

INF 01202 – ALGORITMOS E PROGRAMAÇÃO Turmas A e B - Prof. Anderson Maciel

Trabalho Prático Final – Semestre 2012/1

MOTIVAÇÃO E OBJETIVOS

O objetivo deste trabalho é exercitar as habilidades e conceitos de programação desenvolvidos ao longo da disciplina pela implementação de um jogo em C, em grupos de 2 alunos.

Deverá ser feito um jogo inspirado no clássico jogo *Arkanoid*, em modo texto, onde o usuário do programa (jogador) controlará a movimentação de uma espécie de raquete, movendo-a para esquerda a direita. A raquete representa a nave do jogador. A nave se chama *Vaus*, que após escapar da nave mãe (*Arkanoid*) que foi destruída, está aprisionada em uma dobra espacial. Para escapar, o jogador precisa rebater a bola, evitando que caia e saia pelo plano inferior do jogo, perdendo uma vida quando isso ocorrer. Ao mesmo tempo, a bola deve ser rebatida contra um conjunto de tijolos. Quando a bola atinge um tijolo, este desaparece. Quando todos os tijolos tiverem sido destruídos, o jogador passa ao próximo nível/fase, onde aparece um novo conjunto de tijolos. Podem existir diferentes tipos de tijolos, alguns que necessitam múltiplas boladas para desaparecerem, por exemplo. Outros deixam cair supervitaminas que ao serem colhidas pela *Vaus* a expandem (tornam a *Vaus* mais larga), facilitando a vida do jogador. Ainda, pode haver um supertijolo que ativa uma segunda bola no ambiente.

"THE TIME AND ERA OF THIS STORY IS UNKNOWN. AFTER THE MOTHERSHIP "ARKANOID" WAS DESTROYED, A SPACECRAFT "VAUS" SCRAMBLED AWAY FROM IT. BUT ONLY TO BE TRAPPED IN SPACE WARPED BY SOMEONE....."



Figura 1: Uma versão arcade do *Arkanoid*

O jogo encerra quando o jogador perder suas 5 vidas. Caso sobreviva após 33 fases, deve enfrentar o chefe: Doh, um cabeça de moai que é quem controla a fortaleza dimensional. Uma implementação online deste jogo se encontra em http://www.brincar.pt/jogos/jogos-de-pericia/arkanoid_arkanoid.html

O PROGRAMA

As duplas terão liberdade para a implementação do programa. Entretanto, os seguintes requisitos mínimos devem ser respeitados:

- O jogo deverá ter todos os elementos exibidos em modo Texto (modo gráfico será considerado extra);
- O jogo deverá ter uma tabela de pontuação que deverá ser carregada através de um arquivo binário chamado `highscore.bin`. Nesse arquivo serão salvos a pontuação e o nome do jogador em ordem decrescente, no máximo 10;
- O jogador poderá salvar o estado atual do jogo e mais tarde carregá-lo para voltar a jogar de onde parou. Para tanto o estado deverá ser salvo em um arquivo binário chamado `save.bin`;
- A área de tijolos deverá ser carregada através de um arquivo de texto chamado 'bricks1.txt' que informe a posição inicial dos tijolos, e seu tipo. Cada caractere do arquivo texto representará um tijolo. O caractere X representará um tijolo comum. Os caracteres Y, Z e W representarão variações de tipo de tijolo à escolha dos programadores;
- As dimensões da área de tijolos são: máximo de 10 (altura) e 13 (largura) tijolos, sendo que cada tijolo é representado na tela por dois caracteres lado a lado;
- Os tijolos são fixos, mas a bola que se move não poderá sair do tabuleiro e deve haver uma detecção de colisão com as paredes do mesmo;
- Os tijolos serão representados pelo caractere '@' ou outro à escolha conforme o tipo, e os espaços vazios da área de tijolos pelo caractere ' ' (espaço em branco);
- A posição da bola no espaço do jogo deverá ser manipulada através de um *struct*, contendo sua posição atual em x (coluna) e y (linha) no espaço do jogo;
- Cada elemento deverá possuir sua própria *struct* tipo tijolo, *Vaus*, bola, etc.;
- A bola irá se mover de modo autônomo, sendo que o seu contato com outros elementos pode mudar sua direção de movimento (como? discutiremos isso.);
- As seguintes teclas deverão ser utilizadas para controlar a interação no jogo:
 - Esc – encerra o jogo a qualquer momento. Antes de encerrar confirma com o usuário que deseja mesmo encerrar;
 - S – Salva o estado do jogo;
 - L – Carrega o estado do jogo;
 - R – Reinicia o nível ; (extra)
 - Enter – Pausa o jogo; (extra)
 - Seta ← (ou caractere 'a') – move a *Vaus* uma posição para esquerda;
 - Seta → (ou caractere 's') – move a *Vaus* uma posição para direita;
- O jogo deve possuir uma tela que mostre um menu inicial com as seguintes opções: iniciar jogo, carregar jogo, exibir pontuações;
- O jogo deve reconhecer quando o jogo acabou, ou seja, quando todas as vidas acabarem ou a última fase for vencida;
- O jogo deverá conter um contador de pontos, a ser exibido em uma linha logo abaixo do final da área de jogo, ou em uma posição que achar mais conveniente;
- Ao final do jogo, o programa deverá verificar se a pontuação do jogador entra na tabela dos *High Scores*. Caso afirmativo, o programa deve solicitar a leitura do nome do jogador e armazenar seu nome e pontuação no arquivo `highscore.bin` (possivelmente excluindo o score mais baixo até então, se a tabela já se encontrar com seu número máximo de entradas);

