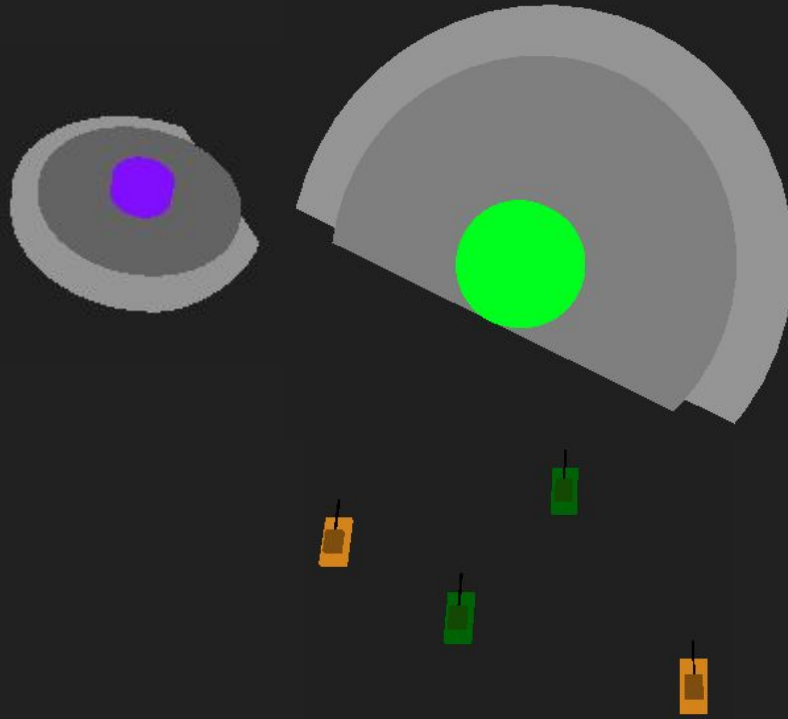


CG_N4: Space Defenders

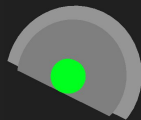


Alunos: Célio Rodrigues Junior e Henrique José Wilbert

Introdução ao Tema

- Jogo simples em 3D parecido com space invaders, porém com os times invertidos

- O jogador controla uma nave  e tem que defender a nave-mãe caída dos tanques que avançam em sua direção



- A nave pode disparar um Laser para destruir os tanques.



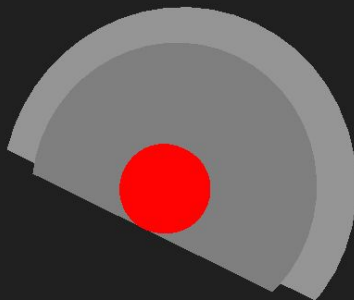
- Tanques especiais  precisam de três tiros de laser para serem destruídos

Introdução ao Tema

- Um laser especial pode ser utilizado a partir da nave-mãe caída que destrói vários tanques próximos



- A vida total das naves é demonstrada pela variação da cor do cockpit da nave-mãe principal, quando chegar a vermelho as naves-mãe são destruídas



Controles

Setas do teclado controlam a nave em X e Z (Y não muda)

