

РАЗРАБОТКА МОБИЛЬНОЙ ИГРЫ ЖАНРА ПЛАТФОРМЕР-ГОЛОВОЛОМКА

ПЛАТФОРМЕР-ГОЛОВОЛОМКА



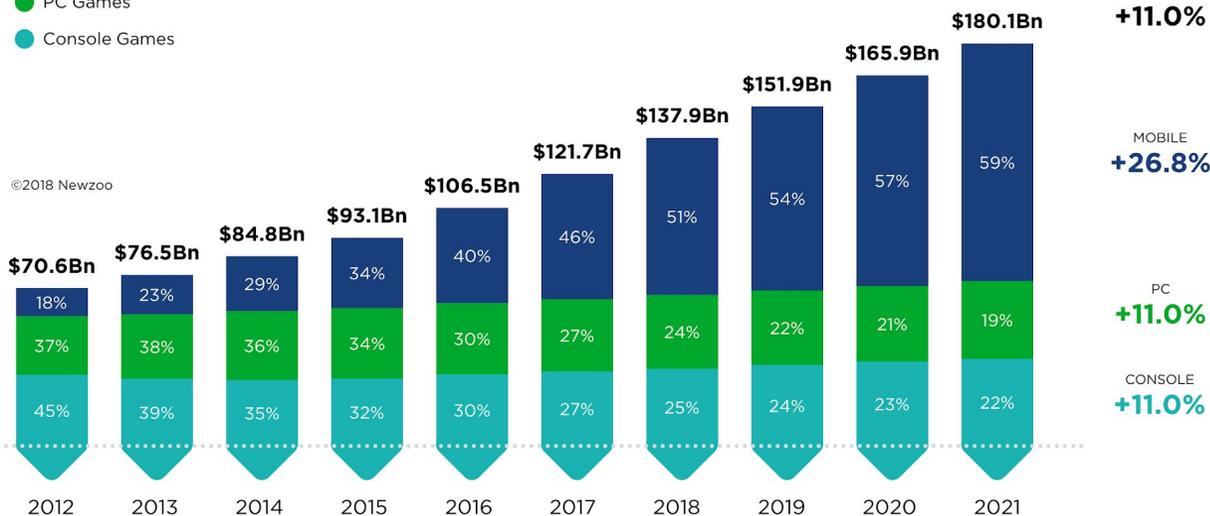
Выполнил студент группы
ИВТ-16(ускр)



2012-2021 GLOBAL GAMES MARKET

REVENUES PER SEGMENT 2012-2021 WITH COMPOUND ANNUAL GROWTH RATES

- Mobile Games
- PC Games
- Console Games



Source: ©Newzoo | April 2018 Quarterly Update | Global Games Market Report
newzoo.com/globalgamesreport



По данному графику и прогнозу на 2021 год
влиятельной аналитической
организацией Newzoo,
видно, что мобильный
рынок будет неуклонно расти.

Рисунок 1. – Динамика роста прибыли игровой индустрии от Newzoo

Целью выпускной квалификационной работы является разработка мобильной игры жанра платформер-головоломка.

Задачи:

- ❖ Изучение соответствующей литературы;
- ❖ Создание блок-схемы игры;
- ❖ Написание кода программы;
- ❖ Апробация тестового варианты игры.

Сравнение игр аналогов



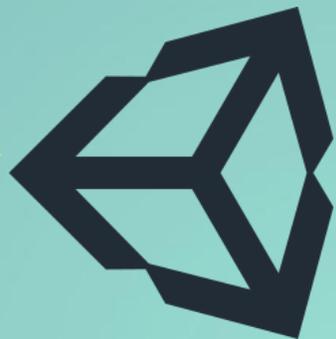
Название: Limbo
Разработчик: Playdead
Издатель: Microsoft Game Studios,
Playdead (PSN, Windows, Mac OS X, Linux, iOS, Android)
Дата выпуска: 21 июля 2010.
На мобильные платформы: 3 июля 2014.
Жанр: Платформер, головоломка, survival horror



Название: Brothers: A Tale of Two Sons
Разработчик: Starbreeze Studios
Издатель: 505 Games
Даты выпуска:
Xbox 360 – 7 августа 2013, Microsoft Windows – 3 сентября 2013,
PlayStation 3 – 3 сентября 2013, PlayStation 4, Xbox One – Лето 2015
iOS, Android – 22 октября 2015
Жанр: Головоломка, платформер