Dica(s):

- A biblioteca conio.h disponibiliza funções básicas para a inclusão de caracteres em lugares definidos pelo programador na tela, como contém varias funções úteis, como *putchxy*, *kbhit* e *getch*. Como não é padrão do C, não há garantia de que esta biblioteca esteja disponível para todos os compiladores;
- Uma solução é usar a biblioteca conio2, que tem aproximadamente as mesmas funções. A versão dela para o compilador MinGW que usamos no CodeBlocks pode ser baixada em: <http://www.inf.ufg.br/~amacieli/software/conio2.zip> (vejam as instruções no leia.me.txt)
- No Linux, a biblioteca ncurses funciona de maneira similar (<http://www.tldp.org/HOWTO/NCURSES-Programming-HOWTO/index.html>);
- Para o controle da velocidade do jogo pode-se utilizar a função “clock()” e a constante “CLOCKS_PER_SEC”, que informa quantos clocks são necessários para 1 segundo. Ambos da biblioteca “time.h”;
- para o jogo não executar muito rápido, pode-se utilizar a função *sleep*;
- Para o cálculo dos pontos pode seguir o que é usado no Arkanoid original: 10 pontos por tijolo comum da primeira fase, aumentando nas outras fases e com pontuação maior para tijolos especiais;
- Para representar todos os símbolos da tabela ASCII estendida é necessário que a variável correspondente seja declarada como *'unsigned char'*.



Figura 2: Exemplo de Arkanoid em Modo Texto

REQUISITOS:

- O trabalho deverá ser feito em duplas. Informar os componentes da dupla até o dia 11 de maio, por e-mail ao professor Anderson.
- Até o dia 12 de julho, para ambas as turmas, a dupla deverá submeter via Moodle um arquivo zip cujo nome deve conter o(s) nome(s) do(s) aluno(s). O arquivo zip deve conter:
 - Uma descrição do trabalho realizado contendo a especificação completa das estruturas usadas e uma explicação de como usar o programa
 - Os programas-fonte devidamente organizados e documentados (arquivos.c e arquivos.h)
 - Executável do programa

- O trabalho será obrigatoriamente apresentado durante a aula prática do dia 13 de julho para todas as turmas. Ambos membros da dupla deverão saber responder perguntas sobre qualquer trecho do código.
- No dia da apresentação serão realizados diferentes testes de condições de jogo para testar o programa.
- Os seguintes itens serão considerados na avaliação do trabalho: estruturação do código em módulos, documentação geral do código (comentários, **identação**), “jogabilidade” do jogo e atendimento aos requisitos definidos.
- **Importante:** trabalhos copiados não serão considerados. Saibam que há ferramentas que possibilitam a detecção automática de plágio.