

# Frogger for MINIX

---

Laboratório de Computadores

Especificação do Projeto



Ana Amaral – up201303169

João Guarda – up201303463

## Conteúdo

Descrição.....	3
Modo <i>Single-Player</i> .....	3
Modo <i>Two-Player's</i> .....	3
Periféricos.....	4
Teclado .....	4
<i>Timer</i> /Contador 18254.....	4
Rato .....	4
Placa de vídeo.....	4
<i>Real Time Clock (RTC)</i> .....	5
Porta de Série .....	5
Módulos .....	5
Planeamento.....	7

## Descrição

### **Modo *Single-Player***

O jogador assume o controlo de um sapo que deve atravessar, com tempo limitado, uma estrada movimentada e um rio cheio de obstáculos até chegar ao seu ninho. A cada cinco travessias bem-sucedidas, o jogo progride de nível, aumentando a sua dificuldade (carros mais rápidos, menos troncos e tartarugas, corrente do rio mais rápida).

A estrada movimentada contém 4 faixas de rodagem com sentidos intercalados (esquerda para a direita e vice-versa). O sapo deve chegar ao lado oposto da estrada sem ser atropelado por nenhum dos veículos que circula nela. Superados estes primeiros obstáculos, o jogador, para alcançar o ninho do sapo, terá de atravessar um rio (onde não poderá cair nele) onde a corrente conduz troncos, possibilitando assim a passagem para a outra margem. O rio é compostos por 3 blocos horizontais, sendo que o primeiro e o último contêm troncos. No bloco central, a corrente mobiliza tartarugas. Estas são mais imprevisíveis que os troncos visto que submergem e emergem ao longo da sua passagem.

### **Modo *Two-Player's***

Os dois jogadores irão competir num sistema de pontos, onde o vencedor ganhará um ponto e o perdedor nada ganhará. Em caso de empate nenhum dos jogadores recebem pontos.

Utilizando as mesmas regras do *single-player*, um jogador para bater o adversário deve colocar mais sapos nos ninhos. Os jogadores utilizaram a mesma cena de jogo e poderão visualizar o sapo do adversário no seu computador. Para distinguir os sapos eles serão de cores diferentes.

Caso o tempo de jogo acabe antes de todos os ninhos serem preenchidos, o vencedor é aquele que tiver colocado mais sapos nos ninhos.

## Periféricos

O jogo utilizará os seguintes periféricos:

### Teclado

Modo: O teclado será utilizado com interrupções.

Descrição: Este dispositivo será utilizado para controlar o sapo durante o jogo, ler o nome do utilizador (para gravar no quadro de pontuações máximas caso seja necessário). Também poderá ser utilizado para navegação entre menus.

### Timer/Contador 18254

Modo: O *timer* será utilizado com interrupções.

Descrição: Será utilizado para contar o tempo de *respawn* dos carros, dos troncos e das tartarugas. Terá também um papel fulcral na contagem de *frames* por segundo.

### Rato

Modo: O rato será utilizado com interrupções.

Descrição: Utilizado na navegação entre menus. Será utilizado também como alternativa ao teclado para movimentar o sapo durante o jogo (ALTERAÇÃO EM RELAÇÃO AO PROJECTO).

### Placa de vídeo

Descrição: A placa de vídeo será utilizada em Modo Gráfico.

### ***Real Time Clock (RTC)***

Modo: O *RTC* será utilizado com interrupções.

Descrição: Será utilizado para saber a data para guardar *highscores* e para programar um alarme para notificar o utilizador sobre quando o tempo de cada nível chegar ao fim.

### **Porta de Série**

Modo: A porta de série será utilizada com interrupções.

Descrição: Será utilizada para um modo de 2 jogadores. Será utilizada para comunicar as posições dos sapos e na fase inicial será utilizada para sincronizar a mesma cena de jogo.

## **Módulos**

### **Rato**

Membro: João Guarda;

Descrição: Subscrição das interrupções e cancelamento das mesmas. Fornecerá as coordenadas X e Y do rato assim como se algum botão está a ser premido.

### **Teclado**

Membro: Ana Amaral;

Descrição: Subscrição das interrupções e cancelamento das mesmas. Corresponder o *make-code* a tecla correta, permitindo assim saber qual a tecla que o utilizador está a premir.

### **Timer**

Membro: Ana Amaral;

Descrição: Subscrição das interrupções e cancelamento das mesmas. Fornecer contagens de tempo.

**Modo Gráfico**

Membro: João Guarda;

Descrição: Inicializar o modo gráfico, ler ficheiros *.bmp* e testar colisões e libertar memória das imagens que já não serão utilizadas.

**RTC**

Membro: João Guarda;

Descrição: Subscrição das interrupções e cancelamento das mesmas. Fornece a data atual assim como permitirá o agendamento de alarmes.

**Porta de Série**

Membro: João Guarda;

Descrição: Subscrição das interrupções e cancelamento das mesmas. Estabelecerá um protocolo de comunicação com outra máquina.

**Motor de Jogo *Single-Player***

Membro: João Guarda;

Descrição: Conterá a informação relevante sobre o estado do jogo (ex. Posição dos carros, dos troncos e do sapo etc...). Conterá vários sub-módulos como: carro, camião, sapo, tronco, tartaruga, faixa.

**Motor de Jogo *Two-Player***

Membro: João Guarda;

Descrição: Semelhante ao módulo de *Single-Player* com os devidos ajustes.

## **Tratador de Interrupções**

Membro: Ana Amaral;

Descrição: Este módulo conterà informação de quais os periféricos estão ativos assim como será responsável por chamar o *handler* adequado para cada interrupção.

## **Menu Principal**

Membro: Ana Amaral;

Descrição: Este módulo conterà informação do menu principal. Conterà a posição dos botões das opções disponíveis, assim como a imagem do menu.

## **Menu *High-Score***

Membro: Ana Amaral;

Descrição: Este módulo conterà informação das pontuações máximas anteriormente obtidas, assim como a imagem deste menu.

## **Planeamento**

01 de Dezembro: Completar os módulos do Rato, Timer e Teclado.

08 de Dezembro: Completar os módulos do Modo Gráfico e dos Menus.

15 de Dezembro: Completar o Motor de Jogo do modo *Single-Player*.

22 de Dezembro: Completar os módulos do RTC e da Porta de série.

02 de Janeiro: Completar o Motor de Jogo do modo *Two-Player's*.

O módulo do tratamento de interrupções irá sendo atualizado ao longo do projecto.