Laboratório de Produção de Jogos 3 - Física e Simuladores

Objetivo: Neste laboratório iremos implementar o ataque aos inimigos spawnados. Vamos mirar e movimentar a "cabeça" da torre para a posição onde o alvo está. Vamos também iniciar a lógica de disparo de projéteis.

Atividade 1

Tempo entre Waves (DESAFIO) Coloque um componente de Texto no cenário e faça ele receber o contador de tempo entre waves.

Atividade 2

Definindo uma Torre - Pt1 Primeiramente vamos fazer o import de um modelo de torre. Você pode usar o exemplo que está no canvas ou fazer seu próprio. Após importar:

1. Defina o scale apropriado, e as cores conforme seu gosto.
2. Altere o nomes dos componentes do modelo para que o componente de cima se chame "Head" e o debaixo se chame "Base".
3. Crie uma prefab dessa torre.
4. Em seguida, adicione uma torre no cenário.

Atividade 3

Definindo uma Torre - Pt2 Agora vamos criar o mecanismo para fazer a rotação da Head quando houver um alvo em movimento. Passo a passo:

1. Crie um Game Object filho da prefab Torre chamado Engrenagem
2. Posicione a Engrenagem no meio do modelo e entre a Head e a Base. Dica: utilize a visualização Wireframe no Shading Mode.
3. Defina a Head como filha de Engrenagem
4. Garanta que o eixo Y da engrenagem esteja na mesma direção que o eixo Y da Head.
5. Crie o script Metralhadora e adicione o seguinte código:

```
1   public class Metralhadora : MonoBehaviour
2   {
3       /* Temos dois objetivos com esse script:
4           1) Queremos encontrar o alvo mais proximo da torre e mirar nele
5           2) Queremos movimentar a Head da torre para seguir o movimento do
                alvo */
6
7       // Guarda a posicao do alvo atual
8       [SerializeField] private Transform target;
```

```
9
10          // Define a distancia maxima para a mira
11          [SerializeField] private float range = 15f;
12
13          // Essa funcao nos permite visualizar a distancia maxima do tiro
14          private void OnDrawGizmosSelected()
15          {
16              Gizmos.color = Color.red;
17              Gizmos.DrawWireSphere(transform.position, range);
18          }
19      }
```

6. Adicione o script na torre.

7. Teste e veja se está tudo ok. Você deverá observar uma esfera desenhada ao redor da metralhadora com raio igual ao *range* especificado. Ou seja, essa é a área de cobertura da metralahdora. Note que o Gizmo deve estar habilitado para você visualizar a esfera.

Atividade 4

Definindo uma Torre - Pt3 Em seguida, vamos utilizar a área de cobertura da torre para mirar em inimigos. Passo a passo:

1. Adicione o seguinte código no script Metralhadora:

```
1       public class Metralhadora : MonoBehaviour
2       {
3           /* Definicoes dos atributos */
4
5           [SerializeField] private string enemyTag = "inimigo";
6
7           private void Start()
8           {
9               // Vamos chamar o metodo de encontrar alvo assim que entrarmos no
                    Start e apos isso, a cada meio segundo --> dessa forma nao
                    sobrecarregamos o Update
10              InvokeRepeating("UpdateTarget", 0f, 0.5f);
11          }
12
13          private void UpdateTarget()
14          {
15              // Encontra inimigos spawnados na cena
16              GameObject[] inimigos = GameObject.FindGameObjectsWithTag(enemyTag);
17
18              // Variaveis para armazenar informacoes do inimigo mais proximo
19              GameObject inimigoMaisProximo = null;
20              float distanciaInimigoMaisProximo = Mathf.Infinity;
21
22              // Faz loop no vetor de inimigos e descobre o inimigo mais proximo
23              foreach (GameObject inimigo in inimigos)
24              {
25                  float distanciaAteInimigo = Vector3.Distance(transform.position,
                        inimigo.transform.position);
26
```

```
27                 if (distanciaAteInimigo < distanciaInimigoMaisProximo)
28                 {
29                     distanciaInimigoMaisProximo = distanciaAteInimigo;
30                     inimigoMaisProximo = inimigo;
31                 }
32             }
33 
34             // Verifica se o inimigo mais proximo esta dentro da area de cobertura
35             if (inimigoMaisProximo != null && distanciaInimigoMaisProximo < range)
36             {
37                 target = inimigoMaisProximo.transform;
38             }
39             else
40             {
41                 target = null; // limpa o alvo caso nao encontre um alvo valido
42             }
43 
44         }
45 
46         /* Demais definicoes dos metodos */
47     }
```

2. Teste e veja se está tudo ok. Você deverá ver o atributo de alvo no Inspector sendo atualizado.

Atividade 5

Definindo uma Torre - Pt4 Uma vez que temos o alvo, vamos rotacionar a Head da torre para estar de acordo com a posição do alvo. Passo a passo:

1. Adicione o seguinte código no script Metralhadora:

```
1     public class Metralhadora : MonoBehaviour
2     {
3          /* Definicoes dos atributos */
4 
5        [SerializeField] private Transform engrenagem;
6 
7         private void Update()
8         {
9             // Executa a logica somente se tivermos um alvo
10            if (target != null)
11            {
12                Vector3 direcaoParaMirar = target.position - transform.position;
13 
14                // Pega a rotacao necessaria para virar para posicao do alvo
15                Quaternion rotacaoNecessariaParaVirar =
                      Quaternion.LookRotation(direcaoParaMirar);
16                // Faz o calculo do vetor de rotacao
17                Vector3 rotacaoParaMirar = rotacaoNecessariaParaVirar.eulerAngles;
18                // Define a rotacao da engrenagem para este vetor de rotacao
19                engrenagem.rotation = Quaternion.Euler(0f, rotacaoParaMirar.y,
                      0f);
20            }
21        }
```

```
22
23          /* Demais definicoes dos metodos */
24      }
```

2. Teste e veja se está tudo ok. Você deverá ver a Head da torre sendo atualizada.

Atividade 6

Definindo uma Torre - Pt5 (DESAFIO) Defina uma prefab para o projétil. Defina um ponto de spawn em frente ao cano da metralhadora. Faça o spawn dos projéteis de tempo em tempo. Adicione efeitos de som no momento do disparo.