

**V Международная конференция учащихся  
Научно – творческий форум**

**Проектная работа по информатике**

# **Компьютерная игра «СВОИМИ РУКАМИ»**



**Автор: Балакин Егор Сергеевич ученик 9В класса  
МАОУ СОШ №7 города Когалыма, Россия.**

**Руководитель: Мимеев Алексей Владимирович,  
учитель информатики МАОУ СОШ №7 города  
Когалыма, Россия**

**2024 г.**

## Содержание

Введение .....	2
Этапы разработки проекта.....	3
1. Теория разработки игр.....	3
2. Выбор и изучение функций программы реализации проекта .....	4
3. Создание игры .....	5
4. Диагностика работы игры и корректировка параметров.....	9
5. Анкетирование и исследование мнения респондентов.....	9
Итоги работы над проектом .....	12
Список литературы.....	13
Приложения.....	14

## **Введение**

Почти у каждого современного человека на компьютере или телефоне присутствуют - игры. Они позволяют нам отдохнуть, увлекательно провести время, окунуться в мир волшебства и тайн. Есть много различных видов игр, но прежде всего их можно разделить на 2 типа – это полезные игры и приносящие вред.

Когда мы включаем игру, рассчитываем не только на то, чтобы забыться или просто расслабиться, но стоит знать, что есть некоторые игры, которые наносят психический вред человеку. Имеются ввиду игры, которые включают в себя жестокие сцены. Такие игры могут пагубно повлиять на психику человека и сделать его жестоким, особенно если это ребенок. На различных уровнях и интернет-площадках широко обсуждается тема полезности и вреда компьютерных игр. И многие склоняются к мнению, что игры нужны и важны для человека, только надо избирательно подходить к выбору программного продукта и нормировать время их использования, при этом обращая внимание на рекомендации по возрасту.

Желание попробовать создать собственную игру, которая не навредила любому школьнику, появилось ещё в восьмом классе. Именно тогда началось знакомство с теорией игр, поиск доступной среды для работы. Придумать сюжет игры, «создать игру своими руками» - эта мечта стала реальностью.

Возможно, моя игра поможет вытеснить «вредные» игры, т.к. не будет содержать жестоких сцен и в скором времени привлечь большую аудиторию людей, которые позабудут о развлечениях, содержащих насилие.

## **Цель проектной работы**

Целью проектной работы является создание компьютерной игры «SpaceJump» в среде Unity 3D.

## **Задачи**

Для реализации цели определяются следующие задачи:

1. познакомиться с теорией и правилами создания игр;
2. выбрать и изучить среду программирования;
3. придумать сюжет и правила игры;
4. выполнить разработку игры в среде;
5. провести отладку и редактирование параметров игровых ситуаций;
6. представление проекта сверстникам.

## **Этапы разработки проекта**

1. Изучение теории разработки игр
2. Выбор и изучение функций программы реализации проекта
3. Создание игры
4. Диагностика работы игры и корректировка параметров
5. Анкетирование и исследование мнения респондентов

### **1. Теория разработки игр**

Для того, чтобы понять, как устроены игры в целом, я начал смотреть видео, размещенные в интернете, которые давали совет юным разработчикам игр. Благодаря сайту YouTube, я смог найти много обзоров на игровые движки, советы для создания игр и вообще с чего всё нужно начинать. [1,2,3]

Первое и главное решение, которое необходимо принять любому создателю игры – это выбор тематики игры. Обычно компьютерные игры делятся на несколько типов: квесты, экшн, ролевые игры, стратегии, симуляторы, логические и азартные. [4]

- Квесты – осуществляют путешествие одного или нескольких персонажей к поставленной цели путем преодоления разнообразных трудностей.

- Экшн (action) - игры от первого лица – популярные бродилки-стрелялки.

- Рольевые игры или РПГ – игрок исполняет роль определенного персонажа и выполняет поставленные перед ним задачи.
- Стратегии и логические игры подражают деятельности управленца.
- Симуляторы имитируют управление автомобилем, космическим кораблем, самолетом и т.п.
- Азартные и логические хорошо развивают мыслительную деятельность.

Используя рекомендации экспертов был определен вид игры и отдано предпочтение «квесту».

Также компьютерные игры можно сгруппировать по следующим признакам: есть или нет персонаж. Игры без персонажа – логические, симуляторы. Они не имеют сюжета, не вызывают сильного привыкания и не влияют на психику. Часто предлагают интеллектуальные задания, что хорошо развивает мышление.