Espaço atira o laser

WASD deslocam a câmera para a direita/esquerda/Frente/Atrás

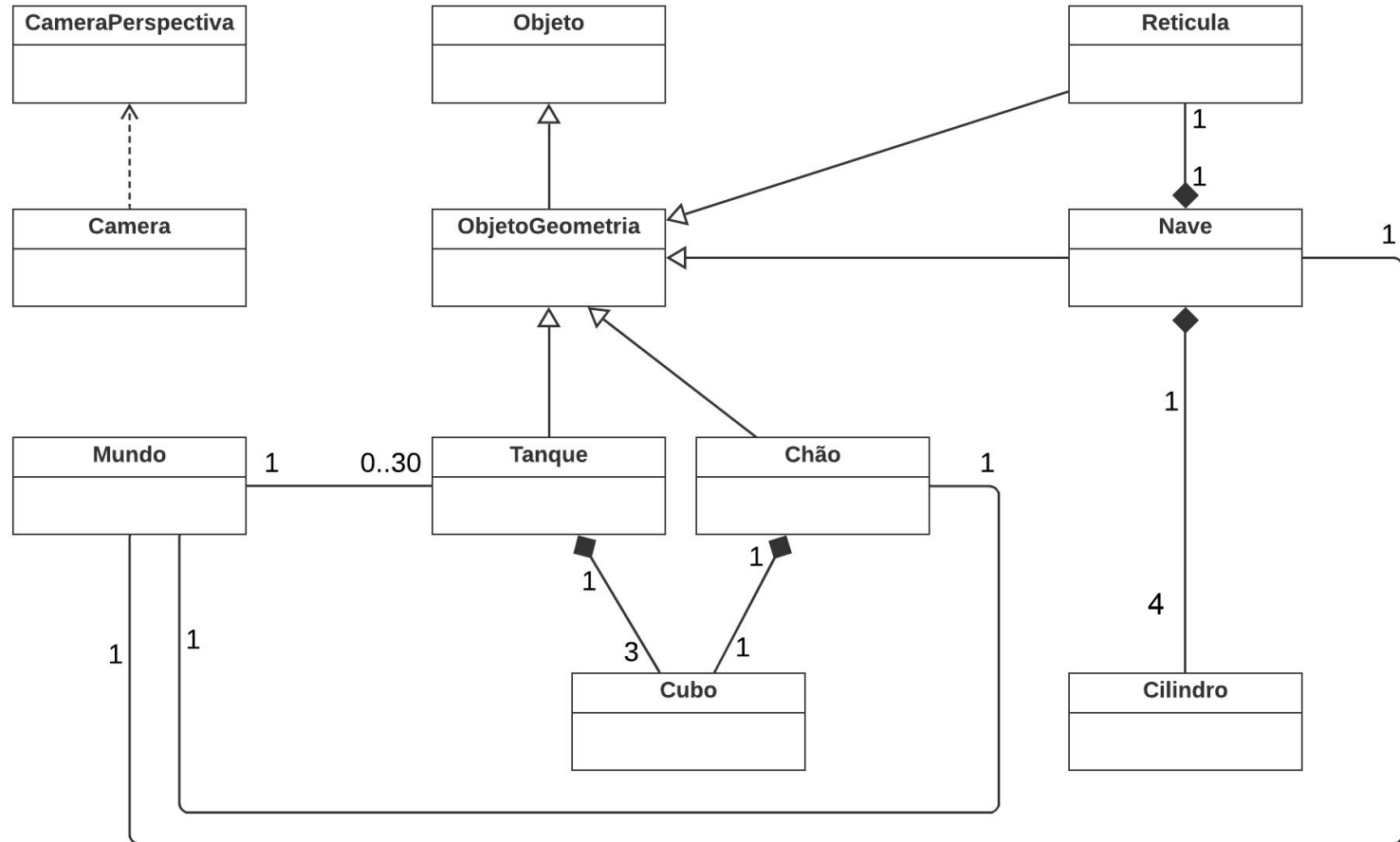
R ativa um poder que destrói vários tanques perto da nave-mãe



