд = 4286,70 ∙ 15 = 643,01 р.

100

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяем в соответствии с действующими законодательными актами по формуле [19]:

(Зо + Зд) ∙ Нсоц соц 100 %

(4.3)

где Нсоц – норматив отчислений в фонд социальной защиты населения (34.6 %).

Вычислим отчисления на социальны нужды

Рсоц = (4286,70 + 643,01) ∙ 34,6 = 1705,68 р.

100

69

Прочие затраты процентах от затрат по формуле [20]:

включаются в на основную

себестоимость разработки ПО в заработную плату разработчиков

Рпз = 100 % , (4.4)

З ∙ Н

о пз

где Нпз – норматив прочих затрат равен 100 %.

Рпз = 4286,70 ∙ 100 = 4286,70

100

Полная сумма затрат на разработку ПО находится путём суммирования всех рассчитанных статей затрат. Результаты приведены в таблице 4.2.

Таблица *4.2 –* Затраты на разработку программного обеспечения

Наименование статьи затрат

1 Основная заработная плата разработчиков

2 Дополнительная заработная плата разработчиков 3 Отчисления на социальные нужды

4 Прочие затраты

Общая сумма затрат на разработку

Сумма, р. 4286,70 643,01 1705,68 4286,70

10922,09

**4.3 Оценка экономического эффекта и расчёт показателей эффективности инвестиций при разработке программного обеспечения для свободной реализации на рынке IT**

Полная сумма затрат на разработку программного обеспечения составляет 10922,09 р.

Разработка программного средства велась для свободной реализации на рынке IT. При таком условии экономический эффект организации-разработчика представляет собой прибыль от показа рекламы множеству потребителей.

Прибыль от реализации в данном случае напрямую зависит от объёмов показа рекламы, цены реализации и затрат на разработку данного ПО.

Цена одного показа рекламы 0,1 р. определена на основе проведённых маркетинговых тестов.

Налог на добавленную стоимость определяется по формуле [20]:

70

НДС = 100 + Ндс , (4.5)

Ц ∙ 𝑁 ∙ Н

дс

где Ц – стоимость показа рекламы (руб.); *N* – количество показа рекламы;

Ндс – ставка налога на добавленную стоимость согласно действующему законодательству 20 %.

НДС = 0,1 ∙ 720000 ∙ 20 = 12000 р.

100 + 20

Сумма налога на добавленную стоимость равна 12000 р.

Прибыль, полученная разработчиком от реализации ПО на рынке, определяется по формуле [20]:

П = Ц ∙ 𝑁 − НДС − Зр, (4.6)

П = 0,1 ∙ 720000 − 12000 − 10922,09 = 49077,91 р.

Организация является плательщиком налога на прибыль, поэтому экономический эффект рассчитывается по формуле [20]:

Пч = П ∙ (1 − 100 % ), (4.7)

Н

п

где Нп – ставка налога на прибыль –18 %.

Пч = 49077,91 ∙ (1 − 100 ) = 40243,89 р.

18

Уровень рентабельности рассчитывается по формуле [20]:

Ур = Прч ∙ 100 %, (4.8)

З

Ур = 10922,09 ∙ 100 = 368,46 %

40243,89

71

Средняя процентная ставка в национальной валюте по новым банковским вкладам (депозитам) для юридических лиц на срок свыше 1 года на февраль 2020 года равняется 13,35 %.

На основании полученныхрезультатов можно сделать вывод, что проект будет экономически эффективным, так как рентабельность затрат на данную разработку программного обеспечения составляет 368,46 %, что не меньше средней процентной ставки по банковским депозитным вкладам.

Для расчёта показателей эффективности использования программного средства необходимо полученные суммы результата (прироста чистой прибыли) и затрат (капитальных вложений) по годам приводят к единому времени расчётному году путём умножения затрат за каждый год на коэффициент привидения, который определяется по формуле [20]:

𝑎𝑡 = (1 + Ен)𝑡, (4.9)

1

где Ен – норма привидения разновременных затрат и результатов равна 11 % (не ниже ставки рефинансирования НБ РБ);

*t* – порядковый номер года, результаты и затраты которого приводят расчётному.

Рассчитаем коэффициенты приведения по годам

𝑎1 = (1 + 0,11)0 = 1

1

𝑎2 = (1 + 0,11)1 = 0,9

1

𝑎3 = (1 + 0,11)2 = 0,81

1

Дисконтированная стоимость будущих поступлений денежных средств находится по формуле [20]:

ЧДД = ∑𝑡=1(𝑃 − З𝑡) ∙ 𝑎𝑡, (4.10)

𝑛

𝑡

где *n* – расчётный период, лет;

72

*Pt* – результат (экономический эффект – прибыль или чистая прибыль), полученный в году t, р.;

З*t* – затраты (инвестиции – затраты на разработку (модернизацию) или на приобретение и внедрение ПО) в году t, р.

Разработка ПО была начата в 2020 года, а реализация приложения планируется в 2021 году. В последующие годы реализация приложения не планируется так как игры жанра Idle не реализуются более двух лет.

ЧДД1 = (0 − 10951,67 ) ∙ 1 = −10951,67р.

ЧДД2 = 40243,89 ∙ 0,9 = 36219.50 р.

ЧДД3 = 0 ∙ 0,81 = 0 р.

В таблице 4.3 представлен расчёт показателей эффективности инвестиций.

Таблица *4.3 –* Расчёт эффективности инвестиционного по разработке программного обеспечения

Наименование показателя обоз. 1-й Расчётный год

Усл.

2-й

проекта

3-й

1 2 3 4 5 Результат

1 Прирост результата (чистой прибыли)

2 Коэффициент дисконтирования

3 Результат с учётом фактора времени

Затраты (инвестиции)

4 Инвестиции в разработку (модернизацию) ПО

5 Инвестиции с учётом

фактора времени

Рt 0

t 1

Рtt 0

З 10922,09

Зtt 10922,09

40243,89 0

0,9 0,81

36219,50 0

0 0

0 0

73

Продолжение таблицы *4.3* 1

6 Чистый дисконтированный доход по годам

7 Чистый дисконтированный

доход нарастающим итогом

2 3

ЧДДt –10922,09

ЧДД –10922,09

4

36219,50

25297,41

5

0

25297,41

Среднегодовая величина чистой прибыли за расчётный период определяется по формуле [19]:

Пчср = ∑𝑡=1 Пч𝑡, (4.12)

𝑛

𝑛

где Пч*t* – чистая прибыль, полученная в году t, р.

Пчср = 40243,89 = 13414,63 р.

3

Рентабельность инвестиций определяется по формуле [20]:

Ри = Пчср ∙ 100 %, (4.13)

З

Ри = 10922,09 ∙ 100 = 122,82 %

13414,63

В результате технико-экономического обоснования применения программного средства были получены следующие значения показателей их эффективности:

− чистый дисконтированный доход в период первых трёх лет, составит 25297,41;

− проект окупается на 2 год;

− рентабельность инвестиций составляет 122,82 %.

Таким образом, применение программного продукта является эффективным и инвестиции в его разработку целесообразными.

74

**5 ОХРАНА ТРУДА. ОБЕСПЕЧЕНИЕ БЕЗОПАСНЫХ УСЛОВИЙ ТРУДА** **ПРОГРАММИСТА:** **ВЫБОР** **И** **ОБОСНОВАНИЕ КОМПЛЕКСА МЕР ПО ЗАЩИТЕ ОТ СТАТИЧЕСКОГО ЭЛЕКТРИЧЕСТВА**

Целью дипломного проектирования является разработка мобильного приложения для 3D-игры на Unity. Приложение представляет собой мобильную игру, реализующую следующий функционал:

управление городом;

управление командой художников; рисование граффити;

хранение и загрузка внутриигровых данных пользователя о прогрессе игры.

