



Интерфейс. Part I

Часть первая. Основы взаимодействия. Кнопки, текст, картинки.

Что такое интерфейс?



Худ.Фильм - Джони
Мнемоник



Что нам говорит вики?

Интерф́ейс (от [англ. interface](#)) — общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил [взаимодействия](#) (управления, контроля и т. д.) между элементами [системы](#).

Примеры:

- элементы [электронного аппарата](#) ([телевизора](#), [автомагнитолы](#), [часов](#) и т. п.), такие как дисплей, набор кнопок и переключателей для настройки плюс правила управления ими, относятся к [человеко-машинному интерфейсу](#);
- [клавиатура](#), [мышь](#) и пр. [устройства ввода](#) — элементы [интерфейса «человек — компьютер»](#).


Что мы только что прочитали?



Давайте подумаем...







Дополнительная информация

<https://habr.com/ru/company/mosigra/blog/232197/>

Рекомендую прочитать, ссылка так же в телеграмме

habr



Когда сам интерфейс игры становится
частью сюжета и мира

Перейдём к Unity





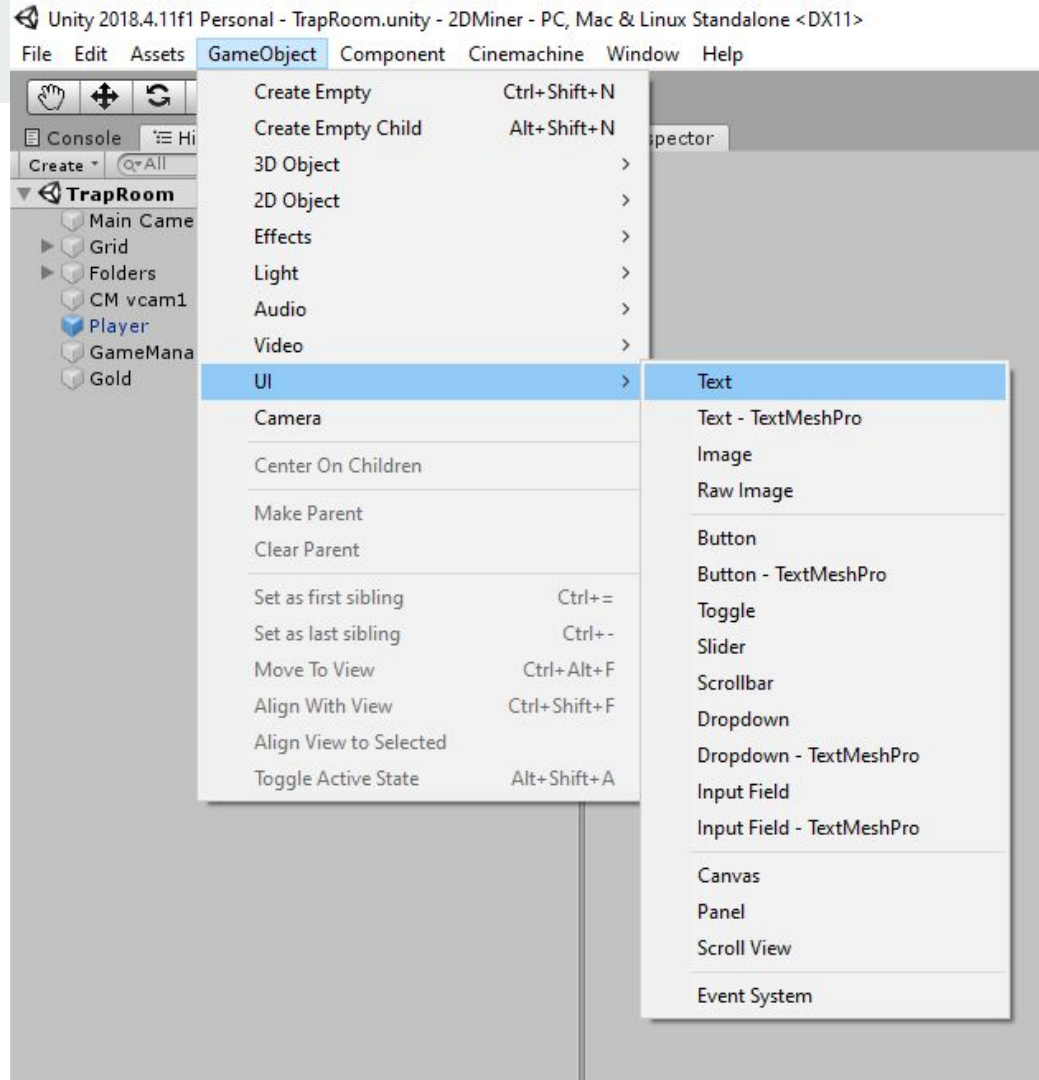
Жмяк сюда

Не сюда

И не сюда

В Unity об интерфейсе уже Подумали

И подготовили много элементов
упрощающих разработку





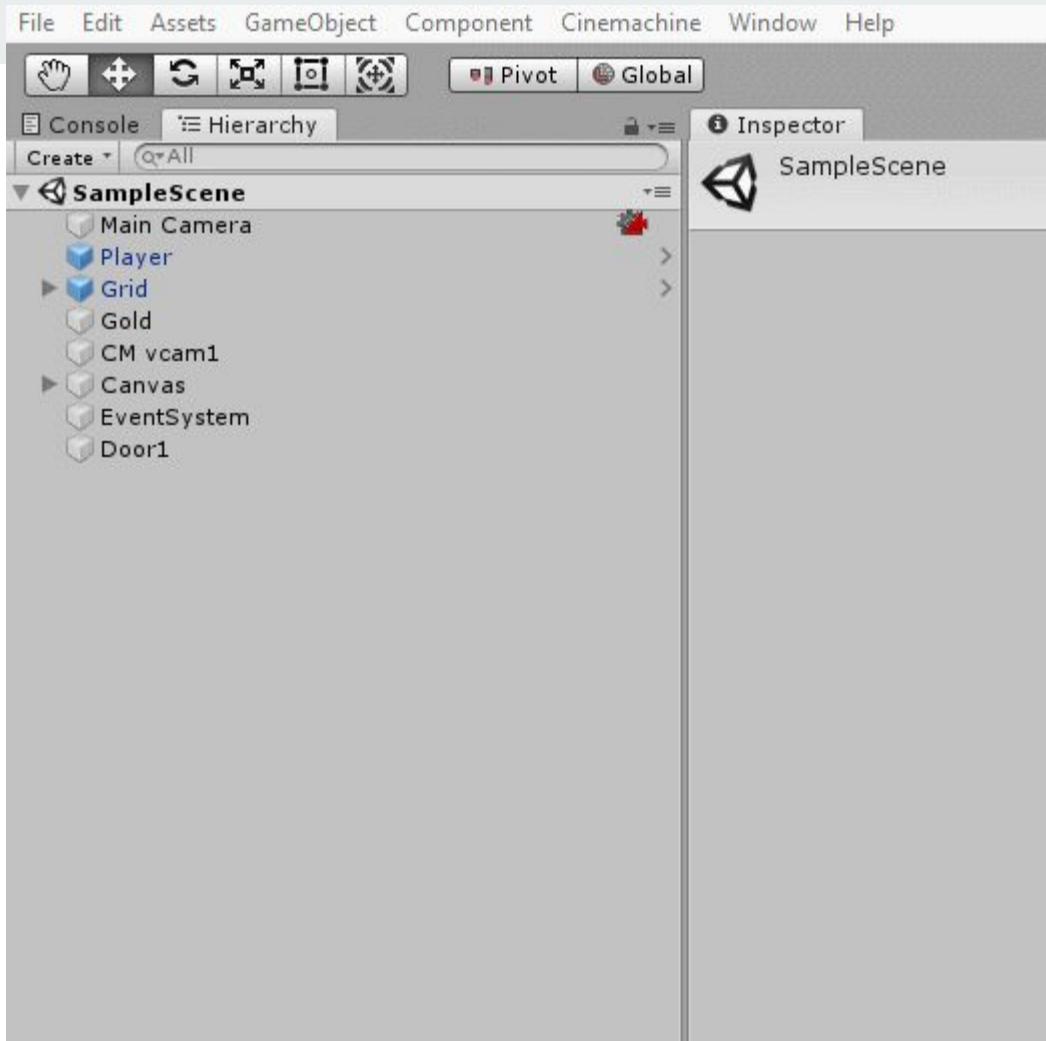
Для начала...

Брать информацию можно отсюда -> <https://docs.unity3d.com/Manual/UIToolkits.html>

Тут вся документация, но на АНГЛИЙСКОМ.

