Janeiro de 2016

BATTLE TETRIS

Laboratório de Computadores - FEUP

António Jorge Aguiar do Vale, up201404572
Telmo João Vales Ferreira Barros, up201405840
TURMA 5 GRUPO 4

Indíce

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

1. Instruções de utilização

Imagens de Apresentação

Ao iniciar o jogo surgem nos primeiros oito segundos duas imagens de apresentação com informação da faculdade e dos desenvolvedores do projeto. Apresentação esta que pode ser "saltada" premindo a tecla ESC.

Menu

Fig. 1 – Menu

O menu principal é constituído por 6 botões que o utilizador pode pressionar usando o rato.

Singleplayer

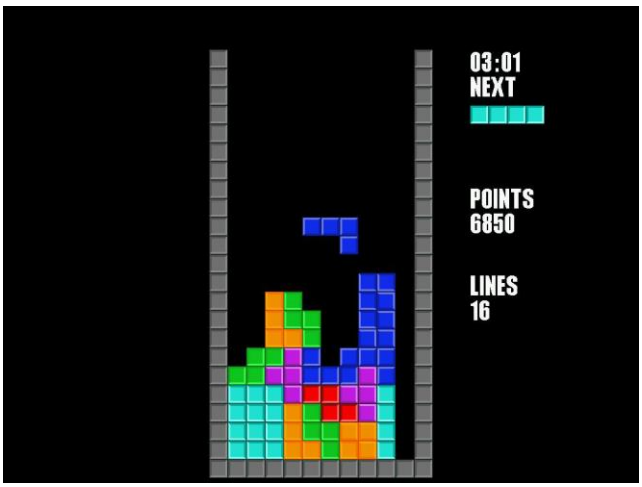


Fig. 2 – Singeplayer

Modo de jogo de Tetris comum (recriação do clássico), peças são sorteadas aleatoriamente e vão descendo uma a uma até atingirem o fundo do tabuleiro, quando combinadas preencherem uma linha completa esta é retirada do tabuleiro e as peças acima descem. O jogo termina e o jogador perde quando as peças atingem o topo do tabuleiro.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

Multiplayer

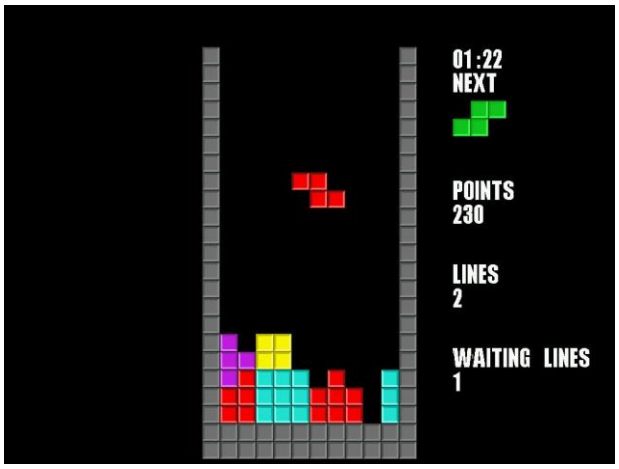


Fig. 3 – Multiplayer

Este modo de jogo tem por base o Tetris Clássico só que é jogado por dois jogadores em máquinas virtuais distintas. Sempre que um jogador completa uma linha, esta é "lançada" para o jogodo jogador adversário e fica em espera até que este deixe cair a próxima peça. O jogador adversário tem a possibilidade de reenviar a linha que recebeu juntamente com uma sua para o adversário caso complete uma linha com a peça que está a jogar atualmente. Assim sucessivamente e o primeiro a atingir o topo do tabuleiro é o derrotado como no jogo normal de Tetris.

