Faculdade de Engenharia da Universidade do Porto

Projeto Final

Laboratório de Computadores

Turma 8 - Grupo 4

Realizado Por:

André Tiago Moreira Soares da Costa e Silva (up202108724)

Bernardo Ferreira Campos (up20206056)

João Tomás Pinho Cardinal Teixeira (up202108738)

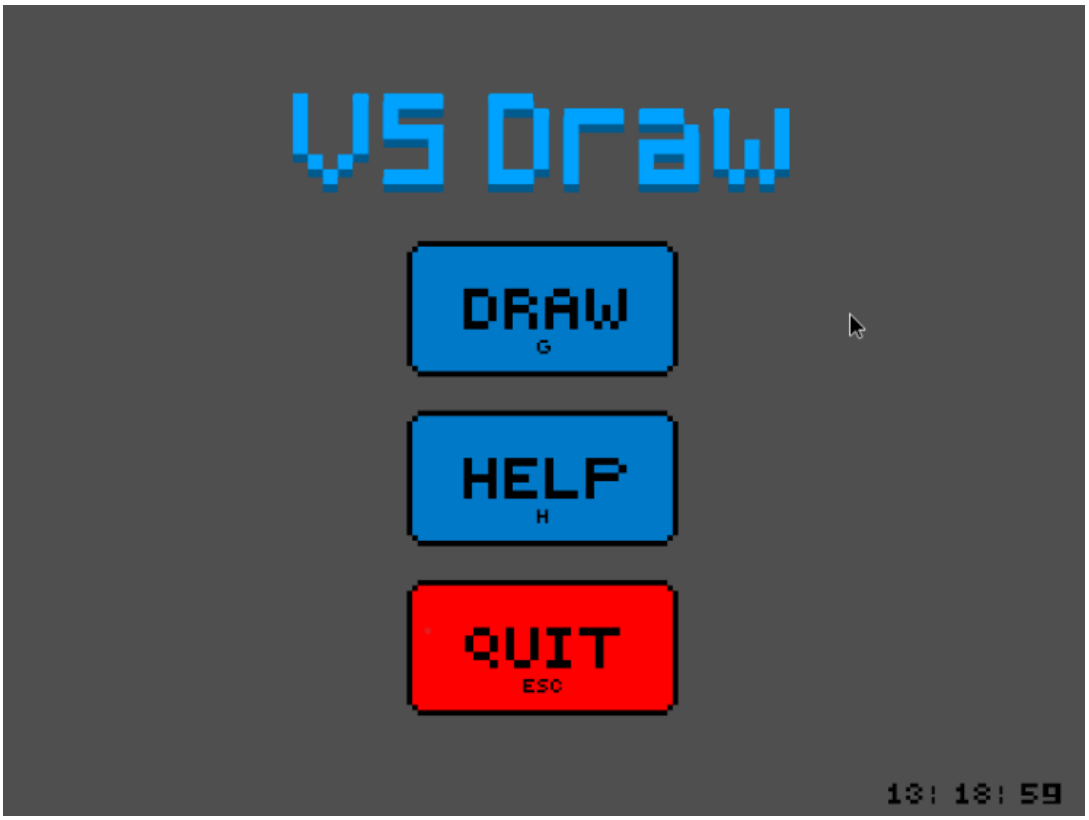
Índice

1. Instruções de utilização de programa

1.1 Menu Inicial

O Menu Inicial é uma interface na qual as opções iniciais do programa são apresentadas. Assim,quer com a utilização do rato, quer com certos atalhos do teclado, esta assenta sobre a seleção de um entre de três botões:

