

РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ ЖАНРА ПЛАТФОРМЕР НА ДВИЖКЕ UNITY С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ C#: «LEGENDS OF PLANET Z-ZERRA: BLACK AND WHITE WORLD»

Зайнутдинов А.Д.

г.Казань, МБОУ «Гимназия №6», 6 «З» класс

Руководитель: Смиронова А.В., г.Казань, МБОУ «Гимназия №6»,
учитель информатики,

Данная статья является реферативным изложением основной работы. Полный текст научной работы, приложения, иллюстрации и иные дополнительные материалы доступны на сайте V Международного конкурса научно-исследовательских и творческих работ учащихся «Старт в науке» по ссылке: <https://school-science.ru/5/4/34488>

21 век – век стремительного развития информационных технологий. Они позволяют быстро решать поставленные задачи, экономить время и достигать максимально комфортного уровня жизни. Современные смартфоны, планшеты и другие гаджеты люди носят с собой повсюду, поэтому мобильные технологии стали неотъемлемой частью нашей жизни. Вместе с увеличением рынка персональной электроники так же стремительно быстро растет рынок развлечений – поэтому осуществление проектов по созданию игр в настоящее время очень актуально и можно смело утверждать, что востребованность таковых не уменьшится в ближайшие десятилетия.

Целью данной работы является разработка игры жанра платформер на движке Unity с использованием языка программирования C#.

Основные задачи исследования:

1. Выбор инструментального программного обеспечения;
2. Изучение работы с графикой;
3. Анализ возможностей использования редактора Unity;
4. Реализация программы.

Методы исследования:

1. Анализ литературы;
2. Метод моделирования.

Платформеры, как игровой жанр, появились в начале 1980-х годов, когда игровые консоли не были достаточно мощными. Действие игры не выходило за пределы одного экрана. Персонаж лазал вверх и вниз по лестницам или прыгал с платформы на платформу, часто сражаясь с противниками и преодолевая препятствия. Вскоре процесс прохождения уровня перестал быть в основном вертикальным и стал горизонтальным с появлением длинных многоэкранных прокручивающихся игровых миров. Самым яр-

ким представителем игры платформера того времени до сих пор считается видеоигра Super Mario Bros., выпущенная в 1985 году фирмой Nintendo [4].

Теоретическая значимость данной работы заключается в том, что приобретенный мною опыт на всех этапах разработки данного мобильного приложения позволит в будущем гораздо увереннее ставить перед собою цели по проектированию более сложных игр и достигать их. К тому же данная работа сможет привлечь внимание других ребят школьников к программированию, покажет, что процесс создания игры увлекательный и познавательный, позволяет воплотить в жизнь все свои идеи и задумки. И в конечном итоге воспользоваться всеми полезными свойствами конечного продукта – игры.

Приспособленность игры к разным мобильным устройствам, и, соответственно к самым распространённым платформам (Android, iOS, MacOS, Windows и др.) является отличной перспективой для ее успешного продвижения, привлечения большого количества аудитории при реализации конечному пользователю.

Для начала можно разработать игру для одной операционной системы (ОС), в нашем случае для Android. В дальнейшем, проанализировав все недостатки, чтобы избежать их повторения, портировать игру на другие ОС.

Компьютерные игры пользуются неослабевающим интересом. Они приносят немало пользы. Многие обучающие программы в своей основе содержат игровую составляющую. Игры учат человека быстро реагировать, принимать решения, при этом ощущая свою ответственность за это. Они способствуют развитию образного мышления, критического, стратегического мыш-

ления. Развивают мелкую моторику, учат планировать свои действия. Могут помочь с выбором профессии. Кроме развивающей и обучающей функции, игры позволяют человеку провести досуг, отвлечься от дел и просто отдохнуть.