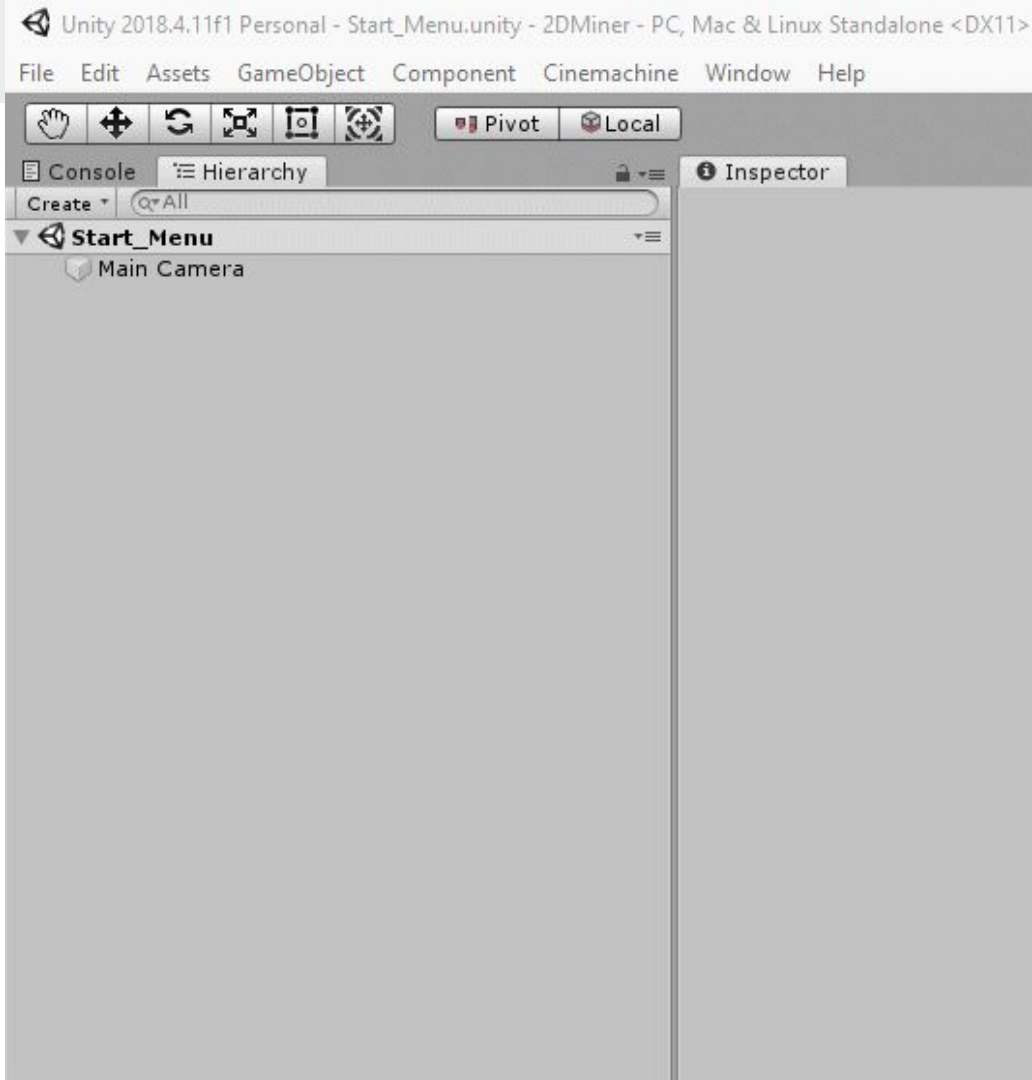
Создадим новую сцену

- 1) Назовите её Menu.
- 2) Переключитесь на эту сцену



Добавим в сцену кнопку

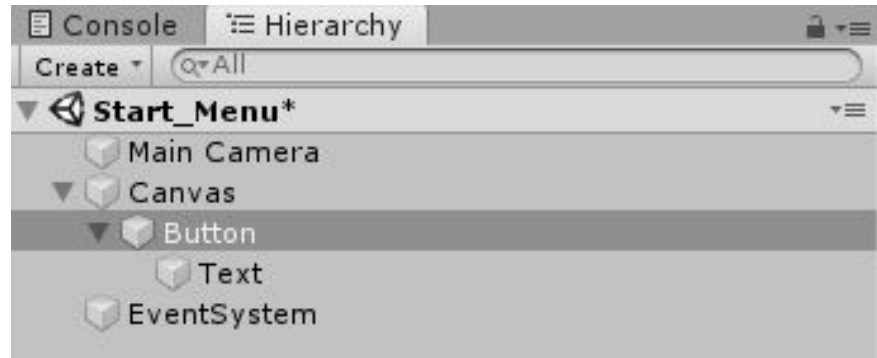
Для желающих, можно вернуться на слайд 6 и продумать из каких элементов состоит интерфейс меню игры HeartStone



Изучим дерево проекта

Видно что добавились объекты:

- Canvas
- Button
- Text
- EventSystem

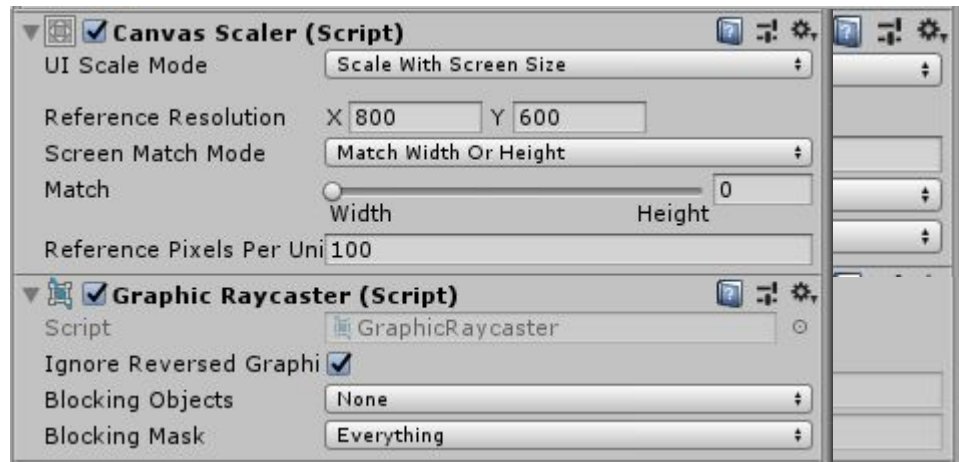


Элемент первый - *Canvas*

Свойство определяющее размеры полотна.