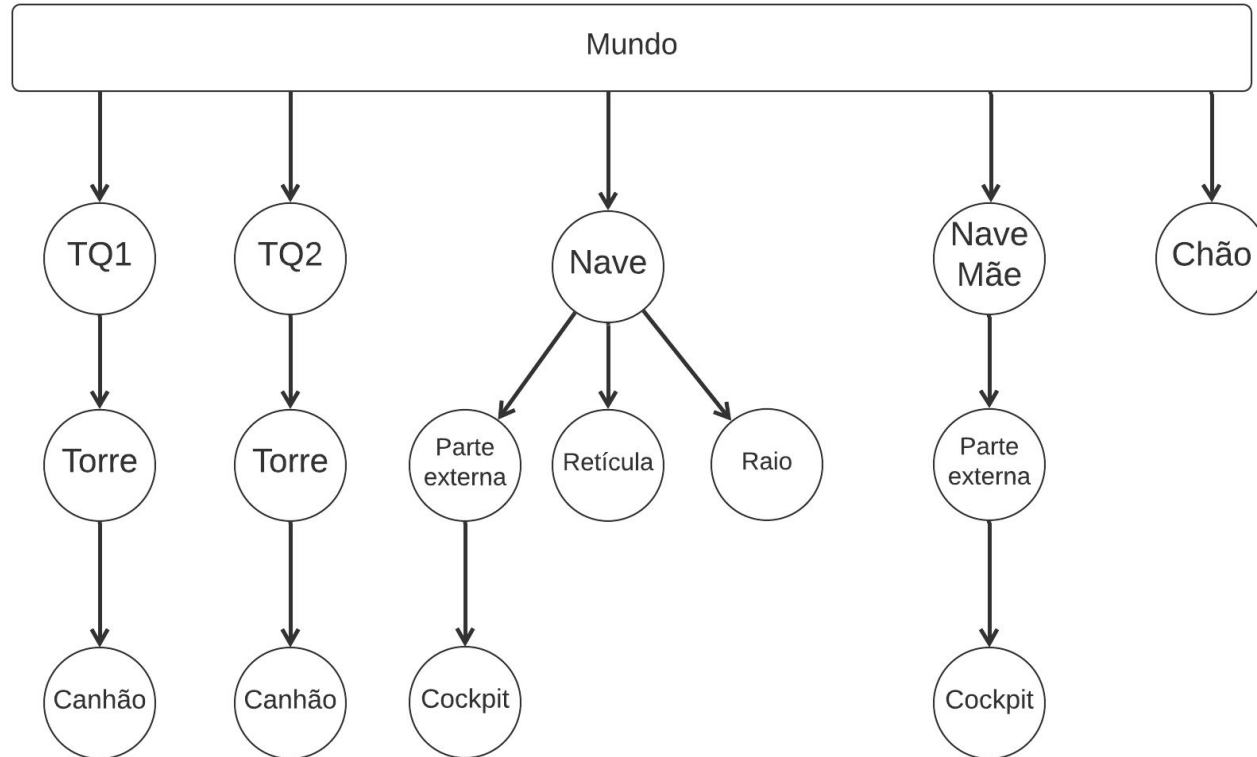
Demonstração



Diagrama de Classes



Grafo de Cena



Trechos relevantes: Movimentação da câmera

```
34 public void deslocarEsquerda()  
35 {  
36  
37     if (eye.X - 10 > 0)  
38     {  
39         eye.X -=10;  
40         at.X -=10;  
41     }  
42  
43 }  
44
```

```
45 public void deslocarDireita()  
46 {  
47  
48     if (eye.X + 10 < 150)  
49     {  
50         eye.X +=10;  
51         at.X +=10;  
52     }  
53 }  
54
```



Trechos relevantes: Geração dos tanques

```
64      System.Timers.Timer geradorT = new System.Timers.Timer();
65      geradorT.Elapsed += new ElapsedEventHandler(gerarT);
66      geradorT.Interval = 1500;
67      geradorT.Enabled = true;
68
```

```
440     private void gerarT(object source, ElapsedEventArgs e)
441     {
442         if (tanques.Count < 30)
443         {
444             float novaPosicao = 0.0f;
445             int posicao = 0;
446
447             do
448             {
449                 Random r = new Random();
450                 posicao = r.Next(1, 50);
451                 novaPosicao = (posicao * 3f) - 1.2f;
452             } while (posicoes.Contains(posicao) && posicoes.Count < 30);
453             posicoes.Add(posicao);

```

Trechos relevantes: Geração dos tanques

```
458      objetoId = Utilitario.charProximo(objetoId);
459      TQ obj_TQ = new TQ('a', null, new Ponto4D(novaPosicao, 0.0f, 70.0f), true);
460      obj_TQ.Posicao = posicao;
461      objetosLista.Add(obj_TQ);
462      tanques.Add(obj_TQ);
463      obj_TQ.RotacaoZBBox(180, 'y');
```

Trechos relevantes: Tanques

```
47 base.PontosAdicionar(new Ponto4D(x - 1.2, y - 0.75, z + 2.5));
48 base.PontosAdicionar(new Ponto4D(x + 1.2, y - 0.75, z + 2.5));
49 base.PontosAdicionar(new Ponto4D(x + 1.2, y + 0.75, z + 2.5));
50 base.PontosAdicionar(new Ponto4D(x - 1.2, y + 0.75, z + 2.5));
51 base.PontosAdicionar(new Ponto4D(x - 1.2, y - 0.75, z - 2.0));
52 base.PontosAdicionar(new Ponto4D(x + 1.2, y - 0.75, z - 2.0));
53 base.PontosAdicionar(new Ponto4D(x + 1.2, y + 1, z - 2.0)); //
54 base.PontosAdicionar(new Ponto4D(x - 1.2, y + 1, z - 2.0)); //
```

```
59 base.FilhoAdicionar(new TQ(rotulo, paiRef, centro, 1, tqEspecial));
```

