Estevam Galvão Albuquerque – 16/0005663 – Turma C

PROJETO 1 - GAME OF LIFE

OBJETIVO

Este trabalho objetiva a prática da programação em assembler do RISC-V utilizando os recursos do simulador RARS. O trabalho consiste em desenvolver uma implementação em assembler do Jogo da Vida, exibindo o resultado de sua evolução através da interface gráfica do RARS, com o auxílio da ferramenta de exibição de saída gráfica mapeada em memória.

DESCRIÇÃO

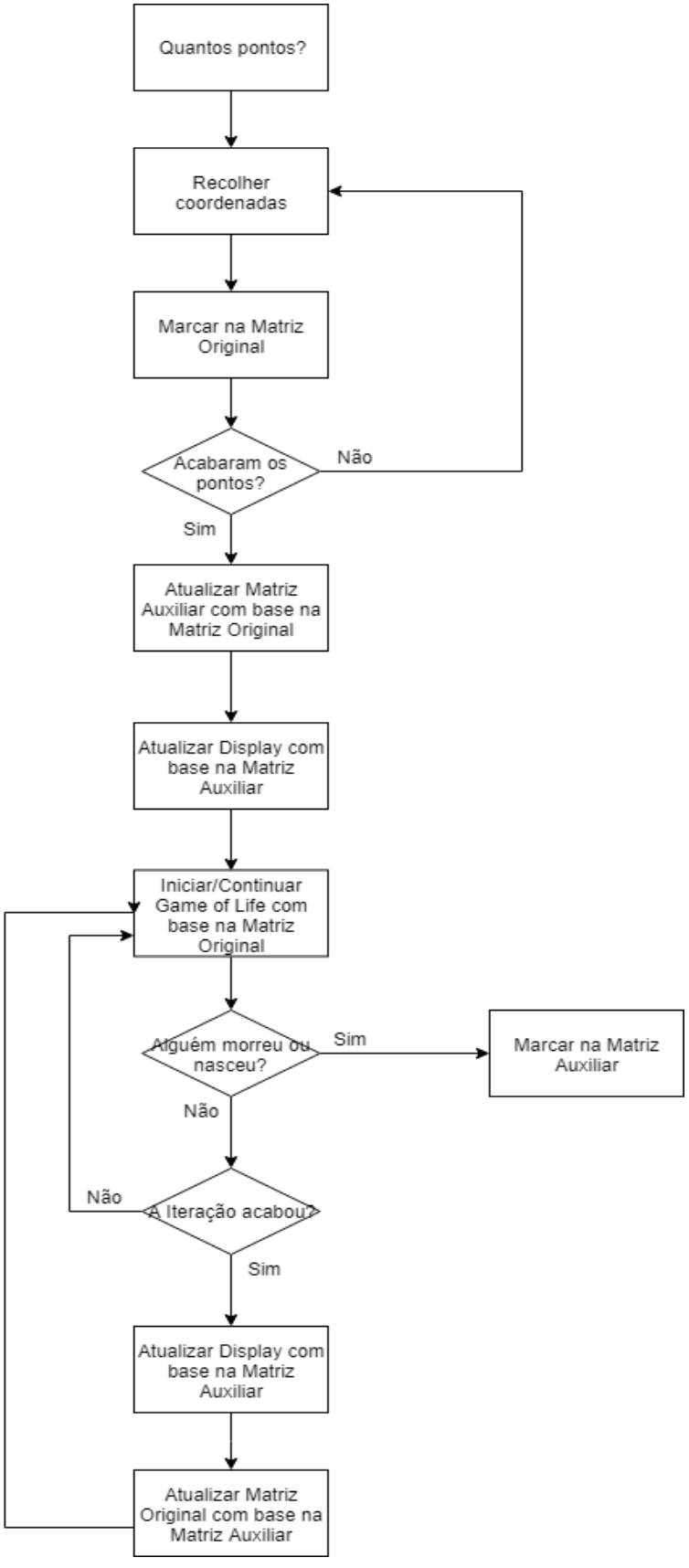
Para concretização do algoritmo, pensei em uma estratégia utilizando 3 matrizes.

i. Matriz Original: É uma matriz do tipo ".word", 18x18. É nela que são feitas as verificações do jogo. E é com base nela que é decidido quem vive, morre e sobrevive.
ii. Display: É uma matriz do tipo ".word", 16x16. É com ela que fazemos a exibição do jogo no Bitmap Display do RARS.
iii. Matriz Auxiliar: É uma matriz do tipo ".word", 16x16. Ela foi criada para auxiliar a matriz original em armazenar as decisões do jogo na rodada "n", sem que isso afete a rodada "n+1" antes que a "n" finalize.

Dito isso, o algoritmo começa pedindo ao usuário os pontos iniciais do jogo, após esses pontos serem definidos, recolhemos e marcamos na matriz original cada um deles e em seguida copiamos para matriz auxiliar e então passamos os pontos para o display. Iniciamos o jogo logo em seguida utilizando a matriz original como base e cada decisão de vida ou morte é escrita na matriz auxiliar, enquanto a matriz original permanece intacta. Depois disso copiamos o conteúdo da matriz auxiliar para o display e para matriz original , onde então, reiniciamos o jogo.

FLUXOGRAMA:

Ao invés de pensar em um pseudo-código, antes de implementar o algoritmo, eu criava fluxogramas pois encontrei mais facilidade em visualizar o código em si com essa estratégia.



FUNÇÕES IMPLEMENTADAS

userStampMatrix(x, y, matrix adress):

i. É uma função que percorre a matriz original, inicializada com zeros, até o ponto digitado pelo usuário e então inverte seu valor.

updateDisplay(auxiliar matrix adress, display adress):

i. É uma função que percorre a matriz auxiliar e o display para comparar seus pixels e alterar no display os que estiverem diferentes dos da matriz auxiliar. Basicamente, replica os pixels da matriz auxiliar no display.

updateAuxMatrix(matrix adress, auxiliar matrix adress):

i. É uma função que basicamente replica os pixels da matriz original na matriz auxiliar. É usada somente no início do código para receber os pixels marcados pelo usuário.

updateMatrix(matrix adress, auxiliar matrix adress):

i. É uma função que basicamente replica os pixels da matriz auxiliar na matriz original.

play(matrix adress, auxiliar matrix adress):

i. É a função que executa as regras do jogo propriamente dito.
ii. Percorre a matriz original fazendo a verificação de 8 vizinhos mais próximos e se o pixel atual está vivo ou morto, com base nisso julgará quais pixels morreram, sobreviveram ou nasceram na próxima iteração. O resultado é armazenado na matriz auxiliar.

“getPoints”():

i. É uma função que está na “main” e não recebe nada pela pilha pois pede diretamente ao usuário via syscalls.
ii. Recebe quantos pontos o usuário irá digitar e quais serão eles.
iii. Funciona em conjunto com a userStampMatrix.