При разработке системы и работе с ней важно обеспечить безопасные условия труда. Для этого необходимо определить опасные и вредные производственные факторы, которые могут оказать неблагоприятное воздействие на состояние человека и повлиять на производительность труда, а также разработать перечень мероприятий по защите от данных факторов.

Статическая электризация веществ имеет сложную природу. Существует несколько механизмов образования статического электричества: контактная электризация, электрохимический механизм, асимметричное заряжение в результате индукции в сильном электростатическом поле и др. Образование электростатических зарядов при контактной электризации происходит при разделении контактирующих поверхностей (разрыве контакта). Величина образовавшегося заряда определяется зарядами двойного слоя, электрическим сопротивлением материала и скоростью отрыва поверхностей (интенсивностью технологического процесса). В обычных же условиях при контакте двух материалов на их поверхностях вследствие действия внутриатомных электрических сил образуется двойной электрический слой. На поверхностях одного материала в месте контакта преобладают отрицательные заряды, на поверхности другого – положительные. При сохранении контакта суммарный заряд контактирующих материалов будет равен нулю. Образовавшиеся заряды при контактной электризации могут оставаться на поверхностях после их разделения только в том случае, если время разрушения контакта окажется меньше времени релаксации (рассеивания) зарядов. Образование электростатических зарядов в результате контактной электризации представлено на рисунке 5.1. Последнее в значительной степени определяет величину зарядов на разделительных

75

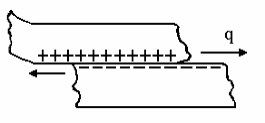
поверхностях, то есть чем выше скорость отрыва (чем интенсивнее ведется процесс), тем больший заряд остается на поверхностях. Толщина двойного электрического слоя, то есть пространственного разделения электрических зарядов на границах соприкосновения двух фаз (поверхностей) соответствует диаметру иона, равного 10-10 м. При контактной электризации положительные заряды возникают на материале, имеющем большее значение диэлектрической постоянной [21].

Рисунок *5.1* – Образование электростатических зарядов в результате контактной электризации

Так как процесс разработки и эксплуатации системы предполагает работу с персональными компьютерами и другой офисной техникой, пользователь подвергается воздействию статического электричества.

Заряды статического электричества в производственных условиях могут накапливаться на теле работающих и их одежде [21].

Дисплеи и кабели питания могут генерировать рентгеновское, радиочастотное, электромагнитные и ультрафиолетовое излучение. Наличие электростатического поля приводит к уменьшению содержания отрицательных ионов в воздухе помещения и загрязнению экрана и других периферийных устройств в результате притягивания к ним мелких частиц пыли. Пыль как вредное вещество может оказывать на организм фиброгенное, токсическое, раздражающее, аллергенное, канцерогенное действие [22].

Увеличение электростатического заряда и разности потенциалов на разделённых поверхностях может привести к электрическому пробою разделяющей среды. Возникновение искрового разряда и высоких потенциалов представляет собою наиболее опасное проявление статического электричества.

Так как персональные компьютеры и периферийные устройства имеют токоведущие части и могут подключаются к сети, то создаётся опасность статического электричества.

76

Со статическим электричеством знакомы все люди. Это совокупность явлений, которые связаны с возникновением, сохранением и свободного накопления электрического заряда. Последний возникает на поверхности диэлектрика, который плохо проводит ток, или на изолированным проводнике, не имеющим доступ к постоянному току.

При прикосновении человека к предмету, несущему электрический заряд, происходит разряд последнего через тело человека. Величины возникающих при разрядке токов не велики, и они очень кратковременны. Поэтому электротравм не возникает. Однако, разряд, как правило, вызывает рефлекторное движение человека, что в ряде случаев может привести к резкому движению руки, падению человека с высоты или его попаданию в опасную производственную зону [22].

Кроме того, при образовании заряда с большим электрическим потенциалом вокруг них создаётся электрическое поле повышенной напряжённости. Установлено, что электрическое поле повышенной напряжённости вредно для человека. При длительном пребывании человека в таком поле наблюдаются функциональные изменения в центральной нервной системе, сердечно- сосудистой и других системах. Для человека, находящегося в электростатическом поле, характерна повышенная утомляемость, сонливость, снижение внимания, скорости двигательных и зрительных реакций [22].

На сегодняшний день влияние статического электричества на здоровье и иммунную систему человека не исследовано в полной мере.

Но на основе уже проведённых исследований можно классифицировать негативные воздействия на организм человека в случае долговременного нахождения в поле статистического заряда:

функциональные нарушения в центральной нервной системе;

спазм сосудов, способный вызвать повышение артериального давления;

чрезмерная эмоциональность и раздражительность; головные боли;

нарушения аппетита и сна;

возникновение фобий, постоянная боязнь вновь получить электрический разряд и ту боль, которая за ним последует.

В настоящем разделе рассмотрим вопросы, связанные с обеспечением электростатической безопасности при разработке мобильного приложения для 3D-игры на Unity.

77

Разряд статического электричества возникает тогда, когда напряжённость электрического поля над поверхностью диэлектрика или проводника, обусловленнаянакоплениемнанихзарядов, достигает пробивной величины [21].

При достижении энергией искрового разряда величины энергии воспламенения пылегазовоздушных и других взрыво- и пожароопасных смесей возникает опасность взрыва и пожара. Электростатическая искробезопасность объекта достигается при выполнении следующего условия безопасности:

𝑊 ≤ 𝑘 ∙ 𝑊 𝑖𝑛, (5.1)

𝑝 𝑚

где *Wp* – максимальная энергия зарядов, которые могут возникать внутри объекта или на его поверхности, Дж;

*k* – коэффициент безопасности, выбираемый из условий допустимой (безопасной) вероятности зажигания *(k <* 1,0*)*;

*Wмин* – минимальная энергия зажигания веществ и материалов, Дж.

Энергия искрового заряда с заряженной проводящей поверхности определяется по формуле:

𝑊 = 0,5 ∙ 𝐶 ∙ φ2*,* (5.2)

𝑝

где *C* – электрическая ёмкость проводящего объекта относительно земли, Ф;

φ – потенциал заряженной поверхности относительно земли, В. Заряды статического электричества и высокие потенциалы часто ведут

к отказам отдельных элементов аппаратуры (полупроводниковых приборов, микросхем), являются причиной ухудшения условий труда, вызывая у работающих при разрядах неприятные болезненные ощущения.

Степень опасности статического электричества определяется электростатическими свойствами веществ и материалов, используемых на производстве, а также чувствительностью изделий к электростатическим разрядам.

Нормируемым параметром статического электричества, характеризующим условия труда персонала, является напряжённость электростатического поля *Е*, кВ/м, допустимые уровни которой устанавливаются в зависимости от времени пребывания персонала на рабочем месте [21].

78

Предельно допустимый (или наибольший) уровень напряжённости устанавливается равным 60 кВ/М в течение одного часа, то есть:

𝐸

𝐸пд = 60 , (5.3)

√

𝑡

При напряжённости поля равной или меньше 20 кВ/м время пребывания в таком поле не регламентируется.