Trechos relevantes: Movimentação dos tanques

```
69      System.Timers.Timer movedorT = new System.Timers.Timer();
70      movedorT.Elapsed += new ElapsedEventHandler(moverT);
71      movedorT.Interval = 150;
72      movedorT.Enabled = true;
```

```
482      private void moverT(object source, ElapsedEventArgs e)
483      {
484          foreach (TQ tanque in tanques)
485          {
486              if (!(tanque.BBox.obterCentro.Z < 0))
487              {
488                  tanque.TranslacaoXYZ(0, 0, -0.5);
489                  ObjetoGeometria tq = (ObjetoGeometria)tanque;
490                  tq.atualizaBBoxTransformacao();
491              }
492              else
493              {
494                  tanque.mirar(centros);
495              }
496          }
497      }
498  }
499
500 }
```

Trechos relevantes: Tanques Mirando

```
200 //gira pra esquerda
201 if (centros[indiceMaisProx].X < this.BBox.centro.X)
202 {
203     base.ObjetosLista[0].RotacaoZBBox(Math.Abs(centros[indiceMaisProx].X - this.BBox.centro.X) * 1.5, 'y');
204 }
205 else
206 { //gira pra direita
207     base.ObjetosLista[0].RotacaoZBBox(-Math.Abs(centros[indiceMaisProx].X - this.BBox.centro.X) * 1.5, 'y');
208 }
```



Trechos relevantes: Geração das Naves

```
37     this.cor = cor;
38     this.centro = centro;
39     criaCilindro(qualidade, raio, altura, centro);
40     base.FilhoAdicionar(new Nave(rotulo, this, qualidade, raio, altura, new Cor(150, 150, 150), centro, 1));
41     base.FilhoAdicionar(new Nave(rotulo, this, qualidade, raio, altura, new Cor(196, 255, 255), centro, 2));
42     base.FilhoAdicionar(new Reticula(rotulo, this, centro));
43     this.raio = new Nave(rotulo, this, 50, 0.1, 100, new Cor(255, 255, 255), centro, 3);
```



Trechos relevantes: Geração das Naves

```
74 public Nave(char rotulo, Objeto paiRef, int qualidade, double raio, double altura, Cor cor, Ponto4D centro, int parte)
75 {
76     this.cor = cor;
77     switch (parte)
78     {
79         case (1):
80             criaCilindro(qualidade, raio * 1.25, altura * 0.75, centro);
81             break;
82         case (2):
83             criaCilindro(qualidade, raio * 0.3, altura * 1.75, centro);
84             break;
85         case (3):
86             criaCilindro(qualidade, 0.1, -50, centro);
87             break;
88     }
89 }
90 }
```



Trechos relevantes: Laser da Nave

```
64 public Ponto4D atirar()
65 {
66     if(trava == false){
67         trava = true;
68         base.FilhoAdicionar(raio);
69         return new Ponto4D(this.BBox.obterCentro.X,this.BBox.obterCentro.Y,this.BBox.obterCentro.Z);
70     }
71     else return null;
72 }
```



Trechos relevantes:

Atualização da BBox com base na matriz de transformação (alteração em ObjetoGeometria)

```
32 public void atualizaBBoxTransformacao()
33 {
34     this.BBox = new BBox();
35
36     int i = 0;
37     foreach (Ponto4D pto in pontosLista)
38     {
39         double x = pto.X;
40         double y = pto.Y;
41         double z = pto.Z;
42
43         double x1 = ((x * base.Matriz.ObterElemento(0)) + (y * base.Matriz.ObterElemento(4)) + (z * base.Matriz.ObterElemento(8)) + (
44         double y1 = ((x * base.Matriz.ObterElemento(1)) + (y * base.Matriz.ObterElemento(5)) + (z * base.Matriz.ObterElemento(9)) + (
45         double z1 = ((x * base.Matriz.ObterElemento(2)) + (y * base.Matriz.ObterElemento(6)) + (z * base.Matriz.ObterElemento(10)) + (
46
47         if(i == 0){
48             BBox.Atribuir(new Ponto4D(x1,y1,z1));
49         }else{
50             BBox.Atualizar(new Ponto4D(x1,y1,z1));
51         }
52         i++;
53     }
54     BBox.ProcessarCentro();
```