Сравнение игровых ДВИЖКОВ

Выбранный движок →



unity

- + Простой в изучении
- + Удобный интерфейс
- + Множество учебных материалов
- + Язык скриптов C#

Разработка 2D и 3D проектов

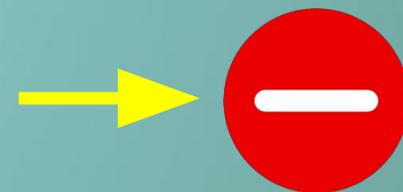
Конкуренты →



UNREAL ENGINE



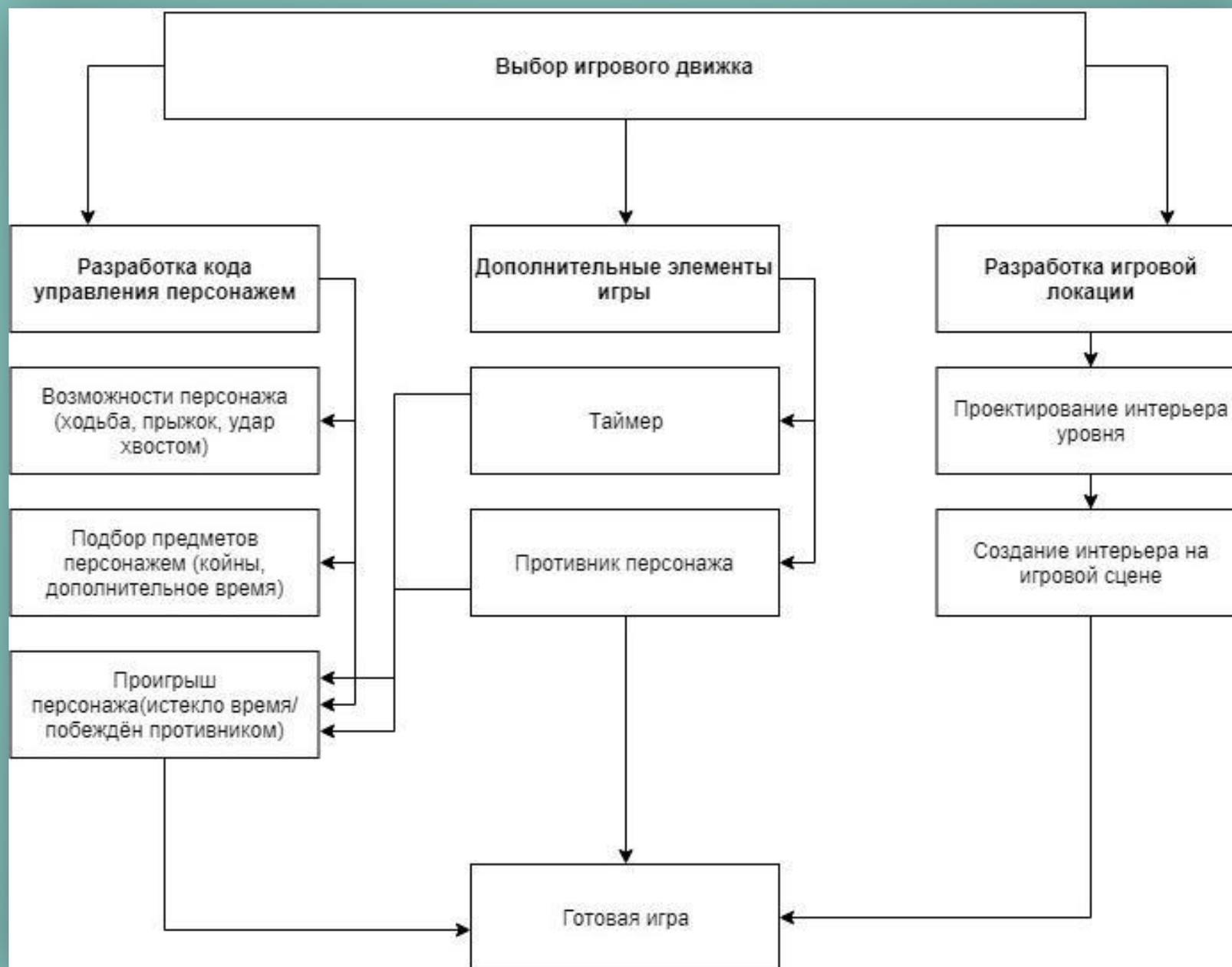
CRYENGINE 3



Этапы разработки игры

ПРОЕКТИРОВАНИЕ	
1. Цель	Идея, жанр, сеттинг.
2. Средство	Программный код, игровой движок
ТВОРЧЕСТВО	
3. Игровая механика	Объекты, управление, физический движок
4. Уровни	Level Design (расстановка объектов)
5. Графика	Арты, 3D модели, анимации, спрайты, фоны
6. Сюжет	События, диалоги, повествования
7. Звук	Звуковые эффекты, музыка, озвучка
ИЗДАНИЕ	
8. Отшлифовка	Устранение ошибок
9. Продажа	Реклама, локализация, система продажи
10. Поддержка	Выпуск дополнений и исправления ошибок

