

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

2ºano – MIEIC – 2015/2016

Copter

Laboratório de Computadores

Turma 6 - Grupo 11



Maria João dos Santos Aguiar e Mira Paulo
Nuno Miguel Mendes Ramos



Índice

1. RESUMO DO JOGO.....	3
2. DESCRIÇÃO DO PROGRAMA	4
2.1. MENU PRINCIPAL.....	4
2.2. MENU DE OPÇÕES	5
2.3. GRÁFICOS POSSÍVEIS DO JOGO	6
2.4. MENU DE "GAME OVER"	8
2.5. CRÉDITOS	9
3. PERIFÉRICOS USADOS	10
3.1. TIMER.....	10
3.2. TECLADO.....	10
3.3. RATO	11
3.3. PLACA GRÁFICA	11
3.4. RTC	12
4. DESCRIÇÃO DA ARQUITETURA DO PROGRAMA.....	13
4.1. MÓDULOS E DESCRIÇÃO.....	13
5. ESTADO DO PROJETO.....	17
5.1. INSTALAÇÃO	17
5.2. POSSÍVEIS MELHORIAS	17
5.3. EMPENHO DOS ELEMENTOS DO GRUPO	17
6. GRÁFICO DE INVOCAÇÃO DE FUNÇÕES	19
7. BIBLIOGRAFIA.....	21



1. Resumo do Jogo

O copter é um jogo que tem um principal objetivo a pilotagem de um helicóptero através de um labirinto de obstáculos. É um jogo bastante simples graficamente, que permite ao utilizador jogar com o teclado ou com o rato. Então, para fazer subir o helicóptero o jogador pode, ou pressionar a tecla do espaço, ou pressionar o botão esquerdo do rato.

O objetivo do jogo é atingir a maior distancia possível sem bater em numa das margens ou em nenhum dos obstáculos que surgem com o passar do tempo.



2. Descrição do Programa

2.1. Menu Principal



O menu principal do jogo é constituído por 4 botões. O botão "Play" que inicia um novo Jogo, um botão "Exit" que sai do Jogo, um botão "Options" que possibilita ao jogador a escolha do seu "copter" e um botão de "Credits".



2.2. Menu de Opções



O jogador pode assim alterar o ícone do jogo. Ao clicar em “Home” terá acesso de novo ao menu principal onde pode começar a Jogar.