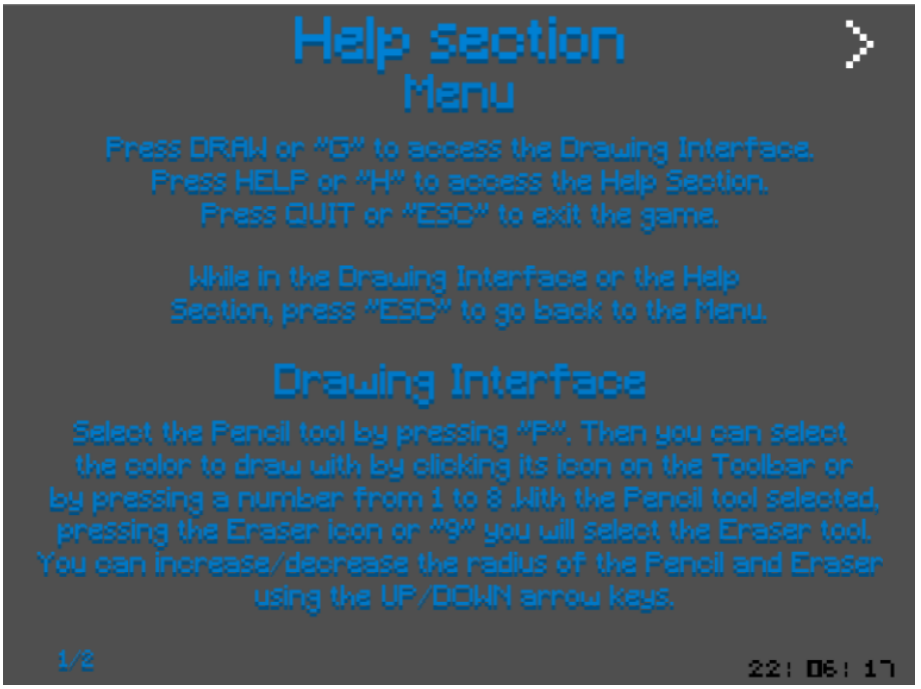
- Desenho: Inicia a ferramenta de desenho em modo de desenho.
- Instruções : Exibe um manual de instruções que revelam a funcionalidade de cada ferramenta disponível(e os seus atalhos de teclado respetivos).
- Saída: Encerra o programa.

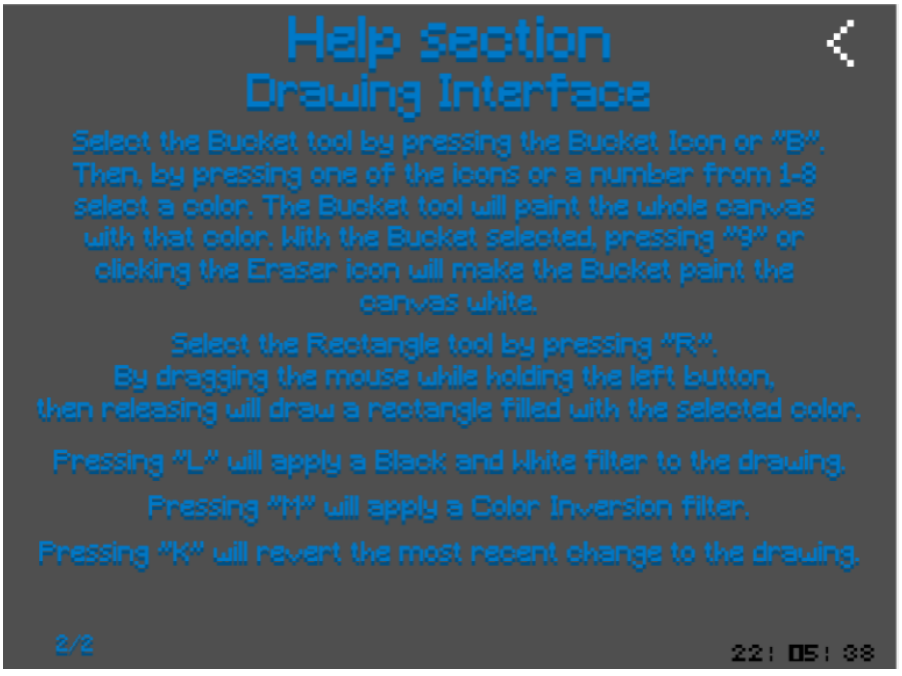


1.2 Instruções

As instruções servem de orientações que o utilizador deverá ter em conta aquando da utilização do programa, pelo que este se encontra dividido em duas páginas. Enquanto a primeira página indica instruções sobre a

interatividade com o menu com o uso do teclado, a segunda funcionalidade indica todas as ferramentas e filtros disponíveis, bem como o seu atalho e funcionalidade correspondentes.

