U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Relatório do Projeto

Ultimate Racing

Laboratório de Computadores 2018/2019

Turma: 6

Grupo: 03

Gustavo Torres (up201706473)

e

Joaquim Rodrigues (up201704844)

Índice

1. Instruções de utilização do programa 3

1.1 Ecrã Inicial (Menu Principal) 3

1.2 Menu de Escolha de Nível 4

1.3 Ecrã de Jogo 5

1.4 Menu de Pausa 6

2. Estado do Projeto 7

2.1 Video Card 7

2.2 Keyboard 8

2.3 Mouse 8

2.4 RTC 8

2.5 UART 8

2.6 Timer 8

3. Estrutura/Organização do código 10

3.1 Bitmap 10

3.2 Camera 10

3.3 Car 10

3.4 Circle 10

3.5 Game 10

3.6 GameObject 11

3.7 Keyboard 11

3.8 List 11

3.9 Mouse 11

3.10 MouseD 11

3.11 Numbers 11

3.12 Obstacle 12

3.13 Pista 12

3.14 Powerup 12

3.15 Proj 12

3.16 Quadtree 12

3.17 Rectangle 12

3.18 RTC 13

3.19 Serial_port 13

3.20 Sprite 13

3.21 Tile 13

3.22 Timer 13

3.23 Vector2d 13

3.24 Video_gr 14

