

PAC-MARIO

Universidade De Brasília, Departamento de Ciência da Computação

Disciplina: CIC0003 – Introdução aos Sistemas

Computacionais – Turma 01 2024/1

Professor: Marcus Vinicius Lamar

Paulo Henrique Sales Freitas

Daniel Florencio Hollenbach

Davi Brasileiro Gomes

1-Resumo:

Este trabalho apresenta uma releitura do jogo Pac-Man, incorporando elementos do jogo Super Mario Bros. Desenvolvido pelos alunos Davi Brasileiro Gomes, Daniel Florencio Hollenbach e Paulo Henrique Sales Freitas, da Universidade de Brasília, o projeto faz parte da disciplina "Introdução aos Sistemas Computacionais", ministrada pelo professor Marcus Vinicius Lamar. O jogo foi produzido utilizando a arquitetura RISC-V e o simulador RARS. Serão abordados neste arquivo, o jogo original, as dificuldades enfrentadas durante o desenvolvimento, a arquitetura utilizada, os resultados obtidos e a conclusão sobre o projeto.

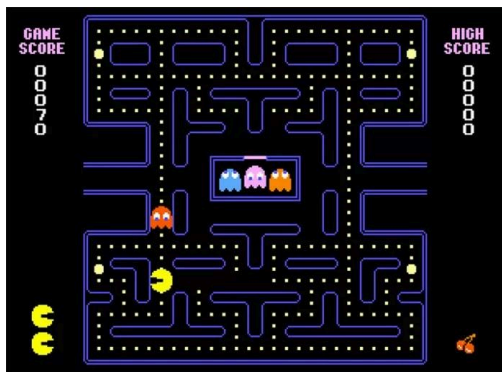


Figura 1: Pac-man 1980

2-Introdução:

A cada dia percebemos a expansão dos jogos no meio social, digital e intelectual. Isso ocorre principalmente porque os jogos, em geral, têm uma forma de interação muito maior e diferente da maioria dos tipos de mídia. Por isso, neste projeto, resolvemos homenagear dois famosos jogos da "era de ouro dos jogos": o "Pac-Man", que foi um marco em 1980 (Figura 1), e o "Super

Mario Bros", um sucesso estrondoso lançado em 1985. Neste projeto, mesclamos os dois jogos, utilizando como base o "Pac-Man", porém com sprites e mapas baseados em "Super Mario Bros". Os inimigos são os clássicos Koopa Troopas, uma raça fictícia de personagens semelhantes a tartarugas da franquia Super Mario Bros. No entanto, a jogabilidade é similar à de Pac-Man, utilizando as teclas W, A, S e D do teclado para movimentar o personagem, seguindo a ordem das teclas (Cima, Esquerda, Baixo,

Direita). O jogo apresenta um nível de dificuldade considerável, pois a inteligência dos inimigos pode parecer um pouco aleatória, embora não seja. Assim, no segundo mapa, a dificuldade aumenta, tornando mais desafiador escapar e concluir o jogo.

3-Metodologia:

O projeto foi desenvolvido utilizando o simulador RARS na arquitetura RISC-V. Inicialmente, concentramos nossos esforços na parte que consideramos mais desafiadora, focando especialmente na primeira fase, que incluía a renderização do mapa, dos inimigos e do protagonista. Acreditamos que essa abordagem nos permitiria entender melhor o que estávamos criando, pois podíamos visualizar todo o processo em ação. Após finalizar essa etapa, implementamos as animações tanto dos inimigos quanto do protagonista. Em seguida, trabalhamos nas mecânicas principais, como movimentação e colisão com paredes, inimigos e moedas. Por fim, desenvolvemos a inteligência artificial dos inimigos.

3.1-Arquitetura RISC-V e Montador RARS



A arquitetura RISC-V é um conjunto de instruções (ISA) aberto e livre, permitindo que qualquer pessoa a utilize e modifique sem restrições. Projetada para ser simples e modular, adota o conceito de "Reduced Instruction Set Computer", tornando a execução de instruções mais eficiente. Essa flexibilidade a torna popular em pesquisa, educação e desenvolvimento de sistemas embarcados, favorecendo

Figura 2: Logo do risc V inovações e personalizações. Neste projeto, utilizamos o montador RARS (RISC-V Assembly and Runtime Simulator), uma ferramenta desenvolvida para criar e simular programas em linguagem de montagem RISC-V. O RARS oferece uma interface gráfica intuitiva, facilitando a escrita, montagem e execução de código de maneira acessível.

4-Resultados Obtidos:

Dificuldade 4.1-Implementação dos controles

Uma das primeiras dificuldades foi a implementação dos controles. Além da movimentação, precisávamos de um botão para facilitar a transição entre as fases, o famoso “cheat” (Figuras 3, 4 e 7). Na movimentação padrão, o protagonista deveria continuar se movendo, permitindo que o botão fosse pressionado apenas uma vez para que ele seguisse seu caminho até colidir com um objeto. Além disso, era necessário que ele se movesse enquanto apagava o rastro que deixava para trás.

