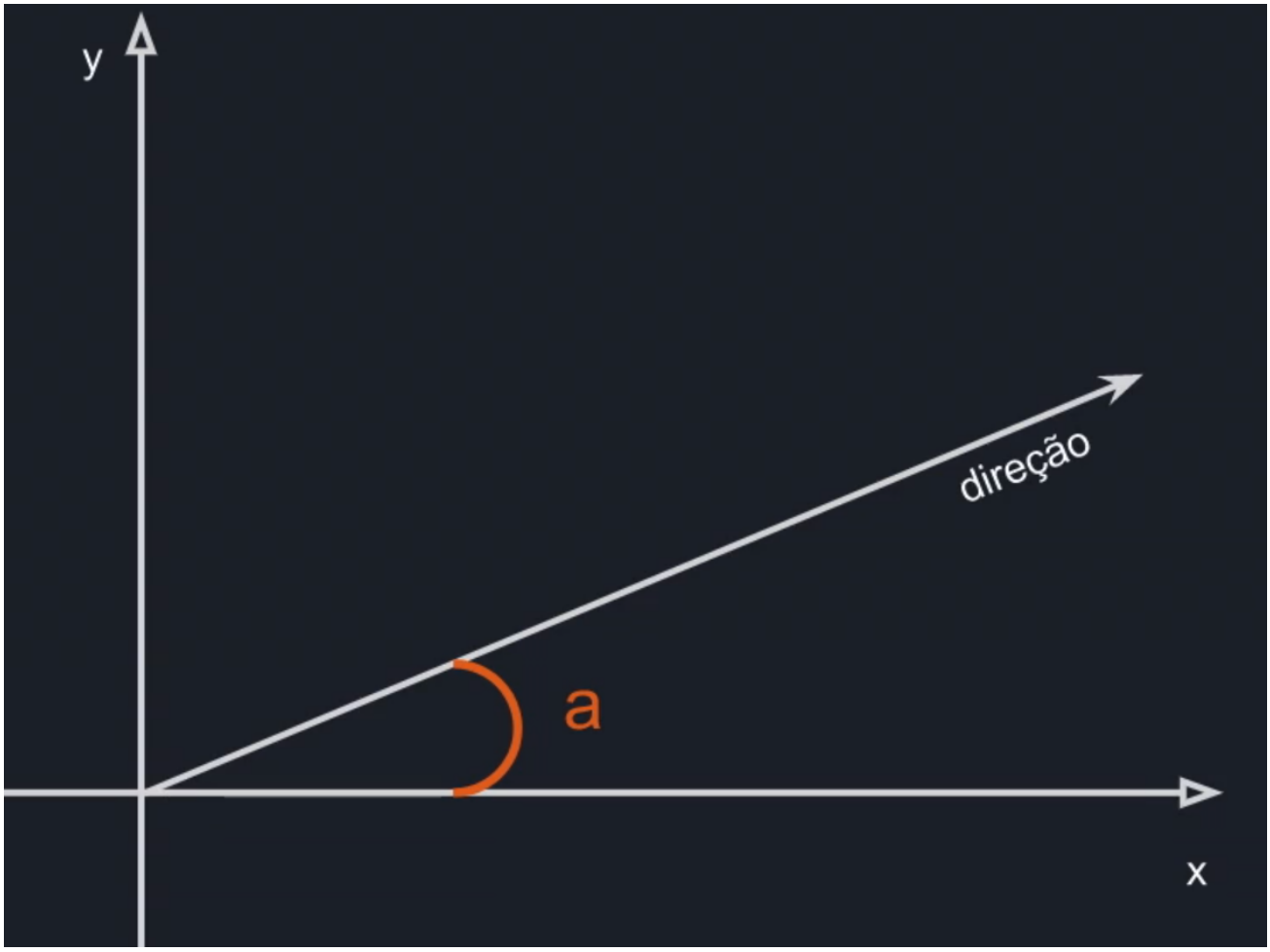
05

Rotação do Jogador e Inimigos

Transcrição

Um detalhe ao qual não demos muita importância no começo mas que vai deixar nosso jogo ainda mais legal e polido é o problema da rotação dos objetos. Note que as naves inimigas e a do jogador se movimentam em uma direção, mas elas não rotacionam. Isso passa a impressão de que as vezes a nave não faz uma curva, ela anda de lado ou para trás. Isso quando não está indo para frente.

Voltando a considerar o plano cartesiano, temos os eixos X e Y e o vetor direção. Isso era o que tínhamos que saber para poder movimentar a nave: O vetor direção. O que precisamos encontrar agora é o ângulo formado entre este vetor e o eixo X, o qual chamaremos por enquanto de `a`.



Esse não é um problema novo. Na matemática ele é conhecido como ângulo arco-tangente e pode ser descoberto usando o método `atan2` da classe `Math` no *JavaScript*, informando para ela o valor da direção `Y` e `X` nessa origem, apesar de parecer estranho.

O primeiro passo é obter o ângulo entre o eixo `X` e o vetor direção. Faremos isso no método `mudarDirecao` do `Jogador.js` primeiro.

```
let angulo = Math.atan2(direcao.y, direcao.x);
```

O segundo é converter esse valor para graus. Isto por que o método `atan2` retorna esse valor do ângulo em radianos. Para converter para graus basta multiplicarmos esse valor pela divisão de `180` por `PI` . Assim teremos:

```
let angulo = Math.atan2(direcao.y, direcao.x);
angulo = angulo * (180 / Math.PI);
```

O terceiro passo é inverter esse valor. Isso porque o eixo de rotação da *Cocos* é inverso ao eixo padrão da matemática. Isso faz com que o grau `90` seja direcionado para baixo, enquanto o `270` seja para cima.

```
let angulo = Math.atan2(direcao.y, direcao.x);
angulo = -angulo * (180 / Math.PI);
```

Por último, precisamos atribuir esse valor a propriedade `rotation` do componente `node` da nave. Todo componente `node` possui uma propriedade chamada `rotation` que configura exatamente a rotação do elemento na cena. Ao final teremos:

```
mudarDirecao: function(event){
    let posicaoMouse = event.getLocation();
    posicaoMouse = new cc.Vec2(posicaoMouse.x, posicaoMouse.y);

    let direcao = posicaoMouse.sub(this.node.position);
    direcao = direcao.normalize();

    this._direcao = direcao;

    let angulo = Math.atan2(direcao.y, direcao.x);
    this.node.rotation = -angulo * (180 / Math.PI);
},
```

Como nossa nave segue a posição do mouse. Teremos a nave também girando em relação a essa posição.

Faremos a mesma coisa no método `mudarDirecao` do `Inimigo.js` .

```
mudarDirecao: function(){
    let direcao = this._alvo.position.sub(this.node.position);
    direcao = direcao.normalize();
    this._direcao = direcao;

    let angulo = Math.atan2(direcao.y, direcao.x);
    this.node.rotation = -angulo * (180 / Math.PI);
},
```

Agora os inimigos serão rotacionados de acordo com a posição da nave do jogador.