

Universidade Federal de Minas Gerais

Trabalho Prático

AEDS 1

Tema:
Arcade

Grupo:
João Gabriel de Oliveira Bicalho
Rafael Castro Araujo Beirão

Belo Horizonte
Junho de 2017

Introdução

Neste trabalho, foram implementadas interpretações dos jogos clássicos Pong, Breakout e Space Invaders.

Os jogos utilizam várias funções em comum, que foram colocadas em um único arquivo "*common.h*". Já as funções específicas de cada jogo estão divididas em 3 módulos separados (arquivos *.h*) com seus respectivos nomes. Cada um desses módulos pode ser dividido em três partes, sendo elas as funções de entrada e saída de dados, a seção de renderização, que utiliza uma matriz pré-construída e a renderiza com sprites em ASCII e a seção de lógica de jogo, que realiza todas as operações lógicas do jogo e prepara a próxima "frame". Todas as três partes foram construídas para funcionar em Windows e Linux através de funções que abstraem as especificidades desses sistemas ou através de *defines*.

Toda a renderização do jogo é feita através de comandos *printf()*, sendo que, em Windows, é usada a biblioteca *conio.h* para otimizar a velocidade desse processo.

Todos os jogos leem o seu "mapa" de um arquivo de texto contendo uma matriz de números, cada um com um significado dependendo do jogo, e explicado no código-fonte. Este arquivo está localizado no mesmo diretório que o executável, e o executável o usa sempre que o "mapa" é reexibido na tela. Dessa forma, para colocar paredes no Pong, por exemplo, basta alterar o arquivo "helper" e reabrir o programa.

Além disso, os jogos Pong e Breakout salvam o histórico de partidas num arquivo de log, que pode ser lido através do próprio programa. Cada arquivo de log possui diferentes informações, pertinentes ao jogo.

Pong

Em Pong, é possível escolher entre jogar com dois jogadores, um jogador e um computador, ou computador contra computador. A inteligência artificial foi programada simplesmente para seguir a coordenada vertical da bola quando esta está vindo em sua direção, possuindo um fator aleatório que define se ela será ou não alcançada, de modo a tornar o modo Player vs COM algo possível de vencer.

Space Invaders

Em Space Invaders, o jogador controla uma nave e tem que destruir todos os inimigos antes que eles atinjam o solo, além de desviar dos tiros dos oponentes. Nesse jogo, os inimigos são expressos como objetos em um vetor, que são transcritos em uma matriz de renderização. Também são objetos o jogador, e os projéteis de ambas as partes, tendo as colisões desses objetos analisadas quando existem dois objetos com as mesmas coordenadas.

O jogo acaba quando os inimigos atingem o chão, quando jogador perder todas as vidas, ou quando todos os inimigos forem destruídos.

Breakout'

Em Breakout', o jogador controla uma barra, e seu objetivo é destruir os blocos da parte superior da tela usando uma bolinha. O jogo termina quando 90% dos blocos forem destruídos ou quando o jogador perder as 6 vidas (quando não consegue pegar a bola antes dela alcançar o chão). O jogo recebe a apóstrofe no fim do nome para diferenciar do breakout original, que exigia que todos os blocos fossem destruídos para que o usuário ganhasse o jogo.

Este é o único módulo colorido. Isso foi feito através da biblioteca conio.h no Windows e de comandos de terminal no Linux.

Instalação

Todos os três jogos são compilados ao mesmo tempo, já que são módulos .h carregados a partir de um arquivo principal "arcade.c".

Como nenhuma biblioteca não-padrão foi utilizada no Linux, basta executar o comando abaixo na pasta do projeto para compilar e rodar os jogos:

```
$ gcc main.c -o game && ./game
```

No Windows é necessário linkar a biblioteca "conio" e a "conio_unicode" para que o projeto possa ser compilado. Os compilados das bibliotecas estão incluídos no pacote. Use o Code Blocks para compilação.

Futuro

Embora os jogos estejam completos em si, é possível que mais jogos ou novas funcionalidades sejam adicionados, além de otimizações do código já existente.

Dessa forma, caso tenha interesse, você pode seguir o repositório do projeto no GitHub através do endereço: <https://github.com/castro-joao/arcade>.