Таким образом, будущая игра – это квест без персонажа с элементами симулятора.

## **2. Выбор и изучение функций программы реализации проекта**

Знакомясь со средами создания игр я обратил внимание на технологию Unity 5. Компания Unity Technologies официально объявила о выходе Unity 5 — новой версии популярного мультиплатформенного движка для разработки приложений и видеоигр. Что примечательно, распространяется он на значительно более выгодных условиях, чем предшественники. Так, базовую версию технологии под названием Unity 5 Personal Edition можно загрузить с официального сайта и пользоваться ею совершенно бесплатно, а отчислений за полученную прибыль с вас не потребуют. Данный вариант предназначен для простых пользователей и компаний, у которых годовая выручка и объём инвестиций не превышают \$100 тыс. В Personal Edition доступны все возможности движка и редактора Unity 5, а ограничения касаются лишь дополнительных функций вроде сервисов Unity Cloud Build Pro, Game Performance Reporting, Unity Analytics Pro и прочих.

Unity 5 отличается улучшенной графикой и расширенными функциями редактора. Так, в пятой версии движка стали доступны новые шейдеры и интеграция с технологией глобального освещения в режиме реального времени Geomerics Enlighten. Это позволяет привносить в игры визуальные эффекты и освещение консольного уровня, а также математически точное отображение теней. Также Unity 5 может похвастаться новыми инструментами WebGL, которые позволяют создавать в браузерах более продвинутые проекты с высокой производительностью. [5]

Выбор игрового движка стал очевиден – это игровой движок Unity 5. Этот движок является более известным и многие наши современники начинают создавать игры именно с его помощью, так как он более прост в некоторых деталях

### **3. Создание игры**

1) Для того чтобы создать игру, нужно придумать идею для игры и логическую цепочку действий, по которой игрок в итоге должен закончить игру. Таким образом, сюжет заключался в том, что нужно преодолевать препятствия, перепрыгивать через кубы и сделать всё возможное, чтобы дойти до контрольной точки уровня.

2) После того, как был придуман сюжет, мне нужно было создать обстановку, в которой будет происходить геймплей игры. Было решено создать обстановку в космосе и использован Skybox картинка космоса (рис 1). Естественно, чтобы игроку не надоедали однообразные уровни, добавлены разные Skybox на каждом уровне.

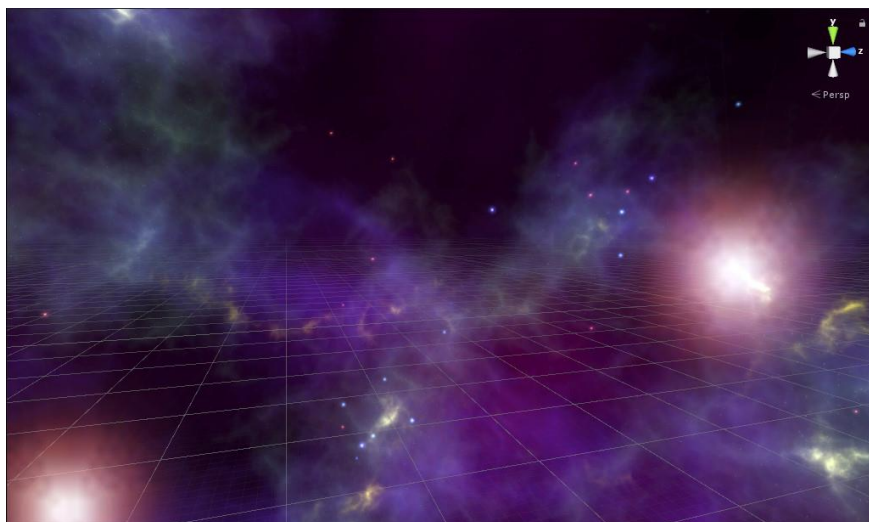


Рис 1

3) Затем началось создание мест, по которым игрок будет передвигаться.

Объектом, по которому будет передвигаться персонаж стал куб (рис 2). Просто расставить кубы было недостаточно, и вскоре были добавлены разные формы поверхностей и анимации на некоторые объекты. Таким образом повышалась сложность игры, и на каждом уровне по сюжету должны появляться новые объекты с их своеобразными свойствами.



