

Faculdade de Engenharia da Universidade do Porto



**Projecto Final
Laboratório de Computadores
Turma 5 - Grupo 4**

Realizado por:

- Fábio Araújo de Sá (up202007658@fe.up.pt)
- Inês Sá Pereira Estêvão Gaspar (up202007210@fe.up.pt)
- Lourenço Alexandre Correia Gonçalves (up202004816@fe.up.pt)
- Marcos William Ferreira Pinto (up201800177@fe.up.pt)

Índice

Introdução	3
Instruções de Utilização	4
Menu inicial	4
Instruções	5
Desenhar	6
Adivinhar	8
Jogo a carregar	10
Fim de jogo	10
Estado do Projeto	12
Tabela de funcionalidades	12
Tabela de dispositivos	12
Dispositivos	13
Mouse	13
Teclado	13
Timer	13
Video Graphics	14
Serial Port	14
Organização do código/estrutura	15
Keyboard Module - 8%	15
Mouse Module - 10%	15
Queue Module - 3%	15
Serial Port Module - 8%	15
Timer - 5%	16
Video Graphics Module - 10%	16
Utils Module - 1%	16
Drawing Module - 10%	16
Guessing Module - 10%	17
Instructions Module - 2%	17
Menu Module - 10%	17
Game Module - 7%	17
Read Module - 3%	17
Sprites Module - 3%	18
Main - 10%	18
Function call Graph	19
Detalhes de Implementação	20
Envio de dados com a serial port	20
Conclusões	21
Referências	22

Introdução

O nosso Projeto “Gertic--” consiste numa adaptação do conhecido jogo [Gartic](#) para dois jogadores.

No primeiro caso, o jogador seleciona a opção de desenhar e é-lhe atribuída uma palavra aleatória que tem que recriar num tempo limitado.

No segundo caso, após a escolha do modo de jogo (desenhar ou adivinhar), ao jogador que selecionou a opção de desenhar é dada uma palavra aleatória que ele tem que recriar num desenho num tempo limitado. Para desenhar o jogador tem disponíveis várias cores e diferentes espessuras do traço. O segundo jogador terá de adivinhar a palavra de acordo com o desenho que recebe, escrevendo a sua tentativa numa caixa de texto.

O jogo acaba quando o segundo jogador acerta na palavra ou quando o tempo acaba.

Instruções de Utilização

Menu inicial

No início do jogo, o menu inicial é mostrado. Este contém 4 botões: Desenhar, Adivinhar (ambos para escolha do modo de jogo), Instruções (para indicar as regras de utilização) e Sair. A ação de cada botão pode ser acedida ou por movimentos do rato (clique com o botão esquerdo sob a área pretendida) ou pressionando as teclas indicadas em cada secção.

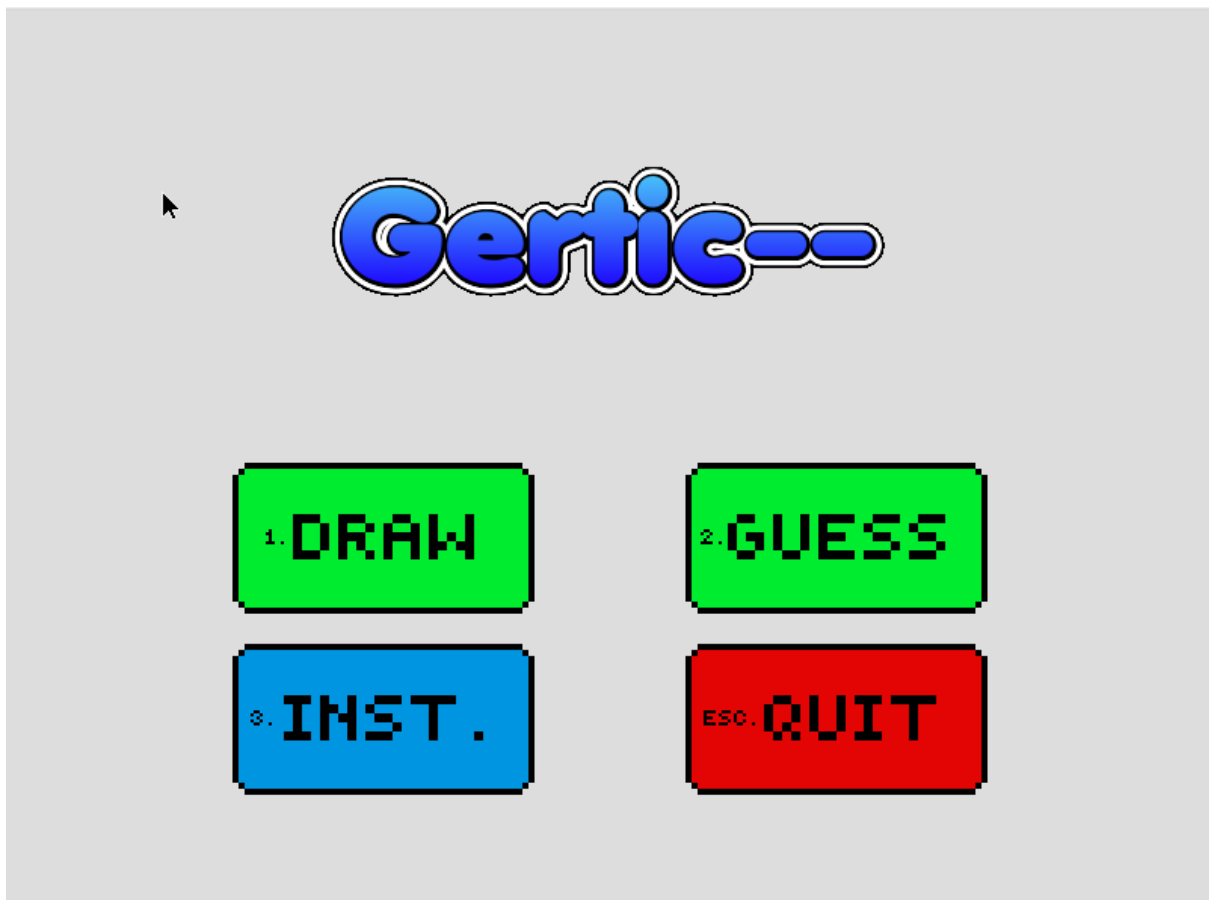


Fig 1. Menu inicial

Instruções

Esta janela mostra as instruções e regras para jogar. Para regressar ao menu inicial deve ser pressionada a tecla ESC, tal como indicado.

