

Проектная работа

Информатика

**Индивидуальный проект на тему:**

**«Разработка видеоигры»**

*Выполнила:*

***Дубина Анастасия Александровна***

*учащаяся 10 «А» класса*

*МБОУ СОШ №46 г. Калуги*

*Руководитель:*

***Иванова Татьяна Анатольевна***

*Учитель физики*

*МБОУ СОШ №46 г. Калуги*

## **Введение**

В современном мире довольно-таки трудно представить жизнь без своей любимой игры. Также трудно встретить подростка, который не проводит свое свободное время в какой-либо игре.

Мы живем в мире, когда развитие науки идет быстрым ходом, и помимо медицины, математики, военного дела и прочих других сфер, требующих точных данных и знаний, так же развиваются технологии. IT или же информационные технологии совершенствуются с каждым днем, появляются новые профессии и специализации.

Одно из прибыльных и манящих на первый взгляд подразделений - разработка игр. И вправду, своей яркостью, необычностью и разносторонностью, компании увлекают покупателей всех возрастов, не оставляя никого безразличным к своему продукту.

В своем проекте я решила примерить на себя роль разработчика игр, и с нуля провести исследование и изучение всех аспектов этой профессии и попробовать создать самостоятельно часть игры.

### **Задачи:**

1. Изучение всевозможных программных обеспечений и выбрать наиболее подходящее и удобное для работы.
2. Создание графики, необходимой для проекта.
3. Изучить новый язык программирования для дальнейшего его применения в разработке продукта.
4. Прорисовать всю необходимую графику для продукта.
5. Создание первоначального шаблона игры.

## **1. Теоретическая часть.**

### **1.1. Изучение этапов создания и разработки игры.**

В современных реалиях игра уже не такой трудоемкий и длительный процесс, какой могут позволить себе только крупные компании. За счет большого спроса на продукт, компании стали делать большое количество программ, библиотек, которые всячески упрощали жизнь разработчикам. Благодаря этому программисты оптимизируют свое время и силы, в результате чего у всех сейчас есть возможность сделать самостоятельно интересную игру, которая понравится многим. Тем не менее, создание игры до сих пор остается одним из трудных заданий, даже со всеми привилегиями на данный момент. В технологии разработки по-прежнему входит уникальность, наличие графического и звукового сопровождения, сюжета и механики. Все эта работа лежит на разработчике и его команде. Сам процесс состоит из нескольких частей. Первая — обдумывание задачи и цели своего продукта, поиск аудитории и так же создание сюжета. Вторая — прорисовка прототипов персонажей и окружающей среды. Третья — собрать все найденное и созданное вместе. Написать код, добавить звуковые эффекты. После чего, игра проходит проверку на ошибки. После этой стадии проект можно считать готовым.

Из этого следует план дальнейшей разработки игры:

- 1) Изучение необходимых для создания материалов.
- 2) Определение цели и замысла проекта.
- 2) Создание графической части проекта.
- 3) Подборка аудио материалов.
- 4) Планирование и распределение работы по времени.
- 5) Создание сюжета и истории игры.

- 6) Изучение языка программирования C#.
- 7) Изучение работы Unity.
- 8) Написание кода.
- 9) Соединение всего в одно целое.
- 10) Исправление ошибок.

## **1.2. Определение Исходного замысла, цели и назначения проекта**

Исходный замысел данного проекта заключается в базовом освоении навыков создания и разработки 2d видео игр, создании базовой модели игры или ее первой (пилотной) части. К окончанию работы над проектом, будет возможность использовать продукт в развлекательных целях. Предусматривается наличие движка, дизайна и начальный сюжет.

## **1.3. Изучение и подбор программ.**

Изучая все возможные программы для благоприятного и удобной разработки можно найти бесконечное множество программ и библиотек. Мой выбор остановился на Unity. Эта программа поддерживает C#, что по отзывам большинства людей является простым языком для изучения, по отношению к другим. Так же Unity удобен в своем пользовательском интерфейсе, что позволяет пользоваться им на интуитивном уровне. Так же этой программе можно найти намного больше уроков и решений проблем, по сравнению с другими.

Выбор программы для графического обеспечения не составил труда. Критерии подбора были просты: наличие анимационной панели, и возможно рисовать в Pixel стиле.

По этим параметрам, мною была выбрана Krita. Она бесплатная и поддерживает все необходимые функции для создания анимации и графики для проекта.

