Mestrado Integrado em Engenharia Informática e Computação

DUELNIX

Um projeto elaborado para a cadeira de Laboratório de Computadores

Prof. Pedro Souto

Monitor: Edgar Passos

Prof. Tiago Boldt Sousa

Turma 4 – Grupo 13

João Almeida - up201504874@fe.up.pt

João Mendes - up201505439@fe.up.pt

2 de janeiro de 2016

Índice

1. Instruções de Utilização……..…..……3
2. Estado do Projeto……8
3. Organização do Código……11
4. Detalhes da Implementação……16
5. Conclusões……..17
6. Apêndice……..18

1. Instruções de Utilização

1.1. MENU INICIAL

Figura 1 e 2 – Menu Inicial

O menu inicial permite a escolha de quatro opções:

“PLAY”, onde se pode entrar no modo de jogo de singleplayer (versus computador).

“PRACTICE”, onde se pode atirar a alvos.

“EXIT”, onde se pode sair do programa.

“MULTIPLAYER” onde dois jogadores podem defrontar-se no mesmo computador.

As opções de “PLAY”, “PRACTICE” e “MULTIPLAYER” podem ser ativadas através do clique no botão esquerdo rato, na zona que aparece envolta a verde (visível na figura 2). O rato é representado pela imagem duma pistola e pode ser controlado livremente (o “ponto” da localização do rato é o canto superior esquerdo da sua imagem).

A opção “MULTIPLAYER” pode ser escolhida carregando na barra de espaço, quando no menu inicial.

Se for carregada a tecla Escape, o programa também sairá, tal como no botão “EXIT”.

As horas aparecem no canto superior direito do ecrã durante toda a duração do programa.

1.2. MODO SINGLEPLAYER

Figura 3 e 4 – Modo Singleplayer (nível 1 e nível 2, respetivamente)

No modo Singleplayer um jogador defronta o computador.

O jogador pode mover-se usando as teclas – “W” move para cima, “A” move para a esquerda, “S” move para baixo e “D” move para a direita.

O rato serve para o jogador disparar. Só pode disparar uma bala de cada vez (pode disparar outra quando não houver nenhuma outra bala sua no ecrã). O jogador dispara carregando no botão esquerdo do rato, só sendo aceite o clique se este for “à frente” do seu cowboy e dentro do retângulo castanho, o recinto do jogo. O cowboy não pode ultrapassar o obstáculo nem colocar-se acima ou abaixo do mesmo.

O jogador pode sair do jogo a qualquer momento pressionado a tecla Escape, voltando ao menu inicial.

O objetivo é chegar aos 8 pontos, no primeiro e segundo níveis, antes do cowboy adversário. A dificuldade aumenta pela mudança de obstáculo no meio de ecrã e a velocidade da bala do cowboy controlado pelo computador aumenta.

Se as balas embaterem no obstáculo são “absorvidas” por ele desaparecendo do ecrã. Se atingirem o topo ou fundo da zona de jogo, a bala resvala e inverte o seu sentido vertical, mas se atingir o lado oposto sem atingir o adversário ou o obstáculo, desaparece. O obstáculo é fixo no nível 1 (cato) e no nível 2 o obstáculo passa a ser móvel (diligência), movendo-se na vertical. O obstáculo deve ser usado como um escudo.

Atingir um adversário dá um ponto. O resultado aparece no topo do ecrã e é restabelecido após a passagem de nível.

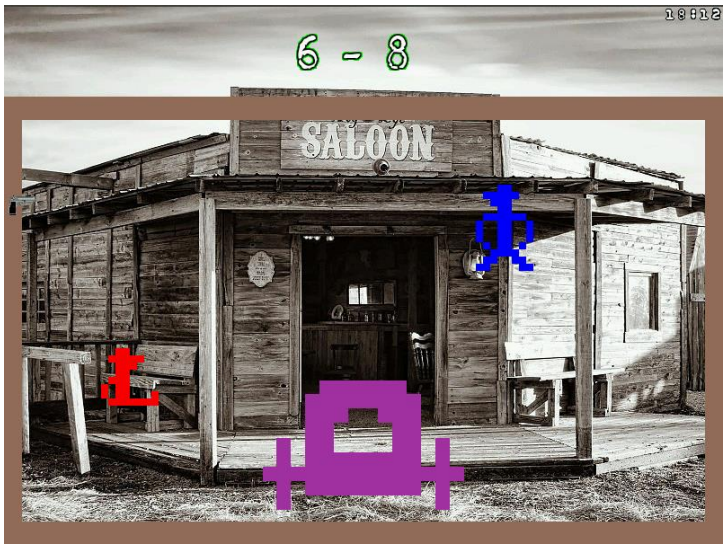


Figura 5 e 6 - Modo Singleplayer (nível 2 e nível 3, respetivamente)

Na figura 5 vemos um utilizador que acabou de perder no nível 2 e por isso, foi derrotado. Podemos ver o seu cowboy "morto" e nos 3 níveis, no caso de derrota, o ecrã fica "preso" à espera que o utilizador carregue na tecla Escape para voltar ao menu inicial.