Рис 2

4) Создание поверхности – это только часть работы по созданию объекта. Чтобы игрок мог закончить уровень, предстояло создать конечный пункт, дойдя

до которого, он переходит на следующий этап (рис 3). Для осуществления перехода на новый уровень, создан скрипт, по которому будет срабатывать сценарий игры (Приложение 2).

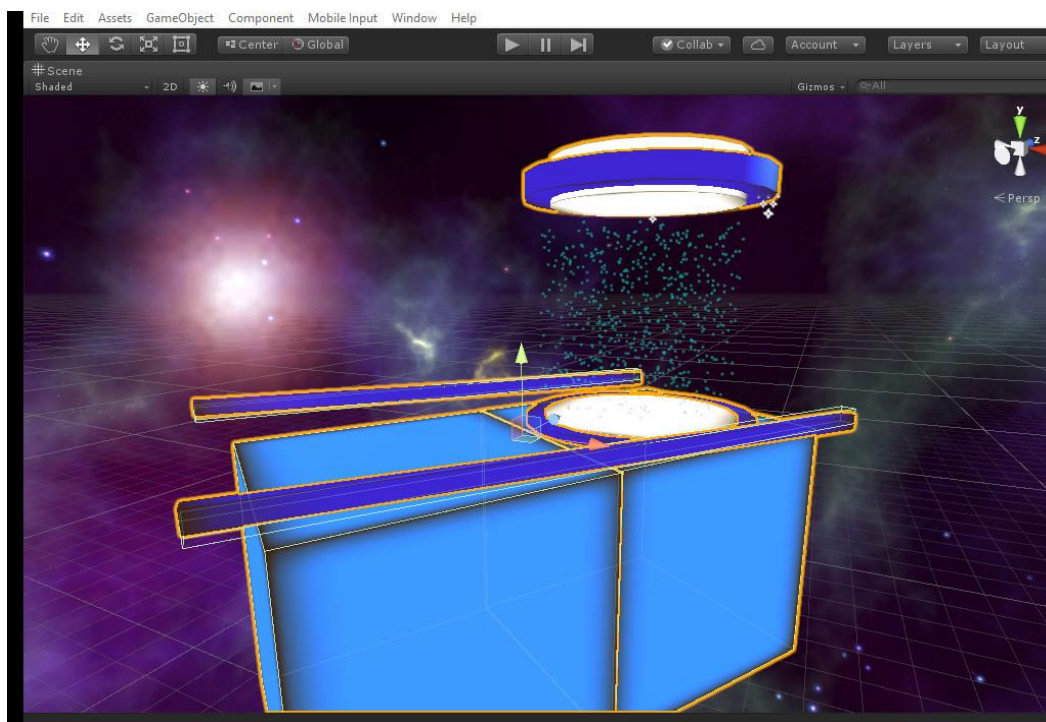


Рис 3

5) Меню – незаменимая часть, с которой начинают игру, настраивают системные требования и вообще лицо игры (рис 4). Пришлось тщательно потрудиться над его созданием, чтобы оно было максимально приближено к нашим современным стандартам. Главное меню игры включает в себя:

-Начать игру;

-Настройки

- ◆ Громкость музыки;
- ◆ Разрешение экрана (от 1024x768 до 2560x1440);
- ◆ Качество графического изображения (Низкое, Среднее, Высокое);
- ◆ Язык (Русский или English);

-Автор (Можно посмотреть информацию об авторе игры);

-Выход (Перед выходом, запрашивается предварительное соглашение о выходе).

(Скрипт в Приложении 3)