Так же был необходим текстовый редактор для кода. Так как это не первый раз мой в программировании, я выбрала Visual Studio Code. Он удобны, быстрый, бесплатный и привычный для меня редактор.

### **1.3. Продумывание сюжета.**

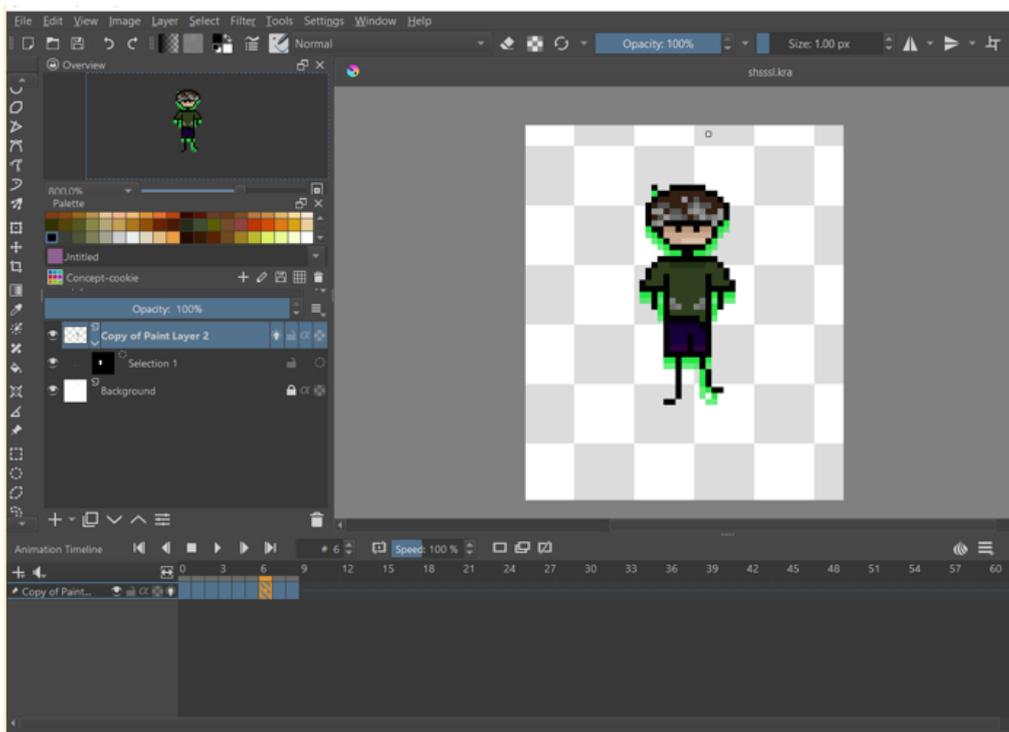
Так как в планах дальше данный продукт развить до полноценной игры необходимо определиться с сюжетом и задачей этой игры. Идеей в дальнейшем применить эту игру для интерактивного изучения физики для 7-9 классов.

Данная игра будет представляет из себя инди квестовую игру, с различными загадками и препятствиями. Для простоты стиль и рисовка была определена в пиксельном стиле. На данном этапе так же был прописан изначальный сюжет, который в продолжении разработки проекта видоизменялся.

## **2. Практическая часть**

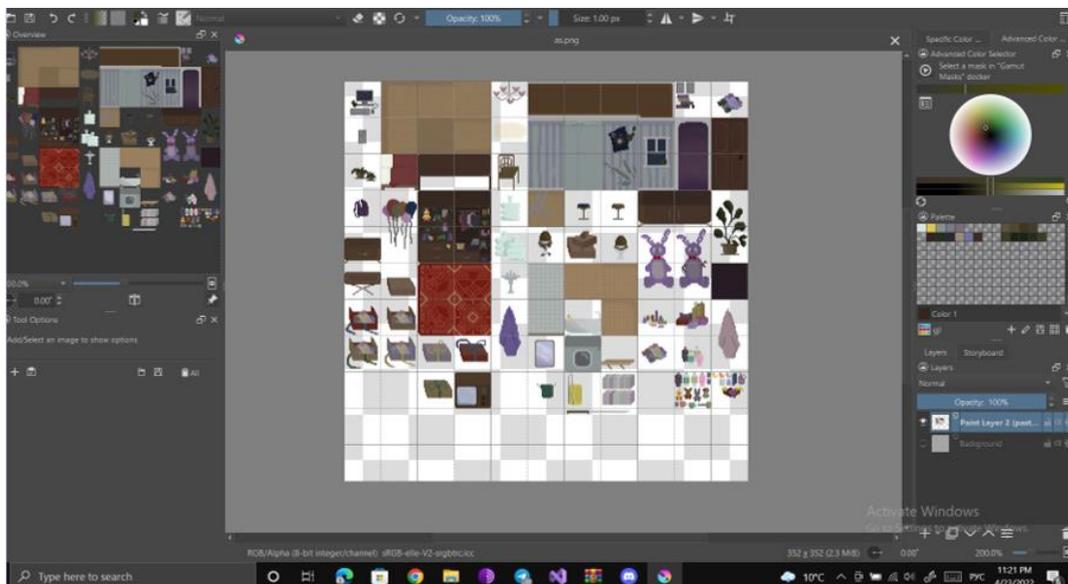
### **2.1. Создание графической части проекта.**

