

## Segundo Trabalho

### Cena Simples Interativa com Câmara Móvel e Colisões

#### Objectivos

Os objectivos do segundo trabalho de laboratório são explorar o conceito de câmara virtual, as diferenças entre câmara fixa e câmara móvel, as diferenças entre projecção ortogonal e projecção perspectiva, a compreensão das técnicas básicas de animação e a detecção de colisões.

A avaliação do segundo trabalho será realizada na semana de **21 a 26 de Outubro** e corresponde a **5 valores** da nota do laboratório. A realização deste trabalho tem um esforço estimado de **10 horas** por elemento do grupo, distribuído por duas semanas.

#### Tarefas

As tarefas do segundo trabalho são as seguintes:

1. Criar um campo de jogo composto por uma cerca com três paredes com uma altura "H" (ver Fig. 1.) As seguintes características devem ser seguidas:
  - a) A altura das paredes deve ser tal que não permita que as bolas caiam da cerca (por ex.  $H > 2 \cdot R$ , onde R é o raio da bola).
  - b) As bolas são disparadas por três canhões, tendo cada canhão a sua direção de disparo.
  - c) A direção de cada canhão deverá ser programada, ou seja, o ângulo de disparo deve ser selecionado. Poderá haver um ângulo pré-definido (por defeito) para cada canhão. As teclas " $\rightarrow$ ", " $\leftarrow$ ", deverão ser usadas para incrementar/decrementar o ângulo de disparo de cada canhão.
  - d) Os canhões podem ser modelados recorrendo a um cilindro.
  - e) As bolas são disparadas sobre o piso e não devem saltar, deslocando-se sempre sobre o solo e em linha reta.
  - f) As teclas "Q(q)", "W(w)", "E(e)", quando ativadas, devem acionar um dos três canhões que dispara uma bola.
  - g) Deve ainda ser criado um conjunto de N (arbitrário) bolas. Estas bolas deverão ser inicialmente colocadas em posições aleatórias no interior da cerca (ver Fig. 1).
2. Definir uma câmara fixa com uma vista de topo sobre a cena utilizando uma projecção ortogonal que mostre toda a cena (tecla "1"). [**1,5 valores**]. Definir

ainda duas câmaras adicionais tendo o cuidado de manter a câmara definida anteriormente. Deve ser possível alternar entre as três câmaras utilizando as teclas “1”, “2” e “3”. A câmara 2 deve ser fixa e permitir visualizar todo o terreno de jogo através de uma projeção perspectiva. A câmara 3 deve também utilizar uma projeção perspectiva mas é móvel. Esta deve estar colocada atrás de uma bola e acompanhar o seu movimento (essa bola deve ser visível). **[1,5 valores]**

3. Realizar o movimento das bolas. Este deve ser um movimento retilíneo uniformemente retardado. Mais especificamente, após algum tempo decorrido sobre a colisão, as bolas devem parar, ou seja, há perda de energia cinética. Para isso deve ser pré-definido um valor para a força de atrito.

Deve ser possível mostrar ou esconder os eixos das bolas através da tecla ‘R’. Diferentes bolas devem movimentar-se com velocidades diferentes (atribuídas aleatoriamente) e a sua velocidade vai diminuindo com o tempo. Deve-se detetar e tratar a colisão das bolas. As colisões podem ser (i) bola-bola ou (ii) bola-parede. Na primeira (colisão bola-bola), esta deve tratada usando esferas envolventes. Na segunda (bola-parede), esta deve ser tratada usando bounding boxes alinhadas, ou usando limites. Nota, caso uma bola colida com uma parede, a bola deve ricochetear nesta, ficando a parede imóvel.

**Nota:** Dado que a cerca não tem uma parede, se as bolas saírem por este lado, estas deverão cair para o infinito. **[1,5 valores]**

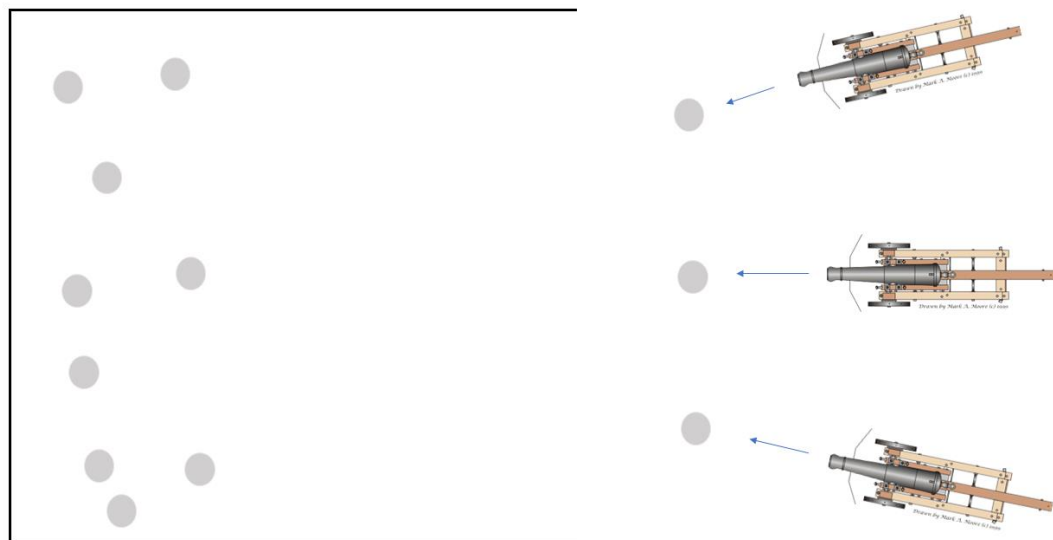


Figura 1 – Imagem ilustrativa da disposição dos objetos visíveis na cena: campo com cerca de 3 paredes, bolas alvo, bolas de canhão e 3 canhões (vista de topo).

## Nota Importante

**Nota1:** Não devem utilizar bibliotecas externas nem funções do three.js para detetar colisões ou implementar a física inerente ao movimento. Esperamos ver o vosso código e não chamadas a funções de bibliotecas.

**Nota 2:** Existem dois tipos de colisão. Colisão **bola-bola**, colisão **bola-parede**. Esta última colisão pode ser realizada usando limites ou usando bounding boxes alinhadas.

## Sugestões

1. A posição, direção e velocidade inicial das bolas podem ser obtidas recorrendo a *Math.randFloat(low, high)*.
2. Para criar o eixo das bolas pode-se usar o objecto *AxesHelper(size)*.
3. Para decrementar a velocidade das bolas com o tempo é recomendado o uso de um temporizador com algumas dezenas de segundos. Ao disparar o temporizador, a velocidade das bolas diminui ligeiramente.
4. Para mais informação relativa a colisões e conservação de momento, consultar:

[https://pt.wikipedia.org/wiki/Conserva%C3%A7%C3%A3o\\_do\\_momento\\_linear](https://pt.wikipedia.org/wiki/Conserva%C3%A7%C3%A3o_do_momento_linear)