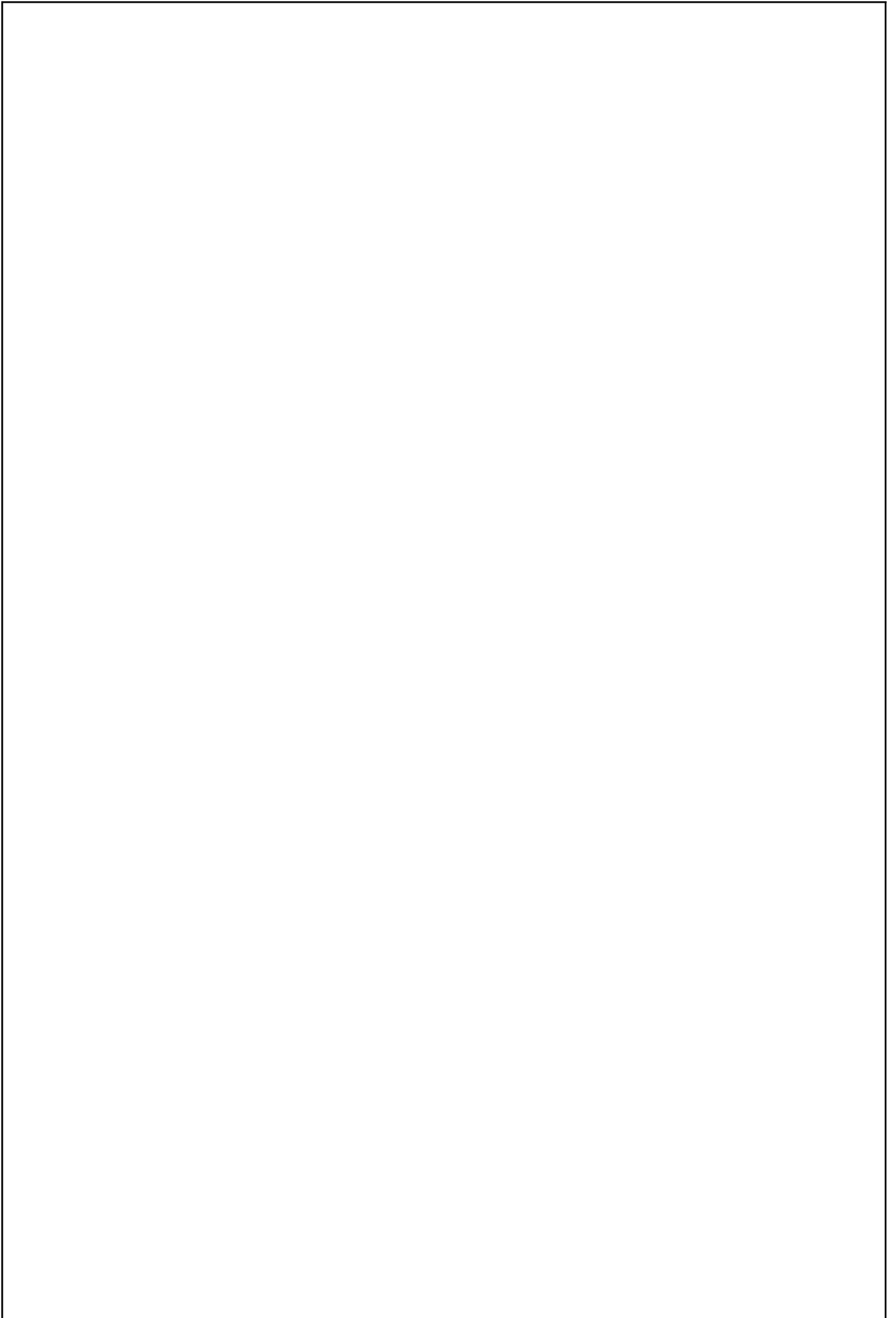


Фаткуллин А.А.

405

# ДИПЛОМНАЯ РАБОТА

2021



**Министерство образования и науки Республики Татарстан**

**Государственное автономное профессиональное образовательное учреждение**

**«Тетюшский государственный колледж гражданской защиты»**

ПЦК математики и естественно-научных  
дисциплин

ВКР соответствует установленным  
требованиям и направлениям ГАК для  
защиты

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

Председатель \_\_\_\_\_ В.Г. Мальгин

**ДИПЛОМНАЯ РАБОТА**

Разработка игры для мобильных устройств

на базе Android на Unity

Выполнил: Фаткуллин Айрат

Арсланович

специальность 09.02.05

«Прикладная информатика»

Курс IV, группа 405

Руководитель: Паулкин К.А.

Рецензент: Зубарькова Н.С.

Защита состоялась

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Оценка \_\_\_\_\_

Тетюши, 2021

## СОДЕРЖАНИЕ

Введение.....	3
Глава 1. Мобильные игры в нашей жизни.....	7
1.1 Полезность игр для человека.....	7
1.2 Сегменты рынка мобильных игр.....	8
1.3 Обзор существующих мобильных платформ.....	9
1.4 Описание предметной области.....	11
1.5 Сравнение сред Сравнение Unity и Construct 2.....	25
Глава 2. Разработка мобильного приложения.....	28
2.1 Обзор программного обеспечения.....	28
2.2 Разработка игрового приложения.....	30
2.3 Перенос на Android.....	40
Заключение.....	43
Список литературы.....	45
Приложение 1.....	48

Подп. и дата

Инв. № дубл

Взим. инв. №

Подп. и дата

**ТГКГЗ. ПИ.22.00.000.КР**

Изм	Лист	№ докум.	Подп.	Дата

Инв. №

Разраб.

Фаткуллин

Пров.

Паулкин

Н.контр

Утв.

*Дипломная работа*

Лит.	Лист	Листов
	2	

**09.02.05, 305**

## ВВЕДЕНИЕ

Человек, телефон и игры – эта связь появилась ещё, когда выходили первые кнопочные телефоны, на которых была всеми узнаваемая «Змейка», с тех пор и начала расти популярность мобильных игр. С каждым новым выпущенным телефоном производители ставили в виде предустановок на него игры, как старые, так и новые. Игры менялись, по графике, по управлению и возможностям внутри игры, это зависело от самих телефонов и средств разработки. С появлением первых сенсорных телефонов, начали появляться и новые игры, управление перешло с кнопок на касания пальцем, нажатия и взмахи. С появлением компаса в телефонах, следственно гироскопа, появилась возможность управления наклоном телефона - например такие игры как Doodle Jump. Разработчики игр быстро осваивали новую среду, а людям нравилось играть при этом не сидя за компьютером дома. С прогрессом роста производительности смартфонов, игры взяли ещё больший горизонт возможностей. На смартфоны не только стали портировать компьютерные игры, но так же стал возможным запуск эмуляторов для запуска игр с консолей, а общее все возможное количество мобильных игр стало невероятно огромным.

В наши дни люди могут играть в очереди, коротая время до встречи или в ванной, при этом играть можно, как в ту же «змеюку», так и в 3D шутеры от первого лица. Люди оценили такую возможность расслабиться, игры помогают отвлечься или же просто скоротать время. На данный момент развитие смартфонов в аппаратной части очень быстро набирает темпы, уменьшается техпроцесс изготовления процессоров, на меньшей площади кристалла располагается больше транзисторов, следовательно, растёт производительность. Так же растут и частоты работы процессора, современные процессоры способны работать на частоте до 3.1GHz, что выше некоторых процессоров для ПК.

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	3
Изм.	Лист	№ документа	Подпись	Дата		

Так же графический модуль в новых процессорах способен поддерживать графику компьютерного уровня в разрешение FHD, UHD и в некоторых случаях. Развитие графической стороны в приложениях немало важно, пользователям нравится красивая картинка. На телефоны уже портировали такие масштабные игры как: PUBG, Call of Duty, Fortnite. Разберём каждую игру по отдельности, почему же они интересны в данном случае. Начнём с PUBG и Fortnite, это главные представители жанра BattleRoyal, это игры где на одной карте происходит сражение с участием 100 человек по принципу: каждый сам за себя (1vs100); отряды из двух человек (50 команд по 2 человека в команде); отряды из четырех человек (25 команд по 4 человека). Главная задача победить всех и остаться единственным выжившим. При этом существует безопасная зона, которая уменьшается в каждый определенный отрезок времени и с каждым этапом сужается быстрее, а урон за пределами безопасной зоны становится больше. И вот уже 2 года на смартфонах доступен PUBG Mobile и по 1 году Call of Duty: Mobile и Fortnite Mobile. Эти 3 проекта позволяют наглядно увидеть, как далеко продвинулось развитие смартфонов и мобильных игр, на столько, что человек с телефона может поучаствовать в битве из 100 человек. Мобильный игры не просто стали объединять людей по интересам, они так же сохранили соревновательный интерес, это все возможные таблицы лидеров, разделение на лиги по достигнутым очкам. На основе этого киберспорт перешёл и в мобильный гейминг, уже существует достаточное количество дисциплин: PUBG Mobile; Call of Duty:Mobile; Fortnite; World of Tanks Blitz; Mobile Legends Bang Bang; Mobile Legends Wild Rift; Arena of Avalor; Hearthstone (mobile). Дисциплины хоть и «малые» братья от оригиналов на ПК, но бюджет в них не малый. Рынок мобильных игр далеко не бедный, не меньше чем у консолей и ПК. А количество участников в турнирах на мобильных устройствах достигает до 2 млн. человек.

					ТГКГЗ.ПИ.22.00.000.КР	Лист
						4
Изм.	Лист	№ документа	Подпись	Дата		

Теперь про интерес пользователей, мы обсудили масштабные проекты, но это не значит, что все играют только в эти игры, рассматривая список популярных игр по загрузкам, видно такую картину, что людям нравятся аркады. Именно аркады составляют  $\approx 70\%$  списка топ 100 по загрузкам в Play Market. Многие аркады так называемые «таймкиллеры» (time-время, kill-убить), игры для «убийства» времени. Это игры зачастую подразумевают управление игровым процессом одним пальцем (удобно в очереди, автобусе). Такие игры очень выгодны для обеих сторон, для разработчика и для пользователя. Они не требуют много времени на разработку, но при этом способны приносить доход с помощью рекламы и встроенных покупок. Если на счёт рекламы всё понятно, то на счёт покупок немного стоит разъяснить. В таких играх зачастую есть одна основная валюта, которую можно зарабатывать, играя, но кто не хочет ждать может её купить, либо может быть валюта, получение которой возможно только при покупке за реальные деньги (в редких случаях можно получить в каком либо внутри игровом событии), а эта валюта позволяет покупать дополнительные внутри игровые предметы. Так же есть тип внутри игровых покупок как «отключение рекламы» и в последнее время популярный «сезонный пропуск», который даёт право получать особые награды за достижения нового уровня.

**Актуальность:** Выбранная нами тема актуальна на протяжении многих лет, с самого начала развития и роста популярности мобильных устройств по сей день, развитие смартфонов не стоит на месте, как и развитие мобильных игр. А интерес пользователей растёт в геометрической прогрессии.

**Цель** дипломной работы - разработать игру для мобильных устройств на базе Android на Unity

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	5
Изм.	Лист	№ документа	Подпись	Дата		

**Задачи:**

1. Изучить влияние игр на жизнь людей;
2. Изучить сегменты рынка мобильных игр;
3. Изучить среды разработки приложения для Android;
4. Разработать игру для мобильных устройств на базе Android в программе Unity.

					ТГКГЗ.ПИ.22.00.000.КР	Лист
						6
Изм.	Лист	№ документа	Подпись	Дата		



# ГЛАВА 1. МОБИЛЬНЫЕ ИГРЫ В НАШЕЙ ЖИЗНИ

## 1.1 Полезность игр для человека

Обсудим влияние игр на человека в целом, ведь можно бесконечно спорить, приводя примеры за и против, потому что существуют громкие примеры, как в пользу игр, так и против. Против игр ситуация куда более спорная, в основном все приводят примеры когда игры служат триггерами для нездоровой психики человека. Например, когда подростки убивают своих сверстников, а потом обнаруживается что они играли в какой-нибудь шутер. Но тут всё просто, миллионы людей играют в игры и никто из них не идёт убивать людей, угонять машины, ведь это идёт от начальной психи, я не отрицаю что можно ограничить доступ к играм, путём только лицензионного распространения и с соблюдением возрастных ограничений. Вернёмся к положительному, так как всё отрицательное это то, во что люди хотят верить. Игр на Android множество, они делятся на множество жанров и поджанров. Самые популярные жанры: аркады, гонки, шутеры, головоломки, симуляторы. Так как речь зашла о пользе, рассмотрим головоломки и аркады для детей. Головоломки хорошо заставляют подумать над прохождением той или иной загадки, а некоторые переросли в серию, как например The Room, это самая известная головоломка в которой вам предстоит не просто разгадывать секреты, но и следовать сюжетной линии. А детские аркады помогают детям развивать логику, реакцию и моторик, как пример «Мой говорящий Том». Теперь о пользе не зависимо от возраста. Игры помогают людям отвлечься от повседневных проблем либо просто расслабиться, скоротать время или же провести его с друзьями. На счёт друзей, игры так же хорошо объединяют людей разных возрастов.

