**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский политехнический университет»**

Методическое пособие

**«Создание интерфейса в Unity»**

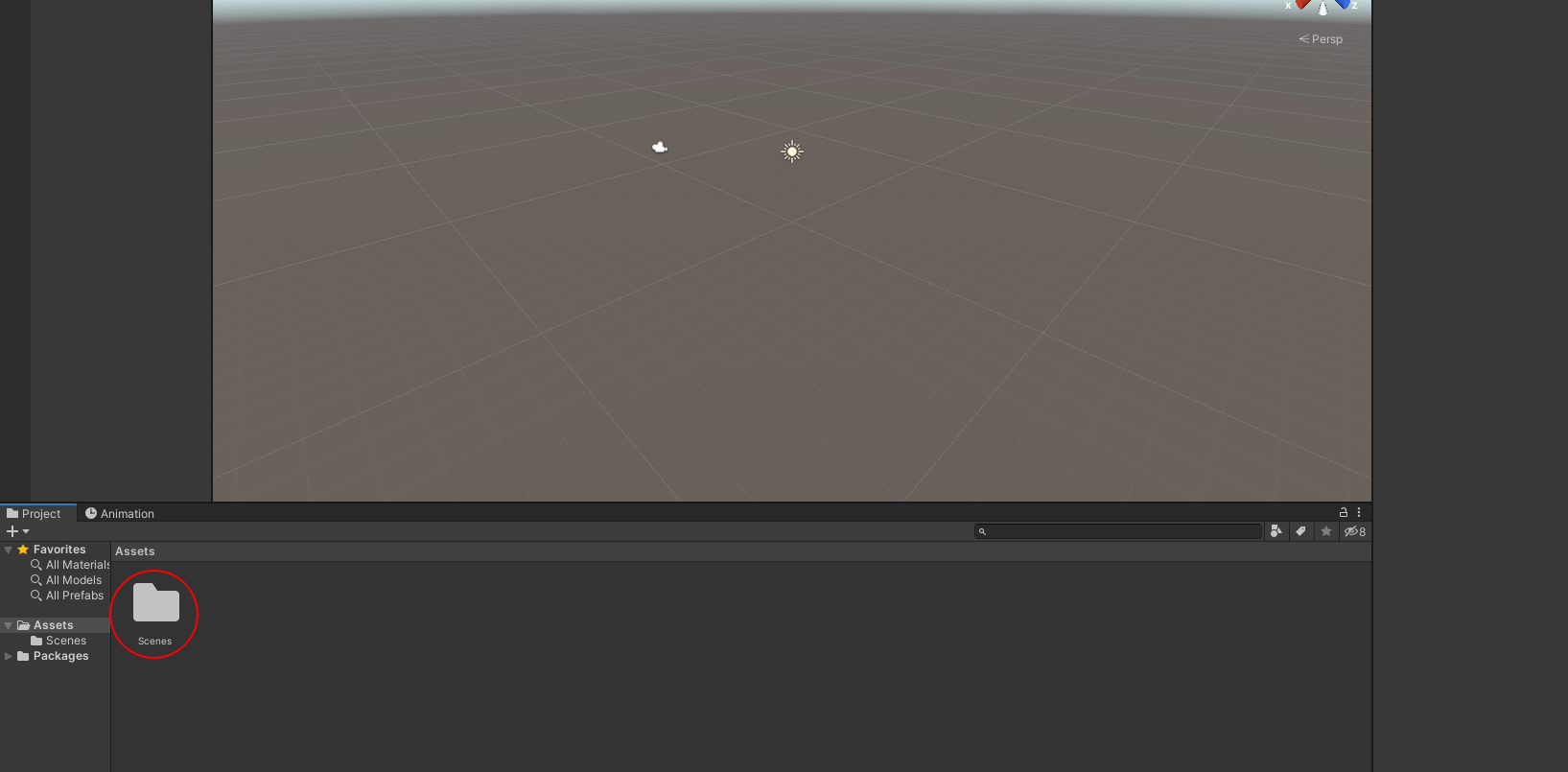
Выполнили: Гайнулин А.В

Москва, 2020

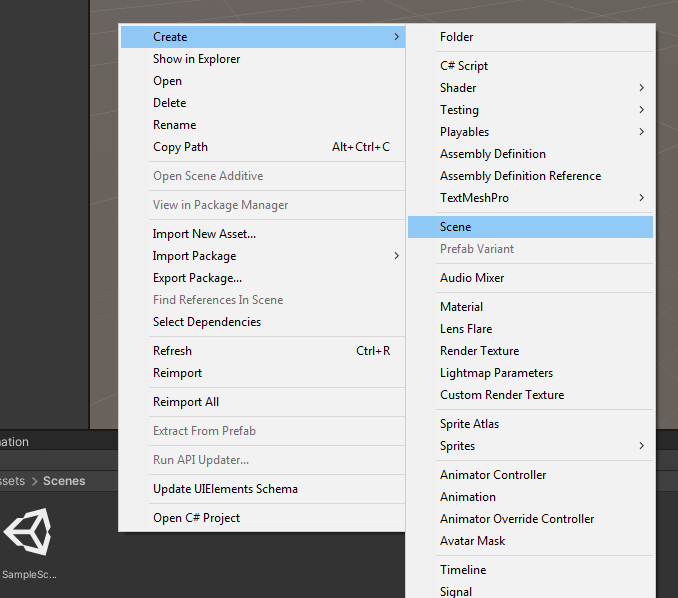
**Методические указания по созданию пользовательского интерфейса**

1. Создание панели, фон, кнопки.

