01

Barra de vida

Transcrição

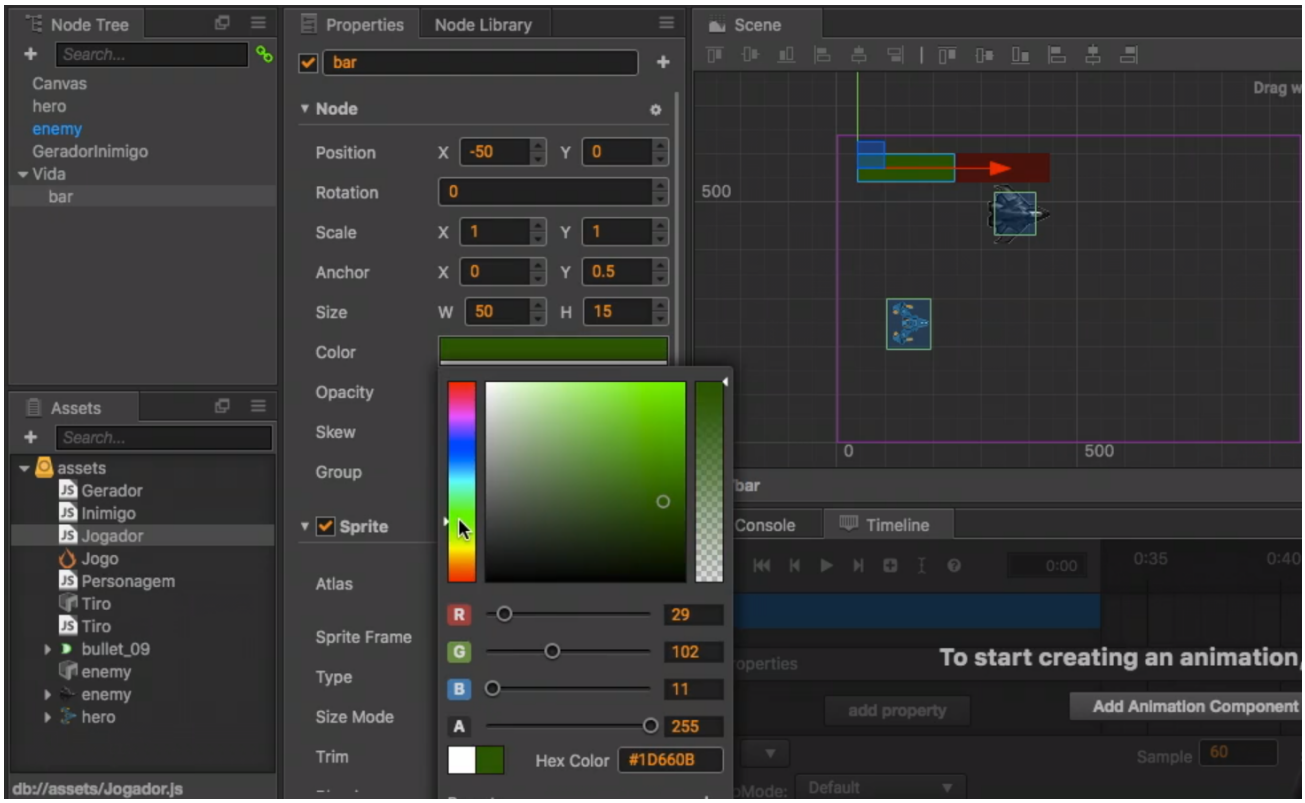
Nesta aula trabalharemos nos detalhes mais visuais do nosso jogo. O jogador, por exemplo, ao ser atingido por um tiro, recebe seu dano, mas não temos nenhuma indicação visual disso.

Como vamos trabalhar com recursos visuais, a Cocos disponibiliza alguns objetos que facilitam todo o processo. Na aba `Node Library` temos vários objetos dos quais podemos fazer uso. Um deles é o `ProgressBar` que pode ser usado como a barra de vida do jogador.

Para utilizá-la, basta a arrastarmos para a cena do jogo. Note que ela é dividida em duas cores, uma branca e a outra cinza. Este objeto na verdade é composto por dois componentes, um deles é o plano de fundo cinza e o outro é a parte branca que aumenta ou diminui seu tamanho de acordo com o controlador `progress` em suas propriedades.

Na aba `Node Tree` e possível ver que o objeto `ProgressBar` - que passaremos a chamar de `Vida` - possui esse outro componente interno chamado vida, basta clicar na seta ao lado do nome do objeto. Nossa primeira configuração para

este componente é fazer com que o elemento pai tenha uma cor vermelha e o elemento filho `bar` tenha a cor verde. Para isso basta clicar no campo `Color`, na aba `Properties`, do objeto e de seu filho.