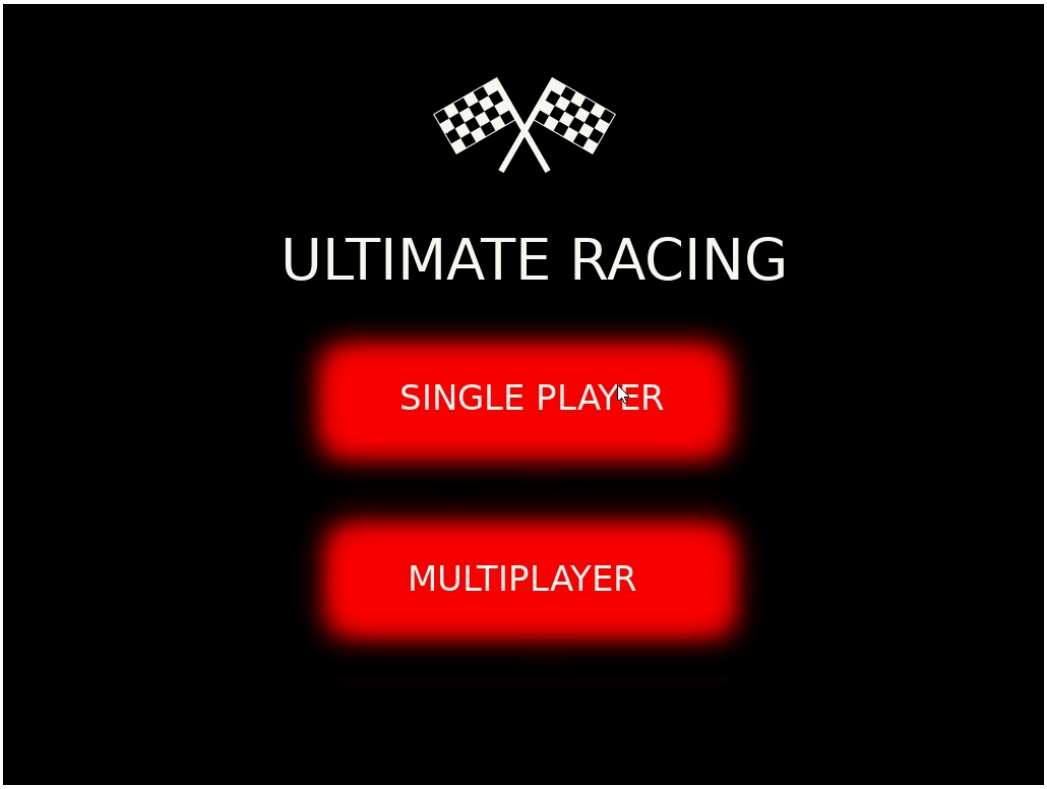
3.25 Function call graph 15

4. Detalhes de implementação 16

5. Conclusão 17

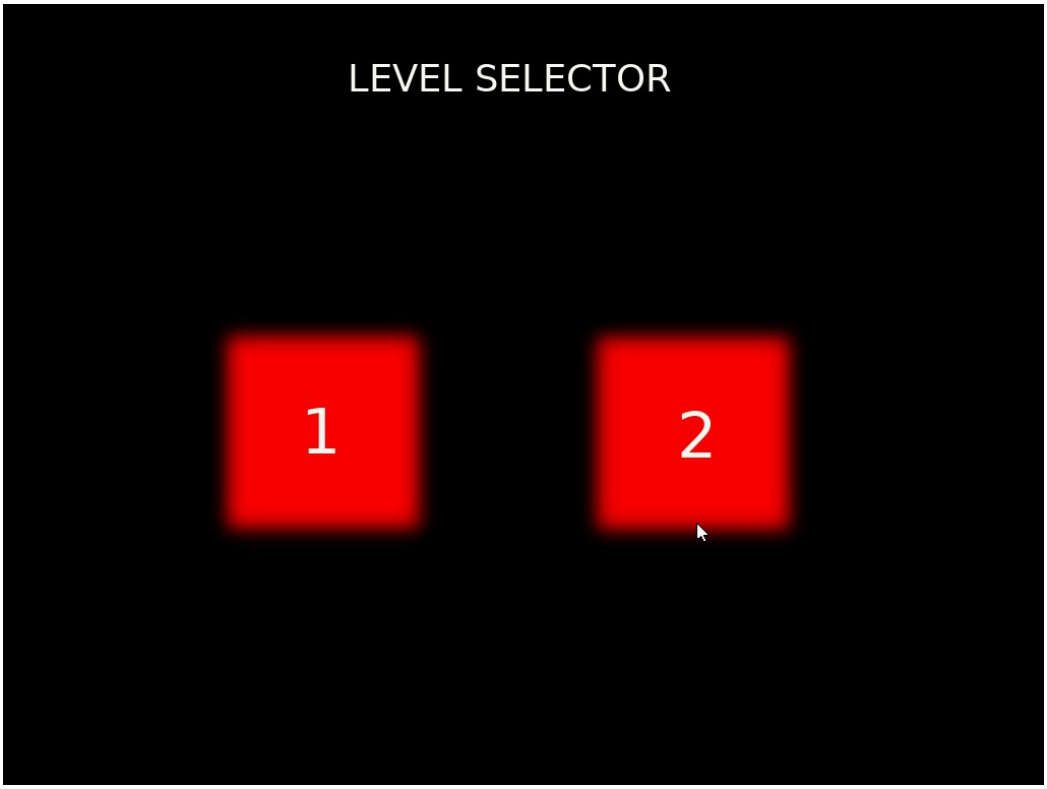
1. Instruções de utilização do programa

1.1. Ecrã Inicial (Menu Principal)



Ao iniciar o programa é apresentado o menu principal do jogo, no qual estão presentes duas opções: “Single Player” e “Multiplayer”, podendo estas ser selecionadas com o rato. Ao escolher-se a primeira, o jogador competirá sozinho contra Bots, enquanto que na segunda poderá jogar com outra pessoa, através da serial-port. Para sair deste menu, basta premir a tecla ‘esc’.

1.2. Menu de Escolha de Nível



Ao escolher uma das opções iniciais, o utilizador tem que escolher qual o nível (pista) a escolher, tendo duas opções: 1 (pista número 1) e 2 (pista número 2), podendo ser selecionadas com o rato. Cada pista terá 3 carros controlados automaticamente (Bots) e 1 carro controlado pelo utilizador.

