

Создание врага

Теперь пришло время сделать врагов, от которых нашему игроку предстоит уклоняться. Их поведение будет не очень сложным: толпы (мобы) будут появляться случайно по краям экрана, выбирать случайное направление и двигаться по прямой линии.

Мы создадим сцену `Mob`, которую затем сможем *инстанцировать*, чтобы создать любое количество независимых мобов в игре.

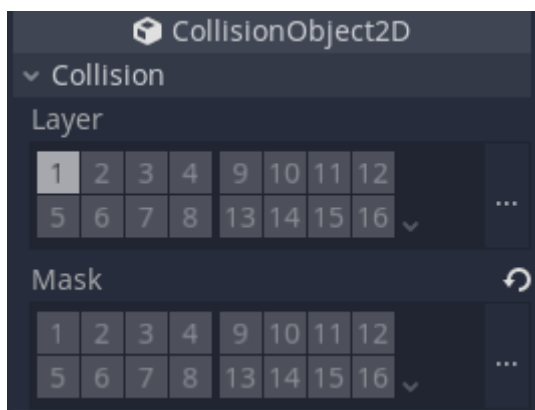
Настройка узла

Нажмите "Сцена" -> "Новая сцена" и добавьте следующие узлы:

- `RigidBody2D` (названный `Mob`)
 - Анимированный спрайт
 - `CollisionShape2D`
 - Уведомление о видимости2D

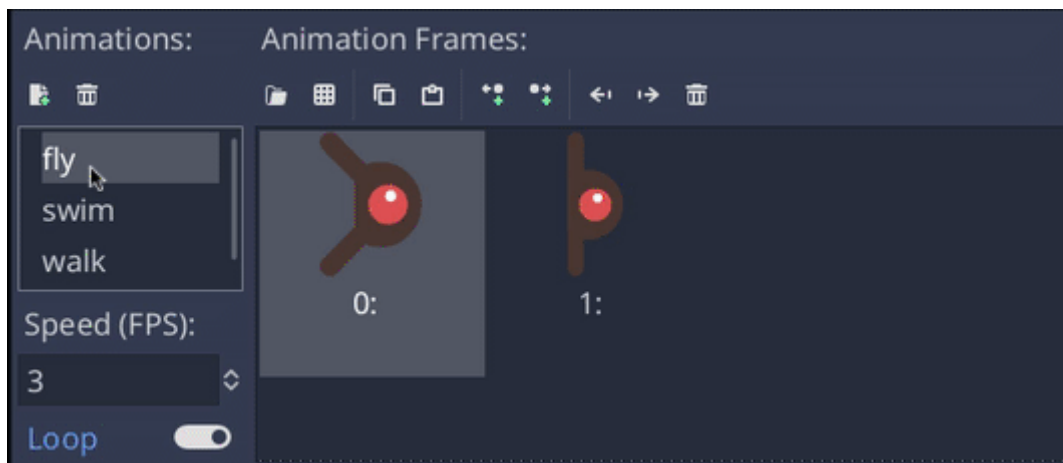
Не забудьте настроить дочерние узлы, чтобы они не могли быть выбраны, как вы это делали со сценой игрока.

В свойствах `RigidBody2D` установите в поле `Gravity Scale` значение `0`, чтобы моб не упал вниз. Кроме того, в разделе `PhysicsBody2D` выберите свойство `Mask` и снимите первый флажок. Это гарантирует, что мобы не столкнутся друг с другом.



Настройте `AnimatedSprite`, как вы сделали это для игрока. На этот раз у нас есть 3 анимации: `fly` (лететь), `swim` (плыть) и `walk` (ходить). В папке art есть две картинки для каждой анимации.

Установите "Speed (FPS)" на `3` для всех анимаций.



Установите свойство `Playing` в инспекторе на "Вкл".

Мы выберем одну из этих анимаций случайным образом, так что мобы будут иметь некоторое разнообразие.

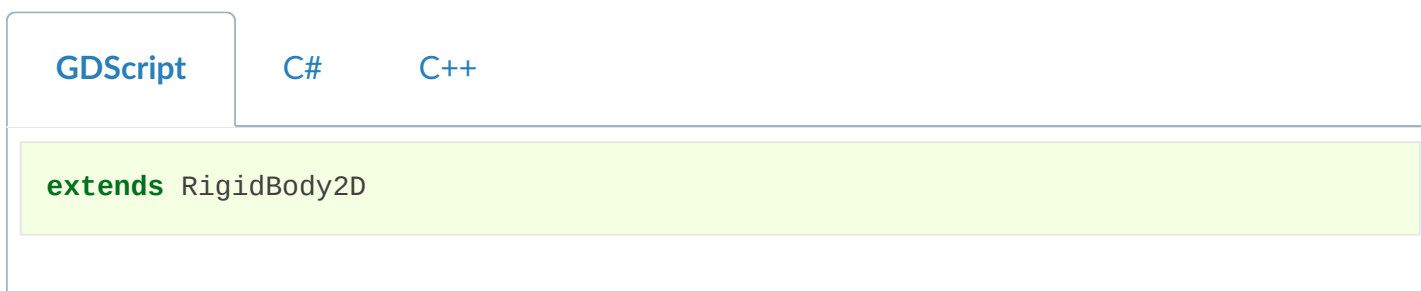
Также как изображения игрока, изображения мобов нужно уменьшить. Установите для `AnimatedSprite` в свойстве `Scale` значение `(0.75, 0.75)`.

Как и в сцене `Player`, добавьте форму `CapsuleShape2D` для расчета столкновений. Чтобы выровнять фигуру с изображением, вам нужно установить свойство `Rotation Degrees` на значение `90` (в разделе "Transform" в инспекторе).

Сохраните сцену.

Скрипт врага

Добавьте скрипт к `Mob` вот так:



Теперь давайте посмотрим на остальную часть скрипта. В `_ready()` мы проигрываем анимацию и случайным образом выбираем один из трех типов анимации:

GDScript

C#

C++

```
func _ready():
    $AnimatedSprite.playing = true
    var mob_types = $AnimatedSprite.frames.get_animation_names()
    $AnimatedSprite.animation = mob_types[randi() % mob_types.size()]
```

Во-первых, мы получаем список имен анимаций из `frames` свойства `AnimatedSprite`. Это возвращает массив, содержащий все три имени анимации: `["walk", "swim", "fly"]`

Дальше нам нужно выбрать случайное число между `0` и `2`, чтобы выбрать имя одной из анимаций из списка (индексы массива начинаются с `0`). Функция `randi() % n` выбирает случайное целое число между `0` и `n-1`.

Примечание

Вы должны использовать `randomize()`, если вы хотите, чтобы ваша последовательность "случайных" чисел отличалась при каждом запуске сцены. Мы будем использовать `randomize()` в нашей сцене `Main`, поэтому здесь она нам не понадобится.

Последняя часть - заставить мобов удалять себя, когда они покидают экран. Подключите сигнал `screen_exited()` узла `Visibility Notifier 2D` и добавьте этот код:

GDScript

C#

C++

```
func _on_VisibilityNotifier2D_screen_exited():
    queue_free()
```

На этом заканчиваем работу над сценой *Mob*.

Теперь, когда игрок и враги готовы, в следующей части мы перенесём их на новую сцену. Мы заставим врагов появляться случайно по границам экрана и двигаться вперёд, сделав наш проект играбельным.