

Создание показателя здоровья у врага

Первым делом необходимо создать в объекте “mannequin” объект “UI” - “Canvas”, как указано на рисунке 91. После, необходимо в объекте “Canvas” создать “UI” - “Slider”, как указано на рисунке 92. Также необходимо удалить объект “Handle Slide Area”, который находится в объекте “Slider” и изменить в объекте “Background” параметр “Color”, чтобы цвет был не яркий, как указано на рисунке 93.

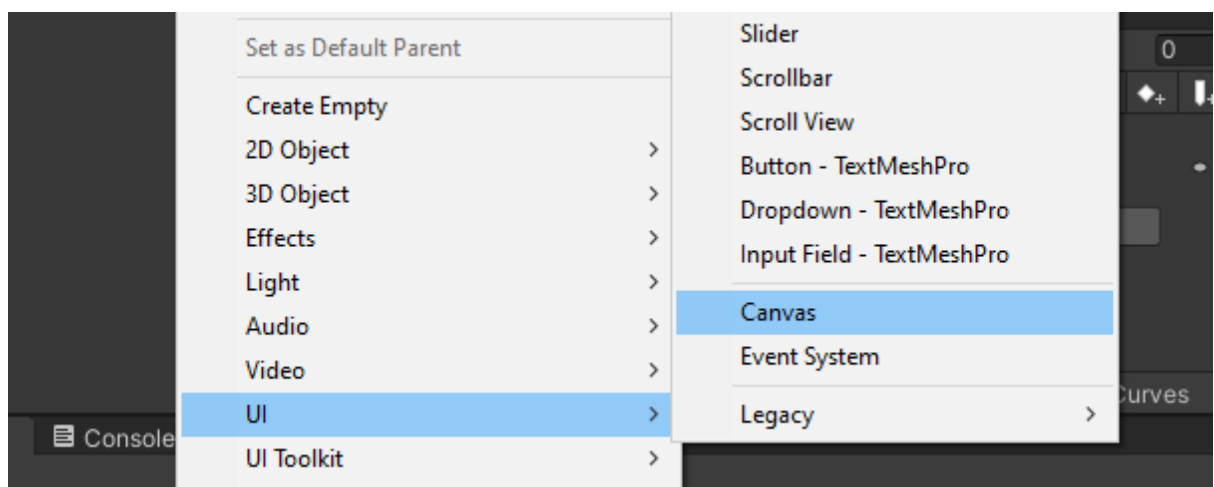


Рисунок 91. Создание “Canvas”

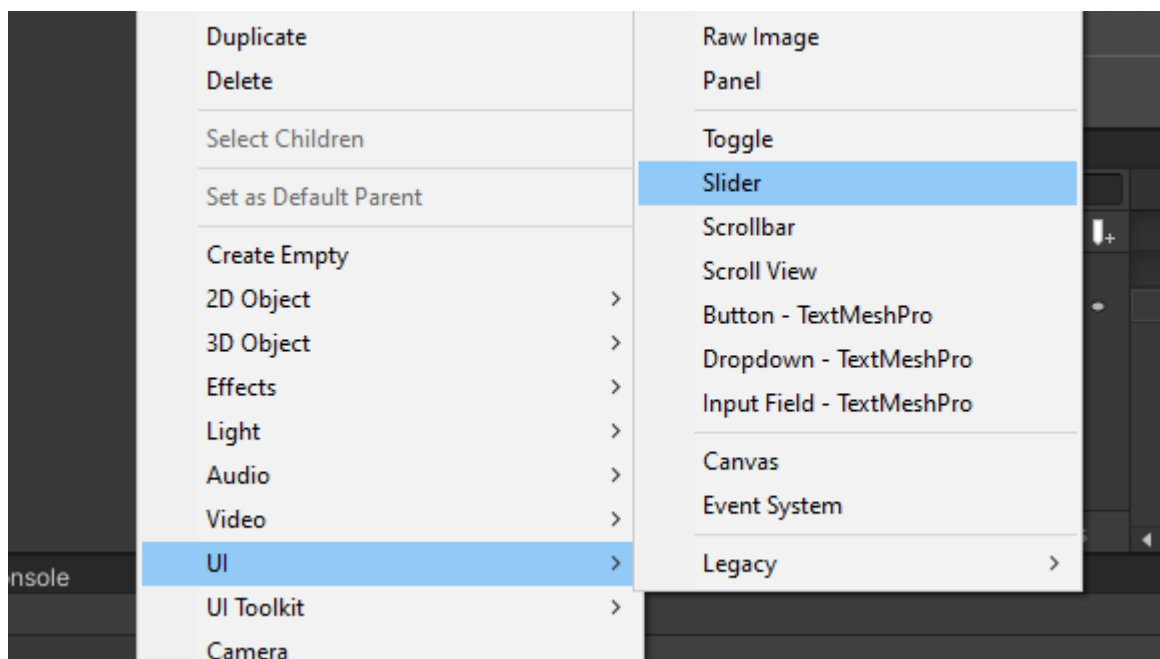


Рисунок 92. Создание “Slider”

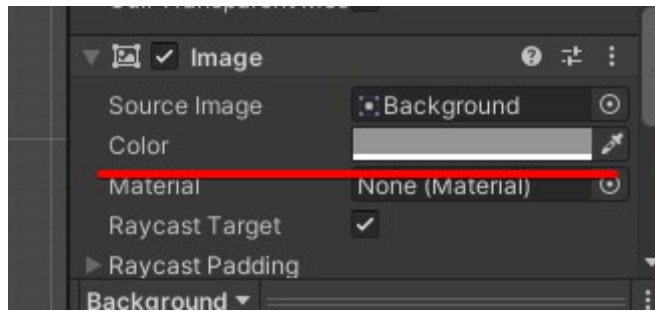


Рисунок 93. Изменение цветового параметра в объекте “Background”