Создадим новый проект, после чего в папке **Scene** на панели снизу **(Рис.1)** создадим новую сцену **(Рис.2)** и откроем её.

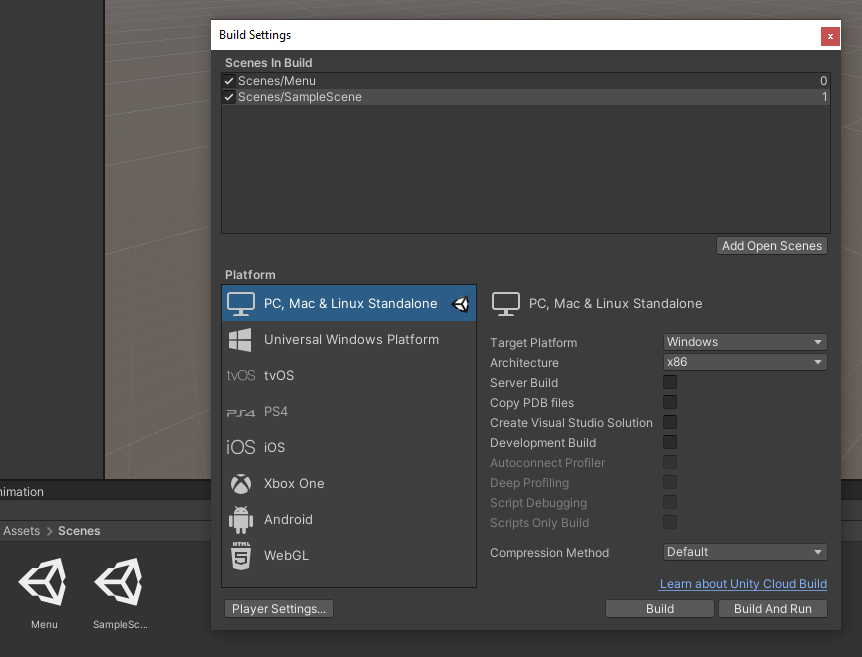


**Рис.1 Папка Scenes**



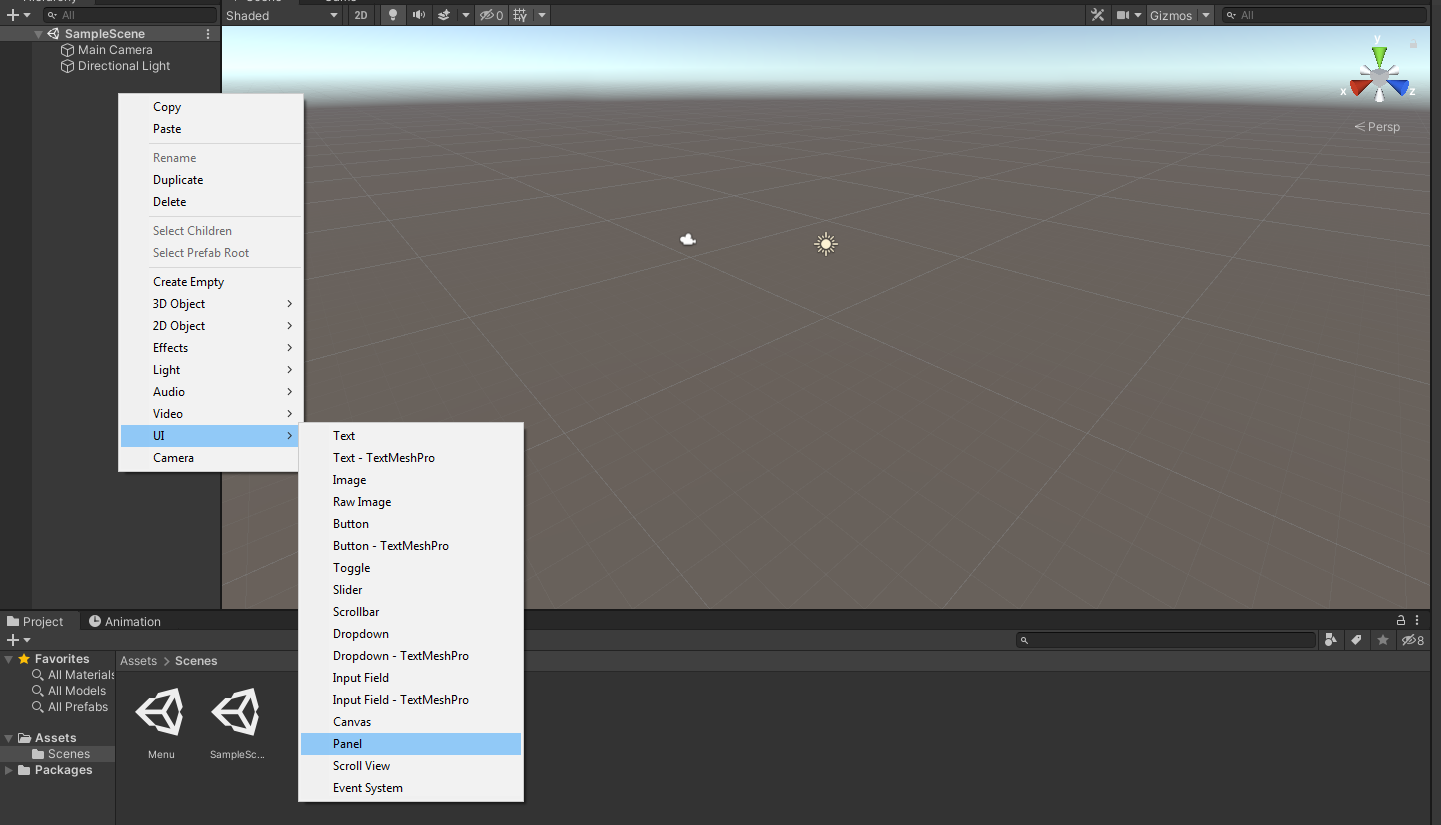
**Рис.2 Создание новой сцены.**

Теперь открываем **File->BuildSettings** и перетаскиваем созданные сцены в Scenes in build. **(Рис.3)**

