Computação Gráfica - Trabalho Prático 1

Luiz Otávio

Maio 2018

1 Introdução

Foi realizado a implemetnação de uma versão do jogo Breakout utilizando a linguagem C++, OpenGL e a lib SDL2 para gerenciamento de janelas, inputs, etc.

Breakout é um jogo arcade desenvolvido e publicado pela Atari, Inc., lançado em 13 de maio de 1976. Foi conceituado por Nolan Bushnell e Steve Bristow, influenciado pelo jogo de arcade Pong, de 1972, da Atari, e construído por Steve Wozniak, auxiliado por Steve Jobs. O jogo foi portado para múltiplas plataformas e atualizado para videogames como Super Breakout. Além disso, Breakout foi a base e inspiração para certos aspectos do computador pessoal Apple II.

No jogo, uma camada de tijolos está no topo da tela. Uma bola viaja através da tela, saltando fora das paredes superiores e laterais da tela. Quando um tijolo é atingido, a bola é quicada e o tijolo é destruído. O jogo reinicia quando a bola toca a parte inferior da tela. Para evitar que isso aconteça, o jogador movimenta um *paddle* para rebater a bola para cima, mantendo-a em jogo.

2 Execução

O projeto está disponível no repositório: https://github.com/Luiz0tavio/breakout

2.1 Dependências

Distribuição Linux e as seguintes libs:

- SDL2
- SDL2_image
- GL
- GLEW

2.2 Compilação

Dentro da pasta raiz se encontra um Makefile e um CMakeLists.txt (cmake).

O Makefile irá gerar um binário chamado *breakout* dentro da pasta *bin*. Entre dentro dela para executá-lo:

```
$ make
$ cd bin
$ ./breakout
```

3 Implementação

3.1 Memória

Foi utilizado um *Pool de Memória* para evitar alocação dinâmica (alto overhead) e manter vertices, texturas, etc contíguos na memória, o que torna a iteração mais rápida.
Para isso, logo no início do programa, alocamos nosso *Pool de Memória* (5MB). Então, fazemos overload no operador *new* de todas as classes. Nosso novo *new* fará uma requisição no Pool de memória e não alocará nada!
Templates da STL também aceitam um *Allocator* para alocarmos memória para ele, ou seja, nosso std::vector fará uma requisição em nosso Pool de Memória a cada push_back().
O Pool de Memória é circular e é limpo apenas no fim da execução do programa.

3.2 Jogabilidade

O jogo é ajustado para rodar a 60fps.
Iniciará pausado. A bola é refletida nos blocos e nas paredes. Quando toca em um bloco, ele é destruído. São 33 blocos no total. Caso ela toque o chão, o jogo irá reiniciar.
A bola ganha velocidade linearmente a cada iteração.
O paddle é movimentado pelo mouse. Sua velocidade varia: quanto mais perto das bordas, mais veloz naquela direção ele irá.
O botão esquerdo do mouse pausa/despausa.
O botão 'q' fecha o programa.
O botão 'r' reincia o jogo.

3.3 Texturas

O componentes do jogo têm textura simples (sem transparência). Os blocos possuem textura aleatória. Há 25 texturas únicas para 33 blocos.

4 Conclusão

Apesar de um jogo simples, possibilidades de funcionalidades para o jogo são infinitas. Com o poder computacional atual, até mesmo em dispositivos móveis, podemos utilizar de uma idéia de 1976 e recriar um jogo que pode se tornar altamente viciante e competitivo adicionando vários modos de jogos, rankings, tipos de blocos e bolas, efeitos, transiços etc