SaintMan

Enzo Araújo Teles

Lucas Mendes

Victor Yan Martinez

Universidade de Brasília, Depto. Ciências da Computação, Brasil

Figura 1: Menu principal

Resumo

O presente artigo tem por objetivo detalhar o processo de desenvolvimento de "SaintMan" que é uma versão modificada do jogo PacMan (1980 - Atari) como parte de um projeto da disciplina de ISC – Introdução aos Sistemas Computacionais – da Universidade de Brasília ministrada pelo professor Marcos Vinícius Lamar. O desenvolvimento será feito em Assembly RISC-V 32 no simulador RARS e FPGRARS.

Palavras-chave: Assembly, RISC-V, RARS, FPGRARS, PacMan, jogo.

1 Introdução

Pac-Man, lançado em 1980 para o Atari, é um clássico jogo de arcade desenvolvido pela Namco. No jogo, o jogador controla Pac-Man, uma criatura amarela circular, em um labirinto cheio de bolinhas (conhecidas como "pellets"). O objetivo é comer todas as bolinhas do labirinto enquanto evita ser capturado por quatro fantasmas coloridos: Blinky, Pinky, Inky e Clyde. Pac-Man também pode encontrar quatro "power pellets", que o tornam temporariamente invulnerável, permitindo que ele coma os fantasmas, o que concede pontos extras. Cada fase termina quando todas as bolinhas do labirinto são comidas, e o jogo se repete com dificuldade crescente.

Figura 2: PacMan

Foi então atribuído aos alunos da disciplina de ISC – Introdução aos Sistemas Computacionais - do curso de Ciências da Computação da Universidade de Brasília a realização de um projeto de um jogo baseado no PacMan que tem por objetivo testar os seus conhecimentos em Assembly RISC-V 32 adquiridos em aula.

2 Metodologia

O projeto foi realizado no RARS na ISA RISC-V 32 e no simulador FPGRARS, utilizando programação em Assembly. Foram utilizadas as ferramentas disponibilizadas pelo RARS, como o Keyboard MMIO e o Bitmap Display. O FPGRARS serviu principalmente para executar o código e verificar potenciais erros na hora da execução. Para o desenvolvimento em conjunto da equipe, foi utilizado Git e GitHub para controle e versionamento do código.

2.1 Personagem

Na releitura do jogo, alteramos o PacMan para um humano que se chama Zenon habitante das moradas celestes, mas que perdeu todas as suas graças por uma força maligna e agora precisa recuperá-la para chegar nas alturas onde sua amada Nandora habita. Seu desenvolvimento foi feito com o software paint.net com as grids 16x16 e teve como inspiração cavaleiros medievais.

Figura 3: Zenon & SaintZenon

O arquivo do personagem é manuseado pegando-se o .data de cada direção do personagem e colocando num .data geral, e durante o código, apenas são realizadas operações de multiplicação que pulam certas linhas para a posição desejada quando precisássemos usar. Também armazenamos em outro .data os estados da animação do personagem, como eram três estados de animação, utilizamos os frames disponibilizados pelo RARS e fizemos a animação com base nisso, apenas trocando os frames entre si enquanto houvesse movimentação.

Figura 4: Animações de movimentação.

2.2 Mapas e Colisões

Alteramos também o estilo dos mapas para ser algo mais compatível com a história do jogo, assim seu desenvolvimento foi feito baseando-se nas crenças Cristãs de Céu, Purgatório e Inferno.

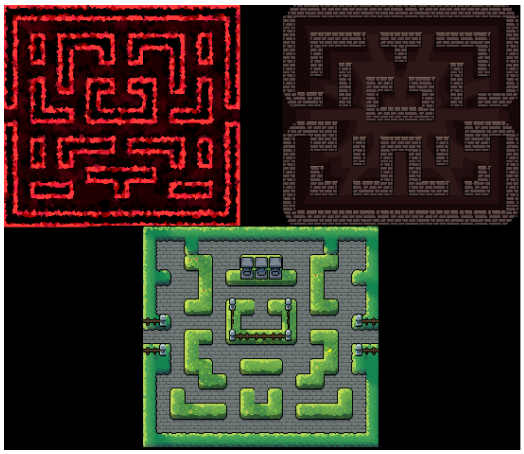

Figura 5: Mapas

Já as colisões foram feitas baseando-se num mapa específico, onde foram colorizados as paredes e os itens coletáveis para verificarmos o .data da cor e realizar a colisão com base nesta cor.

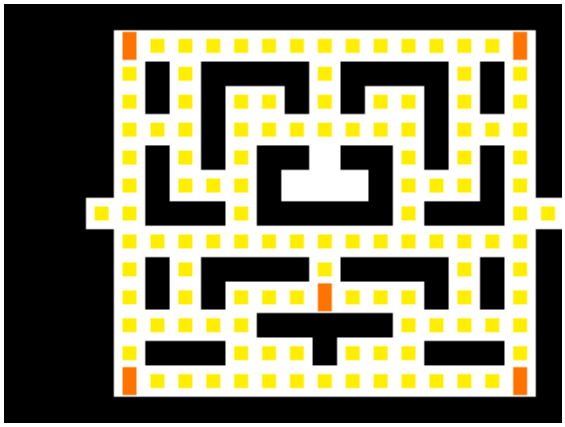

Figura 6: Mapa de colisões.

Como podemos ver, existem 4 tipos de cores que são tratadas de formas diferentes. O preto sinaliza a parede e o que o personagem nem ninguém pode atravessar, o branco representa o local que o personagem pode andar livremente, o laranja é a colisão com a Eucaristia e o amarelo é a colisão com os Rosários (Scores). Separamos a Eucaristia do Rosário, pois eles precisam ter colisões diferentes para que se possa fazer códigos diferentes para quando acontecer esta interação, visto que o personagem deve evidentemente ganhar superpoderes.

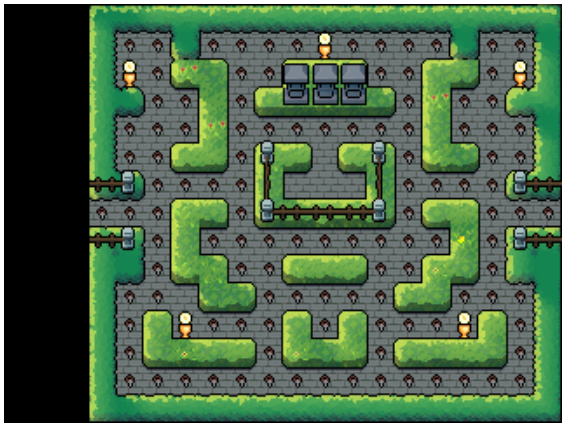

Figura 7: Mapa populado.

Figura 8: Eucaristia e Rosário.

Como dito anteriormente, o Rosário representando os Scores e a Eucaristia representando o que seria as “pallets” de superpoderes.

2.3 Inimigos

Para conformidade com a história fantasiosa do jogo, os fantasmas são agora diferentes tipos de demônios que buscam infernizar o caminho de Zenon até a sua santificação. Percebe-se que cada feição dos demônios é diferente, pois eles representam

cada um uma adversidade demoníaca diferente (Orgulho, Ira, Tristeza, Tédio).

Figura 9: Demônios.

As animações e a montagem do .data seguem o mesmo princípio da do personagem. Ambos têm 3 frames de animação e o .data está num arquivo único.

Figura 10: Animação do inimigo.

E assim como no jogo PacMan quando o jogador come alguma pallet e os fantasminhas mudam ficando vulneráveis a ataques, aqui é o mesmo.

Figura 11: Demônio assustado.

Figura 12: Animação de ataque do personagem.

A colisão com o inimigo para a mecânica de ataque, não funciona da mesma maneira que a colisão com os pontos. A colisão é calculada de acordo com as posições do jogador e do inimigo a cada iteração do game loop, checando para verificar se as caixas de colisão de cada sprite se sobrepõe.

Caso o jogador não tenha colidido com alguma eucaristia e mesmo assim colida com o inimigo, morrerá.

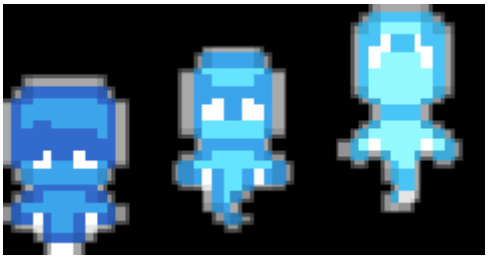

Figura 13: Animação de morte do personagem.

A lógica de movimentação dos inimigos é baseada na posição do jogador, cada inimigo seguindo um ponto diferente. A cada iteração do game loop o inimigo verifica cada uma das direções para verificar se há colisão com a parede, se não houver, ele andará para a direção cujo caminho o leve para o mais próximo do jogador. Quando o jogador come uma eucaristia, os inimigos vão para a direção oposto da que o aproxima, fugindo dele.

2.4 Música e Efeitos Sonoros

A música foi um dos problemas no início, pois precisaríamos arrumar uma forma de tocar a música enquanto tocávamos os efeitos sonoros, além do código que não é tão fácil de implementar. Com ajuda de alguns vídeos no YouTube de como se utiliza o MIDI do RARS, conseguimos implementar diferentes tipos de músicas e os efeitos sonoros respectivos. Um outro problema foi achar uma boa música para implementar, tentamos algumas do famoso jogo Elden Ring e Dark Souls, mas muitas delas ficaram um pouco estranhas no código pois elas possuem notas ocultas que o script de conversão não consegue detectar.

Além disso, criamos um script auxiliar para reduzir 1/4 do tempo da nota da música pois havia um pequeno delay entre o programa tocar uma nota e outra. Com isso, após uma pesquisa mais direta, conseguimos implementar efeitos sonoros para cada uma

das ações dentro do jogo e diversas trilhas sonoras para separar cada momento dele, além de criar um efeito sonoro próprio da mecânica de ataque.

Figura 14: Script Auxiliar

2.5 Game Loop

A cada iteração do game loop, primeiramente o programa muda a posição do player e dos inimigos na memória, em seguida ele imprime essas mudanças na tela, apagando os sprites anteriores e imprimindo na nova posição. Após isso, ele checa possíveis colisões com os inimigos, ao mesmo tempo que, altera a direção deles, de acordo com a lógica de movimentação dita anteriormente. Ao final, ele verifica se o usuário apertou alguma tecla e muda a direção do sprite de acordo com ela.

3 Resultados Obtidos

Após um longo período de desenvolvimento com dificuldades e limitações específicas da linguagem, conseguimos terminar o trabalho tendo o resultado esperado.

Assembly não é uma linguagem fácil de se trabalhar, por ser de baixíssimo nível, há limitações tanto da parte de depurações quando da construção de código, visto que há de se ter um controle maior com os registradores da linguagem.

4 Conclusão

Neste artigo foi detalhado o processo de desenvolvimento do jogo e da arte de SaintMan baseado no PacMan Atari na linguagem de programação Assembly RISC-V 32. Assim cumprindo os requisitos do projeto da disciplina de ISC (Introdução aos Sistema Computacionais).

5 Referências

- Projeto Destaque Gaunlet

https://github.com/Luke0133/The-Assembly-Gauntlet

- Projeto Destaque RadIceCream

https://github.com/NirvaCx/RadIceCream

- Repositório Auxiliar ISC e OAC

https://github.com/victorlisboa/LAMAR

- Instrumentos MIDI

https://www.youtube.com/watch?v=IYdq06l8qXI

- Vídeo explicativo IA dos Fantasmas

https://youtu.be/ataGotQ7ir8?si=Eos-rlo_eOuw-fUm

- Site para as músicas

https://www.hooktheory.com/

- Jogo PacMan https://freepacman.org/