

# **Documentação de Projeto – Parte 2**

## **Design, Estudo da Plataforma**

---

**Projeto:** Enduro

**Autores:** Gabriel Carrico Guerrero

## Parte 2a – Design

### 1 Introdução

Segunda parte do documento de projeto para o desenvolvimento do Enduro em uma plataforma SK-S7G2 da Renesas. O primeiro documento abordava o Domínio do Problema, bem como o CONOPS e a Especificação. Já neste segundo, o foco é no planejamento para a implementação e solução propriamente dita.

Esse projeto é um trabalho de conclusão para a disciplina de Sistemas Embarcados.

### 2 Arquitetura Funcional

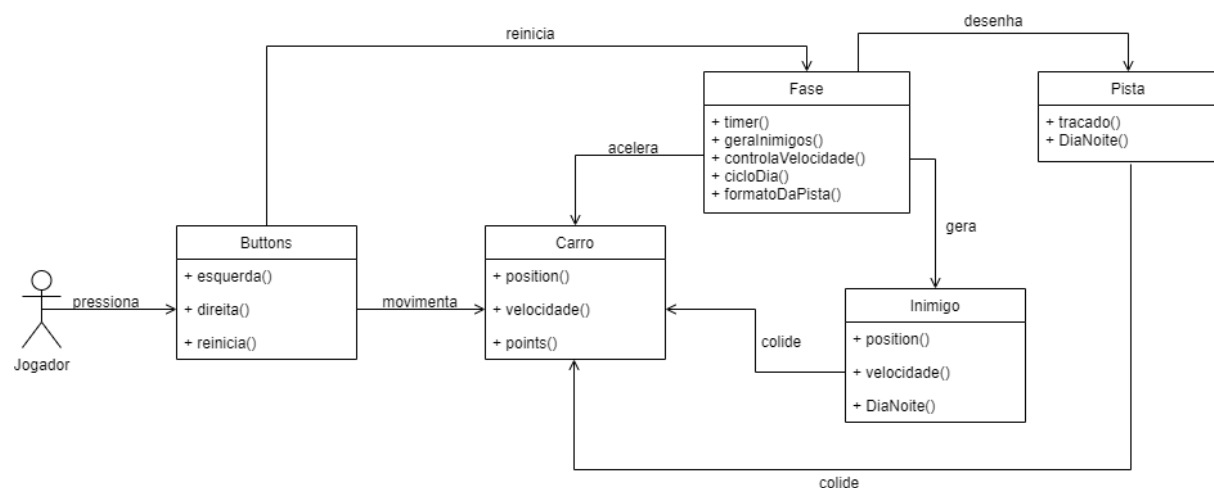


Figura 1 - Arquitetura Funcional

- O jogador interage com o sistema através de 3 botões virtuais.
  - O primeiro deles é um botão de reinício, que deve zerar o timer e a pontuação do carro, bem como retornar os demais parâmetros para as condições iniciais.
  - O segundo botão faz com que o carro do jogador faça uma curva à esquerda.
  - O terceiro botão faz com que o carro do jogador faça uma curva à direita.
- O carro do jogador deve ter como parâmetros sua posição atual, com a qual será feito os cálculos de colisão, bem como sua velocidade e sua pontuação.
- Toda a fase possui um timer, que ao esgotar causa o fim do jogo. Por outro lado, se o jogador ultrapassar o número correto de inimigos, o timer é acrescentado e o ciclo do dia muda. Cada acréscimo do timer causa uma aceleração do carro do jogador. A fase também é responsável pela disposição dos inimigos e o desenho da curva.

- Cada inimigo gerado deve percorrer o desenho da pista e estar a uma velocidade sempre menor que a do carro do jogador. Em caso de colisão com o jogador, a partida deve ser finalizada.
- A pista deve impor os limites do traçado, caso o jogador ultrapasse esses limites, a partida deve ser finalizada.