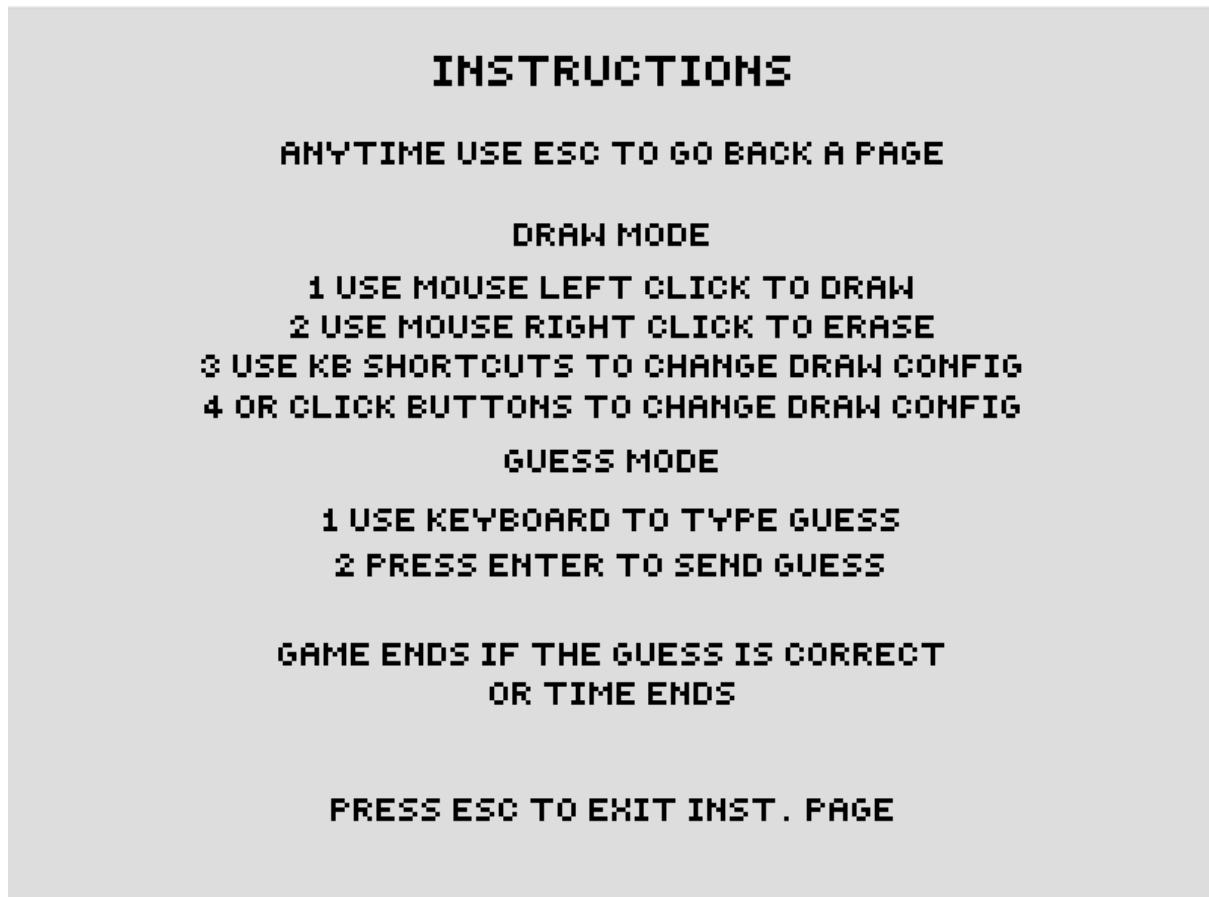


Fig 2. Instruções

Desenhar

Neste modo, o jogador deve desenhar uma representação da palavra aleatória que lhe é atribuída. Esta última aparece no canto inferior esquerdo, enquanto que o tempo restante para o fim do jogo é apresentado no canto inferior direito.

O jogo disponibiliza uma paleta variada de cores. O jogador pode mudar de cor de desenho clicando com o botão esquerdo do rato em cima da cor que pretende ou pode seleccionar no teclado o número correspondente a essa cor.