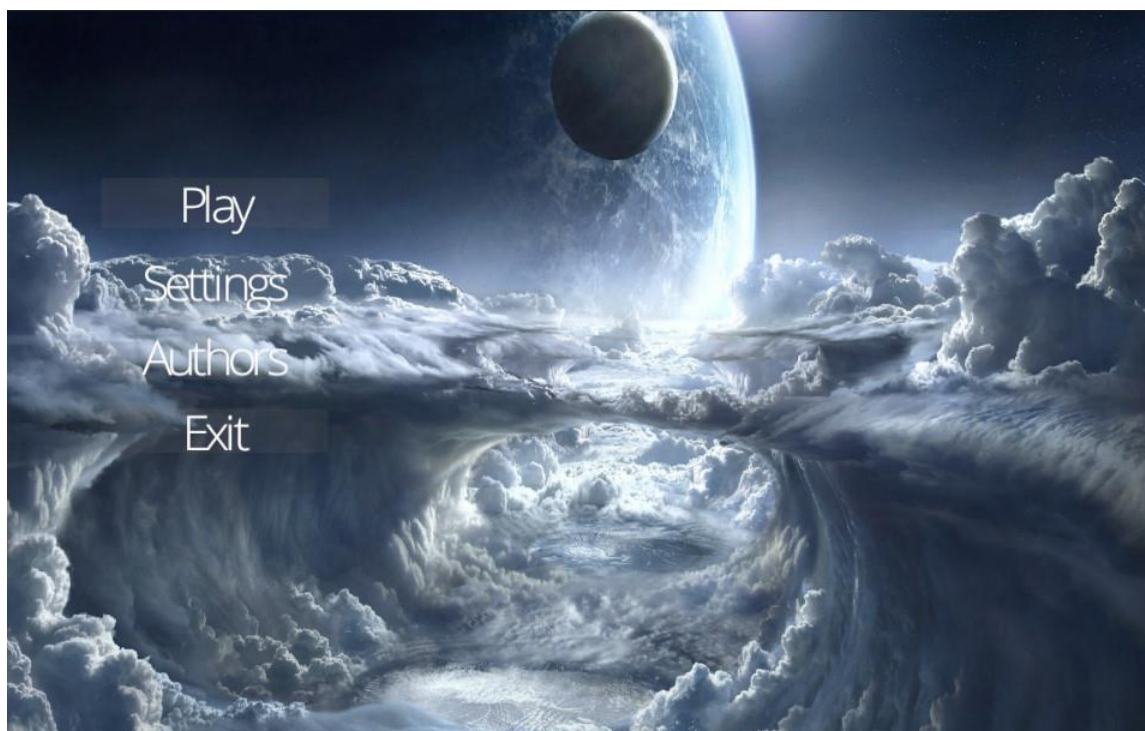


Рис 4

б) Не менее важной частью игрового проекта является меню паузы, что бы геймер мог в нужный момент приостановить игру или же вовсе выйти в меню. Меню паузы включает в себя несколько разделов таких как: продолжить игру, настройки игры и выход в главное меню (рис 5). Структура этого скрипта не менее сложна, чем структура скрипта главного меню. (Скрипт в Приложении 1)

