

Соединение компонентов со скриптом

После сохранения скрипта, необходимо перейти в проект и выбрать объект “Player”, после чего в скрипте появится место для подключения компонента “Rigidbody 2D”, указанное на рисунке 41. Для того, чтобы подключить компонент к скрипту, необходимо нажать на элемент, указанный на рисунке 42, и в открывшемся окне выбрать объект “Player”, чтобы получилось как на рисунке 43.

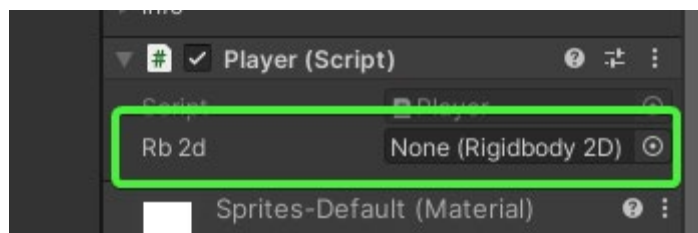


Рисунок 41. Место для подключения компонента

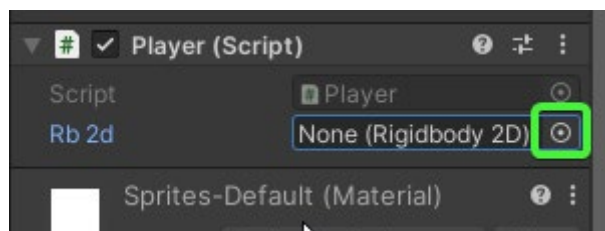


Рисунок 42. Кнопка для добавления объектов

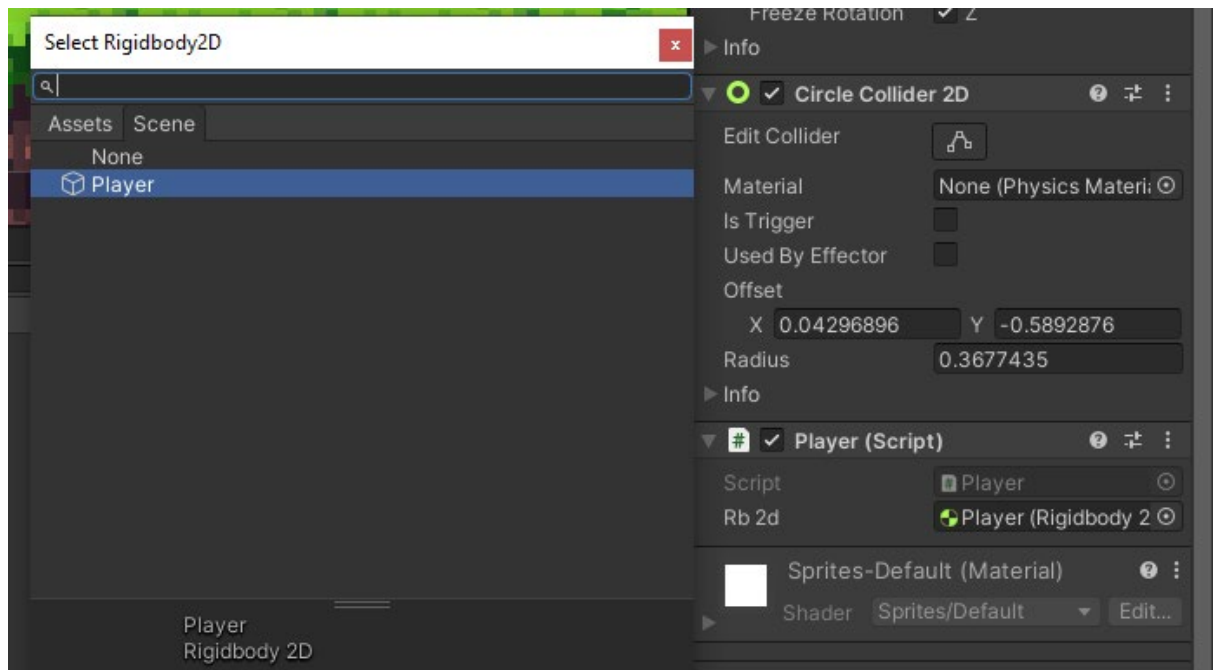


Рисунок 43. Соединение объекта со скриптом

Для смены спрайта в зависимости от вектора движения, необходимо добавить в скрипт “Player” следующий код:

`private bool isRight = true;` приватная булевая переменная, отвечающая за поворот персонажа. Дает понять, что изначально персонаж смотрит направо.

```
void Update()
```

```
{
```

Если переменная `isRight` верна и скорость персонажа по оси `x` меньше нуля, то активируется метод `Flip()`

```
if (isRight && rb2d.velocity.x < 0)
```

```
{
```

```
Flip();
```

```
}
```

Если переменная `isRight` не верна и скорость персонажа по оси `икс` больше нуля, то активируется метод `Flip()`

```
if (!isRight && rb2d.velocity.x > 0)

{

    Flip();

}

}
```

Метод `Flip()` поворачивает объект на 180 градусов и меняет значение переменной `isRight` на противоположное тому, которое имеет сейчас

```
void Flip()

{

    transform.Rotate(0, 180, 0);

    isRight = !isRight;

}

}
```

Для того, чтобы персонаж умел прыгать, необходимо создать новый объект с названием “Groundcheck” в объекте “Player”, как указано на рисунке 44. Этот объект будет называться дочерним, так как он закреплён за другой.

После создания объекта “Groundcheck” ему необходимо:

1. создать новый скрипт с названием “GroundCheck”
2. добавить компонент “Box Collider 2D”

3. поставить галочку напротив “is Trigger” у компонента “Box Collider 2D”, как указано на рисунке 45.
4. переместить измененные границы коллайдера под персонажа, как указано на рисунке 45.

Скрипт “GroundCheck” отвечает за проверку соприкосновения персонажа с землей и должен иметь следующий вид:

```
using UnityEngine;
```

```
public class GroundCheck : MonoBehaviour
```

```
{
```

Создается приватная булевая переменная `ground`

```
bool ground = true;
```

Создается публичный булевый метод хранящий в себе значение переменной `_ground`

```
public bool isGround()
```

```
{
```

```
    return ground;
```

```
}
```

Метод отвечающий за соприкосновение коллайдеров

```
private void OnTriggerEnter2D(Collider2D collision)
```

```
{
```

Если коллайдер имеет тег “Ground”, то переменная `ground` равна истине

```

        if (collision.tag == "Ground")

        {

            ground = true;

        }

    }

```

Метод отвечающий за отсоединение коллайдеров

```

private void OnTriggerExit2D(Collider2D collision)

{

```

Если коллайдер имеет тег “Ground”, то переменная ground равна не истине

```

        if (collision.tag == "Ground")

        {

            ground = false;

        }

    }

}

```

После сохранения скрипта, необходимо добавить в объект “Ground” тег с названием “Ground”. Для того чтобы создать тэг, необходимо нажать на любой объект и в окне “Inspector” перейти в раздел добавления тэгов, указанный на рисунке 46, после чего создать новый тег по той же схеме как создавались слои.

Теперь осталось добавить новые строки в скрипт “Player”, а именно:

Приватная переменная с названием `_jump`, которая отвечает за скорость по оси `y`

```
float _jump = 6f;
```

Публичная переменная “GroundCheck” которая соединяет данный скрипт со скриптом имеющим название этой переменной

```
public GroundCheck GroundCheck;
```

```
void Update()
```

```
{
```

Если нажата клавиша “Space” и метод `isGround` равен истине, дает скорость персонажу по оси `Y` и умножает эту скорость на скорость в переменной `_jump`

```
if (Input.GetButtonDown("Jump") && GroundCheck.isGround())
```

```
{
```

```
rb2d.velocity = Vector2.up * _jump;
```

```
}
```

```
}
```

```
}
```

Последним действием необходимо перенести объект “Groundcheck” в отведенное место у скрипта, как на рисунке 47.