Свойства определяющие размеры окна.
По умолчанию размеры определяются системой Unity.
Режим отрисовки полотна. На что камера делает упор.

Ручные настройки размера полотна и прочие графические примочки

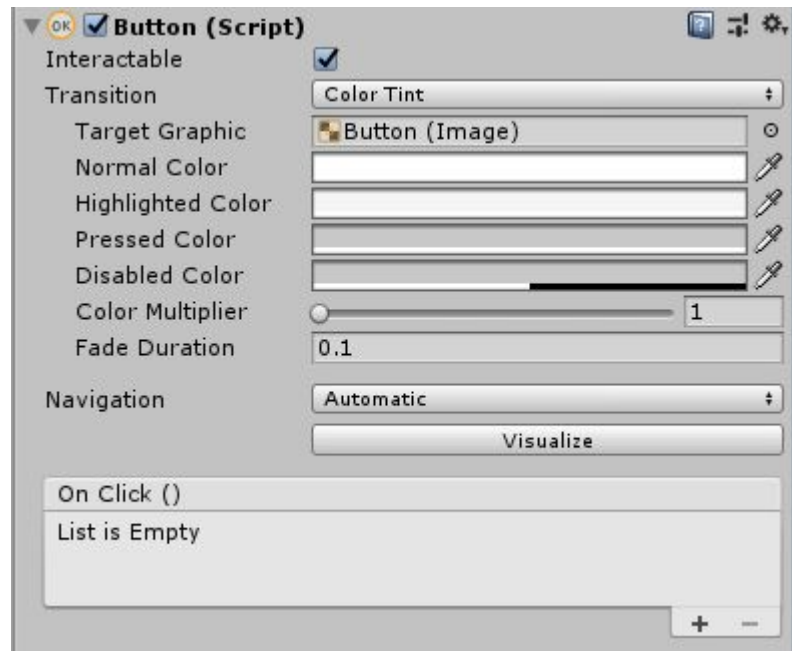


Элемент второй - *Button*

Нас тут интересует только два свойства:

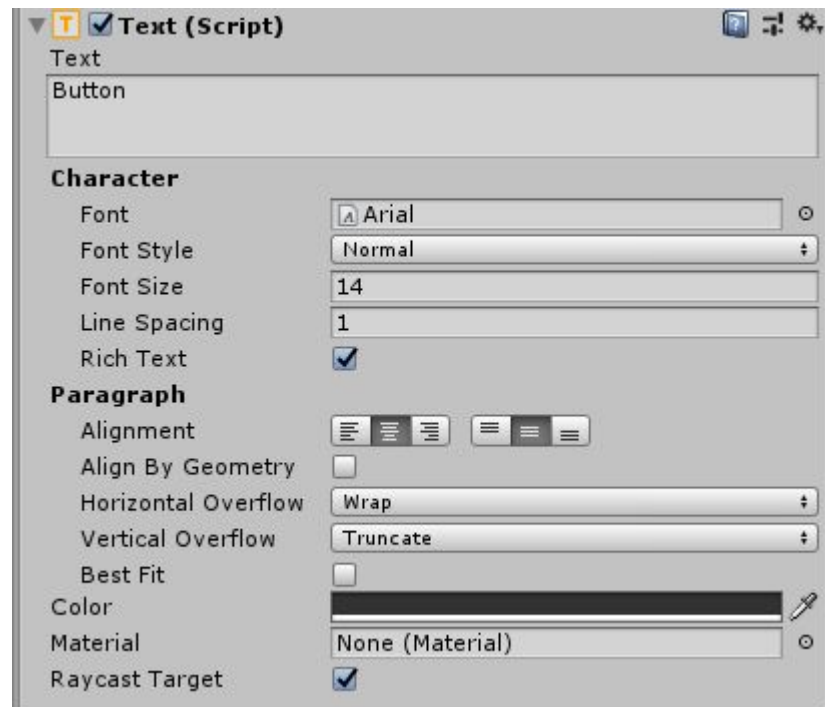
Фон и материалы кнопки

Реакция кнопки на нажатие



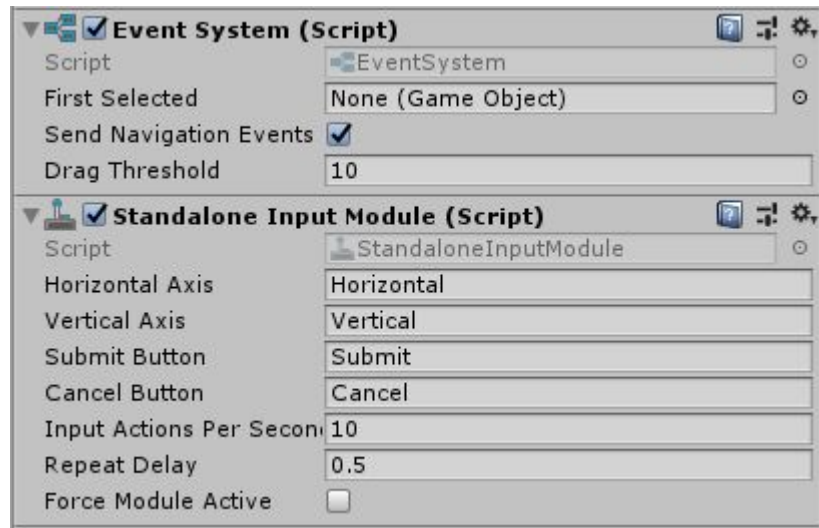
Элемент третий- *Text*

Тут настраивается типографика текста, шрифт выравнивание. Меняется сам текст.