ГЛАВА 1. ОПИСАНИЕ ИНСТРУМЕНТОВ РАЗРАБОТКИ ДЛЯ СОЗДАНИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ В UNITY

1.1 Язык программирования C#

C# (произносится как «си шарп») – простой, современный объектно-ориентированный язык программирования, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных и самоописательных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов – это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие сведения о компоненте, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому C# отлично подходит для создания и применения программных компонентов [9],[10].

Вот несколько функций языка C#, обеспечивающих надежность и устойчивость приложений:

- Сборка мусора автоматически освобождает память, занятую уничтоженными и неиспользуемыми объектами;
- Обработка исключений дает структурированный и расширяемый способ выявлять и обрабатывать ошибки;
- Строгая типизация языка не позволяет обращаться к неинициализированным переменным, выходить за пределы массива или выполнять неконтролируемое приведение типов.

В C# существует единая система типов. Все типы C#, наследуют от одного корневого типа `object`. Таким образом, все типы используют общий набор операций, и значения любого типа можно хранить, передавать и обрабатывать схожим образом [1][2].

1.2 Платформа Unity

Unity – межплатформенная среда разработки компьютерных игр. Unity позволяет создавать приложения, работающие под более чем 20 различными операционными системами, включающими персональные компьютеры, игровые консоли,

мобильные устройства, Интернет-приложения и другие [12].

Выпуск Unity состоялся в 2005 году и с того времени идет постоянное развитие. Изначально Unity предназначался исключительно для компьютеров Mac, но потом постепенно выходили обновления, позволяющие работать под Windows и другие ОС.

Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

На Unity написаны сотни игр, приложений и симуляций, которые охватывают множество платформ и жанров. Вместе с тем Unity используется как крупными разработчиками, так независимыми студиями.

Редактор Unity имеет простой Drag&Drop интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два скриптовых языка: C#, JavaScript (модификация).

Проект в Unity делится на сцены (уровни) – отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты – объекты, которые не имеют модели («пустышки»). Объекты, в свою очередь содержат наборы компонентов, с которыми и взаимодействуют скрипты. К объектам можно применять коллизии (в Unity т. н. коллайдеры – `collider`). В редакторе имеется система наследования объектов – дочерние объекты будут повторять все изменения позиции, поворота и масштаба родительского объекта. Скрипты в редакторе прикрепляются к объектам в виде отдельных компонентов.

При компиляции проекта создается исполняемый(.exe) файл игры (для Windows), а в отдельной папке – данные игры (включая все игровые уровни и динамически подключаемые библиотеки) [5].

1.3 Среда разработки MonoDevelop

MonoDevelop IDE (интегрированная среда разработки) (<http://monodevelop.com>) – свободная мультиплатформенная среда разработки, предназначенная для создания приложений на языках C#, C, C++, Java, Visual Basic.NET, CIL, Nemerle, Boo. Встроен в дистрибутив Unity3D как средство написания скриптов [11].

Возможности:

1. Подсветка синтаксиса – выделение синтаксических конструкций текста с использованием различных цветов, шрифтов и начертаний. Применяется для облегчения чтения исходного текста компьютерных программ, улучшения визуального восприятия.

2. Сворачивание кода, или фолдинг (англ. folding) – одна из функций текстового редактора, позволяющая скрывать определённый фрагмент редактируемого кода или текста, оставляя лишь одну строку.

К примеру, фолдинг функции приводит к сворачиванию всего кода функции в одну строку таким образом, что будет видно только название функции. Обычно, чтобы свернуть фрагмент, нужно нажать на символ «-» слева от него. Чтобы увидеть весь фрагмент, то есть развернуть его, нужно нажать на символ «+», появившийся у свернутых фрагментов.

3. Автодополнение кода – функция в программах, предусматривающих интерактивный ввод текста (редакторы, оболочки командной строки, браузеры и т. д.) по дополнению текста по введённой его части.

