Создание врага

Для того, чтобы создать врага, необходимо добавить его спрайты в проект и подготовить их, как указывалось выше. После того как спрайты были нарезаны, необходимо создать новый объект с названием “Chicken”и добавить в него такие компоненты как:

1. “Sprite Renderer” и изменить его параметры, как указано на рисунке 71.
2. “Circle Collider 2D” и изменить его параметры, как указано на рисунке 72.
3. “Rigidbody 2D” и изменить его параметры, как указано на рисунке 73.
4. Скрипт с названием “EnemyChicken”.

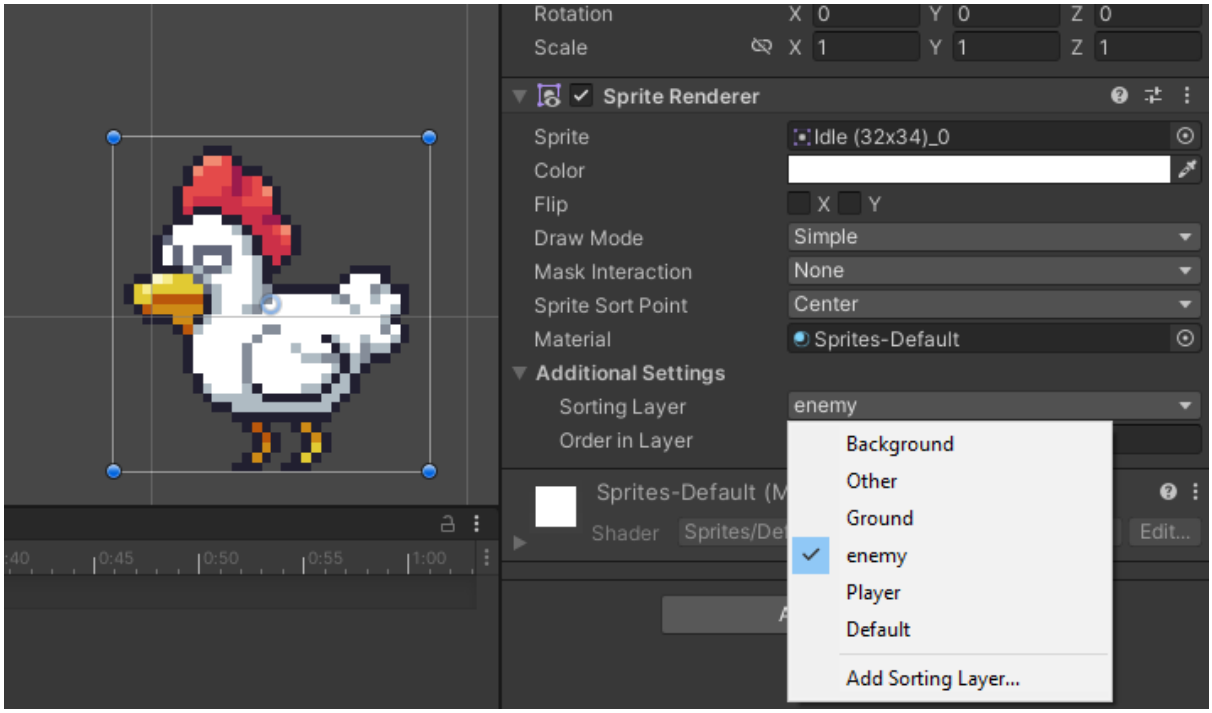

Рисунок 71. Изменение компонента “Sprite Renderer”