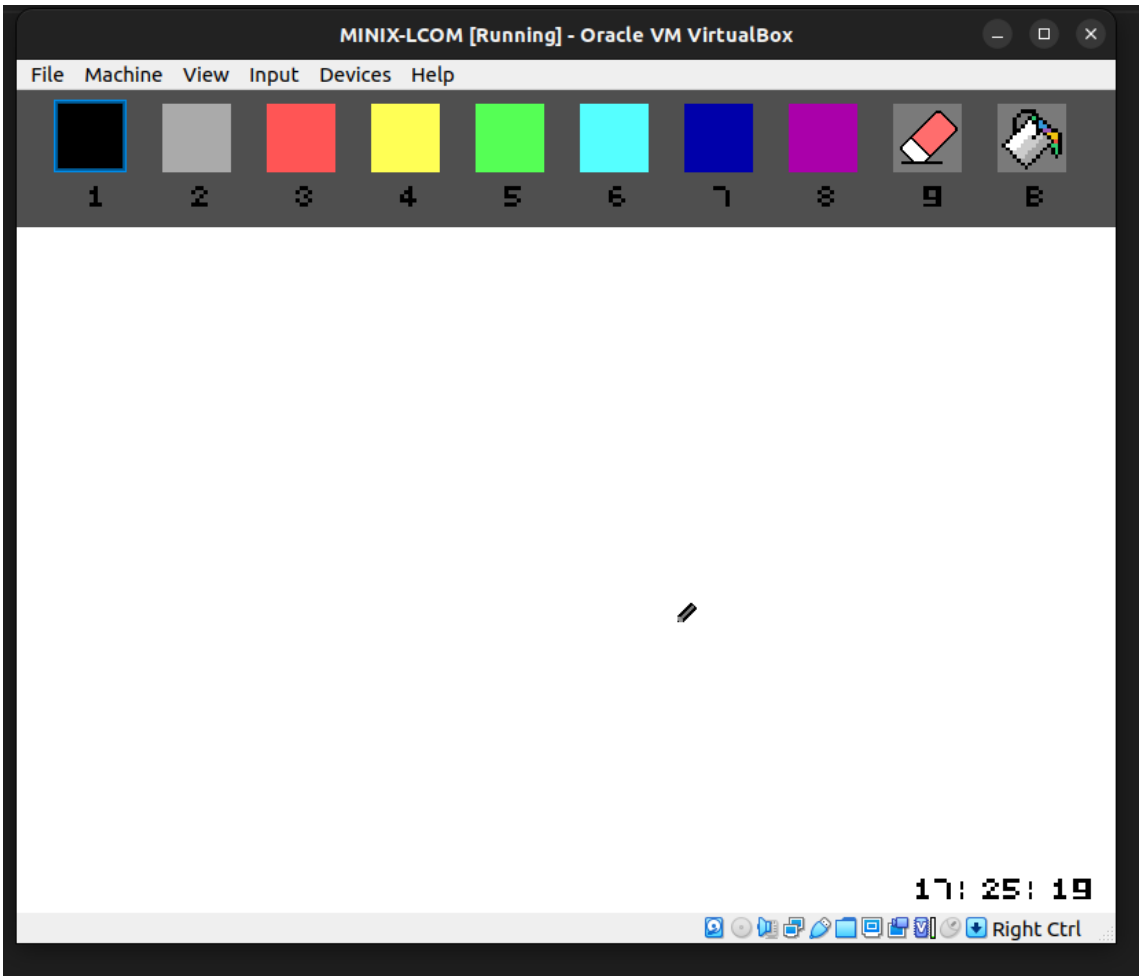




1.3 Modo de Desenho

Neste modo, o utilizador é convidado a desenhar livremente no ecrã, utilizando as ferramentas apresentadas. Assim, é apresentada uma barra

de oito cores diferentes, que , quando, selecionadas definem a cor a ser pintada na tela , seguidas de duas ferramentas: a borracha, que quando selecionada preenche os pixels selecionados pelo ponteiro de branco, e o balde, que preenche todo o ecrã de uma cor selecionada .