****

**Рис.3**

После чего на правой панели справа жмём ПКМ и во вкладке **UI**  выбираем **Panel (Рис.4)**

****

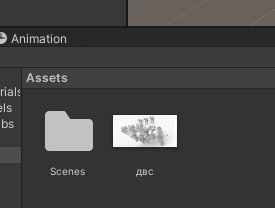
**Рис.4 Создание Canvas**

Сразу добавляется **Canvas** (Холст) и дочерним объектом к нему добавляется **Panel** (Панель) **(Рис.5)**

****

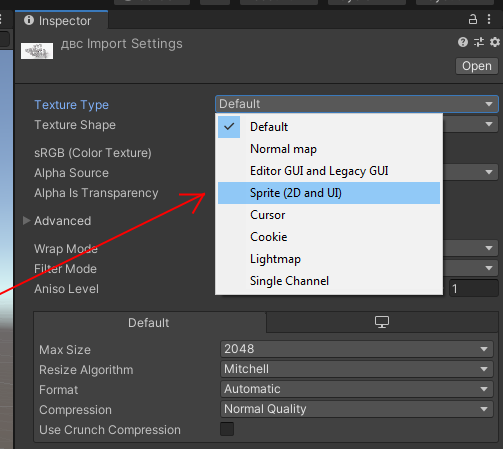
**Рис.5**

Справа в меню **Inspection** есть подменю **Image**, куда можно загрузить картинку для фона в **Source image**, предварительно добавив её в проект **(Рис.6)**

****

**Рис.6 Добавление картинки в проект**

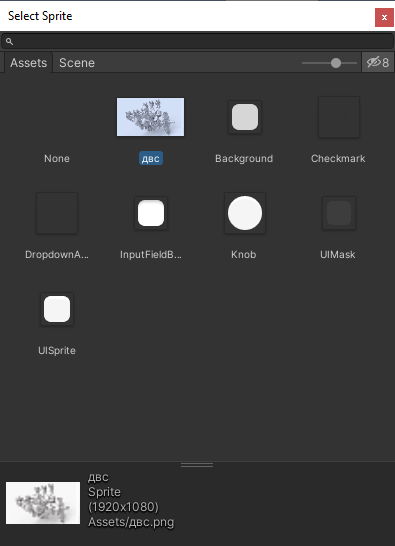
Чтобы **Panel** видел её, нужно выбрать картинку и в меню **Inspection** в графе **Texture type** выбрать **Sprite (2D and UI) Рис.7**

****

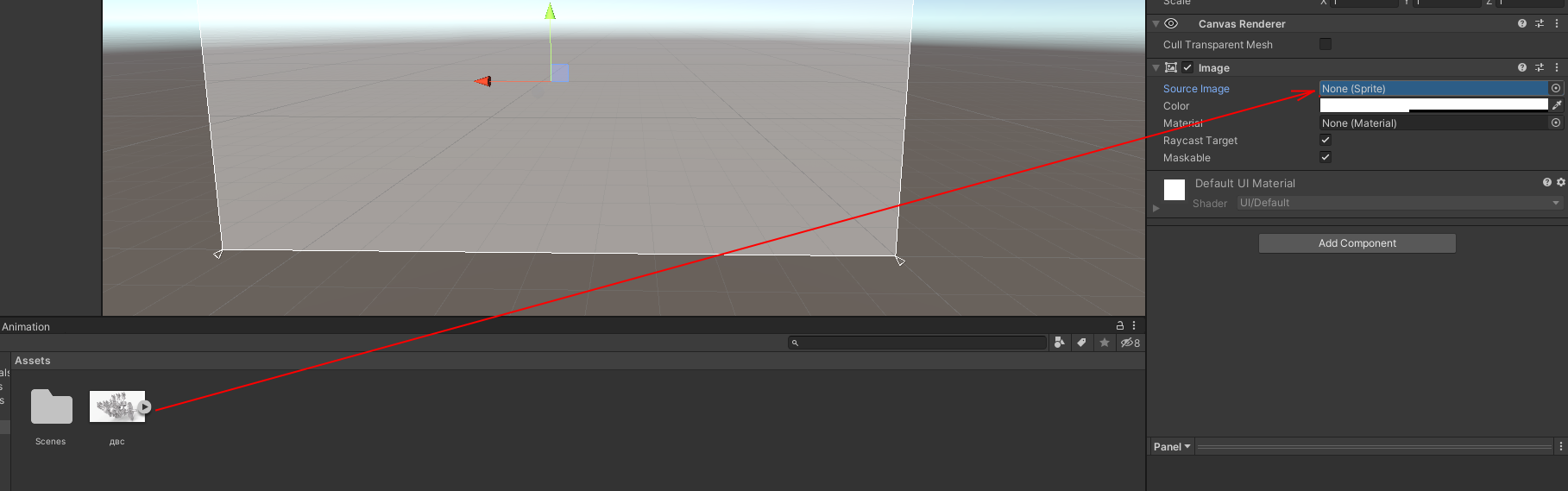
**Рис.7**

После чего можно выбрать эту картинку как фон в **Panel (Рис.8)**

*Примечание: Для всех объектов в Юнити работает drag-and-drop, например картинку фона можно сразу перенести в графу image* ***(Рис.9)***