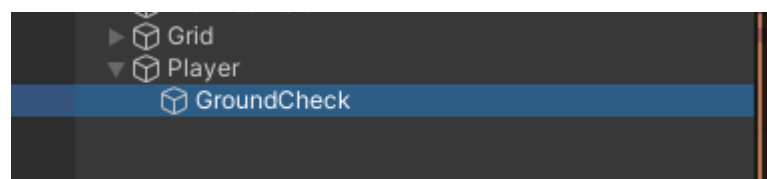


Рисунок 44. Создание дочернего объекта

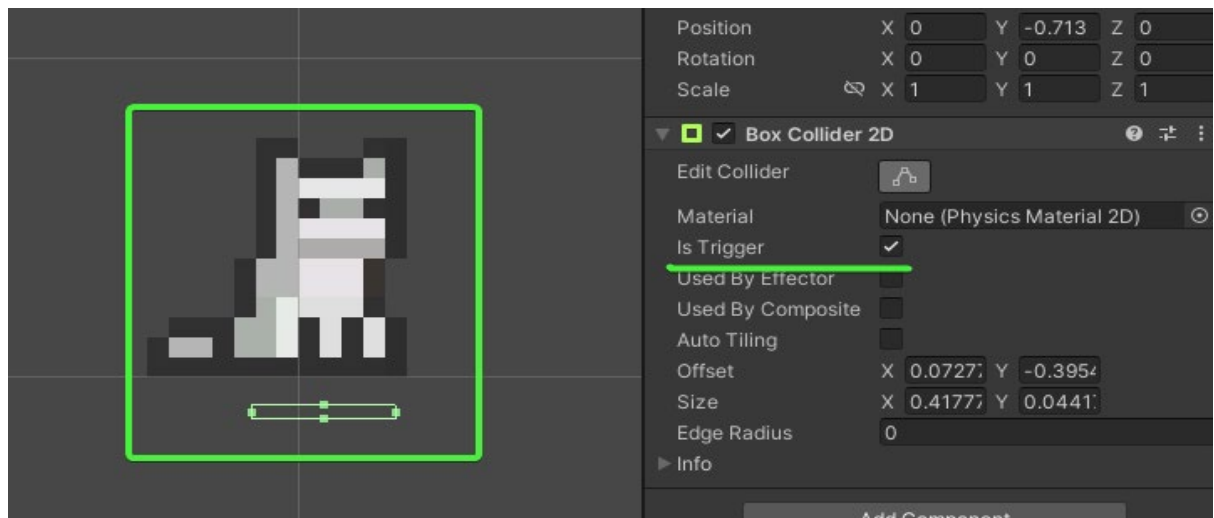


Рисунок 45. Подготовка компонента для использования

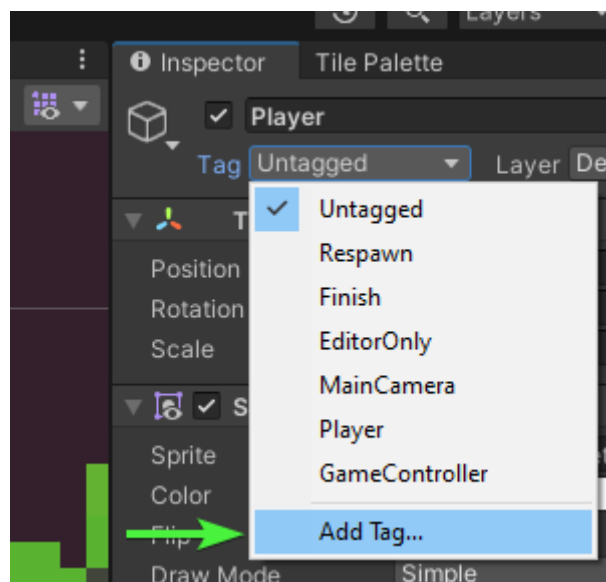


Рисунок 46. Добавление тэга

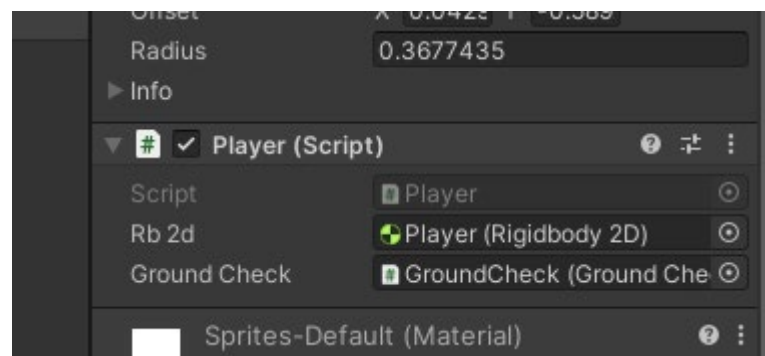


Рисунок 47. Соединение скриптов