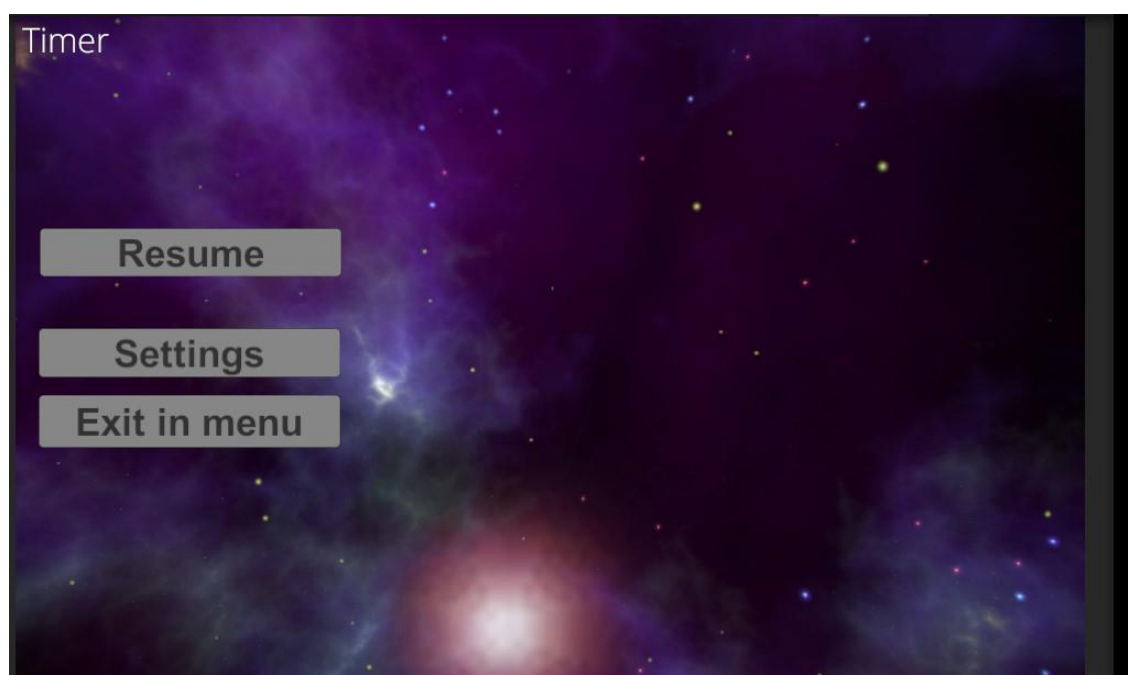


Рис 5

#### 4. Диагностика работы игры и корректировка параметров

На последнем этапе разработки игры она была скомпилирована в первый готовый экземпляр рабочего приложения, который был предложен для тестирования двенадцати респондентам из числа родителей, одноклассников и друзей. В результате опроса респондентов по их замечаниям и предложениям были изменены некоторые параметры игры, а именно добавлена функция замедления движения объектов, добавлен таймер времени прохождения уровня.

Положительным отзывом об игре можно считать удовлетворительную сложность игры, что позволяет заинтересовать игрока на первом этапе и стремиться достичь более высоких результатов.

На этапе доработки были улучшены текстуры для визуальных объектов и добавлены окружающие элементы (рис 6).

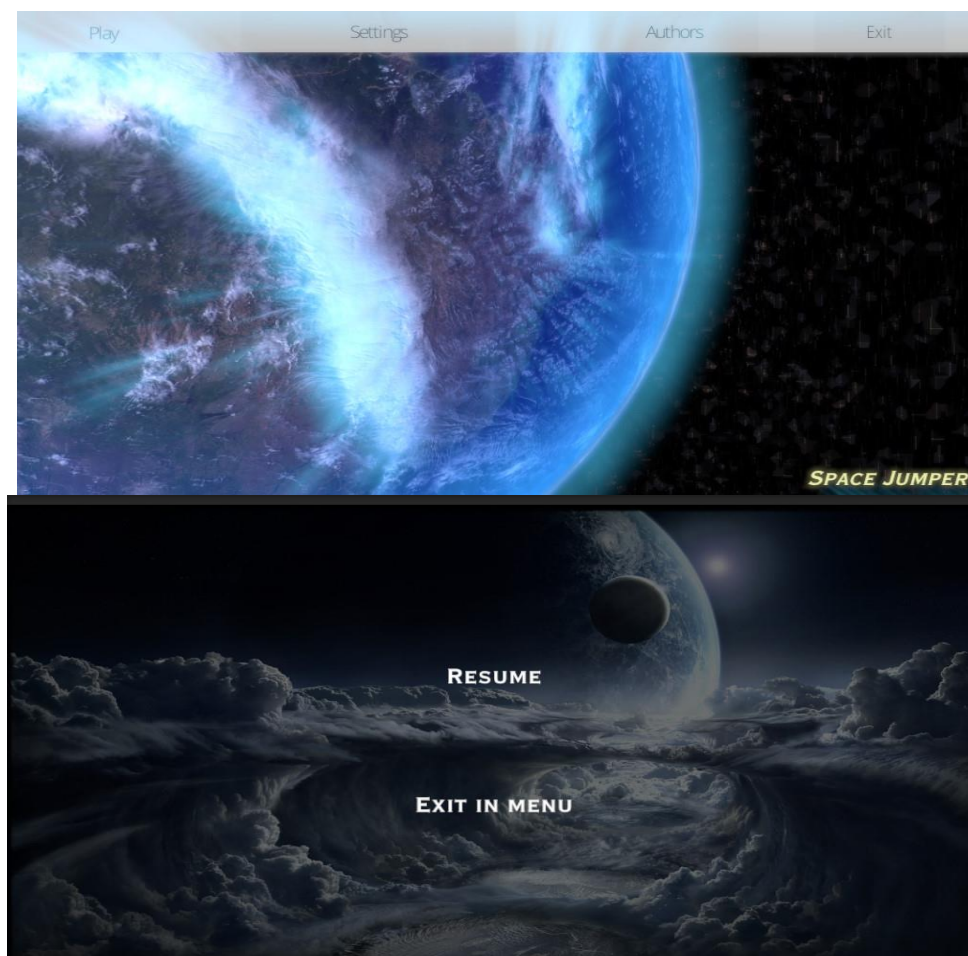


Рис 6

## 5. Анкетирование и исследование мнения респондентов

После создания игры мы предложили протестировать ее двенадцати респондентам из числа родителей, одноклассников и друзей. Для этого была предложена следующая анкета для исследования отношения к игре и ее дальнейшего улучшения. Вопросы анкеты таковы:

1. Понравилась ли вам игра?
2. Нравится ли вам установленный уровень сложности игры?
3. Как вы считаете, стоит ли добавить новые уровни игры?
4. Нравится ли вам интерфейс игры?
5. Вам нравится качество графического изображения?
6. Есть ли недостатки в игре, если да, то какие?
7. Что бы вы хотели изменить в этой игре?
8. Хотите, чтобы вышла вторая часть?

