

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Научный руководитель, преподаватель
департамента программной инженерии
факультета компьютерных наук

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____ О.В. Максименкова
«___» _____ 2018 г.

_____ В.В. Шилов
«___» _____ 2018 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

МОБИЛЬНАЯ AR-ИГРА В ЖАНРЕ «TOWER DEFENCE»

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729. 04.01-01 ПЗ 01-1

Исполнитель
студент группы БПИ 177

_____ / Н.А. Самойлов/
«___» _____ 2018 г.

Москва 2018

УТВЕРЖДЕН
RU.17701729.04.01-01 ПЗ 01-1

МОБИЛЬНАЯ AR-ИГРА В ЖАНРЕ «TOWER DEFENCE»

Пояснительная записка

RU.17701729.04.01-01 ПЗ 01-1

Листов 20

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Москва 2018

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «Protect the reality» («Мобильная AR-игра в жанре «Tower Defence»»). Данный документ содержит разделы «Введение», «Назначение и область применения программы», «Технические характеристики», «Ожидаемые технико-экономические показатели», «Источники, использованные при разработке»

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы;
- описание алгоритма и функционирования программы
- описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [3];
 - 2) ГОСТ 19.102-77 Стадии разработки [4];
 - 3) ГОСТ 19.103-77 Обозначения программ и программных документов [5];
 - 4) ГОСТ 19.104-78 Основные надписи [6];
 - 5) ГОСТ 19.105-78 Общие требования к программным документам [7];
 - 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [8];
 - 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [9].
- Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [10], ГОСТ 19.604-78 [11].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
Содержание	3
1. ВВЕДЕНИЕ	4
1.1. Наименование программы	4
1.2. Документы, на основании которых ведется разработка	4
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	5
2.1. Функциональное назначение	5
2.2. Эксплуатационное назначение	6
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	7
3.1. Постановка задачи	7
3.2. Описание алгоритма и функционирования программы	7
3.2.1. Общая структура работы приложения	7
3.2.2. Реализация взаимодействия пользователя с ARKit.	8
3.2.2.1. Функционирование «prepare_scene_1»	9
3.2.2.2. Функционирование «base_scene»	9
3.2.3. Структура уровня	10
3.2.3.1. Враги	10
3.2.3.2. Башни	10
3.2.3.3. Управление волнами врагов	12
3.2.3.4. Десериализация конфигураций уровня	12
3.2.3.5. «ProjectilesManager»	12
3.2.3.6. Покупка, продажа и «апгрейд» башен	13
3.3. Описание и обоснование выбора состава технических и программных средств	14
3.3.1. Состав технических и программных средств	14
3.3.2. Обоснование выбора состава технических и программных средств	14
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	15
4.1. Предполагаемая потребность	15
4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	15
5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16
ПРИЛОЖЕНИЕ 1	17
Описание и функциональное назначение классов	17
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Protect the reality».

1.2. Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1212-01 от 12.12.2017 «Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук».

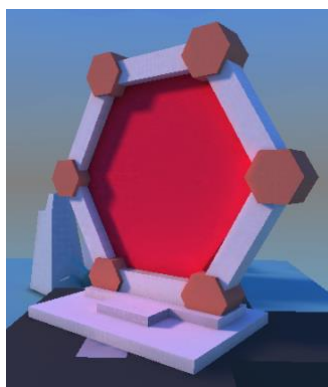
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

Функциональным назначением программы является наложение 3D модели игрового поля на получаемое в реальном времени со встроенной в мобильное устройство камеры изображение. Также программа предоставляет возможность взаимодействия с игровым полем посредством нажатия игроком на сенсорный экран.

Правила игры: Задача игрока защитить красный портал (рис.1а), расправляясь с волнами наступающих из зеленого портала врагов (рис.1б), с помощью покупки башен (рис. 1в), атакующих их, когда те проходят рядом (рис. 1г).