Battle

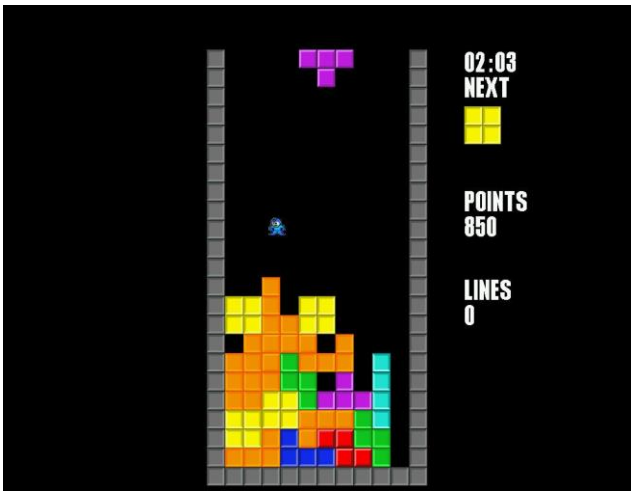


Fig. 4 - Battle

Um jogo é composto por duas rondas. Um dos jogadores é uma personagem na primeira ronda e na seguinte controla a queda das peças, ou vice versa. A personagem encontra-se no fundo do tabuleiro de Tetris e tem como objetivo dificultar a colocação de peças sem no entanto ser atingido pelas mesmas. Ao atingir uma determinada altura o jogador que controla a personagem vence. O jogador que controla as peças tem como objetivo atingir a personagem mantendo-se igualmente focado na finalização de linhas.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

Highscores

Fig. 5 – Sub-menu Highscores

Neste sub-menu são exibidas as três melhores pontuações obtidas no modo de jogo singleplayer.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

