

The Way of the Exploding Fist

Pedro Henrique dos Santos Vitor Manuel
Marcelo Amorim

University of Brasília, Dept. of Computer Science, Brazil

Abstract

Esse projeto teve como objetivo desenvolver um jogo a partir dos estudos na matéria de Introdução aos Sistemas Computacionais. O conjunto de instruções (ISA - Instruction Set Architecture) utilizados no desenvolvimento é o RISC-V. Segundo Pereira (2017), "a iniciativa RISC-V foi criada com o objetivo de desenvolver um processador de alto desempenho e código aberto, que possa ser utilizado no desenvolvimento de aplicações e produtos de uso geral. RISC-V é uma ISA (Instruction Set Architecture ou arquitetura de conjunto de instruções) desenvolvida originalmente na Universidade da Califórnia". Neste artigo ficará descrito toda a metodologia de desenvolvimento do jogo, desde a definição das macros, loops que fazem a dinâmica do jogo, renderização das sprites etc. Assim, seguiremos as seguintes etapas neste artigo: introdução, metodologia, resultados, discussão e referências.

1 Introdução

Jogos digitais tem sido amplamente usados para fins de aprendizagem.

Este trabalho apresenta um exemplo prático de como o desenvolvimento de jogos pode ser interessante para aprendizagem de conceitos fundamentais de computação. No desenvolvimento do jogo foi possível estudar conceitos importantes de programação e os fundamentos desses conceitos.



Figure 1: O que é a figura

A Figura 1 mostra o jogo The Way of the Exploding Fist [Altera Corporation].

Sobre o jogo

Na seção 2 será apresentada a metodologia utilizada. A seção 3 apresenta os resultados obtidos. A seção 4 conclui este trabalho.

2 Metodologia

[Apresentar aqui como o projeto foi desenvolvido e quais as ferramentas usadas]

1) A divisão de arquivos no desenvolvimento ficou da seguinte maneira:

- game.s O arquivo game.s é onde está o centro da aplicação. Nele foi colocado as dependências explicadas a seguir (animations-player1.s, animations-player2.s, GAME-MACRO.s, sprites.s). O

programa inicia com o loop do menu e, posteriormente, o loop do jogo (GAMELOOP). No GAMELOOP fica definido todas as teclas de controle do player 1.

- animations-player1.s + animations-player2.s Aqui ficaram todas as macros ligadas aos movimentos do Player 1 e do Player 2 em seus respectivos arquivos.

- GAME-MACROS.s As game macros foram macros relacionadas ao jogo de forma geral e ficaram disponíveis para uso não só relacionada a um player.

- sprites.s Nesse arquivo ficaram todos os 'includes' relacionados às imagens em formato '.data'.

2) Utilizamos o github para versionar o código e otimizar o trabalho em grupo:

- Repositório do trabalho: <https://github.com/MarceloAmorim25/RISC-V-Assembly-game>

2.1 Arquitetura RISC-V

Exemplo de subseção. A arquitetura RISC-V [Patterson and Hennessy 2005] surgiu como uma forma de alternativa à dependência de padrões definidos apenas por empresas que tivessem com maior força no mercado de desenvolvimento de processadores. Um conjunto de padrões de instruções aberto veio como uma alternativa de se manter esses padrões de forma independente. Segundo o The RISC-V Instruction Set Manual Volume I: Base User-Level ISA Version 1.0, um dos objetivos da criação desse conjunto de instruções é "Provide a realistic but open ISA that captures important details of commercial general-purpose ISA designs and that is suitable for direct hardware implementation.". Esse documento de marco inicial é aberto e pode ser consultado em <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-62.pdf>.

2.2 Simulador/Montador RARS

Exemplo de subseção. O Rars RISC-V Assembler, Simulator, and Runtime [Vollmar and Sanderson 2005], segundo o seu repositório no github: "RARS, the RISC-V Assembler, Simulator, and Runtime, will assemble and simulate the execution of RISC-V assembly language programs. Its primary goal is to be an effective development environment for people getting started with RISC-V.". Esse repositório pode ser acessado no link - <https://github.com/TheThirdOne/rars>.

3 Resultados Obtidos

Apresentar aqui os resultados obtidos, telas e link para vídeos e comentários.

4 Conclusão

Este trabalho apresentou...

Referências:

<https://www.embarcados.com.br/fe310g-microcontrolador-open-source-estrutura-basica-risc-v/>

References

ALTERA CORPORATION. DE2-70 development board. <http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html> [Accessed in 10/05/2012].

PATTERSON, D. A., AND HENNESSY, J. L. 2005. *Organização e projeto de Computadores: A Interface software/hardware*, 3ª edição. Elsevier.

VOLLMAR, K., AND SANDERSON, P. 2005. A MIPS assembly language simulator designed for education. *Journal of Computing Sciences in Colleges* 21, 1, 95–101.