Люди кооперируются в группы и учатся действовать сообща, не редко после совместных игр начинают общение на разные темы, как в игре, так и за пределами в социальных сетях. Хорошо это заметно в командных шутерах и МОБА играх. МОБА используют вариант сражений 5 на 5, и порой люди

									Лист
									7
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

регулярно собираются в команды из 5 человек, что бы побеждать, так как в таких играх нужны не только индивидуальные навыки, но и общение между командой и слаженные действия. Так же МОБА игры хорошо развивают память и мышление, так как там более 100 предметов со своими характеристиками и игроки запоминают какие предметы им нужны, какие предметы лучше использовать и на каком герою, ведь в таких играх порядка 50-100 героев так же со своими особенностями. Как итог в играх больше пользы, чем вреда, да они забирают время, но выбор каждого, чем заняться в свободный час, два. И игры могут стать способом заработка, как уже говорилось, есть киберспорт, турниры и бюджет этих турниров приближается стремительно к отметке 1000000\$.

Так же все люди будучи детьми учатся и развиваются играя. Детские игры развивают моторику, мышление, коммуникативность. Человек играет с самого детства.

## 1.2 Сегменты рынка мобильных игр

На данный момент рынок мобильных игр разбит на аркады, шутеры, гонки, симуляторы и головоломки. В каждом из этих сегментов есть свои главные представители. Аркады на любой вкус разрабатывает компания «Voodoo», их игры не претендуют на место в вашем телефоне на год или два, и сами разработчики это понимают, поэтому выпускают минимум один новый проект каждые две-три недели. В головоломках лидер «Fireproof Games» с своей серией игр «The Room». Жанр гонки, постоянно выходят новые гонки, со своими особенностями, но лидерство занимает так же одна компания «Gameloft» и их серия «Asphalt», которая насчитывает уже 10 игр в серии. Симуляторы делят несколько «гигантов», это «GIANTS Software» и их серия «Farming simulator», «Fun Games For Free» с игрой «Бесплатный 3D-авиасимулятор: самолет изумительный», «Zuuk Games» с симуляторами грузовиков и автобусов, и совсем не давно вышедший симулятор езды по бездорожью «MadRunner» от «Focus Home Interactive». Теперь о самом

									Лист
									8
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

главное и интересном, шутеры, когда-то лидерство здесь принадлежало так же «Gameloft» и их серии игр «Modern combat». Но теперь здесь два гиганта которые относительно совсем не давно на рынке мобильных игр, 2 года «Tencent Games» с игрой «PUBG Mobile», 1 год «Activision Publishing, Inc.» с игрой «Call of Duty: Mobile», около 1 года «Epic Games» с игрой «Fortnite», и 2 года «GARENA INTERNATIONAL I PRIVATE LIMITED» с игрой «Garena free fire».

### **1.3 Обзор существующих мобильных платформ**

Наиболее распространенные операционные системы для смартфонов и платформ:

Symbian OS - ОС занимала большую часть рынка смартфонов до конца 2010 г. В начале 2010 года на базе ОС остается только 1 Платформа: Series 60, которая используется в основном в устройствах Nokia, а также некоторые моделях Samsung;

BlackBerry OS (RIM) – система широко используется в устройствах в первую очередь в Соединенных Штатах, так как спецслужбы некоторых стран не заинтересованы в использовании смартфонов в стране из-за того, что все входящие и исходящие данные зашифрованы с использованием алгоритма шифрования AES;

Windows Mobile и Windows CE - компактная операционная система Microsoft, выпущенная в 1996 году и занимавшая самый большой сегмент рынка ОС для смартфонов к 2010 году, в настоящее время проходит поэтапный отказ от поддержки и развития;

Windows Phone 7 - разработка от Microsoft, радикально отличается от Windows Mobile;

Palm OS - одна из популярных платформ, хотя в настоящее время мобильные телефоны на базе Palm OS мало распространены. Последний смартфон под управлением операционной системы был выпущен в конце 2007 года (Palm Centro);

									Лист
									9
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

Linux - широкого распространения эта операционная система на мобильных устройствах не получила, однако ее развитие традиционно считается перспективным направлением. Смартфоны на базе Linux распространяются главным образом в Азии;

Bada - новейшая мобильная платформа, разработанная компанией Samsung. Первым телефоном на новой платформе стал S8500 Wave;

Android - портативная (сетевая) операционная система для смартфонов, планшетных ПК, электронные книги, цифровых плееров, часов и нетбуков на базе ядра Linux. Первоначально разработанная Android Inc., которую затем купил Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать приложения на основе Java, который управляют устройством через разработанные Google библиотеки. Android Native Development Kit позволяет системе использовать библиотеки и компоненты приложений, написанных на C и других языках;

ОС IOS (до 24 июня 2010 года - iPhone OS) - это мобильная операционная система, разработанная и изготовленная американской компанией Apple. Она была выпущена в 2007 году; первоначально - для iPhone и iPod Touch, а позже для таких устройств, как iPad и Apple TV. В отличие от Windows Phone и Google Android, доступна только для устройств, производимых Apple;

Windows Phone 8 - второе поколение телефонной операционной системы от Microsoft Windows. Выход произошел 29 октября и является прототипом интерфейса, известного как Metro (или Modern UI). Windows Phone 8 использует новую архитектуру Windows NT, которая используется в операционных системах Microsoft. Устройства, работающие под управлением Windows Phone 7. не может выполнить обновление до Windows Phone 8, а новые приложения, созданные для Windows Phone 8 не могут работать на Windows Phone 7.

					ТГКГЗ.ПИ.22.00.000.КР	Лист
Изм.	Лист	№ документа	Подпись	Дата		10

В настоящее время Android развивается в геометрической прогрессии: каждый год число пользователей этой операционной системы постоянно растет. Согласно отчету компании Canalys за 2013 год, ведущего аналитика индустрии высоких технологий, операционная система Android уже тогда уже занимала 69,2% мирового рынка мобильных устройств . Конечно, этот факт привлекает внимание многих разработчиков создавать мобильные приложения специально для Android. Возможно, на сегодняшний день она является самой популярной и интересной системой. Разработчики дают пользователям уникальную возможность – установив набор свободного программного обеспечения, можно создать программы для системы и продавать их в специализированном интернет-магазине.

#### **1.4 Описание предметной области**

Знакомство с разработкой в области приложений под операционную систему Android, предоставляет отличную возможность по написанию программ для мобильных устройств. Android это набор программ с открытым исходным кодом, который включает операционную систему, подпрограммное обеспечение и ключевые мобильные приложения вместе с библиотеками API, предназначенными для написания новых программ, определяющих визуальное представление и функционал мобильных устройств. Самые разнообразные мобильные устройства со временем снабжались такими мощными инструментами, как камера, медиа-плеер, навигатор, сенсорный дисплей. С внедрением новых технологий мобильный телефон превратился в нечто большее, чем просто устройство для звонков. При этом программная платформа и среда разработки отставали от бешеного темпа развития технологий. До недавнего времени мобильные телефоны работали под управлением закрытых платформ, построенных на основе запатентованных операционных систем, для чего требовались запатентованные инструменты разработки. Сами же телефоны функционировали с оригинальным программным обеспечением гораздо

						ТГКГЗ.ПИ.22.00.000.КР	Лист
							11
Изм.	Лист	№ документа	Подпись	Дата			

лучше, чем со сторонним. Это создавало искусственные препятствия для программистов, которые рассчитывали на использование более мощного аппаратного обеспечения мобильных устройств.

В случае с Android встроенное ПО написано на том же самом API, что и программы сторонних разработчиков, при этом время для исполнения и тех, и других одинаково. Данное API позволяет получить доступ к сенсорному управлению, навигационным сервисам, фоновым и картографическим процессам, реляционным базам данных, двумерной и трехмерной графике, к функциям видеозаписи, межпрограммного взаимодействия. В распоряжении разработчика приложений для Android достаточно мощное API и качественная справочная документация. Он может стать членом огромного сообщества, ему не нужно тратить на программное обеспечение или рекламу своего продукта. С ростом популярности мобильных устройств открываются великолепные перспективы разрабатывать инновационные мобильные приложения, причем с любым опытом программирования. Android — одна из операционных систем нового поколения, созданных для работы с аппаратным обеспечением современных мобильных устройств. На сегодняшний день Windows Mobile и Apple iPhone предлагают достаточно мощные и более простые в использовании среды разработки мобильных приложений. Однако в отличие от Android это запатентованные операционные системы, в которых в определенных случаях приоритет отдается встроенному ПО, а не приложениям сторонних программистов. Кроме того, эти операционные системы ограничивают возможности взаимодействия приложений с данными телефона, а также ограничивают или контролируют процесс распространения сторонних приложений, созданных для данных платформ. Android дает новые возможности для мобильных приложений, предлагая открытую среду разработки, построенную на открытом ядре Linux. У всех приложений есть доступ к аппаратным средствам устройства, для чего используются специальные серии API-библиотек.

									Лист
									12
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

Кроме того, здесь включена полная и контролируемая поддержка взаимодействия приложений. На платформе Android все программы имеют одинаковый статус. Сторонние приложения написаны на том же API, что и встроенное ПО, при этом во всех программах одинаковое время исполнения. Пользователи могут удалять или заменять встроенные ПО на альтернативные сторонние разработки, например, игровое приложение или рабочий стол. История Android началась в далеком 2002 году, когда корпорация Google заинтересовались наработками Энди Рубина и решила сделать из этого большой проект. И в 2007 году Google решил организовать большой альянс разработчиков мобильных устройств с целью продвигать Android, как 7 операционную систему для телефонов. Как мы видим на сегодняшний день это у них отлично получилось. Android — операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков и других устройств. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Архитектуру Android принято делить на четыре уровня: уровень ядра; уровень библиотек и среды выполнения; уровень каркаса приложений; уровень приложений. На конференции для разработчиков в мае 2017 года Google объявила, что за всю историю Android было активировано более 2 млрд Android-устройств. Прогнозы относительно данной платформы положительные. Немаловажными аспектами роста динамики явились такие характеристики как, открытость системы, возможность вносить изменения в основные приложения, возможность быстрой и легкой разработки. Поэтому выбор разработки приложения для Android является очень актуальным. Разработчику, работающему с платформой Android дается возможность писать код на Java абстрагируясь от ядра.

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	13
Изм.	Лист	№ документа	Подпись	Дата		

У данной операционной системы имеются такие плюсы как: фреймворк, имеющий широкий набор API для созданий разнообразных видов приложений и дающий возможность повторного использования и замены компонентов, предлагаемые платформой и другими приложениями. А также наличие виртуальной машины Dalvik, обеспечивающий запуск приложений. Кроме того инструментами Android являются база данных SQLite, 2D и 3D графика, Media Player, коммуникации, протоколы обмена и различные библиотеки. Компоненты приложения являются кирпичиками, из которых состоит приложение для Android. Каждый компонент представляет собой отдельную точку, через которую система может войти в приложение. Не все компоненты являются точками входа для пользователя, а некоторые из них зависят друг от друга. При этом каждый компонент является самостоятельной структурной единицей и играет определенную роль — каждый из них представляет собой уникальный элемент структуры, который определяет работу приложения в целом. Компоненты приложения можно отнести к одному из четырех типов. Компоненты каждого типа предназначены для определенной цели, они имеют собственный жизненный цикл, который определяет способ создания и прекращения существования компонента.