Проектирование Игры



Блок-схема 1 –
проектирование

Разработка игровой локации

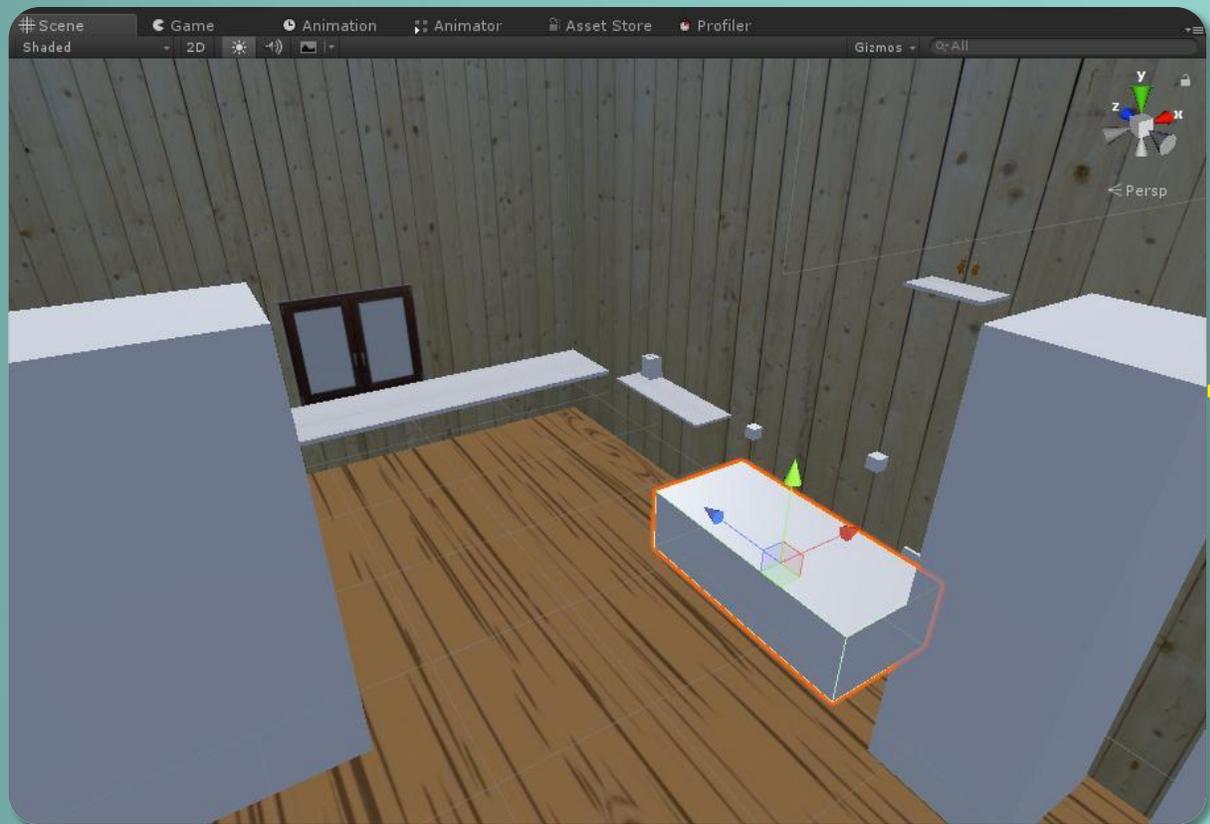


