Segundo Trabalho
Cena Simples Interativa com Câmara Móvel e Colisões

Objectivos

Os objectivos do segundo trabalho de laboratório são: (1) explorar o conceito de câmara virtual, perceber as diferenças entre (2) câmara fixa e câmara móvel, e entre (3) projecção ortogonal e projecção perspectiva; deseja-se ainda (4) a compreensão das técnicas básicas de animação bem como a detecção de colisões.

A avaliação do segundo trabalho será realizada na semana de **26 a 30 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.

Não esquecer de comunicar ao docente do laboratório as **horas despendidas pelo grupo (média do grupo)** na realização deste trabalho.

Tarefas

As tarefas do segundo trabalho são as seguintes:

1. Criar uma mesa de bilhar como a Figura 1 ilustra. Devem ter-se em conta as seguintes características na construção da mesa de bilhar:

   a) A altura H das paredes da mesa deve ser tal que não permita que as bolas caiam para fora da mesa (por ex. H > 2*R, onde R é o raio da bola – ver exemplo que se ilustra na Figura 2). Os alunos devem especificar as dimensões da mesa de bilhar bem como as dimensões das esferas;

   b) As bolas brancas são disparadas por vários tacos e sobre o plano da mesa. As bolas são automaticamente colocadas à ponta do taco (isto é, há uma bola branca acessivel em cada um dos 6 tacos). Na Figura 1, ilustram-se 6 tacos que disparam as bolas, devendo estas deslocar-se sempre sobre o plano da mesa. Não são permitidos quaisquer saltos no movimento da bolas. As bolas devem rodar sobre si mesmas na direção do seu deslocamento**;**

   c) A direção de disparo da tacada deverá ser programada, ou seja, o ângulo da tacada deve ser selecionado. Poderá haver um ângulo máximo pré-definido, por exemplo 60 graus, e com as teclas “<-” e “->” ajustar o ângulo da tacada.

Por exemplo, o ângulo da tacada poderá estar contido no intervalo [-60,60] graus. (A escolha deste ângulo pode ser um parâmetro a ajustar);
De notar ainda que, no processo de seleção do angulo, o taco deve de rodar.

d) Os tacos podem ser modelados recorrendo, por exemplo, a cilindros;

e) As teclas "4" a "9", quando ativadas, devem acionar um dos 6 tacos que dispara uma bola. (damos a liberdade de poder haver disparos em simultâneo);

f) O taco selecionado deve ter uma cor diferente dos restantes; não é mandatório modelar o movimento do taco no processo da tacada (i.e., aproximação do taco à bola);

g) O disparo do taco deve ser efetuado usando a tecla "space";

h) Dá-se a oportunidade de selecionar mais do que um taco para o disparo das bolas;

i) Deve ainda ser criado um número $N \geq 15$ (arbitrário) de bolas. Estas bolas deverão ser inicialmente colocadas em posições aleatórias no interior da mesa de bilhar (ver Figura 1) em movimento e com velocidades diferentes;

j) Quando a bola atingir um dos buracos da mesa, esta deve desaparecer no interior da mesa e ter uma queda no infinito. Será desejável a visualização da queda.

Nota1: Admite-se sempre que a bola que está à saída do taco (a bola à qual vai ser dada a tacada) não colide com as outras. Só se faz o estudo da colisão após a tacada.

Nota2: É deixado ao critérios dos alunos, que a aplicação possa (ou não) permitir tacadas simultâneas. Não é mandatório existirem tacadas simultâneas.

2. Definir uma câmara fixa com uma vista de topo sobre a mesa de bilhar utilizando uma projeção ortogonal que mostre toda a cena (semelhante ao que é mostrado na Figura 1). Esta visualização deverá ser conseguida usando a 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 toda a cena através de uma projeção perspetiva. A câmara 3 deve também utilizar uma projeção perspetiva mas é móvel. Esta deve estar colocada atrás da bola disparada e acompanhar o seu movimento (essa bola deve ser visível). **[1,5 valores]**

Realizar o movimento das bolas. O movimento deve ser retilíneo uniformemente retardado. Ou seja, após algum tempo decorrido sobre uma colisão ou um disparo, as bolas devem perder progressivamente a velocidade. 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 ser 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. No caso de colisão bola-bola, esta deve ser uma colisão elástica.

