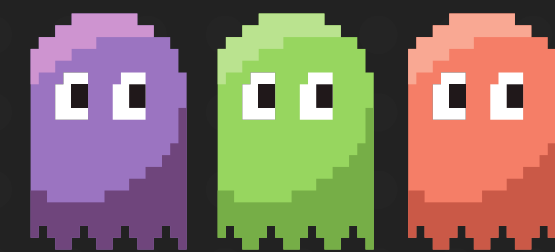


PAC-MAN



FEV / 23

LEONARDO GUEDES DA FONSECA PASSOS
(222031493)



UnB | CIC

Abstract



O presente trabalho consiste em uma versão do Jogo Pac-man desenvolvido pelo aluno Leonardo Guedes da Fonseca Passos, como requisito para parovação na disciplina Introdução aos Sistemas Computacionais (ISC) da Universidade de Brasília, ministrada pelo professor Marcus Vinicius Lamar. Referido jogo tem por objetivo percorrer um labirinto comendo as rações e fugindo de fantasmas, e, para reproduzi-lo, o aluno utilizou a arquitetura RISC-V e o Simulador RARS. Serão tratados aqui, todos os passos para o desenvolvimento do projeto, as dificuldades do aluno na elaboração do jogo, a arquitetura utilizada para desenvolvê-lo e os resultados alcançados.



Introdução

O jogo Pac-man foi criado em 1980 por Toru Iwatani no Japão[1], funcionário da Namco, empresa desenvolvedora de jogos eletrônicos. O conceito do jogo é simples, o jogador controla o personagem que deve comer as rações dentro de um labirinto, com o obstáculo de fugir de quatro fantasmas, que ao encostar no Pac-man, faz com que perca uma vida. Ao comer todas as rações, passa-se de fase, para um novo labirinto.

Devido a limitações técnicas e de tempo, o escopo do presente trabalho não é de reproduzir o jogo original com todas suas regras e complexidades. O presente projeto resultou em um jogo semelhante ao original, mas sem a implementação de algumas funcionalidades que serão enumeradas nos próximos tópicos.

[1] Hadan, F. Pac-man (Multi): o Famoso “come-come completa 40 anos de muita diversão. Tecnoblog.net. 2020. Disponível em <https://www.gameblast.com.br/2020/06/pac-man-arcade-40-anos.html#:~:text=Pac%2DMan%2C%20conhecido%20originalmente%20como,o%20personagem%20%C3%A9%20mundialmente%20conhecido>. Acesso em 07/02/2023.



Metodologia

O jogo foi implementado de acordo com os seguintes passos:

1) Inicialmente, uma função (PRINT) é acionada para imprimir no bitmap display o mapa do jogo em dois frames que ficam se sucedendo na tela.

2) Em seguida, tem-se um loop em que os seguintes comandos são feitos;

2.1) Em primeiro lugar o usuário é acionado, para, através do teclado, fornecer o input, que representa a movimentação do personagem principal para cima, para baixo, para esquerda ou para direita.

2.2) A função PRINT promove a movimentação do personagem conforme input do usuário.

2.3) Posteriormente, a função APAGARASTROFAN exclui a imagem do personagem da posição em que ele estava antes da movimentação. O personagem, então, estará impresso na tela na nova posição e o seu rastro será apagado da última posição.

2.4) Em seguida, um número pseudoaleatório vai ser gerado para cada fantasma e de acordo com esse número, cada fantasma será movimentado em alguma direção e o seu rastro será apagado com a mesma metodologia utilizada para apagar o rastro do personagem. Esse loop vai ficar sendo gerado até que o jogo termine, seja porque o personagem percorreu todos os blocos em que havia ração para ele comer, seja porque ele encostou em algum fantasma e morreu.

O objetivo do jogo é comer todas as rações sem ser tocado ou encostar nos fantasmas.

Dificuldades

O Autor, na elaboração do presente trabalho, esbarrou nas seguintes dificuldades:

1) A primeira e maior dificuldade foi ter que programar em uma linguagem de baixo nível (assembly). Isso porque, em linguagens desse tipo, não existem variáveis à disposição do programador, de forma tal que é necessário usar apenas os registradores do processador para fazer o código.

2) Outra dificuldade foi desenhar os mapas em que o jogo se desenrola. É que o Autor, em virtude de sua inexperiência com esse tipo de projeto, elaborou os mapas especificando manualmente cada um dos bits que o compunha, em vez de utilizar ferramentas mais apropriadas a essa finalidade.

3) Mais uma dificuldade encontrada foi na elaboração da função PRINT, que imprime o mapa, os personagens e todos os componentes que integram o jogo. Isso porque cada um desses elementos tem um formato, uma largura, um comprimento e demorou até que se pensasse em uma lógica que permitisse a impressão de todos esses elementos com desempenho aceitável para o jogo.

4) Encontrou-se dificuldade também para criar a lógica de movimentação dos personagens, especialmente, a obtenção de um número aleatório. Isso porque os fantasmas, no jogo criado, movimentam-se aleatoriamente, e criar uma lógica que trouxesse essa aleatoriedade foi bastante desafiador.

5) Outro ponto desafiador foi identificar no mapa impresso quais espaços os personagens poderiam transitar e o que deveria ocorrer quando eles o fizessem. Por exemplo, os personagens só poderiam transitar pelos quadrados vermelhos e deveriam ser impedidos de percorrer os quadrados pretos. Se o quadrado vermelho contivesse uma ração, o personagem, ao passar por ali, deveria aumentar seu score, ao passo que se o personagem transitasse por um quadrado vermelho sem bolacha, o seu score deveria permanecer inalterado. Como todos os bits do mapa foram escritos manualmente, não foi simples percorrer o mapa para identificar quais os bits que compunham cada quadrado para então dar o tratamento adequado a este quadrado.

Resultados

Elaborado o jogo de acordo com as especificações expostas nos tópicos anteriores, os resultados obtidos foram os seguintes.

1) Foram implementadas três fases com mapas diferentes conforme figuras a seguir:

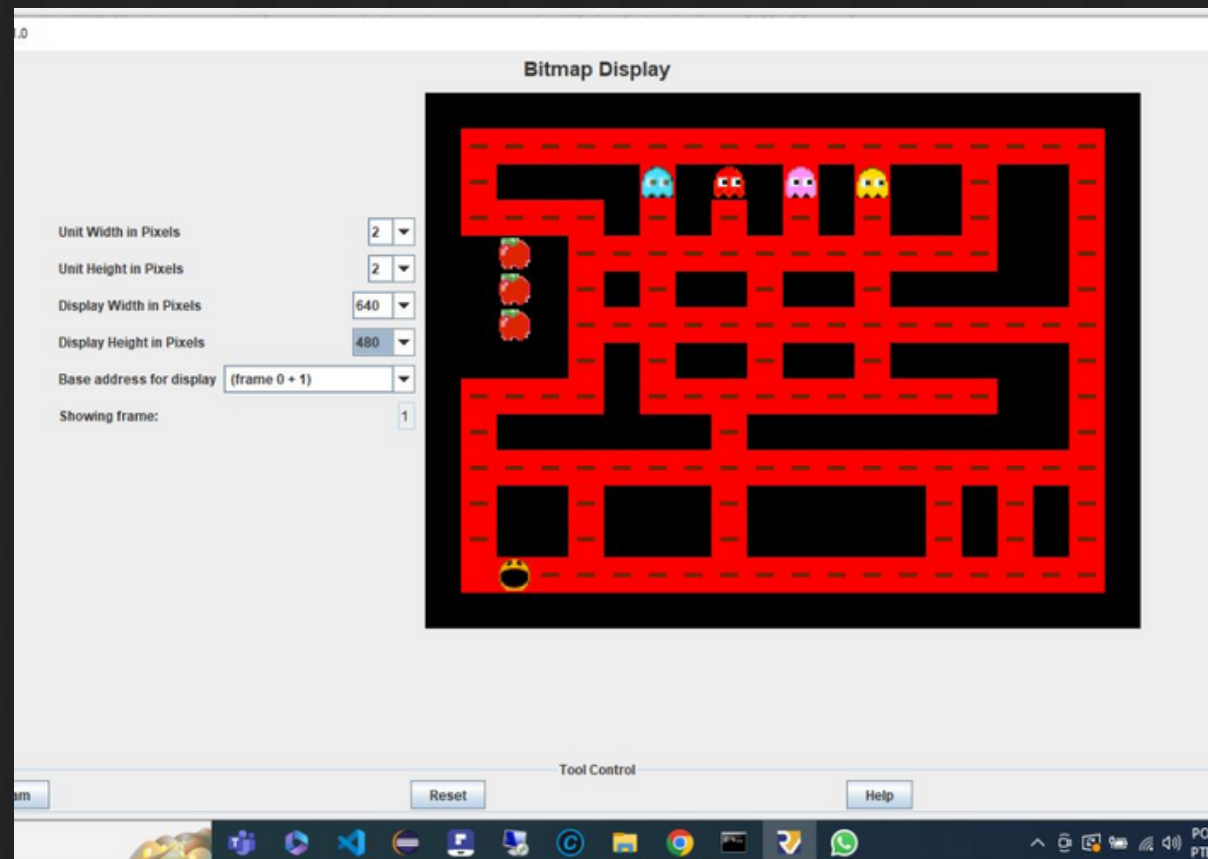


Figura 01

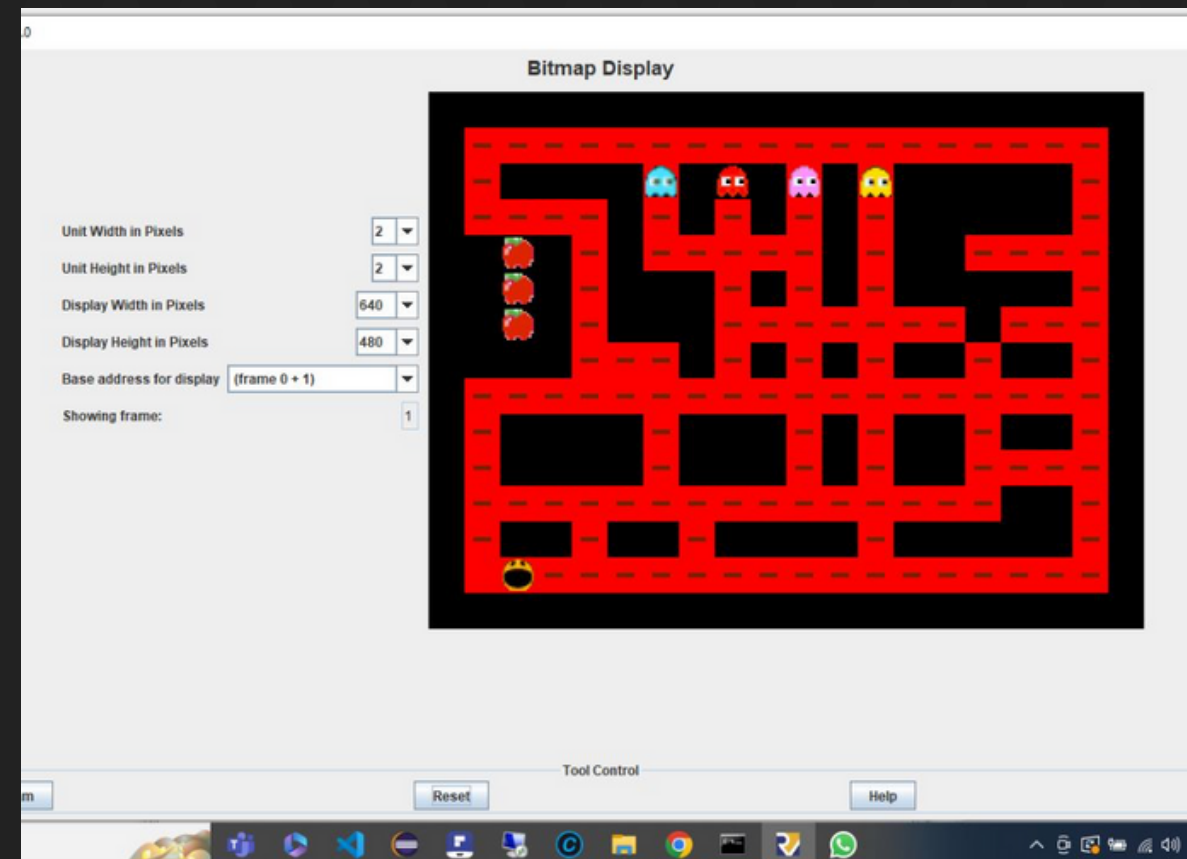


Figura 02

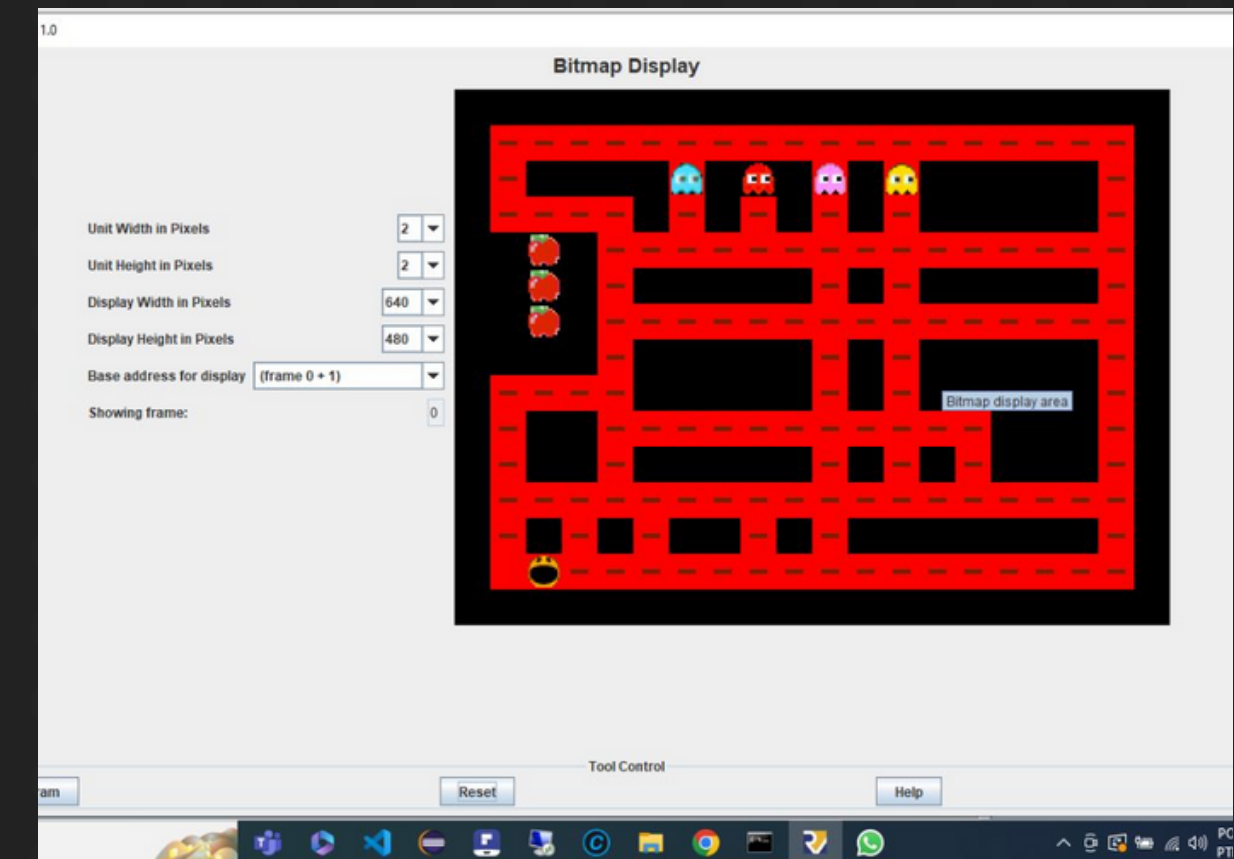


Figura 03

2) Em cada um desses mapas, foram distribuídas as rações que o personagem deveria comer. A condição de vitória quando ele consegue comer todas as rações foi implementada com sucesso.

Resultados

3) Foram elaboradas as telas de início e de final, tanto para o caso de vitória como de derrota do personagem, conforme figuras a seguir:



Figura 04



Figura 05



Figura 06

4) Foram implementadas com sucesso todas as interfaces: interface gráfica (mapas), interface de Input (KDMIO) e de áudio (MIDI), esta última para acionar o som de movimento do personagem.

Infelizmente, devido ao curto espaço de tempo disponível para elaboração do trabalho, especialmente porque o semestre letivo 2022.2 foi reduzido em função dos efeitos da pandemia, não se conseguiu implementar os seguintes aspectos do jogo: a) não foi possível a elaboração de uma lógica em que o personagem e os fantasmas não se sobrepussem em determinados momentos. Assim, em certas ocasiões do jogo, um fantasma acaba tampando o outro ou o personagem (e vice e versa). b) não foi possível implementar a condição de derrota quando o personagem encosta em algum dos fantasmas. c) não foi construído o menu que gerencia a quantidade de vidas, o score do personagem e as informações desse tipo; d) os fantasmas movimentam-se de forma totalmente aleatória, já que não foi possível desenvolver uma lógica que lhes conferisse algum tipo de inteligência artificial.

CONCLUSÃO

Implementado o projeto tal qual descrito nos tópicos anteriores, conclui-se que, apesar das dificuldades de se programar numa linguagem de baixo nível, como assembly, com algum esforço, é possível desenvolver projetos utilizando esse tipo de linguagem a partir da simulação que o RARS faz do processador RISC-V.

Apesar do curto período para o desenvolvimento do projeto, o autor pretende continuá-lo até conseguir implementar todas as funcionalidades que se amoldem às regras do PAC-MAN.