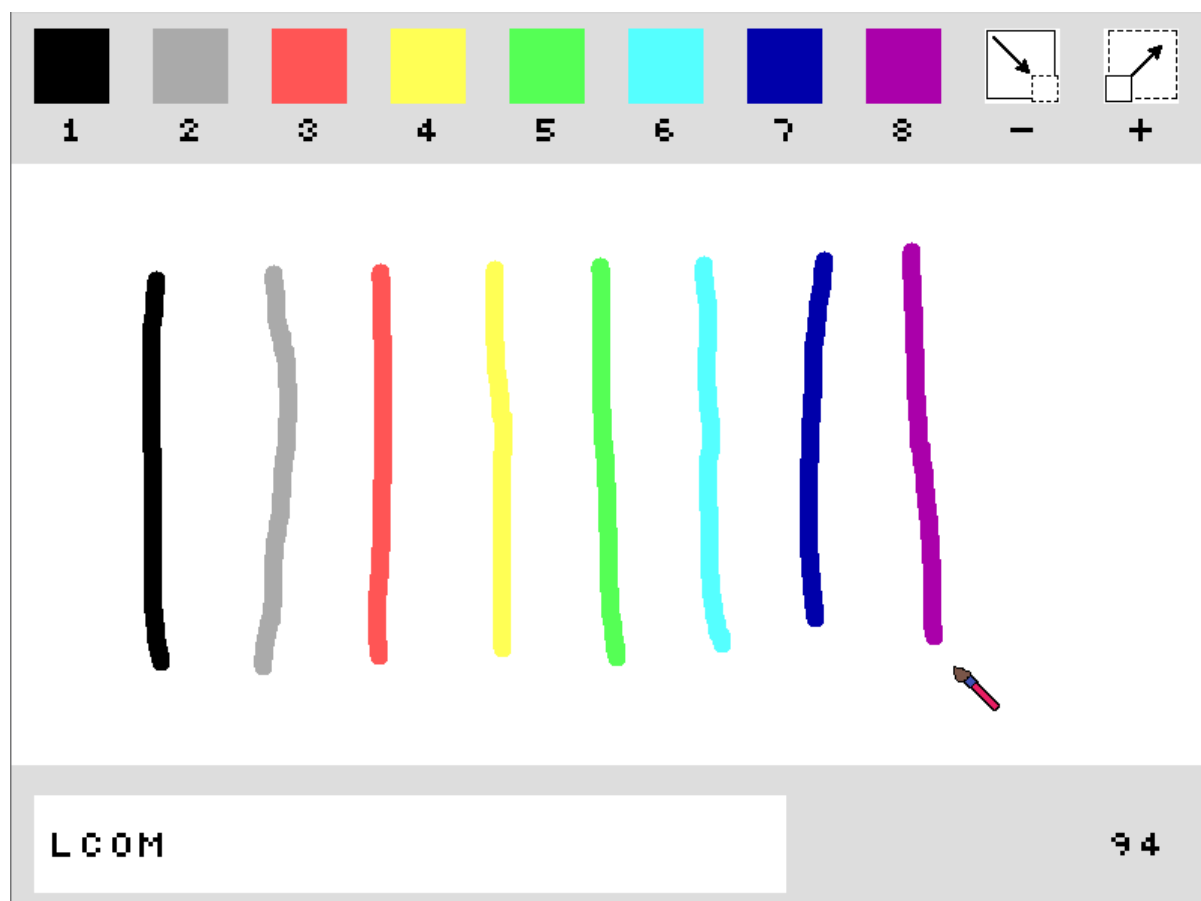


Fig 3. O jogador poderá escolher várias cores para desenhar

O jogo disponibiliza diferentes espessuras do traço. O jogador pode mudar a espessura do traço clicando com o botão esquerdo do rato em cima do botão aumentar ou diminuir ou, de modo similar, pode seleccionar no teclado as teclas '+' e '-' (teclado ANSI) .

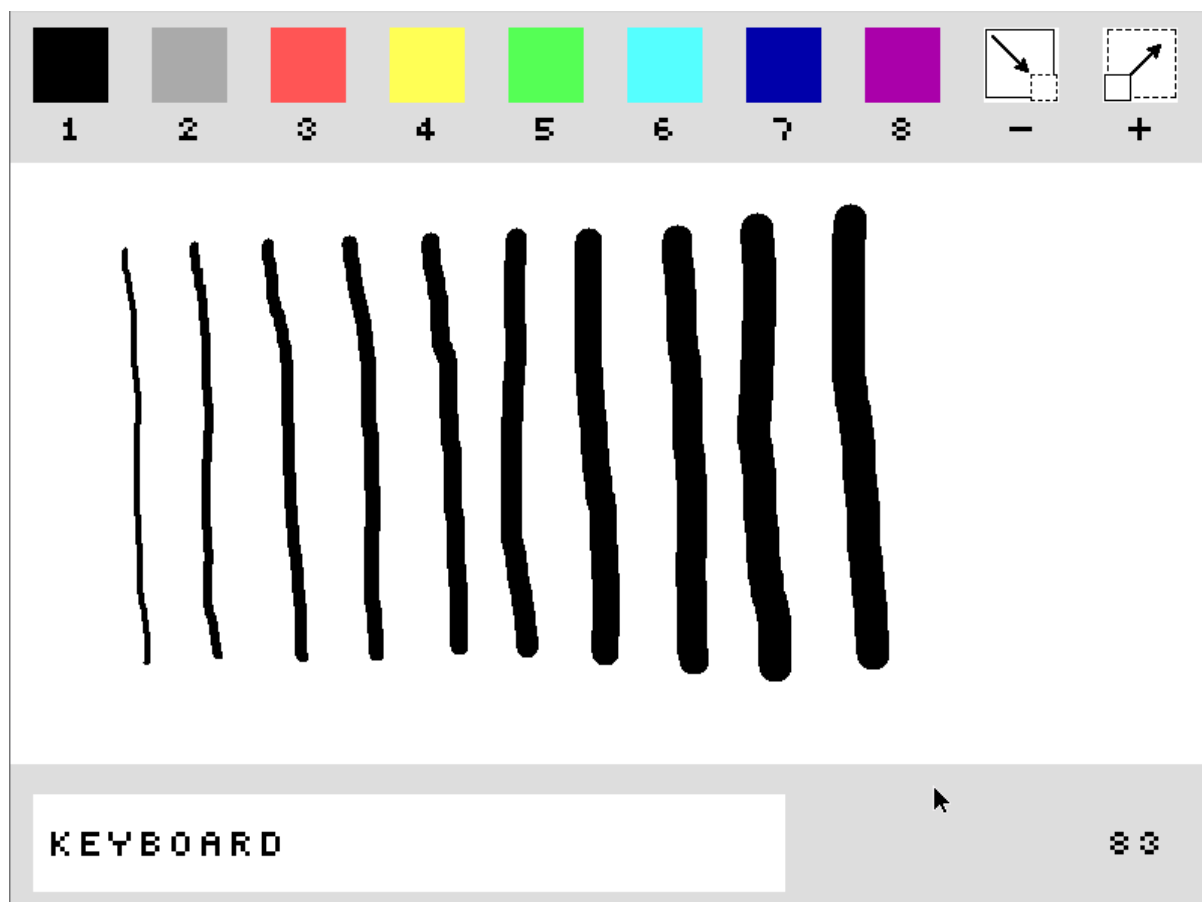


Fig 4. O jogador poderá escolher várias espessuras do traço

O jogo tem também disponível a opção de apagar. O jogador pode apagar partes do desenho clicando com o botão direito do rato e deslizando sobre a área a apagar.



Fig 5. O jogador a qualquer momento pode apagar partes do desenho

Adivinhar

Neste modo, o jogador deve adivinhar a palavra correspondente ao desenho que recebe.