Estratégia Utilizada

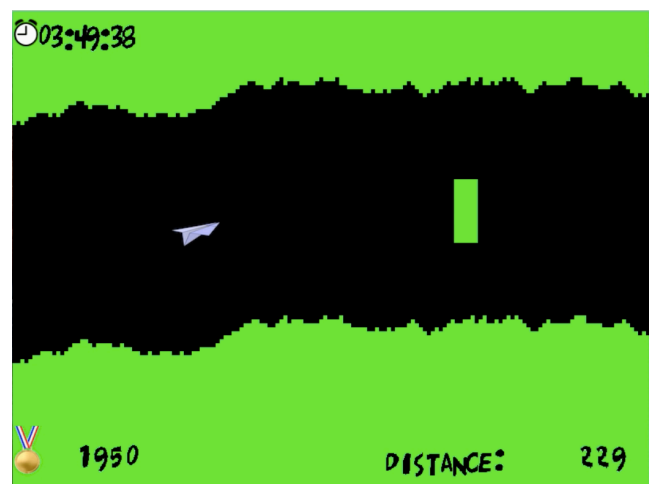
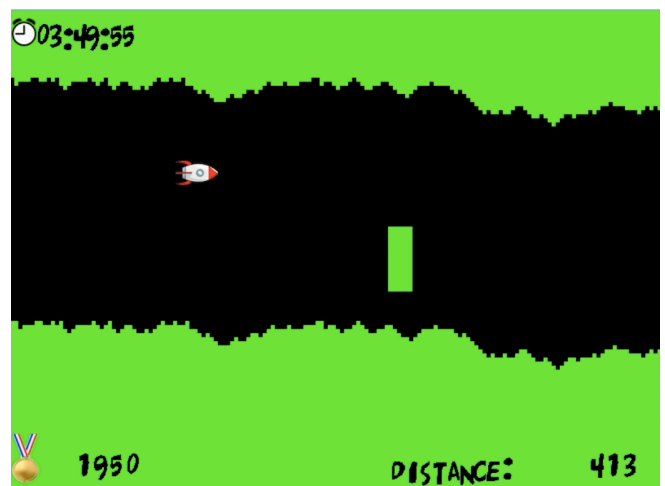
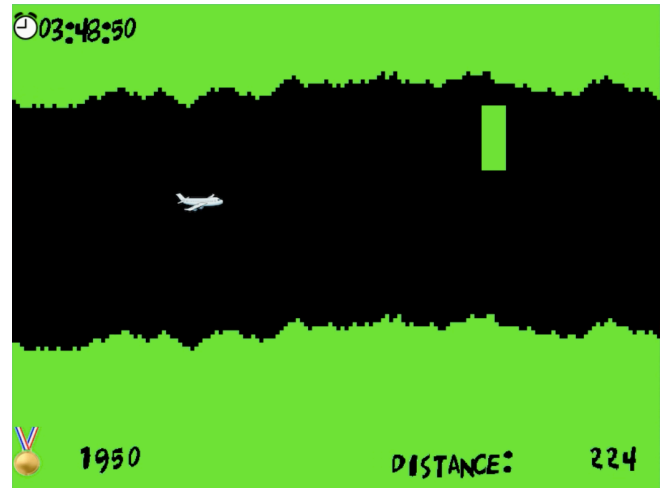
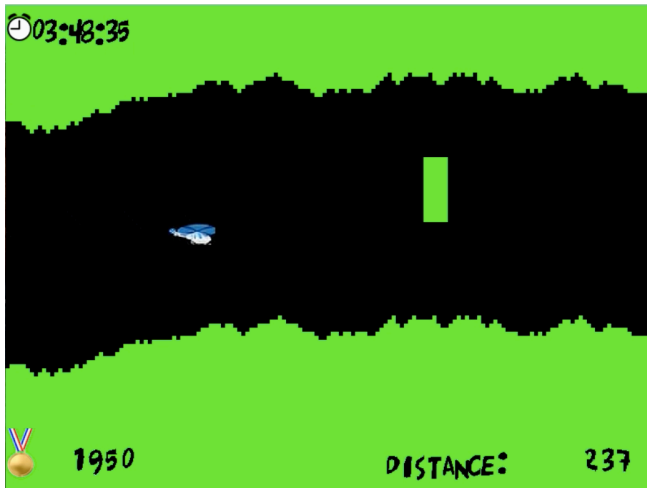
A estratégia utilizada na criação do Menu de Opções foi a criação de uma imagem constituída por 6 pequenas imagens de possíveis helicópteros que distanciam 60 pixéis umas das outras.

Assim, foi criada uma função que recebe um apontador para Bitmap, uma coordenada x e uma coordenada y onde vai ser desenhada a imagem e um char, que significa a imagem respetiva, '0' caso seja a imagem zero, '1' caso seja a imagem um e assim sucessivamente. Seguidamente, o que essa função faz é subtrair o código aschii do char que recebe como argumento com 48 (código aschii do 0), obtendo se assim o número de imagens que se salta. Esse número multiplicado por 60 origina o número de pixéis que o apontador para o início da imagem se terá de descolar, ficando assim a apontar para a imagem pedida.



