

10

## Polimento do jogo

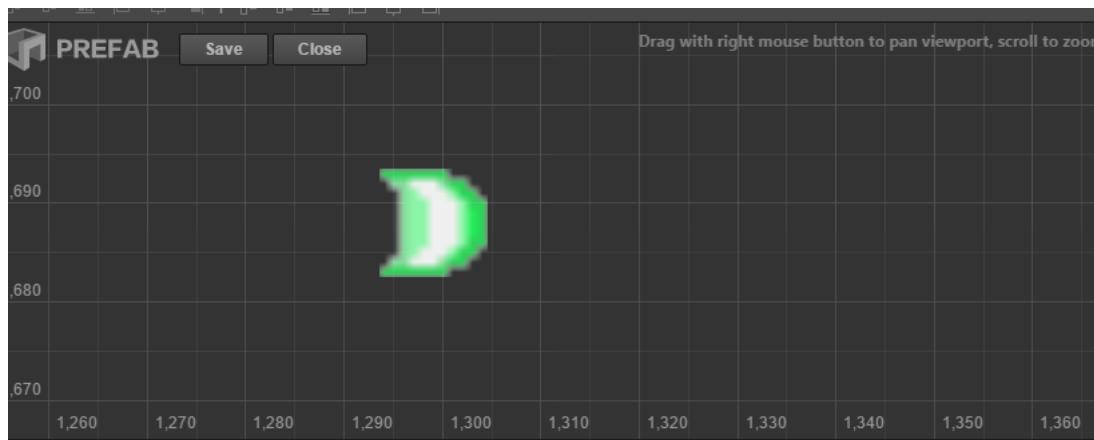
Durante o desenvolvimento de um jogo uma das áreas onde passamos mais tempo trabalhando é no polimento dele. Muitas vezes é mais rápido fazer grandes avanços no jogo como um todo do que ajustar o balanceamento de uma única fase ou escolher quanto de vida um inimigo deve ter.

A etapa de polimento do jogo é extremamente importante e faz muita diferença para o produto final. No nosso caso temos muitas ferramentas que criamos para que o *game designer* possa montar e ajustar o jogo da maneira que ele quiser. Mas podemos ir um passo além e fazer com que nosso tiro rotacione na direção que ele está seguindo.

Parece pouco mas no produto final o jogador irá reparar nesse tipo de detalhe. Por mais que ele não saiba o porquê de uma versão do jogo parecer melhor do que a outra ele vai sentir isso.

E por que isso faz diferença no nosso tiro?

Porque se você reparar nosso tiro não é redondo. Ele tem um formato levemente ovalado, o que indica que ele tem um sentido.



Com isso se rotacionarmos ele na direção em que ele está andando, vamos ter uma diferença no visual dele. Para rotacionarmos o tiro na direção que ele está andando precisamos descobrir o ângulo que devemos colocar na propriedade `rotation` do componente **Node**.

Para descobrirmos essa rotação podemos pegar o vetor direção e passar os valores dele para o método `Math.atan2` que irá nos retornar um ângulo em radianos. Como a Cocos trabalha com ângulos em graus, precisamos chamar o método `cc.radiansToDegrees` para converter nosso ângulo.

Além disso precisamos alinhar o eixo de rotação das bibliotecas de matemática do JavaScript com o eixo de rotação da Cocos. Isso porque na Cocos o 90° aponta para baixo e não para cima. Para fazer essa inversão precisamos apenas multiplicar nosso ângulo por -1.

No final só precisamos aplicar esse ângulo à propriedade `rotation` com componente **Node**

```
inicializa: function (pai, posicao, direcao) {
    this.node.parent = pai;
    this.node.position = posicao;
    this._movimentacao.setDirecao(direcao);
```

```
this.rotacionar(direcao);  
},  
  
rotacionar : function(direcao){  
    let angulo = Math.atan2(direcao.y, direcao.x);  
    angulo = cc.radiansToDegrees(angulo) * -1;  
    this.node.rotation = angulo;  
},
```