No nível 3 a velocidade da bala do adversário do utilizador (cowboy azul) é mais uma vez aumentada, e a velocidade de movimento do obstáculo também é aumentada. Assim, o jogo é consecutivamente dificultado.

Na figura 6 observamos o nível 3 onde o resultado é mais uma vez reposto (0-0) e o utilizador dispõe de 30 segundos (decrementados) para derrotar o computador. Se não conseguir (i.e., empatar ou perder) o cowboy morre e o jogo espera pela tecla Escape. Se eventualmente conseguir, mostra o ecrã de vitória e espera pela tecla Enter. Ambas as teclas redirecionam para o Menu Inicial.

Como o jogo é muito difícil de bater, não dispomos de imagens do ecrã de vitória…

1.3. MODO PRACTICE

Figura 7 e 8 – Modo Practice

No modo “PRACTICE” o jogador pode fazer isso mesmo, praticar a forma de jogar, treinar a sua mira, “skillshots” (tiros com uma ou mais tabelas) e aperceber-se do ângulo a que deve disparar.

Este modo dispõe de um contador de tiros acertados, que quando atinge o valor 5, o obstáculo é alterado de um cato fixo para uma diligência móvel. Existe também o botão “READY” (mais visível na primeira imagem) para o jogador voltar ao menu inicial. Isto também é alcançável pressionando a tecla Escape.

Sempre que o cowboy atinge o alvo, este é relocalizado na parte direita do ecrã. A relocalização é efetuada recorrendo ao “random” ao invés do alvo ser móvel. Optamos por essa via ao invés de um alvo móvel para o utilizador poder efetivamente treinar a mira (alvo móvel seria obtido da mesma forma que o obstáculo móvel, por isso foi meramente uma opção do grupo e não uma dificuldade).

Chegando aos 12 tiros acertados (exibido no lado esquerdo do ecrã), o utilizador é levado automaticamente para o menu inicial (esta opção foi tomada devido à limitação de tempo na demonstração do projeto, sendo que seria possível - apenas retirar um if no código - o jogador pratique por tempo indefinido)

1.4. MODO MULTIPLAYER

No modo multiplayer, dois jogadores podem defrontar-se no mesmo computador. Devido à indisponibilidade do rato para este módulo (não é, normalmente, possível ter doies ratos no mesmo computador – nem seria possível, provavelmente, obter interrupções dos dois), decidimos usar o teclado.

O jogo funciona normalmente, como no modo SINGLEPLAYER (1.2), só que o segundo jogador é controlado por outro utilizador.

Jogador 1:

- "Z" desloca para a esquerda, "X" desloca para baixo, "C" desloca para a direita e "S" desloca para cima.
- "Q" dispara para cima, "W" dispara em frente, "C" dispara para baixo.

Jogador 2:

- "," - ";" desloca para a esquerda, "." - ":" desloca para baixo, "-" – "_" desloca para a direita e "L" desloca para cima.
- "I" dispara para cima, "O" dispara em frente, "P" dispara para baixo.

É recomendável que cada jogador use uma mão, colocando nas teclas de movimento e com um dedo ou rapidamente deslocar a mão para as teclas de disparo, voltando de imediato para as teclas de movimento.

A qualquer momento pode sair-se do jogo usando a tecla Escape, terminando o jogo e voltando ao menu inicial.

Quando um dos jogadores chega aos 5 pontos, o obstáculo cato fixo é alterado para uma diligência móvel, dificultando o jogo. O primeiro a chegar aos 12 pontos vence e o cowboy do adversário "morre". O ecrã fica nesse estado até ser pressionada tecla Escape, voltando ao menu inicial.

A velocidade da bala é alterada chegando os 5 pontos. O jogador que lá chegar primeiro tem uma vantagem até ao outro jogador chegar a essa pontuação.