Анализ ответов на вопросы представлен в таблице и на диаграмме.

Таблица 1

Название вопроса	Да	Нет	Пояснение
Понравилась ли вам игра?	9	3	
Нравится ли вам установленный уровень сложности игры?	6	6	
Как вы считаете, стоит ли добавить новые уровни игры?	7	5	
Нравится ли вам интерфейс игры?	9	3	
Вам нравится качество графического изображения?	8	4	
Есть ли недостатки в игре, если да, то какие?	5	7	1. Мало уровней 2. Слишком лёгкий уровень сложности.

Хотели бы вы изменить что -либо в этой игре, и если да, то что?	5	7	1. Добавить фигуры при прохождении уровней 2. Изменить музыкальное сопровождение уровней.
Хотите, чтобы вышла вторая часть?	9	3	



Как мы видим из представленной диаграммы отношение к игре у всех респондентов положительное.

## **Итоги работы над проектом**

Работая над созданием проекта, я узнал много нового в сфере программирования и разработки игр. Мною было найдено и просмотрено много обучающих уроков по созданию игр. Все это помогло мне научиться систематизировать полученные знания и применить их на практике.

Проведя анкетирование среди родителей, одноклассников и знакомых, мы выяснили, что нужно добавить и изменить в игре. Поэтому она совершенствуется, а я набираюсь опыта. В дальнейшем хочется, чтобы ее оценили в интернет мирах Steam.

И конечно, общаясь с программистами, у меня появились новые друзья, а родители и знакомые поняли, что я могу создать свой интересный проект. Ведь создавать игру это возможность не только программировать, но и приятно общаться на просторах интернета.

## Список литературы

1. Канал «Safing Mirow»  
[https://www.youtube.com/channel/UCCsK8EOie1QbRgk5Yi5Fv\\_g](https://www.youtube.com/channel/UCCsK8EOie1QbRgk5Yi5Fv_g)
2. Канал «Иван Карманов»  
<https://www.youtube.com/channel/UCcbmaBZEEzGnkFIMj7C1xWw>
3. Канал «Игродел Мастер» <https://www.youtube.com/user/igrofresh>
4. Классификация компьютерных игр — Википедия  
[https://ru.wikipedia.org/wiki/Классификация\\_компьютерных\\_игр](https://ru.wikipedia.org/wiki/Классификация_компьютерных_игр)
5. Новости Software. <https://3dnews.ru/910408>

## Пауза:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Pause : MonoBehaviour {
    //если ты будешь делать субтитры тогда нужно русские субтитры добавлять в кавас на
    русском(CanvasRus) и английские в канвас на англ.(CanvasEng)
    //если что куда что вешать потом могу рассказать в скайпе
    public GameObject CanvasRus;//Это канвас на русском
    public GameObject CanvasEng;//Это канвас на английском
    public GameObject SettingsEng;//Это панель настроек на английском
    public Slider SoundSlidEng;// Это слайдер звука на английском
    public Dropdown ResolutionDropEng;// Это DropDown разрешения на английском
    public Dropdown QualityDropEng;// Это DropDown качества на английском
    public GameObject SettingsRus;//Это панель настроек на русском
    public Slider SoundSlidRus;// Это слайдер звука на русском
    public Dropdown ResolutionDropRus;// Это DropDown разрешения на русском
    public Dropdown QualityDropRus;// Это DropDown качества на русском

    void Start(){
        if (Global.Language == 0) {
            SoundSlidEng.GetComponent<Slider> ().value = Global.Sound;
            ResolutionDropEng.GetComponent<Dropdown> ().value =
Global.Resolution;
            QualityDropEng.GetComponent<Dropdown> ().value = Global.Quality;
        }
        if (Global.Language == 1) {
            SoundSlidRus.GetComponent<Slider> ().value = Global.Sound;
            ResolutionDropRus.GetComponent<Dropdown> ().value =
Global.Resolution;
            QualityDropRus.GetComponent<Dropdown> ().value = Global.Quality;
        }
        gameObject.GetComponent<AudioSource> ().volume = Global.Sound;
        Screen.lockCursor = true;
    }
    void Update(){
        if (Input.GetButtonDown ("Cancel")) {
            if (Global.Language == 0) {
                CanvasEng.SetActive (true);
            }
            if (Global.Language == 1) {
                CanvasRus.SetActive (true);
            }
            gameObject.GetComponent<AudioSource> ().volume = 0.04f;
            Screen.lockCursor = false;
            Time.timeScale = 0;
        }
    }
}

```