В диапазоне напряжённостей от 20 до 60 кВ/м допустимое время пребывания персонала в электростатическом поле без средств защиты (*t*пд) в часах определяется по формуле:

пд =

𝑇

(𝐸факт)2, (5.4) пд

где *Eфакт* – фактическое (измеренное) значение напряжённости в рабочей зоне (на рабочем месте).

К общим способам по снижению возможности образования и накопления зарядов статического электричества на рабочих поверхностях, изделиях, одежде и теле работающих относятся:

заземление электропроводных (в том числе и неметаллических) элементов оборудования и инструментов;

общее и местное увлажнение воздуха и его ионизация;

увеличение поверхностной и объёмной проводимости обрабатываемых материалов;

подбор контактирующих материалов, при которых уровень электризации минимален;

ограничение скорости переработки и транспортирования электризующихся материалов (уменьшение скорости перемешивания и переливания жидкостей, возможности вскубливания, разбрызгивания и т.п.).

Неметаллическое оборудование может считаться электростатически заземлённым, если сопротивление растеканию тока в земле с любых точек его внешней и внутренней поверхностей не превышает 107 Ом (при относительной влажности воздуха не выше 60 %). Например, покрытие пола считается электропроводным для статического электричества, если электрическое см2 на пол и прижатой с силой в 25 кг⋅см, и заземлением не превышает 107 Ом (бетон, керамическая плитка, ксилолит, антистатический линолеум и др.) [21].

79

Заземление работающих обеспечивается применением антистатических заземляющих браслетов, антистатической одежды и обуви.

Заземляющий браслет соединяется с заземлением (или с заземлённой нейтралью трёхфазной сети) через резистор сопротивлением не менее в один мегом (для обеспечения электробезопасности) гибким многожильным проводом (сечением не менее 1 мм2). Общее сопротивление цепи «тело человека – земля» не должно превышать 107 Ом.

Для уменьшения плотности зарядов наэлектризованного материала применяется индукционные, высоковольтные и радиационные нейтрализаторы.

Недостатком поверхностного нанесения антистатических веществ является недолговечность их действия, так как они неустойчивы к механическим воздействиям. Наиболее эффективным является внутреннее введение этих веществ в полимеры.

Снижение возможности образования опасной искры с поверхности наэлектризованного материала достигается в некоторых случаях увеличением электрической ёмкости заряженного материала по отношению к земле путём установки заземлённой металлической пластины либо сетки непосредственно под заряженной поверхностью [21].

Таким образом, изложенные выше мероприятия обеспечат электростатическую безопасность при разработке мобильного приложения для 3D-игры на Unity.

80

**ЗАКЛЮЧЕНИЕ**

В результате дипломного проектирования разработано мобильное приложение для 3D-игры на Unity, предназначенное для развлекательного времяпрепровождения людей независимо от пола или возраста.

Для достижения поставленной цели дипломного проекта были решены следующие задачи:

рассмотрены аналоги мобильных игр жанра Idle, проведен их сравнительный анализ, выявлены их достоинства и ограничения;

проведено обоснование выбора среды разработки и используемых технологий;

проведено инженерно-психологическое проектирование приложения; реализовано приложение, с использованием шаблонов

проектирования Abstract Factory, State; и принципов разработки SOLID, KISS, DRY;

проведено тестирование мобильного приложения;

выполнено технико-экономическое обоснование эффективности разработки и использования приложения;

протестирована работоспособность и корректность функционирования приложения;

разработан комплекс мер по защите от статического электричества для обеспечения безопасных условий труда при разработке приложения.

Исходя из требования о кроссплатформенности, в качестве среды разработки выбрал игровой движок Unity 3D, который на данный момент являетсяоднимиз самыхпопулярныхигровыхдвижков. ВUnityприсутствуют визуальная среда разработки, межплатформенная поддержка, модульная система компонентов и возможность создания анимации.

Из-за специфики работы в игровом движке Unity 3D, в качестве языка программирования был выбран C#.

Разработанное мобильное приложение устойчиво к аппаратным и программным сбоям и защищено от неправильных действий пользователей.

Тесты, проведённые для оценки качества разработанной автоматизированной системы были пройдены успешно. Smoke-тестирование определило, что система выполняет свои основные функции и способна запускаться на разных устройствах.

81

В результате технико-экономического обоснования применения программного средства были получены следующие значения показателей их эффективности:

В результате технико-экономического обоснования применения программного средства были получены следующие значения показателей их эффективности:

− чистый дисконтированный доход в период первых трёх лет, составит –2808,19;

− проект окупается на 2 год;

− рентабельность инвестиций составляет 27,54 %.

В результате эргономического проектирования проведён анализ и определено содержание функций системы, осуществлено распределение функций между человеком и приложением, определена структура системы, разработаны алгоритмы работы пользователя и эргономические требования к системе, составлен сценарий информационного взаимодействия человека и мобильного устройства, проведён расчёт оценки эргономичности пользовательского интерфейса.

Определены опасные и вредные производственные факторы при работе с системой, такие как повышенные уровни электромагнитного излучения, влияние статического электричества, напряжение зрения, внимания, длительное статическое напряжение, монотонность труда. Рассмотрены вопросы обеспечения безопасных условий труда и обоснован комплекс мер по защите от статического электричества.

По материалам дипломного проекта опубликованы тезисы «Разработка в среде Unity 3D» в материалах 55-й юбилейной научной конференции аспирантов, магистрантов и студентов Белорусского государственного университета информатики и радиоэлектроники.

82

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Индустрия компьютерных игр-2020 [Электронный ресурс] : Исследование / Высшая школа экономики, институт «Центр развития». – Режим доступа : https://app2top.ru/wp-content/uploads/2020/07/Industriya-komp-yuterny-h-igr-2020.pdf/. – Дата доступа : 21.11.2020.

[2] Medium [Электронный ресурс]. – Режим доступа : https://medium.com/gamedev-мёд-и-дёготь/popularity-of-mobile-game-art-styles-and-genres-8a29bea10ae2/. – Дата доступа : 16.11.2020.

[3] DTF [Электронный ресурс]. – Режим доступа : https://dtf.ru/gamedev/166-idle/. – Дата доступа : 16.11.2020.

[4] iXBT.GAMES [Электронный ресурс]. – Режим доступа : https://www.ixbt.com/live/games/gamedev-tycoon.html/. – Дата доступа : 16.11.2020.

[5] Play Market [Электронный ресурс]. – Режим доступа : https://play.google.com/store/apps/details?id=com.greenheartgames.gdt&hl=ru/. – Дата доступа : 16.11.2020.

[6] App Market [Электронный ресурс]. – Режим доступа : http://appmarketmod.com/arkady/1298-idle-light-city-v210.html/. – Дата доступа : 16.11.2020.

[7] Play Market [Электронный ресурс]. – Режим доступа : https://play.google.com/store/apps/details?id=com.nopowerup.idlelightcity&hl=ru &gl=US/. – Дата доступа : 16.11.2020.

[8] App Store [Электронный ресурс]. – Режим доступа : https://apps.apple.com/us/app/graffiti-idle/id1489301269?ign-mpt=uo%3D4/. – Дата доступа : 16.11.2020.

[9] iXBT.GAMES [Электронный ресурс]. – Режим доступа : https://www.ixbt.com/live/games/luchshie-igrovye-dvizhki-2020.html/. – Дата доступа : 16.11.2020.

[10] Создание игры на Unity 3D [Электронный ресурс]. – Режим доступа : http://orlova.rsue.ru/content/kg/Создание%20игры%20на%20Unity% 203d.htm/.– Дата доступа : 16.11.2020.