2. Estado do Projeto

Dispositivo	Utilização	Interrupções
Timer	Atualizar o estado do jogo e refrescar o ecrã	**SIM**
Teclado	Interação do utilizador com o jogo e usado para sair de modos de jogo e do programa	**SIM**
Rato	Interação do utilizador com o jogo (movimento e posição), e usado para sair de modos de jogo e do programa.	**SIM**
Placa Gráfica	Mostrar ao utilizador o que está a acontecer. Desenho de imagens (bmp, no caso).	**NÃO**
Real Time Clock (RTC)	Mostrar a data durante a execução do ecrã	**SIM**

O programa tem um único ciclo de interrupções (com driver_receive()), que passa a informação e atualiza todo o jogo. Este ciclo encontra-se na função updateDuelnix(). As interrupções dos 4 dispositivos são recebidas e as suas variáveis atualizadas.

As funcionalidades do jogo foram todas cumpridas (as da especificação) e uma funcionalidade adicional, que foi o modo multiplayer. O modo serial port não foi possível devido à falta de tempo após o bom funcionamento dos módulos principais do programa.

2.1. TIMER

O timer foi utilizado para refrescar o ecrã e implementar o nível 3 do modo singleplayer que tem um temporizador – updateGameState() atualiza uma variável ticks que conta as interrupções e gere esse temporizador. O ecrã é refrescado a cada interrupção do timer (tick), logo é atualizado a 60 frames por segundo. Não só o ecrã, mas toda a informação "atual" do jogo é atualizada pelo timer.
Decidimos não usar a função timer_set_square() – que está no módulo do timer, assumindo que o timer já se encontra configurado dessa forma.

2.2. TECLADO

O teclado é usado em todo o programa, e cada modo de jogo recebe o scancode atualizado no ciclo de interrupções (updateGameState(), updateMainMenuState(), updatePracticeState(), updateMultiplayer()). Com esta informação, cada modo de jogo gere a sua informação conforme o scancode recebido. É usado para sair dos modos e voltar ao menu inicial e para os controlos do jogo (sendo utilizado juntamente com o rato), nomeadamente a posição do cowboy. A função tick_delay não foi usada para ler do KBC para um desempenho mais rápido, o mesmo acontece no módulo do rato. A leitura dos packets é feita por updateMouse() recorrendo ao mouse_int_handler() para ler os packets recebidos.

2.3. RATO

O rato é usado como num sistema operativo, aparecendo no ecrã durante todo o programa. É criada a sua estrutura e ao longo do programa, cada interrupção altera essa estrutura com a posição do rato e o botão esquerdo. É usado um pouco por todo o programa com a função getMouse(). Utilizado para o clique em botões, e para os controlos do jogo, nomeadamente para disparar, em todos os módulos do jogo exceto o multiplayer. O disparo e os botões usam o movimento e a posição do rato em simultâneo. De notar, é a fluidez do movimento do rato.

2.4. PLACA GRÁFICA

A placa gráfica é o essencial do projeto, porque sem ela o nosso programa não poderia comunicar o utilizador, mas tivemos um pouco de dificuldade com a mesma de início. Utilizamos a técnica de triple_buffering para atualizar o mouse mais frequentemente e este ficar sempre por cima das imagens (bitmaps) desenhadas no double_buffer. Depois é que copiamos o triple_buffer para a placa gráfica que exibe as imagens ao utilizado. O page flipping não é feito por VBE, mas sim “manualmente” como referido.

Usamos “sprites” com animação, e deteção de colisões através das imagens dos objetos (bitmaps).

Usamos o modo direto RGB 5:6:5 0x114 (800x600 pixels).

A placa gráfica é atualizada a cada interrupção do timer.

A edição de imagens foi feita usando o programa GIMP.

2.5. REAL TIME CLOCK (RTC)

O RTC é atualizado com interrupções (subscrito como o rato, teclado, timer – com pedido de interrupções).

Lemos os registos (A e B) do RTC com modo 24 horas, de modo a atualizar a estrutura que contém a hora atual. O registo C é também limpo.

3. Organização do código

3.1. Bullet

Este módulo cria "objetos" do tipo Bullet que representa uma bala (velocidade, localização). É usado para criar (quando se dispara), desenhar no ecrã, atualizar a posição (recorrendo à velocidade) e apagar quando colide ou desaparece do ecrã.

Responsável: João Almeida (70%).