4. Встроенный отладчик – используется на этапе отладки компьютерной программы, на котором обнаруживают, локализируют и устраняют ошибки. Чтобы понять, где возникла ошибка, приходится:

- узнавать текущие значения переменных;
- выяснять, по какому пути выполнялась программа.

5. Модульное тестирование – единичное тестирование, или модульное тестирование (англ. unit testing) – процесс в программировании, позволяющий проверить на корректность единицы исходного кода, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок [11].

1.4 Редактор скелетной анимации Spriter

Скелетная анимация – способ анимирования двумерных и трёхмерных моделей в мультипликации и компьютерных играх.

Заключается в том, что мультипликатор или модельер создаёт скелет, представляющий собой как правило древообразную

структуру костей, в которой каждая последующая кость «привязана» к предыдущей, то есть повторяет за ней движения и повороты с учётом иерархии в скелете. Далее каждая вершина модели «привязывается» к какой-либо кости скелета. Таким образом, при движении отдельной кости двигаются и все вершины, привязанные к ней. Благодаря этому задача аниматора сильно упрощается, потому что отпадает необходимость анимировать отдельно каждую вершину модели, а достаточно лишь задавать положение и поворот костей скелета.

Также благодаря такому методу сокращается и объём информации, необходимой для анимирования. Достаточно хранить информацию о движении костей, а движения вершин высчитываются уже исходя из них [6].

Spriter – программа для создания костной анимации, работает с растровой графикой. Использует «модульный» метод создания плавных анимаций. Этот способ имеет множество преимуществ [3]:

1. Экономит время – неважно, являетесь ли вы опытным или начинающим художником по дизайну игр, вы сможете тратить гораздо меньше времени на настройку и отладку анимируемых вами персонажей, так как это позволит вам повторно использовать только несколько модульных изображений.

2. Мгновенная итерация. Предположим, что необходимо изменить конфигурацию головы игрового персонажа на этапе окончательной разработки. Если у вас есть Spriter в 2D-панели инструментов, вам нужно будет только изменить небольшую часть изображений головы, потому что модули настроены для использования во всех кадрах анимации.

3. Пользовательская настройка. Spriter намного упрощает работу дизайнера в плане создания любых трюков, поскольку модульные изображения (части тела персонажей) можно свободно подталкивать или поворачивать.

4. Неограниченные изменения персонажей. Этот метод обеспечивает сверхбыстрое и безболезненное создание новых персонажей на основе данных уже созданного персонажа. Так же это эффективный с точки зрения памяти способ создания инструментов персонажа, которые могут изменяться в течении игры (например: сбор бонусов, новое оружие и т.п.).

1.5 Графический редактор Adobe Illustrator CC 2017

Adobe Illustrator – векторный графический редактор, разработанный и распространяемый фирмой Adobe Systems [7]. Одно из передовых приложений для дизайна векторной графики, позволяющее создавать

логотипы, значки, рисунки, типографику и иллюстрации для печатных изданий, веб-публикаций, видео и мобильных устройств.

Превосходное впечатление независимо от размера экрана. Полный набор инструментов для рисования превращает простые фигуры и цвета в изящные логотипы, значки и графические объекты. Приложение Illustrator позволяет создавать векторную графику, которая адаптируется к размерам экранов мобильных устройств или рекламных щитов без ущерба для качества и четкости изображения. Превосходная типографика говорит сама за себя. Можно встраивать названия компаний в логотипы, создавать листовки или макеты вебсайтов с помощью лучших инструментов для работы со шрифтами. Так же можно добавлять эффекты, менять стили и редактировать отдельные символы, чтобы типографический дизайн точно выражал идею.

Illustrator регулярно обновляется, и его функциональность становится шире [8].