Рисунок 2 – начальный этап разработки уровня

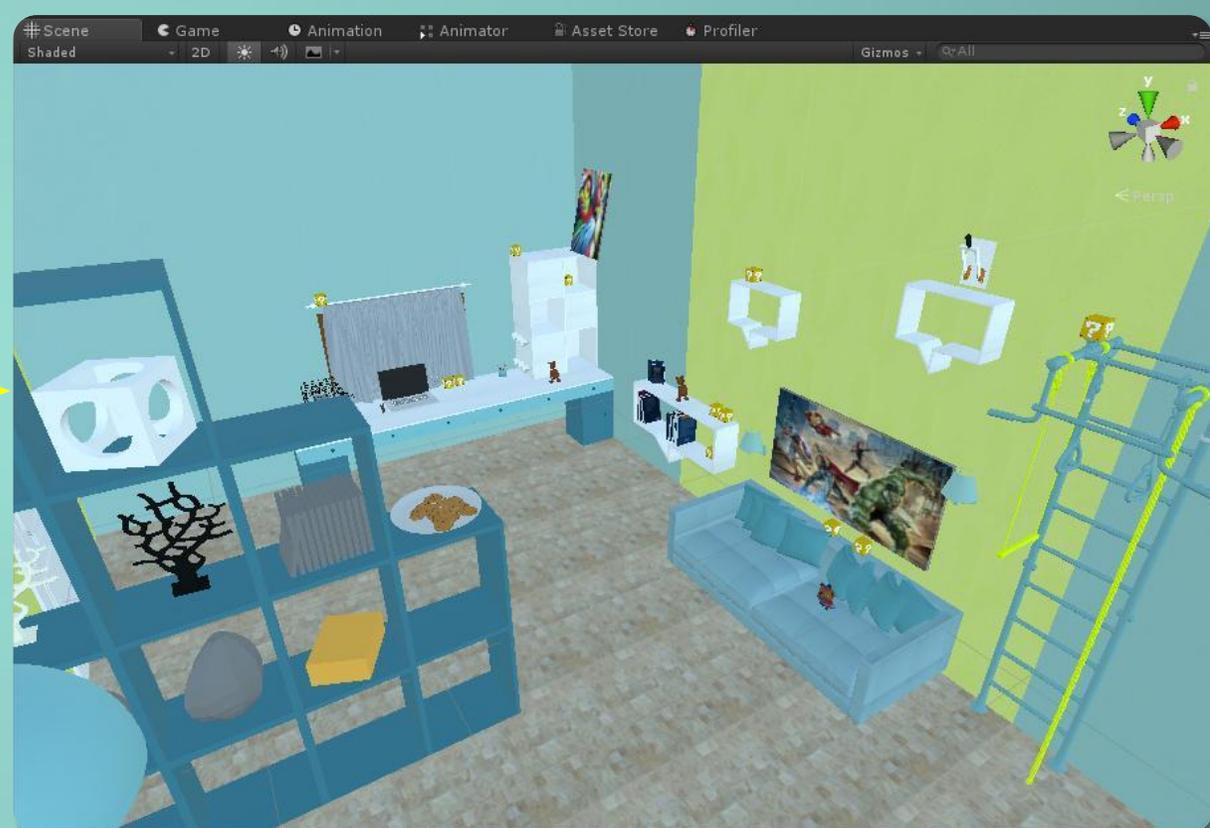
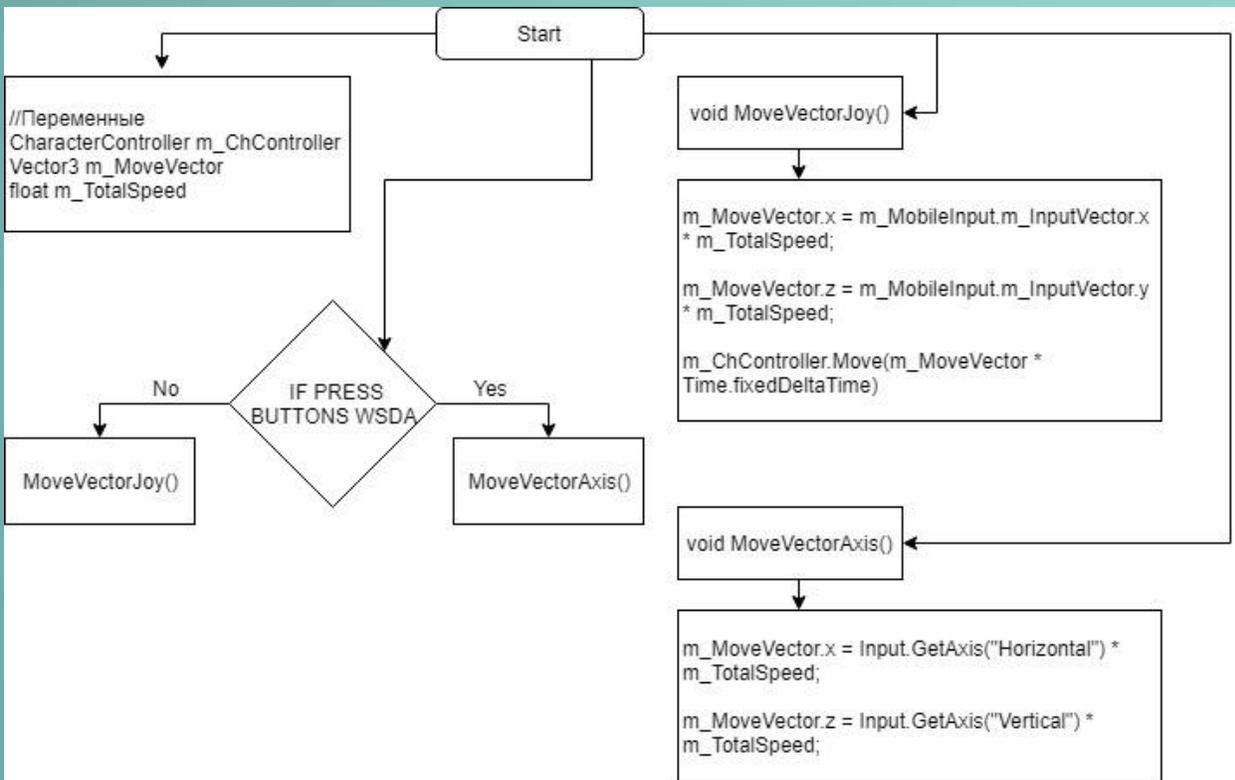