Peso relativo: 2%

3.2. Cowboy

Este módulo cria "objetos" do tipo Cowboy que representa uma bala (localização, tipo, estado). É usado para criar (quando se inicia um modo de jogo) e desenhar no ecrã. O estado é atualizado no modo de jogo (diferentes imagens – andar, morto, a disparar).

Responsável: João Mendes (70%).

Peso relativo: 5%

3.3. Duelnix

Este módulo cria o jogo e a sua máquina de estados. É utilizado para alterar entre estados, atualizar os periféricos através do único ciclo de interrupções que invoca os handlers respetivos, desenhar no ecrã o estado do jogo, verificar se o estado atual terminou. É neste módulo que é atualizada também a placa gráfica. No início são subscritas as interrupções e é inicializado no estado do menu inicial. No fim, é libertada a memória e feito o unsubscribe dos dispositivos para futura utilização.

Responsável: João Almeida (60%).

Peso relativo: 8%

3.4. GameState

Este módulo é o modo singleplayer. A cada interrupção do timer são atualizadas as variáveis como: os dois cowboys, o obstáculo, as balas (se houver), o resultado, o temporizador (quando existir) e a deteção de colisões no caso de existirem balas. Estas variáveis são atualizadas chamando funções dos seus módulos, o que mostra que implementamos uma solução faseada. É sempre verificado se o estado já terminou ou não, e o nível é atualizado consoante o resultado.

Responsável: João Mendes (70%).

Peso relativo: 10%

3.5. Graphics

Importado do lab5 e alterado para suportar triple_buffering – no lab5 usamos diretamente a video_mem.

Responsável: João Almeida (70%).

Peso relativo: 5%

3.6. kbd_asm

Importado do lab3 e alterado. Mesmo assim, não conseguimos colocar a funcionar a leitura de scancodes do kbc em assembly.

Responsável: João Mendes (100%).

Peso relativo: 0%

3.7. keyboard

Código importado do lab3, com a remoção de tick_delay() para um desempenho mais rápido.

Responsável: João Almeida (70%).

Peso relativo: 8%

3.8. main

Chama sef_startup() e srand() para mais tarde usar no programa. Inicializa o modo gráfico com vg_init() e inicia a máquina de estados que entre num ciclo de atualização – desenho, no fim termina a máquina de estados e chama vg_exit() para voltar ao modo de texto.

Responsável: João Mendes (50%).

Peso relativo: 5%

3.9. MainMenuState

Estado inicial (e será o final tamém), que apresenta o menu inicial e "guia" o utilizador pelo programa. Sempre que sai de um jogo, volta-se a este estado. O estado tem três botões e a opção de sair com Escape e de entrar no modo multiplayer com Space Bar. Este é o estado mais vezes repetido ao longo do programa

Responsável: João Almeida (80%).

Peso relativo: 6%

3.10. Mouse

Este módulo foi importado do lab4 e alterado de modo à informação do rato (botão esquerdo e posição) seja acessível a partir de qualquer sítio e em qualquer momento do programa, para se lhe poder recorrer nos diferentes modos de jogo e ser mais fluido o seu uso.

Responsável: João Mendes (70%).

Peso relativo: 6%

3.11. Multiplayer

Este módulo é o modo multiplayer. A cada interrupção do timer são atualizadas as variáveis como: os dois cowboys, o obstáculo, as balas (se houver), o resultado e a deteção de colisões no caso de existirem balas. Estas variáveis são atualizadas chamando funções dos seus módulos, o que mostra que implementamos uma solução faseada. É sempre verificado se o estado já terminou ou não, e o nível é atualizado consoante o resultado. Este modulo denota especial uso do teclado, usando 15 teclas em simultâneo para dois jogadores se poderem defrontar.

Responsável: João Mendes (100%).

Peso relativo: 10%

3.12. PracticeState

Este módulo é o modo de treino. A cada interrupção do timer são atualizadas as variáveis como: o cowboy e o alvo, o obstáculo, a bala, o contador de tiros certos, o temporizador (quando existir) e a deteção de colisões no caso de existirem balas. Estas variáveis são atualizadas chamando funções dos seus módulos, o que mostra que implementamos uma solução faseada. É sempre verificado se o estado já terminou ou não, e o nível é atualizado consoante o resultado.

Responsável: João Mendes (50%).

Peso relativo: 10%

3.13. rectangle

Este módulo cria um “retângulo” através de coordenadas e serve para criar “botões” verificando se a posição do rato é interior ou não a esse retângulo.

Responsável: João Almeida (80%).