### 3 Arquitetura Física

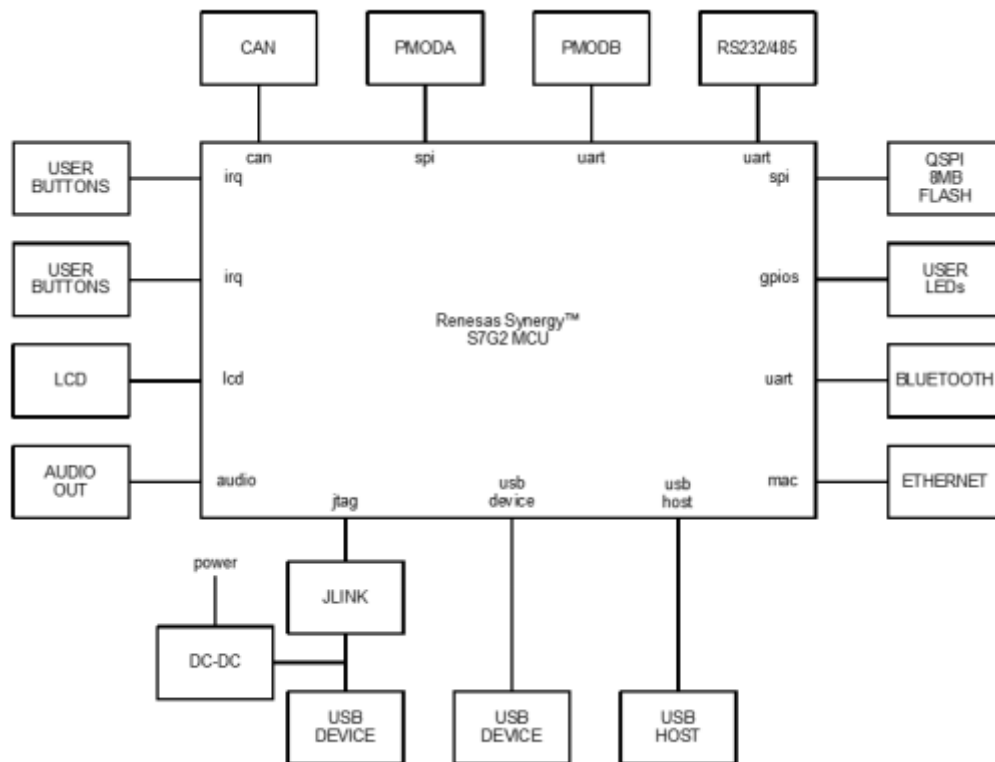


Figura 2 - Diagrama de Blocos SK-S7G2  
[fonte: Renesas Synergy Starter Kit SK-S7G2 User's Manual]

A placa escolhida para a solução do problema é a SK-S7G2. Isso foi decidido com base em uma restrição do projeto, como pode ser visto no Documento 1. A S7G2 é uma escolha adequada, uma vez que apresenta um microcontrolador suficientemente potente para executar o programa, e uma interface integrada de vídeo para a visualização do jogo. Na imagem acima, podemos ver essa imagem como sendo o LCD.

O LCD é conectado diretamente à interface LCD do MCU. O painel LCD possui touch screen e utiliza um driver Ilitek para a Interface de Comunicação Serial (SCI) da MCU. A comunicação é configurada para ser de 4-fios de 8 bits.

O touch screen é sentido através de um controlador resistivo, conectado a S7G2 através de uma interface IIC.