Рисунок 3 – начальный этап разработки уровня

Разработка кода управления персонажем



Блок-схема 2 – методы управления персонажем

```
private void MoveVectors()
{
    m_MoveVector = Vector3.zero;
    if ((Input.GetAxis("Horizontal") != 0 || Input.GetAxis("Vertical") != 0) && m_ModeCanPlayerWalk == ModeCanPlaye
        MoveVectorAxis();
    else
        MoveVectorJoy();
}

1 reference
public void MovePlayer()
{
    PlayerRotation();
    Jump();

    m_ChController.Move(m_MoveVector * Time.fixedDeltaTime);
}

1 reference
private void PlayerRotation()
{
    if (m_ModeCanRotation == ModeCanRotation.canRotation)
    {
        m_camRotation = Quaternion.Euler(0, m_TargetCamera.rotation.eulerAngles.y, 0);
        m_MoveVector = m_camRotation * m_MoveVector;

        if (Vector3.Angle(Vector3.forward, m_MoveVector) > 1f || Vector3.Angle(Vector3.forward, m_MoveVector) == 0f)
        {
            m_direct = Vector3.RotateTowards(transform.forward, m_MoveVector, m_speedRotation * Time.fixedDeltaTime
            m_Transform.rotation = Quaternion.LookRotation(m_direct);
        }
    }
}
```

Рисунок 4 – код методов управления персонажем

Разработка кода управления персонажем



Рисунок 5 – скриншот игры, демонстрация управления персонажа

ScriptableObject (скриптируемый объект)