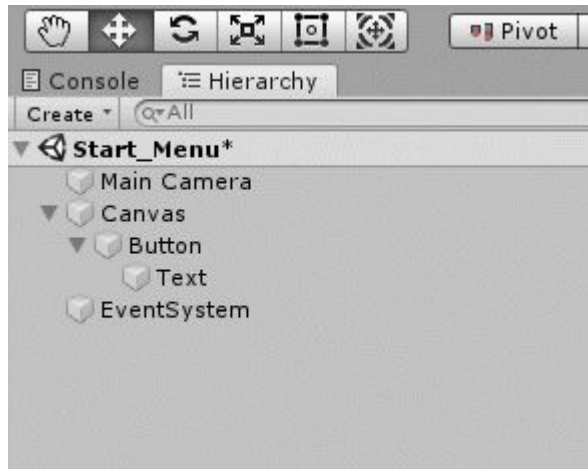
Элемент четвёртый - *EventSystem*

Автоматизированная система обработки взаимодействия с интерфейсом.



Что же, кнопку мы добавили


Теперь нужно её поставить в
нужное место.



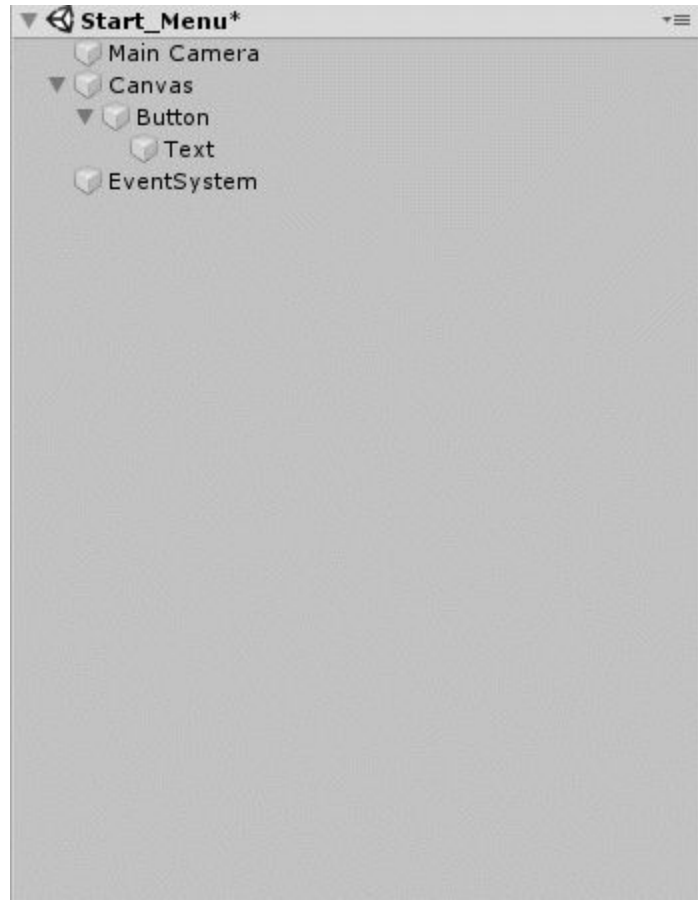


Задание 1

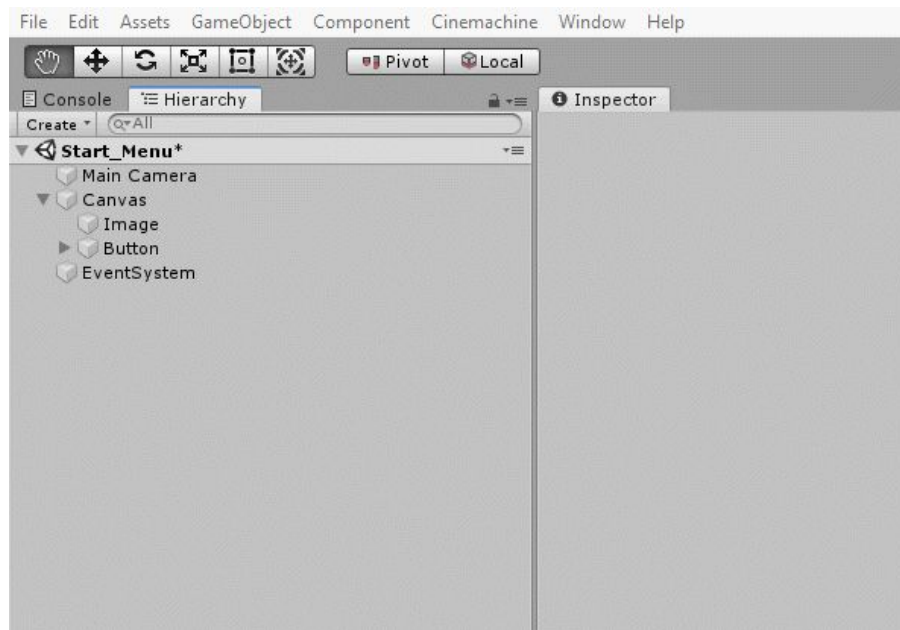
- 1) Добавьте ещё 2 кнопки.
- 2) Измените текст кнопок на:
 - Старт
 - Настройки
 - Выход