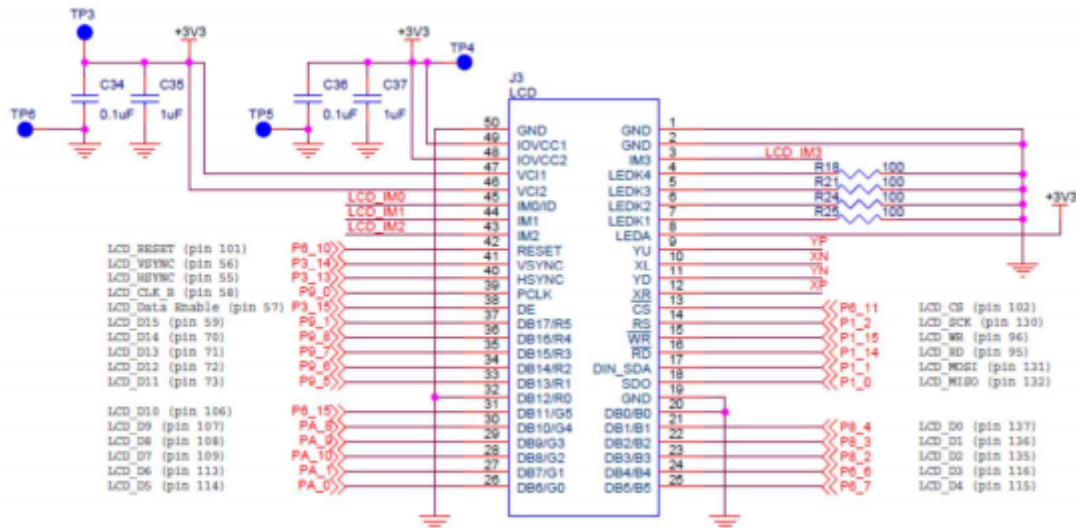


Figura 3 - Conexão entre MCU e LCD  
[fonte: Renesas Synergy Starter Kit SK-S7G2 User's Manual]

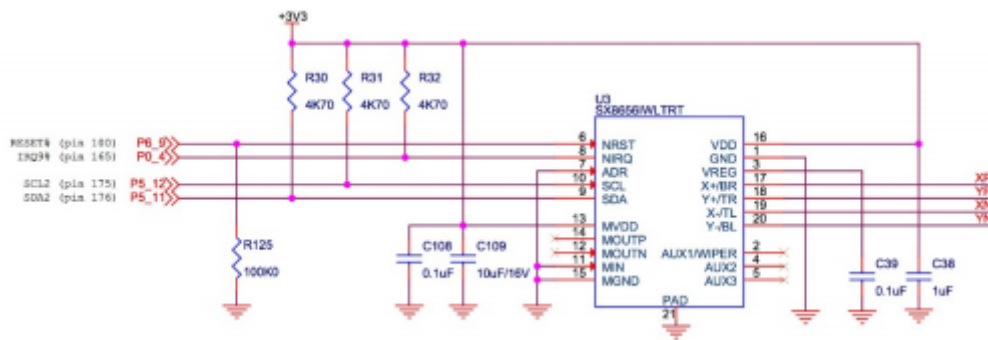


Figura 4 - Controlador do Touch Screen  
[fonte: Renesas Synergy Starter Kit SK-S7G2 User's Manual]

Outro aspecto importante que será utilizado é o timer. Por se tratar de um jogo de sobrevivência, o controle do timer, tanto dos inimigos quanto do tempo restante é vital para uma boa aplicação do Enduro. No caso da S7G2, a placa possui o GPT (General PWM Timer) que pode ser visto na figura abaixo.

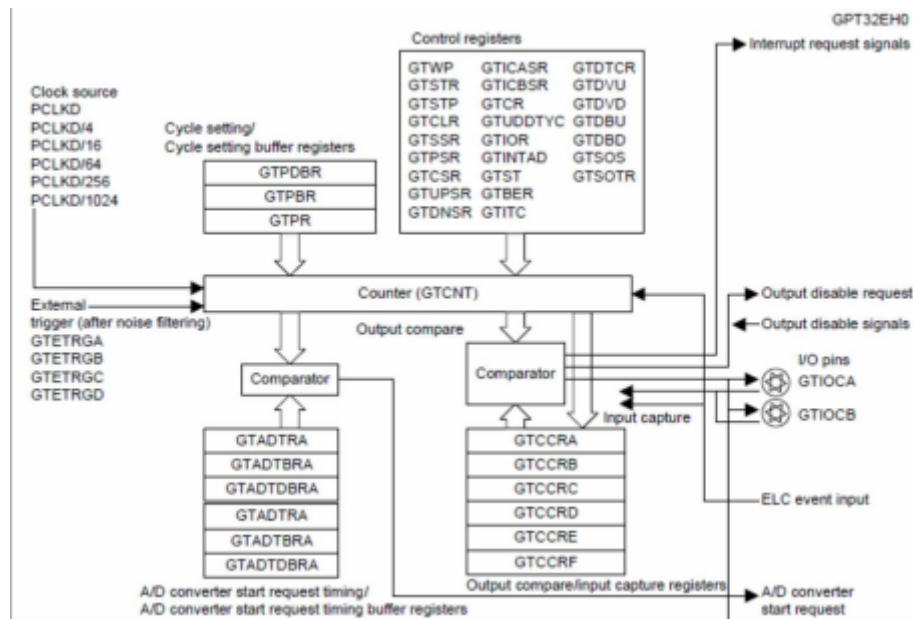


Figura 5 - General PWM Timer [fonte: Renesas S7 Series Microcontrollers User's Manual]

O GPT pode contar tanto para incrementando como decrementando, gerar interrupções, acionar uma conversão analógico-digital e contar os pulsos da PCLKD. O timer pode ser periódico ou one-shot, tendo 14 canais, cada um com um contador de 32-bits.

