

# PACMOUSE

Gabriel Alexandre Gomes  
Henrique de Almeida Santos  
Nyvea Paulla Tavares Monteiro

Departamento de Ciência da Computação, Universidade de Brasília, Brasil

**Resumo:** *Este artigo descreve o processo de criação da releitura do jogo “Pac-Man”, projeto proposto na disciplina de Introdução aos Sistemas Computacionais(ISC), ministrada pelo Prof.Dr. Marcus Vinicius Lamar<sup>1</sup>. Com a arquitetura RISC-V e dos programas RARS<sup>2</sup> e FPGRARS<sup>3</sup>, que permitem simular este ambiente. O projeto teve como fim colocar em prática os conhecimentos adquiridos, durante as aulas ministradas em ISC, sobre as mecânicas da linguagem de programação Assembly, bem como a manipulação de dados armazenados na memória do computador. Serão abordados neste artigo a metodologia utilizada, arquitetura do desenvolvimento, dificuldades enfrentadas e os resultados obtidos.*

## 1 - INTRODUÇÃO

Pac-Man é um clássico dos jogos, lançado em 22 de maio de 1980, pela produtora Namco. A ideia original surgiu após Toru Iwatani, criador do jogo, observar uma pizza sem uma de suas fatias em um jantar entre amigos, e esse fator foi responsável por criar algo que iria revolucionar o mundo dos jogos, com elementos nunca antes vistos, como por exemplo a mecânica de *power-up* (o que concede ao personagem mais poder por um certo período de tempo), além de apresentar uma complexa inteligência artificial, para a época, já que cada um dos inimigos no jogo possuía uma função distinta dos demais. O jogo foi, e até os dias atuais é, um sucesso.

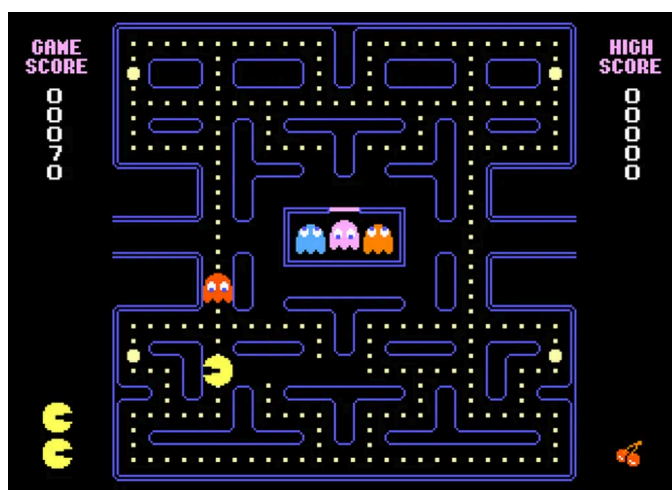


Imagem 1: Layout do jogo Pac-Man

Tomando por referência o jogo original, fizemos uma releitura, chamada “Pacmouse”. Trata-se de um camundongo que está preso em uma cozinha repleta de comida, mas também, de gatos dispostos a eliminar a ameaça à higiene do ambiente. Em um labirinto cheio de comida é preciso recolher as migalhas e super queijos presentes durante o nível. Ao coletar todos os farelos de comida da cozinha, o camundongo avança para o próximo desafio.

<sup>1</sup> <http://lattes.cnpq.br/0125818896954714>

<sup>2</sup> <https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf>

<sup>3</sup> <https://github.com/LeoRiether/FPGRARS>

## 2 - METODOLOGIA

Para o desenvolvimento deste projeto, foram utilizados os simuladores RARS e FPGRARS como ambientes de programação e teste, respectivamente, para trabalhar com a arquitetura de conjunto de instruções RISC-V.

### 2.1 - Análise

O primeiro passo dado para o início do projeto foi a análise do jogo original. A partir das referências disponibilizadas no documento, com as especificações do projeto<sup>4 5</sup> e outras referências encontradas por meio de pesquisas<sup>6 7</sup>, tais como os recursos disponibilizados por alunos de semestres anteriores na plataforma GitHub<sup>8</sup>, pudemos começar a pensar nos elementos que gostaríamos de implementar em nosso jogo, bem como as possíveis maneiras de fazê-lo.

### 2.2 - Design Gráfico

Após decidir como seria nossa releitura do clássico Pac-Man, partimos para o desenvolvimento da parte visual do jogo. Decidimos os padrões dos labirintos, o fundo (que trata-se de um chão de cozinha), o design do nosso personagem principal e de seus inimigos, além dos elementos coletáveis durante as fases do jogo.



Imagem 2: Labirinto 1



Imagem 3: Labirinto 2



Imagem 4: Elementos do jogo

<sup>4</sup> <https://freepacman.org/>

<sup>5</sup> <https://www.youtube.com/watch?v=IKbGgjNvdak>

<sup>6</sup> <https://www.youtube.com/watch?v=ataGotQ7ir8>

<sup>7</sup> <https://shaunlebron.github.io/pacman-mazegen/>

<sup>8</sup> <https://github.com/victorlisboa/LAMAR>

## 2.3 - Música

Inicialmente, a ideia era implementar uma música com diversas linhas melódicas diferentes, o que demonstrou ser extremamente trabalhoso. Com isso, a decisão não seguiu adiante, e optamos por utilizar uma só linha melódica para facilitar a implementação e evitar problemas desnecessários. A escolha para música foi “Der Erlkönig”, de Franz Schubert. Utilizando a partitura adaptada de Ernst para o violino<sup>9</sup>, criamos o arquivo midi, e a partir disso, conseguimos os valores dos tons das notas e suas durações, para que assim, fossem implementadas no RARS.



Imagem 5: Partitura

## 2.4 - Desenvolvimento

Com todas essas ideias, o verdadeiro problema veio na implementação. As dificuldades enfrentadas serão discutidas com mais detalhes adiante. Infelizmente, não foi possível implementar a maior parte dos requisitos apresentados pelo projeto final, seja por uma série de fatores.

## 3 - DIFICULDADES

Listamos aqui algumas das várias dificuldades que enfrentamos durante o projeto.

### 3.1 - Linguagem

A linguagem de programação Assembly é algo completamente novo para todos os integrantes do grupo, o que foi uma grande barreira, por se tratar de uma linguagem de baixo nível, que vai além das demais, principalmente considerando que estávamos acostumados a programar na linguagem C na disciplina de Algoritmos e Programação de Computadores(APC).

### 3.2 - RARS

Apesar de ser uma boa ferramenta, o RARS apresenta diversas limitações que prejudicam a compreensão e identificação de certos tipos de erros, o que atrasou (e muito) o desenvolvimento do nosso projeto. Ainda que o FPGRARS apresenta diversas melhorias em relação ao RARS , não há possibilidade de saber em qual linha o código está com problemas. Se há problemas, o FPGRARS apenas encerra o programa subitamente.

### 3.3 - Registradores e memória

A manipulação de registradores e de dados na memória também mostrou-se mais complicada do que imaginávamos, devido à ausência de variáveis. É extremamente difícil trabalhar com um número tão limitado de registradores, e manipular dados contidos diretamente na memória é algo novo, e até

<sup>9</sup> <https://musescore.com/musemeister/ernst-erlkonig>

mesmo contra-intuitivo. Todos esses aspectos tornaram nossa jornada ainda mais complicada, e culminaram na impossibilidade da entrega do projeto completo no prazo estipulado.

#### 4 - RESULTADOS OBTIDOS

Como frizado neste artigo, não cumprimos com todas as especificações solicitadas para o projeto. Apesar disso, conseguimos um jogo onde o fundo é impresso, tal como o personagem principal, que é capaz de movimentar-se, em 4 (quatro) direções distintas, de acordo o input do usuário no teclado, e há trilha sonora em loop, implementada através das syscalls midi.

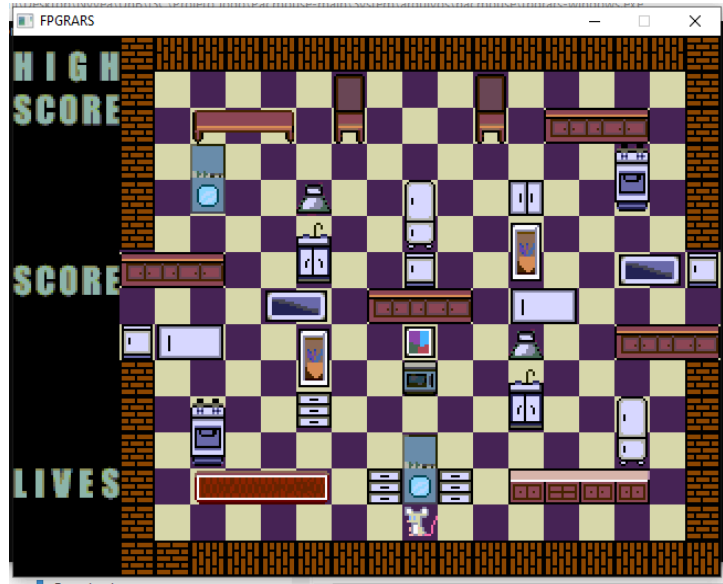


Imagem 6: Jogo no FPGRARS

#### 5 - CONCLUSÃO

Por fim, em virtude do que foi apresentado ao longo do artigo, conclui-se que para programar em Assembly é necessário quebrar certos paradigmas, paciência e conhecimento, por se tratar de uma linguagem de baixo nível. Por fim, apesar de não termos finalizado o projeto, nós fizemos tudo em nosso alcance para desenvolver esse projeto o máximo possível, e esperamos que a experiência obtida neste semestre nos auxilie futuramente na disciplina de Organização e Arquitetura de Computadores(OAC) e em outras também.

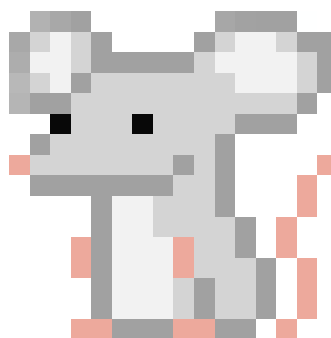


Imagem 7: Nosso protagonista