Добавим фоновое изображение



Изменим размер изображения



W

Inspector

☒ Image ☐ Static

Tag Untagged Layer UI

Rect Transform

stretch

Left	Top	Pos Z
350	356	0
Right	Bottom	
350	356	

Anchors

Min	X 0	Y 0
Max	X 1	Y 1
Pivot	X 0.5	Y 0.5

Rotation X 0 Y 0 Z 0

Scale X 8.3101 Y 8.3101 Z 8.3101

Canvas Renderer

☒ **Image (Script)**

Source Image None (Sprite)

Color

Material None (Material)

Raycast Target ☒

Default UI Material

Shader UI/Default

Add Component

Project

Create

Favorites

- All Materials
- All Models
- All Prefabs

Assets

- Gizmos
- Material
- Prefab
- Scenes
- Script
- Sprites**
- Suond_Sfx
- TextMesh Pro

Packages

Assets > Sprites >

Cavern Cavern2 Palette buzzsaw

buzzsaw-hit Cavern disciple-45x51 output-onlinep...

roguelike-nor... roguelikeDung... roguelikeDung...



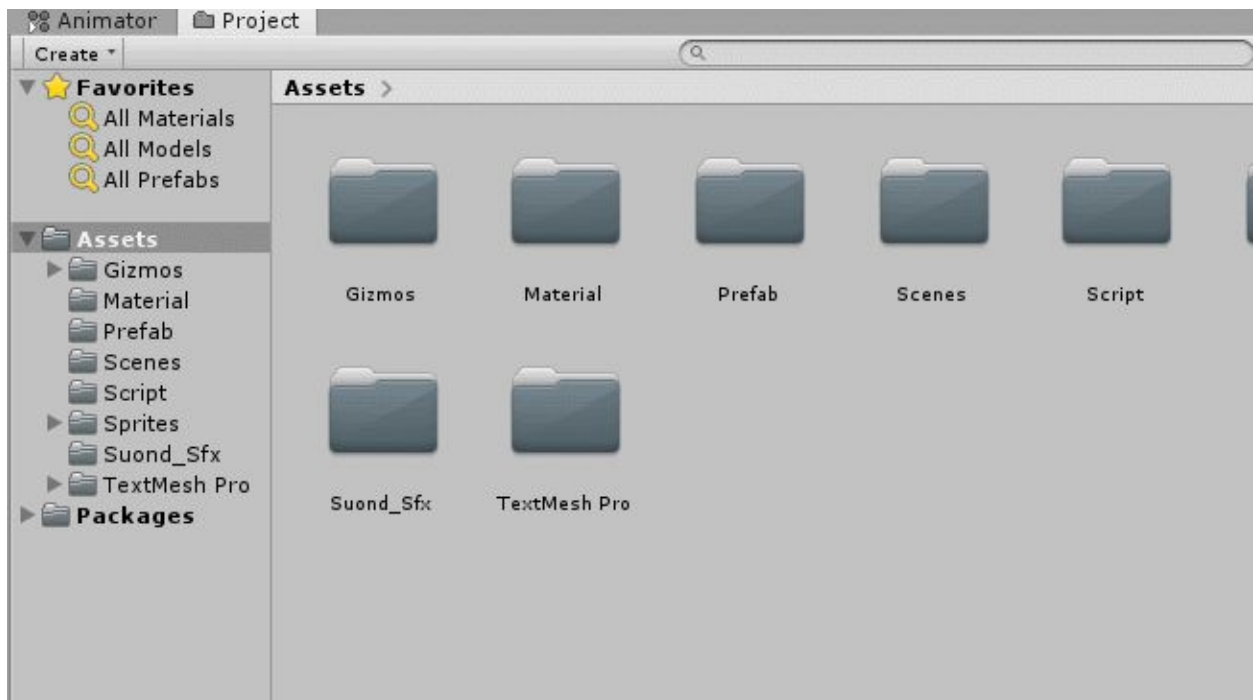
Заскриптуем переход

Создайте файл проекта скрипт C#

Назовите его:

ButtnController

После чего сразу
откройте в редакторе и
перепечатайте текст из
следующего слайда
презентации





Скрипт для работы с кнопками

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using System.Collections;

public class BtnnController: MonoBehaviour
{
    public void PressedStartLv1( int sceneID )
    {
        SceneManager.LoadScene(sceneID);
    }

    public void PressedExit()
    {
        Application.Quit();
    }
}
```

ВНИМАНИЕ! ЧАСТАЯ ОШИБКА

Очень важно делать одинаковые имена для файла и для класса в тексте.

Если в тексте написано `class MySCRIPT` значит и файл должен называться `MySCRIPT`.