How to Play

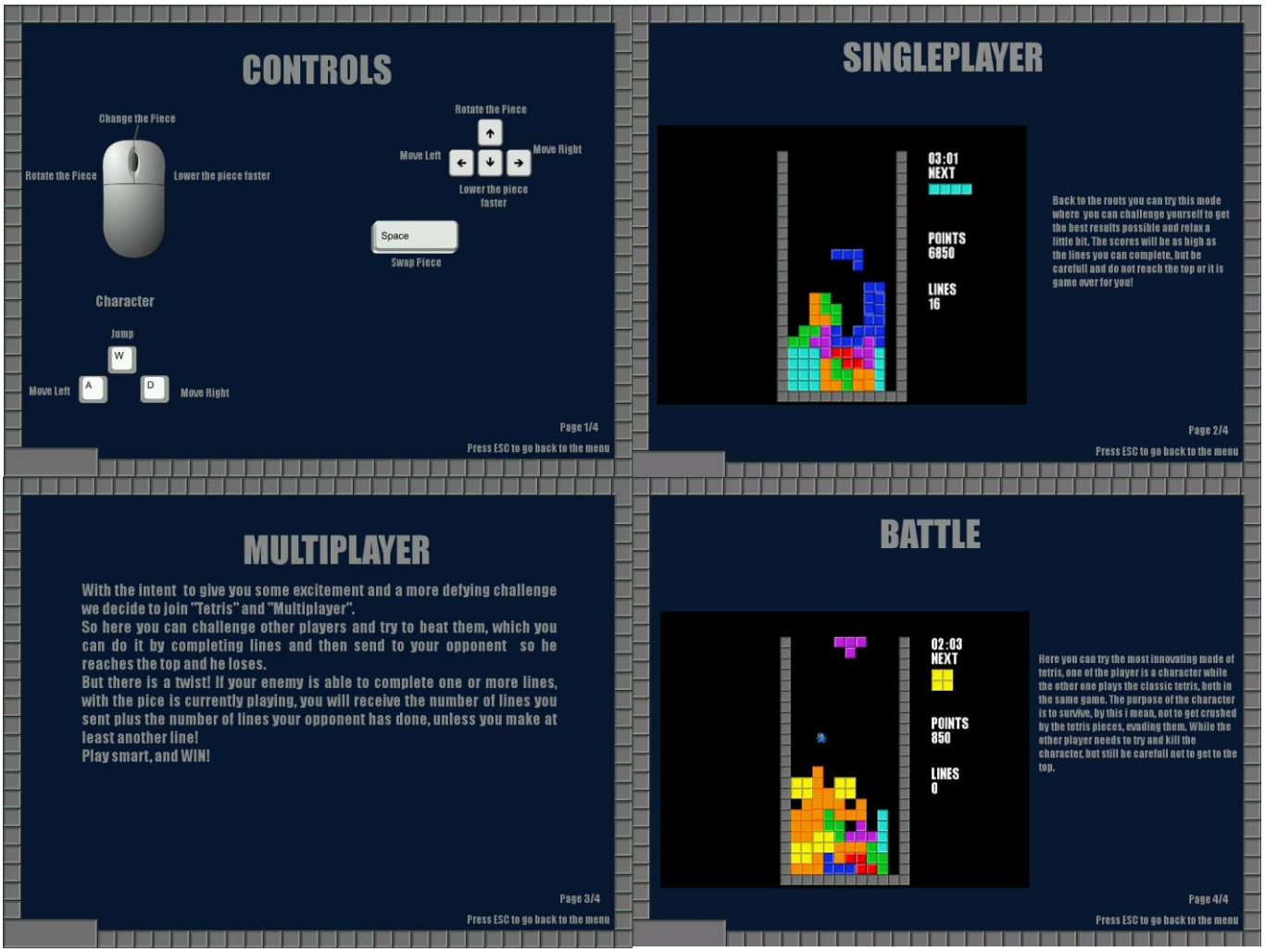


Fig. 6 – Sub-menu How To Play

Neste sub-menu, que permite a navegação pelas suas 4 páginas usando as setas do teclado, encontram-se explicados os controlos do jogo e o funcionamento dos diversos modos de jogo num registo mais informal e apelativo.

Exit

Termina a execução do programa.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

2. Estado do Projeto

Todas as opções do menu estão disponíveis tal como fora planeado e planificado. Contudo na opção Battle, embora seja possível controlar a personagem e as peças, as colisões personagem-peça não estão otimizadas e a jogabilidade devido à falta de tempo fica um pouco comprometida por esse motivo, o resultado final apresentado deste modo de jogo não é o que pretendíamos, no entanto permite já algum entretenimento e foi inserido na mesma pois usamos conceitos como animação de sprites, gravidade e movimento contínuo que não tínhamos usado até ao momento.

Dispositivo	**Uso**	**Modo**
Timer	Controlar taxa de refrescamento de frames Controlar a queda das peças automática Contar tempo de jogo passado	I
Teclado	Mover Peça Mover personagem Sair do sub Menu	I
Rato	Selecionar as opções do Menu Mover Peça	I
Gráficos	Exibir os Menus e o jogo	P
RTC	Ler a hora atual Gravar as três melhores pontuações	P
UART	Enviar o número de linhas finalizadas no modo multiplayer	I

Timer

O timer foi usado ao longo de todo o módulo handler.c e surge em todos os ciclos de tratamento de interrupções, usado para desenhar as imagens a uma taxa de 60Hz.

Ainda no mesmo módulo é usado para gerar eventos que provocam a queda das peças no jogo periodicamente. (timer_event_handler 337 - 341)

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

Teclado

O teclado foi usado ao longo de todo o módulo handler.c e surge em todos os ciclos de tratamento de interrupções. Nos modos de jogo é usado para controlar as peças e a personagem. Nos sub-menus dos Highscores e How To Play é usado para permitir sair do sub-menu.

Para mover as peças cada makecode da tecla respetiva gera um evento. (kbd_event_handler 300 - 322)

Para mover a personagem cada makecode e breakcode da tecla respetiva gera um evento. (kbd_char_event_handler 529-541)

Rato

O rato foi usado ao longo de todo o módulo handler.c e surge em todos os ciclos de tratamento de interrupções.

Nos modos de jogo é usado para controlar as peças.

No menu principal surge desenhado e é usada a sua posição e o clique do botão esquerdo para clicar nos botões que geram eventos tratados pelo módulo do menu.

(mouse_packet_handler/ main_menu_event_handler 148-209)

Gráficos

O modo gráfico usado é o 0x117 que permite uma resolução de 1024x768 pixeis com cores de 16 bits.

O modo gráfico foi usado desde o início até ao término do programa e é controlado no módulo handler.c.