[11] Unity Dojo [Электронный ресурс]. – Режим доступа : https://unitydojo.blogspot.com/2015/04/how-to-detect-collisions-in-unity-like.html?m=1/. – Дата доступа : 16.11.2020.

[12] Рихтер, Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Д. Рихтер. – СПб. : Питер, 2019. – 896 c.

83

[13] Шупейко, И. Г. Теория и практика инженерно-психологического проектирования и экспертизы: учебно-методическое пособие к практическим видам занятий / И. Г. Шупейко. – Минск : БГУИР, 2009.

[14] Шупейко, И. Г. Эргономическое проектирование системы «человек – компьютер – среда»: учебно-методическое пособие к курсовой работе / И. Г. Шупейко. – Минск : БГУИР, 2012.

[15] Сан ПиН 9-131-РБ 2000. – Гигиенические требования к видеодисплейным терминалам, электронно-вычислительным машинам и организации работы. – Минск, 2000. – 18 с.

[16] Monographies [Электронный ресурс]. – Режим доступа : https://www.monographies.ru/ru/book/section?id=4632/. – Дата доступа : 22.11.2020.

[17] Липаев, В. Тестирование компонентов и комплексов программ / В. Липаев. – М. : Синтег, 2010. – 400 с.

[18] Habr [Электронный ресурс]. – Режим доступа : https://habr.com/ru/post/253895/. – Дата доступа : 16.11.2020.

[19] ITC [Электронный ресурс]. – Режим доступа : https://itc.ua/news/dfc-intelligence-chislo-igrayushhih-v-videoigry-lyudej-v-mire-prevysilo-3-mlrd-polovina-iz-nih-igraet-na-pk/. – Дата доступа : 16.11.2020.

[20] Технико-экономическое обоснование дипломных проектов (работ) : методическое пособие / А. А. Горюшкин [и др.]. – Минск : БГУИР, 2020. – 105 с.

[21] Михнюк, Т. Ф. Охрана труда. Учебное пособие для вузов. – Минск. : БГУИР, 2007. – 335 с.

[22] Девисилов, В. А. Охрана труда: учебник. / В. А. Девисилов. – М. : Форум, ИНФРА-М, 2013. – 448 с.

[23] Стандарт предприятия. Дипломные проекты (работы). Общие требования : справ. пособие / Доманов, А. Т. [и др.]. – Минск : БГУИР, 2017. – 174 с.

[24] Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – М.: Издательскоторговый дом "Русская Редакция", 2004. – 576 с.

[25] Официальное сообщество разработчиков на Unity3D [Электронный ресурс] – Режим доступа: //URL: http://unity3d.com/support/community – Дата доступа: 16.11.2020.

[26] Официальная документация по Unity3D [Электронный ресурс] – Режим доступа : http://unity3d.com/support/ – Дата доступа: 16.11.2020.

[27] Константайн, Л. Разработка программного обеспечения / Л. Констанайн, Л. Локвуд. – СПб. : Питер, 2004. – 592 с.

84

[28] Куликов, С. C. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. – Минск: Четыре четверти, 2017. – 312 с.

[29] Барабаш, В. И. Охрана труда специалистов, работающих с видеоматериалами: метод. рекомендации/ В. И. Барабаш. – ЛПИ им. Калинина,1990. – 16с.

[30] Безопасность жизнедеятельности: учебник для вузов / С. В. Белов [и др.]. – М.: Высш. шк., 1999. – 53с.

[31] Лазаренков, А. М. Охрана труда. / А. М. Лазаренков, В. А. Калиниченко. – Минск : ИВЦ Минфин, 2010. – 464 с.

85

**ПРИЛОЖЕНИЕ А** (обязательное)

Фрагмент программного кода

public class CameraController : MonoBehaviour {

#region Fields

[Header("Components")]

[SerializeField] private Camera m\_camera = null;

[Header("Angle Options")]

[SerializeField] private float m\_minAngle = 0f; [SerializeField] private float m\_maxAngle = 0f;

[Header("Zoom Options")]

[SerializeField] private float m\_zoomValue = 40;

[SerializeField] private Vector3 m\_zoomPositionOffset = Vector3.zero;

private float m\_baseZoom = 60;

private Vector3 m\_basePosition = Vector3.zero;

private BaseCameraState m\_cameraState = null;

private float m\_angle = 0;

private CameraConfig m\_cameraConfig = null;

#endregion

#region Properties

public Camera Camera => m\_camera;

public float Angle {

get => m\_angle;

private set => m\_angle = Mathf.Clamp(value, m\_cameraState.MinAngle, m\_cameraState.MaxAngle);

}

public float MinAngle => m\_minAngle;

public float MaxAngle => m\_maxAngle;

public float Zoom => m\_zoomValue;

public float BaseZoom => m\_baseZoom;

public Vector3 ZoomOffset => m\_zoomPositionOffset;

public Vector3 BasePosition => m\_basePosition;

86

public Transform Target { get; set; } #endregion

#region Unity Lifecycle

private void Start() {

m\_baseZoom = Camera.fieldOfView; m\_basePosition = Camera.transform.localPosition;

m\_cameraConfig = ConfigManager.Instance.Get<CameraConfig>(); InputService inputService = ServiceLocator.Instance.Get<InputService>();

inputService.eventOnPointerDown += ActivateCamera; inputService.eventOnPointerDrag += UpdateCameraAngle; inputService.eventOnPointerUp += ResetCamera;

GraffitiManager.Instance.EventGraffitiCompleted += OnGraffitiCompleted;

ChangeCameraState(new IdleCameraState()); }

private void OnDisable() {

InputService inputService = ServiceLocator.Instance.Get<InputService>();

if(inputService == null) return;

inputService.eventOnPointerDown -= ActivateCamera; inputService.eventOnPointerDrag -= UpdateCameraAngle; inputService.eventOnPointerUp -= ResetCamera;

GraffitiManager.Instance.EventGraffitiCompleted -= OnGraffitiCompleted; }

private void LateUpdate() {

HandleState(); }

#endregion

#region Private Methods

private void OnGraffitiCompleted() {

ChangeCameraState(new ResultViewCameraState()); }

private void ActivateCamera() {

87

if(m\_cameraState is ResultViewCameraState) return; DOTween.Sequence().SetDelay(m\_cameraConfig.delayBeforeZoom).OnComplete(() => {

ChangeCameraState(new ZoomViewCameraState()); }).SetId(this);

ChangeCameraState(new ViewCameraState()); }

private void UpdateCameraAngle() {

DOTween.Kill(this);

Vector2 inputDirection = ServiceLocator.Instance.Get<InputService>().Direction;

float angle = m\_cameraConfig.swipeForce \* (inputDirection.x \* inputDirection.magnitude);

Angle += angle;

}

private void ResetCamera() {

if(m\_cameraState is ResultViewCameraState) return;

DOTween.Kill(this);

ChangeCameraState(new IdleCameraState()); UpdateCameraAngle();

}

private void HandleState() {

BaseCameraState cameraState = m\_cameraState.HandleState();

if (cameraState != null && Target != null) {

ChangeCameraState(cameraState); }

}

private void ChangeCameraState(BaseCameraState newState) {

m\_cameraState?.OnStateExit(); m\_cameraState = newState; m\_cameraState.OnStateEnter(this);

} #endregion

}

public abstract class ProductionBuilding : MonoBehaviour {

#region Fields

88

[Header("Components")]

[SerializeField] protected ProductionBuildingView m\_productionBuildingView;

[Header("Options")]