```
public class MySCRIPT : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

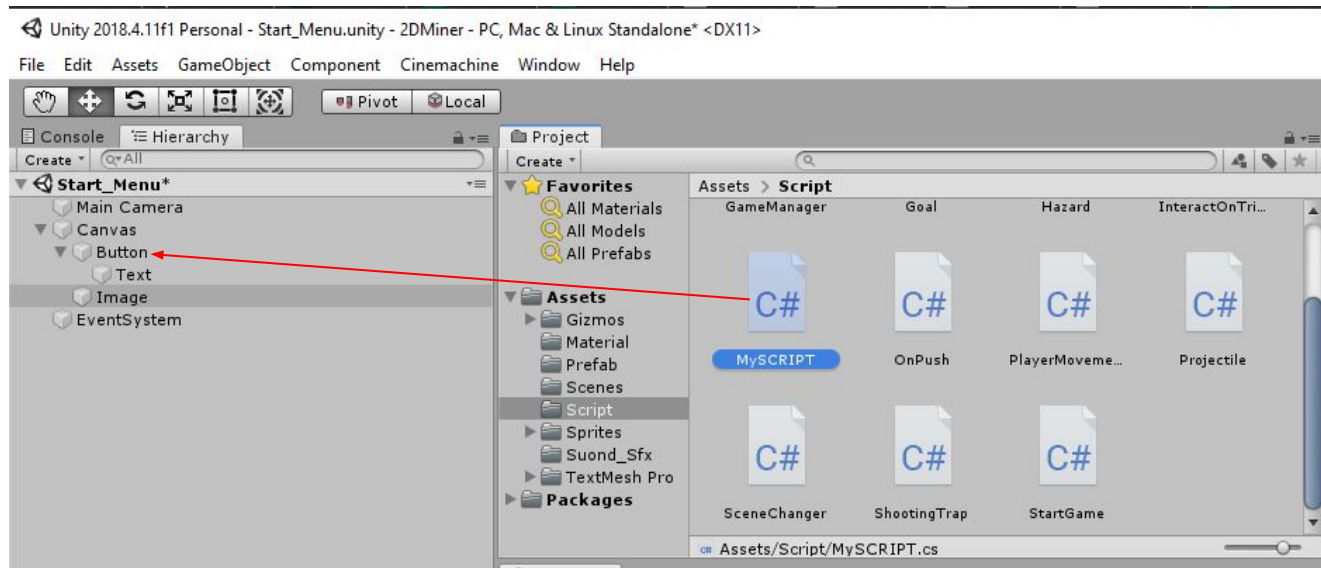
    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