Четыре типа компонентов: Activity представляет собой один экран с пользовательским интерфейсом. Например, в приложении для работы с электронной почтой одна операция может служить для отображения списка новых сообщений, другая – для составления сообщения и третья операция – для чтения сообщений. Несмотря на то что операции совместно формируют связанное взаимодействие пользователя с приложением по работе с электронной почтой, каждая из них не зависит от других операций. Любые из этих операций могут быть запущены другим приложением (если это позволяет приложение по работе с электронной почтой). Например, приложение для камеры может запустить операцию в приложении по работе с электронной почтой, которая составляет новое сообщение, чтобы

						<i>Лист</i>
					<b>ТГКГЗ.ПИ.22.00.000.КР</b>	14
<i>Изм.</i>	<i>Лист</i>	<i>№ документа</i>	<i>Подпись</i>	<i>Дата</i>		



пользователь мог отослать фотографию; Service представляет собой компонент, который работает в фоновом режиме и выполняет длительные операции, связанные с работой удаленных процессов. Служба не имеет пользовательского интерфейса. Например, она может воспроизводить музыку в фоновом режиме, пока пользователь работает в другом приложении, или же она может получать данные по сети, не блокируя взаимодействие пользователя с операцией. Служба может быть запущена другим компонентом, который затем будет взаимодействовать с ней, – например операцией; Content provider управляет общим набором данных приложения. Данные можно хранить в файловой системе, базе данных SQLite, в Интернете или любом другом постоянном месте хранения, к которому у вашего приложения имеется доступ. Посредством поставщика контента другие приложения могут запрашивать или даже изменять данные (если поставщик контента позволяет делать это). Например, в системе Android есть поставщик контента, который управляет информацией контактов пользователя. Любое приложение, получившее соответствующие разрешения, может запросить часть этого поставщика контента (например ContactsContract.Data), для чтения и записи сведений об определенном человеке; Broadcast receiver представляет собой компонент, который реагирует на объявления распространяемые по всей системе. Многие из этих объявлений рассылает система — например объявление о том, что экран выключился, аккумулятор разряжен или был сделан фотоснимок. Объявления также могут рассылаться приложениями, — например, чтобы сообщить другим приложениям о том, что какие-то данные были загружены на устройство и теперь готовы для использования. Несмотря на то, что приемники широковещательных сообщений не имеют пользовательского интерфейса, они могут создавать уведомления в строке состояния, чтобы предупредить пользователя о событии "рассылка объявления". Однако чаще всего они являются просто "шлюзом" для других компонентов и предназначены для выполнения минимального объема работы. Например,

									Лист
									15
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

они могут инициировать выполнение службой определенных действий при возникновении события. Уникальной особенностью системы Android является то, что любое приложение может запустить компонент другого приложения. Например, предоставить пользователю возможность фотографировать, используя камеру устройства, то, поскольку наверняка имеется другое приложение, которое может выполнить это действие, вместо того чтобы разработать операцию фотографирования в своем приложении, вы можете вызвать такое приложение. Вам не нужно внедрять код из приложения для камеры или даже устанавливать на него ссылку. Вместо этого вы можете просто запустить операцию фотографирования из приложения для камеры. По завершении этой операции фотография будет возвращена в ваше приложение, и ее можно будет использовать. Для пользователя это будет выглядеть как одно приложение. Когда система запускает компонент, она запускает процесс для этого приложения (если он еще не был запущен) и создает экземпляры классов, которые требуются этому компоненту. Например, если ваше приложение запустит операцию фотографирования в приложении для камеры, эта операция будет выполняться в процессе, который относится к этому стороннему приложению, а не в процессе вашего приложения. Поэтому, в отличие от приложений для большинства других систем, в приложениях для Android отсутствует единая точка входа (например, в них нет функции `main()`). Поскольку система выполняет каждое приложение в отдельном процессе с такими правами доступа к файлам, которые ограничивают доступ в другие приложения, ваше приложение не может напрямую вызвать компонент из другого приложения. Это может сделать сама система Android. Поэтому, чтобы вызвать компонент в другом приложении, необходимо сообщить системе о своем намерении (`Intent`) запустить определенный компонент. После этого система активирует для вас этот компонент.

										Лист
										16
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР					

Компоненты трех из четырех возможных типов — операции, службы и приемники широковещательных сообщений — активируются асинхронным сообщением, которое называется Intent (намерение). Объекты Intent связывают друг с другом отдельные компоненты во время выполнения, будь то это компоненты вашего или стороннего приложения (эти объекты Intent можно представить себе в виде мессенджеров, которые посылают другим компонентам запрос на выполнение действий). Объект Intent создается с помощью объекта Intent, который описывает запрос на активацию либо конкретного компонента, либо компонента конкретного типа — соответственно, намерение Intent может быть явным или неявным. Для операций и служб Объект Intent определяет действие, которое требуется выполнить (например, просмотреть (view) или отправить (send) что-то), а также может указывать URI (Uniform Resource Identifier – унифицированный идентификатор ресурса) данных, с которыми это действие нужно выполнить (помимо прочих сведений, которые нужно знать запускаемому компоненту). Например, объект Intent может передавать запрос на выполнение операции "показать изображение" или "открыть вебстраницу". В некоторых ситуациях операцию можно запустить, чтобы получить результат. В этом случае операция возвращает результат также в виде объекта Intent (например, можно отправить сообщение Intent, чтобы дать пользователю возможность выбрать контакт и вернуть его вам — в ответном сообщении Intent будет содержаться URI, указывающий на выбранный контакт). Для приемников широковещательных сообщений Intent просто определяет передаваемое объявление (например, широковещательное сообщение о низком уровне заряда аккумулятора содержит только строку "аккумулятор разряжен").

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	17
Изм.	Лист	№ документа	Подпись	Дата		

Компоненты четвертого типа – поставщики контента – сообщениями Intent не активируются. Они активируются по запросу от ContentResolver. Процедура определения контента (content resolver) обрабатывает все прямые транзакции с поставщиком контента, с тем, чтобы этого не пришлось делать компоненту, который выполняет транзакции с поставщиком. Вместо этого он вызывает методы для объекта ContentResolver. Это формирует слой, абстрагирующий (в целях безопасности) поставщика контента от компонента, запрашивающего информацию. Для запуска компонента приложения системе Android необходимо знать, что компонент существует. Для этого она читает файл AndroidManifest.xml приложения (файл манифеста). В этом файле, который должен находиться в корневой папке приложения, должны быть объявлены все компоненты приложения. Помимо объявления компонентов приложения, манифест служит и для других целей, среди которых: указание всех полномочий пользователя, которые требуются приложению, например разрешения на доступ в Интернет или на чтение контактов пользователя; объявление минимального уровня API, требуемого приложению, с учетом того, какие API-интерфейсы оно использует; объявление аппаратных и программных функций, которые нужны приложению или используются им, например камеры, службы Bluetooth или сенсорного экрана; указание библиотек API, с которыми необходимо связать приложение (отличные от API-интерфейсов платформы Android), например библиотеки Google Maps. Основная задача манифеста – это информировать систему о компонентах приложения. Существует огромное количество устройств, работающих под управлением Android, и не все они имеют одинаковые функциональные возможности. Чтобы ваше приложение не могло быть установлено на устройствах, в которых отсутствуют функции, необходимые приложению, важно четко определить профиль для типов устройств, поддерживаемых вашим приложением, указав требования к аппаратному и программному обеспечению в файле манифеста.

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	18
Изм.	Лист	№ документа	Подпись	Дата		



Стремительный рост технологий, связанных с Интернетом и "электронной коммерцией", дополнительно усложняют эту задачу. Модный ныне объектно-ориентированный подход сам по себе не решает этих проблем, более того, часто приносит новые. Именно система программирования на основе языка Java(TM) обладает следующими характеристиками: язык программирования объектно-ориентирован, в то же время довольно прост для освоения -цикл разработки приложений сокращен за счет того, что система построена на основе интерпретатора -приложение получается автоматически переносимым между множеством платформ и операционных систем за счет встроенной системы сборки мусора программист освобождается от необходимости явного управления памятью - в интерактивном графическом приложении удается достичь высокой производительности (быстрого отклика на ввод пользователя) за счет встроенной в систему многопоточности-приложение легко сопровождается и модифицируется, т.к. модули могут быть загружены с сети -в приложения встроена система безопасности, не допускающая незаконного доступа и проникновения вирусов. Система Java создана на основе «простого» языка программирования, техника, использования которого близка к общепринятой, и обучение которому не требует значительных усилий. Java как язык программирования является объектно-ориентированной с момента основания. Кроме того программист с самого начала обеспечивается набором «стандартных» библиотек, обеспечивающих функциональность от стандартного ввода/вывода и сетевых протоколов до графических пользовательских интерфейсов. Эти библиотеки легко могут быть расширены. Несмотря на то, что язык С++ был отвергнут, синтаксис языка Java максимально приближен к синтаксису С++. Это делает язык знакомым широкому кругу программистов. В то же время из языка были удалены многие свойства, которые делают С++ излишне сложным для пользования, не являясь абсолютно необходимыми. В результате язык Java получился более простым и органичным, чем С++.

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	20
Изм.	Лист	№ документа	Подпись	Дата		

Надежность и безопасность Java существенно облегчает создание надежного программного обеспечения. Кроме исчерпывающей проверки на этапе компиляции, система предусматривается анализ на этапе выполнения. Сам язык спроектирован так, чтобы вырабатывать у программиста привычку писать "правильно". Модель работы с памятью, в которой исключено использование указателей, делает невозможными целый класс ошибок, характерных для C и C++. В силу того, что Java предназначена для работы в распределенной среде, безопасность становится чрезвычайно важной проблемой. Требования безопасности определяют многие черты как языка, так и реализации всей системы. Независимость от архитектуры и переносимость. Компилятор Java производит байт-коды, т.е. модули приложения имеют архитектурно-независимый формат, который может быть проинтерпретирован на множестве разнообразных платформ. Это уже не исходные тексты, но еще не платформно-зависимые машинные коды. Следующий шаг - "замораживание" стандарта на формат основных встроенных типов данных. Программа, созданная на одной платформе, работает на всех остальных. Этот стандарт фиксирован в документе, описывающем Java Virtual Machine. Стандарт может быть реализован на любой аппаратно-программной платформе, поддерживающей многопоточность. Производительность Схема работы системы и набор байт-кодов виртуальной машины Java таковы, что позволяют достичь высокой производительности на этапе выполнения программы: анализ кодов на соблюдение правил безопасности производится один раз до запуска кодов на выполнение, в момент выполнения таких проверок уже не нужно, и коды выполняются максимально эффективно; работа с базовыми типами максимально эффективна, для операций с ними зарезервированы специальные байт-коды методы в классах не обязательно связываются динамически автоматический сборщик мусора работает отдельным фоновым потоком, не замедляя основную работу программы, но в то же время, обеспечивая своевременный возврат свободной памяти в систему; стандарт

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	21
Изм.	Лист	№ документа	Подпись	Дата		

предусматривает возможность написания критических по производительности участков программы в машинных кодах.

Интерпретируемая природа языка позволяет сделать фазу линкования простой, инкрементальной и, следовательно, быстрой. Это резко сокращает цикл разработки и тестирования программных фрагментов. Многопоточность позволяет выполнять в рамках одного приложения несколько задач одновременно. Это становится особенно актуально в современных распределенных приложениях, когда процессы сетевого обмена могут идти одновременно и асинхронно. При этом программа продолжает реагировать на ввод информации пользователем без неприятных задержек.

Многопоточность поддерживается на уровне языка - часть примитивов синхронизации встроена в систему реального времени, а библиотека содержит базовый класс Thread. К тому же системные библиотеки написаны thread-safe, т.е. все они могут быть использованы в многопоточковых приложениях. Система обеспечивает динамическую сборку программы. Классы подгружаются по мере необходимости, причем загружены они могут быть с любой точки сети, что позволяет сделать внесение изменений в приложения прозрачным для пользователя. Пользователь может быть уверен, что всегда работает со свежей версией приложения. Полная система Java включает в себя готовый набор библиотек, который можно разбить на следующие пакеты: java.lang - базовый набор типов, отраженных в самом языке. Этот пакет обязательно входит в состав любого приложения. Содержит описания классов Object и Class, а также поддержку многопоточности, исключительных ситуаций, оболочку для базовых типов, а также некоторые фундаментальные классы. java.io - потоки и файлы произвольного доступа. Аналог библиотеки стандартного ввода-вывода системы UNIX. Поддержка сетевого доступа (sockets, telnet, URL) содержится в пакете java.net. -java.util - классы-контейнеры (Dictionary, HashTable, Stack) и некоторые другие утилиты. Кодирование и декодирование.

									Лист
									22
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				



Классы `Date` и `Time`. `java.awt` - Abstract Windowing Toolkit, архитектурно-независимый оконный интерфейс, позволяющий запускать интерактивные оконные Java приложения на любой платформе. Содержит базовые компоненты интерфейса, такие как события, цвета, фонты, а также основные оконные элементы - кнопки, `scrollbars` и т.д. Каждая из перечисленных характеристик по отдельности может быть найдена в уже существующих программных пакетах. Новым является соединение их в стройную непротиворечивую систему, которая должна стать всеобщим стандартом. Основные свойства языка программирования Java Встроенные (примитивные) типы данных. В языке Java существует набор встроенных типов данных, которые не являются объектами. Набор их сходен с набором базовых типов C++ за некоторыми исключениями. Отличаются от C++ как синтаксисом, так и представлением. Тип `character` есть 16-разрядное число без знака (диапазон 0-65,535). Кодировка соответствует стандарту Unicode. В силу того, что эта кодировка в идеале должна охватывать все существующие в мире языки, это представление должно облегчить локализацию приложений. `Boolean` этот тип данных не выделен в C++, однако неявно присутствует практически во всех программах. В Java тип `boolean`, может принимать значения `true` и `false` и не может (в отличие от C++) быть преобразован в другой тип. Добавлен новый оператор логического сдвига вправо (т.к. нет беззнаковых целых чисел). Встроенная операция слияния строк (оператор `+`). В отличие от C++ массивы в Java являются полноценными объектами с определенным runtime представлением Система Java создавалась объектно-ориентированной с самого начала. Объектно-ориентированная парадигма наиболее удобна при создании программного обеспечения типа клиент-сервер, а также для организации распределенных вычислений. Одна из черт, присущих объектам, заключается в том, что объекты обычно переживают процедуру, их создающую. Они затем могут перемещаться по сети и храниться в базах данных.