2. Estado do Projeto

2.1 Tabela de Dispositivos

Dispositivo	Funcionalidade	Interrupções
Keyboard	Uso de atalhos no menu e seleção de cores, filtros e ferramentas no modo desenho	Sim
Mouse	Interação com os diversos Sprites, e seleção de pixels a ser pintados na tela	Sim
Real-Time Clock	Guarda informações sobre o tempo atual, ao atualizar as informações de uma estrutura própria a cada segundo	Não
Timer	A cada segundo, atualiza as informações da estrutura providenciada pelo RTC , e atualiza a tela a cada interrupção	Sim
Video Card	Desenha e mostra o menu, o modo de desenho, as instruções , e o relógio durante toda a instância do programa.	Não

Tabela de Funcionalidades

2.2 Keyboard

O teclado pode ser usado para interagir com o menu, uma vez que possui atalhos que correspondem às suas opções. Assim, pressionando a letra 'G' é iniciado o modo de desenho, pressionando a letra 'H' são mostradas as instruções, e pressionando o ESC termina a instância do programa.
Já no modo de desenho, pressionando em qualquer tecla de 1 a 7 , é selecionada a cor correspondente, que é apresentada no cabeçalho da tela. Pressionando na tecla '8' , é selecionado uma ferramenta (o Balde) e a tecla '9' conduz ao uso da borracha, que se traduz na seleção da cor Branca. Tal como é revelado nas instruções, pressionando no R seleciona-se a ferramenta que constrói retângulos, no P para selecionar novamente o lápis, o M para aplicar o filtro de inversão de cor ,o L para captar as componentes cinzentas de cada cor pertencente ao desenho, o K para reverter a última mudança/pintura feita no desenho global, e por fim as setas para cima e para baixo aumentam e diminuem respetivamente a espessura do traço a ser usado na pintura.

2.3 Mouse

O mouse é usado na interação com o menu, bem como, no modo de desenho, trata da seleção de cores e outras ferramentas (balde e borracha) , bem como baseando na cor e na espessura selecionadas, seleciona o píxel (ou conjunto de píxeis) a ser pintados, tendo como referência o ponteiro.
As posições do ponteiro, transformadas em coordenadas cartesianas , permitem distinguir uma variedade de zonas limitadas nas diversas interfaces que o programa apresenta, a fim de garantir uma interatividade coesa e precisa. Assim, é nomeadamente avaliada a pressão no botão esquerdo, que determina a sequência de píxeis a serem pintados no ecrã (por exemplo, quando é pintada uma linha com o arrasto do rato, ou a seleção de pontos que definem duas extremidades opostas que servem de referência para a pintura de um retângulo entre eles), a fim de distinguir a área de desenho e a área de seleção de cores e ferramentas.

2.4 Real Time Clock

O RTC (Real Time Clock) tem um funcionamento semelhante ao Timer. Possui 6 contadores internos, cada um correspondendo a um parâmetro da data ou hora:
Assim, aquando de um segundo, com o uso da estrutura abstrata real_time_info, as informações por ela guardada (isto é, de ano, mês,

dia e hora) são atualizadas (para tal, há o auxílio do timer que assinala a passagem do segundo o que indica a necessidade de atualização). Assim, em conjunto com a gráfica, a sua informação é traduzida com a impressão dinâmica na tela de sprites que representam números, pelo que a seleção de sprites a serem impressos é feita dígito a dígito, seleção que dependerá da passagem de um segundo , de um minuto , ou de uma hora. A atualização da estrutura é feita com a chamada à função (rtc_update_time), e o desenho do estado do relógio(impressão no buffer secundário) é feita com a chamada updateClock().

2.5 Timer

O Timer é usado para controlar a taxa de atualização de frames feita pela placa gráfica, atualizando o ecrã 60 vezes por segundo, e para atualizar as informações captadas do Real Time Clock, a cada 60 interrupções (isto é, assinalando a passagem de um segundo quando uma variável contadora de interrupções timer counter é incrementada 60 vezes), atualizando a estrutura na qual são guardadas as informações sobre ano, mês, dia , hora , minuto e segundo atuais. A implementação de configurações , funcionalidades e interrupções encontra-se no timer.c

2.6 Video Card

A gráfica é utilizada para imprimir no ecrã os vários estados do programa e o desenho feito em tempo real.
Durante toda a instância do programa, encontramo-nos no modo de vídeo, isto é , um modo VBE 0x115 que define uma resolução de 800 x 600 , cujas cores seguem um modelo de modo direto, visto que são codificadas com base em 8 bits que exprimem a componente vermelha,