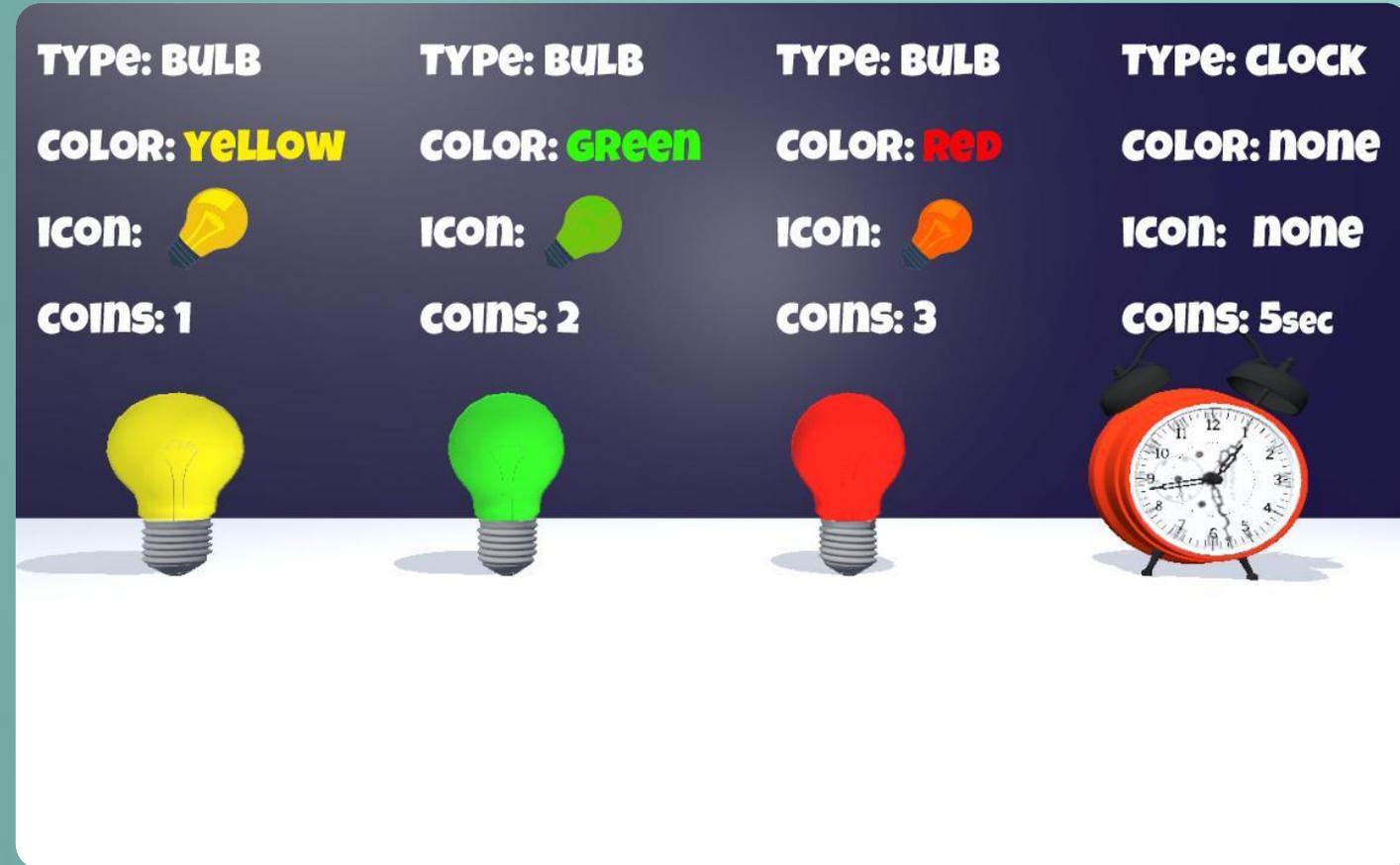


Рисунок 6 – виды объектов собираемых игроком

ScriptableObject это класс, который позволяет хранить большое количество передаваемой информации независимо от образцов скрипта. Существует множество причин для использования ScriptableObjects в Unity. Они могут снизить объём используемой под каждый дополнительный префаб памяти, потому что по своей сути ScriptableObject следуют паттерну разработки Flyweight.