										Лист
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР					23

Основные требования к объектно-ориентированной системе - инкапсуляция - сокрытие реализации за абстрактным интерфейсом - полиморфизм - одно и то же сообщение, посланное различным объектам, приводит к выполнению разных операций -наследование - новые классы могут наследовать данные и функциональность уже существующих классов - динамическое связывание - новые классы могут появляться в системе откуда угодно, в том числе и из сети. Необходимо иметь возможность динамически включать их в систему. Классы Класс есть языковая конструкция, определяющая поля данных объектов данного класса (instance variables) и их поведение (methods). Практически класс в Java сам по себе не является объектом. Это лишь шаблон, который определяет, из каких частей будет состоять объект, созданный с помощью этого класса, и как он будет себя вести. Освещены следующие стороны Java как объектно-ориентированного языка программирования. Классы определяют шаблон, по которому создаются конкретные объекты. Поля данных объекта определяют состояние объекта. Объекты обмениваются сообщениями между собой. Получение сообщения приводит к вызову одного из методов. Методы определяют поведение объекта данного класса. Методы для разных классов могут иметь одно и то же имя, но различное содержание. Система Java предназначена для создания программного обеспечения, которое должно быть интеллектуальным, предельно надежным и безопасным по множеству параметров. Особое внимание уделяется как ранней диагностике возможных проблем, так и поздней, во время выполнения кодов. Система Java достаточно безопасна, чтобы жить в сетевом окружении. Нейтральность к архитектуре и переносимость делают ее достаточно привлекательной для создания распределенных по сети приложений.

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	24
Изм.	Лист	№ документа	Подпись	Дата		

## 1.5 Сравнение сред Сравнение Unity и Construct 2

Для сравнения были взяты конструктор Construct 2 и игровой движок Unity. Первоначально были важны бесплатность и удобство экспортирования на Android. Если с первым у обоих программ нет проблем, то портировать с Cordova, который собирает Construct на Android возникает много сложностей. Так же конструктор не может предложить такой список возможностей для разработки приложения, как Unity. Выбор остановился на Unity по ряду причин. Первое, это бесплатная версия, данная версия не имеет никаких ограничений, имеет полный набор функций, а так же позволяет просто экспортировать программу на Android. Так же у Unity есть возможность использовать готовые «ассеты» для создания игр, разного рода текстуры и объекты, как платные, так и бесплатные. Так же к преимуществам выбранной среды разработки можно отнести большое обилие обучающих материалов в интернете, что очень помогает новичкам. Unity позволяет создавать кроссплатформенные проекты, поддерживает разработки приложения для Windows, MacOS, Android, IOS, а так же приложения для игровых консолей.

Так как мы разрабатываем приложения для Android, то ещё один главный момент, это оптимизация, а Unity достаточно «лёгкий» игровой движок с хорошей оптимизацией, о которой мы ещё поговорим в процессе разработки.

На данный момент Unity один из самых популярных игровых движков, если не самый популярный, на данном игровом движке даже есть «игровой гигант» в лице Heartstone от Blizzard. Это хороший пример правильного использования не самого сложного игрового движка, ведь количество игроков в Heartstone превышает 100 млн игроков. Так же это хорошая демонстрация кроссплатформенности движка.

									Лист
									25
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

Ещё не много о сравнении Construct 2 и Unity, главным отличием является способ программирования. В Construct 2 вы создаёте объект, а дальше лишь задаёте ему набор действий и способы взаимодействия, никакой работы с кодом. Вы выбираете уже готовые варианты тех или иных действий. Это является как плюсом, так и минусом, ведь это легко и быстро, но получается достаточно «грубым» поведение объекта. Так же оптимизация таких приложений желает лучшего, а напоследок это работа с «физикой», то есть с более сложными приложениями уже является не возможным. Работа в Unity представляет собой не только добавление объектов на рабочее поле, но и работа с кодом C#, где прописываете поведение, действия и взаимодействия объектов. В Unity есть специальные готовые компоненты для более сложных приложений, а именно «физика» объектов. Но не стоит думать, что добавив объекту данный компонент, у вас сразу будет готовая игра, так как все тонкости этой самой «физики» вы сами корректируете в коде C#. На рисунке 1. представлено рабочее окно Unity

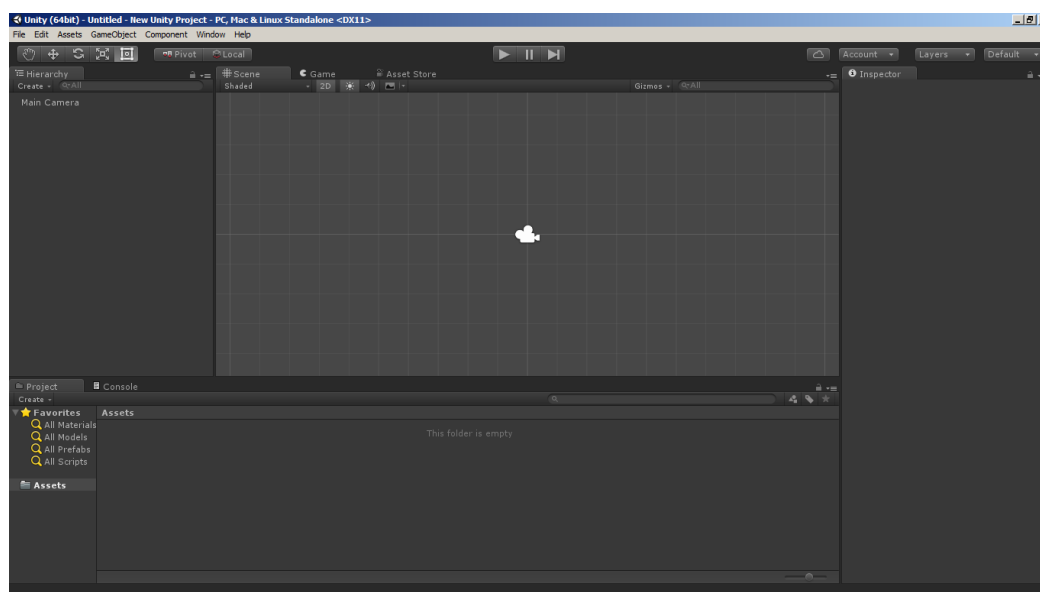


Рис 1. Рабочее окно Unity

На рисунке 2. представлено рабочее окно программы Construct 2.

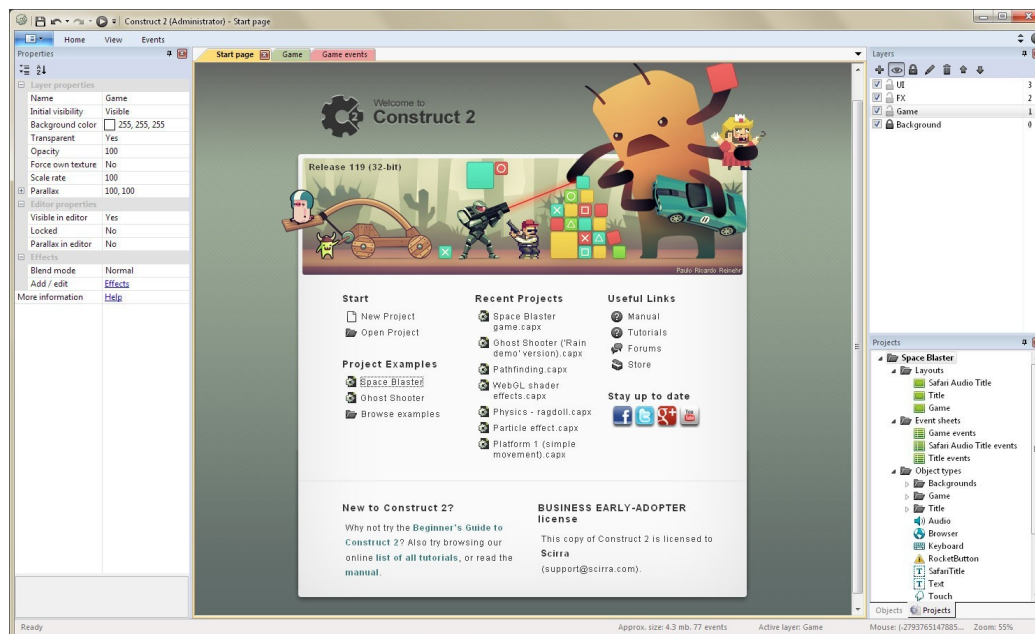


Рис 2. Рабочее окно Construct 2

Подводя итог сравнения, стоит отметить, что Construct 2 не такая и плохая программа для разработки приложений для Android, с очень легким принципом работы, но, к сожалению, уже устаревшая на сегодняшний день, поэтому наш выбор останавливается на Unity.

Ещё раз разберём основные плюсы Unity:

- Простота и удобство использования;
- Кроссплатформенность;
- Оптимизация для мобильных устройств;
- Бесплатная версия;
- Наличие готовых текстур и объектов;
- Огромное количество бесплатных обучающих материалов.

## ГЛАВА 2. РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

### 2.1 Обзор программного обеспечения

Для разработки приложения была выбрана программа Unity. Сейчас мы более подробно на этом остановимся. Unity – больше, чем движок, это среда для разработки компьютерных игр, в которой объединены различные программные средства, используемые при создании ПО – текстовый редактор, компилятор, отладчик и так далее. При этом, благодаря удобству использования, Unity делает создание игр максимально простым и комфортным, а кроссплатформенность движка позволяет разработчику охватить как можно большее количество игровых платформ и операционных систем.

В первую очередь, как мы уже упоминали, движок Unity3D дает возможность разрабатывать игры, не требуя для этого каких-то особых знаний. Здесь используется компонентно-ориентированный подход, в рамках которого разработчик создает объекты (например, главного героя) и к ним добавляет различные компоненты (например, визуальное отображение персонажа и способы управления им). Благодаря удобному Drag & Drop интерфейсу и функциональному графическому редактору движок позволяет рисовать карты и расставлять объекты в реальном времени и сразу же тестировать получившийся результат.

Второе преимущество движка – наличие огромной библиотеки ассетов и плагинов, с помощью которых можно значительно ускорить процесс разработки игры. Их можно импортировать и экспортировать, добавлять в игру целые заготовки – уровни, врагов, паттерны поведения ИИ и так далее. Практически никакой работы с кодом. Многие ассеты доступны бесплатно, другие предлагаются за небольшую сумму, и при желании можно создавать собственный контент, публиковать его в Unity Asset Store и получать от этого прибыль.

									Лист
									28
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

Третья сильная сторона Unity 3D – поддержка огромного количества платформ, технологий, API. Созданные на движке игры можно легко портировать между ОС Windows, Linux, OS X, Android, iOS, на консоли семейств PlayStation, Xbox, Nintendo, на VR- и AR-устройства. Unity поддерживает DirectX и OpenGL, работает со всеми современными эффектами рендеринга, включая новейшую технологию трассировки лучей в реальном времени. Физика твердых тел, ragdoll и тканей, система Level of Detail, коллизии между объектами, сложные анимации – все это можно реализовать силами движка. Стереотипное мнение о том, что движок пригоден только для небольших инди-игр и неспособен выдавать красивую картинку, давно уже не актуально: достаточно посмотреть технодемо ADAM, The Blacksmith и Book of the Dead от создателей среды Unity, чтобы убедиться в ее выдающихся способностях.

Наконец, Unity доступен бесплатно, что открывает перед независимыми разработчиками дверь в игровую индустрию. Конечно, существуют ограничения: бесплатная версия движка демонстрирует лого Unity перед запуском игры, а проект, созданный с ее помощью, не должен приносить разработчику больше \$100 тысяч в год. Впрочем, тарифы на подписку не опустошат кошельки даже начинающей команды: Про-версия стоит \$125 в месяц, что не так уж много в сравнении с другими движками, причем базовая версия содержит ровно тот же функционал, что и профессиональная. Исходя из вышеописанного, движок идеально подходит для начинающих разработчиков, которым мало возможностей более простых инструментов, вроде RPG Maker, и которые, в то же время, не хотят тратиться на более дорогие и продвинутое движки. Unity позволяет быстро создать объекты, расставить и связать их, создать нехитрую сцену, задействовать собственный контент и содержимое магазина ассетов.