## 4 Interface com o Usuário

A interface com o usuário será feita por meio do GUIX. O GUIX Studio é um aplicativo a parte que pode ser instalado juntamente com o e2 Studio e é utilizado para soluções que necessitem de interface gráfica.

Com o GUIX Studio é possível criar botões e ações de forma simplificada, com um extenso tutorial disponível que auxilia no passo a passo do set-up da solução.

Utilizando-se do GUIX o LCD terá suas atualizações, para que o usuário possa acompanhar o desenrolar da partida. O GUIX também interfaceia a tela touch screen, que permite a criação dos botões virtuais que serão aplicados na nossa solução. O Documento 1 possui uma série de imagens com exemplos de como imaginamos a interface gráfica do Enduro, com os botões já posicionados. A leitura do Documento 1 é de extrema importância para o entendimento do projeto. De forma semelhante, a leitura do documento de setup do GUIX se faz necessária, para consultar, buscar por Renesas Application Note R12AN0021EU0118 no site da Renesas.

## 5 Mapeamento da Arquitetura Funcional à Arquitetura Física

Por se tratar de um projeto relativamente simples, não será necessário a conexão de múltiplas partes da arquitetura física. Todo o projeto será implementado utilizando somente o LCD e o MCU da S7G2.

Importante ressaltar que o projeto fará uso do RTOS para o controle sincronizado de todos os seus componentes. Para tal será utilizado o ThreadX, mas por ser parte do Sistema Operacional, também se trata de uma implementação interna da MCU.

## 6 Arquitetura do Hardware

---

Por se tratar de uma plataforma já pré definida, o nosso projeto fará uso da placa SK-S7G2 da Renesas. Como dito anteriormente, os componentes que serão utilizados verdadeiramente pela nossa solução são apenas a MCU e toda a parte de LCD e touch screen. A descrição desses componentes também já foi feita em seções anteriores.

## 7 Design Detalhado

---

Por se tratar de uma plataforma já pré definida, o nosso projeto fará uso da placa SK-S7G2 da Renesas. Portanto, nenhum elemento na Arquitetura Física precisa ser desenvolvido.

Abordando um pouco os elementos de software que irão compor o sistema, temos uma série de máquinas de estado que serão apresentadas a seguir:

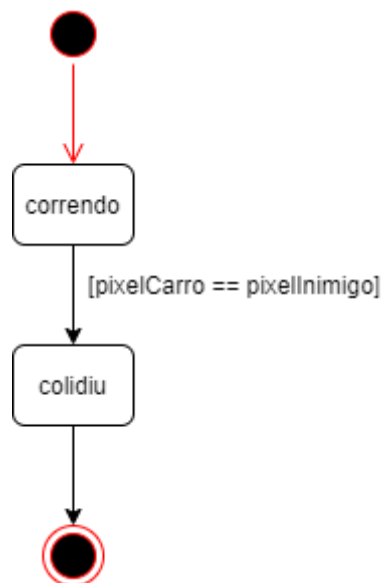


Figura 6 - Máquina de Estados: Colisões

Começando pela mais simples delas, temos o sistema de colisão. Ela basicamente será iniciada ao começo da partida e ficará sempre observando, caso a posição de um pixel do carro seja igual a posição de um pixel do inimigo, isso indica que houve colisão, ocorre a troca de estado e somos informados do ocorrido.