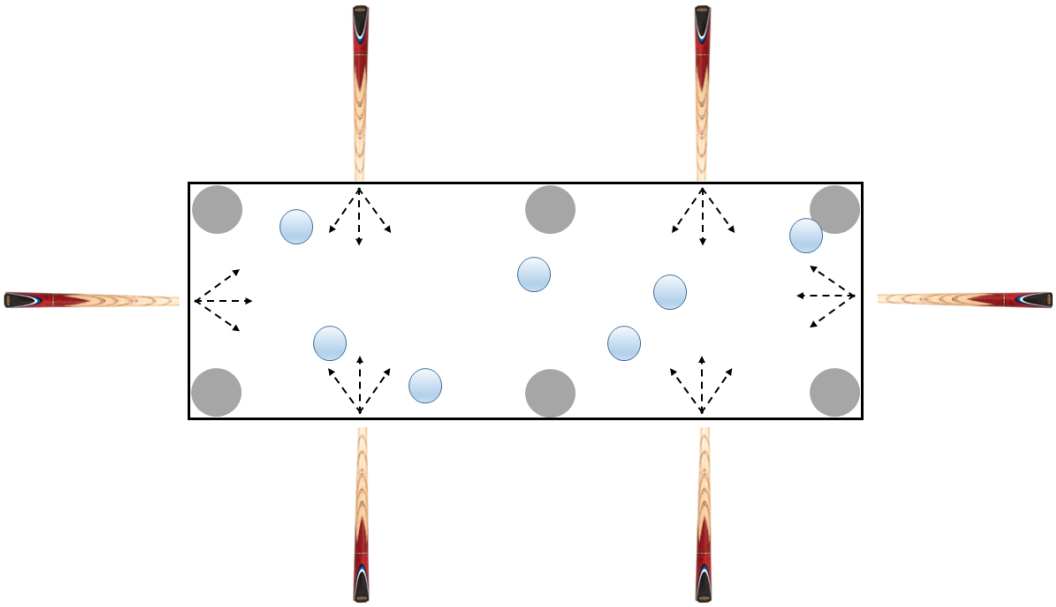

Figura 1 – Imagem ilustrativa da mesa de bilhar. Ilustram-se os seis tacos que permitem o disparo das bolas que são colocadas na mesa numa posição à ponta do taco (vista de topo).

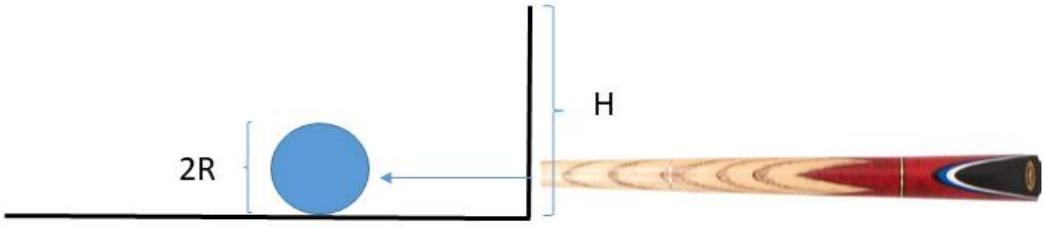

Figura 2 – Imagem ilustrativa do disparo da bola pelo taco. Assume-se que a bola efetua um deslocamento sobre o plano da mesa. Mostra-se ainda as dimensões da bola (2R) e da altura da mesa de bilhar (H), onde se verifica a relação H > 2R, não permitindo que a bola saia do espaço delimitado pela mesa de bilhar. Não poderá haver salto da bola após a colisão com a parede da mesa.

Notas Importantes

Nota 1 : Antes de escrever qualquer linha de código, é necessário esboçar o que se pretende modelar em 3D pois tal actividade ajuda muito a perceber que primitivas e transformações devem ser aplicadas

Nota 2: Não devem utilizar bibliotecas externas nem funções do three.js para detectar colisões ou implementar a física inerente ao movimento. Esperamos ver o vosso código e não chamadas a funções de bibliotecas.

Nota 3: 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. Para além de dos acontecimentos de update e display existem mais um conjunto de acontecimentos, tais como teclas pressionadas ou soltas, temporizadores e redimensionamento da janela. Sugerimos vivamente que tais acontecimentos sejam tratados pelas respectivas funções de callback de forma independente. Tenha em atenção que nos próximos Trabalhos iremos requerer a implementação devida dos acontecimentos de redimensionamento da janela!

2. Por fim, os alunos devem adoptar uma programação orientada a objectos, seguindo sempre boas práticas de programação que permitam a reutilização do código em entregas posteriores e facilitem a escalabilidade.

3. A posição, direção e velocidade inicial das bolas podem ser obtidas recorrendo a *Math.randFloat(low, high)*.

4. Para criar o eixo das bolas pode-se usar o objecto *AxesHelper(size)*.

5. 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.

6. Para mais informação relativa a colisões e conservação de momento, consultar:

   https://pt.wikipedia.org/wiki/Conserva%C3%A7%C3%A3o_do_momento_linear