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	29
Изм.	Лист	№ документа	Подпись	Дата		

Так как движок имеет огромную аудиторию пользователей, найти решение любой проблемы не составит труда – комьюнити с радостью поможет начинающему разработчику, официальные и пользовательские блоги и обучающие курсы (в том числе на русском языке) дадут все необходимые знания. Большие студии также найдут в движке свои преимущества. Возможности движка позволяют создавать отличные проекты знакомые многим. За примерами далеко ходить не нужно: Pillars of Eternity, Firewatch, Inside, Superhot – все эти громкие проекты созданы на Unity. Будь то дорогой сюжетно-ориентированный проект или многопользовательская браузерная игра на Unity3D, разработчики в любом случае получают мощный и гибкий инструментарий для создания максимально качественного продукта.

## 2.2 Разработка игрового приложения

Идей для игры послужила игра Flappy Bird, в данной игре всё управление заключается в нажатии на экран для того, что бы игровой персонаж набирал высоту, что бы предотвратить столкновение с препятствиями, цель игры набрать как можно большее количество очков. Было принято решение воспроизвести игру на Unity, в то время как оригинал создан на Java. Игра крайне простая, механика игры крайне примитивна. Первое, при нажатии на экран птица должна взлетать, затем начинать снижении. Второе это при пролете между труб игроку должно засчитываться очко, при столкновении игра заканчивается. Ну а так же игра не имеет конца, то есть трубы должны постоянно появляться с произвольной высотой, но не выходя за границы экрана. Начинаем разработку с запуском Unity Hub. Здесь хранится список всех проектов, а так же вы можете видеть расположение любого вашего проекта на вашем компьютере.

									Лист
									30
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				



В окне представленном на рисунке 3. окно Unity Hub необходимо нажать кнопку «New», после чего откроется окно настроек проекта представленное на рисунке 4. окно настроек проекта в Unity Hub

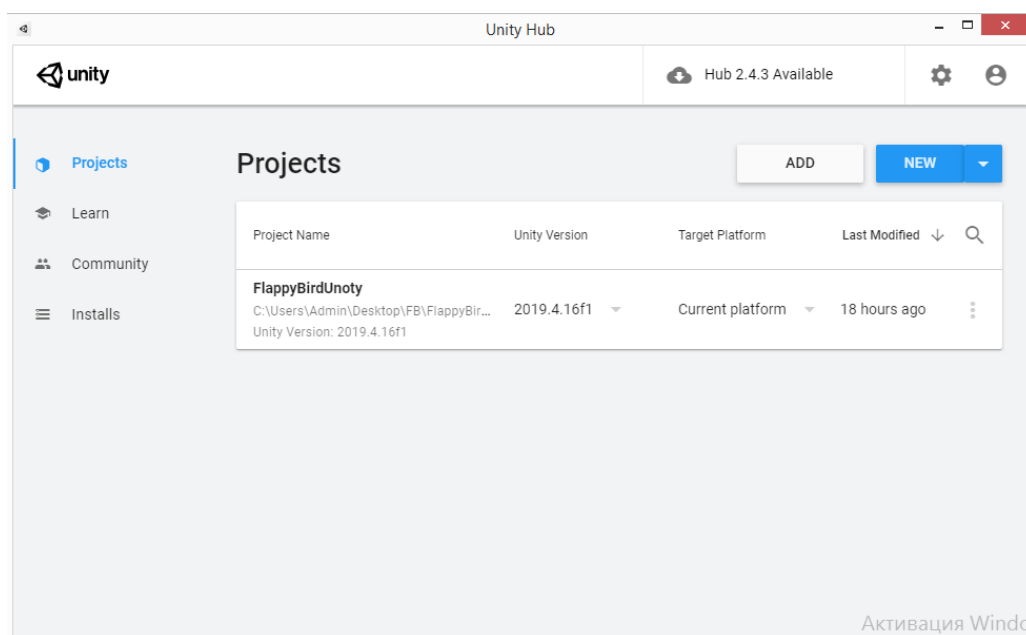


Рис 3. Окно Unity Hub

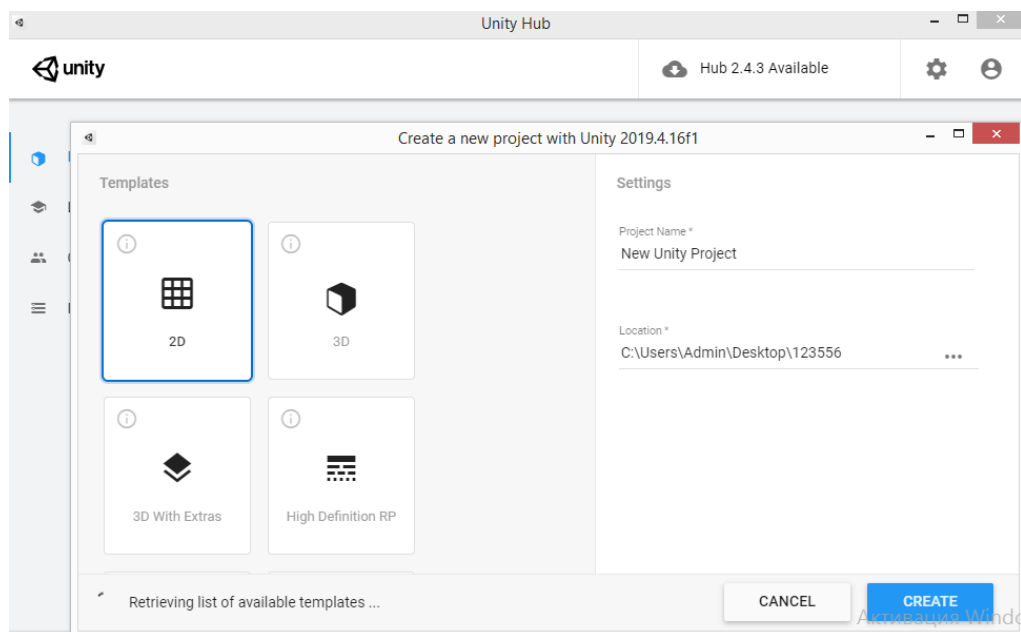


Рис 4. Окно настроек Unity Hub

Нам необходимо выбрать тип приложения 2D, так же как-либо назвать свой проект и выбрать местоположение проекта на компьютере, после чего нажать кнопку «Create». На этом готово, проект создан, но он совсем пустой.

Для создания игры нам так же необходимо использовать какие-либо текстуры объектов, можно использовать текстуры объектов, которые вы сами и нарисуете, можно использовать готовые в формате png. Нами было принято решение использовать готовые текстуры объектов, то что нам нужно было крайне легко найти на яндекс картинки. Заранее скачиваем и переносим все текстуры объектов в отдельную папку, вот что у нас было заготовлено. Всего понадобилось три текстуры объектов, птица, труба и кнопка «Старт».

Текстуры объектов продемонстрированы на рисунке 5. текстуры объектов для приложения.

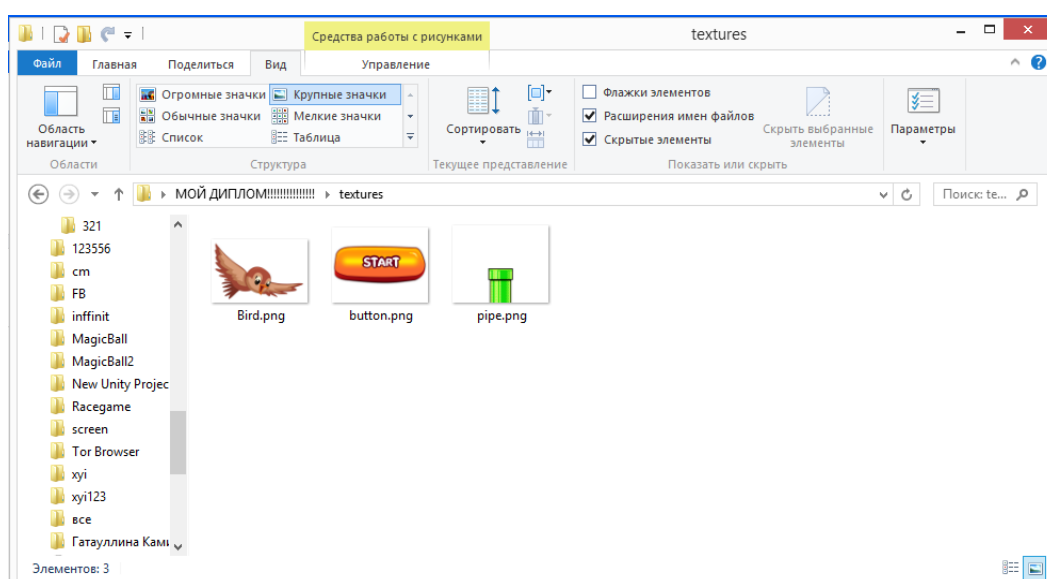


Рис 5. Текстуры объектов для приложения

Так же вы можете найти полный набор текстур - «ассеты» для Unity, где будут текстуры анимации персонажа, разные текстуры врагов, целей, каких-либо объектов окружения и взаимодействия.

Вернёмся к нашему проекту. Первым делом нужно задать настройки для построения приложения под Android системы, для этого выбираем File>Build settings и в открывшемся окне представленном на рисунке 6. Build settings выбираем Android и нажимаем «Switch platform».

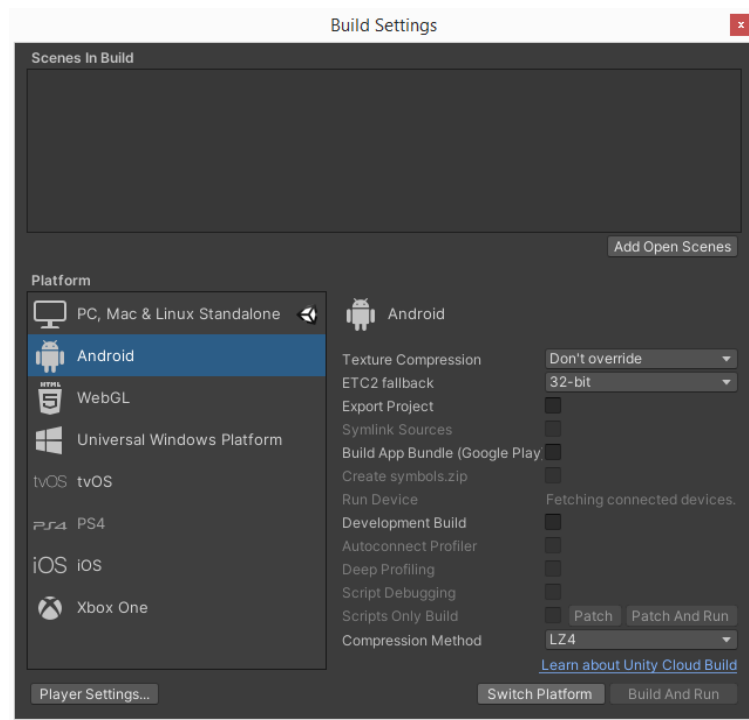


Рис 6. Build settings

Далее переносим все наши текстуры в нижнюю панель Assets в рабочем окне Unity. Что бы перенести объект на сцену, просто нажимаете и удерживаете объект и переносите на сцену. Далее вы можете настроить размер объекта и положение объекта на сцене и экране.

Первым делом мы прописываем код для нашей птицы, код представлен на рисунке 7. код игрового персонажа.ц4

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Bird : MonoBehaviour
6  {
7      public float force; // переменная для силы прыжка
8      Rigidbody2D BirdRigid; // нам нужен Rigidbody, а то птичка не будет прыгать
9
10     public GameObject RestartButton; // это для кнопки
11
12     void Start()
13     {
14         Time.timeScale = 1; // скорость равна 1
15         BirdRigid = GetComponent<Rigidbody2D>(); // получаем компонент Rigidbody
16     }
17
18     void Update()
19     {
20     }
21     if (Input.GetMouseButtonDown(0)) // если жмем на кнопку мыши или экран
22     {
23         BirdRigid.velocity = Vector2.up * force; // как раз силу настраиваем
24     }
25 }
26
27 private void OnCollisionEnter2D(Collision2D collision) // проверяем столкновение
28 {
29     if (collision.collider.tag == "Enemy") // если тэг объекта "Енему"
30     {
31         Destroy(gameObject); // то птичка уничтожается(
32         Time.timeScale = 0; // время останавливается
33         RestartButton.SetActive(true); // кнопка рестарта появляется
34     }
35 }
36
37
38 }

```

Рис 7. Код игрового персонажа

Так же были добавлены комментарии для поправок, в случае если что-то пойдет не так. В данном коде мы подключаем и настраиваем компонент Rigidbody, который отвечает за «физику» нашей птицы, а именно сила прыжка, так же настраивается управления, необходимо что бы при нажатии на экран птица направлялась вверх с определенной силой. Так же в конце мы настраиваем collision – соприкосновения птицы с препятствиями, в нашем случае это трубы, так же есть условие , что объект должен иметь тэг «Enemy». И в заключении в случае если птица коснется трубы, то время остановится и появится кнопка рестарта. По мимо Rigidbody к птице нужно подключить Circle collider 2D, данный компонент задаст так называемый скелет нашей птицы, используется круглый скелет, что бы избежать случайных столкновений. И данный компонент подключается уже непосредственно в Unity.

Теперь разберём код персонажа более подробно. Рассмотрим всё по порядку, начнём с подключения компонента Rigidbody и кнопки перезапуска, рисунок 8. подключение Rigidbody и кнопки перезапуска.