1.3. Ecrã de Jogo

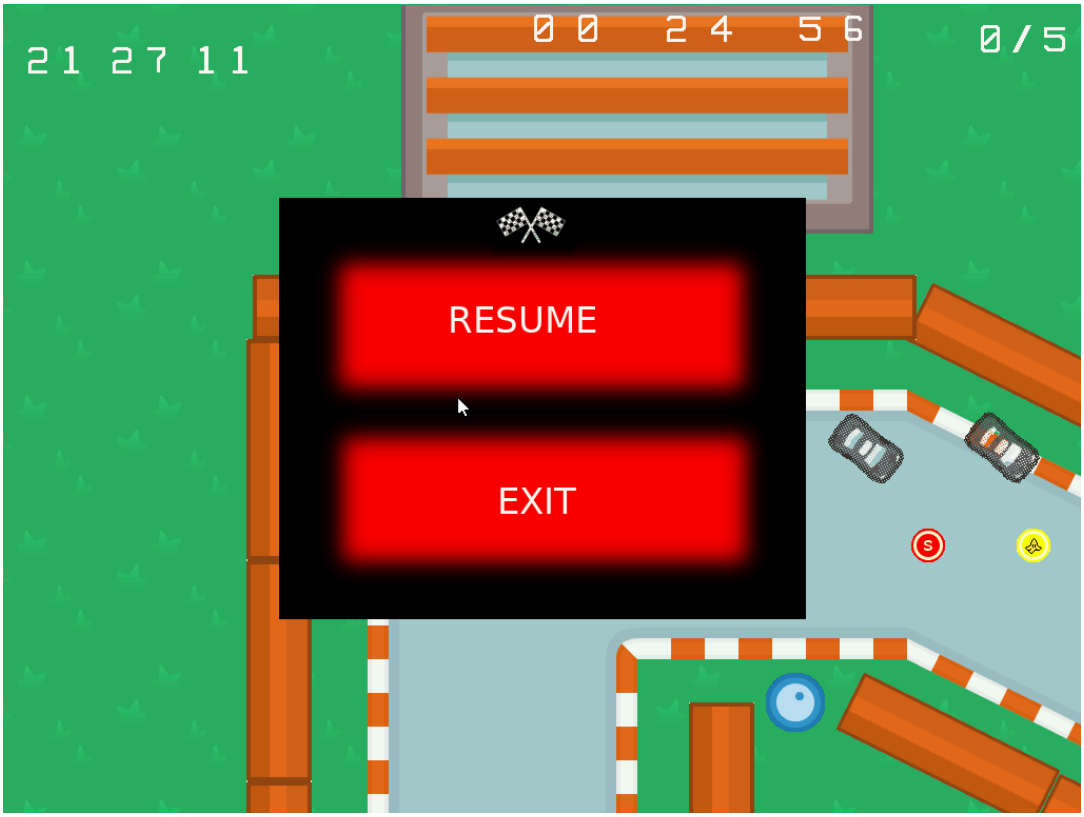
Após escolher uma pista, o utilizador entrará no ambiente de jogo, iniciando a corrida após uma contagem decrescente de 3 segundos. No ecrã estão o número de voltas (completas/totais) no ecrã superior direito e o tempo já decorrido no topo do ecrã. Na corrida (a imagem é um exemplo in-game), existem um conjunto de comandos que o utilizador poderá utilizar:

- Tecla ‘w’: Acelera o carro no sentido do mesmo;
- Tecla ‘d’: Vira o carro para o lado direito;
- Tecla ‘a’: Vira o carro para o lado esquerdo;
- Tecla ‘s’: Trava o carro;
- Tecla ‘esc’: Pausa o jogo, levando-o para o Menu de Pausa (ver ponto 1.4.);
- Premir ‘Mouse1’: Se existir um power-up (ver Power-Ups), utiliza-o, mediante a posição da mira;
- Movimento do rato: Ajusta a mira;

Durante a corrida podem ser usados (como já foi referido), power-ups, os quais após serem apanhados têm um tempo X para serem utilizados:

- Carapaça: É lançada do carro em direção à mira e, se embater num carro, este fica imóvel durante um curto período de tempo;
- Banana: É libertada por um carro e, se algum carro passar na mesma, fica incontrolável durante um curto período de tempo;
- Speed: É utilizado pelo carro, aumentando a velocidade do mesmo (manter premido Mouse1);

1.4. Menu de Pausa



Ao ser premida a tecla ‘esc’ no ecrã de jogo, o utilizador coloca o jogo em pausa, podendo escolher voltar ao mesmo escolhendo a opção “Resume”. Se escolher a opção “Exit”, o jogador abandona o jogo (no caso do Multiplayer, ambos os jogadores abandonam o jogo). Neste ecrã também está apresentada, no canto superior esquerdo, a hora atual.