В объекте “Slider” необходимо изменить параметры, указанные на рисунке 94, а именно:

1. Убрать галочку у “Interactable”
2. “Transition” должен быть равен “None”
3. “Navigation” должен быть равен “None”

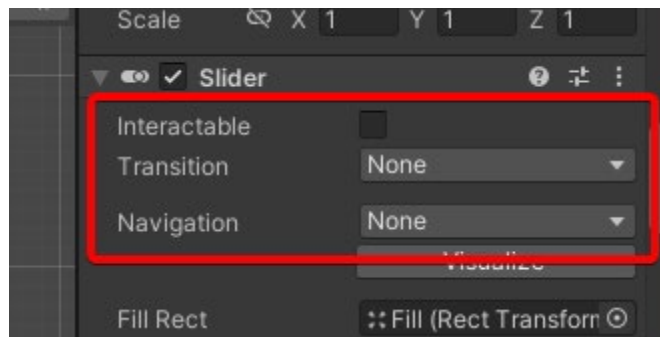


Рисунок 94. Параметры объекта “Slider”

Теперь для изменения цвета в зависимости от получаемого урона, необходимо создать скрипт с названием “HPBar” и добавить его в объект “Canvas”, который будет содержать следующий код:

```
using UnityEngine;
```

```
using UnityEngine.UI;
```

```
public class HPB : MonoBehaviour
```

{

Переменная для выбора объекта с которым будет вестись работа

public Slider Slider;

Переменная для выбора местонахождения слайдера

public Vector3 Offset;

Метод, который принимает два значения, для отображения здоровья

public void SetHealth(float health, float maxHealth)

{

Показывает объект “Slider” после того, как здоровье персонажа будет меньше максимального

Slider.gameObject.SetActive(health < maxHealth);

Отображаемый уровень здоровья равен текущему здоровью

Slider.value = health;

Максимальное здоровье равно взятой из метода переменной

Slider.maxValue = maxHealth;

Необходимо для смены цвета от зеленого до красного

Slider.fillRect.GetComponentInChildren<Image>().color = Color32.Lerp(Color.red, Color.green, Slider.normalizedValue);

Оператор, необходимый для скрытия объекта “Slider” если текущее значение меньше или равно нулю

if (Slider.value <= 0)

```
{
```

Скрывает объект “Slider”

```
Slider.gameObject.SetActive(false);
```

```
}
```

```
}
```

```
void Update()
```

```
{
```

Позиция объекта “Slider” всегда будет находится в позиции персонажа

```
Slider.transform.position =
```

```
Camera.main.WorldToScreenPoint(transform.parent.position + Offset);
```

```
}
```

```
}
```

А также добавить в скрипт “mannequin” следующий код:

Переменная, которая необходима для соединения скрипта этого скрипта со скриптом “HPB”

```
public HPB Healthbar;
```

```
void Start()
```

```
{
```

Метод, вызываемый из скрипта “HPB” для присваивания максимального здоровья к текущему

```
Healthbar.SetHealth(_currentHealth, _maxHealth);
```

```
}
```

```
public void TakeDamage(int damage)
```

```
{
```

Метод, вызываемый из скрипта “HPB” для присваивания максимального здоровья к текущему

```
Healthbar.SetHealth(_currentHealth, _maxHealth);
```

```
}
```

После чего перенести в объект “Canvas”, объект “Slider” в параметр “Slider” у скрипта “HPB”, как указано на рисунке 95. А также объект “Canvas” перенести в параметр “Healthbar” у скрипта “mannequin”, как указано на рисунке 96. И необходимо изменить параметр “Offset”, чтобы объект “Slider” находился над персонажем.

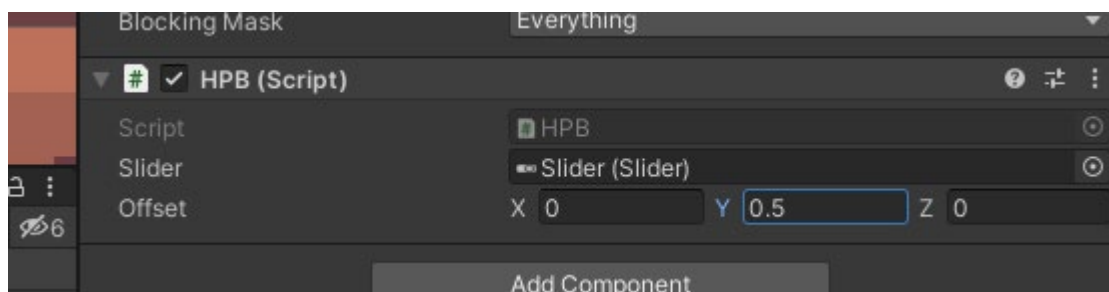


Рисунок 95. Заполнение параметра “Slider”

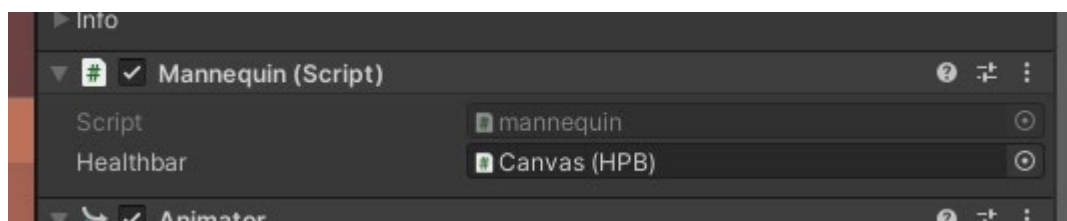


Рисунок 96. Заполнение параметра “Healthbar”