ScriptableObject (скриптируемый объект)

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [CreateAssetMenu(menuName = "CreateBulbsSO")]
6 public class BulbsSO : ScriptableObject {
7     [SerializeField] private string ItemName;
8     [SerializeField] private int QuantityPoins;
9     [SerializeField] private Material[] Material;
10    [SerializeField] private GameObject Obj;
11    [SerializeField] private Sprite Icon;
12
13    public string itemName { get { return ItemName; } }
14    public int quantityPoins { get { return QuantityPoins; } }
15    public Material[] material { get { return Material; } }
16    public GameObject obj { get { return Obj; } }
17    public Sprite icon { get { return Icon; } }
18 }
```

Рисунок 7 – код класса ScriptableObject

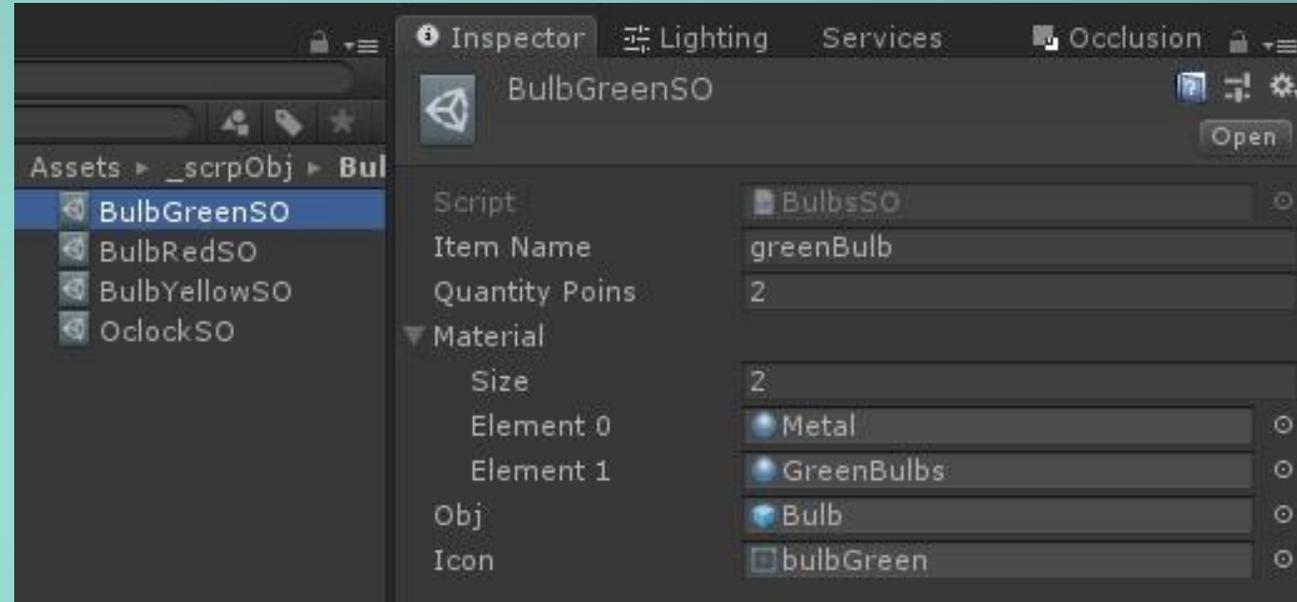
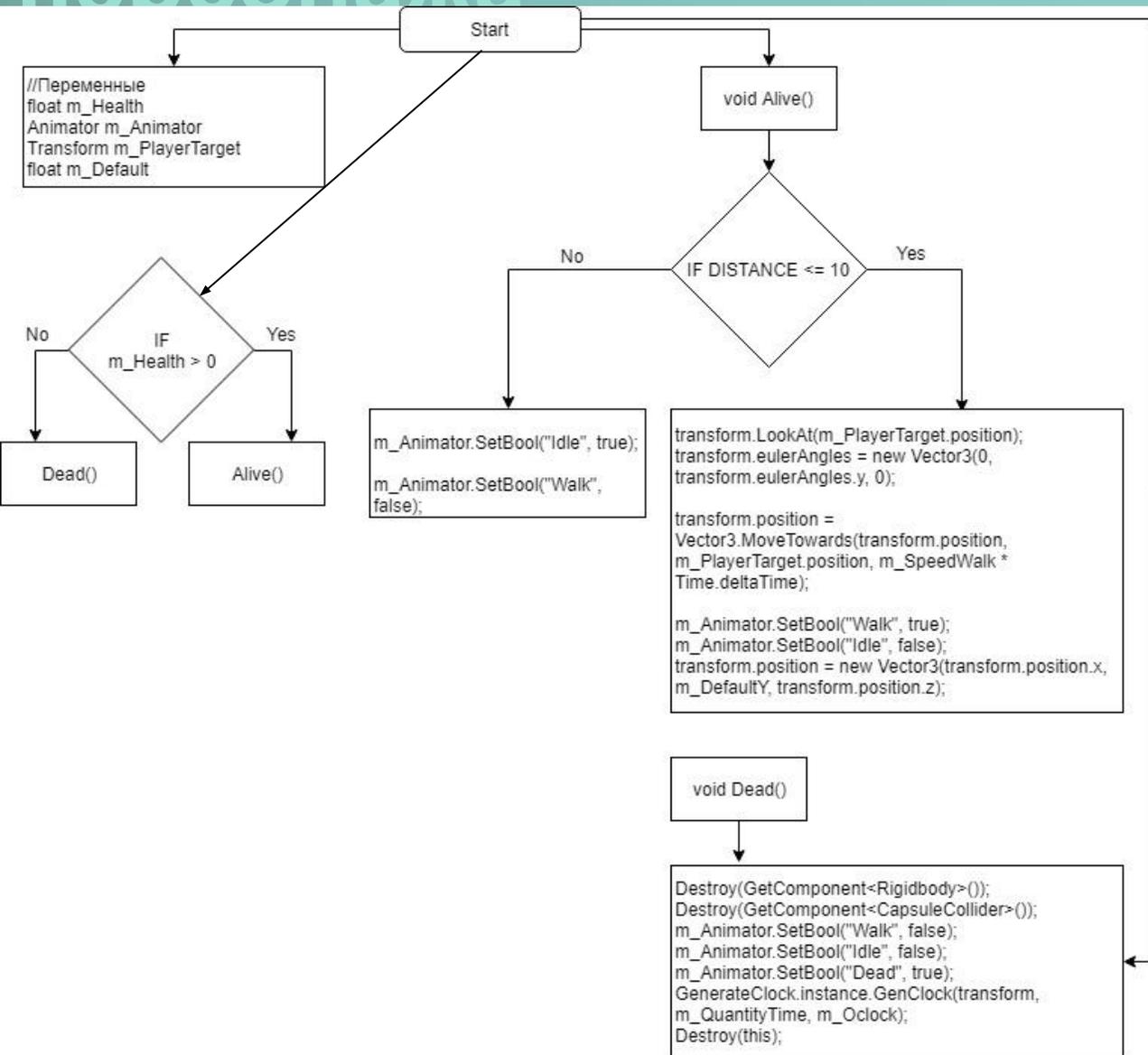


Рисунок 8 – настройка скриптируемых объектов

Разработка противника персонажа