8 bits a exprimir a componente verde e 8 bits a exprimir a componente azul, o que permite uma vasta variabilidade de cores compatíveis (cerca de 2^24 cores)
A fim de evitar a presença de fenómenos visuais como flickering, permitir uma interatividade mais fluída , e permitir a conservação do desenho a ser executado a fim de garantir a sua continuidade de frame para frame, foi aplicado um triple buffering. Enquanto o primeiro buffer atualiza o ecrã (visto que são impressos no ecrã os seus conteúdos), o segundo buffer é usado para carregar elementos estáticos (Principalmente zonas fixas como barras , juntamente com xpm's), e o terceiro buffer onde será pintado e guardado o desenho. Com base no algoritmo do Pintor, após a cópia do desenho do terceiro para o segundo buffer (usando a função copyTheDraw()) são "pintados" posteriormente os elementos estáticos do modo de desenho , pelo que, na interrupção do timer seguinte, todos estes conteúdos são copiados (usando a função swap_buffers()) para o buffer principal, que é usado na impressão.
Usamos alguns sprites para criar os diferentes elementos visuais do nosso programa. Esses elementos têm uma forma fixa e não mudam durante a execução do programa. Para criá-los, utilizamos a função "createSprite(xpm_map_t sprite)", que recebe um XPM predefinido. Alguns exemplos desses elementos são o cursor do mouse, o ícone do pincel, do balde e da borracha no modo de desenho e os botões do menu. A implementação das configurações e funcionalidades relacionadas aos gráficos de vídeo pode ser encontrada no ficheiro "graphics.c".

3. Organização e estrutura do código

É importante salientar que nós seguimos uma estrutura *Model-Viewer-Controller*. Desta forma, a implementação dos diversos controladores encontra-se no âmbito do Controller, e compreendem-se nos seguintes:

3.1 Controlador Keyboard

A implementação do KBC e das funções de keyboard seguem as funções desenvolvidas no Lab3 das aulas práticas que puderem ser adaptadas às necessidades do nosso programa. O propósito destas consiste na leitura de dados emitidos pelo KBC, a fim de captar as teclas necessárias ao desencadeamento de ações e ferramentas no nosso programa. Também inclui as funções necessárias à subscrição e configuração de *interrupts*, com a implementação de um handler. Este módulo tem um peso de 15% no projeto.

Contribuidores:

- André Tiago Moreira Soares da Costa e Silva
- Bernardo Ferreira Campos
- João Tomás Cardinal Pinho Teixeira

3.2 Controlador Mouse

A implementação do KBC e das funções de mouse seguem as funções desenvolvidas no Lab4 das aulas práticas que puderem ser adaptadas às necessidades do nosso programa. O propósito destas consiste na leitura de dados emitidos pelo KBC, a fim de construir pacotes (“packets”). Estes servem para fornecer informações sobre o rato , tais como o deslocamento do rato nos eixos cartesianos, a sua localização atual no plano, e a pressão dos três botões principais (esquerdo, central e direito) . Os packets são construídos à base de interrupts que o rato vai emitindo, pelo que cada interrupt está responsável por construir um pacote e garantir a sua sincronização (visto que a construção de pacotes pode não coincidir com a sequencialização dos interrupts, e gerar pacotes “falsos” caso não haver esta sincronização) a fim de providenciar informação precisa sobre o rato Também inclui as funções necessárias à subscrição e configuração de interrupts, e a implementação de um handler, usando de forma semelhante ao keyboard o KBC . Este módulo tem um peso de 15% no projeto.

- André Tiago Moreira Soares da Costa e Silva
- Bernardo Ferreira Campos
- João Tomás Cardinal Pinho Teixeira

3.3 Controlador Timer

A implementação do Timer baseia-se nas funções desenvolvidas no Lab2 das aulas práticas que puderam ser reutilizadas no contexto do nosso projeto. Neste caso em particular , salientam-se as funções que ,com base nos interrupts recebidos do timer, permitiram marcar o ritmo de execução do nosso projeto (tal como no assinalar a passagem dos segundos) , e permitir a subscrição de interrupts, que contribuem para a sincronização de diversos dispositivos usados (veja-se a título de exemplo a atualização da informação recebida pelo RTC, e a atualização constante da imagem a ser impressa pela gráfica, isto é, a cada interrupt). Este módulo tem um peso de 10% no projeto. Este módulo tem um peso de 10% no projeto.