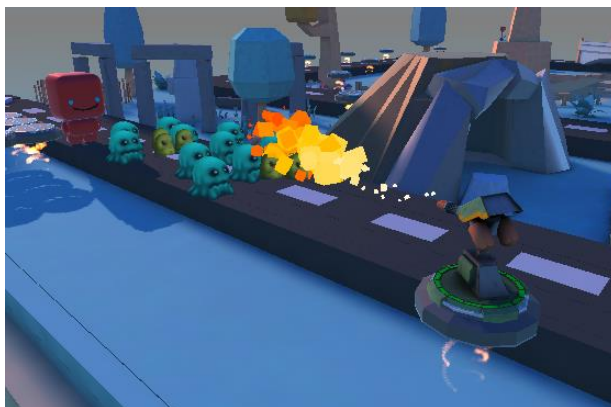
а)



б)



в)



г)

Рисунок 1. Пример игрового процесса. а) – красный портал; б) – зелёный портал; в) – магазин башен; г) – башня атакует врагов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2.2. Эксплуатационное назначение

Программа может быть использована на досуге для развлечения на смартфонах и планшетах на базе IOS.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи

Мотивом создания данного продукта послужило желание предоставить пользователям новый взгляд на игры жанра «Tower Defence». Для этого необходимо:

1. Разработать удобный и красивый пользовательский интерфейс.
2. Обеспечить быстроту работы программы.
3. Разработать архитектуру ПО, которую будет удобно поддерживать и развивать.
4. Смоделировать игровые элементы.

3.2. Описание алгоритма и функционирования программы

3.2.1. Общая структура работы приложения

Общую структуру работы программы и возможное взаимодействие классов между собой можно понять, рассмотрев работу загрузчика сцен.

При запуске программы загружается сцена «Manager», в которой есть объект со скриптами «AppManager» и «SceneLoader» (рис. 2), работу которых мы и рассмотрим.

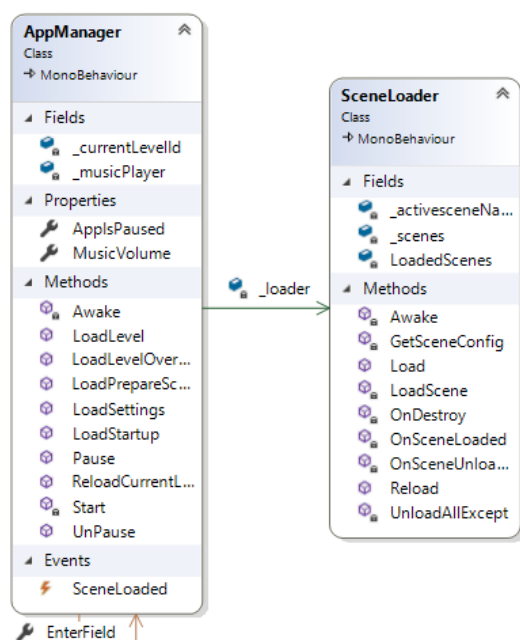


Рисунок 2. Диаграмма классов объекта «ApplicationManager» сцены «Manager».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

«AppManager» предоставляет интерфейс управления базовыми функциями: загрузка сцен, пауза и управление звуком. За загрузку сцен отвечает инкапсулированный класс «SceneLoader», который читает файл «scene_config.json» из папки «Resources» с конфигурациями сцен и безопасно управляет их поведением. В json-файле сериализованы объекты типа «SceneConfig»:

```
[Serializable]
public class SceneConfig
{
    [Serializable]
    public struct Helper
    {
        public string SceneName;
    }

    public string SceneName;

    public bool SetActive; //if it's true scene will be active after loading

    public Helper[] helpers; // what we will need to load with this scene.
}
```

Каждый из которых представляет собой параметры сцены и массив других сцен, без которых данная не может полноценно работать.

В связи с данной реализацией и невозможностью передачи ссылок между сценами во множестве классов, в том числе в «AppManager» (рис.2), будет использоваться паттерн «Singleton».

Краткий обзор каждой из сцен:

«Manager» - управляет загрузчиком. (не участвует в сериализации).

«base_scene» - управляет ARKit сессией, содержит в себе камеры.

«game_gui» - содержит в себе игровой интерфейс.

«prepare_scene_1» - занимается нахождением ARKit поверхности, на которой образуется игровое поле. (загружается с «base_scene»)

«level_1» - первый уровень игры. (загружается с «base_scene» и «game_gui»)

«startup» - главное меню. (загружается после запуска программы)

«SettingsScene» - меню настроек.

«Level_overview» - меню выбора уровня.