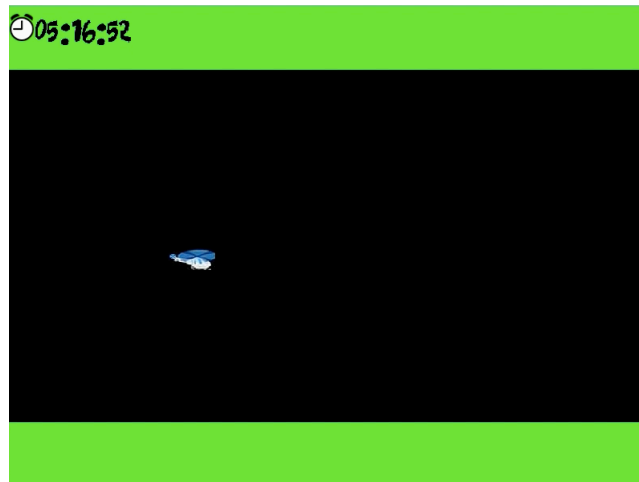


2.3. Gráficos possíveis do Jogo

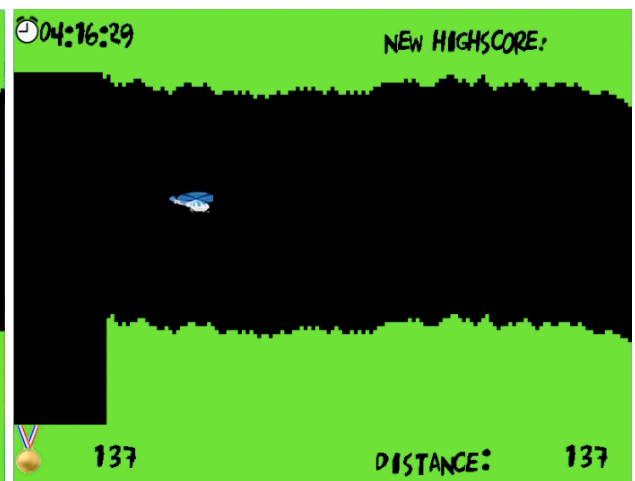
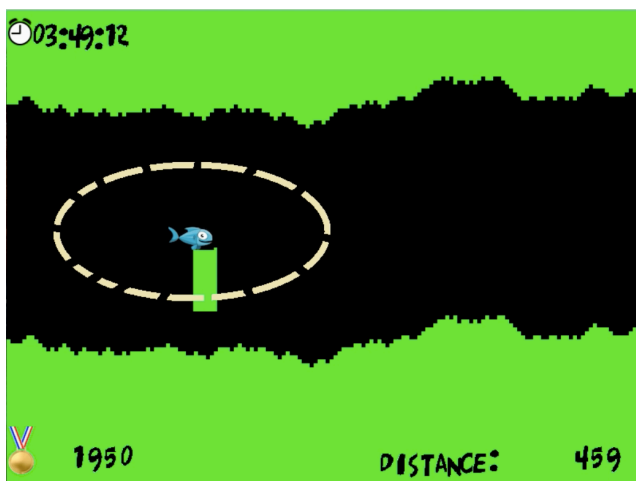




Quando se inicia o jogo, o helicóptero entra num modo de pausa até ser pressionada a primeira vez a tecla do espaço.



Além de ser possível trocar a imagem relativa ao helicóptero, o jogador pode ainda ter acesso ao seu "HighScore", à data atual e à distância percorrida durante o jogo. Sempre que um jogador atinge uma nova "HighScore" uma mensagem aparecerá no lado superior direito do ecrã a avisar do seu novo recorde. Sempre que o jogador perde, ao conduzir o helicóptero para uma margem ou um obstáculo, um círculo aparecerá no local onde o helicóptero embateu.





2.4. Menu de "Game Over"



Assim que o jogador perde, é direcionado para um Menu de "Game Over", onde pode, ou jogar novamente, ou voltar para o menu principal ou sair completamente do jogo.



2.5. Créditos



O jogador pode voltar ao menu principal ao pressionar o botão “Home”.



3. Periféricos Usados

Na realização do projeto foram utilizados todos os dispositivos estudados nas aulas, com exceção da porta de série. De seguida, apresenta-se uma tabela que contém informação acerca das funcionalidades usadas e a maneira como foram tratadas.

