

# Construção de Sistemas Digitais - Trabalho 3

## Introdução

O terceiro trabalho da disciplina consiste na implementação de um sistema de jogo prototipado em FPGA, utilizando a saída de vídeo VGA e a porta USB. Um teclado USB compatível com o padrão PS/2 (projeto desenvolvido no trabalho anterior) será utilizado como dispositivo de entrada para controle do jogo. O adaptador gráfico já encontra-se integrado à plataforma HF-RISC, sendo fornecido um driver para suas funções de baixo nível, uma biblioteca gráfica e exemplos.

São oferecidos diversos recursos a serem utilizados como referência para o desenvolvimento do trabalho (repositório do HF-RISC<sup>1</sup>), que consiste em uma aplicação (jogo) a ser implementada em linguagem C, compreendendo operações sobre o vídeo, movimento de objetos, detecção de colisão, sistema de pontuação e controle. Entre os recursos, está um projeto de hardware do video game que utiliza um processador, memórias, periféricos, um adaptador gráfico VGA e um kit para o desenvolvimento de software.

## Prototipação

A prototipação do projeto de referência pode ser realizada incluindo-se os seguintes arquivos em um projeto no Vivado:

- hf-risc/riscv/core\_rv32e\_verilog/\*
- hf-risc/riscv/platform/rams/boot\_ram\_50mhz.vhd
- hf-risc/riscv/platform/rams/ram.v
- hf-risc/devices/peripherals/basic\_soc.v
- hf-risc/devices/controllers/uart/uart.v
- hf-risc/devices/controllers/vga\_controller/\*
- hf-risc/riscv/platform/artix7\_nexysA7/artix7\_nexysa7\_basic\_soc\_vga.v (top level)
- hf-risc/riscv/platform/artix7\_nexysA7/artix7\_nexysa7\_basic\_soc\_vga.xdc

O projeto de referência inclui a maioria dos recursos de hardware necessários para o desenvolvimento. Cabe a cada grupo adaptar a integração

---

<sup>1</sup><https://github.com/sjohann81/hf-risc>

do periférico desenvolvido no trabalho anterior com a plataforma de hardware fornecida, assim como adaptar o software para realizar a leitura das teclas para controle do jogo.

## Aplicações de referência, drivers e adaptador gráfico

Juntamente com a plataforma de hardware, são fornecidos os seguintes recursos, presentes no diretório *hf-risc/software/app/vga*:

- Driver / GDI - (arquivos *vga\_drv.h*, *vga\_drv.c* e *IBM\_VGA\_8x8.h*);
- VGA Demo (arquivo *vga\_demo.c*);
- VGA Cube (arquivo *vga\_cube.c*);
- VGA Ball (arquivo *vga\_ball.c*);
- VGA Sprite (arquivo *vga\_sprite.c*);

O primeiro exemplo (VGA Demo) ilustra o uso de todas as funcionalidades do driver de vídeo e GDI (*Graphics Display Interface*), que inclui uma API para acesso às funções gráficas. O segundo exemplo ilustra o uso de processamento em 3D utilizando cálculos por software e o terceiro exemplo mostra um exemplo de jogo, contendo uma bola, objetos, detecção de colisão e controles dos botões. O último exemplo apresenta uma forma de construir um jogo utilizando *sprites*. Os exemplos podem ser compilados com o comando *make nome\_exemplo F\_CLK=50000000*, e carregados para a placa FPGA com o comando *make load SERIAL\_DEV=/dev/ttyUSB1*. Antes de carregar a aplicação, recomenda-se que em um terminal seja utilizado para depuração, sendo que sua aplicação poderá utilizar comandos de impressão. Utilize o comando *make debug SERIAL\_DEV=/dev/ttyUSB1* para isso. Assume-se que o script que configura as permissões de acesso à porta serial tenha sido executado. Se nada aparecer no terminal, programe a placa e pressione o botão CPU RESET (deverá aparecer uma mensagem do *bootloader*).

O adaptador gráfico possui uma série de limitações, sendo que o hardware gera um sinal VGA padrão, com resolução de 640x480 pixels compatível com qualquer monitor LCD ou CRT. Em função das limitações na plataforma de prototipação, e escolha deliberada de uma pequena memória de vídeo de 32KB, a resolução do adaptador é de apenas 300x218 pixels com 16 cores (4 bits por pixel). Toda a renderização de objetos é realizada por software, portanto será necessário cuidado ao atualizar os objetos para minimizar o número de pixels a serem modificados, com o objetivo de se obter um desempenho aceitável no jogo. A memória de vídeo é uma memória de porta simples, sendo seu acesso multiplexado pelo controlador gráfico e a CPU<sup>2</sup>.