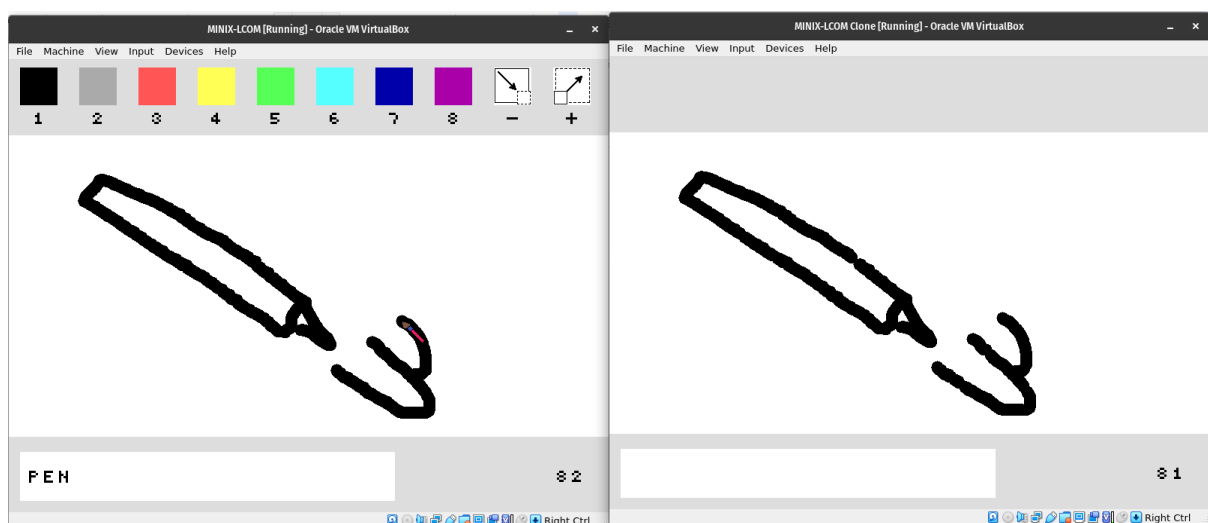


Fig 6. É possível ver o que o outro jogador está a desenhar no momento

A caixa de texto onde pode digitar a sua tentativa, usando o teclado, aparece no canto inferior esquerdo, enquanto que o tempo restante para o fim do jogo é apresentado no canto inferior direito.

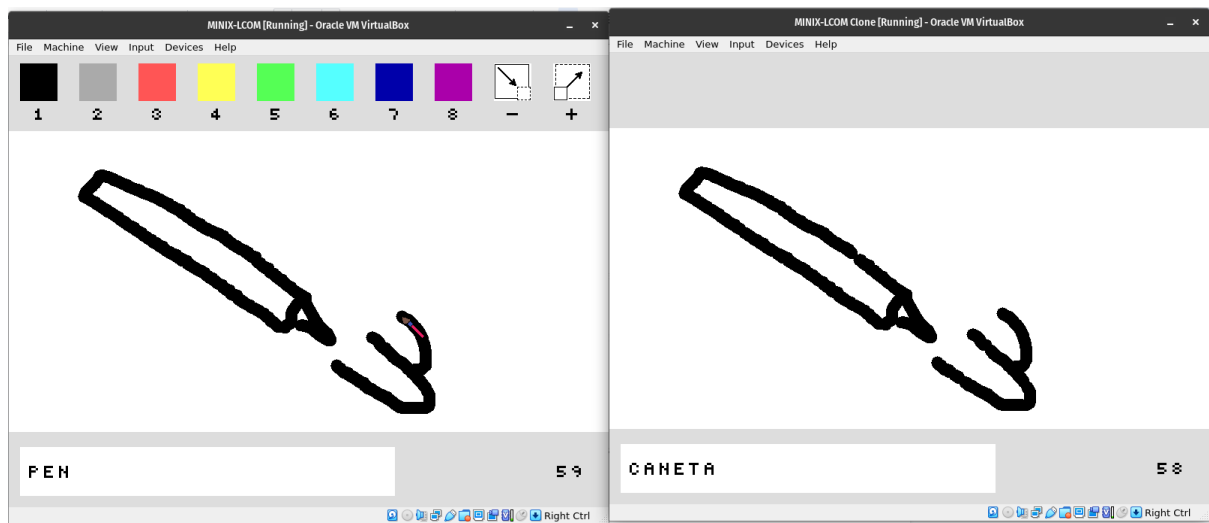


Fig 7. O jogador pode escrever a sua tentativa na caixa de texto

Jogo a carregar

No começo do jogo, enquanto as diferentes instâncias do Gertic-- estão a parear um aviso de loading é mostrado no ecrã.