[SerializeField] protected ProductionBuildingConfig m\_buildingConfig = null;

protected DynamicParameter m\_income; protected DynamicParameter m\_productionSpeed;

private ProductionBuildingData m\_buildingData;

private ProductionBuildingState m\_currentState;

private Building m\_building;

#endregion

#region Properties

public int Id => m\_buildingConfig.Id;

public float ProductionProgress {

get => m\_buildingData.productionProgress;

set => m\_buildingData.productionProgress = value; }

public float DrawProgress {

get => m\_buildingData.drawProgress;

set => m\_buildingData.drawProgress = value; }

public bool IsAutoCollectingEnabled {

get => m\_buildingData.isAutoCollectingEnabled;

private set => m\_buildingData.isAutoCollectingEnabled = value; }

public bool IsDrawn {

get => m\_buildingData.isDrawn;

set => m\_buildingData.isDrawn = value; }

public bool IsUnlocked => m\_buildingData.isUnlocked;

public bool IsReadyToCollect => ProductionProgress >= ProductionSpeed.Value;

public DynamicParameter ProductionSpeed => m\_productionSpeed;

public DynamicParameter Income => m\_income;

public float UnlockPrice => m\_buildingConfig.UnlockPrice;

89

public float AutoCollectPrice => m\_buildingConfig.AutoCollectingPrice;

public ProductionBuildingView ProductionBuildingView => m\_productionBuildingView;

public Sprite CompletedSprite => m\_building.CompletedTileSprite;

public int ArtistsCount {

get => m\_buildingData.artistsCount;

set => m\_buildingData.artistsCount = value; }

public string Status => m\_currentState.Status;

public abstract CurrencyType CurrencyType { get; }

#endregion

#region Unity Lifecycle

private void OnEnable() {

UpdateManager.EventOnUpdate += OnUpdate;

m\_productionBuildingView.EventOnUnlockClick += UnlockBuilding; m\_productionBuildingView.EventOnCollectClick += OnIncomeCollected;

}

private void OnDisable() {

UpdateManager.EventOnUpdate -= OnUpdate;

m\_productionBuildingView.EventOnUnlockClick -= UnlockBuilding; m\_productionBuildingView.EventOnCollectClick -= OnIncomeCollected;

}

private void Awake() {

Initialize(); }

private void Start() {

InitializeBuildingState(); }

#endregion

#region Public Methods

90

public void OnSelected() {

SendClickEvent();

if (m\_currentState is LockedBuildingState) {

m\_productionBuildingView.SetLockedUIActive(true); return;

}

if (m\_currentState is IncomeState && !IsAutoCollectingEnabled && IsReadyToCollect) {

OnIncomeCollected(); SendClaimIncomeEvent();

return; }

if (m\_currentState is ProductionState || m\_currentState is IncomeState) {

var popup = PopupManager.Instance.GetPopup<ProductionBuildingMenu>(); popup.EventAutoCollectUnlocked += UnlockAutoCollecting; popup.EventIncomeUpgraded += UpgradeIncome; popup.EventProductionSpeedUpgraded += UpgradeProductionSpeed;

PopupManager.Instance.ShowPopup<ProductionBuildingMenu>(); }

if (m\_currentState is DrawingBuildingState) {

PopupManager.Instance.ShowPopup<ControlArtistPopup>(); }

}

public void OnUnSelected() {

if (m\_currentState is LockedBuildingState) {

m\_productionBuildingView.SetLockedUIActive(false); }

if (m\_currentState is ProductionState || m\_currentState is IncomeState) {

var popup = PopupManager.Instance.GetPopup<ProductionBuildingMenu>(); popup.EventAutoCollectUnlocked -= UnlockAutoCollecting; popup.EventIncomeUpgraded -= UpgradeIncome; popup.EventProductionSpeedUpgraded -= UpgradeProductionSpeed;

} }

public virtual void CollectIncome() {

ProductionProgress = 0;

91

m\_productionBuildingView.DisplayCollectAnimation(() => {

ChangeState(new ProductionState()); });

}

#endregion

#region Private Methods

protected virtual void UnlockBuilding() {

SendUnlockEvent();

AudioClipData audioClipData = ConfigManager.Instance.Get<AudioConfig>().ButtonClickAudioClipData;

AudioManager.Instance.PlaySoundFx(audioClipData);

PopupManager.Instance.ShowPopup<ControlArtistPopup>();

ChangeState(new DrawingBuildingState()); m\_buildingData.isUnlocked = true;

}

protected virtual void UnlockAutoCollecting() {

IsAutoCollectingEnabled = true; SendUnlockAutoCollectEvent();

}

protected virtual void UpgradeIncome() {

m\_income.Upgrade(); m\_buildingData.incomeParameterLevel = m\_income.Level; m\_buildingData.incomeValue = m\_income.Value;

SendUpgradeProductionBuildingEvent(DynamicParameterType.BuildingIncome.ToFriendlyName()); }

protected virtual void UpgradeProductionSpeed() {

m\_productionSpeed.Upgrade(); m\_buildingData.productionSpeedParameterLevel = m\_productionSpeed.Level; m\_buildingData.productionSpeedValue = m\_productionSpeed.Value;

SendUpgradeProductionBuildingEvent(DynamicParameterType.ProductionSpeed.ToFriendlyName()); }

private void OnUpdate(float deltaTime) {

ProductionBuildingState newState = m\_currentState?.HandleState(deltaTime);

92

if (newState != null) {

ChangeState(newState); }

}

private void Initialize() {

CityProgressData cityProgressData = ServiceLocator.Instance.Get<IUserProfileModel>().CurrentCityProgress;

m\_buildingData = cityProgressData.GetProductionBuildingData(m\_buildingConfig.Id); m\_buildingData.currency = (int)CurrencyType;

ParameterFactory parameterFactory = DynamicParametersManager.Instance.ParameterFactory; m\_income = parameterFactory.Get(DynamicParameterType.BuildingIncome,

m\_buildingConfig.IncomeData, m\_buildingData.incomeParameterLevel); m\_productionSpeed = parameterFactory.Get(DynamicParameterType.ProductionSpeed,

m\_buildingConfig.ProductionSpeedData, m\_buildingData.productionSpeedParameterLevel); m\_buildingData.incomeValue = m\_income.Value; m\_buildingData.productionSpeedValue = m\_productionSpeed.Value;

string cityNumber = ServiceLocator.Instance.Get<IUserProfileModel>().CurrentCity; CityData cityData = ConfigManager.Instance.Get<CityConfig>().GetCityData(cityNumber); LevelData levelData = cityData.GetLevelData(Id);

m\_building = levelData.Building;

Sprite buildingSprite = m\_buildingData.isDrawn ? m\_building.CompletedTileSprite : m\_building.DefaultTileSprite;

m\_productionBuildingView.InitBuilding(buildingSprite); }

private void InitializeBuildingState() {

if (!m\_buildingData.isUnlocked) {

ChangeState(new LockedBuildingState()); }

else if (!IsDrawn) {

ChangeState(new DrawingBuildingState()); }

else if(!IsReadyToCollect) {

TeamManager.Instance.BusyArtistsCount -= ArtistsCount; ArtistsCount = 0;

ProductionProgress += TimeManager.Instance.SecondsOffline; ChangeState(new ProductionState());

} else {

ChangeState(new IncomeState()); }

}

93

private void ChangeState(ProductionBuildingState newState) {

m\_currentState?.OnStateExit(); m\_currentState = newState; m\_currentState?.OnStateEnter(this);

}

private void OnIncomeCollected() {

AudioClipData audioClipData = ConfigManager.Instance.Get<AudioConfig>().CollectIncomeAudioClipData;