Periférico	Interrupção
Timer	Sim
Teclado	Sim
Rato	Sim
Placa Gráfica	Não
RTC	Não

3.1. Timer

Principal função do timer é a construção dos "frames". É também usado para atualizar o helicóptero: as suas coordenadas e a distância alcançada ("SinglePlayer.c", linha 94) e para regular o tempo das imagens, por exemplo na "highScore", sempre que avisa o jogador de um Novo "HighScore" ("GameState.c", linha 43).

3.2. Teclado

É usado na lógica principal do jogo. Para fazer o helicóptero subir o jogador deve pressionar a tecla do espaço, e para fazer descer, deve largá-la alcançada ("SinglePlayer.c", linha 69).



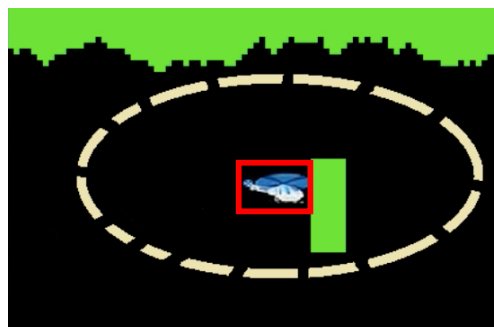
3.3. Rato

É usado na lógica principal do jogo. Para fazer o helicóptero subir o jogador pode, além de pressionar a tecla do espaço, pressionar o botão do lado esquerdo do rato e larga-lo para o fazer descer.

Além disso é usado em toda a parte relativa aos menus do Jogo: menu principal, menu de opções e menu de Game Over ("SinglePlayer.c", linha 81).

3.3. Placa Gráfica

Utilizada para desenhar todos os bitmaps utilizado no programa, tanto no jogo, como nos menus. É utilizado o modo vídeo 0x114, de resolução 800x600. É utilizado o double buffer. As colisões são detetadas por cores.



O código baseia-se num ciclo que percorre toda a imagem do helicóptero, criando uma espécie de retângulo para lá dos limites da imagem do copter, verificando se algum dos seus pixéis exteriores têm a mesma cor que a cor das margens ou que a cor dos obstáculos.

As fontes são usadas através dos bitmaps, foi criada uma imagem com o alfabeto ordenado segundo os seus códigos aschii, sendo que a distancia entre dois char foi declarada como 20 pixéis.

Assim, foi criada uma função que recebe um apontador para Bitmap, uma coordenada x e uma coordenada y a partir de onde é desenhada a imagem e um char, neste caso o char correspondente à letra, número ou símbolo que queremos imprimir



no ecrã de jogo. Como sabemos que o código aschii do primeiro elemento é 48, ó código apenas trata de subtrai o código aschii do char que queremos imprimir por 48. O resultado será o número de letras que teremos que ultrapassar para chegar a letra pedida. Se multiplicarmos esse resultado por 20 e somarmos ao apontador para o início da imagem obtemos um apontador para o início letra que queríamos imprimir.

```
0123456789:;[]=[]?[]ABCDEFGHIJKLMN O PQRSTU VWXYZ
```

3.4. RTC

Utilizado sempre que o jogador inicia uma tentativa de jogo, ao mostrar no canto superior esquerdo as horas atuais ("rtc.c", linha 37).



4. Descrição da Arquitetura do Programa

4.1. Módulos e Descrição

Bitmap

Responsável por desenhar todas as imagens do programa. É constituído por funções de carregamento de imagens, impressão de imagens com e sem fundo.

Main

Como o nome indica, é o módulo que chama a função que executa o jogo. É responsável por chamar as funções de "subscribe" e "unsubscribe", "vg_init" e "vg_exit", carregamento de algumas imagens e remoção das mesmas. Além disso, chama uma função que é responsável por abrir um ficheiro chamado HighScore.txt e guardar a pontuação lá escrita numa variável.

Button

Criação de um botão e de uma função que testa se este foi ou não pressionado pelo rato.

Copter

Criação de um helicóptero e funções de atualização do mesmo, por exemplo a posição. É constituído por funções como desenhar helicóptero, funções que incrementam a distância percorrida e que verificam se o jogador bateu ou não um novo recorde.