2. Estado do Projeto

Device	**What for?**	**Interruptions**
Timer	Contar o tempo que um carro demora a completar a pista	Y
Keyboard	Controlar o carro	Y
Mouse	Selecionar opções nos menus e usar power-ups	Y
Video Card	Desenho da pista	N
RTC	Apresentar a data atual	Y
Serial-Port	Função multiplayer	Y

2.1. Video Card

Mencionar modo, resolução, color mode, numero de cores e se usamos triple ou double buffer, colisões, funções de alteração de palete.

No video card foi utilizado o modo 117, direto, com resolução 1024x768, 64K de cores (5:6:5), sendo utilizado double buffer, efetuando uma escrita num buffer secundário e no fim do *tick* é efetuada uma copia desse segemento de memória para a VRAM. São utilizados *sprites* ao longo da aplicação e durante o jogo é feito o uso de um algoritmo de colisão baseado no SAT (*Separating Axis Theorem*) conseguindo colisões pixel a pixel com as formas geométricas básicas que disponibilizamos (rectângulos e circulos). Também são utilizados *sprites* nas representações numéricas de algumas variáveis.

Neste I/O device possuímos três funções de desenho de bitmaps (drawBitmap)(), drawBitmapFast() e drawBitmapAng()), no ficheiro bitmap.c e uma de desenho dos tiles da pista (drawTile()) de importância extrema para tudo o que envolve o projeto, embora existam outras funções de desenho também importantes (drawObstacle(), draw_digit(), drawMouse(), etc.).

2.2. Keyboard

Para o *keyboard* são usadas interrupções e este serve como meio de comunicação tanto no jogo para controlar o carro como para sair de alguns menus. É esencialmente usado no controlo do *gameflow*.

É importante referir a relevância do keyboard, uma vez que tem a função de controlo do carro no Ecrã de Jogo, possuindo também a função de saída do Ecrã Inicial (para o caso de se querer encerrar o jogo). Também existe uma funcionalidade extra,

premindo a tecla ‘l’, que liga e desliga as luzes do carro. Para isto são utilizadas as interrupções deste dispositivo.

As funções mais importantes do *keyboard* são a getInputKbd() (presente no pista.c) que atualiza o input do teclado para a pista e a kbc_ih() (game.c) que atualiza a última tecla premida, permitindo a navegação entre menus.

2.3. Mouse

O mouse é utilizado no navegação inter-menus fazendo uso dos botões e da posicão do rato, bem como durante a execução do jogo fazendo usos também da posição e botões.

Nos menus é usado para saber quando e qual botão é clicado para poder ir para um *gameState* diferente. Durante o jogo, a posição é usada como mira e o botão é usado para utilizar os *powerups* quando estes estão disponíveis. Para isto são utilizadas interrupções do dispositivo.

As funções mais importantes do *keyboard* são a getInputMouse() (presente no ficheiro pista.c) que atualiza o input do rato para a pista e a mouse_ih() (game.c) que permite a navegação entre menus.

2.4. RTC

O RTC é utilizado como modo de o utilizador saber a hora enquanto joga, podendo ser consultada no Menu de Pausa, na possibilidade de este se perder nas horas . Também existe a funcionalidade, que pode ser ativada aquando do desenvolvimento do codigo, em que, à medida que a hora vai entrando em regime noturno a pista começa a escurecer e é possível ligar as luzes do carro, no entanto a funcionalidade não foi totalmente desenvolvida e encontra-se muito ineficiente.

São utilizadas interrupções periódicas.

Uma função de relevo deste I/O device é a função getDate(), presente no ficheiro pista.c.

2.5. UART

A UART é usada na comunicação entre computadores na funcionalidade multiplayer. São utilizadas interrupções e é feita uma comunicações nos dois sentidos, em que o computador que começa por tentar estabelecer comunicação, e quando esta é estabelecida, este compuatdor é definido como *server* ou *master* e o outro como *client*

ou *slave*. No server é feita toda a lógica de jogo e este manda as informações pertinentes ao outro computador para o manter atualizado. O client envia possíveis inputs ao server para este poder simular o jogo.

