

# Documentação do Jogo: Fix-It Felix Jr. em Assembly RISC-V

## 1. Título e Introdução

### 1.1. Nome do Jogo

#### **Fix-It Felix Jr.**

### 1.2. Descrição Geral

O jogo é inspirado no clássico **Fix-It Felix Jr.**. Desenvolvido em **Assembly RISC-V**, ele é projetado para rodar na plataforma **Rars16\_Custom1** ou **FPGRars**. O jogador assume o papel de **Felix**, um carpinteiro que deve consertar janelas quebradas em um prédio enquanto desvia de obstáculos jogados por **Ralph**, o vilão do jogo.

O jogo se passa em uma tela vertical com um prédio dividido em andares. Ralph move-se de um lado para o outro. O objetivo é consertar todas as janelas de um nível para progredir para o próximo, acumulando pontos enquanto evita ser atingido por tijolos que são arremessados por Ralph ao longo do jogo.

### 1.3. Plataforma

- **Simulador:** Rars ou FPGRars para RISC-V.
- **Hardware:** O jogo é projetado para rodar em simuladores de arquitetura RISC-V e pode ser adaptado para execução em FPGA.

### 1.4. Tecnologias Utilizadas

- **Linguagem de Programação:** Assembly RISC-V.
- **Simuladores:** Rars ou FPGRars.
- **Recursos Gráficos:** Interfaces gráficas e animações para representar o prédio, os personagens e os obstáculos.
- **Efeitos Sonoros:** Sons MIDI, reproduzidos conforme as ações do jogo, como consertos de janelas.

## 2. Gameplay e Mecânicas

### 2.1. Objetivo do Jogo

O objetivo principal do jogador é **consertar todas as janelas quebradas** do prédio para avançar para o próximo nível. Durante o jogo, é necessário **desviar dos tijolos** jogados por Ralph.

### 2.2. Regras do Jogo

- **Pontuação:** O jogador ganha pontos ao consertar janelas.
- **Perda de Vidas:** Se o jogador for atingido por tijolos ou pássaros, ele perde uma vida.
- **Game Over:** O jogo termina quando todas as vidas do jogador acabam.

### 2.3. Mecânicas de Jogo

- **Movimentação do Personagem:**
  - Felix pode se mover **para cima, para baixo, para a esquerda e para a direita** dentro dos limites da tela.
  - As bordas do prédio delimitam a área de movimentação, impedindo que o personagem saia da tela.
- **Conserto de Janelas:**
  - Ao chegar na posição de uma janela quebrada, o jogador deve pressionar um botão específico para consertá-la.
  - Cada janela consertada adiciona pontos ao placar do jogador.
- **Obstáculos:**
  - **Tijolos:** Jogados por Ralph em padrões previsíveis, caindo verticalmente.

### 2.4. Controles

- **W** - Move para cima
- **A** - Move para a esquerda
- **S** - Move para baixo
- **D** - Move para a direita
- **M** - Conserta a janela

### 2.5. Fluxo do Jogo

1. **Tela inicial:** O jogador deve pressionar uma tecla para iniciar o game.
2. **Gameplay:** O jogador controla Felix para consertar janelas enquanto desvia de obstáculos.
3. **Fim de Jogo:** Ao perder todas as vidas, é exibida uma tela de **Game Over** com a pontuação final.

## 3. Design de Interface e Menus

### 3.1. Interface do Jogo

- **Topo da Tela:**
  - Exibe o **número de vidas** restantes.
- **Centro da Tela:**
  - Mostra o **prédio** com janelas quebradas e os **obstáculos** em movimento.
- **Base da Tela:**
  - Felix se move para consertar as janelas quebradas.

### 3.2. Feedback Visual

- **Animações:**
  - Movimentação do Felix com sprites que representam as ações de andar e consertar janelas.

- Animações para os tijolos jogados por Ralph e movimentação do mesmo.
- **Mudança de Cor:**
  - As janelas quebradas mudam de cor ao serem consertadas para indicar progresso.

## 4. Sistema de Pontuação

### 4.1. Como Pontuar

- **Janelas Consertadas:** Cada janela consertada adiciona pontos à pontuação total do jogador.

## 5. Arquitetura do Sistema e Estrutura de Arquivos

### 5.1. Organização do Código

/Fix-it-Felix-Jr

```
|—— exemplos úteis    // Para auxiliar a produção do projeto
|—— Imagens            // Arquivos.data para os sprites
|—— fpgrars-x86_64-pc-windows-msvc-unb
|—— main.asm           // Arquivo principal do jogo
|—— Rars16_Custom1     // Simulador de programas em Assembly RISC-V
```

### 5.2. Main

- **main.asm:** Inicialização do jogo e controle do loop principal, renderização do prédio, janelas, obstáculos e Felix, leitura de teclado para movimentação e ações do jogador, lógica de colisão e movimento dos obstáculos e execução de sons MIDI durante o jogo.

## 6. Instalação e Execução

1. Abra o simulador **Rars**.
2. Carregue o arquivo `main.asm`.
3. Configure a memória e a tela conforme as especificações do simulador.
4. Compile e execute pressionando **F5**.

## 7. Testes e Bugs Conhecidos

- **Bugs Conhecidos:**
- Durante o desenvolvimento do jogo, nos deparamos com diversos bugs, alguns simples de corrigir, outros que exigiram mais tempo e dedicação.

Um dos primeiros problemas ocorreu na renderização do personagem principal, Felix Jr., que ficava piscando na tela. Conseguimos solucionar esse erro ajustando a lógica de atualização do sprite.

Outro bug que nos deu bastante trabalho foi relacionado ao som, especificamente à trilha sonora de fundo. O jogo travava sempre que a música era iniciada, só retomando a execução

após a conclusão completa do áudio. Esse problema exigiu uma investigação mais aprofundada até encontrarmos a solução adequada.

Também enfrentamos um erro na lógica de movimentação do Ralph, que estava se movendo para trás de maneira inesperada, como se estivesse fazendo um "Moonwalk".

Além disso, ainda há um bug esporádico que faz o jogo travar sem uma causa aparente, o que seguimos investigando.

## 8. Créditos

- **Autores:** Isabella Lemos Alvarenga, Márcio Vinícius da Silva Guimarães e Danilo Pereira Santos.
- **Inspirações e Referências:**
  - Jogo **Fix-It Felix Jr.**
  - Documentação do simulador **Rars**.
  - Jogos modelos feitos em assembly por outros estudantes da UnB.