Peso relativo: 2%

3.14. rtc

Este módulo contem funções para lidar com o real time clock. Funções para subscrever(modo 24 horas), unsubscribe, disable e enable de interrupções e também a sua gestão, onde é atualizada uma estrutura com horas e minutos, correspondendo à hora atual.

Responsável: João Mendes (80%).

Peso relativo: 5%

3.15. shield

Este módulo cria "objetos" do tipo Shield que representa os obstáculos (velocidade, localização). É usado para criar (quando se inicia um jogo, ou se passa um nível), desenhar no ecrã, atualizar a posição (recorrendo à velocidade) e apagar quando termina o jogo (o obstáculo é indestrutível – daí o nome do módulo, os jogadores devem usar como escudo).

Responsável: João Almeida (60%).

Peso relativo: 2%

3.16. table

Este módulo cria "objetos" do tipo Table que representa os resultados (localização e tipo – pode ser só um jogador ou dois). É usado para criar (quando se inicia um jogo, e reset se passado um nível), desenhar no ecrã, atualizar o resultado e apagar quando termina o jogo. Os resultados são atualizados nos módulos de jogo.

Responsável: João Almeida (60%).

Peso relativo: 5%

3.17. timer

Importado do lab2 e usado para o temporizador e para refrescar o ecrã em cada interrupção.

Responsável: João Almeida (60%).

Peso relativo: 5%

3.18. vbe

Importado do código das aulas práticas do laboratório 5, onde a função implementada obtém as informações sobre o modo de vídeo – 0x114 neste caso.

Responsável: João Mendes (50%).