GameState

Contém funções que atualizam os dados do helicóptero, e permitem escrever a cada novo "frame", os dados já atualizados (coordenada x e y, distância, "highScore"). Também é responsável por atualizar o relógio, por desenhar as margens e os obstáculos.

HighScore.txt

Ficheiro que contém a melhor "highScore".



KBD

Contém grande parte das constantes presentes no programa.

Keyboard

Responsável pelas funções que se relacionam com o teclado.

MainMenu

Contém todos os Menus existentes no programa (Menu de "Game Over", Menu de Opções e Menu de Créditos) assim como as funções que se relacionam com cada menu (construtor, destrutor e carregamento de imagens).

Margins

Responsável por desenhar as margens do programa.

Menu

Funções de subscribe, unsubscribe, carregamento e remoção de imagens.

Mouse

Responsável pelas funções que se relacionam com o rato.

Obstacles

Responsável por desenhar os obstáculos do programa.

Rtc

Responsável pelas funções que se relacionam com o rtc.

RTCC

Contém constantes relativas ao rtc.

Singleplayer

Contém o ciclo principal que corre o jogo. Foi implementada uma máquina de estados. Sempre que o helicóptero embate contra uma margem ou um obstáculo entra no estado "HIT; antes de iniciar o jogo, ou seja antes de ser pressionada a primeira vez a tecla do espaço, encontra-se no estado "WAITING". Caso contrário, e se ainda estiver em jogo, encontra-se em modo "NO_HIT". É aqui que se chamam as funções que atualizam o helicóptero, as suas coordenadas, distância percorrida, a hora atual e a "HighScore".



Timer

Responsável pelas funções que se relacionam com o timer.

vbe

Responsável pelas funções que se relacionam com a placa gráfica. Funções que guardam informação sobre o modo de vídeo atual.

vídeo_gr

Responsável pelas funções que se relacionam com o carregamento e tratamento de imagens e frames.



4.2. Estruturas Criadas

- Singleplayer
- Copter
- MouseInfo
- Obstacles
- Button
- MainMenu
- MenuGameOver
- OptionsMenu



5. Estado do Projeto

5.1. Instalação

Para iniciar o programa basta entrar na pasta `proj` e, como super user (su) escrever os seguintes comandos: **sh install.sh** (copia os ficheiros de imagem para um ficheiro específico e compila o jogo), e depois **sh run.sh** (correr o programa).

5.2. Possíveis Melhorias

Como previsto, não foi implementado o periférico da porta de série. Embora tenha sido referido na especificação do projeto a opção de jogo de "Multiplayer", não foi possível a sua implementação. Apercebemo-nos que seria impossível fazer subir os dois helicópteros ao mesmo tempo, pois como só é possível ler um "scancode" de cada vez, este modo de jogo só funcionaria se os dois jogadores pressionassem nas teclas desfasadamente, o que deixaria de fazer sentido.

Seria também interessante criar a opção de mudar a cor das margens, opção essa que não foi implementada por falta de tempo.

5.3. Empenho dos Elementos do Grupo

Os elementos do grupo envolveram-se de forma igual na realização do trabalho.



5.4. Detalhes de Implementação

As aulas de Lcom foram bastante benéficas para a aprendizagem dos periféricos e, por isso bastante úteis para o projeto. Pensamos, no entanto, que tanto o RTC como a porta de série deveriam ter sido mais explorados durante as aulas.