Изначальной задачей было собрать необходимую графическую базу, за счет которой можно собирать игру как конструктор. Для этого нужно было: создать прототип персонажа, прорисовать некоторые локации, за счет которых можно развивать сюжет и сделать игру более интересной и неповторимой. Для живой и реалистичной игры помимо прототипа была создана анимация для этого персонажа. Таким образом, было создано 5 различных видов анимации передвижения персонажа, а именно: походка вверх, вправо, вниз, влево, прыжок.

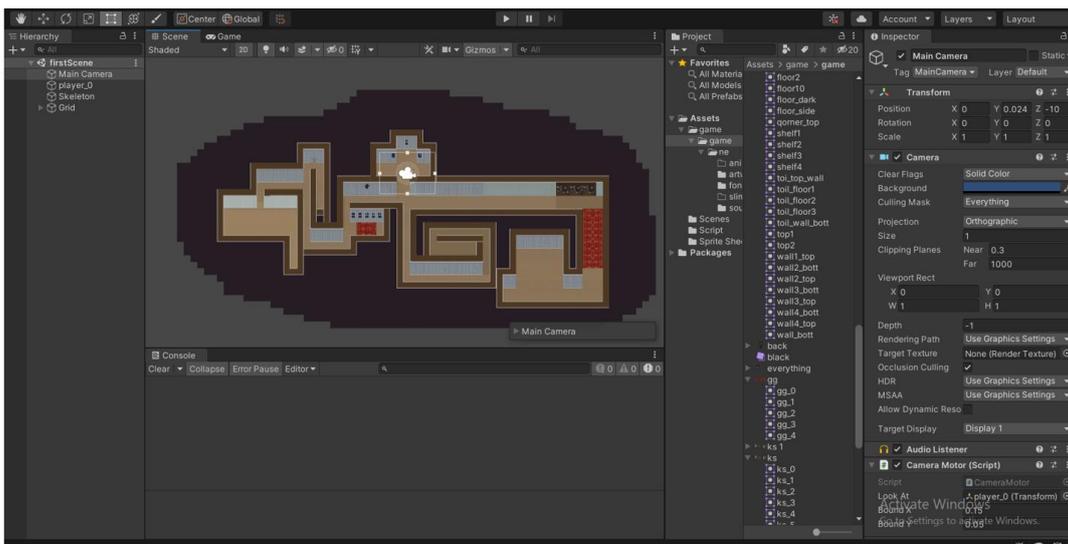


Далее, была проделана работа над созданием окружающей среды. Для того, чтобы создание проекта было максимально эффективным шаблоны окружающей среды должны быть отформатированы по определенным правилам. Масштаб каждой детали составлял 32x32 пикселей. Прорисовывая отдельные детали в данном масштабе, они не должны повторяться, так как далее их предстоит склеивать между собой как маленькие кусочки картины. Данный способ

помогает не только сэкономить время на прорисовке, но также он уменьшает общую скорость самой игры в дальнейшем. Так как если один кусок будет повторяться несколько раз, будет лучше для памяти компьютера прорисовывать одни и те же маленькие кусочки, чем один большой.



После чего, уже в программе Unity, необходимо разделить готовые материалы на отдельные части и с группировать, по уровням и локациям, тем самым создавать карты. После чего отформатированным материалом можно прорисовывать карту игрового поля.



## 2.2 Написание программы.

Для данной игры было необходимо написать программы на ранее выбранном языке программирования C#.