É usado double buffering logo alteramos as funções previamente desenvolvidas para escrever no double buffer em vez de diretamente na memória vídeo. É possível observar isso em todo o módulo vídeo_gr.c. A cópia do buffer para a memória vídeo encontra-se na função vg_buffer() (vg_buffer 265-268).

As colisões também fazem uso do modo gráfico tendo em conta que verificam a cor dos pixéis em determinada posições para verificar se é possível a movimentação da personagem (vg_get_pixel 121-125).

A animação de sprites também é aplicada na personagem do modo battle.

O desenho de frases/letras também é realizado usando uma imagem de um tipo de letra a partir da qual obtemos as letras/símbolos pretendidas. (vg_string 185-230)

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

Outras funções auxiliares para desenhar retângulos, sprites e imagens com e sem cor transparente encontram-se no módulo vídeo_gr.c.

RTC

O RTC é gerido no módulo rtc.

É usado para obter a hora atual. (rtc_current_time 35-79)

Guarda os valores das melhores pontuações nos seus registos que estão disponíveis para alterar. (rtc_get_highscores/rtc_set_highscores 127-158)

Embora seja usado em polled mode a subscrição dos interrupts está pronta. (rtc_subscribe_int/rtc_unsubscribe_int 10-27)

UART

A porta serial é configurada e as suas interrupções subscritas no módulo serial.c. (9 - 45)

Visto que a porta serial apenas transfere o número de linhas realizadas e alguns caracteres como ‘G’ , gameover, ‘R’, ready para estabelecer concordância entre as duas máquinas virtuais, optamos por usá-la por interrupts e de forma simples escrever e ler um caracter por cada interrupt. (handler.c 420 – 535)

Os caracteres são enviados quando uma linha é completada, quando se entra no modo de jogo multiplayer e quando um jogador perde.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

3. Estrutura/organização do código

Toda a documentação do código encontra-se definida no html gerado pelo doxygen e está localizada na mesma pasta do relatório para consulta. As informações referentes a esta secção encontram-se no início de cada módulo.

Como não conseguimos implementar a função driver_receive() para surgir nos gráficos aqui se encontra uma listagem das funções que fazem o seu uso:

- int mainhandler();
- int menu_handler();
- int game_handler();
- int multi_game_handler();
- int battle_game_handler();
- int highscores_handler();
- int instructions_handler();

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

4. Detalhes de implementação

Máquinas de estados e funcionamento do programa

A conceção do funcionamento da máquina de estados foi um processo demorado e optamos por conceber um conjunto de estados para cada módulo que requeira: menu, game e character.

Numa forma simples cada um destes módulos possui a sua máquina de estados que é alterada consoante os eventos atuais gerados a partir do módulo handler.c.

O módulo handler.c é o elo de ligação entre todas as estruturas e é ele que gere e interpreta as interrupções geradas pelos diferentes dispositivos, escreve no buffer, cria as estruturas necessárias,e a partir dos dados que os dispositivos geram e a necessidade da estrutura altera os eventos da mesma. Ex: no decorrer do jogo interpreta o Makecode da seta para o lado esquerdo como o evento LEFTKEY_DOWN e lança-o.

Por sua vez é chamada a função que atualiza o estado do jogo consoante o evento e que neste caso, na eventualidade da peça se poder mover o estado do jogo passaria a MOVE_LEFT.

Quando for chamada a função que atualiza o jogo, esta terá em conta o estado atual do jogo e fará as alterações necessárias que no exemplo dado corresponderia à incrementação da velocidade no eixo do x e posterior atualização da posição.