3.2.2. Реализация взаимодействия пользователя с ARKit.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

После того как пользователь выберет уровень, в который он хочет сыграть, происходит загрузка «prepare_scene_1», и вместе с этим передается id уровня в статическую переменную класса «PrepareLevelControl». (В связи с этим, данную сцену можно использовать для любого другого уровня, не тратя времени на создание новой.) Далее идет поиск подходящей поверхности, о чем пользователю сообщает движущаяся иконка по центру экрана. (рис. 3а) После того, как поверхность найдена, она визуализируется и на ней появляется объект, несущий в себе информацию о положении и размере игрового поля. (рис. 3б) Взаимодействуя с интерфейсом, пользователь выбирает размер, положение объекта и нажимает кнопку «Play», и происходит загрузка сцена уровня.



Рисунок 3. а) – поиск поверхности; б) - настройка объекта

3.2.2.1. Функционирование «prepare_scene_1»

Перед тем как загрузится сцена уровня, класс «PrepareLevelControl» обращается к классу «PositionSaver», что задает новое положение камер таким образом:

новое положение камеры = положение камеры – положение объекта настройки.

Камеры становятся таким образом что кажется, будто мы передвигали сцену уровня. Такое решение принято в связи с тем, что объекты с пометкой «static» имеют две особенности:

- 1) Выгодны с точки зрения производительности.
- 2) Недвижимы во время работы программы.

3.2.2.2. Функционирование «base_scene»

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Функционирование данной сцены основано на том, что одна из камер («Main Camera») захватывает изображение с камеры IOS устройства, а другая («ScaledCamera») всё остальное. Так решение сводит на нет проблему с изменением размера ARKit сцены [1].

3.2.3. Структура уровня

Краткий обзор объектов сцены:

- 1) «LevelControl» - содержит в себе все скрипты, необходимые для работоспособности сцены.
- 2) «WorldSpaceCanvas» - служит родительским объектом для UI элементов, которые создаются в игровом мире. (например рис. 1в)
- 3) «Nodes» - содержит в себе объекты для построения пути движение врагов.
- 4) «Platforms» - содержит в себе объекты для создания башен.
- 5) Другие – в основном «static» объекты игрового поля и его окружения.

3.2.3.1. Враги

Рассмотрим базовые компоненты объекта врага:

- 1) «BoxCollider» - нужен для работы «EnemyLifeCycle».
- 2) «NavMeshAgent» - нужен для работы «Agent».
- 3) «Animator» - нужен для переключения анимаций ходьбы.
- 4) «Rigidbody» - нужен для работы «NavMeshAgent».
- 5) «Agent» - управляет передвижением врага.
- 6) «EnemyLifeCycle» - управляет количеством здоровья.

Функциональность врага начинается и заканчивается на том, что он идет от метки к метке попутно собирая в свой «BoxCollider» различные снаряды построенных игроком башен.

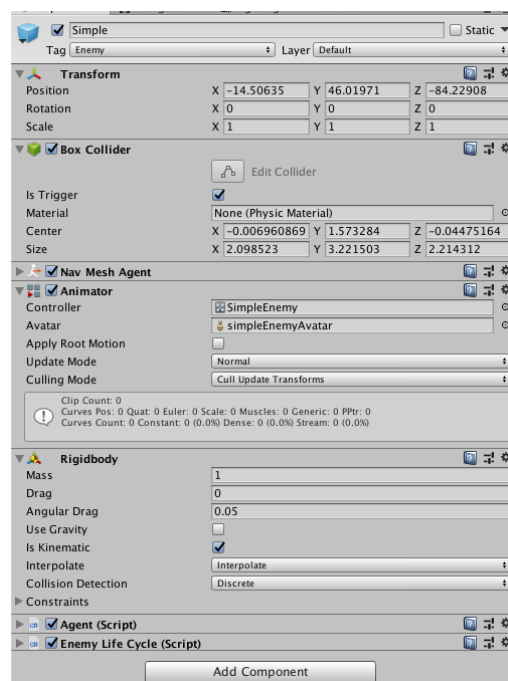


Рисунок 4. Компоненты врага

3.2.3.2. Башни

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Рассмотрим базовые компоненты объекта башни (рис. 5):

- 1) «SphereCollider» - обозначает область действия класса «Targetter».
- 2) «SimpleShooter» - один из наследников абстрактного класса «Shooter», представляет собой базовые функции стрельбы по врагам.
- 3) «TowerData» - содержит в себе изменяемую информацию об уроне, времени перезарядки и дальности стрельбы башни.
- 4) «Targetter» - находит и предоставляет данные о враге в радиусе «SphereCollider» и передает его «Shooter».