Peso relativo: 2%

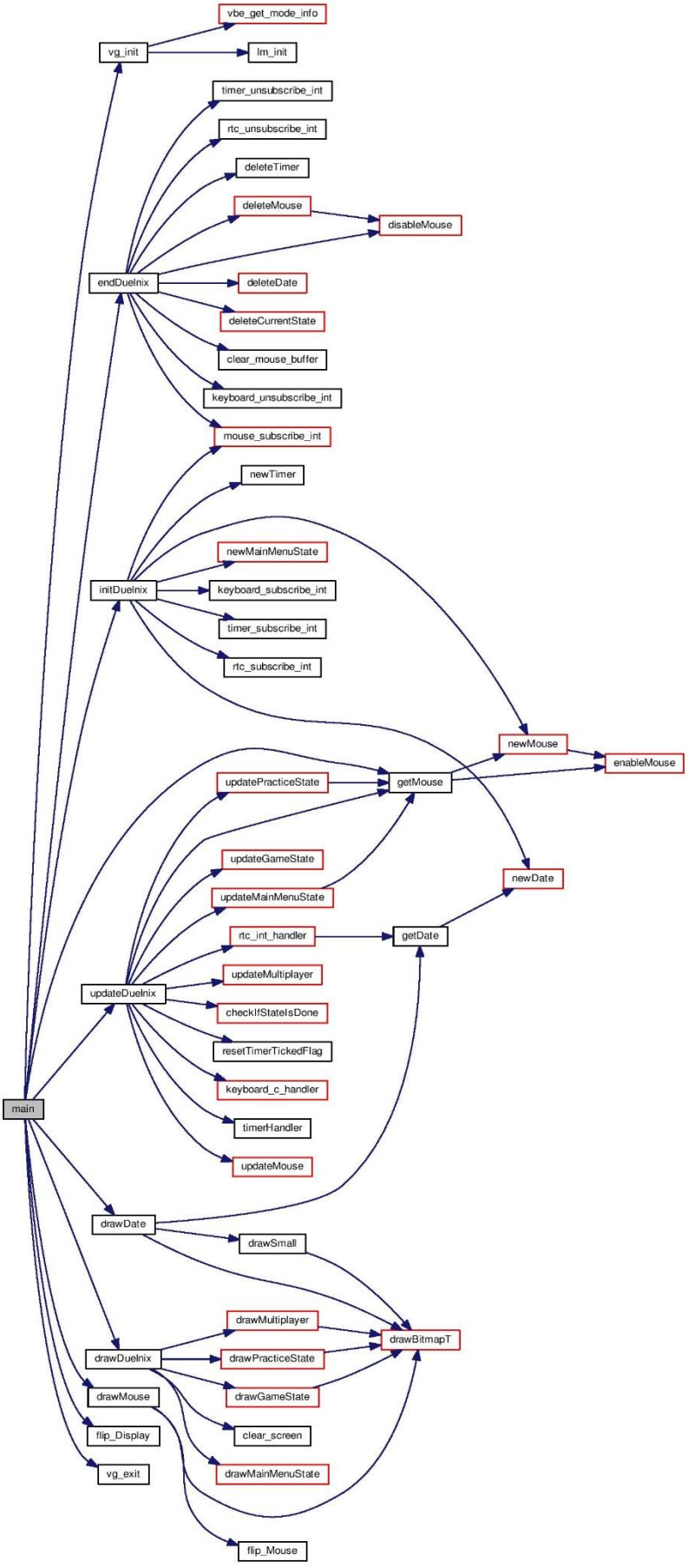
3.19. Bitmap

http://difusal.blogspot.pt/2014/09/minixtutorial-8-loading-bmp-images.html
Código autorizado pelo professor, desenvolvido por Henrique Ferrolho.
O código foi alterado para corresponder às nossa procura.
Uma função adicional para não pintar o fundo das imagens criadas pelo grupo.

Responsável: João Almeida (50%).

Peso relativo: 4%

3.20. Function Call Graph



4. Detalhes da Implementação

A colisão de objetos, a criação de botões, a utilização do rato durante todo o programa são muito importantes para o resultado. A criação de um "placar" para amostragem dos resultados e o aproveitamento do seu uso num temporizador.

O RTC também é um dos detalhes, pois a sua aprendizagem foi feita fora das aulas. Foi usado o modo de 24 horas, e foram subscritos os interrupts e com a verificação do registo A e a "limpeza" do registo C a cada interrupção.

O uso de uma struct para cada objeto representado no ecrã contribuiu para uma mais fácil abstração para fazer o projeto.

O uso de uma só struct para representar os dois tipos de cowboys, os dois tipos de "placar" e os dois tipos de escudo também é um pormenor a assinalar porque nos permitiu estruturar o que seria complicado apenas no módulo dessa struct, e no módulo da lógica de jogo ser mais fácil de lidar com as situações.

O uso de triple buffering permitiu-nos ter muito menos "lag" e o flicker do ecrã desapareceu a 100%, até agora.

Lidar com as structs e diferentes tipos de objetos como por exemplo as balas, foi um pouco difícil a princípio e levou-nos a alguma pesquisa para encontrarmos uma solução.

A AI do computador é de resposta, responde aos inputs do utilizador, assim dificulta o jogo, mas é também misturada com rand() para o jogo não ser "impossível". Os disparos são efetuados na direção do rato, este truque permite ao computador disparar na direção que o utilizador vai pretender disparar (ele é que controla o rato), e isto faz com que os disparos vão na direção do cowboy do utilizador, isto foi também misturado com rand() para que o jogador não esteja sempre na mira do "computador", tornando o jogo não tão difícil. Sabendo este truque, torna-se mais fácil jogar contra o computador.

No modo multiplayer, como só se pode atirar em 3 direções, os tiros para baixo e para cima são efetuados com declive constante, com vx = vy, o que leva a que a bala seja disparada sempre com 45 graus.

5. Conclusões

LCOM foi uma experiência positiva, porque aprendemos a lidar com os periféricos do computador duma forma mais profunda. Talvez por isso é que a unidade curricular tenha sido um pouco complicada, mas a aprendizagem foi gratificante e aprendemos melhor a programar em linguagem C (o que ainda não tínhamos feito).

Posto isto, apenas temos a apontar os guiões do lab4 e lab5 (lab6 e lab7 agravando-se) como um ponto negativo, pois são demasiado vagos comparativamente ao lab2 (principalmente) e lab3. As datas de entrega não são as melhores para os grupos com aulas às segundas (por vezes aprenderam matéria na teórica que tiveram de aplicar passado umas horas).

Durante os primeiros labs em que estamos um pouco no "desconhecido" seria uma ajuda ter mais um monitor na sala, pois nessas aulas o professor e o monitor são muito requisitados.

Quanto ao projeto, à primeira vista parecia complicado, mas pouco a pouco e com uma boa estruturação tornou-se mais fácil a sua implementação.

6. Apêndice

Instruções de instalação:

- Download da pasta proj para /home/lcom
- o diretório do projeto será /home/lcom/proj
- entrar em src - /home/lcom/proj/src
- comando “su”
- comando “sh install.sh”
- comando “sh compile.sh”
- comando “sh run.sh”

Estes comandos devem-se aos recursos do jogo (imagens bmp).