```
public class Bird : MonoBehaviour
{
    public float force; // переменная для силы прыжка
    Rigidbody2D BirdRigid; // нам нужен Rigidbody, а то птичка не будет прыгать
    public GameObject RestartButton; // это для кнопки
}
```

Рис 8. Подключения Rigidbody и кнопки перезапуска

На рисунке выше добавляем новую величину - force, которая обозначает силу прыжка. А так же подключаем компонент Rigidbody, что бы наш персонаж имел «твёрдое тело», иначе персонаж не будет летать.

Далее задаем скорость движения и приводим в активное состояние компонент Rigidbody. Наглядно нужный код вы можете видеть на рисунке 9. скорость движения.

```

void Start()
{
    Time.timeScale = 1;           // скорость равна 1
    BirdRigid = GetComponent<Rigidbody2D>(); // получаем компонент Rigidbody
}

```

Рис 9. Скорость движения

Теперь нам необходимо настроить управление и настроить сила, а конкретно направление силы. Нам необходимо что бы сила была направлена вверх, что бы при нажатии на клавишу мыши или же на экран, персонаж совершал прыжок, с силой, которую мы задали ранее. Для этого нужен код, представленный на рисунке 10. управление и направление силы

```

void Update()
{
    if (Input.GetMouseButtonDown(0)) // если ждем на кнопку мыши или экран
    {
        BirdRigid.velocity = Vector2.up * force; // как раз силу настраиваем
    }
}

```

Рис 10. Управление и направление силы

Всё записываем как условие, к контрено фиксируем нажатие, а затем даём персонажу силу направленную вверх.

Переходим к ещё одному очень важному моменту, а именно столкновение птицы с трубам. Здесь пригодится цикл, который будет фиксировать касания птицы с препятствиями, имеющими тэг «Enemy». Если столкновение происходит, то птица исчезает и время останавливает. Более подробно код можно рассмотреть на рисунке 11. столкновение объектов.

```

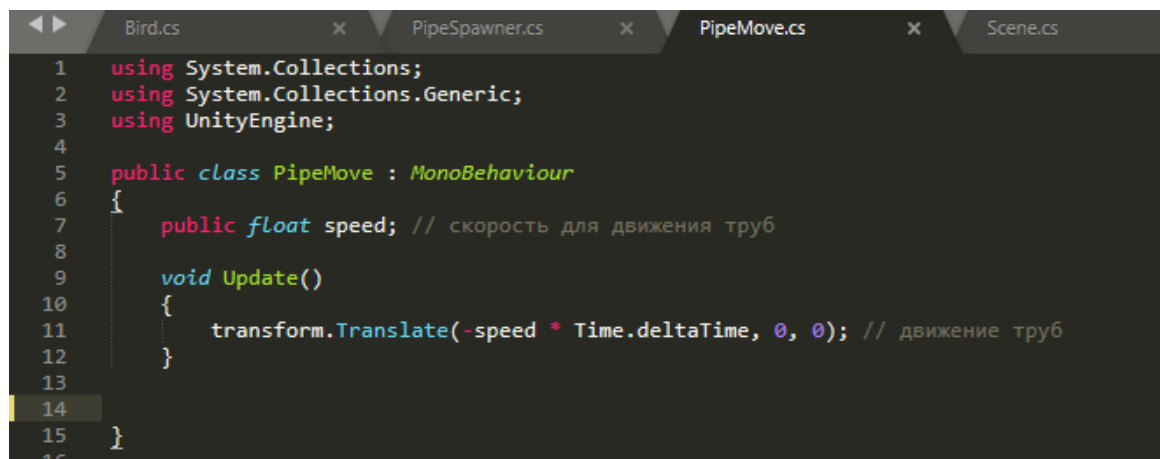
private void OnCollisionEnter2D(Collision2D collision) // проверяем столкновение
{
    if (collision.collider.tag == "Enemy") // если тэг объекта "Enemy"
    {
        Destroy(gameObject); // то птичка уничтожается
        Time.timeScale = 0; // время останавливается
    }
}

```

Рис 11. Столкновение объектов

Далее переходим к скриптам для труб. Для труб используется два скрипта. Первый PipeMove, а второй PipeSpawner. PipeMove используется для задания направления движения труб и скорости движения труб. PipeSpawner отвечает за периодическое появление труб перед игроком.

Так же в данном скрипте задаётся периодичность появления труб и позиция появления, позиция произвольная, но не выходящая за верхние или нижние границы экрана. Дополнительно в PipeSpawner производится оптимизация приложения, задаётся удаление труб которые покидают пределы экрана, трубы «существуют» 10 секунд, после чего удаляются, что бы не нагружать устройство. Наглядно увидеть код скрипта для движения труб вы можете на рисунке 12. код для движения труб



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PipeMove : MonoBehaviour
6 {
7     public float speed; // скорость для движения труб
8
9     void Update()
10    {
11        transform.Translate(-speed * Time.deltaTime, 0, 0); // движение труб
12    }
13
14 }
15
16
```

Рис 12. Код для движения труб

Далее переходим к скрипту для появления труб. Что бы было проще, группируем объекты труб сверху и снизу в группу трубы. Далее уже скрипт будет предназначен для этой группы и при настройке появление труб на произвольной высоте, свободное место для пролёта птицы будет везде одинаковых, будет лишь меняться высота труб, а так же их общее положение на экране. Так же здесь прописываем частоту появления группы труб. Как вы можете наблюдать на рисунке 13. код появления труб, мы используем цикл, каждые две секунды появляется новая группа труб, с произвольной позицией по координате Y. Так же указываем диапазон положений Y, в нашем случае от -1 до 4. И в заключении прописываем условие удалять предыдущие объекты труб, а не нашу группу, в общем, что бы Unity было что создавать.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PipeSpawner : MonoBehaviour
6  {
7      public GameObject Pipes; // переменная для префабов
8
9      void Start()
10     {
11         StartCoroutine(Spawner()); // включаем корутину "Spawner"
12     }
13
14     IEnumerator Spawner() // собсна сама корутина
15     {
16         while (true) // бесконечный цикл while - работает
17         {
18             yield return new WaitForSeconds(2); // ждем 2 секунды
19             float rand = Random.Range(-1f, 4f); // случайная позиция от -1 до 4 (чтоб удобнее было)
20             GameObject newPipes = Instantiate(Pipes, new Vector3(2, rand, 0), Quaternion.identity); // переноси
21             Destroy(newPipes, 10); // удаление нового gameObject'a через 10 секунд (если б удаляли Pipes - то ниче
22         }
23     }
24
25 }
26
27

```

Рис 13. Код появления труб

Теперь перейдем к подсчету очков. Для подсчета очков в самом Unity создали пустой объект и дали ему компонент Box Collider 2D, что бы наша птица могла, как бы соприкоснуться с ним, на самом же деле это просто прозрачное поле между верхней и нижней трубой. Всё это назвали ScoreZone и присвоили тэг «Score», наглядно вы можете увидеть это на рисунке 14.

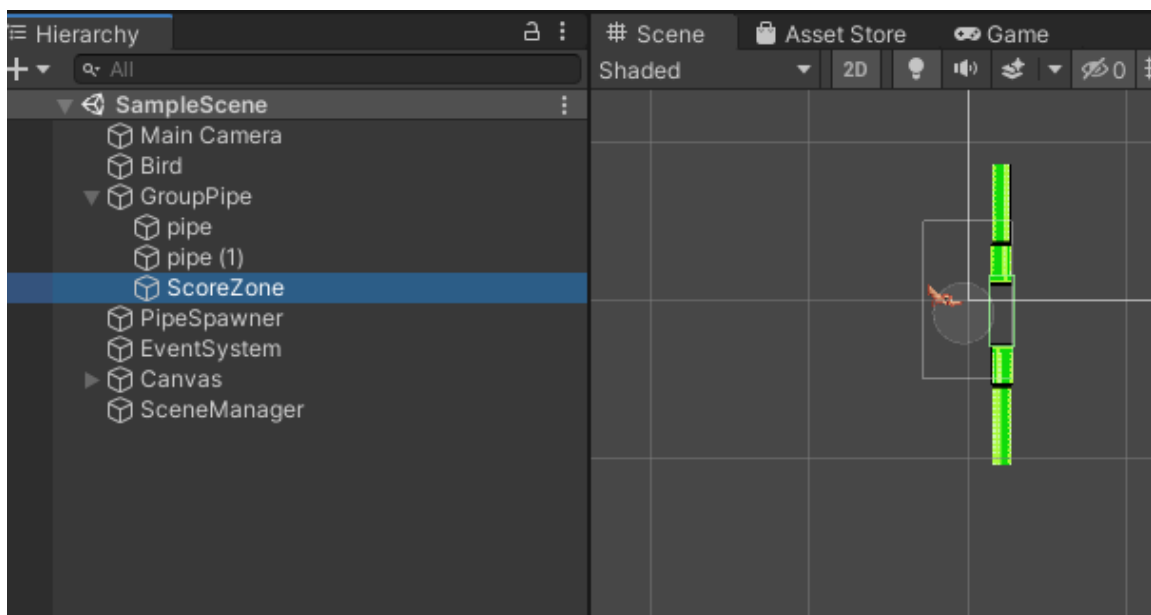


Рис 14. ScoreZone

Тем самым пролетая между трубами, фиксируется преодоления препятствия и игроку присваивается одно очко.

Для отображения очков на экране, мы добавляем новый объект Canvas в Unity, в который сразу же добавляем как подгруппу подчинённых объектов добавляем объект текст, который и будет показывать текущее количество очков. На рисунке 15. подключение Canvas показана структура проекта с добавлением двух новых объектов, а так же можете наблюдать всю иерархию объектов добавленных в проект.

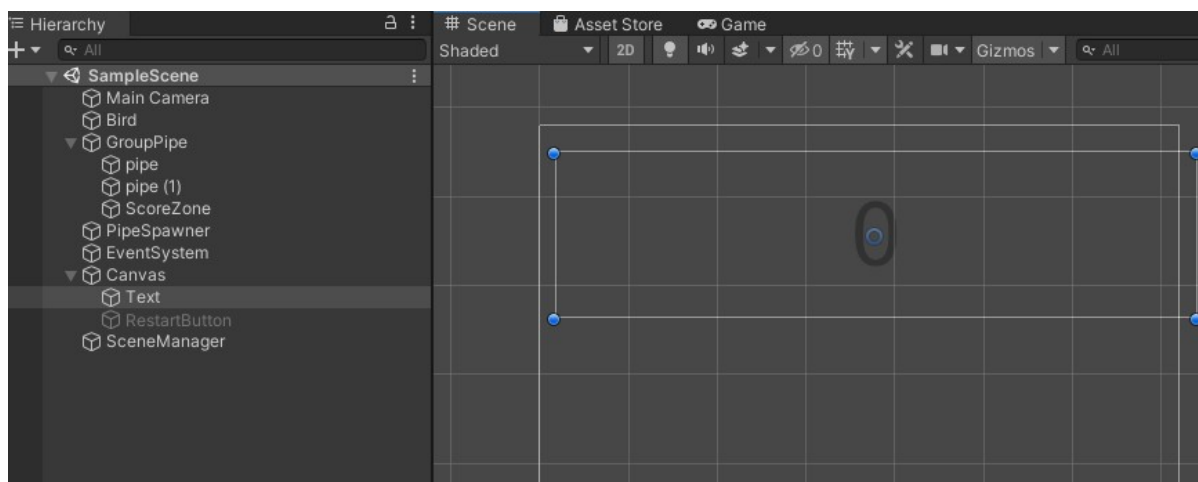


Рис 15. Подключение Canvas

Скрипт для подсчета очков представлен наглядно на рисунке 16. код накопления очков. Здесь добавляется переменная для очков и переменная для текста который будет отображать количество набранных очков. При старте количество очков всегда будет 0. Так же в данном скрипте будет связь с нашей ScoreZone, а именно если птичка будет проходит через объект с тэгом «Score», то будет прибавляться одно очко. Не забываем в данном скрипте подключить библиотеку UI, для этого после всех подключенных библиотек необходимо дописать:

```
Using UnityEngine.UI
```



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Score : MonoBehaviour
7  {
8      public int score; // переменная для очков
9      public Text scoreText; // переменная для текста
10
11     void Start()
12     {
13         score = 0; // при старте кол-во очков будет равнять 0
14     }
15
16     void Update()
17     {
18         scoreText.text = score.ToString(); // это связь очков и текста
19     }
20
21     private void OnTriggerEnter2D(Collider2D collision) // метод для прохода через объект
22     {
23         if (collision.tag == "Score") // если птичка проходит через объект с тэгом "Score"
24         {
25             score++; // то прибавляется одно очко (ну типо score = score + 1)
26         }
27     }
28
29 }
30
31 }
32

```

Рис 16. Код накопления очков

Переходим к добавлению последнего элемента, а именно кнопки рестарта. Добавляем объект кнопка на сцену и накладываем на него текстуру, которую скачали, а далее переходим к написанию скрипту, код скрипта на рисунке 17. код перезапуска игры.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class Scene : MonoBehaviour
7  {
8      public void Restartlevel()
9      {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().name); // перезагрузка сцены (но это я чет запарился)
11
12         // можно было написать вот так: SceneManager.LoadScene(0) - тот же эффект
13     }
14
15 }
16
17

```

Рис 17. Код перезапуска игры

Здесь просто прописываем действие перезагрузки сцены, а далее добавляем данный скрипт на объект кнопка.

## 2.3 Перенос на Android

Теперь, когда всё собрано, готово, проверено и работает, можно смело приступать к сборке APK файла, но перед этим следует внести несколько изменений в настройки проекта. Для этого снова следуем File > Build settings и там нажимаем «Player settings».

Теперь в открывшемся окне пишем название компании которое нам нравится и название приложение, а так же указываем версию. Дополнительно можно изменить иконку приложения на свою. Пример представлен на рисунке 18. настройки проекта.

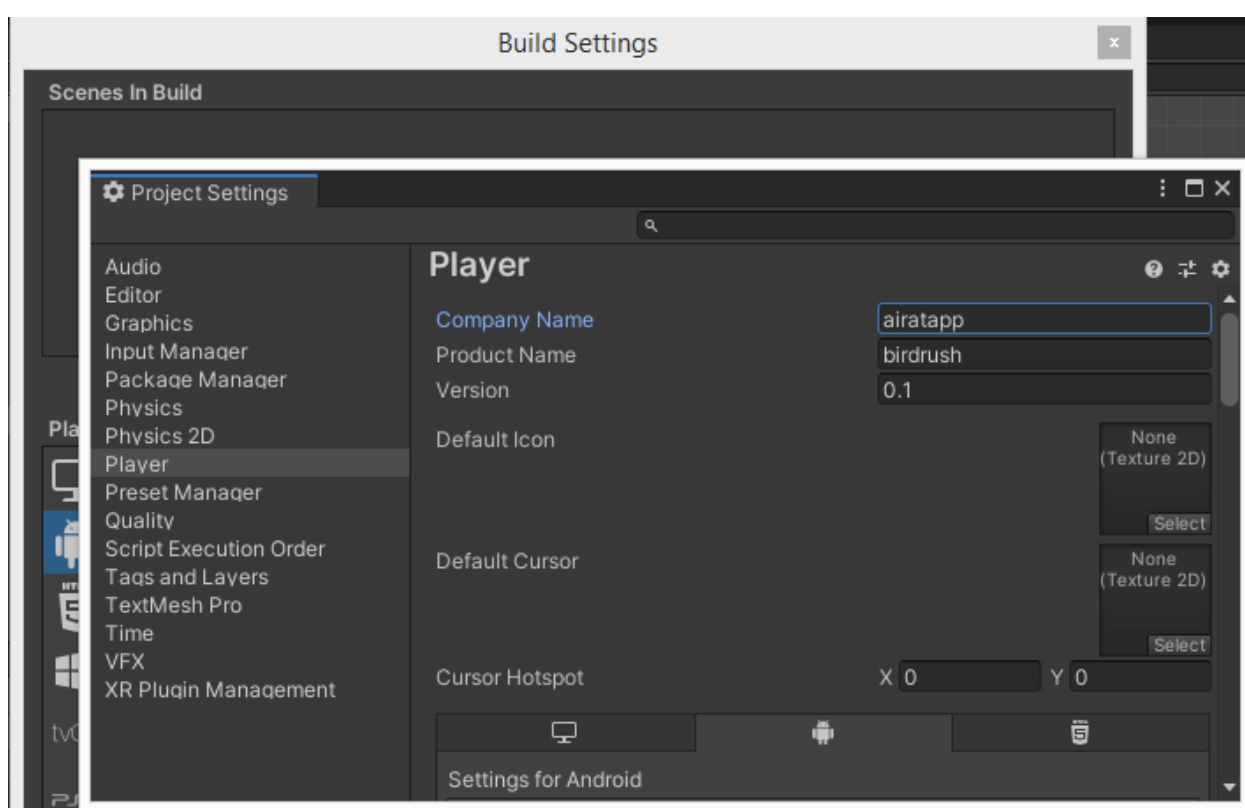


Рис 18. Настройки проекта

Так же в данном окне следует найти раздел Resolution and Presentation и найти пункт Default Orientation, в котором следует выбрать Portrait, данный пункт даст понять смартфону, что приложение всегда должно быть в портретной ориентации. Пример на рисунке 19. ориентация приложения.

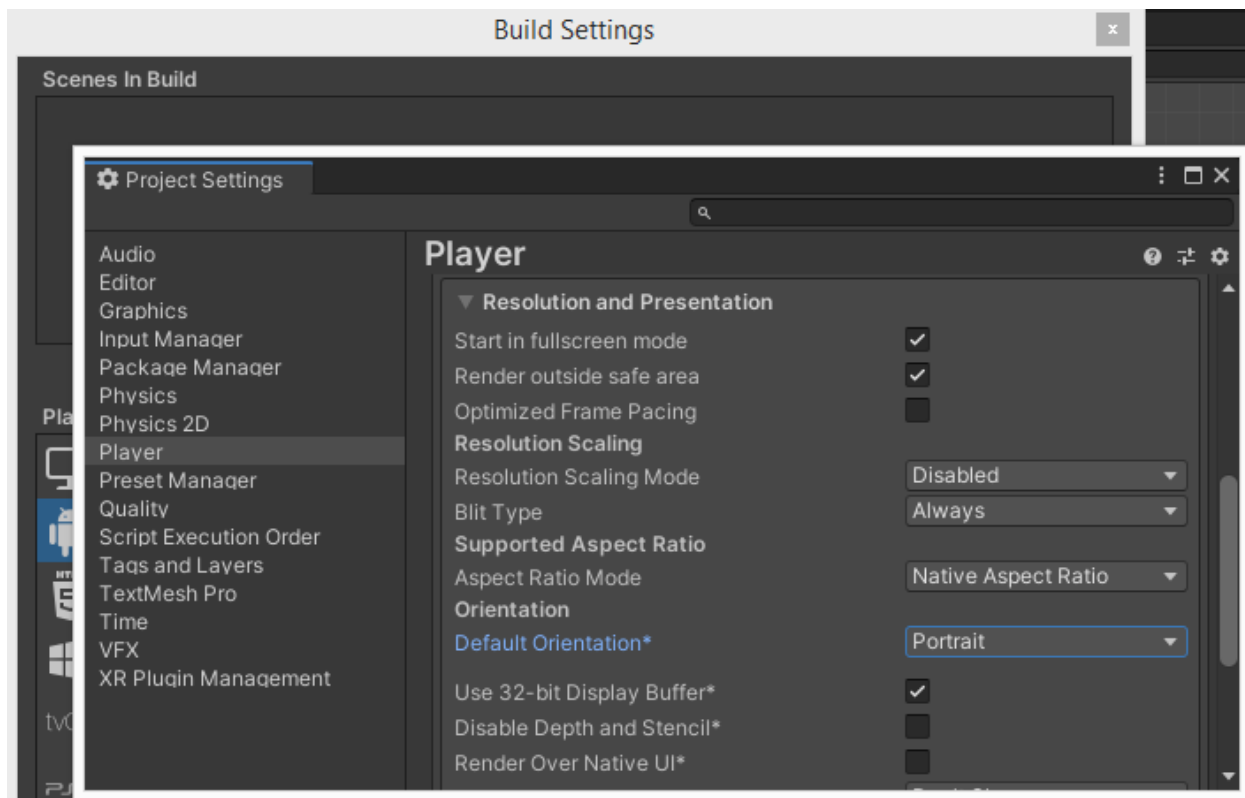


Рис 19. Ориентация проекта

В настройках изображения так же можно поставить галочку, что бы приложение всегда запускалось в полноэкранном режиме, что бы лишний раз не беспокоить пользователей с вырезом под «челку» на экране лишним уведомлением.

Остаётся последнее по списку, но не последнее по важности, а именно архитектура приложения. В пункте Other setting, необходимо найти пункт Scripting Backend и вместо Mono выбрать IL2CPP, после чего ниже станет доступна к выбору архитектура ARM64, которая используется для большинства приложений. Наглядно данные пункты можно увидеть на рисунке 20. настройка архитектуры приложения.

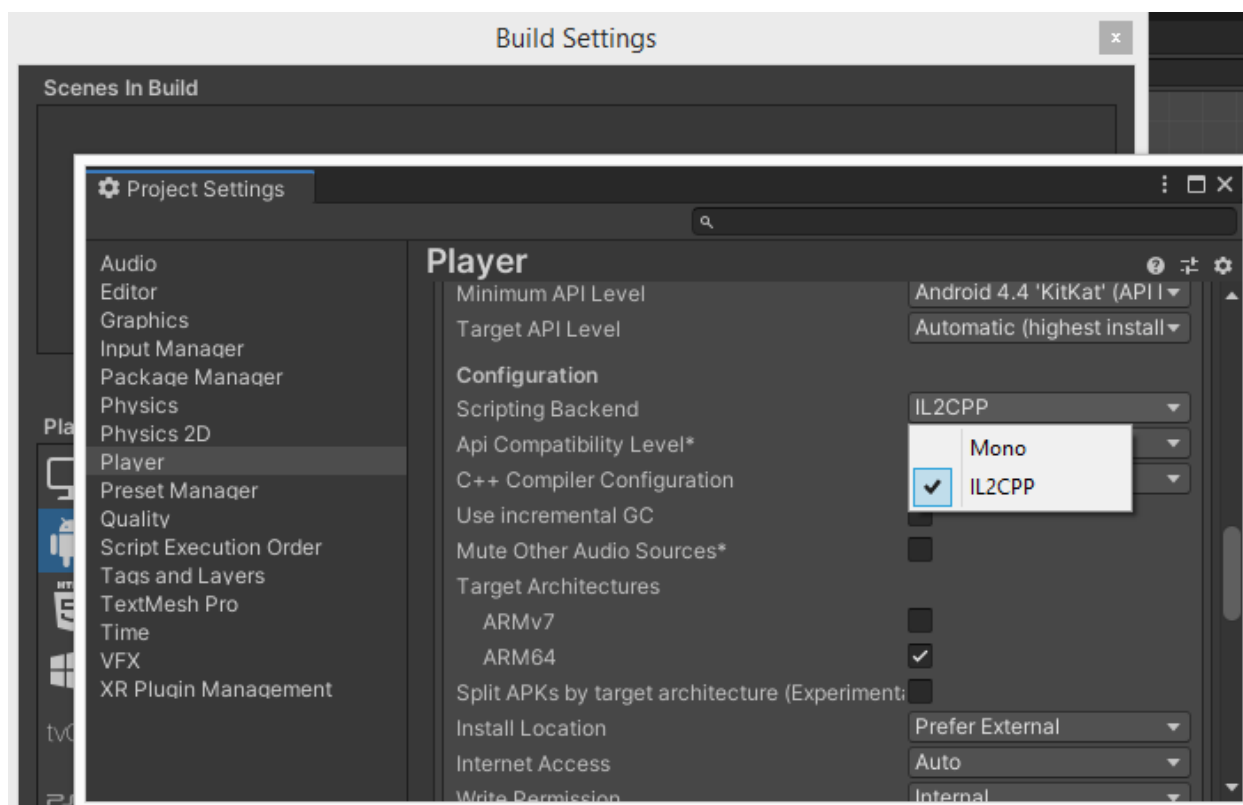


Рис 20. Настройка архитектуры приложения

Когда все выше перечисленные пункты выполнены, можно закрыть окно Player settings и нажать на кнопку «Build», после чего вам будет предложено выбрать местоположение, куда будет сохранен собранный APK файл, нажимая «Ок» вы дожидаетесь завершения компиляции, после чего можете перенести свой APK на Android, установить и играть.

## ЗАКЛЮЧЕНИЕ

Количество среднего времени в день, которое человек проводит в телефоне, играя в игры или находясь в социальных сетях и просто просторах сети интернет, следовательно, всё больше нового и интересного необходимо создавать для телефонов. Так и мы освежили старую игру Flappy Bird, заменив текстуры и перенесли её на новый игровой движок. Люди используют телефон и играют, где бы они ни были, а кто-то даже на ходу, и для этого нужна простая, но интересная игра, что мы и сделали. Игра в одно нажатия достаточно проста для любых условий, где бы вы ни были, а соревноваться с друзьями по набранным очкам добавит интереса вам и вашим знакомым.

По результату проделанной работы были достигнуты следующие цели и задачи:

1. Изучено влияние игр на жизнь людей;
2. Изучены сегменты рынка мобильных игр;
3. Произведено сравнение сред разработки и выбрана нужная нам среда разработки;
4. Разработанная игра для мобильных устройств на базе Android в программе Unity.

Выделены перспективы дальнейшего развития нашего мобильного приложения:

1. Добавление сохранения лучшего результата – таблица личных рекордов;
2. Создание вкладки с достижениями, например, достигнуть 100 очков, что тоже добавит интереса и подтолкнет соревновательный момент;
3. Создание онлайн таблицы рекордов, что бы соревноваться с игроками по всему миру;
4. Добавление отдельного главного экрана отдельно от основной сцены;

									Лист
									43
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР				

5. Добавление магазина с возможностью изменения облика птицы;
6. Выбор уровней сложности, чем выше уровень, тем быстрее движется птицы.

					ТГКГЗ.ПИ.22.00.000.КР	Лист
Изм.	Лист	№ документа	Подпись	Дата		44

## СПИСОК ЛИТЕРАТУРЫ

1. Android Studio Features. [Электронный ресурс] URL: <https://developer.android.com/studio/features.html>
2. Appery.io: Enterprise Mobile App Builder & MBaaS. [Электронный ресурс] URL: <https://appery.io/>
3. Download Android Studio and SDK Tools | Android Studio. [Электронный ресурс] URL: <https://developer.android.com/studio/index.html>
4. Eclipse - The Eclipse Foundation open source community website. [Электронный ресурс] URL: <https://www.eclipse.org/downloads/>
5. Fork of Tesseract Tools for Android. [Электронный ресурс] URL: <https://github.com/rmtheis/tess-two/>
