



Projeto Aplicativo - Kirby's Adventure -



Objetivo:

O projeto final de OAC (vulgo trabalho) visa testar e aferir os conhecimentos dos alunos sobre as duas grandes áreas abordadas no curso, a linguagem Assembly RISC-V e a organização de processadores, por meio da criação de um jogo temático por parte dos estudantes. O trabalho deve ser feito pelo mesmo grupo dos laboratórios. O jogo deve ser escrito em Assembly RISC-V (ISA RV32IMF) e apresentado para a turma e banca avaliadora no LINF, durante o horário da aula, através da FPGA e acessórios (monitor, teclado, caixa de som etc.). O projeto que porventura não for implementado na FPGA atingirá no máximo 70% da nota máxima possível desta atividade. Neste semestre, decidiu-se que o tema do jogo será *Kirby's Adventure* (NES). Aqueles que não estão familiarizados com esse jogo podem assistir a uma *gameplay* clicando [aqui](https://youtu.be/rJXM4EPbPe0?si=lot6oAZBA7p-Y-7Q) ou copiando e colando o link abaixo no *browser*.

Gameplay (2h18): <https://youtu.be/rJXM4EPbPe0?si=lot6oAZBA7p-Y-7Q>

Para garantir que os trabalhos sigam o tema, assim como objetividade na correção, estipulou-se a seguinte lista de requerimentos. O vídeo acima apresenta ilustrações da maioria dos requerimentos dentro dos primeiros 5 minutos.

Requerimentos:

- 1) (0,5) Música e efeitos sonoros.
- 2) (0,5) Ataques do jogador.
- 3) (1,0) Movimentação e animação do(s) personagem(s) jogáveis.
- 4) (1,5) Mínimo de 2 habilidades distintas de cópia do Kirby, permitindo que ele utilize a habilidade de um inimigo após aspirá-lo, impactando no combate e/ou na movimentação.
- 5) (0,5) Informações sobre a vida, habilidade e pontuação do Kirby.
- 6) (1,0) Pelo menos 2 áreas distintas, isto é, dois ambientes separados por uma porta.
- 7) (1,5) Mínimo de 3 tipos de inimigos diferentes com IA (número de inimigos em aberto), sendo um deles um chefe.

- 8) (1,5) Background móvel que acompanhe o movimento do Kirby (horizontal **ou** vertical).
- 9) (2,0) documentação: descreva seu projeto no formato de um Artigo Científico IEEE para o SBGames (modelo no Moodle), com 6 páginas, contendo:
- | | |
|--|---|
| i. Título; | vi. Fundamentação Teórica e Trabalhos Relacionados; |
| ii. Autores, Filiação Acadêmica e Contato; | vii. Metodologia; |
| iii. Resumo; | viii. Resultados Obtidos; |
| iv. Palavras-chave; | ix. Conclusões e Trabalhos futuros; |
| v. Introdução; | x. Referências Bibliográficas. |

Prepare arquivo .zip com o artigo do projeto e com todos os códigos e relatórios dos laboratórios, projeto e seus códigos fontes e faça o upload no Aprender3.

A ocorrência de requerimentos relativamente vagos é proposital: serve para dar margem de folga aos programas dos alunos. Dúvidas/sugestões de interpretação devem ser negociadas com os monitores, preferencialmente, via Discord, seja publicamente ou em privado.

Sugestões e Comentários finais

Embora seja permitido, não é aconselhado a criação de *sprites* do zero; usem, por exemplo, o Paint.net e o executável bmp2oac3.exe para transformar as imagens de sprites já existentes em .data's do RARS. Finalmente, recomenda-se obter músicas para o jogo a partir da conversão de arquivos .mid já existentes.

Sprites: <https://www.sprisers-resource.com/nes/kirbyadv/>

Músicas (algumas): <https://www.khinsider.com/midi/nes/kirby-s-adventure>

Para a conversão das músicas, sugere-se usar o programa em Python desenvolvido por Gabriel B. Gomes e Davi Paturi ou seguir o tutorial de como importar músicas do [Hooktheory](#). É FORTEMENTE recomendado assistir ao tutorial do Davi Paturi no Youtube, [RISC-V RARS - Renderização dinâmica no Bitmap Display](#).

Conversor de .mid: https://github.com/Zen-o/Tradutor_MIDI-RISC-V

Conversor de .midi, separando as vozes: <https://github.com/Luke0133/Midi2RiscV>

Tutorial do Hooktheory: <https://youtu.be/mBmOkyqejHU?si=r6XRZLv1Rpfo2zJ8>

Tutorial: https://youtu.be/2BBPNgLP6_s?si=eE6QsBbDuH9bE7nW

Além disso, no repositório do GitHub presente [aqui](#), há diversos materiais e jogos antigos úteis para inspirar e ajudar no desenvolvimento desse trabalho.

Repositório LAMAR: <https://github.com/victorlisboa/LAMAR>

Para agilizar a programação, deem preferência ao FPGRARS e deixem o RARS apenas para debug. Não se sintam amedrontados com a magnitude do trabalho: o segredo é fazê-lo de pouco em pouco e de dia em dia, começando pelos requerimentos mais atingíveis. Com isso dito, boa sorte e bom trabalho a todos.

Bom divertimento!