Глава 2. Описание игрового приложения

2.1 Актуальность проекта

Люди хотят постоянно находиться онлайн. Современные смартфоны, планшеты и другие гаджеты они носят с собой повсюду. Возможность подключиться к сети Интернет есть практически в любой точке планеты. Вместе с появлением мобильного интернета огромной актуальностью стала пользоваться такая услуга, как разработка мобильных приложений и игр. Рынок игр увеличивается в геометрической прогрессии. С каждым годом все больше и больше людей играют в мобильные игры. Причем пользоваться популярностью на рынке сейчас могут не только известные производители игр с большим штатом сотрудников, но и один человек при наличии достаточного количества знаний и времени. Инструменты для разработки мобильных приложений постоянно совершенствуются, поэтому игры не становятся хуже от того, что разрабатываются в менее масштабной среде. Они могут выглядеть проще, меньше, но быть при этом очень интересными и востребованными.

2.2 Общая постановка проекта

Основной целью данного проекта является разработка игрового приложения на Unity с использованием языка программирования C# и такого инструментального программного обеспечения, как среда разработки MonoDevelop, редактор скелетной анимации Spriter, графический редактор Adobe Illustrator CC 2017.

2.3 Назначение программного продукта

В проекте представлено игровое приложение на языке программирования C#. Основное назначение игры заключается в том, что с ее помощью можно отлично скоротать время. В любую скучную минуту время пролетит очень быстро. Кроме этого игра может значительно поднять настроение и надолго зарядить позитивными эмоциями. С помощью игры развивается мышление. Игра учит анализировать свои действия. Развивается настойчивость и выносливость. Ребенок может улучшить навыки чтения, уровень знания иностранного языка. Улучшается реакция, так как в игре надо быстро принимать решения. Игра тренирует память, поскольку необходимо многое запомнить: сюжет, управление, свойства предметов.

2.4 Комплекс технических средств

Для работоспособности игрового приложения приведены минимальные системные требования:

- Android 4.4 или выше;
- Процессор 1.2 GHz или лучше;
- Память 1 GB RAM или выше;
- Свободное место 60 Мб.

2.5 Выполнение проекта

Запустить приложение. После запуска на экране появляется окно с предысторией, где раскрывается сюжетная линия игры, рассказывается о персонажах. В правом нижнем углу данного окна находится кнопка «далее». Ее можно нажать, если Вы хотите пропустить предысторию или после прочтения предыстории (Приложение1, Рисунок 1).

Далее мы попадаем в основное меню игры, которое содержит 3 кнопки (Приложение1, Рисунок 2):

Start – после ее нажатия мы выходим в меню с выбором уровней;

Звук on/off – включение/выключения звуков/музыки;

Предыстория – выход в окно с предысторией игры;

Меню выбора уровней содержит кнопки уровней:

- 1 – уровень 1;
- 2 – уровень 2;
- 3 – уровень 3;
- 4 – уровень 4;
- 5 – уровень 5;

Quit – для выхода в главное меню.

Каждый следующий уровень становится доступным только после прохождения предыдущего.

На следующий уровень можно попасть двумя способами:

- не переходя в меню выбора уровней, непосредственно после прохождения предыдущего уровня, нажав на кнопку «следующий»;

- переходя в меню выбора уровней, нажав на кнопку следующего уровня.

Для того чтобы завершить любой уровень, необходимо его пройти от начала до конца. При этом необходимо преодолеть все препятствия, побороть всех монстров-злодеев, собрать бонусы «звезды» по желанию.

Управление героем осуществляется тремя кнопками, которые находятся внизу экрана:

Стрелка вправо – движение вправо;

Стрелка влево – движение влево;

Стрелка вверх – прыжок.

В верхнем левом углу экрана находится кнопка «пауза» – при ее нажатии игра временно останавливается, появляется окно выбора с кнопками – «продолжить» и «выход» в меню выбора уровней.

В правом верхнем углу отражаются жизни героя в виде сердечек.

У героя есть 4 жизни. Жизнь теряется, если герой не преодолел препятствие, или

его «убил» монстр-злодей. При потере жизни герой возвращается в последнюю точку авто-сохранения и продолжает проходить уровень с этого места. Если герой потерял все 4 жизни, то на экране появляется окно с сообщением «Игра Окончена» и предложением вернуться в меню с выбором уровней или начать этот уровень сначала.

