Pinball

Ronaldo Yang & Yoshio Mori
7576750 & 6432393

Junho de 2014

Sumário

1 Estrutura

1.1 Função Principal (main - pinball.js)

É a primeira função a ser chamada pelo cliente. Ela carrega o canvas, a interface do webgl, o scene e é reponsável por fazer a integração entre esses sistemas.

A rotina dela consiste em carregar os buffers, inicializar o shader, criar os objetos. Após isso ele entra em loop de atualização dos objetos e renderização.

1.2 Classe principal (Scene - scene.js)

Essa classe contém as informações das imagens, de como desenhá-las na tela e das posições, tamanhos e orientações de cada peça.

1.2.1 Atributos da Classe

gl Interface gráfica.

[shaders] Conjunto de Object shader.

images Conjunto de Object image.

pieces Conjunto de Object piece.

camera Conjunto de Object camera.

ready Boolean informa true se todos os Object image foram carregados.

1.2.2 Principais Métodos da Classe

createShaders Inicia um novo Object shader obtendo a posição, normal e textura de cada variável no shader.

createImage Faz a leitura do arquivo Obj, coletando os vértices para serem armazenados num buffer da gpu. Com isso ele instancia um Object image.

isReady Para cada Object image instanciado, verifica se a imagem já está pronta. Dessa forma é possível determinar quando começar as iterações do jogo.

1.3 Object image

Esse objeto contém os endereços dos buffers de posições, normais, textura e indices de cada um dos vértices, assim como o shader reponsável pela renderização.

E seu único método, draw, é chamado pelo Object piece com os parâmetros das matrizes de projeção vértice e modelo. Com esses parâmetros inicia-se o processo de renderização.

1.4 Object piece

Cada Object piece está associado à um Object image. Ela contém as informações de posição, tamanho e orientação.

No processo de construção as informações de posição, tamanho e orientação são usados para gerar a matriz de modelo, e sempre que esse atributos são atualizados a rotinha se repete.

1.4.1 Principais Métodos

isReady Verifica se a imagem foi inicializada.

show É chamada pela câmera com os parâmetros das matrizes de projeção e visualização. O método envia esses parâmetros para o Object image associado, assim como sua matriz de modelo.

1.5 Object camera

Seus atributos são posição, ponto da direção que a câmera está voltado e vetor que indica o cima da camera.

Assim como no object piece, aos ser instanciado as matrizes de projeção e visualização são calculadas e sempre que seus atributos são atualizados novas matrizes são geradas.

Seu método show é chamado pelo scene, que por sua vez mostra a imagem de cada peça.

2 Objetos

Cada objeto da mesa de pinball está na pasta pinballOBJ. Todos foram feitos usando o software Blender 2.7.

Dentro da pasta pinballOBJ, há outra pasta que contém um mesaInteira.obj, onde está representado toda mesa de pinball, ou seja, todos os objs do diretório anterior juntos. Assim dá para ter uma noção de como é a mesa pinball através do software Blender.

3 Eventos do Teclado

Os eventos descritos aqui estão no projeto flippers-keyboard.

Para a verificação dos eventos do teclado, assim como a movimentação das palhetas, foi feito um pequeno simulador. Esse, consiste em dois triângulos simulando o que seriam as palhetas e se movimentando(rotacionando no eixo z até um certo ângulo e depois voltando a posição original) quando as teclas certas são pressionadas. Para ser mais semelhante as palhetas, os triângulos deveriam ter sido iniciados transladados um pouco para o eixo x positivo, para que o eixo de rotação fosse mais parecidos com os das palhetas. Como era apenas um teste, não foi dada preocupação a isso.

Foi implementado também nesse simulador o botão de start, pause, unpause e restart. A lista de teclas e suas funções são:

- ENTER para start

- Z para movimentação do triângulo esquerdo
- X para movimentação do triângulo direito
- P para pause
- U para unpause
- R para reiniciar

Ao iniciar esse simulador, só haverá algum movimento(início) quando se dá start. Ao apertar ENTER, podemos começar a movimentar as palhetas com Z e X. Caso apertamos P em algum momento, a cena congela e só continua a sua movimentação assim que apertar U. Para restart, aperta-se R e as palhetas voltam a posição inicial. Para iniciar a movimentação novamente basta apertar ENTER.

Outros eventos que foram descritos, mas não implemetados são:

- SPACE para a força da mola
- PAGE UP para o aumento da inclinação da mesa
- PAGE DOWN para a diminuição da inclinação da mesa

Para o caso da mola, iríamos achatar a mola conforme o pressionamento da tecla SPACE. Quanto maior pressionamento, maior seria a força na direção da bola.

4 Colisões

Para a detecção de colisões iríamos fazer o seguinte:

4.1 Bola-Bola

Basta verificar se a distância entre os centros é menor ou igual a soma dos raios.

4.2 Bola-Poliedro

- Encontramos os limitantes do poliedro nos eixos x, y e z. Assim criamos uma espécie de bounding box, um cubo que envolve/encapsula todo o poliedro
- Verificamos se existe uma potencial intersecção entre esse cubo e a esfera(bola). Assim evitamos mais cálculos caso não exista colisão, pois é mais fácil verificar a intersecção entre uma esfera e um cubo
- Caso não exista intersecção, não há colisão. Fim
- Se sim, pode existir uma possível colisão
- Verificar a colisão através de pontos convenientes do poliedro e da esfera