### 1. Логику перемещения камеры:

Камера должна следовать за персонажем во время передвижения. Для чего необходимо обратиться к библиотекам самой программы и написать небольшой код, который будет отвечать за передвижение главного героя и передвижение камеры за ним. Благодаря библиотекам упрощается написание кода, так как уже по умолчанию установлены клавиши для перемещения (W- вверх, D - вправо, A- влево, S - вниз). Импортируя библиотеки, мы так же ссылаемся на координатное положение персонажа. Изначальное положение принимается за (0, 0) при нажатии вышеперечисленных кнопок изменяется координатное положение самого персонажа на экране (W, S – на оси OY и A, D – на ось OX). В функции `fixedUpdate()` как раз таки проходит ежесекундная проверка на подачу данных с клавиатуры. После получения данных персонаж двигается на 1 пиксель. Следом за ним двигается с задержкой камера. Задержка нужна для того, чтобы играющий успевал воспринимать информацию без напряжения. Также в этот этап входит программирование анимации при движении персонажа.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5
6 public class player : MonoBehaviour
7 {
8     private BoxCollider2D boxCollider;
9     private Vector3 moveDelta;
10    private RaycastHit2D hit;
11
12    Animator anim;
13
14    private void Start()
15    {
16        boxCollider = GetComponent<BoxCollider2D>();
17        anim = GetComponent<Animator>();
18    }
19
20    private void FixedUpdate()
21    {
22        float x = Input.GetAxisRaw("Horizontal");
23        float y = Input.GetAxisRaw("Vertical");
24
25        // Move moveDelta
26        moveDelta = new Vector2(x, y);
27
28        // Don't update direction, whether you're going right or left
29        if (moveDelta.x > 0)
30        {
31            transform.localScale = new Vector3(1, 1, 1);
32            anim.SetBool("isWalking", true);
33        }
34        else if (moveDelta.x < 0)
35        {
36            transform.localScale = new Vector3(-1, 1, 1);
37            anim.SetBool("isWalking", true);
38        }
39
40        // Make sure we are open in this direction, by getting a box there first. If the box returns null, we're free to move
41        hit = Physics2D.BoxCast(transform.position, boxCollider.size, 0, new Vector2(0, moveDelta.y), Mathf.Abs(moveDelta.x * Time.deltaTime), LayerMask.GetMask("Actor", "Blocking"));
42        if (hit.collider == null)
43        {
44            // Make this thing move
45            transform.Translate(0, moveDelta.y * Time.deltaTime, 0);
46        }
47
48        hit = Physics2D.BoxCast(transform.position, boxCollider.size, 0, new Vector2(moveDelta.x, 0), Mathf.Abs(moveDelta.y * Time.deltaTime), LayerMask.GetMask("Actor", "Blocking"));
49        if (hit.collider == null)
50        {
51            // Make this thing move
52            transform.Translate(moveDelta.x * Time.deltaTime, 0, 0);
53        }
54    }
55 }
```

## 2. Логика окружающей среды:

Окружающие объекты: шкаф, стол и стены должны иметь как границы, так и физику. И для того, чтобы обеспечить правильную работу команд, необходимо прописать код, который предусматривает границы объектов, за которые нельзя переступить игроку. Для этого так же необходимо воспользоваться библиотеками Unity, а также сначала в программе распределить границы данных объектов.

### Краткое описание полученных результатов

Поставленная цель была осуществлена. В течение работы над проектом были освоены все необходимые навыки, также были приобретены навыки программирования. Продукт, был сделан как пилотный проект игры, который функционирует по задумке.



Имя	Дата изменения	Тип	Размер
MonoBleedingEdge	23.10.2022 20:42	Папка с файлами	
My project_Data	23.10.2022 20:43	Папка с файлами	
My project.exe	13.10.2021 12:51	Приложение	639 КБ
UnityCrashHandler64.exe	13.10.2021 12:52	Приложение	1 204 КБ
UnityPlayer.dll	13.10.2021 12:52	Расширение при...	27 490 КБ
важно.txt	26.04.2022 0:29	Текстовый докум...	1 КБ

## Список используемой литературы

1. <https://docs.microsoft.com/en-us/dotnet/csharp/>
2. <https://docs.unity3d.com/Manual/UnityasaLibrary.html>
3. <https://stackoverflow.com/questions/42597501/unity>