Contribuidores:

- André Tiago Moreira Soares da Costa e Silva
- Bernardo Ferreira Campos
- João Tomás Cardinal Pinho Teixeira

3.4 Controlador da Gráfica (VBE)

A implementação da Gráfica baseia-se nas funções desenvolvidas no Lab5 das aulas práticas que puderam ser reutilizadas no contexto do nosso projeto. Desta forma, salientam-se as funções que permitiram fazer a configuração do modo gráfico e construir certos elementos no ecrã, estes que poderiam (ou não) ser à base de sprites, que eram construídos conforme os *xpms* que lhes eram atribuídos. No contexto do projeto, foi importante para servir de base à estrutura geral onde assentam as funcionalidades principais do nosso projeto, conferindo-lhes uma interface gráfica. Este módulo tem um peso de 15% no projeto.

Contribuidores:

- André Tiago Moreira Soares da Costa e Silva
- Bernardo Ferreira Campos
- João Tomás Cardinal Pinho Teixeira

3.5 Uso de utils

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas , e tiveram um papel fundamental na implementação dos controladores anteriores (de mouse , keyboard e timer), pelo que acabaram por ser reutilizadas no nosso Projeto. Este módulo tem um peso de 5% no projeto.

Contribuidores:

- André Tiago Moreira Soares da Costa e Silva
- Bernardo Ferreira Campos
- João Tomás Cardinal Pinho Teixeira

3.6 As funções da componente Model

O Model no âmbito do nosso projeto serviu para controlar a lógica por detrás dos estados que compõem o nosso programa. Isto é, estes estados definem as ações a ser executadas com o uso dos controladores da componente controller, e sinalizam o que deve ser apresentado pelo View. Sendo assim, estes incorporam a criação dos Sprites interativos (interação conseguida com o controller), e que vêm a ganhar forma com o Viewer. Neste sentido , também é no model que são definidos os diversos estados que constituem e se conjugam no nosso programa. Estes consistem no System State, um estado binário que define se a instância do programa está ativa , pelo que este estado é verificado constantemente no loop de interrupções, que deverá acabar quando este estado muda, no MenuState, que define em que página é que a instância do programa do programa se encontra, isto é, no âmbito do nosso projeto, se deverá ser no view desenhado o menu principal, as instruções, ou o menu de desenho, e as interações a nível de controller podem vir a mudar este state, desencadeando um novo comportamento no View. E , por fim , um ToolState , que apenas tem relevância e é verificado quando o MenuState corresponde ao menu de desenho, uma vez que este indica qual ferramenta o

ponteiro vai utilizar na interação com a tela de desenho, pelo que este variará entre o lápis, a borracha , e o pincel. Este módulo tem um peso de 15% no projeto.

Contribuidores:

- André Tiago Moreira Soares da Costa e Silva
- Bernardo Ferreira Campos
- João Tomás Cardinal Pinho Teixeira

3.7 A influência das funções construídas a nível do View

O View, enquanto parte constituinte da organização do código, trata de materializar e expor no ecrã o significado das interações que acontecem a nível do Controller e a importância do estado refletido no Model. Desta forma, é no view que são chamadas as funções principais de pintura, tais como a sinalização dos pixels circundantes do ponteiro, o resultado do desenho de retângulos , a consequência de alterar a espessura do pincel após interação do teclado , o uso dos filtros que atuam apenas a nível aspectual e evidenciam certas características da imagem, e a ação proveniente de certas ferramentas , como o balde , ou o pincel. Assim, o impacto do View acaba por ser mais notório apenas com a presença e interação do VBE. Este módulo tem um peso de 20% no projeto.