Modo Battle

A inserção deste modo de jogo só foi possível depois da realização dos primeiros dois modos e como tal optamos por não alterar o código já desenvolvido. Face isto decidimos criar uma estrutura character que corresponde à personagem e funciona como uma estrutura independente que seria capaz de “mover em qualquer tipo de imagem com fundo preto”, isto é, decidimos criar uma personagem independente que se move livremente pelo tabuleiro e que faz a sua ligação com a peça em queda a partir do handler.c. Deste modo evitamos as alterações faladas e permitimos a versatilidade da personagem poder ser inserida noutros contextos.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

Colisões

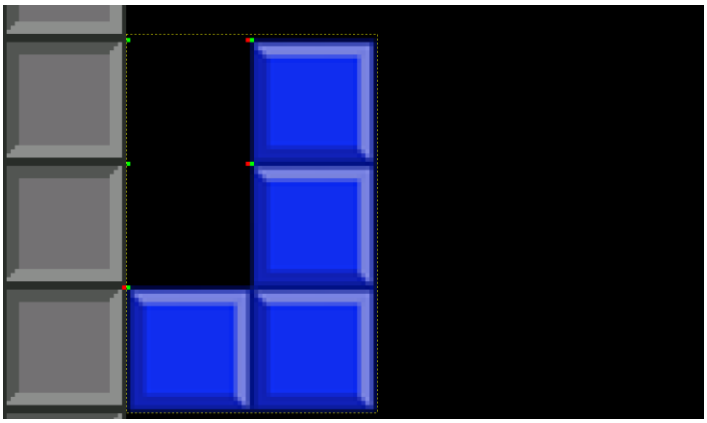


Fig. 6 – Colisão; Peça-Tabuleiro

Peças – Tabuleiro

Ao refletir sobre o assunto das colisões, chegamos a conclusão que objetos usados no nosso "Battle Tetris" eram constituídos por peças com altura e largura múltiplos de trinta e que estes também se moviam trinta pixéis de cada vez, o que nos facilitou o tratamento das colisões. Assim, aquando da resolução das colisões usamos esse aspeto a nosso favor, por exemplo no movimento para a esquerda, verificamos se existe algum bloco a esquerda dos blocos mais a esquerda da peça, isto é, verificamos se o pixel a esquerda do pixel no canto superior esquerdo do bloco é de cor preta, se for pode-se mover. Sendo a cor preta a cor do espaço onde as peças se podem mover livremente. Para o movimento à direita é o mesmo raciocínio, só que em vez de verificarmos o pixel a esquerda verificamos o pixel a direita. Para baixar a peça, verificamos os pixéis abaixo dos blocos.

Personagem – Peças

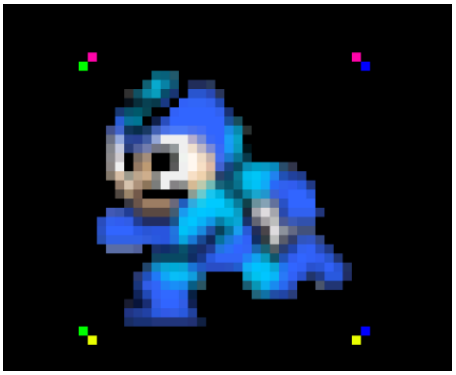


Fig. 7 – Colisão; Caractér

A colisão entre a personagem e as peças do tabuleiro foi simplificada à verificação da cor dos pixels adjacentes às extremidades. Para verificar a possibilidade do movimento para a esquerda são verificadas as cores dos pixels adjacentes ao canto superior e inferior esquerdo. O raciocínio é equivalente para as outras direções e sentidos. Esta não era a verificação que pretendíamos implementar mas é a possível tendo em conta que o tabuleiro não guarda blocos como uma estrutura mas sim cores/pixels.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

Gravidade e saltos da personagem

A gravidade na personagem é concretizada através da incrementação da velocidade no eixo dos y sempre que a personagem não colide com nenhum objeto abaixo de si. Para "fazer" a personagem saltar foi necessário atribuir uma velocidade inicial na mesma direção mas com sentido oposto à gravidade. Foi necessário também estabelecer a proibição do salto quando a personagem tem um objeto em cima de si e quando já se encontra no "ar". Para tal verificamos se a sua velocidade no eixo dos y é nula.

Multiplayer - UART

A porta Serial foi usada por interrupções e através da escrita/leitura de um caracter de cada vez. Esta opção foi escolhida por ser a mais simples e ao mesmo tempo a mais otimizada para a finalidade do nosso projeto visto que unicamente necessitamos enviar caracteres que informam o estado da outra máquina virtual e o número de linhas completadas, número este sempre inferior a 256.