Uma vez que a comunicação é lenta e o jogo de carros é um jogo rápido tivemos diversos obstáculos e o resultado obtido não foi o melhor. A comunicação servidor-cliente é feita em diferentes frequências, o server manda a posição do carro a cada 3 ticks, enquanto que a posição do client é feita a uma frequência menor com o objetivo de apenas corrigir a posição simulada no client para a do server, sendo apenas a cada 5 ticks. Se se diminuir os ticks, nota-se um atraso na comunicação que se vai acumulando ao longo do tempo.

A informação enviada é apenas dos carros e por questoes de eficiência só existem 2 carros e não existem powerups. A informação é então enviada em pacotes de 4 bytes, em que as posições x e y do carro ocupam cada uma 16 bits do primeiro pacote.

No segundo pacote são enviados 16 bits do ângulo e 8 bits de cada uma das componentes da velocidade. Devido a conversões de ordens de grandeza e de *data types* é preciso fazer uma manipulação de bits de forma a tentar não perder informação e também ser eficiente. Assim são enviados 2 pacotes por carro. Suspeita-se que o atraso da comunicação se deva não só à comunicação lenta da UART mas também à computação feita pelo resto do programa.

O estabelecimento da comunicação é feito através da troca de caracteres. O primeiro computador, quando tenta estabelecer comunicação, envia o caracter ‘?’, se o segundo computador tentar estabelecer comunicação e receber um ‘?’, envia então um ‘y’, ficando à espera que o primeiro escolha a pista. Quando a pista é escolhida, é enviado o número da pista e ambos os compuatdores a carregam.

Deste I/O device destacam-se a função sp1_int_handler() que recebe as interrupções da Serial Port, que posteriormente vão ser interpretadas e atualizadas na pista pela função getInputSp1 (modo multiplayer).

2.6. Timer

O timer possui duas funções no projeto. São utilizadas as suas interrupções e, aquando no Ecrã de Jogo, este possui a função de contar o tempo que o utilizador demora para completar a pista e a função de, a cada *tick*, ser feita a atualização do ambiente de jogo.

É de salientar a importância da função *updatePista()* que depende deste I/O device (ficheiro pista.c).

3. Estrutura/Organização do código

3.1. Bitmap

Neste ficheiro incluem-se as funções que carregam, eliminam e desenham os bitmaps. Estes são carregados de ficheiros ".bmp". Foi elaborado em igual percentagem pelos dois elementos do grupo mas foi baseado no codigo de um ex-aluno do MIEIC, Henrique Ferrolho, sobre o qual foi autorizada a sua utlização. Foram desenvolvidas funções auxiliares e melhoradas. Corresponde a 9% do projeto final.

3.2. Camera

Neste ficheiro incluem-se as funções que envolvem a struct Camera, taiscomo a sua iniciação e alteração de diferentes parâmetros. Esta struct é responsável pela imagem visível da pista. Foi elaborado em igual percentagem pelos dois elementos dos grupo. Corresponde a 1% do projeto final.

3.3. Car

Neste ficheiro incluem-se as funções que envolvem a struct Car, tais como a sua iniciação, desenho, comportamento (ao receber inputs / interpretação dos sensores, utilização e atualização de power-ups, etc) e libertação de espaço de memória. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 9% do projeto final.

3.4. Circle

Neste ficheiro inclui-se a inicialização da struct Circle. Foi elaborado pelo membro do grupo Gustavo Torres na sua totalidade. Corresponde a 1% do projeto final.

3.5. Game

Neste ficheiro lida-se com a inicialização dos elementos constituintes do próprio jogo, dando-se também a identificação e resposta às diferentes interrupções (de todos os I/O devices). Com o auxílio de uma máquina de estados, diferentes *gameStates* terão respostas e significados diferentes de momentos do jogo. Este ficheiro corresponde a

uma agregação dos diferentes ficheiros “.c”. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 14% do projeto final.

3.6. GameObject

Neste ficheiro iniciam-se diferentes objetos existentes nos jogos (retângulos e círculos), tendo funções que lidam com estes objetos, tais como a sua atualização, diferentes tipos de movimento e colisões (e o seu handle). Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 10% do projeto final.

3.7. Keyboard

Neste ficheiro, elaborado para o Lab3 , encontram-se funções relativas às interrupções do keyboard, como a subscrição, leitura do buffer, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 1% do projeto final.

