Adventures of Iolo

João Pedro Felicio Pereira

Gustavo Lopez Dezan

Universidade de Brasília Departamento de Ciência da Computação, Brasília-DF

Resumo: Esse projeto teve como objetivo desenvolver um game em **Assembly RISC-V**, interpretado pelo simulador RARS, utilizando de ferramentas como **Bitmap Display**, **Keyboard MMIO** e **interface MIDI**. Esse projeto tem como base o game *Adventures of lolo*.

1 Introdução

Adventures of lolo foi lançado em 1989 pela HAL laboratory, originalmente para o console Nintendo Entertainment System(NES) e posteriormente foi relançado na Virtual Console da Wii e posteriormente na Wii U e Nintendo 3DS.

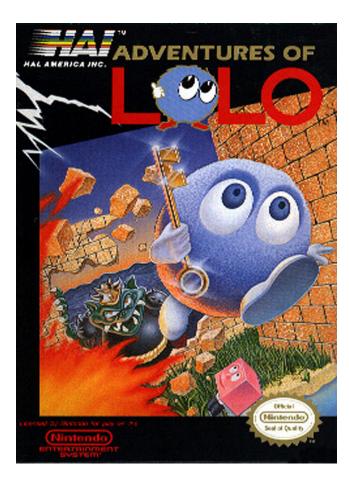


Figure 1: Capa do jogo para NES

2 Metodologia

Para criar esse jogo utilizamos o simulador da ISA do processador Assembly RISC-V, RARS, nele utilizamos ferramentas como Bitmap Display, Keyboard MMIO e interface MIDI para fazer uma integração do jogo com os dispositivos de saída e entrada.

2.1 Movimentação e interação com os elementos

Movimento e interação com os elementos No uso de macros, o grupo buscou produzi-las de forma modular, de modo que elas se encaixassem em diversas situações, o que dificultou. Os primeiros estágios da programação para facilitar os estágios mais avançados. Para a movimentação do personagem Lolo, foram criadas funções que o deslocavam de acordo com um plano cartesiano implementado também por código. Além disso, para checar por colisões o personagem encontra sua posição na matriz e busca pelos valores dos elementos correspondentes. Os códigos de cada ele-

mento do jogo foram pensados como objetos, com suas "funções" entrelaçadas para que fosse possível alcançar os resultados desejados, o que pode ser notado no momento em que o Lolo colide com um coração, onde ele detecta a posição do objeto colidido e chama a função de apagar um sprite na posição específica do coração. De modo semelhante, o poder disparado por Lolo checa inicialmente por sua própria existência, de modo que caso ela seja verdadeira, ele se propaga pelo mapa buscando por colisões e, no caso de ele acertar um "objeto" inimigo, ele chama o código que passará para o inimigo o estado de LIVE = FALSE. O inimigo, por sua vez, só poderá se redesenhar no mapa a cada loop do jogo caso ele possua seu estado de live como true.

2.2 Manipulação de imagens

Foi necessário utilizar programas externos para construir as imagens dos elementos do jogo e o mapa, como o paint.net e piskel. Após isso, utilizamos um programa para converter o arquivo de imagem para .data. Sempre reparando em detalhes como deixar os arquivos em dimensões, cujo o x e y são pares, pois como o RARS tabalha em byte adressing facilitaria o cálculo e o funcionamento mais dinâmico do jogo. Para a criação do mapa, foi criado um tileset com todos os objetos estáticos do jogo, e a partir disso o mapa foi montado como um conjunto de blocos de 16x16 pixels. Para as imagens que possuem mais de um estado, ou uma animação completa (que de modo geral são os sprites), utilizou-se uma imagem com todas as posições que o sprite poderia assumir ordenadas por suas direções, desse modo, foi criado um script que itera por essas posições encontrando a direção do sprite e o último frame que ele assumiu. Para objetos que o Lolo "consumiria", eles foram gerados como fazendo parte da imagem "mapa" e foram printados fechados por cima dela.