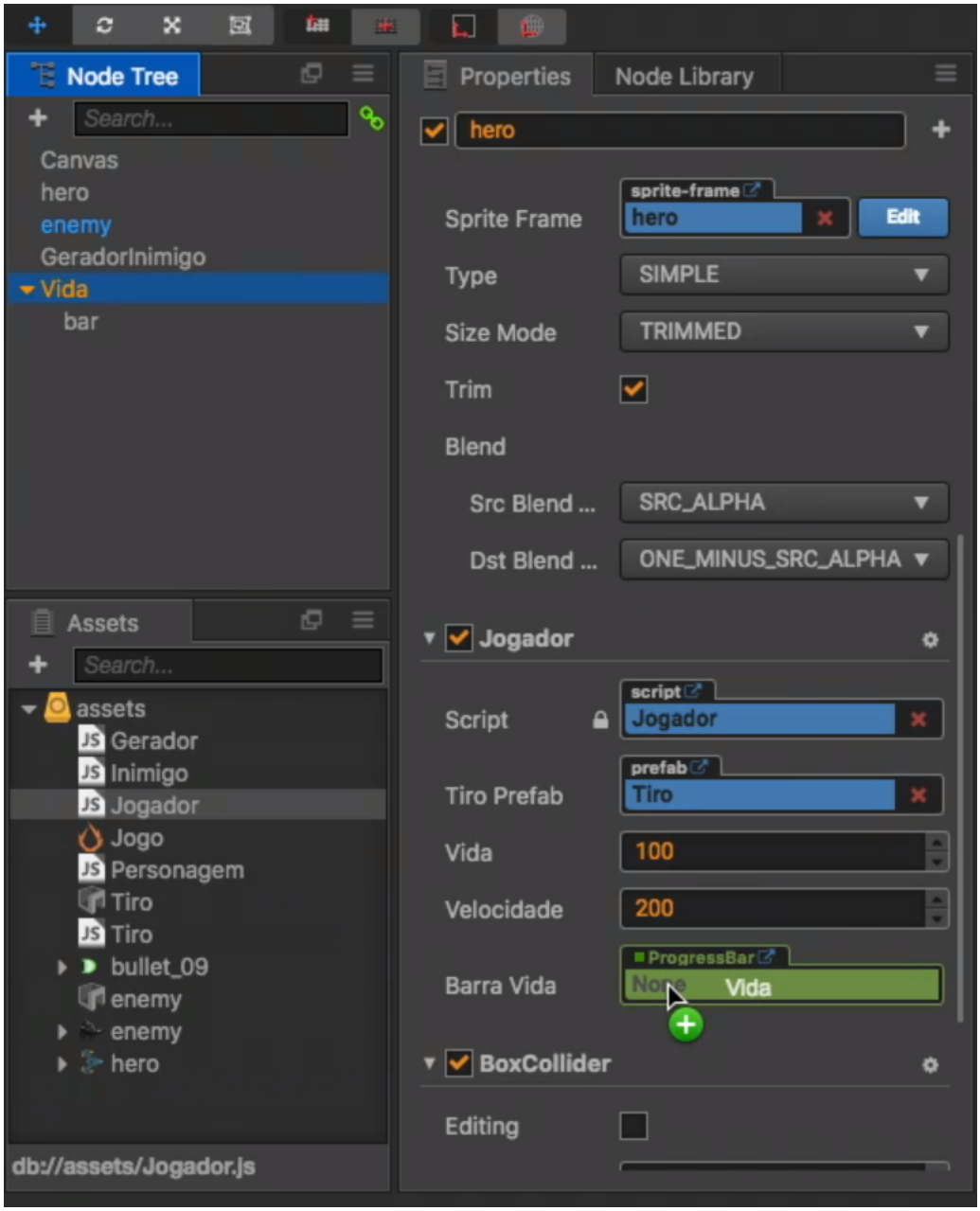


A ideia agora é que a cada dano recebido por nosso jogador, tenhamos um *feedback* visual informando quanto de vida o jogador ainda tem. Porém, nós apenas adicionamos a barra de vida ao jogo, não há nenhuma associação com o jogador ainda.

Para associar a barra de vida ao jogador, criaremos uma nova propriedade pública no `Jogador.js` que será do tipo `ProgressBar` e a chamaremos de `barraVida`.

```
properties: {
    _acelerando: false,
    velocidade: 10,
    vida: 100,
    barraVida: cc.ProgressBar,
},
```

Com isso, podemos ir na aba `Properties` do objeto `hero` e ver que temos um novo campo chamado `Barra Vida`, onde podemos simplesmente arrastar e soltar o objeto `Vida` que está na `Node Tree` para este campo. Assim os dois estarão associados.



Já temos uma propriedade chamada `vida` que tem o valor de `100`. Como este valor representa o máximo de vida, renomearemos esta propriedade para `vidaMaxima`, assim não confundiremos com a propriedade que criaremos agora para manter o registro de quanto de vida ainda sobra para o jogador, esta propriedade terá valor inicial zero e se chamará `vidaAtual`, sendo ela também *privada*.

```
properties: {
    _acelerando: false,
    _vidaAtual: 0,
    vidaMaxima: 100,
    velocidade: 10,
    barraVida: cc.ProgressBar,
},
```

A `vidaAtual` do jogador não fará muito sentido se, ao iniciar o jogo, ela continuar valendo zero. Caso fosse, o jogador já nasceria morto. O fato é que o jogador só nasce de fato no método `onLoad` e será nele onde diremos que a `vidaAtual` é igual a `vidaMaxima`. Outra coisa que precisamos configurar no método `onLoad` é a porcentagem de preenchimento inicial da nossa barra de vida, que no caso será de '100%', visto que não tomamos nenhum dano ainda.

```
onLoad: function () {
    this._vidaAtual = this.vidaMaxima;
    this.barraVida.progress = 1;
    // restante do código
},
```

Um detalhe importante é que a propriedade `progress` da nossa barra de vida trabalha com valores entre zero e um, que representam uma porcentagem de quanto preenchida ela está.

Para finalizar, precisamos calcular o dano recebido pelo jogador e atualizar a barra de vida de acordo. Para obtermos a porcentagem de vida atual, precisamos dividir o valor da vida atual pela vida máxima. Lembrando que a subtração do dano precisa continuar sendo feita. Assim teremos em nosso método `tomarDano` .

```
tomarDano: function(dano){
    this._vidaAtual -= dano;
    let porcentagemVida = this._vidaAtual / this.vidaMaxima;
    this.barraVida.progress = porcentagemVida;
},
```

Já podemos testar e verificar que a cada tiro recebido pelo jogador, temos nossa barra de vida sendo alterada visualmente.