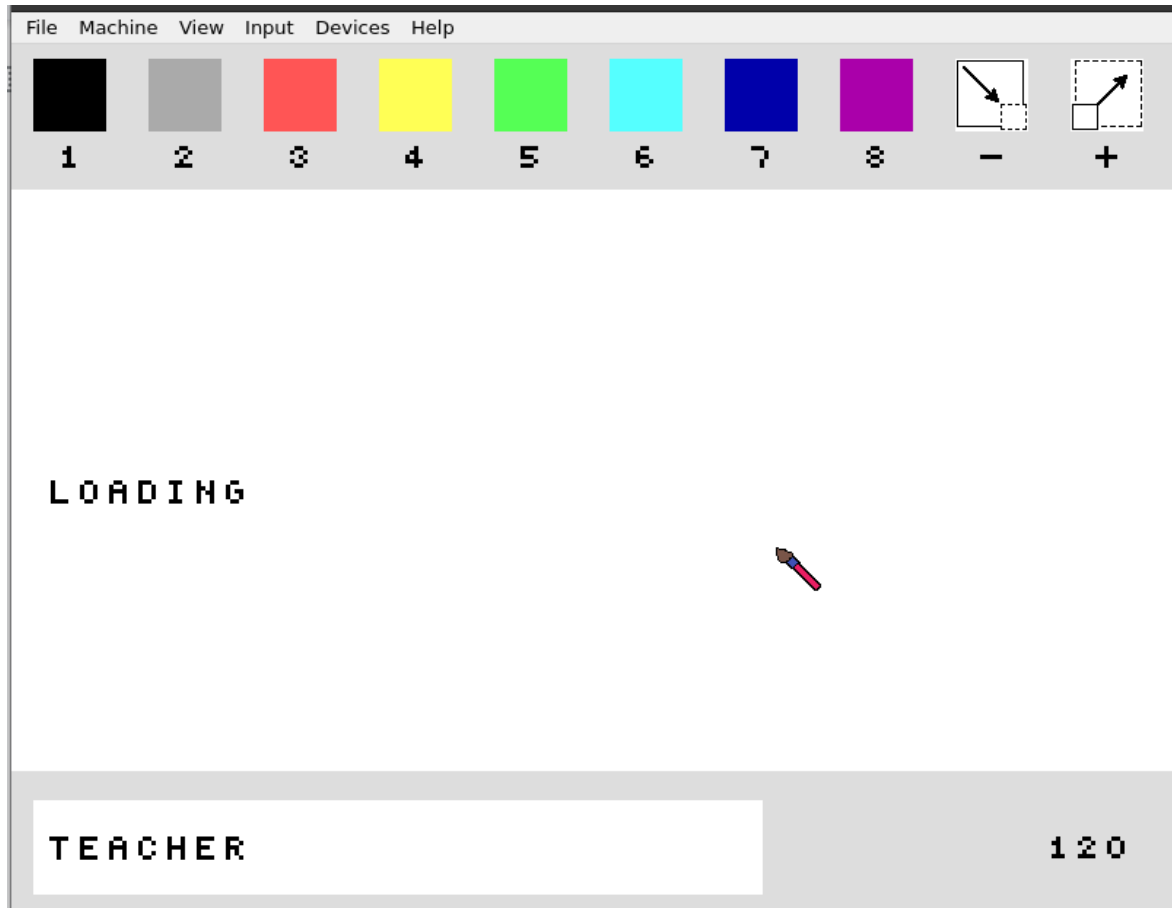


Fig 8. Enquanto o segundo jogador não entra em jogo, aparece este ecrã

Fim de jogo

O jogo pode acabar de três formas diferentes. Sendo elas, se o jogador acertar a palavra, o tempo acabar, ou um dos jogadores desistir da partida. Caso isso aconteça, o jogo exibirá uma mensagem de fim de jogo.