```
private void EnemiesChasingPlayer()
{
    if (m_Health > 0)
        Alive();
    if (m_Health <= 0)
        Dead();
}

1 reference
private void Alive()
{
    if (Vector3.Distance(m_PlayerTarget.position, transform.position) <= 10.0f)
    {
        transform.LookAt(m_PlayerTarget.position);
        transform.eulerAngles = new Vector3(0, transform.eulerAngles.y, 0);
        if (m_PointRay.RayValue() == 1)
        {
            transform.position = Vector3.MoveTowards(transform.position, m_PlayerTarget.position, m_SpeedWalk * Time.deltaTime);
            m_Animator.SetBool("Walk", true);
            m_Animator.SetBool("Idle", false);
        }
        else
        {
            m_Animator.SetBool("Idle", true);
            m_Animator.SetBool("Walk", false);
        }

        transform.position = new Vector3(transform.position.x, m_DefaultY, transform.position.z);
    }
    else
    {
        m_Animator.SetBool("Idle", true);
        m_Animator.SetBool("Walk", false);
    }
}
}
```

Рисунок 9 – код методов противника

Блок-схема 3 – методы управления противником

Заключе ние

В ходе выполнения выпускной квалификационной работы, была разработана полноценная мобильная игра в жанре платформер с элементами головоломок для операционной системы Android с последующим выпуском игры в мобильном рынке Google Play. Данная игра ориентирована для любителей игр в жанре платформеров-головоломок в масштабах всего мира.

Во время разработки были проанализированы современные рынки мобильных приложений, современные игровые движки, сетевые серверы, а также сервисы для сбора аналитических данных. Наиболее подходящими для выполнения поставленной задачи оказались рынок Google Play, игровой движок Unity3D.

Разработанная игра удовлетворяет всем требованиям,

Спасибо за внимание!