3.8 Controlador do RTC

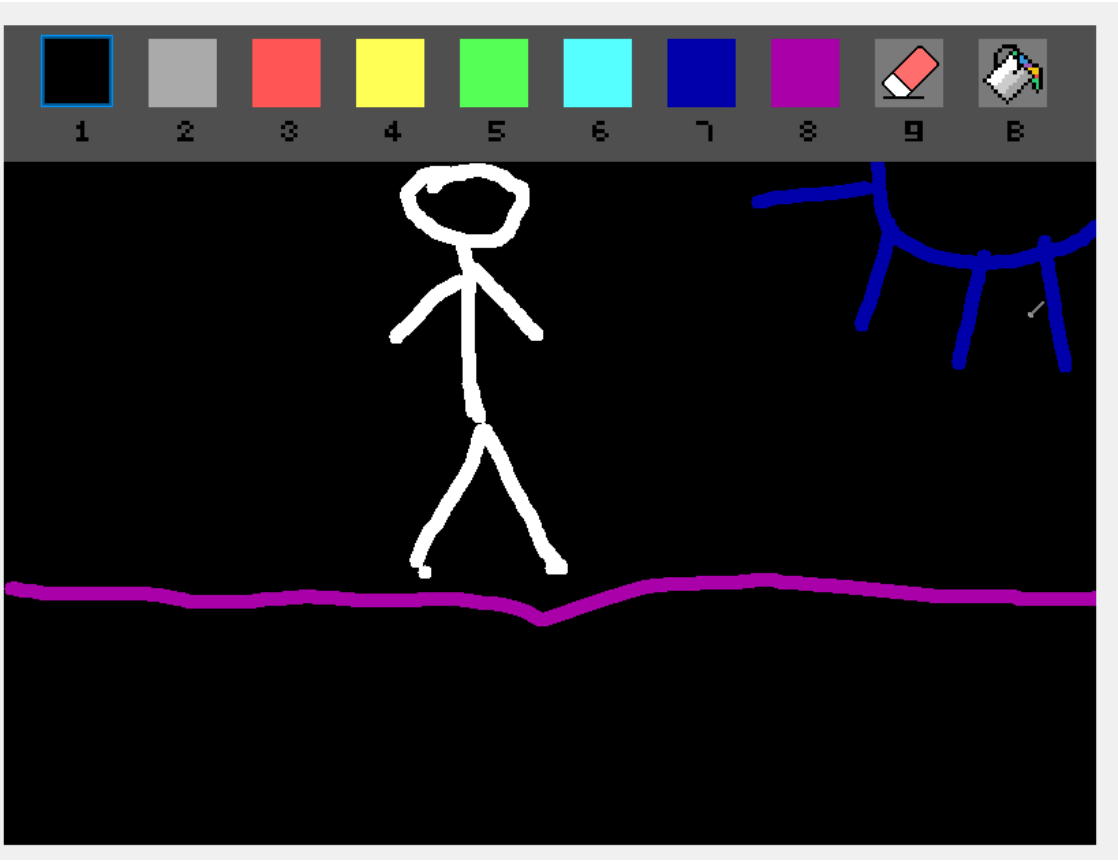
Visto que o RTC não foi abordado nas aulas práticas, este acabou por ser fruto de uma pesquisa pelos labs antigos , a fim de perceber quais as adaptações possíveis das suas funções para certas funcionalidades do nosso projeto. Neste caso em particular, o RTC apenas trata de guardar informações em tempo real sobre a data/hora, e serviu apenas para ser uma fonte impressa do estado do relógio atual. Assim, o uso do RTC é apenas evidenciado juntamente com o uso da gráfica e das funções providenciadas pelo módulo View.
Este módulo tem um peso de 5% no projeto.

3.9 Função “call graph”

4. Detalhes de implementação

Uso de Triple Buffering e alteração remota de desenho

Tendo em conta o papel que os diversos dispositivos tiveram no desenvolvimento deste projeto, a implementação que dá lugar à funcionalidade chave deste projeto é a implementação de um buffer dedicado à conservação e futura cópia do desenho a ser produzido. Assim, tendo em conta que qualquer alteração no terceiro buffer acabaria por ter impacto no desenho, foi possível implementar com relativa facilidade as funcionalidades que vieram do acréscimo do desenho. Isto é, assim é possível manipular num todo o desenho , ao influenciar a gama de cores que este apresenta, bem como guardar com uma relativa facilidade o seu estado, a fim de ser possível reverter ações a nível gráfico. Veja-se , como exemplo, o uso de filtros implementados, e a capacidade singular de reverter a última alteração feita ao desenho (isto é, reverter , quer um retângulo, quer o uso do balde/borracha, quer um traço irregular contínuo, numa dada instância).



5. Conclusões

O esforço foi bem distribuído pelos três elementos, com uma comunicação ativa e frequente, e constante divisão de tarefas. Contudo , nem todas as funcionalidades propostas para o início do projeto foram cumpridas, uma vez que houve dificuldades em implementar a Serial Port, pelo que não foi possível implementar o desenho cooperativo, bem como também não foi estendida a funcionalidade da gráfica, uma vez que não foi possível criar uma animação entre os desenhos selecionados, visto que não foi possível exportar esses desenhos para o formato pretendido.