Dificuldade 4.2-Renderização de imagens

A renderização das imagens foi um dos primeiros desafios enfrentados pelo grupo, já que era a nossa primeira experiência com o RISC-V e o RARS. Utilizando a ferramenta dentro do simulador chamada Bitmap Display, que apresentou suas próprias complexidades e exigiu um aprendizado significativo.

Dificuldade 4.3-IA dos inimigos

A inteligência artificial dos inimigos representou um desafio significativo, pois era preciso considerar como cada objeto e colisão interagem no mapa. Cada inimigo precisava adotar estratégias distintas para alcançar o protagonista e ter comportamentos variados. Se o protagonista utilizasse seu poder ao coletar a maior moeda, os inimigos deveriam fugir de maneiras diferentes. Caso contrário, a ideia de todos se reunirem perderia o sentido de desafio.

Dificuldade 4.4-Pontuação e Pontuação máxima

Na questão da pontuação, não tivemos grandes desafios com a pontuação normal, que exigiu poucos ajustes. No entanto, a pontuação máxima se tornou um ponto complicado, pois precisávamos de uma forma de armazenar essa informação mesmo ao fechar o jogo. Devido à nossa falta de familiaridade com a linguagem e a plataforma, não conseguimos implementar essa funcionalidade, que envolvia também o uso de arquivos.

Dificuldade 4.5-Escassez de informação

Um dos principais desafios do projeto foi a escassez de informações sobre a linguagem e o montador. Apesar de diversas pesquisas, muitas vezes não encontramos os resultados necessários, o que se tornou um grande obstáculo durante o desenvolvimento. Além disso, enfrentamos problemas com o alcance das instruções de branch, que exigiram uma refatoração significativa do código.

Conclusão:

Tendo em vista tudo que foi apresentado neste trabalho e as dificuldades enfrentadas, concluímos que, apesar dos obstáculos, conseguimos demonstrar nosso entendimento sobre a matéria e a linguagem. Embora algumas implementações tenham ficado de lado, conseguimos criar um jogo funcional e até divertido, com um desafio adequado para a maioria das pessoas. Com este projeto, aprendemos a utilizar o RARS e o RISC-V, compreendendo melhor os processos de criação.



Figura 3: Tela inicial do jogo Pac-Mario



Figura 4: Tela de vitória do jogo

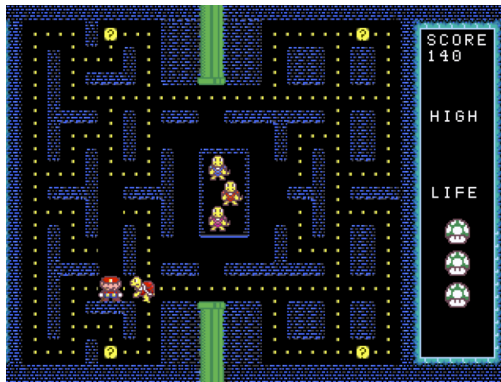


Figura 5: Fase 1 do jogo Pac-mario

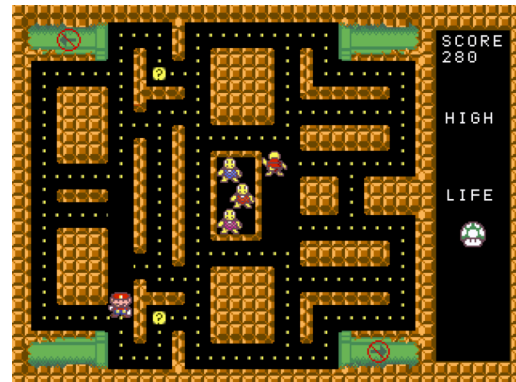


Figura 6: Fase 2 do jogo Pac-Mario

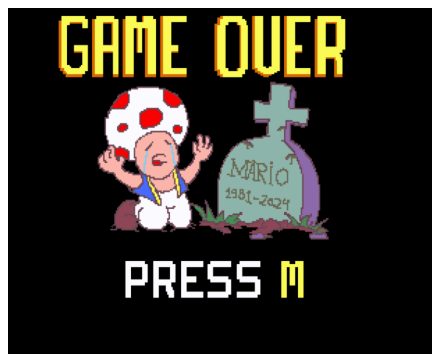


Figura 7: Tela de Game Over

Referências Bibliográficas:

<https://github.com/victorlisboa/LAMAR>
<https://supermario-game.com/pt>
<https://nirvacx.github.io/RarsHelp/>
<https://freepacman.org>
<https://github.com/TheThirdOne/rars>