3.8. List

Neste ficheiro encontra-se uma implementação da estrutura de uma lista, assim como os métodos de inserção, remoção, procura, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 2% do projeto final.

3.9. Mouse

Neste ficheiro, elaborado para o Lab4, encontram-se funções relativas ao mouse, como a subscrição, escrita de comandos, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 3% do projeto final.

3.10. MouseD

Neste ficheiro inicia-se a struct Mouse, havendo outras funções como o desenho, interpretação de packets e update dos mesmos. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 1% do projeto final.

3.11. Numbers

Neste ficheiro lida-se com a inicialização das imagens dos números, assim como o seu desenho para diferentes fins (ex. Hora atual). Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 2% do projeto final.

3.12. Obstacle

Neste ficheiro dá-se a inicialização de obstáculos, a libertação do espaço de memória e o desenho dos mesmos. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 1% do projeto final.

3.13. Pista

Neste ficheiro dá-se a gestão de todos os acontecimentos que ocorrem na struct Pista atual. Inicialização do mapa, tiles, verificação de passagem na meta, posições finais, desenho do nível, verificação de colisões dos carros assim como a libertação do espaço alocado por toda a pista são a os objetivos da maior parte das funções neste ficheiro. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 12% do projeto final.

3.14. Powerup

Neste ficheiro incluem-se as funções que dizem respeito à struct Powerup, assim como a sua inicialização, desenho, ativação e libertação de espaço de memória. Também se inicia a struct Item, desenha, ativa e liberta-se o espaço de memória da mesma. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 3% do projeto final.

3.15. Proj

Neste ficheiro está a função que faz correr o projeto. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 1% do projeto final.

3.16. Quadtree

Neste ficheiro encontra-se a implementação da estrutura de dados Quadtree, contendo métodos de inicialização, inserção, procura, libertação de espaço de memória alocado e interseção. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 6% do projeto final.

3.17. Rectangle

Neste ficheiro inclui-se a inicialização da struct Rectangle. Foi elaborado pelo membro do grupo Joaquim Rodrigues na sua totalidade. Corresponde a 1% do projeto final.

3.18. RTC

Neste ficheiro encontram-se funções relativas às interrupções do RTC como a subscrição, leitura, escrita, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 3% do projeto final.

3.19. Serial_port

Neste ficheiro incluem-se funções relativas à Serial Port como a subscrição, envio e receção informação, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 3% do projeto final.

3.20. Sprite

Neste ficheiro incluem-se as funções que dizem respeito à struct Sprite, assim como a sua inicialização, desenho, ativação e alteração de parâmetros da mesma. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 1% do projeto final.

3.21. Tile

Neste ficheiro incluem-se as funções que dizem respeito à struct Tile, inicializando todos os tiles que possam ser utilizados na pista, possuindo uma função de desenhar e outra de libertar a memória utilizada por cada um. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 3% do projeto final.

3.22. Timer

Neste ficheiro, elaborado para o Lab2 , encontram-se funções relativas às interrupções do timer, como a subscrição, handler, mudança de modo, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 4% do projeto final.