Диаграмма связей между классами показана на рис.6.

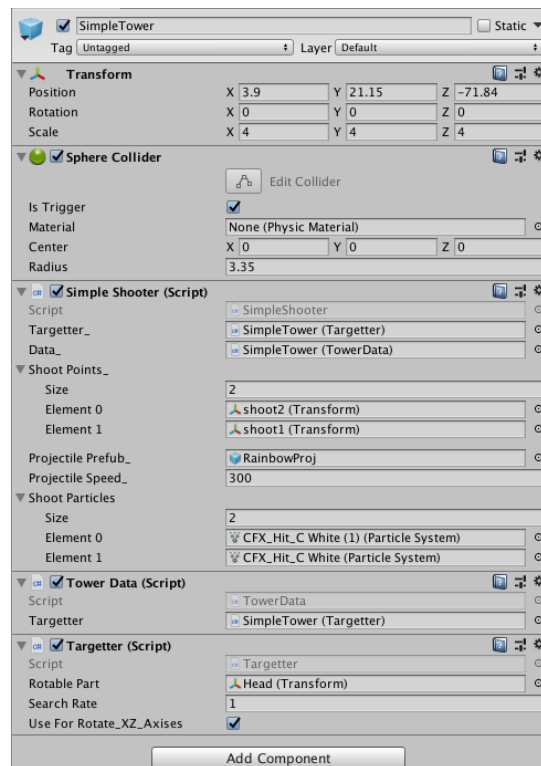


Рисунок 5. Компоненты башни.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

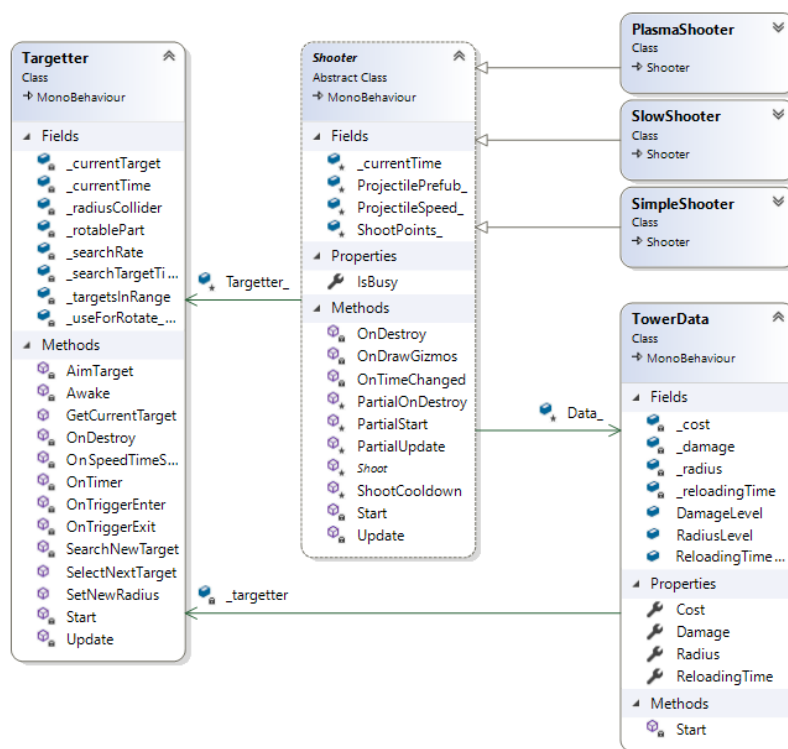


Рисунок 6. Диаграмма классов башни.

3.2.3.3. Управление волнами врагов

Класс «WavesManager» занимается тем, что в соответствии с конфигурацией (о ней речь пойдет позже) инициализирует волны врагов на игровое поле, а также сообщает о состояниях волн.

3.2.3.4. Десериализация конфигураций уровня

Класс «DataStorage» десериализует конфигурации башен, врагов и волн врагов и предоставляет к ним глобальный доступ, что облегчает получение информации различными классами и не нагромождает исходный код.

3.2.3.5. «ProjectilesManager»

Данный менеджер занимается тем, что рассчитывает положение каждого снаряда, перемещающегося на сцене и, в соответствии с парадигмой «Data Oriented Design (DOD)», приветствуемой самой компанией Unity [2], оказывается намного производительней управления положением из скрипта самого снаряда.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.3.6. Покупка, продажа и «апгрейд» башен

Покупка башен происходит по следующему алгоритму:

- 1) Игрок выбирает платформу и нажимает на нее.
- 2) Всплывает меню покупки.
- 3) Пользователь выбирает нужную ему башню и нажимает на соответствующую иконку.