A sua implementação foi realizada da seguinte forma. Quando um jogador entra no modo multiplayer envia um 'M' ao adversário e entra no modo multiplayer aguardando a receção de um 'R' vindo do adversário, indicação que o outro adversário se encontra pronto e o jogo inicia. Do ponto de vista de quem recebe o 'M' o jogo multiplayer é iniciado imediatamente após o envio do 'R'.

No decorrer do jogo sempre que um jogador completa pelo menos uma linha envia o número de linhas concretizadas ao adversário que as lê e incrementa ao seu valor de linhas em espera. O oposto acontece da mesma forma.

Quando um jogador perde o jogo, ou seja, o estado de jogo é gameover, é enviado um 'G' ao adversário informando-o assim que ganhou.

Rotação

Ao pensar sobre este assunto, deparamo-nos com duas possibilidades para resolver este problema. Uma forma mais simples seria ter quatro imagens referentes a cada rotação da peça, mas, por ser mais desafiante optamos por fazer a rotação das peças por código a partir de uma única imagem.

Começamos por alocar memória do tamanho da peça original. Nesse novo mapa escrevemos a imagem original mas percorrendo-a verticalmente, isto é, copiando as colunas da direita para a esquerda para o novo mapeamento. Ao terminar a escrita invertemos as dimensões

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

largura <-> altura e verificamos a possibilidade do seu posicionamento no tabuleiro. Ao confirmar-se alteramos a peça inicial, caso contrário, descartamos as alterações.

Fonts

Fig. 8 - Fonts

De modo a podermos desenhar frases inspiramo-nos na ideia apresentada pelo Henrique Ferrolho na sua aula que lecionou.

Começamos por criar uma imagem com os caracteres entre 38 e 126 da tabela ASCII (escolhemos estes caracteres por serem os mais comuns, incluem os números, maiúsculas, minúsculas e pontuação). Acrescentamos uma linha de pixels no topo que contém um pixel preto no seguimento vertical do início do caracter e um pixel branco no seguimento vertical do final do caracter.

Para desenhar o caracter desejado, usamos um método de pesquisa, que conta o número de pixels pretos da primeira linha da imagem até encontrar o pixel preto com o número do caracter correspondente ao desejado, obtendo assim a "coordenada x" do caracter na da imagem. Seguidamente fazemos a busca do próximo pixel branco e obtemos assim a largura do caracter.

Tendo a posição x inicial do caracter na imagem e a sua largura podemos escrever a secção retangular correspondente no buffer do vídeo.

Fade In

Para criar o efeito das imagens de apresentação, partindo de um ecrã fomos somando 1 (componente de cor vermelha e azul) e 2 (verde) do ecrã até as componentes da cor forem iguais a componentes da imagem.

Fade Out

Foi concluído subtraindo 1 (componente de cor vermelha e azul) e 2 (verde) até que todas as componentes ficassem a preto.

T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

5. Conclusões

Há vários aspetos e diversos detalhes que podem ser retirados deste projeto para que em futuras ocasiões e perante a necessidade da sua implementação possa tudo correr da melhor forma. Contudo consideramos que o resultado obtido desta experiência é manifestamente positivo.

Em primeiro lugar consideramos importante destacar a apresentação de um produto final como desejamos e planeamos. Cumprimos com as metas que estabelecemos e desenvolver o terceiro modo de jogo (Battle), que na nossa especificação estava descrito como uma mera possibilidade na existência de tempo para tal, foi realmente satisfatório pois tivemos oportunidade de introduzir outros conhecimentos que ainda não tinham sido mostrados até então.

Fazendo uma análise mais geral do desenvolvimento do trabalho é possível retirar uma evidente melhoria em execuções futuras e que se prende com a estruturação, organização e documentação do código. Sendo um ponto fulcral em qualquer desenvolvimento de código deveríamos ter dedicado mais atenção a este aspeto desde o início e ir comentando de forma gradual o nosso projeto. Com a ânsia de ver os resultados fomos por vezes negligentes na escrita de comentários o que nos fez dedicar muito do nosso tempo na "reta final" a corrigir essa falha, que foi suprimida com sucesso no entanto.

