

## Prova 3 – Counter Strike

**Professor:** Dr. Rodrigo Fernandes de Mello ([mello@icmc.usp.br](mailto:mello@icmc.usp.br))  
**Colaborador:** Felipe Simões Lage Gomes Duarte ([fgduarte@icmc.usp.br](mailto:fgduarte@icmc.usp.br))

### 1 Counter Strike

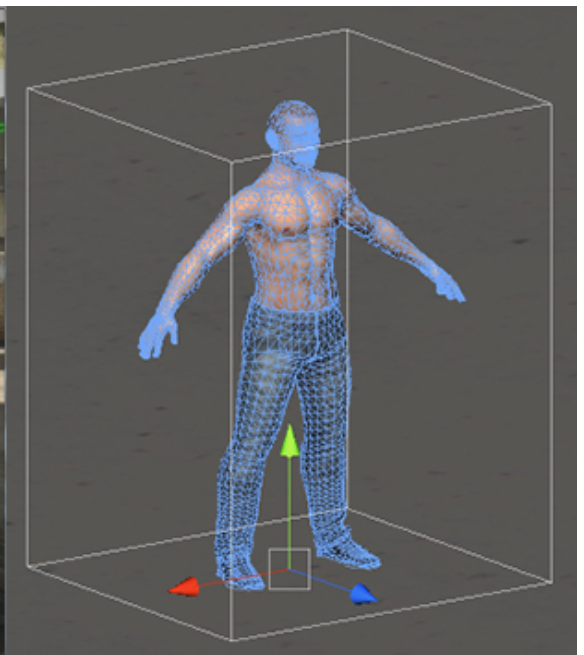
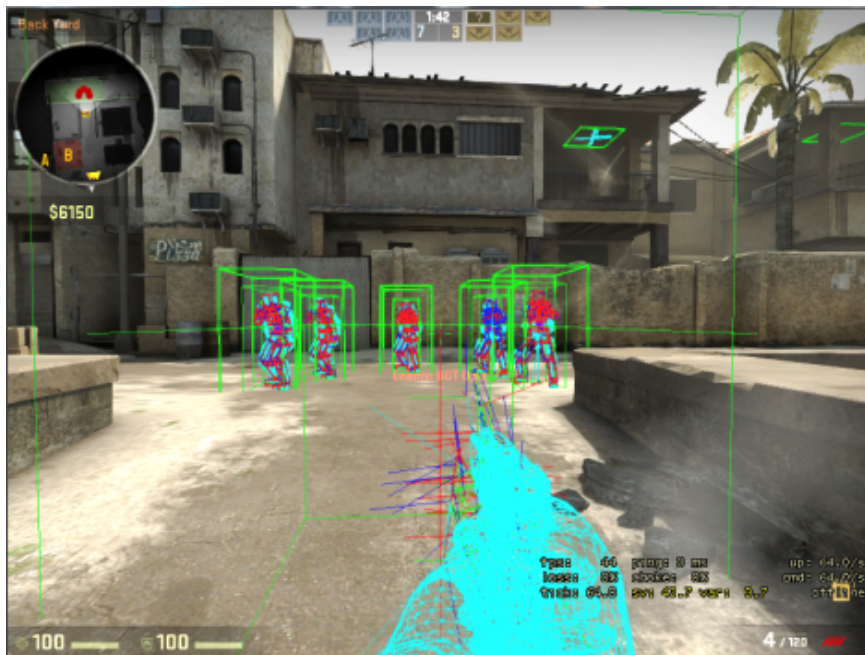
**Counter-Strike** é um popular jogo eletrônico de tiro em primeira pessoa. Inicialmente criado como um “mod” de Half-Life para jogos online, foi desenvolvido por Minh Le e Jess Cliffe e depois adquirido pela Valve Corporation. Foi lançado em 2000 para Windows, e posteriormente foram feitas versões para Xbox, Mac OS X e Linux. O jogo é baseado em rodadas nas quais equipes de contra-terroristas e terroristas se combatem até a eliminação completa de um dos times, e tem como objetivo principal plantar e desarmar bombas, ou sequestrar e salvar reféns.

### 2 Objetivo

Você deverá implementar uma das funcionalidades mais importantes do jogo Counter Strike: **Detecção de Colisão da Bala com o personagem**. Para isso implemente a leitura de um modelo (coordenadas vetoriais em espaço 3D que correspondem ao personagem), calcule o *bounding box* mínimo deste objeto e detecte a colisão entre a bala e o personagem.

