



## Problema #3 – 2024.1

### Sistema Digital

#### 1. Tema

[https://www.youtube.com/watch?v=bC\\_KPLe5dYg](https://www.youtube.com/watch?v=bC_KPLe5dYg)

■ Space\_Invaders\_2.mp4

■ Car\_Race\_2.mp4

■ Asteroids\_game\_2.mp4

#### 2. Objetivos de Aprendizagem

Ao final da realização deste problema, o/a discente deverá ser capaz de:

- Programar em C para um processador com arquitetura ARM;
- Entender o conjunto de instruções do processador gráfico e saber como utilizá-las de acordo com a necessidade do sistema;
- Utilizar as interfaces disponíveis na placa DE1-SoC.

#### 3. Problema

Desenvolver um jogo para arquitetura desenvolvida. O usuário deve interagir como ator do jogo, única e exclusivamente, utilizando o mouse. Para acessar e controlar o jogo (iniciar, pausar, continuar, etc.) o usuário pode utilizar os botões disponíveis na placa. O sistema deve utilizar a interface VGA para visualização do jogo.

#### 4. Requisitos

O problema a ser desenvolvido no Kit FPGA DE1-SoC deve atender às seguintes restrições:

- 4.1. O código deve ser escrito em linguagem C;
- 4.2. O sistema só poderá utilizar os componentes disponíveis na placa;
- 4.3. Um novo sprite deve ser colocado na memória e utilizado no jogo;
- 4.4. As ações do ator do jogo (pulo, tiro, etc.) devem ser comandadas pelos botões do mouse;
- 4.5. A variação da velocidade no movimento do mouse deve ser refletida na ação do ator do jogo. Por exemplo, no jogo breakout a barra se move com velocidade maior se o movimento do mouse for brusco;

- 4.6. Informações do jogo (placar, vidas, etc.) devem ser exibidas no display de 7-segmentos;
- 4.7. O jogo deve permitir ações do usuário através dos botões da DE1-SoC, no mínimo: a pausa, o retorno, o reinício e o término do jogo.
  - 4.7.1. O usuário poderá parar e reiniciar o jogo em qualquer momento;
  - 4.7.2. O usuário poderá sair do jogo em qualquer momento.
- 4.8. Pelo menos um elemento passivo do jogo deverá se mover.

## 5. Produto

Todo o projeto deverá ser disponibilizado na plataforma GitHub. No prazo indicado no cronograma a seguir, cada equipe deverá apresentar:

- 5.1. Levantamento de requisitos;
- 5.2. Código
  - 5.2.1. Código em linguagem C;
  - 5.2.2. Todos os códigos deverão estar detalhadamente comentados;
- 5.3. Script de compilação tipo Makefile para geração do código executável;
- 5.4. Documentação técnica escrita no arquivo READ.ME do projeto no GitHub, contendo, no mínimo:
  - 5.4.1. Detalhamento dos software usados no trabalho, incluindo softwares básicos;
  - 5.4.2. Arquitetura do computador usado nos testes;
  - 5.4.3. Descrição de instalação, configuração de ambiente e execução;
- 5.5. Descrição dos testes de funcionamento do sistema, bem como, análise dos resultados alcançados.

## 6. Cronograma

Semana	Data	Descrição
14	ter. - 18/jun.	Problema 3 – Apresentação
	qui. - 20/jun.	Problema 3 – Seção Desenvolvimento #1
15	ter. - 25/jun.	Feriado – Recesso Junino
	qui. - 27/jun.	Problema 3 – Seção Tutorial #2
16	ter. - 02/jul.	Feriado – Independência da bahia
	qui. - 04/jul.	Problema 3 – Seção Tutorial #3
17	ter. - 09/jul.	Problema 3 – Seção Desenvolvimento #2
	qui. - 11/jul.	Problema 3 – Seção Tutorial #4
18	ter. - 16/jul.	Problema 3 – Seção Desenvolvimento #3
	qui. - 18/jul.	Problema 3 – Seção Tutorial #5
19	ter. - 23/jul.	Problema 3 – Seção Desenvolvimento #4
	qui. - 25/jul.	Problema 3 – Entrega/Avaliação

## 7. Avaliação

Para avaliar o envolvimento do grupo nas discussões e na apresentação, o tutor poderá fazer perguntas variadas a qualquer membro, tanto nas sessões tutoriais quanto na apresentação. O estudante que não comparecer, ou se atrasar, no dia da sessão de apresentação, terá automaticamente nota 0,0 (zero) no problema, excetuando-se as condições que permitem 2ª chamada de avaliações, conforme regulamento do curso.

A nota final será a composição de 3 (três) notas parciais:

Critério	Critérios para a nota	Peso
Desempenho Individual	Participação individual nas sessões tutoriais, de acordo com o interesse e entendimento demonstrados pelo aluno, assim como sua assiduidade, pontualidade e contribuição nas discussões. Essa nota inclui o desempenho do estudante na apresentação do problema no laboratório.	4
Documentação	Documentação técnica de cada grupo, considerando qualidade da redação (ortografia e gramática), organização dos tópicos, definição do problema, descrição da solução, explicação dos experimentos, análise dos resultados, detalhando os itens não atendidos, se for o caso.	2
Códigos	Qualidade do código fonte (organização e comentários), e execução correta dos códigos binários de acordo com testes de validação que explorem as situações de uso.	4