6. IntelliJ IDEA the Java IDE – JetBrains. [Электронный ресурс] URL: <https://www.jetbrains.com/idea/>
7. Introduction to Material Design. [Электронный ресурс] URL: <https://material.google.com/>
8. Kaner, Falk, Nguyen. Testing Computer Software. – USA: Wiley Computer Publishing, 1999. – 42 p.
9. MVP and MVC Architectures in Android. [Электронный ресурс] URL: <https://www.techyourchance.com/mvp-mvc-android-1/>
10. Shoutem - Make an App - Build Apps with Easy Application Creator. [Электронный ресурс] URL: [www.shoutem.com/](http://www.shoutem.com/)
11. Арлоу Дж., Нейштадт А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. 2-е издание. – М.: Издательство «Символ-Плюс», 2007. – 624 с. 37
12. Архитектура Android-приложений. Часть II – архитектурные стили и шаблоны. [Электронный ресурс] URL: <https://habrahabr.ru/post/140655/>
13. Бурнет Э. Привет, Android! Разработка мобильных приложений. 2-е издание. – СПб.: Издательство «Питер», 2012. – 256 с.

										Лист
										45
Изм.	Лист	№ документа	Подпись	Дата	ТГКГЗ.ПИ.22.00.000.КР					

14. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. – СПб.: Издательство «Питер», 2003. – 432 с.

15. ВЕДОМОСТИ – Интернет-аудитория России растет за счет мобильных устройств. [Электронный ресурс] URL: <https://www.vedomosti.ru/technology/articles/2016/01/28/625779-internet-auditoriya-rossiirastet-schet-mobilnih-ustroistv/>

16. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е издание. – СПб.: Издательство «Питер», 2016. – 512 с.

17. Нильсен Я., Будиу Р. Mobile Usability. Как создавать идеально удобные приложения для мобильных устройств. – М.: Эксмо, 2013. – 256 с.

18. Операции. [Электронный ресурс] URL: <https://developer.android.com/guide/components/activities.html>

19. Основы создания приложений. [Электронный ресурс] URL: <https://developer.android.com/guide/components/fundamentals.html>

20. Приложения в Google Play – Clever Bill Splitter. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.cleverturtles.splitter&hl=ru>

21. Основные этапы разработки мобильных приложений [Электронный ресурс] – Режим доступа: <https://spark.ru/startup/componentix/blog/4499/osnovnie-etapirazrabotki-mobilnih-prilozhenij>

22. Официальная документация по Android [Электронный ресурс]. - <https://developer.android.com/guide/> 54

23. Приложение Google I / O 2017 для Android [Электронный ресурс] - <https://github.com/google/iosched>

24. Программирование для android, java [Электронный ресурс] <http://davidmd.ru/%D1%83%D1%80%D0%BE%D0%BA%D0%B8-%D0%BF%D0%BE-android/>



25. Программирование на Java [Электронный ресурс] -  
<http://studyjava.ru/category/uroki-java/>
26. Руководство разработчика [Электронный ресурс] -  
<http://mybiblioteka.su/tom3/7-59852.html>
27. Русская документация Android [Электронный ресурс] -  
<http://easyandroid.ru/index.php?p=721>
28. Тестирование android приложений с помощью реальных устройств [Электронный ресурс] - <http://www.fandroid.info/testirovanie-androidprilozhenij-s-pomoshhyu-realnyh-ustrojstv/>
29. Уроки по Java [Электронный ресурс] - <https://javarush.ru/>
30. Уроки программирования [Электронный ресурс] -  
<http://learnandroid.ru/index.html>
31. Уроки программирования от Google [Электронный ресурс] -  
<https://developer.android.com/training/index.html>

						Лист
					ТГКГЗ.ПИ.22.00.000.КР	47
Изм.	Лист	№ документа	Подпись	Дата		

**Руководство пользователя мобильного приложения.**

**1. Основные требования к взаимодействию**

Для запуска мобильного приложения лишь необходимо скачать и установить APK. Соединение с сетью Интернет требуется только для скачивания, установка и сам игровой процесс не требуют подключения к сети Интернет.

**1.1 Взаимодействие пользователя с Интерфейсом**

Взаимодействие пользователя с Интерфейсом происходит через нажатия на экран смартфона. Используются следующие виды нажатий:

- Одиночное касание без удержания.

**1.2 Уровень подготовки пользователя**

Никаких требований к уровню подготовки пользователя не имеется. Весь игровой процесс максимально простой.

**1.3 Требования к аппаратному и программному обеспечению**

Мобильное приложение крайне нетребовательное, поэтому запуститься даже на самом простом смартфоне на Android.

Требования к рабочему месту пользователя:

- Операционная система: Android 4.4+;
- Процессор и оперативная память: от 1.2GHz и от 1Gb;

**2. Доступ к мобильному приложению.**

Для того чтобы скачать наше мобильное приложение необходимо отсканировать QR-код (см. рис. 1). Данный QR-код содержит ссылку на файл

APK, который хранится на Google Диск. После сканирования откроется Google Диск и будет предложена установка приложения



Рис. 1 QR-код для скачивания

### 3. Работа с мобильным приложением

Всё управление состоит из нажатий на экран. Как только игра запускается, сразу же начинается игровой процесс. Вам необходимо нажимать на экран что бы персонаж двигался вверх, либо не касаться экрана, что бы снизить высоту и избежать столкновения.

После запуска приложения открывается главное игровое окно мобильного приложения (см. рис. 2)



Рис. 2 Главное окно мобильного приложения

Вот как выглядит игра в случае столкновения или вылета за пределы экрана. Время в игре останавливается и появляется кнопка для перезапуска уровня (см. рис. 3)

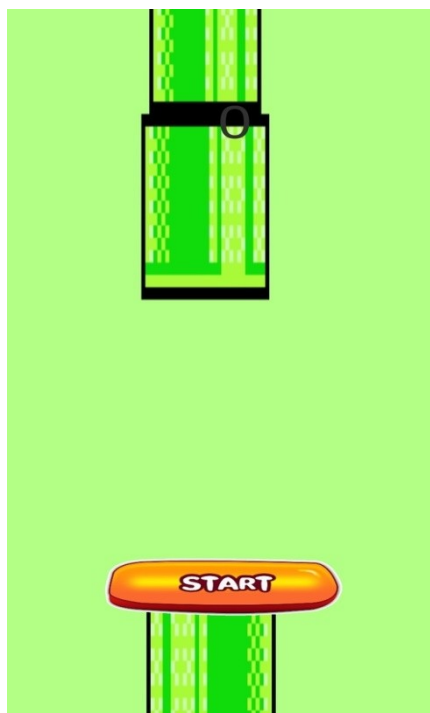
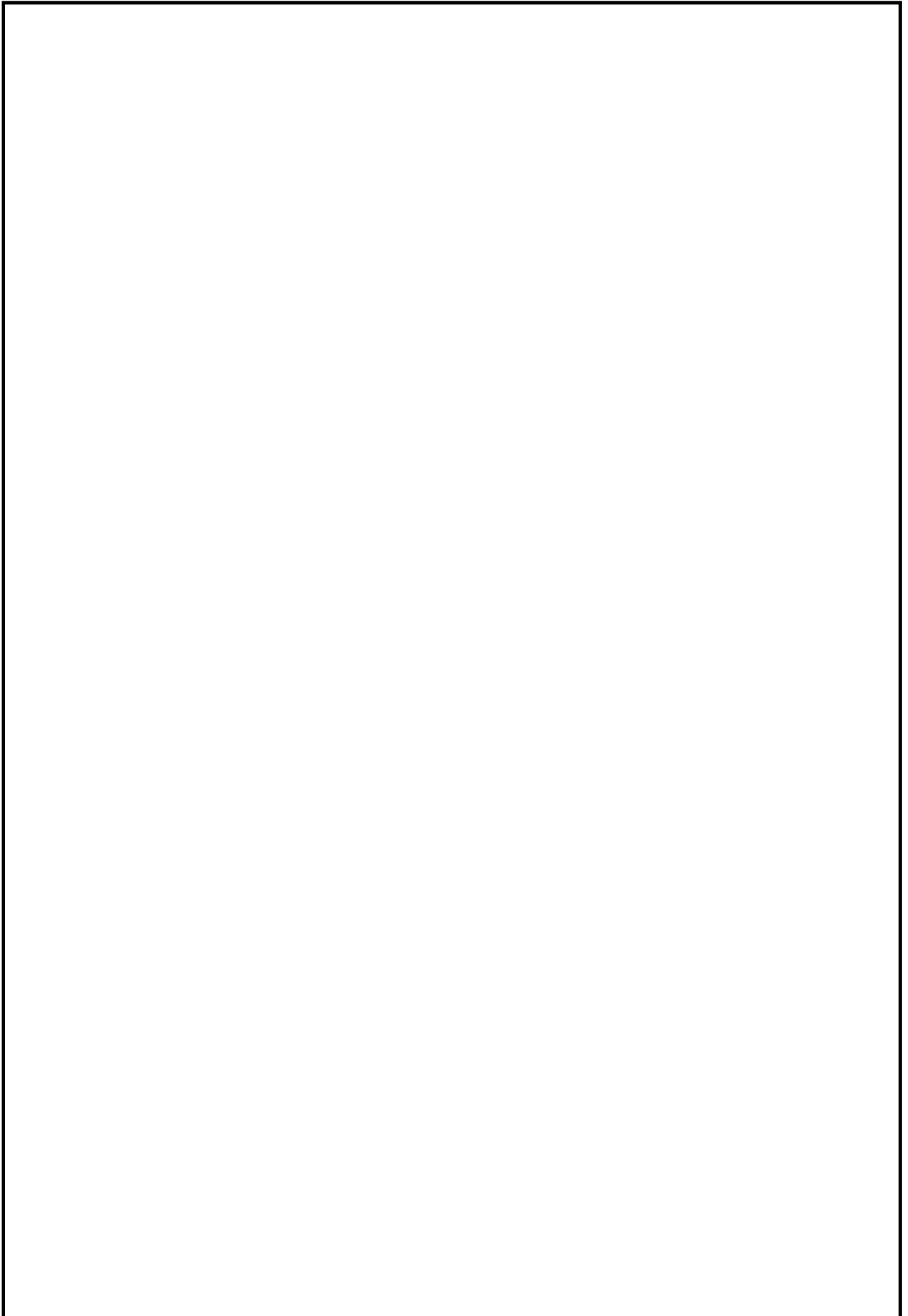


Рис. 3 Окно игры после проигрыша

#### 4. Общие возможности пользователей мобильного приложения

Любой пользователь может скоротать время за игрой в наше мобильное приложение. При этом неважно, где пользователь находится, ведь для игры не нужен интернет, а так же не нужно много усилий.



					ТГКГЗ.ПИ.22.00.000.КР	Лист
						51
Изм.	Лист	№ документа	Подпись	Дата		