---

<sup>2</sup>A CPU acessa a memória gráfica apenas quando pixels não estiverem sendo apresentados no display, durante a sincronização vertical e horizontal.

## O jogo

O jogo a ser desenvolvido é um clone do clássico Space Invaders<sup>3</sup>, que consiste tentar proteger sua base de um conjunto de inimigos. Para isso, o jogador deverá tentar eliminar os inimigos atirando de um canhão. Os inimigos também irão atirar e deslocarem-se em direção ao jogador, com uma cadência cada vez mais rápida. Existem diversas versões para esse jogo, sendo recomendado consultar as informações<sup>4</sup> sobre a pontuação, os tipos de inimigos, e outras questões para que o seu jogo seja uma versão semelhante. O exemplo no site<sup>5</sup> permite jogar o jogo para se ter uma ideia do seu funcionamento, e no vídeo<sup>6</sup> o jogo é apresentado.

Você pode organizar o seu sistema de cores e apresentação geral do jogo, as regras de pontuação e sua apresentação, tamanho dos inimigos e do seu canhão, velocidade e progressão da mesma com relação aos inimigos, entre outros aspectos. Recomenda-se utilizar *sprites* com o tamanho pequeno, como apresentado no exemplo *vga\_sprite.c* (com exceção da nave mistério, canhão e explosões, que podem ser um pouco maior). As proteções da base (escudos) podem ser maiores, pois são objetos estáticos e não comprometerão o processamento. Anexo ao enunciado, são apresentados os *sprites* sugeridos a serem utilizados para a construção do seu jogo, que foram baseados da versão arcade.

## Avaliação

Os seguintes critérios de avaliação serão utilizados:

1. Organização, modularidade e comentários no código: 2 pontos
2. Funcionamento do jogo (movimentação, detecção de colisão, pontuação): 3 pontos
3. Elaboração do jogo (aparência, jogabilidade): 2 pontos
4. Integração do teclado para o controle do jogo: 3 pontos

Este trabalho deverá ser realizado em duplas ou trios e apresentado no dia 27/11 (apresentação em torno de 5 a 10 minutos). Para a entrega, é esperado que apenas um dos integrantes envie pelo Moodle, até a data e hora especificadas, um arquivo *.tar.gz* ou *.zip* do projeto desenvolvido.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Space\\_Invaders](https://en.wikipedia.org/wiki/Space_Invaders)

<sup>4</sup><https://www.classicgaming.cc/classics/space-invaders/play-guide>

<sup>5</sup>[https://www.retrogames.cz/play\\_016-Atari2600.php](https://www.retrogames.cz/play_016-Atari2600.php)

<sup>6</sup><https://www.youtube.com/watch?v=aj43i9Az4PY>

## Anexo: assets para uso no desenvolvimento

Os *sprites* foram construídos com base na versão arcade do jogo. Você pode utilizar os mesmos, definir as cores ou criar os próprios para o seu jogo.



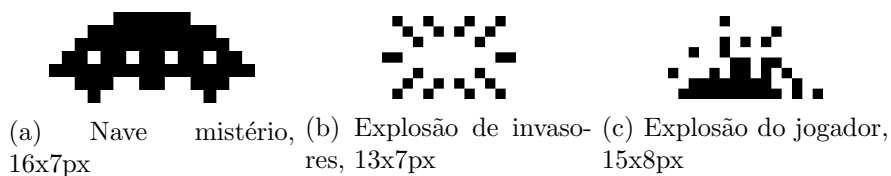
Figura 1: Invasor tipo 1, 8x8px, 30 pontos



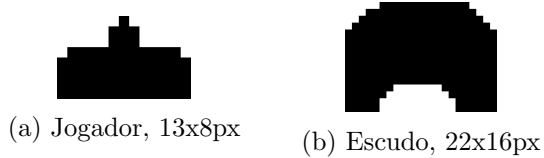
Figura 2: Invasor tipo 2, 11x8px, 20 pontos



Figura 3: Invasor tipo 3, 12x8px, 10 pontos



(a) Nave mistério, 16x7px (b) Explosão de invasores, 13x7px (c) Explosão do jogador, 15x8px



(a) Jogador, 13x8px (b) Escudo, 22x16px