

Part 5 - Ready Player One

By *Richard Davey* on 20th February 2018 @photonstorm

Temos algumas plataformas prontas para uso, mas ninguém para correr nelas. Vamos corrigir isso.

Crie uma nova variável chamada `player` e adicione o seguinte código à função `init`. Você pode encontrar em [part5.html](#) :

```
player = this.physics.add.sprite(100, 450, 'dude');

player.setBounce(0.2);
player.setCollideWorldBounds(true);

this.anims.create({
  key: 'left',
  frames: this.anims.generateFrameNumbers('dude', { start: 0, end: 3 }),
  frameRate: 10,
  repeat: -1
});

this.anims.create({
  key: 'turn',
  frames: [ { key: 'dude', frame: 4 } ],
  frameRate: 20
});

this.anims.create({
  key: 'right',
  frames: this.anims.generateFrameNumbers('dude', { start: 5, end: 8 }),
  frameRate: 10,
  repeat: -1
});
```

Há duas coisas diferentes acontecendo aqui: a criação de um Physics Sprite e a criação de algumas animações que ele pode utilizar.

Sprite com Física

A primeira parte do código cria o sprite:

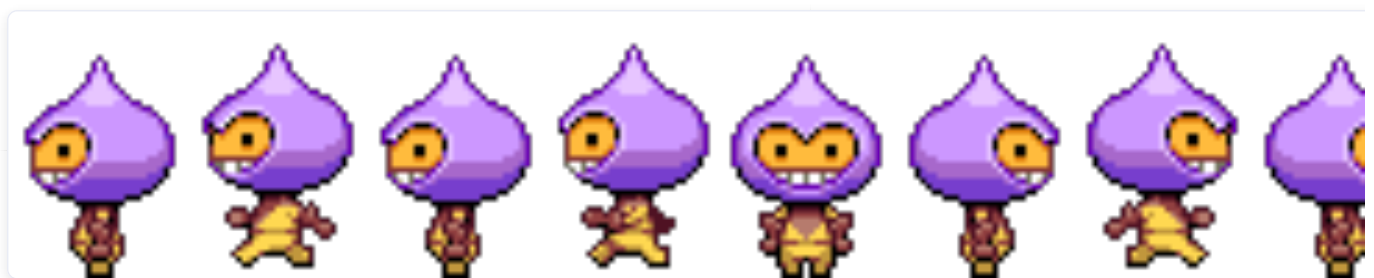
```
player = this.physics.add.sprite(100, 450, 'dude');  
  
player.setBounce(0.2);  
player.setCollideWorldBounds(true);
```

Isso cria um novo sprite chamado `player`, posicionado a 100 x 450 pixels da p do jogo. O sprite foi criado através do Factory Physics Game Object (`this.phy` que significa que ele possui um corpo de Física dinâmico por padrão.

Depois de criar o sprite, é dado um leve fator de ressalto de 0,2. Isto significa c aterrissar depois de pular, ele saltará muito levemente. O sprite é então definic com os limites do mundo do jogo. Os limites, por padrão, estão do lado de fora dimensões do jogo. Como definimos o jogo como sendo 800 x 600, o jogador correr fora desta área. Isso impedirá que o player saia das bordas da tela ou sa superior.

Animações

Voltando sua atenção para a função `preload`, verá que o personagem 'dude' fo como uma sprite sheet, não uma imagem. Isso é porque contém quadros de a assim que o sprite sheet completo se parece:



Há 9 quadros no total, 4 para correr à esquerda, 1 para olhar para a câmera e 4 direita. Nota: O Phaser suporta virar os sprites para salvar quadros de animaçã este tutorial, vamos mantê-lo no modo antigo.

Nós definimos duas animações chamadas 'left' e 'right'. Aqui está a animação d

```
this.anims.create({
```

```
key: 'left',  
frames: this.anims.generateFrameNumbers('dude', { start: 0, end: 3 } ),  
frameRate: 10,  
repeat: -1  
});
```

A animação 'left' usa quadros 0, 1, 2 e 3 e roda a 10 quadros por segundo. O valor diz para a animação fazer um loop.

Este é o nosso ciclo de corrida padrão e o repetimos para correr na direção oposta a tecla 'direita'.

Informação Extra: No Phaser 3, o Animation Manager é um sistema global. As animações criadas dentro dele estão disponíveis globalmente para todos os Objetos do Jogo. Todos os Objetos compartilham os dados básicos de animação enquanto gerenciam suas próprias animações. Isso permite definir uma única animação uma vez e aplicá-la a quantos Objetos forem necessários. Isso é diferente do Phaser 2, no qual as animações pertenciam especificamente aos Objetos em que foram criados.