Trechos relevantes: Hitbox

```
294     else if (e.Key == Key.Space)
295     {
296         Ponto4D xz = nave.atirar();
297         if (xz != null)
298         {
299
300             TQ tanqueRemover = null;
301             foreach (TQ tanque in tanques)
302             {
303                 if (xz.X >= tanque.BBox.obterMenorX &&
304                     xz.X <= tanque.BBox.obterMaiorX &&
305                     xz.Z >= tanque.BBox.obterMenorZ &&
306                     xz.Z <= tanque.BBox.obterMaiorZ)
307                 {
308                     tanqueRemover = tanque;
309                     break;
```

Trechos relevantes: Hitbox

```
312     if (tanqueRemover != null)
313     {
314         if (tanqueRemover.Tquespecial)
315         {
316             tanqueRemover.Vida--;
317             if (tanqueRemover.Vida == 0)
318             {
319                 objetosLista.Remove(tanqueRemover);
320                 tanques.Remove(tanqueRemover);
321                 posicoes.Remove(tanqueRemover.Posicao);
322             }
323         }
324         else
325         {
326             objetosLista.Remove(tanqueRemover);
327             tanques.Remove(tanqueRemover);
328             posicoes.Remove(tanqueRemover.Posicao);
329         }
```

Trechos relevantes: Poder

```
521 private void ativaPoder(object source, EventArgs e)
522 {
523     for (int i = 0; i < tanques.Count; i++)
524     {
525         Ponto4D centro = tanques[i].BBox.centro;
526         if (centro.Z < 3)
527         {
528             centro.Y = 101;
529             Nave raio = new Nave('a', null, 50, 2, -100, new Cor(128, 0, 255), centro, 3);
530             raios.Add(raio);
531             objetosLista.Add(raio);
532             objetosLista.Remove((Objeto)tanques[i]);
533             posicoes.Remove(((TQ)tanques[i]).Posicao);
534             tanques.Remove(tanques[i]);
535         }
536     }
537 }
538
539 this.poder.Enabled = false;
```

Trechos relevantes: Vida da Nave

```
tanque.mirar(centros);

if (((Nave)naves[0].ObjetosLista[1]).Cor.CorR < 255)
{
    ((Nave)naves[0].ObjetosLista[1]).Cor.CorR += 1;
}
else if (((Nave)naves[0].ObjetosLista[1]).Cor.CorG != 0)
{
    ((Nave)naves[0].ObjetosLista[1]).Cor.CorG -= 1;
}
else
{
    foreach (Nave nave in naves)
    {
        nave.Cor.CorR = 33;
        nave.Cor.CorG = 33;
        nave.Cor.CorB = 33;
        ((Nave)nave.ObjetosLista[0]).Cor.CorR = 33;
        ((Nave)nave.ObjetosLista[0]).Cor.CorG = 33;
        ((Nave)nave.ObjetosLista[0]).Cor.CorB = 33;
        ((Nave)nave.ObjetosLista[1]).Cor.CorR = 33;
        ((Nave)nave.ObjetosLista[1]).Cor.CorG = 33;
        ((Nave)nave.ObjetosLista[1]).Cor.CorB = 33;
    }
}
```

Trechos relevantes: Chão

```
16 public Chao(char rotulo, Objeto paiRef, Cor cor) : base(rotulo, paiRef)
17 {
18     this.cor = cor;
19     base.PontosAdicionar(new Ponto4D(300, -0.2, 200));
20     base.PontosAdicionar(new Ponto4D(300, -0.2, 200));
21     base.PontosAdicionar(new Ponto4D(300, 0, 200));
22     base.PontosAdicionar(new Ponto4D(-300, 0, 200));
23     base.PontosAdicionar(new Ponto4D(-300, -0.2, -200));
24     base.PontosAdicionar(new Ponto4D(300, -0.2, -200));
25     base.PontosAdicionar(new Ponto4D(300, 0, -200));
26     base.PontosAdicionar(new Ponto4D(-300, 0, -200));
27 }
```