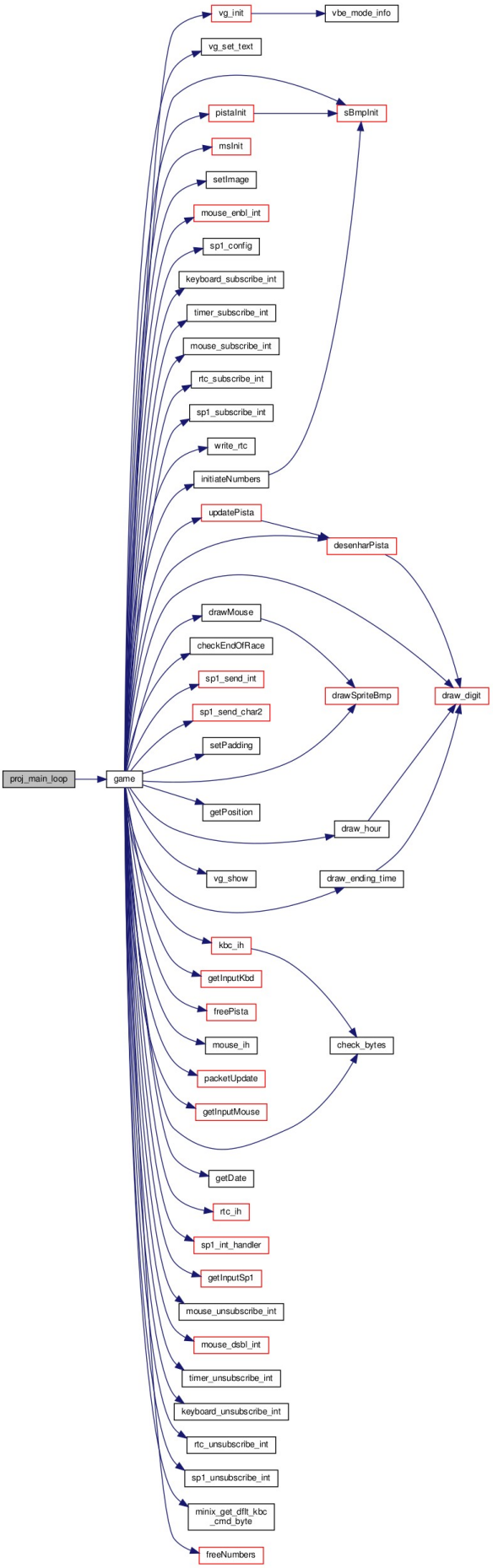
3.23. Vector2d

Neste ficheiro encontra-se a implementação de um vetor bidimensional contendo métodos de inicialização e de operações como rotação, produto escalar, vetorial, soma, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 1% do projeto final.

3.24. Video_gr

Neste ficheiro, elaborado para o Lab5 , encontram-se funções relativas ao Video Card, como a mudança de modo, leitura do modo, desenhar, etc. Foi elaborado em igual percentagem pelos dois elementos do grupo. Corresponde a 8% do projeto final.

3.25. Function call graph



4. Detalhes de implementação

Achamos relevante mencionar neste tópico a facilidade de implementar os diferentes estados no nosso jogo através da utilização de uma estrutura abordada durante as aulas da unidade curricular, a máquina de estados. Apesar disso, achamos também pertinente um estudo mais aprofundado acerca do RTC e da UART, uma vez que a abordagem a estes temas ainda deixa dúvidas na implementação dos mesmos, facto que não ocorre nos outros devices dado o estudo detalhado acerca dos mesmos.

Para a deteção de colisões, utilizamos (como já foi mencionado) o SAT (Separating Axis Theorem) que teve que ser estudado de forma independente, pelo que seria um tópico que teríamos gostado de ter abordado nas aulas da unidade curricular. Outro tópico que gostaríamos que tivesse sido abordado era o modo de carregar bitmaps e não só de carregar e ler pixmaps (uma vez que usamos os primeiros).

No nosso projeto também implementamos uma quadtree como método de guardar, e aceder a dados bidimensionais de uma maneira eficiente. Foi também implementada uma lista como estrutura de dados auxiliar às operações dinamicas da quadtree de aceder e retornar elementos. Foi útil também o desenvolvimento de vetores bidimensionais que facilitaram o uso de coordenadas, velocidades, forças e calculos geométricos.

5. Conclusão

O proveito geral da disciplina é positivo, quer a nível da material abordada como da maneira que esta é abordada e do tempo disponibilizado para tal.

É de salientar um primeiro choque com o desenvolvimento dos periféricos, em que consideramos um bocado dispersa a introdução, onde é explicada a maneira de aceder a funções priveligiadas e de interagir com os periféricos. Uma vez ultrapassada essa fase, a matéria começa a fazer sentido, o que torna possível o aproveitamento da cadeira.

Quanto à cadeira em si, achamos pertinentes os conteúdos lecionados e a sua ordem, bem como os intervalos de tempo em projetos/labs. Concordamos com a metodologia dos mesmos, na medida em que só ao desenvolver é que conseguimos de facto perceber o nosso conhecimento sobre a matéria.

Em relação ao projeto, deparamo-nos com diversos problemas mas que conseguimos superar, o maior desafio foi conciliar a complexidade do projeto com o tempo disponível, mas devido a um bom empenho dos dois elementos e organização de trabalhos, pensamos que o resultado é satisfatório e que foi superado o desafio proposto a nós mesmos.