



1 Introdução, Objetivos e Requerimentos

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 por ventura 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 **Pokémon FireRed/LeafGreen**. Aqueles que não estão familiarizados com esse jogo podem assistir a uma gameplay clicando [aqui](https://www.youtube.com/watch?v=2l4FGPkDwss) ou copiando e colando o link abaixo no *browser*.

Gameplay (2h40): <https://www.youtube.com/watch?v=2l4FGPkDwss>

Para garantir que os trabalhos sigam o tema, assim como objetividade na correção, estipulou-se a seguinte lista de requerimentos.

Requerimentos:

- 1) (0,5) história do jogo (com caixas de diálogo);
- 2) (1,0) música e efeitos sonoros;
- 3) (0,5) mínimo de 1 item utilizável;
- 4) (1,0) mínimo de 5 tipos de pokémons diferentes, com um sistema “pedra-papel-tesoura” entre eles (normal perde para luta, luta para psíquico, etc.);
- 5) (1,0) mínimo de três tipos de telas jogáveis: seleção de pokémons iniciais, área aberta, e ginásio;
- 6) (1,0) fases com número crescente de inimigos (inclusa a animação deles), espaços abertos e paredes;
- 7) (1,0) IA que controla os inimigos e sistema de turnos da luta;
- 8) (0,5) cenas de batalhas, podendo ser apenas uma imagem mostrando os personagens e as vidas deles;
- 9) (0,5) menu de ações do jogador;
- 10) (0,5) movimentação dos personagens;
- 11) (0,5) três tipos de terrenos especiais;
- 12) (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. |

Lembre-se que um artigo científico deve sempre se basear em uma proposta inovadora, logo se deve deixar claro no texto qual a sua contribuição científica. Nomeie o .pdf da documentação `GrupoX_Trabalho.pdf`.

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.

Figura 1: Ilustração dos requerimentos



(a) cena de luta com menu



(b) escolha do pokemon inicial com caixa de diálogo



(c) área aberta com terreno especial e uso de item



(d) ginásio com obstáculos

2 Diretrizes da entrega

Prepare arquivo .zip, nomeado GrupoX_Trabalho.zip, com o artigo do projeto e com todos os códigos e relatórios dos laboratórios, projeto e seus códigos fontes.

Na eventual detecção de indícios de plágio, os autores dos trabalhos envolvidos serão questionados e a cola, caso confirmada, levará à anulação da nota de todos os trabalhos envolvidos. Para evitar que seu trabalho seja copiado sem consentimento, considere tornar o repositório Git do projeto privado até o dia da apresentação.

Finalmente, também será analisado, grupo a grupo, como cada integrante contribuiu para a criação do jogo, quantitativa e qualitativamente. Isso será feito por chamada do Teams e visualização de commits no Git do trabalho.

O estudante que deixou de contribuir, sem justificativa razoável, para a confecção do jogo, terá a nota do trabalho reduzida na proporção de sua ausência, conforme estipulado pelos monitores.

3 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.sprites-resource.com/game_boy_advance/pokemonfireredleafgreen/

Músicas: <https://www.khinsider.com/midi/gameboy/pokemon-red-blue-yellow->

Para a conversão das músicas, sugere-se usar o programa em Python desenvolvido por Gabriel B. Gomes e Davi Paturi. 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

Tutorial: https://www.youtube.com/watch?v=2BBPNgLP6_s&list=PLL0Kob75DU32afhLBN5nY2KzOJ5k6lw-Q&index=3&t=21s

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.