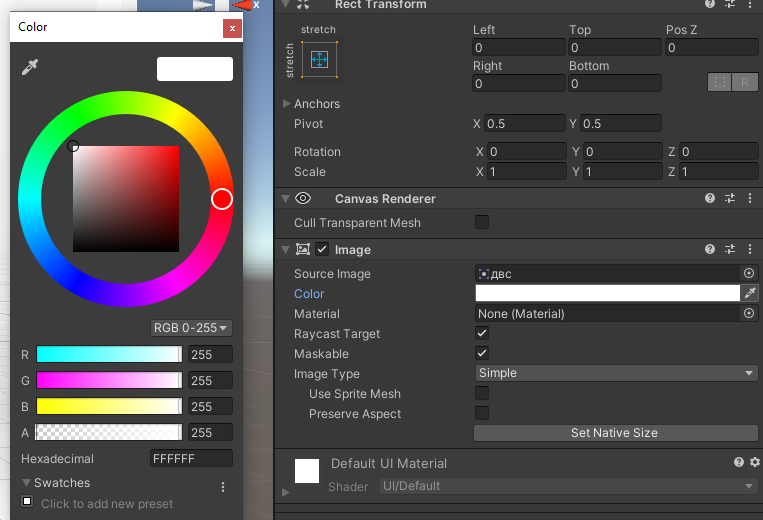


**Рис.8**



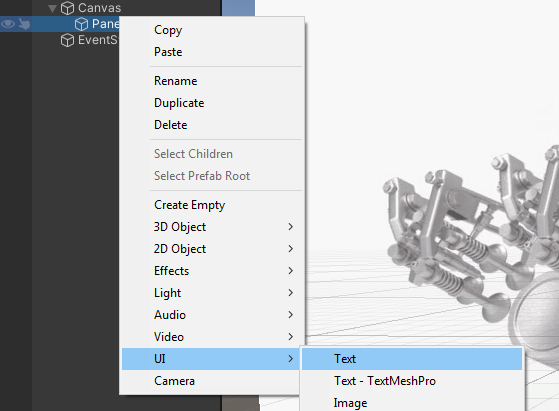
**Рис.9**

По умолчанию картинка на фоне имеет сильную прозрачность, чтобы её убрать открываем меню **Color** и выставляем для **A** значение 255 (по умолчанию стоит 100) **(Рис.10)**

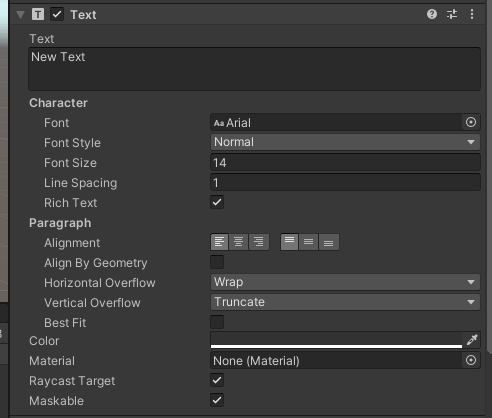
****

**Рис.10 Изменение прозрачности**

Теперь добавим текст к нашей панели, для этого также через ПКМ добавляем дочерний элемент для **Panel** **(Рис.11)** После чего Настраиваем текст так, как хочется с помощью компонента Text(Script) в инспекторе **(Рис.12)**

****

**Рис.11 Добавление текста на панель**

****

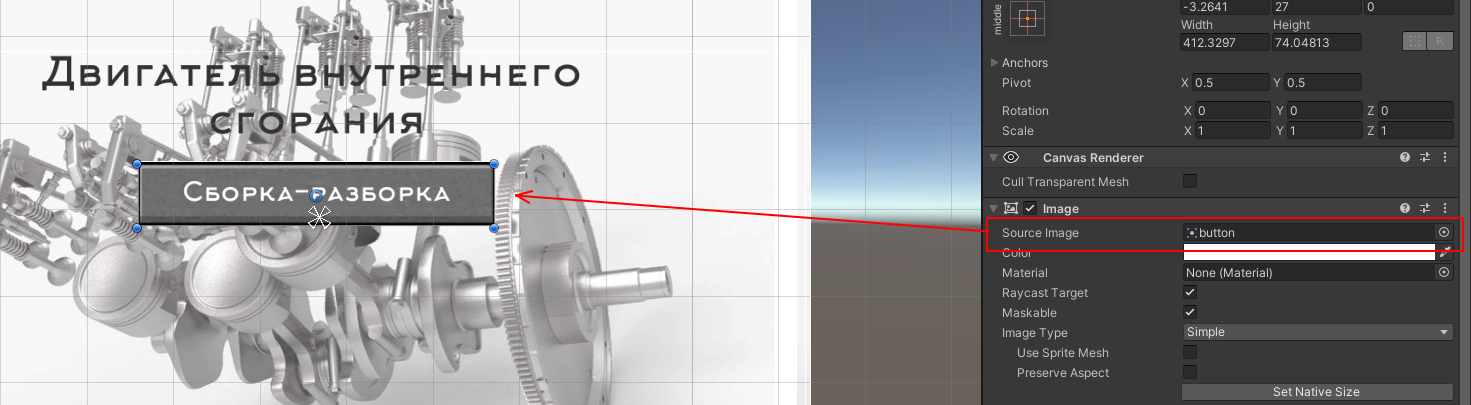
**Рис.12**

После этого добавим кнопки, для этого на **Panel** аналогично предыдущим действиям через ПКМ добавляем кнопку , у каждой кнопки есть дочерний элемент **Text**, отвечающий за текст на кнопке. **(Рис.13)**