AudioManager.Instance.PlaySoundFx(audioClipData);

CollectIncome(); }

private void SendClickEvent() {

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.HouseTapped); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.BuildingId,

m\_buildingConfig.Id); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.HouseType,

m\_currentState.Status); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CurrencyType,

CurrencyType.ToString().ToLower());

AnalyticsManager.Instance.SendCustomEvent(analyticEvent); }

private void SendUnlockEvent() {

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.HouseUnlocked);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.BuildingId, m\_buildingConfig.Id);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CurrencyType, CurrencyType.ToString().ToLower());

AnalyticsManager.Instance.SendCustomEvent(analyticEvent); }

private void SendClaimIncomeEvent() {

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.IncomeCollected);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.BuildingId, m\_buildingConfig.Id);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CurrencyType, CurrencyType.ToString().ToLower());

AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

94

}

private void SendUnlockAutoCollectEvent() {

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.UnlockAutoCollect);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.BuildingId, m\_buildingConfig.Id);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CurrencyType, CurrencyType.ToString().ToLower());

AnalyticsManager.Instance.SendCustomEvent(analyticEvent); }

private void SendUpgradeProductionBuildingEvent(string parameterType) {

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.UpgradeProductionBuilding);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.BuildingId, m\_buildingConfig.Id);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CurrencyType, CurrencyType.ToString().ToLower());

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.ParameterType, parameterType);

AnalyticsManager.Instance.SendCustomEvent(analyticEvent); }

#endregion }

public class IArtist : MonoBehaviour, IArtist {

#region Fields

[Header("Components")]

[SerializeField] private ArtistView m\_artistView = null; [SerializeField] private NavMeshAgent m\_navMeshAgent = null;

private GraffitiBlockData m\_graffitiBlockData = null; #endregion

#region Properties

public IPaintSpray PaintSpray { get; set; }

public IArtistState State { get; set; }

public GraffitiBlockData GraffitiBlockData {

get => m\_graffitiBlockData; set

{

m\_graffitiBlockData = value; m\_graffitiBlockData.HasArtist = true;

95

} }

public ArtistView ArtistView => m\_artistView;

public NavMeshAgent NavMeshAgent => m\_navMeshAgent;

public Transform Transform => transform;

public bool IsFinished { get; set; } = false;

#endregion

#region Unity Lifecycle

private void OnEnable() {

UpdateManager.EventOnUpdate += OnUpdate; }

private void OnDisable() {

UpdateManager.EventOnUpdate -= OnUpdate; }

private void Start() {

m\_navMeshAgent.autoTraverseOffMeshLink = false; m\_navMeshAgent.updateRotation = false; ChangeState(new IdleState());

}

#endregion

#region Private Methods

private void OnUpdate(float deltaTime) {

HandleState(); HandleSpeedBonus();

}

private void HandleState() {

UpdateSpeed();

IArtistState state = State?.HandleState();

ChangeState(state);

96

}

private void HandleSpeedBonus() {

bool isBonusActive = BonusManager.Instance.IsBonusActive(BonusType.MovingSpeed);

m\_artistView.SetTrailActive(isBonusActive); }

private void ChangeState(IArtistState state) {

if (state == null) return;

State?.OnStateExit(); State = state; State.OnStateEnter(this);

}

private void UpdateSpeed() {

m\_navMeshAgent.speed = DynamicParametersManager.Instance.Get(DynamicParameterType.RunningSpeed).Value;

}

#endregion }

public class CityManager : Singleton<CityManager>, IManager {

#region Fields

[Header("Options")]

[SerializeField] private LayerMask m\_clickableLayer; [SerializeField] private LayerMask m\_productionBuildingLayer = 0;

private InputService m\_inputService;

private ProductionBuilding m\_selectedProductionBuilding;

private Camera m\_mainCamera;

#endregion

#region Properties

public ProductionBuilding SelectedProductionBuilding => m\_selectedProductionBuilding;

#endregion

#region Unity Lifecycle

private void OnEnable()

97

{

m\_inputService = ServiceLocator.Instance.Get<InputService>(); m\_inputService.eventOnPointerUp += OnPointerUp;

}

private void OnDisable() {

m\_inputService.eventOnPointerUp -= OnPointerUp; }

private void Start() {

m\_mainCamera = FindObjectOfType<CityCameraController>().Camera; TryShowOfflineIncome();

TryShowTutorialOpenBuildingTutorial(); }

#endregion

#region Public Methods

public void Initialize() {

IUserProfileModel userProfileModel = ServiceLocator.Instance.Get<IUserProfileModel>();

string cityNumber = userProfileModel.CurrentCity; AudioClipData audioClipData =

ConfigManager.Instance.Get<CityConfig>().GetCityData(cityNumber).AudioClipData;

AudioManager.Instance.PlayBackgroundAudio(audioClipData);

userProfileModel.CurrentCityProgress.enterTime = DateTime.Now; }

#endregion

#region Private Methods

private void OnPointerUp() {

IUserProfileModel userProfileModel = ServiceLocator.Instance.Get<IUserProfileModel>();

Vector2 mousePosition = Input.mousePosition;

if(m\_inputService.IsDragged || EventSystem.current.IsPointerUnderUI(mousePosition, m\_clickableLayer)) return;

Ray ray = m\_mainCamera.ScreenPointToRay(mousePosition); RaycastHit2D hit = Physics2D.GetRayIntersection(ray, Mathf.Infinity,

m\_productionBuildingLayer);

98

if (hit) {

ProductionBuilding productionBuilding = hit.collider.gameObject.GetComponentInParent<ProductionBuilding>();

if(!IsCanChooseBuilding(productionBuilding)) return;

m\_selectedProductionBuilding?.OnUnSelected(); m\_selectedProductionBuilding = productionBuilding; m\_selectedProductionBuilding.OnSelected(); userProfileModel.OpenedBuildingId = m\_selectedProductionBuilding.Id;

userProfileModel.OpenedBuildingCurrencyType = m\_selectedProductionBuilding.CurrencyType;

if (userProfileModel.TutorialStage == (int) TutorialType.OpenHouse) {

userProfileModel.TutorialStage += 1; }

TryToShowBuyTutorial(productionBuilding); return;

}

m\_selectedProductionBuilding?.OnUnSelected(); }

private bool IsCanChooseBuilding(ProductionBuilding building) {

IUserProfileModel userProfileModel = ServiceLocator.Instance.Get<IUserProfileModel>();

bool isTutorialCompleted = userProfileModel.TutorialStage > (int) TutorialType.BuyHouse; bool isItCheapestBuilding = building.Equals(GetCheapestBuilding());

bool hasFewUnlockedBuildings = userProfileModel.CurrentCityProgress.UnlockedBuildings > 1;

return isTutorialCompleted || isItCheapestBuilding || hasFewUnlockedBuildings; }

private void TryShowOfflineIncome() {

bool isPlayerWasOfflineEnough = TimeManager.Instance.MinutesOffline >= ConfigManager.Instance.Get<GameConfig>().MinMinutesForGetOfflineIncome;

OfflineIncome offlineIncome = (OfflineIncome) ServiceLocator.Instance.Get<IncomeService>().GetIncome(IncomeType.Offline);

if (!ServiceLocator.Instance.Get<IUserProfileModel>().IsFirstSession && isPlayerWasOfflineEnough && offlineIncome.HasIncome)

{

PopupManager.Instance.ShowPopup<OfflineEarningPopup>(); }

}

private void TryShowTutorialOpenBuildingTutorial() {

if(ServiceLocator.Instance.Get<IUserProfileModel>().CurrentCityProgress.UnlockedBuildings > 0) return;

