

UNIVERSIDADE DE BRASÍLIA



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

TURMA C

---

## Trabalho 1

---

*Nome:*

*Johannes Peter Schulte*

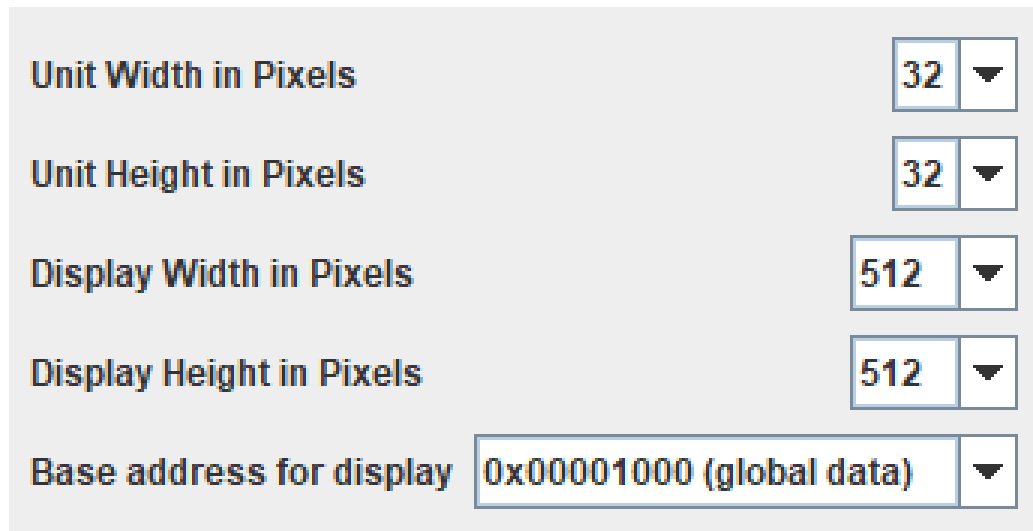
*Matrícula:*

15/0132662

ABRIL DE 2019

## 1 Configuração

Configuração de memória: Compact, Text at Address 0.  
Endereço bitmap: 0x00001000 (global data).  
Resolução: 512x512(com pixel 32x32) ou 128x128(pixel 8x8).



The image shows a configuration window with five settings, each with a text label and a value in a box with a dropdown arrow:

Label	Value
Unit Width in Pixels	32
Unit Height in Pixels	32
Display Width in Pixels	512
Display Height in Pixels	512
Base address for display	0x00001000 (global data)

Figura 1: Configuração recomendada.

## 2 Descrição

Antes de começar o programa, são criados os espaços para as duas matrizes, utilizando .space 0x400 (16x16 words) na memória. Também é declarado o endereço inicial do bitmap em 0x1000. O motivo da escolha foi para que os valores armazenados nos registradores não interferissem com as matrizes.

Logo ao iniciar o programa, a matriz A e B são criadas e populadas com 0s. Com elas criadas, são então inicializados os endereços que serão utilizados no decorrer do programa nos registradores de s0 até s3. Seguindo a inicialização, o programa pega a entrada das coordenadas iniciais pelo teclado, até que 0 seja apertado e a execução do jogo começa.

A execução do jogo começa percorrendo a matriz A, que possui as coordenadas iniciais do jogo. Para cada pixel, são testados os 8 endereços vizinhos para decidir seu próximo estado, que é armazenado em B. O programa continua sua execução alternando entre as matrizes A e B.

### 3 Funções

Função init: Responsável por inicializar a execução do programa. Primeiro armazena os endereços nos registradores estáticos(s0-s3) que serão utilizados no resto do programa, depois recebe a entrada de dois inteiros para a coordenada inicial, usando e chamada de sistema 5. Para cada coordenada (x, y) o programa calcula o endereço que essa coordenada tem dentro da matriz(64 endereços por linha, 4 por coluna), esse valor é armazenado em s4 e a função de estado é chamada, que altera o valor na matriz para 1.

Função estadoA/B: Utilizada para alternar entre a matriz A e B, com o valor armazenado em s5(0 para matriz A, 1 para matriz B).

Função tela: É a função que preenche a tela com a cor de fundo e pausa o programa com a chamada de sistema 32, que pausa a execução por 1000ms. Ela também retorna s0 para o endereço inicial (após percorrer s0 até o final para preencher o fundo) e inicializar as variáveis que serão utilizadas na próxima função.

Função percorre: Percorre a matriz A/B pixel por pixel, verificando se a posição deve ser preenchida(chama a função pixel), se a matriz terminou(chama a função fim) e também incrementa as variáveis que armazenam a posição (x,y) atual com s10 e s9, que são usados para fazer os testes com os vizinhos na função verifica.

Função verifica: Faz a verificação de cada um dos oito endereços vizinhos, incrementando s11 sempre que houver um vizinho. Baseado no valor de s11, a função chama morte, caso o estado do pixel deva ser 0, vida, caso seja 1, ou cont para manter o estado atual(0 ou 1).

No final da execução, a função estado é chamada novamente com a próxima matriz.