****

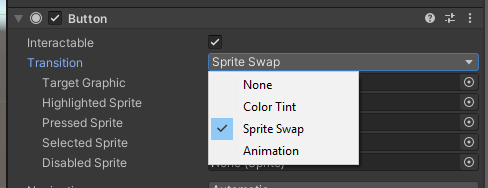
**Рис.13**

Чтобы добавить на кнопку спрайт, аналогично тому, что мы делали для фона, добавляем картинку в проект и делаем её спрайтом, после чего переносим её в поле **Source image (Рис14)**

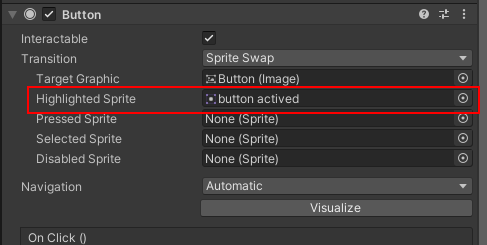
****

**Рис.14**

Чтобы изменить поведение кнопки при наведении, нужно изменить поле **Transition** c **Color tint** на **Sprite Swap (Рис.15)**, после чего добавить нужный спрайт в поле **Highlighted Sprite (Рис.16)**

****

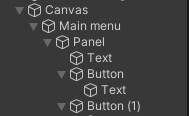
**Рис.15**

****

**Рис.16 Смена спрайта**

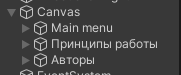
1. Переходы между панелями, программирование кнопок

Что бы переходить на меню опций не обязательно делать новую сцену.  
Для начала создаем пустой **GameObject (Create Empty)** как **Child** нашего Холста. Назовем его **Main menu**. Потом нашу панель, со всеми инструментами сделаем дочерними элементами этого объекта. Должно получиться так: **Рис.17**

****

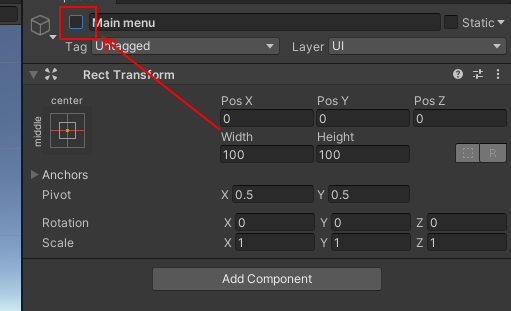
**Рис.17**

Выбираем наш MainMenu объект и сделаем его дубликат. С выбранным элементом нажимаем комбинацию клавиш Ctrl+D. У нас появится новый объект, переименуем его и создадим ещё один дубликат, соответственно для кнопок «Принципы работы» и «Авторы» **Рис.18**

****

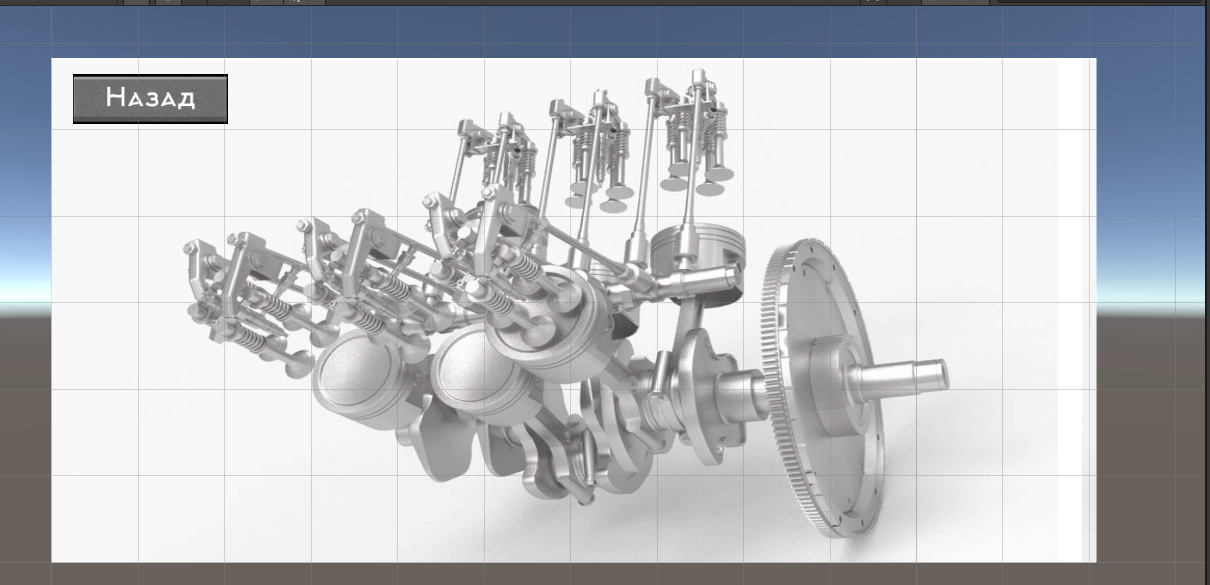
**Рис.18**

Для удобства дальнейшей работы инактивируем **MainMenu**, после чего будет видена следующая в иерархии панель – **Принципы работы**, с которой мы будем проводить дальнейшие действия **Рис.19**