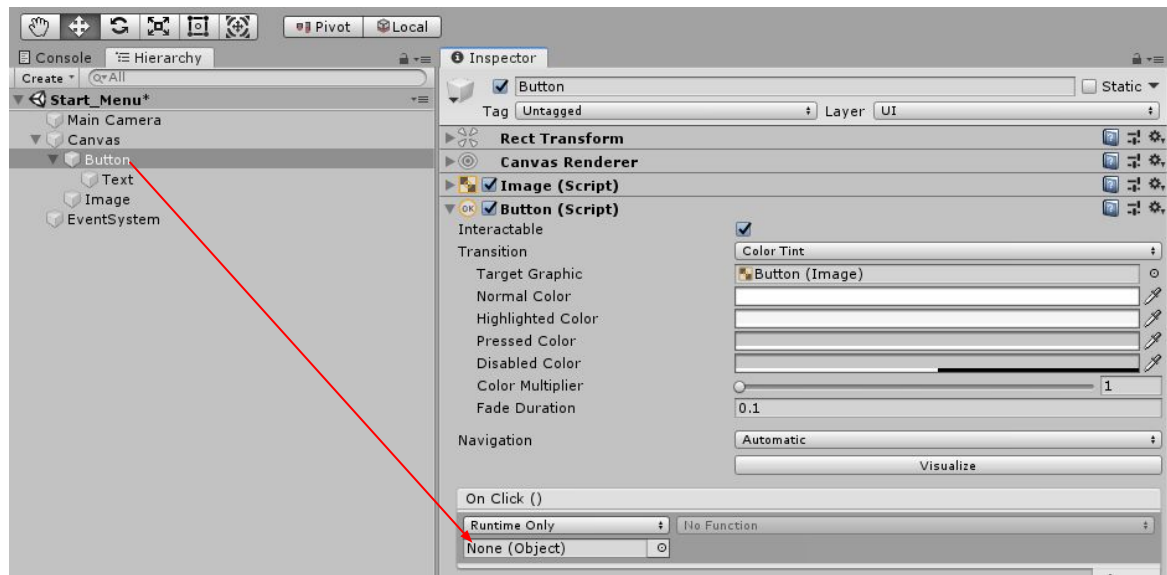
Примените скрипт на наши кнопки

Просто перетащите
скрипт на элемент
кнопки

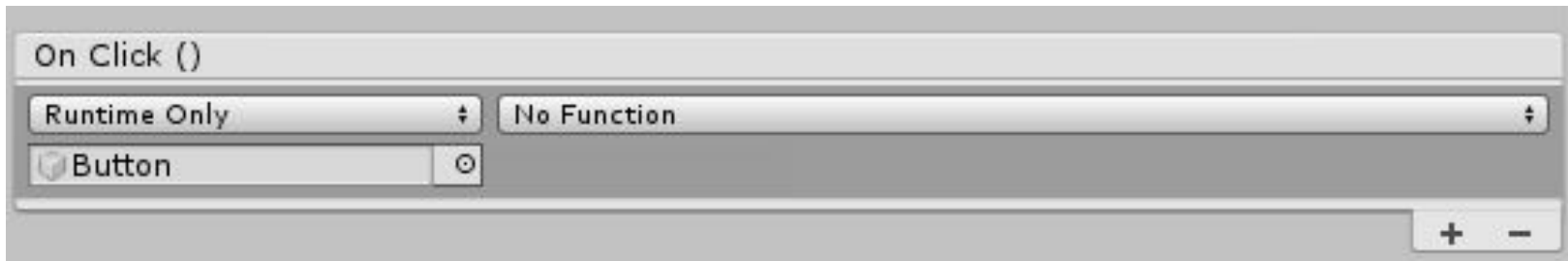


В свойствах кнопки

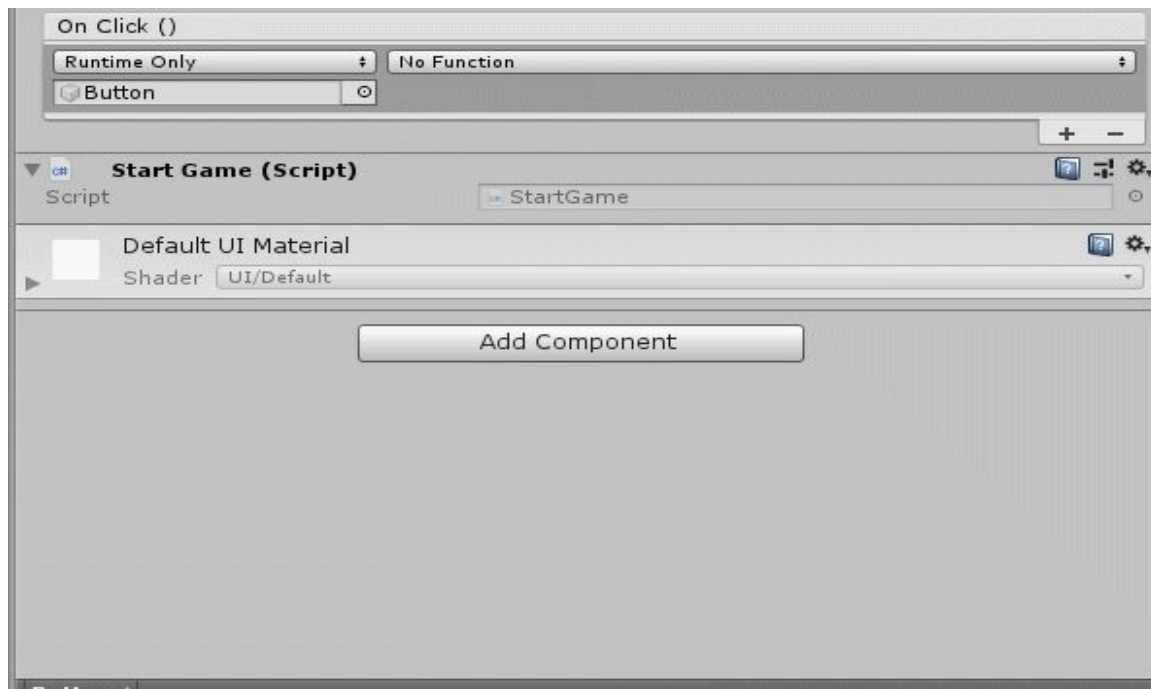
Перетащите объект кнопки в
свойство *on Click()*



Должно получиться так

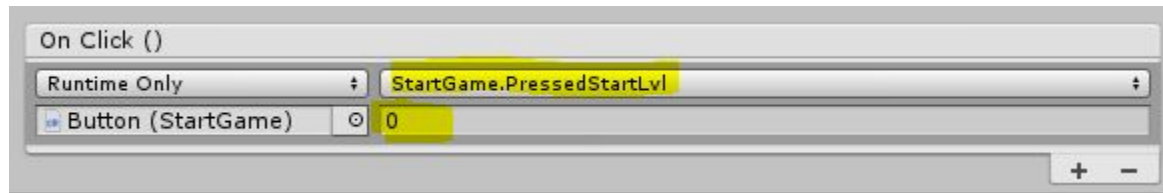


Выберем функцию которая выполнится



Для переключения между сценами

Выберите функцию *PressedStartLvl* и укажите номер сцены на которую вы хотите переключиться.



Номер сцены узнаете в окне BuildSettings

Чтобы в него попасть
нажмите комбинацию

Ctrl+Shift+B



У нас осталось время!

Теперь вы можете создавать кнопки и
делать их обработку. А также
переключаться между сценами.
Или можете консультироваться по вашей
игре у меня.
Задавайте вопросы, не стесняйтесь =)