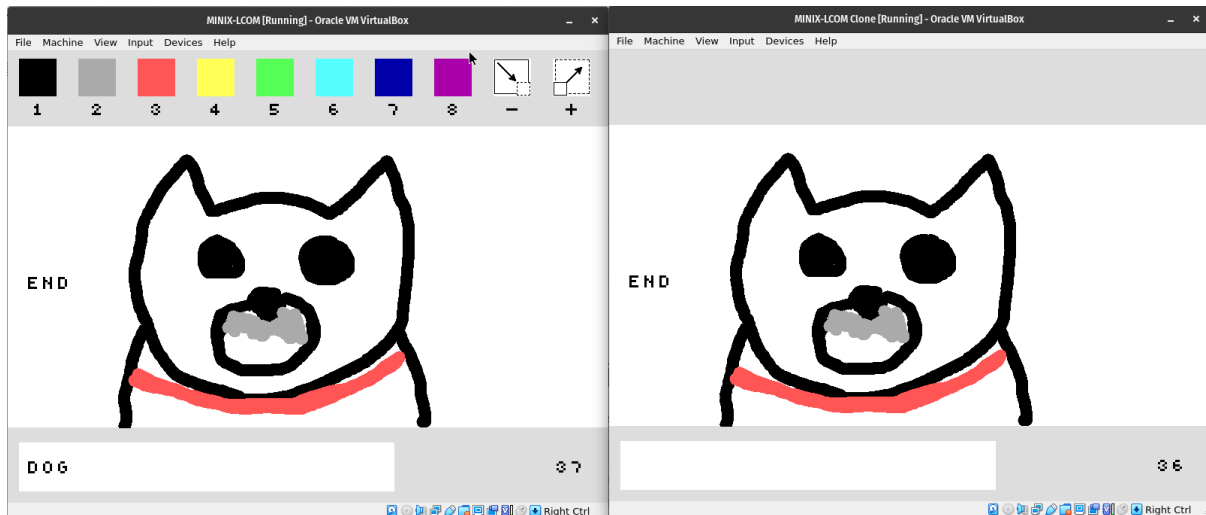


Fig 9. Fim do jogo caso a palavra seja descoberta

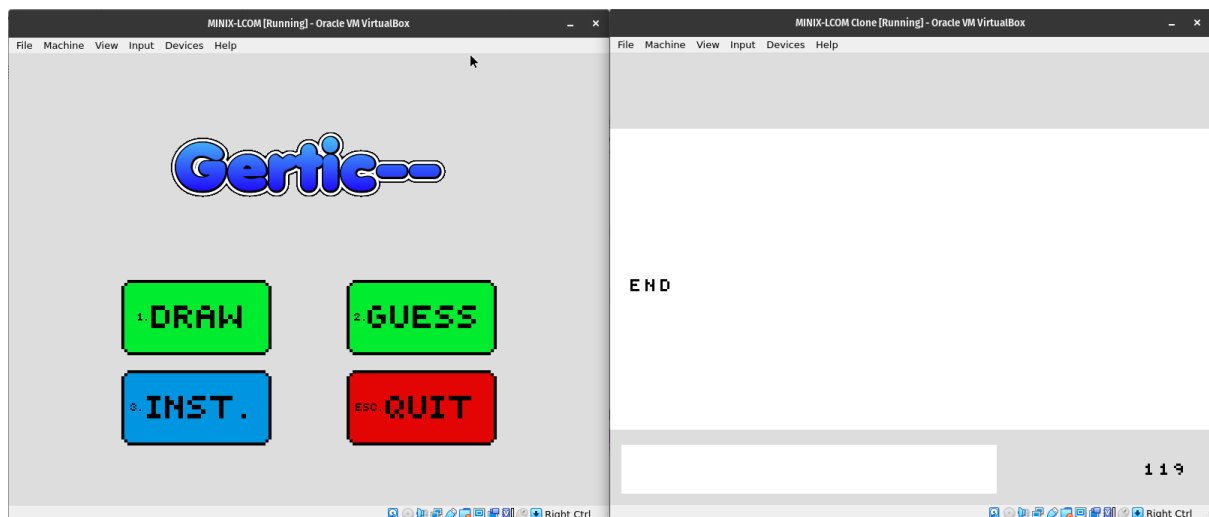


Fig 10. Fim do jogo por desistência

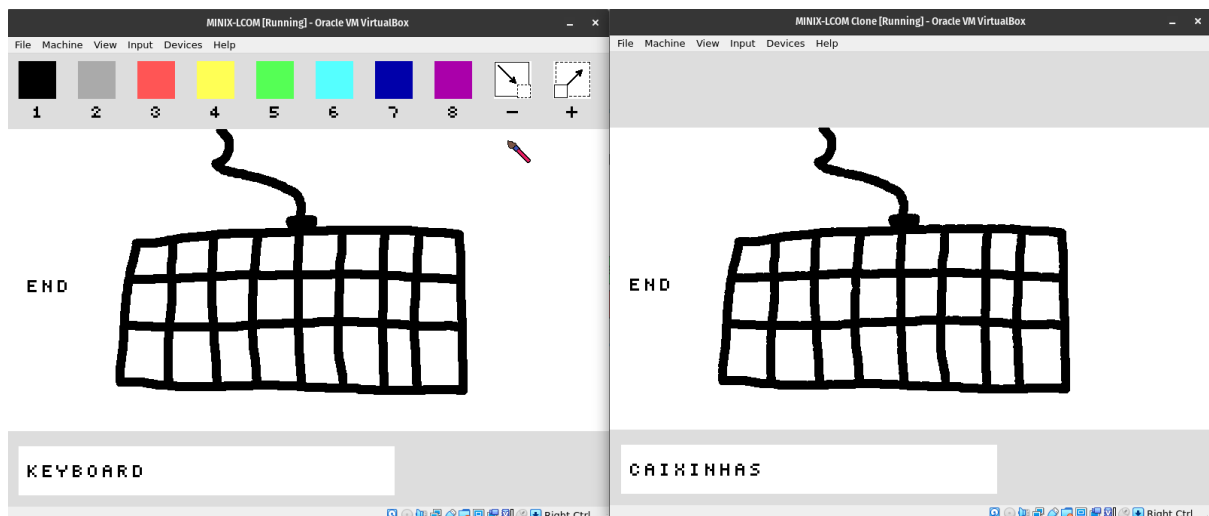


Fig 10. Fim do jogo caso o tempo termine

Estado do Projeto

Tabela de funcionalidades

Funcionalidades	Dispositivos	Estado de implementação
Desenhar	Mouse e video graphics	Completo
Apagar	Mouse e video graphics	Completo
Mudar de cor	Mouse, teclado e video graphics	Completo
Mudar a espessura	Mouse, teclado e video graphics	Completo
Mostrar a palavra	Video graphics	Completo
Escrever a tentativa	Teclado e video graphics	Completo
Mostrar e limitar o tempo de jogo	Timer e video graphics	Completo
Fazer a comunicação entre os dois jogadores	Serial Port, Timer	Completo
Navegação entre menus	Mouse, teclado e video graphics	Completo

Tabela de dispositivos

Dispositivos	Funcionalidades	Interrupções
Mouse	Desenhar, apagar, selecionar opções no menu e no modo desenho	Sim
Teclado	Selecionar opções no menu e no modo desenho, escrever a tentativa de adivinhar	Sim
Timer	Limitar o tempo de jogo, frame-rate	Sim
Video Graphics	Mostrar toda a interface com os jogadores, o desenho, o tempo do jogo e a palavra selecionada	Não
Serial Port	Fazer a comunicação entre os dois jogadores	Sim

Dispositivos

Mouse

O mouse é usado para seleccionar as opções do menu, as cores, a espessura e desenhar/apagar no modo de desenho.

As diferentes posições do rato, transformadas depois em coordenadas cartesianas, permitem distinguir os botões do menu que o jogador selecciona, as diferentes cores disponíveis para o jogador desenhar e a espessura do traço que o jogador pretende. Por fim, as posições do mouse dentro da tela branca permitem detetar os diferentes movimentos para compor o desenho.

No menu, o botão esquerdo do mouse é usado para seleccionar a opção que o jogador pretende. No modo desenho, quando o jogador pressiona o botão esquerdo enquanto desloca o mouse permite desenhar o movimento que o jogador efetuou. O botão esquerdo pode também ser utilizado para seleccionar uma nova cor e espessura do traço. Quando o jogador pressiona o botão direito enquanto desloca o mouse permite apagar o desenho de acordo com o movimento que o jogador efetuou.

A implementação das configurações, funcionalidades e interrupções do mouse estão no ficheiro mouse.c.

Teclado

Tal como o mouse, o teclado também pode ser usado para seleccionar as opções do menu, as cores e a espessura no modo de desenho. Para além disso, o teclado é utilizado para escrever a tentativa no modo de adivinhar.

Em todas as nossas funções que utilizam o teclado fazemos o processamento do input através do make code gerado por cada tecla pressionada.

No menu inicial, podemos pressionar a tecla '1' para entrar em modo desenhar, a tecla '2' para o modo de adivinhar, '3' para a página de instruções e 'ESC' para fechar o programa.

No modo desenhar, pressionando as teclas de '1' a '7' podemos escolher as diferentes cores apresentadas no início da página. Por sua vez, premindo a tecla '+' ou '-' podemos aumentar ou diminuir a espessura do traço, respetivamente.

No modo de adivinhar, o jogador pode fazer a sua tentativa utilizando o teclado para escrever na caixa de texto apresentada no canto inferior esquerdo do ecrã.

A implementação das configurações, funcionalidades e interrupções do teclado estão no ficheiro keyboard.c.

Timer

O timer, mais concretamente o timer 0, é usado para limitar o tempo de cada jogo, para sincronização das duas consolas e para controlar a frame rate da placa gráfica, atualizando o ecrã 30 vezes por segundo.

A implementação das configurações, funcionalidades e interrupções do timer estão no ficheiro timer.c.

Video Graphics

O video graphics é utilizado para dar display das várias páginas que criamos, bem como do desenho realizado por um dos jogadores e a tentativa de adivinhar do segundo jogador.

Todo o projeto está realizado no modo de video, pelo que o display dos caracteres é efetuado através de XPMs. Estes foram gerados com base em imagens PNG e JPEG através do software [GIMP](#). Usamos uma resolução de 800x600 píxeis (modo VBE 0x115), com cores de 3 bytes em modo direto e como tal permite 2^{24} cores (16,8 milhões, aproximadamente) diferentes no nosso programa.

De forma a tornar o jogo mais fluido e dinâmico sob o ponto de vista de cada um dos jogadores, implementamos a técnica de triple buffering. Enquanto o primeiro buffer atualiza o ecrã (o que estiver lá é enviado para o display), um segundo buffer é usado para montar o próximo frame que será enviado para o buffer principal (função [copyBufferToVideo\(\)](#)). Além disso, um terceiro buffer é utilizado para carregar elementos estáticos, evitando a perda de performance ao carregar sempre os mesmos elementos; também é utilizado para guardar o desenho durante um jogo (num espaço definido dentro do buffer). Esse buffer é sempre enviado para o buffer secundário para montar o próximo frame (função [copyDrawingToBuffer\(\)](#)).