****

**Рис.20**

Удалим все кнопки и надписи, оставим одну и назовём её «Назад» **(Рис.21)**

****

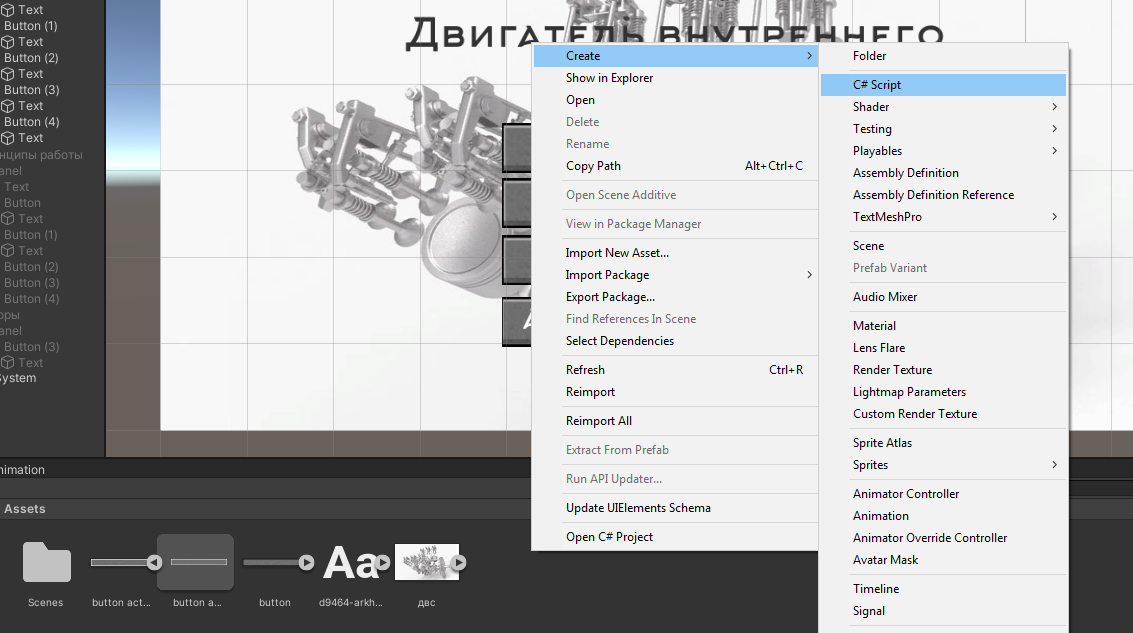
**Рис.21**

После чего можно добавить на панель нужный текст, аналогичные действия проводим с вкладкой **Авторы.**

Перейдем к написанию скрипта меню.

Нам нужно, чтобы по нажатию кнопок **«Сборка-разборка»** и **«Принципы работы»** у нас запускалась другая сцена, а по нажатию кнопки **Выход** приложение закрывалось. Это мы и пропишем в нашем скрипте.

Создаём в ассетах новый скрипт и отрываем его **(Рис.22)**.



**Рис.22 Создание нового скрипта**

Первое что надо сделать – удалить существующие методы **Start() и Update()** – тут они нам не нужны. Дальше нам надо подключить следующее:

using UnityEngine.SceneManagement;

После этого напишем свой метод для нажатия кнопки **Сборка-разборка** Метод должен быть **public** — нам нужно иметь возможность видеть его за пределами нашего скрипта

За загрузку сцены отвечает **SceneManager** и у него есть метод **LoadScene**. Существует несколько перегрузок метода. Можно передавать имя сцены, которую вы хотите загрузить. В нашем случае это сцена «SampleScene».

В итоге функция будет выглядеть следующим образом.

public void PlayPressed()

{

SceneManager.LoadScene("SampleScene");

}

Также создадим метод для выхода из приложения

public void ExitPressed()

{

Application.Quit();