Продажа и апгрейд башни происходит по следующему алгоритму:

- 1) Игрок выбирает башню и нажимает на нее.
- 2) Всплывает меню продажи и «апгрейда».
- 3) Пользователь выбирает нужную ему опцию и нажимает на соответствующую иконку.

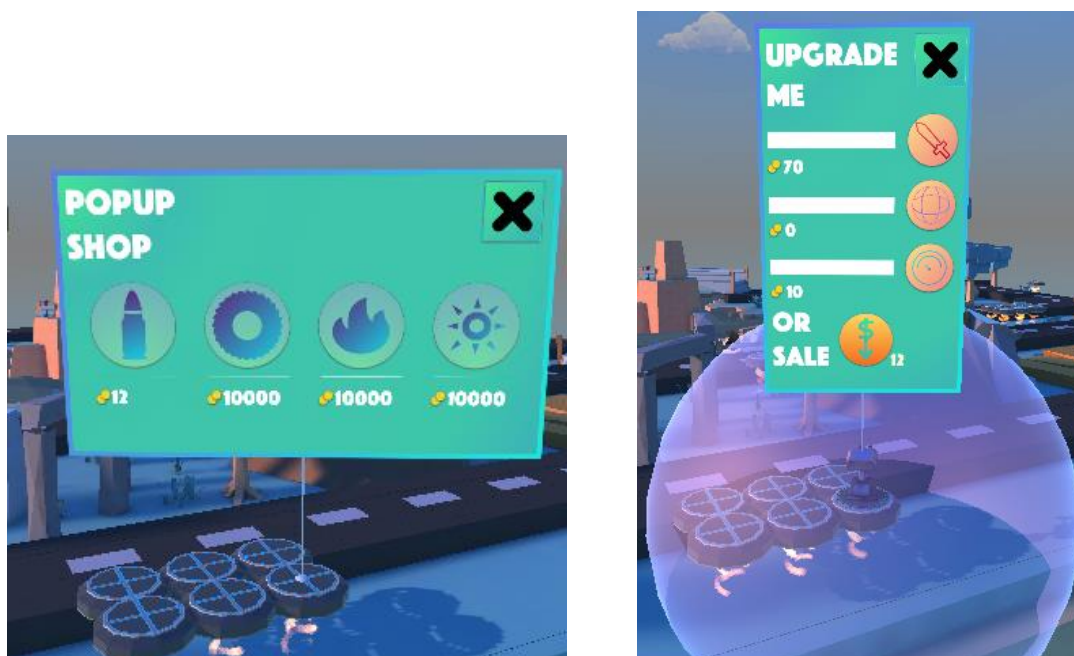


Рисунок 7. Интерфейс покупки, продажи и «апгрейда» башни.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование выбора состава технических и программных средств**3.3.1. Состав технических и программных средств**

Для работы программы необходимо устройство новее iPhone 5s или iPad (5-ого поколения), работающее на IOS11 или более поздних версиях ОС.

3.3.2 Обоснование выбора состава технических и программных средств

На старте разработки данного продукта, только ARKit-plugin для Unity вышел из стадии бета-тестирования.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Приложение будет востребовано, так как мобильные игры все больше набирают популярность в наше время и такое молодое направление как AR-игры безусловно привлечет игроков.

4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Существует множество аналогов данного продукта, но большинство из них не реализованы в дополненной реальности, а те, что используют данную технологию имеют ряд недостатков, связанных с тем, что идентификация поверхности происходит с помощью определенного рисунка на листе бумаги.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Unity at GDC – Job System & Entity Component System [Электронный ресурс] / Unity – Электрон. дан. - 2018 – Режим доступа: <https://youtu.be/kwnb9Clh2Is> , свободный. – Загл. С экрана.
2. Dealing with Scale in AR [Электронный ресурс] / Rick Johnson – Электрон. дан. - 2016 – Режим доступа: <https://blogs.unity3d.com/ru/2017/11/16/dealing-with-scale-in-ar/>, свободный. – Загл. С экрана.
3. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
4. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
5. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
6. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
7. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
8. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
9. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
10. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
11. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