99

TutorialPopup tutorialPopup = PopupManager.Instance.GetPopup<TutorialPopup>(); tutorialPopup.InitializeTutorial(TutorialType.OpenHouse);

tutorialPopup.Show(); }

private void TryToShowBuyTutorial(ProductionBuilding productionBuilding) {

if(ServiceLocator.Instance.Get<IUserProfileModel>().CurrentCityProgress.UnlockedBuildings > 0) return;

TutorialPopup tutorialPopup = PopupManager.Instance.GetPopup<TutorialPopup>(); tutorialPopup.InitializeTutorial(TutorialType.BuyHouse);

tutorialPopup.Show(); }

private ProductionBuilding GetCheapestBuilding() {

var buildings = FindObjectsOfType<ProductionBuilding>(); ProductionBuilding cheapestBuilding = buildings

.Where(building => building.CurrencyType == CurrencyType.Respect) .Aggregate((i1, i2) => i1.UnlockPrice < i2.UnlockPrice ? i1 : i2);

return cheapestBuilding; }

#endregion }

public class AdManager : Singleton<AdManager>, IManager, IAds, IDisposable {

[SerializeField] private AdsConfig m\_adsConfig;

private InterstitialAd m\_interstitial = null; private RewardedAd m\_rewardedAd = null;

private Action m\_onSuccessRewarded = null; private Action m\_onFailRewarded = null;

private string m\_placement = "";

public bool IsInitialized { get; private set; }

public bool IsInterstitialLoaded => m\_interstitial.IsLoaded();

public bool IsRewardedLoaded => m\_rewardedAd.IsLoaded(); public void Initialize()

{

MobileAds.Initialize((status) => { IsInitialized = true; });

LoadInterstitial(); LoadRewarded();

}

100

public void Dispose() {

DisposeInterstitial(); DisposeRewarded();

}

public void LoadInterstitial() {

m\_interstitial = new InterstitialAd(m\_adsConfig.InterstitialAdUnit);

m\_interstitial.OnAdLoaded += OnInterstitialAdLoaded; m\_interstitial.OnAdFailedToLoad += OnInterstitialAdFailedToLoad; m\_interstitial.OnAdOpening += OnInterstitialAdOpen; m\_interstitial.OnAdClosed += OnInterstitialAdClosed; m\_interstitial.OnAdLeavingApplication += OnInterstitialAdLeavingApplication;

AdRequest adRequest = new AdRequest.Builder().Build(); m\_interstitial.LoadAd(adRequest);

}

public void LoadRewarded() {

m\_rewardedAd = new RewardedAd(m\_adsConfig.RewardedAdUnit);

m\_rewardedAd.OnAdLoaded += OnRewardedAdLoaded; m\_rewardedAd.OnAdFailedToLoad += OnRewardedAdFailedToLoad; m\_rewardedAd.OnAdOpening += OnRewardedAdOpen; m\_rewardedAd.OnAdFailedToShow += OnRewardedAdFailedToShow; m\_rewardedAd.OnAdClosed += OnRewardedAdClosed; m\_rewardedAd.OnUserEarnedReward += OnRewardedAdRewarded;

AdRequest adRequest = new AdRequest.Builder().Build(); m\_rewardedAd.LoadAd(adRequest);

}

public bool ShowInterstitial(string placement) {

if (IsInterstitialLoaded) {

m\_placement = placement; m\_interstitial.Show(); return true;

}

Debug.Log("ShowInterstitial: not ready"); return false;

}

public bool ShowRewarded(string placement, Action onSuccess, Action onFail) {

if (IsRewardedLoaded) {

m\_onSuccessRewarded = onSuccess; m\_onFailRewarded = onFail;

101

m\_placement = placement; m\_rewardedAd.Show(); return true;

} onFail?.Invoke();

Debug.Log("ShowRewarded: not ready"); return false;

}

private void DisposeInterstitial() {

m\_interstitial.OnAdLoaded -= OnInterstitialAdLoaded; m\_interstitial.OnAdFailedToLoad -= OnInterstitialAdFailedToLoad; m\_interstitial.OnAdOpening -= OnInterstitialAdOpen; m\_interstitial.OnAdClosed -= OnInterstitialAdClosed; m\_interstitial.OnAdLeavingApplication -= OnInterstitialAdLeavingApplication;

m\_interstitial.Destroy(); }

private void DisposeRewarded() {

m\_rewardedAd.OnAdLoaded -= OnRewardedAdLoaded; m\_rewardedAd.OnAdFailedToLoad -= OnRewardedAdFailedToLoad; m\_rewardedAd.OnAdOpening -= OnRewardedAdOpen; m\_rewardedAd.OnAdFailedToShow -= OnRewardedAdFailedToShow; m\_rewardedAd.OnAdClosed -= OnRewardedAdClosed; m\_rewardedAd.OnUserEarnedReward -= OnRewardedAdRewarded;

}

private void OnInterstitialAdLoaded(object sender, EventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_interstitial\_loaded"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnInterstitialAdFailedToLoad(object sender, AdFailedToLoadEventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_interstitial\_failed"); analyticEvent.AddParameter("placement", m\_placement); analyticEvent.AddParameter("reason", args.Message); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnInterstitialAdOpen(object sender, EventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_interstitial\_opened"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnInterstitialAdClosed(object sender, EventArgs args) {

102

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_interstitial\_closed"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

DisposeInterstitial(); LoadInterstitial();

}

private void OnInterstitialAdLeavingApplication(object sender, EventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_interstitial\_left\_app"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnRewardedAdLoaded(object sender, EventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_reward\_loaded"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnRewardedAdFailedToLoad(object sender, AdErrorEventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_reward\_failed"); analyticEvent.AddParameter("placement", m\_placement); analyticEvent.AddParameter("reason", args.Message); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnRewardedAdOpen(object sender, EventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_reward\_opened"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

}

private void OnRewardedAdFailedToShow(object sender, AdErrorEventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_reward\_failed"); analyticEvent.AddParameter("placement", m\_placement); analyticEvent.AddParameter("reason", args.Message); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

m\_onFailRewarded?.Invoke(); m\_onSuccessRewarded = null; m\_onFailRewarded = null;

}

private void OnRewardedAdClosed(object sender, EventArgs args) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_reward\_closed"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

103

DisposeRewarded(); LoadRewarded();

}

private void OnRewardedAdRewarded(object sender, Reward reward) {

AnalyticEvent analyticEvent = new AnalyticEvent("ad\_reward\_earned"); analyticEvent.AddParameter("placement", m\_placement); AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

m\_onSuccessRewarded?.Invoke(); m\_onSuccessRewarded = null; m\_onFailRewarded = null;

} }

public class DynamicParametersManager : Singleton<DynamicParametersManager>, IManager {

#region Fields

private List<DynamicParameter> m\_dynamicParameters = new List<DynamicParameter>();

private ParameterFactory m\_parameterFactory;

#endregion

#region Properties

public ParameterFactory ParameterFactory => m\_parameterFactory;

#endregion

#region Public Methods

public void Initialize() {

m\_parameterFactory = new ParameterFactory(ServiceLocator.Instance.Get<IUserProfileModel>().CurrentCity);