Utilizamos sprites para criar os vários elementos visuais do nosso programa cuja forma não é alterada durante a execução do programa, através da função [Sprite *createSprite\(xpm_map_t sprite\)](#) e de um XPM previamente definido. São exemplos o cursor do rato, o pincel do rato no modo desenho, e os botões do menu. Ao contrário da detecção de colisões, pois tal não fazia sentido para o nosso jogo, existem pequenas animações (através de estados) quando os botões do menu inicial são clicados.

A implementação da configuração e funcionalidade referentes ao video graphics encontram-se no ficheiro `graphics.c`.

Serial Port

O serial port permite fazer a comunicação entre as duas máquinas virtuais, onde uma das máquinas joga no modo desenho e a outra em modo adivinhar (ou seja, simula uma partida entre dois jogadores).

Para o jogo funcionar, os dois modos (desenhar e adivinhar) devem estar sincronizados. Isso é feito a partir de um handshake. A instância que estiver em modo desenho, a cada segundo, envia um pedido de sincronização. Quando a outra instância receber o pedido envia uma resposta de volta confirmando o pedido, e como confirmação, deve receber de volta uma resposta da primeira máquina a dizer que o emparelhamento foi realizado.

Durante a partida, para o desenho ser transmitido, é enviado 3 bytes de verificação, as posições inicial e final, para que a máquina em modo de adivinhar possa recriar o que está a ser desenhado.

Os três bytes de verificação, bem como os bytes enviados a seguir, podem ser diferentes, isso permite distinguirmos entre diferentes tipos de envio de dados: desenhar uma nova linha, apagar o desenho, ou configurar cor e espessura da linha.

Além disso, durante o jogo as duas máquinas podem receber uma mensagem de fim de jogo, sendo feita a partir dos bytes de verificação (as mensagens são diferentes para as duas instâncias). A implementação das configurações, funcionalidades e interrupções do serial port encontram-se no ficheiro `serialPort.c`.

Organização do código/estrutura

Keyboard Module - 8%

Este módulo contém as funções desenvolvidas no Lab3 das aulas práticas sobre o teclado que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem configurar e invocar as interrupções do dispositivo.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Mouse Module - 10%

Este módulo contém as funções desenvolvidas no Lab4 das aulas práticas sobre o mouse que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem configurar, invocar as interrupções do dispositivo e sincronizar os 3 bytes de informação retornados pelo controlador em cada interrupção.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Queue Module - 3%

Utilizamos a implementação da queue em C do site seguinte:

<https://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/>

Este módulo contém as funções que permitem implementar uma estrutura de dados FIFO (First In First Out), que é utilizada pelo serial port para transferência de bytes de uma máquina virtual para outra.

Serial Port Module - 8%

Este módulo contém as funções que permitem fazer a configuração e as interrupções do serial port.

Estas funções permitem ler e receber bytes da porta série, configurar as interrupções e interpretar os *bytes* recebidos no contexto e estado atual do jogo.

Contribuidores:

- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Timer - 5%

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas sobre o timer que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem fazer a configuração, as interrupções do timer e a sincronização entre os vários dispositivos usados.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Video Graphics Module - 10%

Este módulo contém as funções desenvolvidas no Lab5 das aulas práticas sobre o video graphics que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem fazer a configuração do video graphics e desenhar algumas das formas que aparecem nos ecrãs, nomeadamente a estrutura mais geral das páginas de desenhar e adivinhar.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Utils Module - 1%

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas sobre o timer que posteriormente continuaram a ser usadas nos Labs seguintes e por esse motivo puderam ser também reutilizadas no nosso Projeto.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Drawing Module - 10%

Este módulo contém as funções que permitem fazer toda a interface do modo desenhar, a que permite mostra a palavra e a própria função de desenhar e apagar.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Guessing Module - 10%

Este módulo contém as funções que permitem fazer a interface do modo adivinhar, escrever a tentativa e verificar se a tentativa está correta.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Instructions Module - 2%

Este módulo faz a interface da página de instruções.

Contribuidores:

- Marcos William Ferreira Pinto

Menu Module - 10%

Este módulo contém as funções que fazem a interface do menu, cria os botões “Draw”, “Guess”, “Inst” e “Exit” e faz a verificação do input dado pelo jogador, redirecionando-o para a página da opção que escolheu.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Game Module - 7%

Este módulo contém as funções que permitem desenhar o movimento do mouse, as palavras e o tempo tanto nos dois modos de jogo.

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Read Module - 3%

Este módulo contém a função que permite ler o ficheiro com palavras a desenhar e escolher uma aleatória para a partida.

Contribuidores:

- Fábio Araújo de Sá
- Marcos William Ferreira Pinto

Sprites Module - 3%

Este módulo cria cada elemento visual usado nas funções que fazem a interface com os jogadores. Baseia-se no código do professor João Cardoso.

Contribuidores:

→ Marcos William Ferreira Pinto

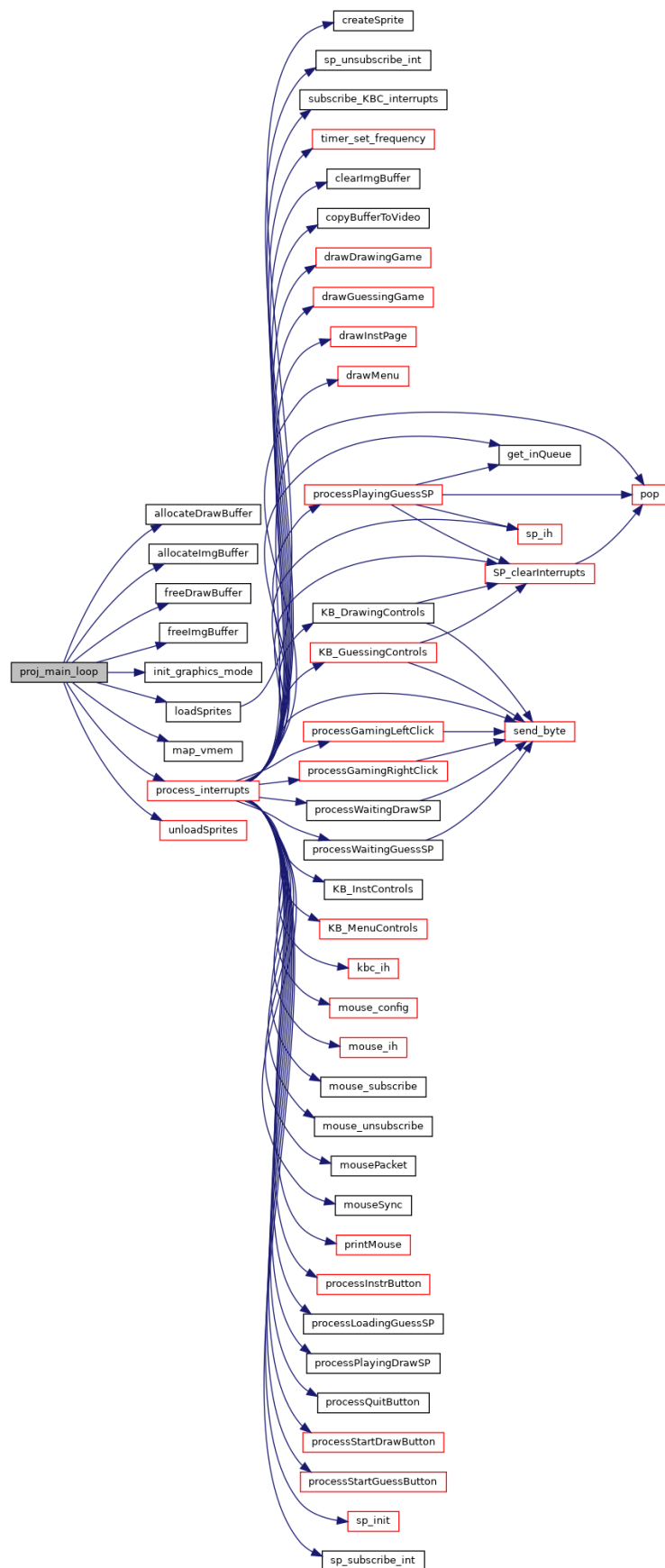
Main - 10%

Este módulo contém o loop principal do projecto assim como o loop que recebe e trata de cada interrupção (void process_interrupts()).

Contribuidores:

- Fábio Araújo de Sá
- Inês Sá Pereira Estêvão Gaspar
- Lourenço Alexandre Correia Gonçalves
- Marcos William Ferreira Pinto

Function call Graph

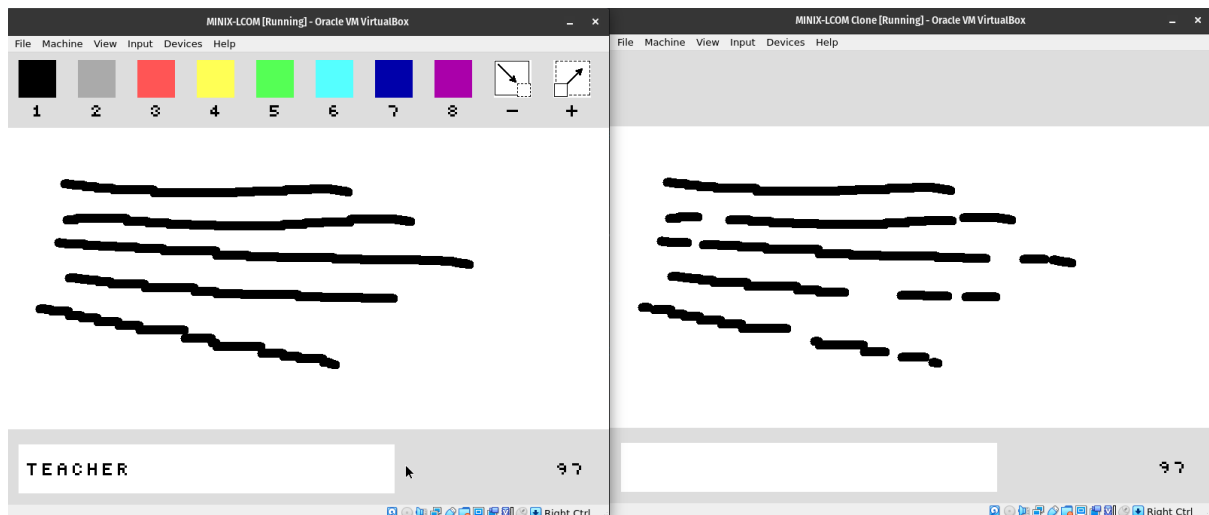


Detalhes de Implementação

Envio de dados com a *serial port*

Mencionamos anteriormente no relatório como mandamos informação entre os computadores dos dois jogadores, no entanto, o método referido resultou em vários problemas causados por erros na comunicação. Em primeiro lugar, o programa desenhava, de forma muito frequente, linhas em posições aleatórias da zona de desenho do lado da pessoa que o recebe. Em segundo lugar, quando ao desenhar uma linha muito rápido, esta ficava cortada, com espaços em branco.

Para resolver este problema, para além da informação necessária para desenhar a linha, enviamos também bytes de controlo para validar a informação e verificar que não tenham ocorrido erros de comunicação. Desta forma já não conseguimos replicar o primeiro problema, acontecendo apenas muito raramente. O segundo problema melhorou bastante, mas não ficou totalmente resolvido.



Conclusões

Neste Projeto conseguimos implementar com sucesso todas as features descritas na proposta inicial.

O jogo proposto, uma adaptação de uma plataforma multijogador mundialmente famosa, foi escolhido para ir além dos obrigatórios dispositivos a implementar. Exigiu, por isso, uma necessidade do Mouse e do Serial Port. Com isto tivemos uma visão ampla da funcionalidade e exigência de todos os dispositivos abordados nas aulas teóricas.

O serial port, tal como previsto, foi o módulo que se destacou: tanto pela complexidade, como pelo tempo necessário a afinar cada detalhe de transferência de bytes e, já numa fase de testes, pela forma com que alguns erros foram contornados.

Foi com trabalho de equipa, com tarefas bem distribuídas e quase sempre em diárias reuniões que tudo foi desenvolvido e entregue em tempo útil.

Com uma proposta inicial equilibrada, permitindo explorar as diversas funcionalidades de cada um dos dispositivos abordados nas aulas do semestre, e a correcta implementação de cada módulo auxiliar, podemos concluir que a execução deste projecto foi em tudo benéfica para a aprendizagem do grupo.

Referências

- Implementação da Queue, consultado em 2/06/2022:
<https://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/>
- Vídeo de demonstração:
https://youtu.be/-W_GAy7G14M
- Sprites baseado na implementação do Professor João Cardoso