Описание и функциональное назначение классов

Таблица 1 – Описание и функциональное назначение классов

Класс	Назначение
PositionSaver	Класс, отвечающий за сохранение положения камеры.
HealthBarController	Класс, отвечающий за отображение здоровья игрока.
LevelCompleteControl	Класс, отвечающий за отображения сообщений об окончании уровня
MoneyCounterController	Класс, отвечающий за отображение количество денег игрока.
PauseButtonControl	Класс, отвечающий за функционирование кнопки паузы.
PauseWaveButtonController	Класс, отвечающий за функционирование кнопки паузы волны.
PopUpShopController	Класс, отвечающий за функционирование магазина башен.
SettingsButton	Класс, отвечающий за функционирование кнопки изменения качества теней и текстур.
SoundButton	Класс, отвечающий за функционирование кнопки изменения громкости.
SpeedButtonControl	Класс, отвечающий за функционирование кнопки изменения скорости волны врагов.
UpgradeMenu	Класс, отвечающий за функционирование меню продажи и апгрейда башни.
WavenumberControl	Класс, отвечающий за отображение номера башни.
WorldSpaceCanvas	Класс, отвечающий за работоспособность «WorldSpaceCanvas».
EnemyConfig	Класс, созданный для сериализации данных о враге.
EnemyCounter	Класс, отвечающий за подсчет кол-ва врагов на сцене и оповещения об изменении их числа.
EnemyLifeCycle	Класс, отвечающий за здоровье вражеских существ.
Platform	Класс, отвечающий за работоспособность платформы под башню.
ProjectileData	Класс, содержащий информацию о снаряде.
ProjectilesManager	Класс, отвечающий за управлением траекторией полета всех снарядов на сцене.
SlowProjectile	Класс, отвечающий за специальную функциональность снаряда башни замедления.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

18
RU.17701729.04.01-01 ПЗ 01-1

ClickableTower	Класс, отвечающий за обработку нажатия игрока и инициализации меню продажи и апгрейда.
FireCaster	Класс со специальной реализацией огненной стрельбы.
PlasmaShooter	Класс наследник от "Shooter". Реализует стрельбу плазменной башни.
Shooter	Класс реализует базовый функционал нанесения урона вражеским существам.
SimpleShooter	Класс наследник от "Shooter". Реализует стрельбу обычной башни.
SlowShooter	Класс наследник от "Shooter". Реализует стрельбу замедленной башни.
Targetter	Класс реализует функционал нахождения врагов в радиусе доступа.
Int_Upgrade	Апгрейд для «TowerConfig»
Float_Upgrade	Апгрейд для «TowerConfig»
TowerConfig	Класс, созданный для сериализации информации о башне.
TowerData	Класс, содержащий в себе изменяемую информацию о башне.
DataStorage	Класс, предоставляющий доступ к различным конфигурациям
Economy	Класс, занимающийся денежной стороной игры.
NavigationNode	Класс, обеспечивающий работоспособность меток для движение вражеский существ.
PlayerHealth	Класс, контролирующий здоровье игрока.
MiniWaveConfig	Класс, созданный для сериализации конфигурации волн врагов.
WaveConfig	Класс, созданный для сериализации конфигурации волн врагов.
TimeManager	Хранит состояние игрового времени.
WavesManager	Занимается инициализацией волн врагов.
AppManager	Класс, управляющий загрузкой сцен, звуком, паузой.
SceneLoader	Реализация загрузчика сцен.
SceneConfig	Класс, созданный для сериализации конфигурации сцен.
CameraScale	Класс, занимающийся контролем за масштабируемой камерой.
GeneratePlanes	Визуализирует AR-поверхность.
ARHit	Реализация функционала «удара» по AR-поверхности.
MyARSession	Реализация функционала AR-сессии.
DestroyParticleAfterPlay	уничтожение системы частиц после проигрывания.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

19
RU.17701729.04.01-01 ПЗ 01-1

JsonHelper	Вспомогательный класс для сериализации массива объектов.
LookAtCamera	Класс для изменения стороны, в которую «смотрит» объект.
LevelOverviewManager	Управление сценой выбора уровня.
PrepareLevelControl	Управление сценой поиска Ar-поверхности.
StartupCameraMovement	Управление перемещением камеры на начальной сцене.
StartupManager	Управление стартовой сценой.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата