

# BomberPac

Artur C. Rogério, João Pedro Barros S. Azevedo, Thiago Matheus M. Gomes

Departamento de Ciência da Computação – Universidade de Brasília (UnB)  
Brasília – DF – Brasil

**Resumo.** *Este trabalho objetiva uma releitura do jogo Bomberman (NES) desenvolvida em linguagem Assembly para ser executada nos simuladores RISC-V Assembler and Runtime Simulator (RARS) e ou Fast Pretty Good RISC-V Assembly Rendering System (FPGRARS) que emulam o processador RISC-V 32.*

**Palavras-chave—** *Assembly, RISC-V, Bomberman*

## 1. Introdução



**Figura 1. Recorte da Tela Inicial do BomberPac**

Bomberman(NES) foi um jogo lançado pela empresa Hudson Soft. O jogo foi posteriormente adquirido pela Konami que desenvolveu a popular série de jogos Bomberman. Trata-se de um jogo *Top-Down* em que o protagonista munido de bombas deve escapar um campo fechado explodindo blocos para revelar a saída secreta, além de eliminar todos os inimigos presentes para a destravar.

A fim de atender ao cronograma de produção do jogo, fez-se necessário simplificar aspectos do jogo original – movimentação contínua do personagem, *pathing* dos inimigos, presença da saída secreta. Optou-se por tematizar o projeto como um confronto entre os universos do Bomberman e da igualmente famosa franquia de jogos Pacman; desenvolvida pela empresa Bandai Namco na década de 80.

## 2. Metodologia

### 2.1 Colisões



**Figure 2. Comparativo Tilemap com Frame Renderizado para a Fase 1.**

O bitmap Display foi agrupado em grupos de 16x16 pixels (256 pixels) chamados de "matriz" cada um com somente um valor entre 0 e 9. Esse agrupamento gerou um conjunto de 20x15 matrizes (300 matrizes) que, por sua vez, foi usado para implementar as colisões e renderização de sprites das fases do jogo.

Foram convencionadas as correspondências visuais: 0 – *Campo Vazio*; 1 – *Pilastra (Bloco Indestrutível)*; 2 – *Fundo Preto*; 3 – *Tijolo (Bloco Destrutível)*; 5 – *PowerUp Chute*, 6 – *PowerUp Soco*; 7 – *Bomba*; 8 – *Campo com Ponto*.

### 2.2. Personagem



**Figura 3. Sprites do Personagem Principal e da Bomba da esquerda para direita**

O personagem foi implementado a partir da constante memória de sua posição, sendo renderizado sobre o mapa. Suas colisões com o mapa são orientadas pelo Tilemap, mas suas colisões com Inimigos são calculadas pela sobreposição de suas localizações.

### 2.3. Comandos

Definiu-se os seguintes comandos para o jogo: 'w' - *Movimenta Personagem para cima*; 'a' - *Movimenta Personagem para esquerda*; 's' - *Movimenta Personagem para baixo*; 'd' - *Movimenta Personagem para direita*; 'j' - *Deposita Bomba*; 'x' - *Destrói Tijolo (Permitido somente durante evento Força)*; 'K' - *Mata todos os inimigos*; 'L' - *Destrói todos os Tijolos da fase atual*; 'P' - *Passa para a próxima fase*.

Toda as letras são *case-sensitive* e as telcas 'K', 'L', e 'M' foram implementadas como *cheats* para a apresentação do jogo.

## 2.4. Power Ups

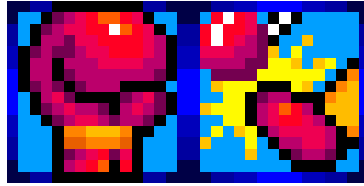


Figura 4. PowerUps (Soco à esquerda e Chute à direita)

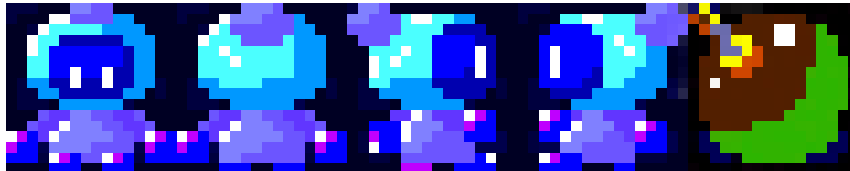


Figura 5. Visuais do PowerUp Força e Chute da esquerda para a direita

**Força:** Dura 6 segundos após coletado. Permite ao personagem a habilidade de destruir o Tijolo a sua frente com um "soco" ao pressionar a tecla 'x'. O Personagem recebe um novo visual pela mesma duração do evento Força.

**Chute:** Dura 8 segundos após coletado. Permite ao personagem a habilidade de “chutar” uma Bomba ao entrar em contato com ela no sentido que está olhando. O temporizador de explosão da Bomba fica estático durante o deslocamento, permitindo que a Bomba exploda somente 2 segundos após colidir com uma Pilastra ou Tijolo. A Bomba recebe um novo visual enquanto durar o evento Chute.

## 2.5. Inimigos

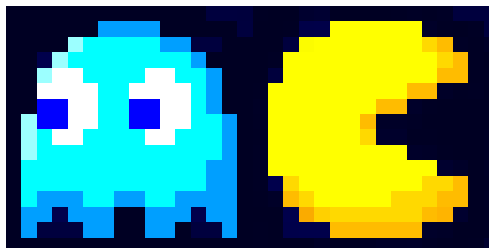


Figure 6. Inimigos (Fantasma à esquerda e Pacman à direita)

Atuam como obstáculos ao Personagem. Dividem-se em duas categorias:

**Fantasma:** Está presente somente na Fase 1. Possui movimentação unidirecional, com alternância de sentidos. Fantasmas que se movimentam horizontalmente alternam entre direita e esquerda, enquanto os que se movem verticalmente alternam entre cima e baixo.

**Pacman:** Está presente somente na Fase 2. Movimenta-se conforme um caminho cíclico predeterminado, alterando seu sentido de percorrimento ao colidir com uma Pilastra (Tilemap 1), Tijolo (Tilemap 3) ou Bomba (Tilemap 7).

## 2.6. Gameplay



**Figura 7. Da esquerda para direita, Tela de Vitória e Tela de Derrota.**

O objetivo do jogo é passar de todas as fases e derrotar o Pacman e seus Fantasmas. Há dois modos para avançar uma fase – Coletar todos os pontos do estágio (Modo Pacifista) ou eliminar todos os inimigos (Modo Agressivo).

O jogador é recompensado com a Tela de Vitória ao concluir as duas fases. O jogador é penalizado com a Tela de Derrota ao perder todas as três vidas iniciais ou ao não concluir a fase no tempo estipulado no Heads-Up Display (HUD).

## 3. Resultados Obtidos

Por fim, um jogo funcional foi construído. Provido de trilha e efeito sonoros; de animações visuais (Troca de sprites dos inimigos e do personagem); de telas de transição (Tela inicial, Tela de instruções, Tela de Vitória e Tela de Derrota); de movimentação do Personagem e dos Inimigos; além da contabilização e retorno visual de elementos do HUD (Vidas, Pontuação e Temporizador).

## 4. Conclusão

A princípio, a criação do projeto apresentado aparentou ser impossível em virtude do prazo para sua conclusão, das aparentes limitações da linguagem Assembly e da falta de proficiência nas ferramentas exigidas. Tratou-se de um projeto de progressão lenta, mas desafiante e, ao final, recompensador visto que o aprendizado adquirido na matéria de Introdução aos Sistemas Computacionais foi exercido na prática.

## 5. Referências Bibliográficas

Petrus, R. and Lisboa, F., Hugo, V. (2025) "Projeto Learning Assembly for Machine Architecture and RISC-V (L.A.M.A.R)". Disponível em: <https://github.com/victorlisboa/LAMAR>

Emulador do Bomberman (NES). Disponível em: [https://www.retrogames.cz/play\\_085-NES.php](https://www.retrogames.cz/play_085-NES.php)

Riether, L. "Fast Pretty Good RISC-V Assembly Rendering System". Disponível em: <https://github.com/LeoRiether/FPGRARS>

Aulas de Laboratório 1 ministradas pelo Dr. Marcus Vinícius Lamar.