InitializeDynamicParameters(); }

public DynamicParameter Get(DynamicParameterType type) {

DynamicParameter parameter = default;

for (int i = 0; i < m\_dynamicParameters.Count; i++) {

if (m\_dynamicParameters[i].ParameterType == type) {

return m\_dynamicParameters[i]; }

104

}

return parameter; }

public void DynamicParameterUpdated(DynamicParameterType type, int level) {

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.UpgradeDynamicParameter);

analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.ParameterType, type.ToFriendlyName());

AnalyticsManager.Instance.SendCustomEvent(analyticEvent);

CityProgressData cityProgressData = ServiceLocator.Instance.Get<IUserProfileModel>().CurrentCityProgress;

Dictionary<int, int> dynamicParametersLevels = cityProgressData.dynamicParametersLevels;

dynamicParametersLevels[(int) type] = level; }

#endregion

#region Private Methods

private void InitializeDynamicParameters() {

//Переменные влияющие на геймплей m\_dynamicParameters.Add(m\_parameterFactory.Get(DynamicParameterType.ArtistsCount));

m\_dynamicParameters.Add(m\_parameterFactory.Get(DynamicParameterType.SprayBottleCapacity)); m\_dynamicParameters.Add(m\_parameterFactory.Get(DynamicParameterType.DrawingSpeed)); m\_dynamicParameters.Add(m\_parameterFactory.Get(DynamicParameterType.RunningSpeed));

//Переменные влияющие на доход

m\_dynamicParameters.Add(m\_parameterFactory.Get(DynamicParameterType.RespectIncomePerBlock)) ;

}

#endregion }

public class GraffitiManager : Singleton<GraffitiManager>, IManager {

#region Fields

public event Action EventBlockCompleted; public event Action EventGraffitiCompleted;

private BaseGraffitiDrawer m\_graffitiDrawer;

private ProductionBuildingData m\_productionBuildingData;

#endregion

105

#region Properties

public float Progress => m\_graffitiDrawer.GraffitiData.Progress;

public bool IsGraffitiCompleted => m\_graffitiDrawer.GraffitiData.Progress >= m\_graffitiDrawer.GraffitiData.blocksCount;

#endregion

#region Public Methods

public void Initialize() {

IUserProfileModel userProfileModel = ServiceLocator.Instance.Get<IUserProfileModel>();

m\_productionBuildingData = userProfileModel.CurrentCityProgress.GetProductionBuildingData(userProfileModel.OpenedBuildingId) ;

}

public void SetGraffitiDrawer(BaseGraffitiDrawer graffitiDrawer) {

m\_graffitiDrawer = graffitiDrawer;

m\_graffitiDrawer.EventBlockCompleted += OnBlockCompleted; m\_graffitiDrawer.EventGraffitiCompleted += OnGraffitiCompleted;

}

public void DrawBlock(GraffitiBlockData blockData) {

m\_graffitiDrawer.Draw(blockData);

m\_productionBuildingData.drawProgress = Progress; }

public bool HasEmptyBlock() {

List<GraffitiBlockData> blocks = m\_graffitiDrawer.GraffitiData.BlocksData;

for (int i = 0; i < blocks.Count; i++) {

if (blocks[i].IsComplete || blocks[i].HasArtist) continue;

return true; }

return false; }

106

public GraffitiBlockData GetEmptyBlock() {

List<GraffitiBlockData> blocks = m\_graffitiDrawer.GraffitiData.BlocksData;

for (int i = 0; i < blocks.Count; i++) {

if (blocks[i].IsComplete || blocks[i].HasArtist) continue;

return blocks[i]; }

return null; }

public Vector3 GetBlockPosition(GraffitiBlockData blockData) {

int blockIndex = blockData.blockID;

int blocksX = m\_graffitiDrawer.GraffitiData.BlocksX; int blocksY = m\_graffitiDrawer.GraffitiData.BlocksY; int blockXIndex = (blockIndex % blocksX);

int blockYIndex = ((blockIndex - 1 ) / blocksX);

float m\_halfOfBlockWidth = (1f / blocksX) / 2f;

float xTime = Mathf.Clamp01((float) blockXIndex / blocksX + m\_halfOfBlockWidth); float yTime = (float) blockYIndex / blocksY;

return m\_graffitiDrawer.Holst.GetBlockPosition(xTime, yTime); }

#endregion

#region Private Methods

private void OnBlockCompleted() {

EventBlockCompleted?.Invoke(); ServiceLocator.Instance.Get<IncomeService>().GiveIncomeFor(IncomeType.PerPixel); ParticleManager.Instance.SpawnParticle(ParticleType.Income, m\_graffitiDrawer.Holst.Transform);

}

private void OnGraffitiCompleted() {

SendDrawnEvent();

m\_productionBuildingData.isDrawn = true;

ParticleManager.Instance.SpawnParticle(ParticleType.Confetti, FindObjectOfType<GraffitiAreaTile>().ConfetiSpawnPos);

DOTween.Sequence().SetDelay(ConfigManager.Instance.Get<GameConfig>().DelayBeforeEndGame) .OnComplete(() =>

107

{ LevelManager.Instance.Building.HideScaffolding(ConfigManager.Instance.Get<GameConfig>().Scaffol dingHideDuration);

EventGraffitiCompleted?.Invoke(); });

ScreensManager.Instance.GetScreen<GameScreen>().TryToShowBackButtonTutorial(); }

private void SendDrawnEvent() {

IUserProfileModel userProfileModel = ServiceLocator.Instance.Get<IUserProfileModel>(); DynamicParametersManager dynamicParametersManager = DynamicParametersManager.Instance;

AnalyticEvent analyticEvent = new AnalyticEvent(StringConstants.AnalyticsEvents.DrawnGraffiti); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.BuildingId,

userProfileModel.OpenedBuildingId); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CountOfArtists,

dynamicParametersManager.Get(DynamicParameterType.ArtistsCount).Value); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.MorePerBlock,

dynamicParametersManager.Get(DynamicParameterType.RespectIncomePerBlock).Level); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.RunningSpeed,

dynamicParametersManager.Get(DynamicParameterType.RunningSpeed).Level); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.DrawingSpeed,

dynamicParametersManager.Get(DynamicParameterType.DrawingSpeed).Level); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CapacityOfCans,

dynamicParametersManager.Get(DynamicParameterType.SprayBottleCapacity).Level); analyticEvent.AddParameter(StringConstants.AnalyticsEventsParameters.CurrencyType,

userProfileModel.OpenedBuildingCurrencyType.ToString().ToLower()); }

#endregion }

108

Обозначение

БГУИР ДП 1-58 01 01 010 ПЗ

ГУИР.508700.001 ПД

ГУИР.508700.002 ПД

ГУИР.508700.003 ПД

ГУИР.508700.004 ПД

ГУИР.508700.005 ПЛ

ГУИР.508700.006 ПЛ

Наименование

Текстовые документы Пояснительная записка

Отзыв руководителя Рецензия

Графические документы Блок-схема алгоритма работы пользователя

Диаграмма классов

Диаграмма вариантов использования

Структурная схема

Основные экранные формы

Расчёт игрового баланса

Дополнитель-ные сведения

109 с. 1 лист 1 лист

Формат А1 1 лист

Формат А1 1 лист

Формат А1 1 лист

Формат А1 1 лист

Формат А1 1 лист

Формат А1

1 лист

БГУИР ДП 1-58 01 01 010 Д1 Изм Лист N докум. Подп. Дата

Разраб. Проверил Т.контр. Н.контр.

Утв.

Евтух Малинина Малинина Малинина

Яшин

Мобильное приложение

для 3D-игры на Unity

и его испытание

Ведомость дипломного проекта

Лит. Лист Листов Т 109 109

Кафедра ИПиЭ гр.780961