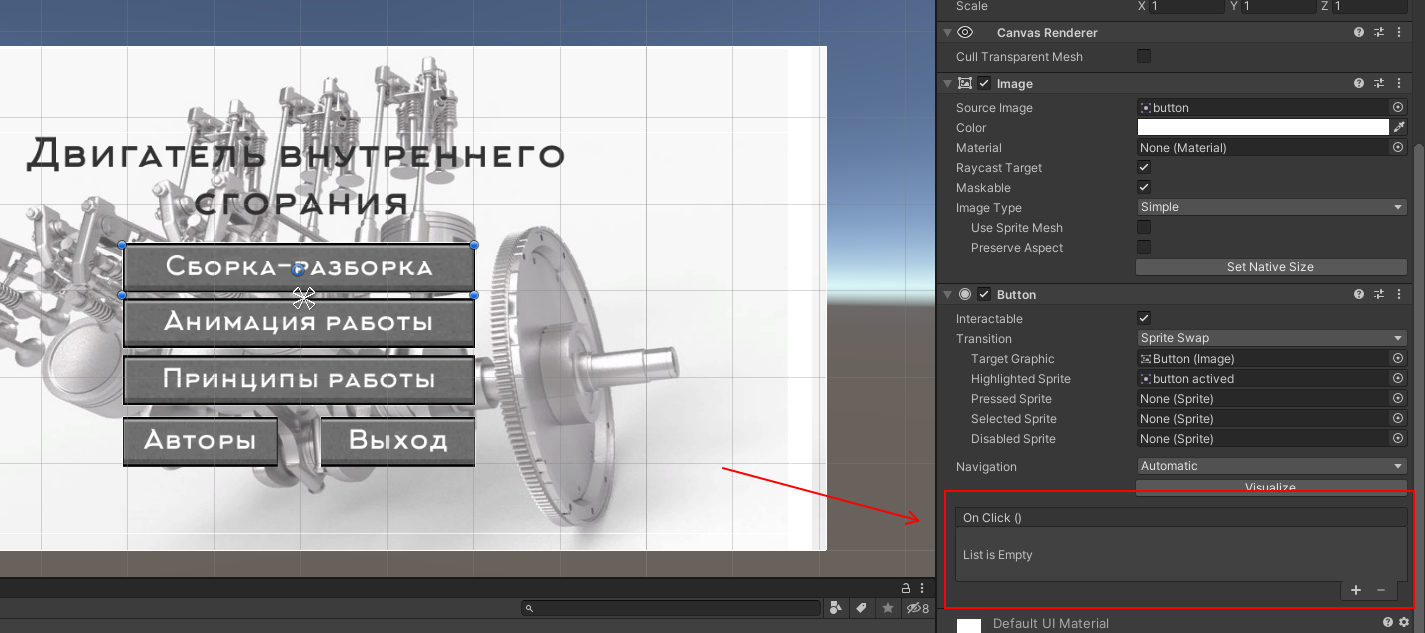
}

Однако в Юнити мы не увидим результата работы этого метода, так как подобное работает только в билде. Для того что бы проверить, что все работает правильно, добавляем в метод строчку

Debug.Log("Exit pressed!");

После чего сохраняем скрипт и возвращаемся в Юнити.

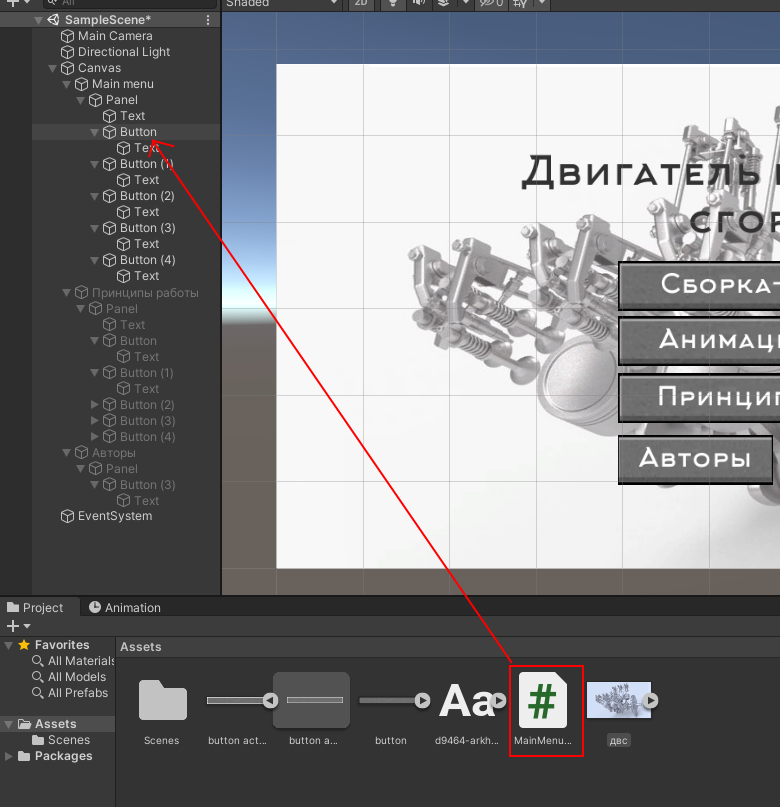
Теперь нам необходимо прикрепить события кнопок на эти методы, для этого выбираем нужную кнопку и находим в инспекторе в подменю **Button, On click() (Рис.23)**