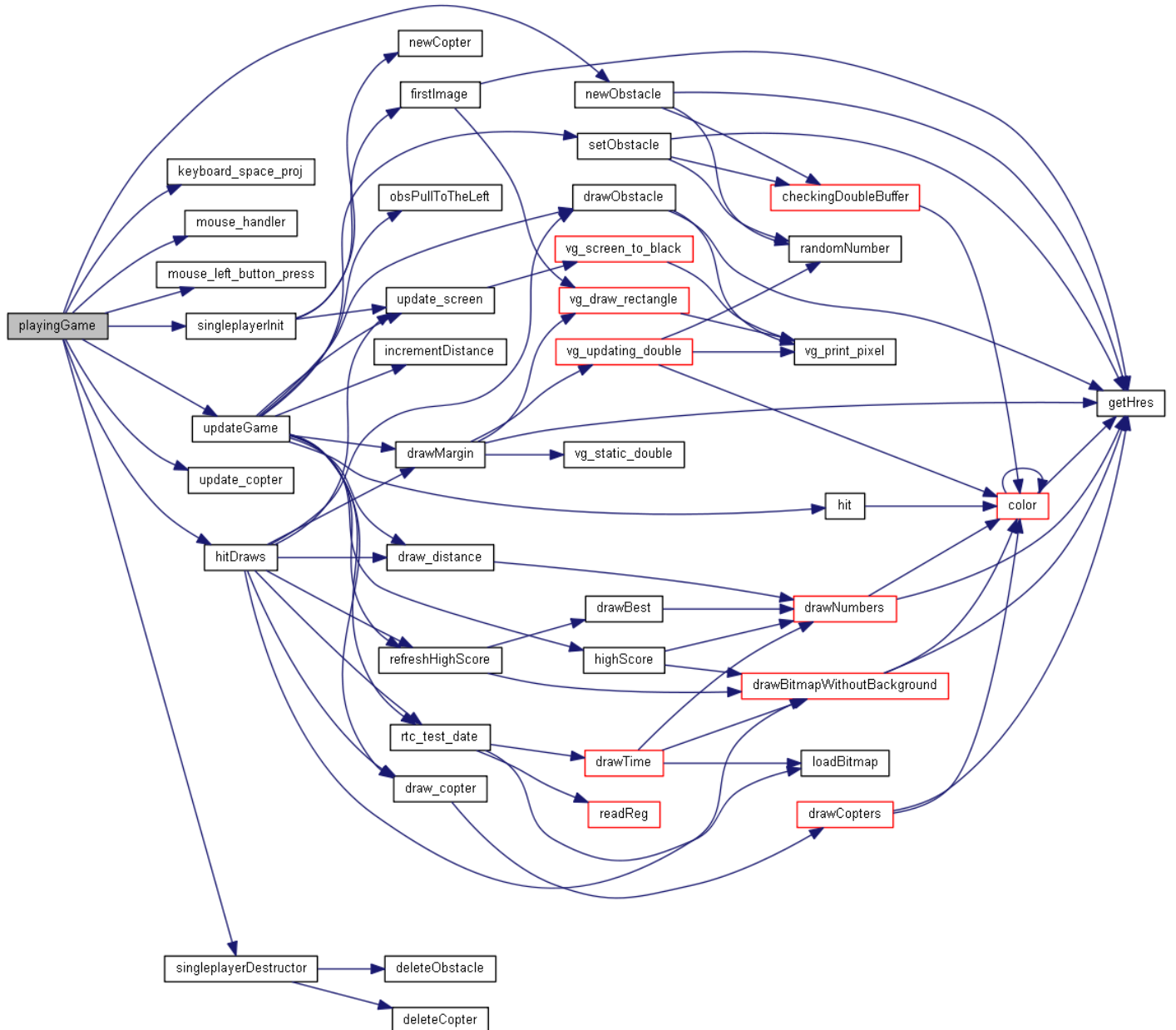
5.6. Dificuldades Sentidas

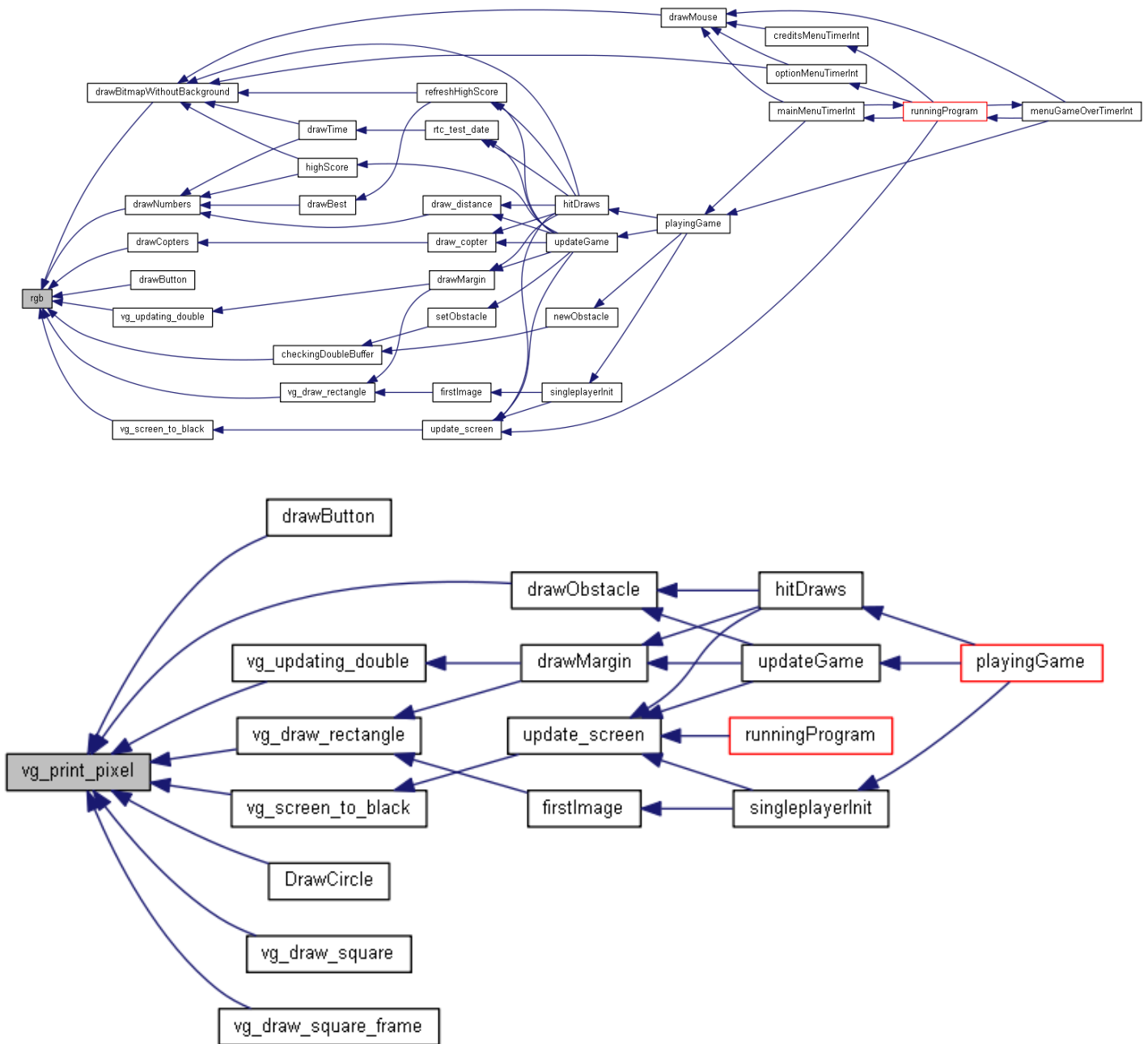
Deparámo-nos com bastantes dificuldades no que diz respeito à gestão de memória e também na implementação dos bitmaps.

A criação das margens foi também uma tarefa complicada. Inicialmente foi criada uma estrutura de nome margens, mas, por trazer bastantes problemas no jogo, decidimos fazer uma implementação sem objetos margens, mas, escrevendo diretamente no vídeomem tentando manter a lógica do jogo, implementação essa que resultou.



6. Gráfico de Invocação de Funções







7. Bibliografia

O código relativo aos bitmaps (vídeo_gr), mais especificamente a função de carregamento de imagens foi inspirada do blog do aluno Henrique Ferrolho: <http://difusal.blogspot.pt/2014/09/minixtutorial-8-loading-bmp-images.html>, baseado na página <http://forums.fedoraforum.org/archive/index.php/t-171389.html>.