В каждом уровне присутствует несколько точек авто-сохранения, которые активируются после прохождения каждой из данных точек.

Цель прохождения 5 уровня заключается в том, чтобы победить главного злодея База. Для этого надо сразиться с роботами, которых он выпускает. По мере того, как уничтожается каждый робот, главный злодей начинает терять цвет и становится более прозрачным. Когда все роботы будут побеждены, Баз исчезнет. Главный герой зайдет в потайную комнату, нажмет на красную кнопку и разрушит машину, которая забрала все цвета у планеты Z-zetta. Мир снова станет цветным и все будут счастливы. Игра пройдена, главный герой прыгает от радости, идут титры!

Приложение 1

Снимки экрана игры. Коды программы

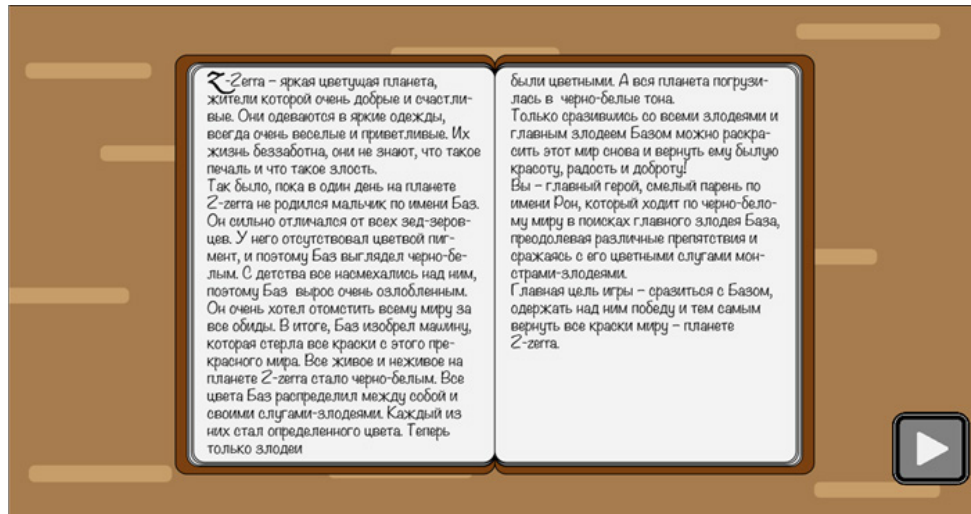


Рис. 1. Предыстория игры

Данный код реализует первый запуск игры, при котором сразу открывается ее предыстория:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class bookStart : MonoBehaviour {
    public GameObject book, start, bg, text;
    void Awake () {
```

```
if (PlayerPrefs.GetInt ("book") != 3) {
    book.SetActive (true);
    start.SetActive (false);
    text.SetActive (false);
    bg.SetActive (false);
    PlayerPrefs.SetInt ("book", 3);
}
}
```

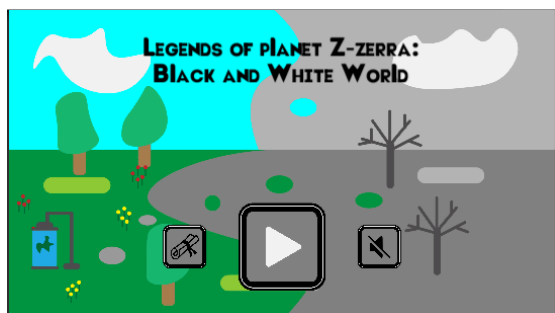


Рис. 2. Основное меню игры

Данный код реализует нажатие центральной кнопки «Start», при котором происходит переход в меню уровней:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class LoadLevelBasic :
MonoBehaviour {
public string sceneName;
void OnMouseDown () {
Time.timeScale = 1;
SceneManager.LoadScene (sceneName);
}
}
```

Заключение

В ходе выполнения данного проекта была проделана значительная работа, начиная с изучения литературы по программированию и заканчивая тестированием и отладкой готового программного продукта. Безусловно это неоценимый опыт в получении теоретических знаний и практических навыков, который пригодится для дальнейших разработок.