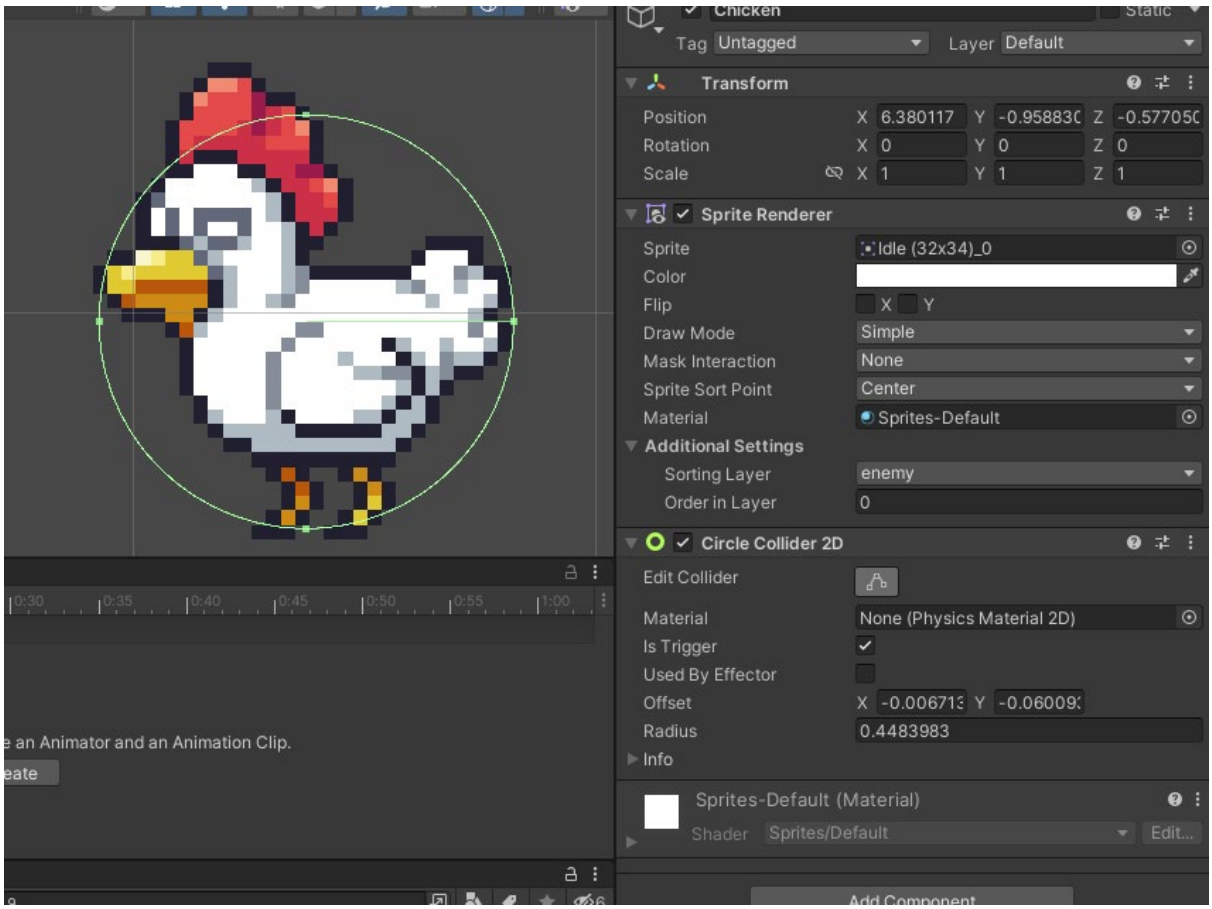

Рисунок 72. Изменение компонента “Circle Collider 2D”

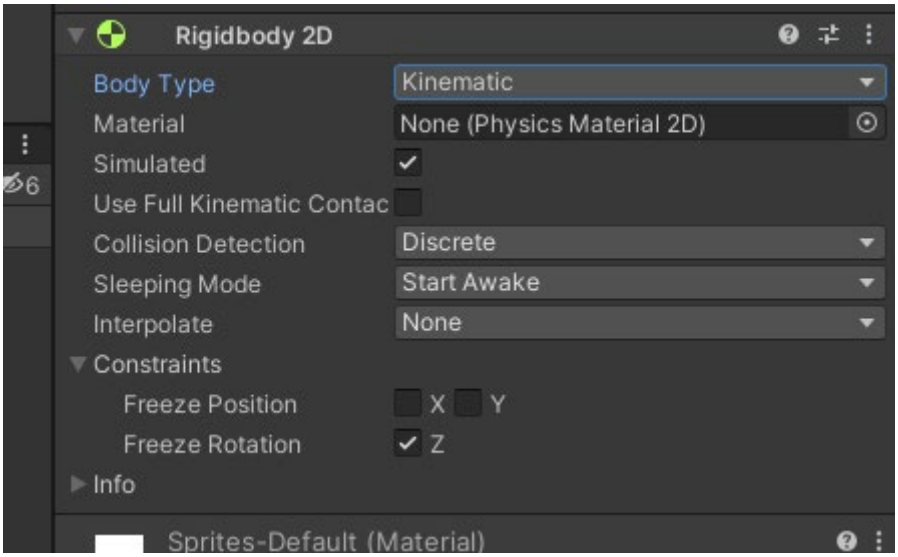

Рисунок 73. Изменение компонента “Rigidbody 2D”

Скрипт “EnemyChicken” должен иметь следующий код:

using UnityEngine;

public class EnemyChicken : MonoBehaviour

{

Создаем переменную, в которую будет положен компонент отвечающий за физику объекта

Rigidbody2D rb2d;

Создаем приватную переменную видную в юнити отвечающую за скорость передвижения

[SerializeField] float _speed = 2.0f;

Создаем приватную переменную, для смены движения

int dir = 1;

Создаем начальную среднюю точку для расчёта пути

Vector2 _startPoint;

Создаем приватную переменную видную в юнити, которая отвечает за дистанцию в обе стороны от начальной средней точки

[SerializeField] float maxRng = 2.0f;

Создаем приватную переменную, которая отвечает за направление персонажа

bool isRight = false;

Метод, активирующийся один раз в начале запуска игры

void Start()

{

Присвоение компонентов переменным

rb2d = GetComponent<Rigidbody2D>();

_startPoint = transform.position;

}

Метод, активирующийся с одной частотой

void FixedUpdate()

{

Создаем приватную переменную, для отображения центральной точки персонажа

float rng = Vector2.Distance(_startPoint, transform.position);

Если центральная точка больше разрешенной максимальной дистанции, значение переменной становится -1

if (rng > maxRng)

{

*dir *= -1;*

}

Изменяет направление движения в зависимости от переменной “dir”

*rb2d.velocity = Vector2.right * dir;*

}

Если переменная isRight верна и скорость персонажа по оси икс 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;
    }
```

Метод, который рисует в окне сцены два круга, обозначающих максимальную дистанцию

```
    private void OnDrawGizmos()
    {
```

```
            Gizmos.DrawWireSphere(_startPoint + new Vector2(maxRng, 0), 0.1f);

            Gizmos.DrawWireSphere(_startPoint + new Vector2(-maxRng, 0), 0.1f);

        }

        }
```

После чего разместить врага в необходимом месте, указать параметры в его скрипте и добавить анимацию с названием run как было продемонстрировано выше.