2.3 Trilha Sonora

Foi utilizada duas macros, uma para efeitos sonoros e outra para a trilha principal. A macro de *sound effects* recebe quatro parâmetros, todos são registradores de argumento, a1, a2, a3 e a4, correspondente a nota, duração,instrumento e volume,respectivamente. A macro de *play music* é um pouco mais complexa por ter várias notas e a duração para cada uma delas. Sendo assim, tem de ser calculado a quantidade de notas para servir como comparação com o contador para saber quando parar de tocar tal música. A **interface MIDI** conta, também, com as syscalls 31 e 33, **MidOut** e **MidOutSync** que são os serviços que vão nos auxiliar para tocar a trilha.

3 Resultados Obtidos

3.1 O jogo

O resultado obtido foi bem satisfatório, apesar do tempo ter sido um grande obstáculo, fizemos um game com efeitos sonoros, animação, tratamento de colisões e outras funções, e isso nos deixou bastante animados, pois nunca tinhamos feito um projeto em grupo dessa magnitude.

3.2 Problemas

Programar em Assembly não é algo muito atrativo. Durante o programa, o computador de um dos integrantes do grupo sofreu sérios problemas mecânicos por causa da execução de uma música. Além disso, problemas como "branch target" acabou nos tomando bastante tempo e não conseguimos implantar diversas funções no código por causa desse erro. Somando isso com as vezes que o simulador simplesmente parava de executar um programa que ele executava com facilidade anteriormente.

4 Conclusão

Esse projeto foi de imensa importância para a nossa apredizagem em assembly.No início, o maior desafio foi o desanimo, pois não estávamos muito preparados para esse jogo, justamente pela falta de conhecimentos na linguagem. Porém evoluímos bastante no decorrer do projeto.

References

https://en.wikipedia.org/wiki/Adventures_of_Lolohttps://www.nintendo.com/games/detail/adventures - of - lolo - 3ds/ :

MIDI number	Note name	Keyboard	Frequency Hz		riod ns	
21 22 24 25 28 27 29 30 33 34 35 37 38 39 41 42 446 447 49 50 51 53 54 65 66 66 67 68 68 71 72 73 74 75 77 78 18 82 88 88 87 89 99 99 99 99 99 99 99 99 99 99 99 99	######################################	Keyboard	Hz 27,500 29,12 32,703 30,868 34,64 41,203 38,85 44,203 43,654 46,299 55,000 51,91) 61,735 58,27 65,406 69,25 82,407 77,76 83,307 110,00 103,8 144,81 123,47 116,5 123,47 116,	5 36.40 8 22.91 9 20.41 3 18.18 9 16.20 1 15.29 1 10.20 9 11.45 9 10.20 1 8.09 7 .645 8 4.050 8 3.405 8 3.405 3 3.034 9 2.551 8 4.050 8 3.405 3 3.034 9 2.551 9 10.20 1 1.136 1 1.136 1 1.136 1 1.136 1 1.136 3 1.012 7 0.8513 5 0.7584 0 0.7589 0 0.6378 0 0.7759 0 0.6378 0 0.4778 0 0.7380 0 0.3792 0 0.3792 0 0.3792 0 0.3780		
98 00		J. Wolfe, UNSW	0040.0	0 0.3792 0.3580 0 0.3189 4 0.2841		

Figure 2: Código do RARS das notas musicais

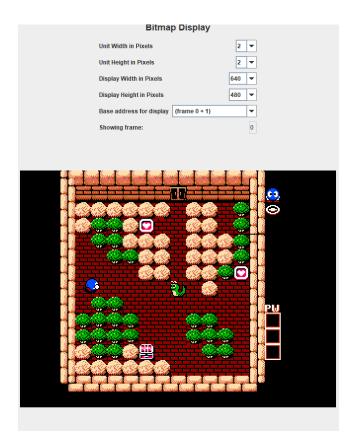


Figure 3: Jogo rodando no RARS



Figure 4: Criação dos sprites