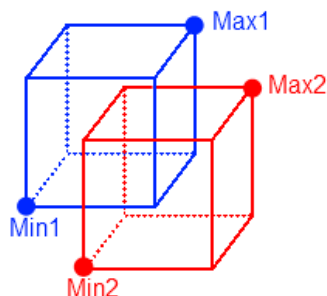
### 3 Bounding Box e Colisão

Em geometria espacial, o bounding box mínimo é dado pelo paralelepípedo reto com as menores dimensões (altura, comprimento, largura e, conseqüentemente, volume) no qual todos os pontos do objeto ou personagem estão contidos dentro. No contexto de computação gráfica, o bounding box é o paralelepípedo que envolve o objeto e simplifica sua descrição para cálculo de colisão e interação entre os objetos na cena.



Para detectar se a bala atingiu o personagem alvo, o bounding box da bala deve estar totalmente contido dentro do bounding box do personagem, i.e., as coordenadas mínimas do bounding box da bala devem ser maiores que as coordenadas mínimas do objeto e as coordenadas máximas da bala devem ser menores que as coordenadas máximas do objeto.

Observe o caso abaixo em que dois paralelepípedos retos, descritos por seus dois pontos (mínimo e o máximo), não estão contidos um dentro do outro. No caso da bala, somente há colisão se o paralelepípedo da bala estiver contido (ou dentro) do paralelepípedo que corresponde ao personagem.



## 4 Comandos do Sistema:

O seu programa deverá iniciar lendo uma opção inteira e o nome do arquivo que estão contidos os pontos que descrevem o personagem (o nome do arquivo será uma string sem espaços e com tamanho menor que 50). Este arquivo contém na primeira linha um número inteiro que representa a quantidade de pontos tridimensionais ali presentes. Na sequência estão as coordenadas  $x$ ,  $y$  e  $z$  de cada ponto (vide exemplo de arquivo disponível no run.codes) todas elas em ponto flutuante com precisão dupla (ou seja, double). Seu programa deverá carregar esses pontos para a memória heap.

Em seguida, para avaliação progressiva de seu programa e validação de todas as etapas de desenvolvimento, seu programa deverá responder a um comando de entrada numérico definido pela opção inteira lida logo no início. Esse inteiro indicará qual ação seu programa deverá executar, conforme segue:

- ‘1’ - Você deverá imprimir na tela todas as coordenadas do modelo (personagem) carregado. A impressão deve ser feita uma coordenada por linha com o seguinte comando (na mesma ordem lida do arquivo de entrada):

```
printf("%.4lf %.4lf %.4lf \n", x, y, z);
```

- ‘2’ - Você deve calcular o bounding box do modelo carregado e imprimir na tela a coordenada mínima e a coordenada máxima (uma em cada linha). Para isso utilize o seguinte comando:

```
printf("Min:  %.4lf %.4lf %.4lf \n", xmin, ymin, zmin);
printf("Max:  %.4lf %.4lf %.4lf \n", xmax, ymax, zmax);
```

- ‘3’ - Você deve ler as coordenadas mínimas e máximas do bounding box da bala e determinar se o personagem foi ou não atingido. As coordenadas serão lidas da seguinte forma:

```
scanf("%lf", &bala_xmin);
scanf("%lf", &bala_ymin);
scanf("%lf", &bala_zmin);
scanf("%lf", &bala_xmax);
scanf("%lf", &bala_ymax);
scanf("%lf", &bala_zmax);
```

Em seguida seu programa deve imprimir:

```
printf("COLISAO\n");
```

caso o personagem tenha sido atingido, ou

```
printf("NULL\n");
```

caso contrário.

## 5 Observações importantes

- Utilize alocação dinâmica (memória heap) para armazenar as coordenadas do modelo (personagem);
- não esqueça de liberar toda memória alocada antes de encerrar a execução do seu programa;
- implemente no mínimo três funções distintas, uma para cada uma das três opções de execução do seu programa. Observe que é um número mínimo, nada impede de você criar outras funções de acordo com sua necessidade;
- A prova deverá ser resolvida durante o período de aula e submetida para correção no run.codes. O sistema abrirá para submissão a 15 minutos do final da aula e fechará, em definitivo, no horário exato de término da aula;
- Não é permitido pesquisar na Internet ou acessar qualquer outro arquivo local de seu computador ou site que não seja o run.codes.