****

**Рис.23**

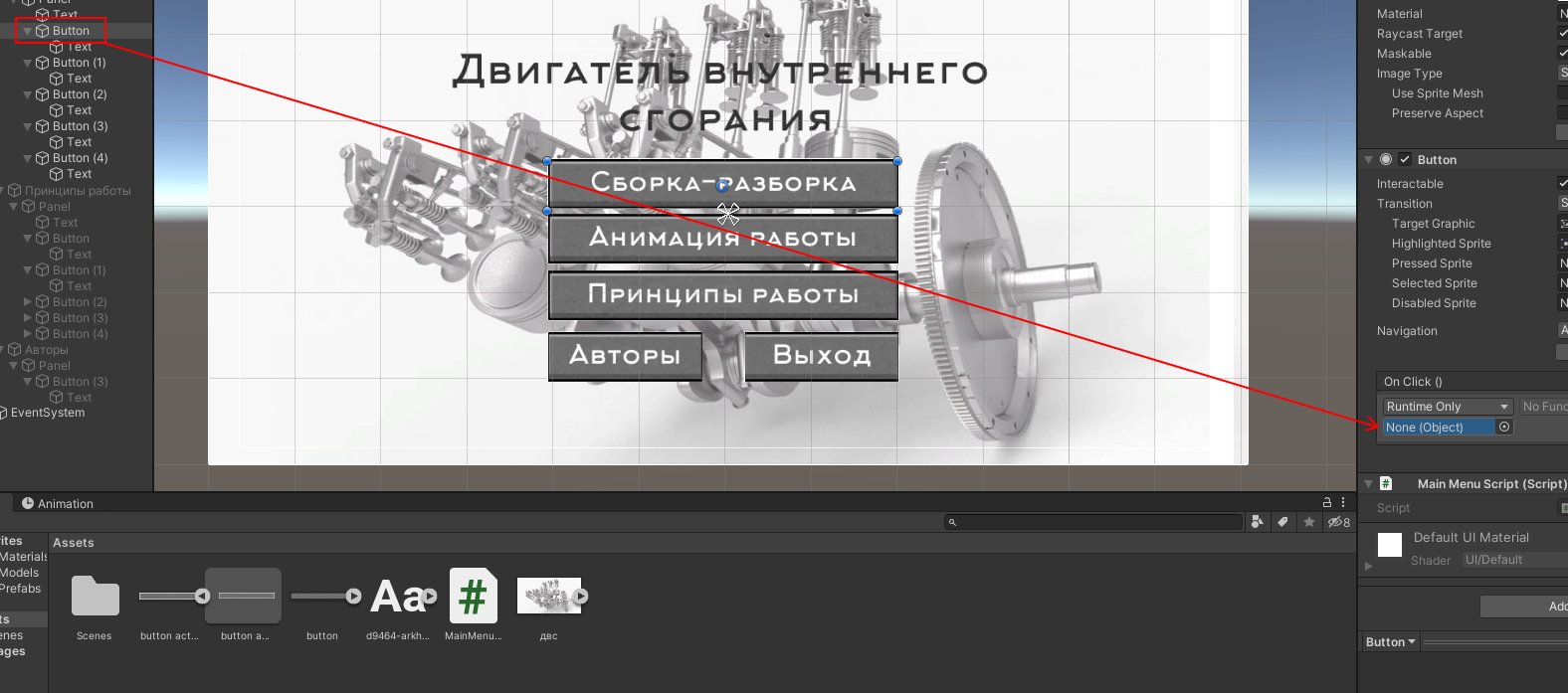
Это событие кнопки, которое по нажатию будет вызывать подписанные на событие методы. Добавляем метод нажимая на +.

После чего нам нужно добавить скрипт в саму кнопку для этого Drag-and-drop’ом перетаскиваем скрипт прямо на кнопку в дереве **(Рис.24)**

****

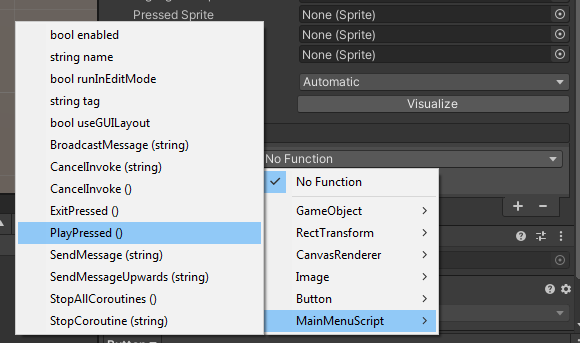
**Рис.24**

После чего в подменю **Button, On click()** в поле **None (Object)** переносим нашу кнопку (в данный момент она является контейнером со скриптом, если его добавить сразу, не будут отображаться методы, которые мы писали) **(Рис.25)**

****

**Рис.25**

После чего в поле **No function**  выбираем нужную функцию **(Рис.26)**

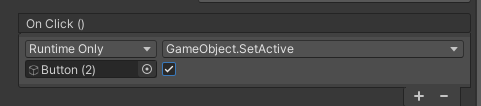
****

**Рис.26 Выбор скрипта для кнопки**

Аналогично поступаем для кнопки «Анимация работы», предварительно поменяв имя сцены, на которую будет совершён переход, и также для кнопки «Выход».

Для кнопок **Принципы работы** и **Авторы** нам не нужно писать никакой код, так как некоторый функционал уже встроен.

Суть в том, что мы будем активировать **Принципы работы** или **Авторы**. На этом этапе у вас должен быть активным **MainMenu**, а **Принципы работы** или **Авторы** не должно быть видно. Так же видим, что когда мы активируем **Принципы работы** или **Авторы**, он полностью перекрывает **Menu**. Суть в том, что играет роль порядок расположения дочерних объектов Холста – в каком порядке они расположены в иерархии в том порядке они и будут прорисовываться. Поскольку **Принципы работы** и **Авторы** у нас находятся над **Main Menu**, то они перекрывают меню. Это мы и будем использовать.  
  
Выбираем кнопку **Принципы работы** и в **OnClick()** перетаскиваем нашу кнопку **Принципы работы,** аналогично тому как мы делали с переносом кнопки со скриптом ранее . В функциях выбираем **GameObject ->SetActive()**; и ставим галочку. **(Рис.27)**

****

**Рис.27**

Это означает, что при нажатии на кнопку эта панель будет активировать и перекрывать панель с главным меню.

Аналогичные действия производим с кнопкой **Авторы**.

Также для того, чтобы всё работало корректно, нужно перед запуском приложения инактивировать панели **Принципы работы** и **Авторы.**

Для кнопок назад производятся аналогичные действия, только галочка не ставится, следовательно при нажатии эта панель будет снова неактивна.

На этом методическое пособие заканчивается.