```

public void Resume()
{
    if (Global.Language == 0) {
        CanvasEng.SetActive (false);
    }
    if (Global.Language == 1) {
        CanvasRus.SetActive (false);
    }
    gameObject.GetComponent<AudioSource> ().volume = Global.Sound;
    Screen.lockCursor = true;
    Time.timeScale = 1;
}
public void Restart ()
{
    Application.LoadLevel (0);
}
public void SettingsGame()//Функция настроек
{
    SettingsEng.SetActive (!SettingsEng.activeSelf);
    SoundSlidEng.GetComponent<Slider> ().value = Global.Sound;
    ResolutionDropEng.GetComponent<Dropdown> ().value = Global.Resolution;
    QualityDropEng.GetComponent<Dropdown> ().value = Global.Quality;
    SettingsRus.SetActive (!SettingsRus.activeSelf);
    SoundSlidRus.GetComponent<Slider> ().value = Global.Sound;
    ResolutionDropRus.GetComponent<Dropdown> ().value = Global.Resolution;
    QualityDropRus.GetComponent<Dropdown> ().value = Global.Quality;
}
public void SetSound(float value){
    if (Global.Language == 1) {
        Global.Sound = SoundSlidRus.GetComponent<Slider> ().value;
    } else {
        Global.Sound = SoundSlidEng.GetComponent<Slider> ().value;
    }
}
public void SetResolution(int value){
    if (Global.Language == 1)
        Global.Resolution = ResolutionDropRus.GetComponent<Dropdown>
().value;
    else {
        Global.Resolution = ResolutionDropEng.GetComponent<Dropdown>
().value;
    }
    switch (Global.Resolution) {
    case 0:
        Screen.SetResolution (2560, 1440, true);
        break;
    case 1:
        Screen.SetResolution (1920, 1080, true);
        break;
    case 2:
        Screen.SetResolution (1600, 900, true);
}

```



```
        break;
    case 3:
        Screen.SetResolution (1500, 1024, true);
        break;
    case 4:
        Screen.SetResolution (1440, 900, true);
        break;
    case 5:
        Screen.SetResolution (1280, 768, true);
        break;
    case 6:
        Screen.SetResolution (1280, 720, true);
        break;
    case 7:
        Screen.SetResolution (1024, 768, true);
        break;
    default:
        Screen.SetResolution (1600, 900, true);
        break;
    }
}
public void SetQuality(int value){
    if (Global.Language == 1)
        Global.Quality = QualityDropRus.GetComponent<Dropdown> ().value;
    else {
        Global.Quality = QualityDropEng.GetComponent<Dropdown> ().value;
    }
    switch (Global.Quality) {
    case 0:
        QualitySettings.SetQualityLevel (5, true);
        break;
    case 1:
        QualitySettings.SetQualityLevel (3, true);
        break;
    case 2:
        QualitySettings.SetQualityLevel (0, true);
        break;
    }
}
public void ExitGame()//Функция выхода
{
    Application.LoadLevel (0);
}
}
```

## Переход на следующий уровень:

```
using UnityEngine;
using System.Collections;

public class NextLevel : MonoBehaviour {

    public int nextlevel = 0;

    void OnTriggerEnter(Collider col)
    {
        if(col.tag == "Player") Application.LoadLevel(nextlevel);
    }
}
```

Начать уровень заново:

```
using UnityEngine;
using System.Collections;

public class ReloadOnClick : MonoBehaviour {

    void Update () {
        if (Input.GetKeyDown (KeyCode.Mouse1))
        {
            Application.LoadLevel (Application.loadedLevel);
        }
    }
}
```

## Меню:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Menu : MonoBehaviour {

    public GameObject settings; // Панель настроек
    public GameObject SoundSlid; //Слайдер громкости
    public GameObject ResolutionDrop; //Дроп разрешения
    public GameObject QualityDrop; //Дроп качества
    public GameObject LanguageDrop; //Дроп языка
    public GameObject Exit; // Панель выхода
    public GameObject CanvasRUS;//Канвас на русском
    public GameObject CanvasENG;//Канвас на английском
    public GameObject authors;//Панель авторы
    public Text textRus;
    public Text textEng;

    void Start(){
        SoundSlid.GetComponent<Slider> ().value = Global.Sound;
        ResolutionDrop.GetComponent<Dropdown> ().value = Global.Resolution;
        QualityDrop.GetComponent<Dropdown> ().value = Global.Quality;
        LanguageDrop.GetComponent<Dropdown> ().value = Global.Language;
        Screen.lockCursor = false;
    }
    public void StartGame()//Функция начала игры
    {
        if(settings.activeSelf == true)
            settings.SetActive (false);
        if(authors.activeSelf == true)
            authors.SetActive (false);
        if (Exit.activeSelf == true)
            Exit.SetActive (false);
        Application.LoadLevel (1);
    }
    public void SettingsGame()//Функция настроек
    {
        settings.SetActive (!settings.activeSelf);
        if(authors.activeSelf == true)
            authors.SetActive (false);
        SoundSlid.GetComponent<Slider> ().value = Global.Sound;
        ResolutionDrop.GetComponent<Dropdown> ().value = Global.Resolution;
        QualityDrop.GetComponent<Dropdown> ().value = Global.Quality;
        LanguageDrop.GetComponent<Dropdown> ().value = Global.Language;
    }
}

```

```
public void Authors() //Функция отображения панель авторы
{
    if (settings.activeSelf == true)
        settings.SetActive (false);
    if (Exit.activeSelf == true)
        Exit.SetActive (false);
    authors.SetActive (!authors.activeSelf);
}
public void ExitGame()//Функция выхода
{
    if(settings.activeSelf == true)
        settings.SetActive (false);
    if(authors.activeSelf == true)
        authors.SetActive (false);
    Exit.SetActive (!Exit.activeSelf);
}
public void NotExit()//Функция отказа от выхода из игры
{
    Exit.SetActive (false);
}
public void YesExit()//Функция подтверждения выхода
{
    Application.Quit ();
}
public void SetSound(float value)//Функция регуляции звука
{
    Global.Sound = SoundSlid.GetComponent<Slider> ().value;
}
public void SetResolution(int value)//Функция выбора разрешения
{
    Global.Resolution = ResolutionDrop.GetComponent<Dropdown> ().value;
    switch (Global.Resolution) {
    case 0:
        Screen.SetResolution (2560, 1440, true);
        break;
    case 1:
        Screen.SetResolution (1920, 1080, true);
        break;
    case 2:
        Screen.SetResolution (1600, 900, true);
        break;
    case 3:
        Screen.SetResolution (1500, 1024, true);
        break;
    case 4:
        Screen.SetResolution (1440, 900, true);
        break;
    case 5:
        Screen.SetResolution (1280, 768, true);
        break;
    case 6:
        Screen.SetResolution (1280, 720, true);
    }
```

```

        break;
    case 7:
        Screen.SetResolution (1024, 768, true);
        break;
    default:
        Screen.SetResolution (1600, 900, true);
        break;
    }
}

public void SetQuality(int value)//Функция выбора качества
{
    Global.Quality = QualityDrop.GetComponent<Dropdown> ().value;
    switch (Global.Quality) {
    case 0:
        QualitySettings.SetQualityLevel (5, true);
        break;
    case 1:
        QualitySettings.SetQualityLevel (3, true);
        break;
    case 2:
        QualitySettings.SetQualityLevel (0, true);
        break;
    }
}

public void SetLanguage(int value)//Функция выбора языка
{
    Global.Language = LanguageDrop.GetComponent<Dropdown> ().value;
    switch (Global.Language) {
    case 1:
        CanvasENG.SetActive (false);
        CanvasRUS.SetActive (true);
        settings.SetActive (false);
        textRus.text = "Русский";
        Global.Sound = SoundSlid.GetComponent<Slider> ().value;
        Global.Quality = QualityDrop.GetComponent<Dropdown> ().value;
        Global.Resolution = ResolutionDrop.GetComponent<Dropdown> ().value;
        break;
    case 0:
        CanvasRUS.SetActive (false);
        CanvasENG.SetActive (true);
        settings.SetActive (false);
        textEng.text = "English";
        break;
    }
}
}
}

```