Unidade curricular

Nós consideramos que esta unidade curricular está bem organizada e o ensino sequencial dos diferentes dispositivos, com posterior aplicação nos labs e projeto final, foi excelente do nosso ponto de vista. Contudo consideramos que os dois últimos dispositivos (RTC e Porta Serial) deveriam ter sido lecionados com mais margem para podermos esclarecer algumas dúvidas presencialmente nas aulas práticas, visto que podem influenciar muito positivamente a nota obtida no projeto final.

Referente ao esclarecimento de dúvidas todos os docentes foram muito prestáveis, quer via presencial no decorrer das aulas práticas, quer pelo sistema de fórum do moodle que consideramos um ótimo meio de comunicação. A introdução de um "professor ajudante" de anos seguintes nas aulas práticas é também uma boa prática pois nos ajudou a compreender alguns erros melhor ainda.

Contudo, gostaríamos de sugerir o melhoramento do acompanhamento dado na primeira e segunda semana letiva, primeiras aulas práticas. Compreendendo que é necessário ter um espírito de autonomia grande e capacidade de auto preparação das aulas, achamos que poderia

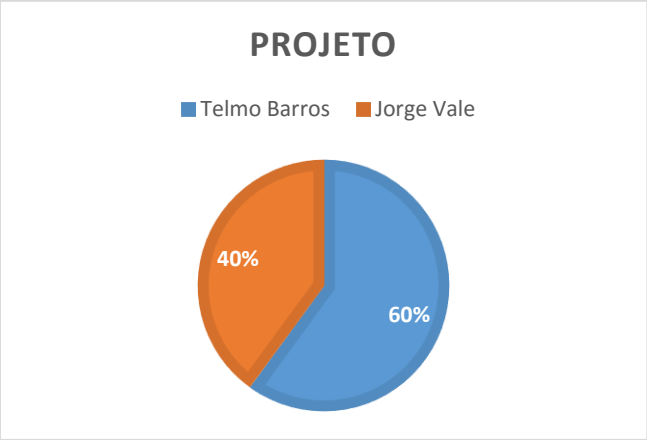
T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840

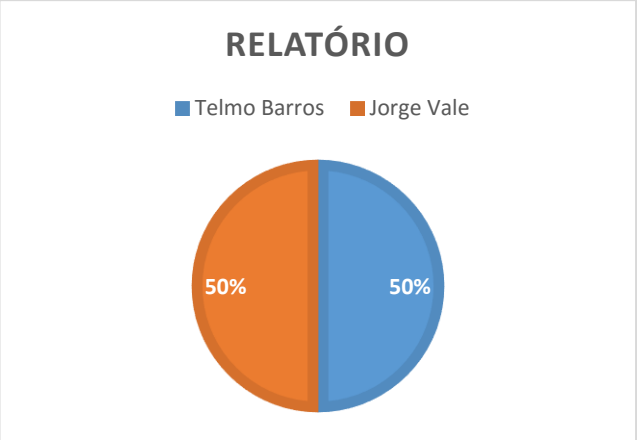
ter sido dada mais atenção e preocupação na adaptação ao ambiente de trabalho (Redmine, VirtualBox, Repositório SVN) pois, no nosso caso, eram conceitos totalmente desconhecidos e mesmo consultando os suportes de apoio fornecidos foi muito difícil a ambientação.

Contribuição

A contribuição no projeto final poderia ter sido mais equilibrada e a percentagem inferior do membro do grupo Jorge Vale deve-se essencialmente às suas dificuldades de adaptação ao ambiente de linguagens de baixo nível, procurando no entanto suprimir as suas dificuldades empenhando-se e realizando outro tipo de tarefas em que não sentia tanta dificuldade.

O relatório foi sendo realizado de forma gradual e igualmente distribuída pelo que consideramos que a contribuição é igual para os dois membros do grupo.





T5G04
Jorge Vale, up201404572
Telmo Barros, up201405840