В результате мною была создана игра для мобильного устройства, которая в данный момент работает на ОС Android.

Работая самостоятельно у тебя есть прекрасная возможность изучить сразу несколько профессий: программист, художник, аниматор, сценарист, гейм-дизайнер. Это очень интересно и сложно одновременно. Ведь не всегда на практике получается все так гладко, как должно быть в теории. В таких случаях приходится прибегать к нестандартным методам исправления ситуации, в частности исправления ошибок при тестировании.

Создание игрового приложения начинается с идеи. И я очень рад, что сумел воплотить свою идею в жизнь. Это творческий процесс, который несколько месяцев занимал все мое свободное время.

В перспективе было бы интересно загрузить игру на GooglePlay и протестировать ее в реальных условиях.

Данная работа может быть интересна всем, кто хочет попробовать себя в качестве разработчика мобильных приложений, не обязательно только игр. Здесь очень важно понять, что главное, это желание, упорство и труд! И тогда у вас обязательно все получится!

Список литературы

1. Джон Шарп. Microsoft Visual C#. Подробное руководство. – 8-е издание – СПб: Издательство «Питер», 2017. – 836 с.;
2. Краткий обзор языка C#: [Электронный ресурс] // Microsoft – 2016 // Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/> (Дата обращения: 17.03.2018);
3. Особенности Spriter: [Электронный ресурс] // Сайт BrashMonkey // Режим доступа: <https://brashmonkey.com/spriter-features/> (Дата обращения: 17.03.2018);
4. Платформер // Википедия – свободная энциклопедия. – 2018 // Режим доступа: <https://ru.wikipedia.org/wiki/Платформер> (Дата обращения: 10.03.2018);
5. Руководство пользователя Unity (2017,3): [Электронный ресурс] // Unity. Documentation – 2017 // Режим доступа: <https://docs.unity3d.com/Manual/index.html> (Дата обращения: 22.12.2017);
6. Скелетная анимация: [Электронный ресурс] // Википедия – свободная энциклопедия, 2016 // Режим доступа: https://ru.wikipedia.org/wiki/Скелетная_анимация (Дата обращения: 25.03.2018);
7. Adobe Illustrator: [Электронный ресурс] // Википедия – свободная энциклопедия, 2016 // Режим доступа: https://ru.wikipedia.org/wiki/Adobe_Illustrator (Дата обращения: 25.03.2018);
8. Adobe Illustrator CC: [Электронный ресурс] // Официальная страница на русском языке. // Режим доступа: <https://www.adobe.com/ru/products/illustrator.html> (Дата обращения: 25.03.2018);
9. C#: [Электронный ресурс] // Прогopedia – свободная энциклопедия языков программирования. – 2018 // Режим доступа: <http://progopedia.ru/language/csharp/>
10. C Sharp: [Электронный ресурс] // Википедия – свободная энциклопедия. – 2018 // Режим доступа: https://ru.wikipedia.org/wiki/C_Sharp (Дата обращения: 17.03.2018);
11. MonoDevelop: [Электронный ресурс] // Википедия – свободная энциклопедия, 2016 // Режим доступа: <https://ru.wikipedia.org/wiki/MonoDevelop> (Дата обращения: 28.03.2018);
12. Unity (игровой движок): [Электронный ресурс] // Википедия – свободная энциклопедия. – 2018 // Режим доступа: [https://ru.wikipedia.org/wiki/Unity_\(игровой_движок\)](https://ru.wikipedia.org/wiki/Unity_(игровой_движок)) (Дата обращения: 28.03.2018);