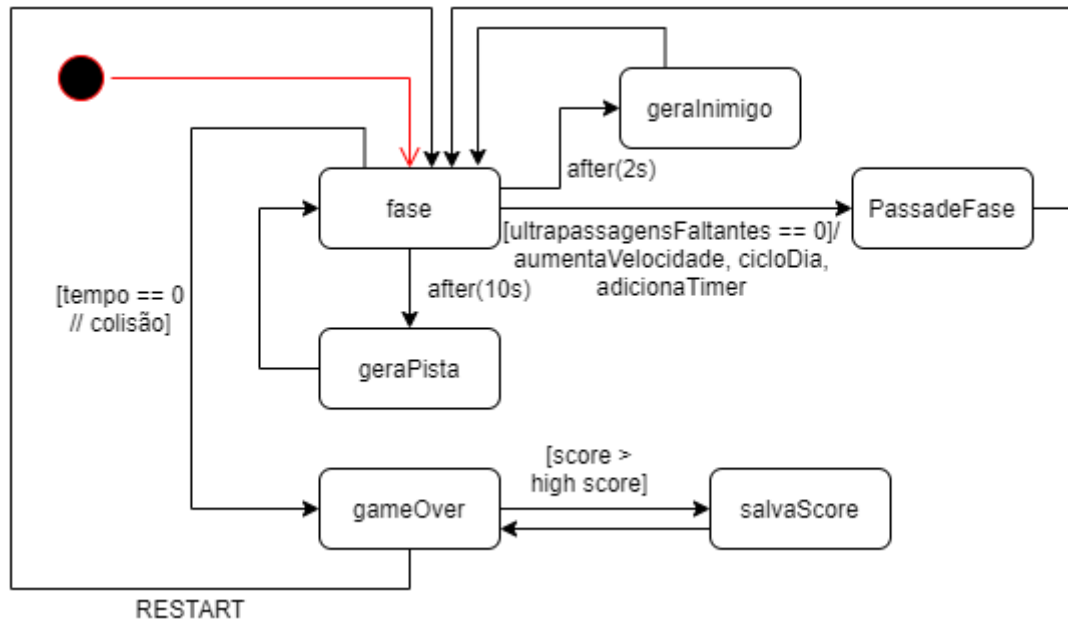


Figura 7 - Máquina de Estados: Pista

Essa informação é importante pois, caso ocorra colisão, o jogo deve encerrar a partida e ir para a tela de Game Over, como pode ser observado pela máquina de estados da fase. Também temos que a cada 10s ela deve gerar um novo formato de pista, este podendo ser uma curva para qualquer um dos lados bem como uma linha reta. A cada 2s ela deve gerar um inimigo, que será abordado na própria máquina de estados dele. E caso o jogador consiga ultrapassar inimigos o suficiente para passar de fase, deve ocorrer um aumento da velocidade, uma alteração no ciclo do dia e um aumento do timer. Também é importante adicionar que caso o tempo chegue a zero, a partida se encerra. Um último aspecto importante é que o score só é salvo caso seja o maior de todos, e um restart pode ser feito durante a tela de Game Over.

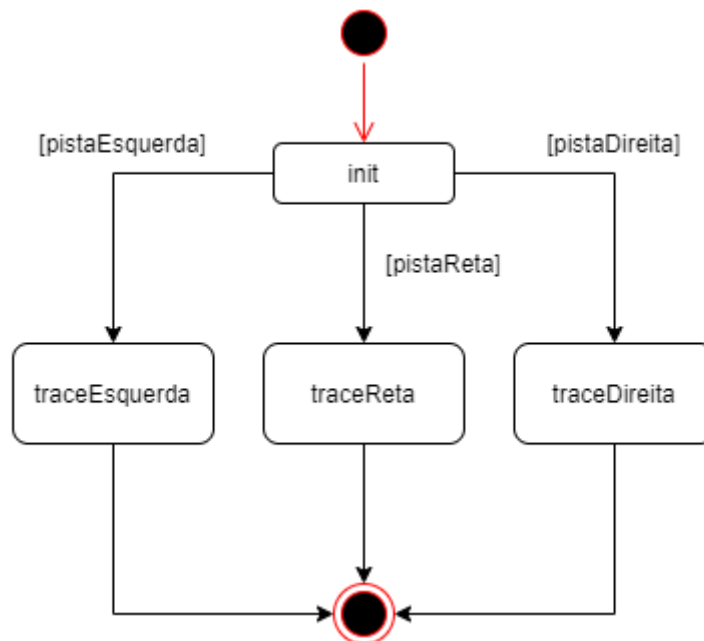


Figura 8 - Máquina de Estados: Inimigos

Falando do inimigo, ele é gerado no topo da tela pela fase e, dependendo do tipo de curva que a pista está apresentando, ele seguirá um traçado correspondente. Quando o carro chegar ao fim da tela, sua execução é finalizada e ele desaparece.

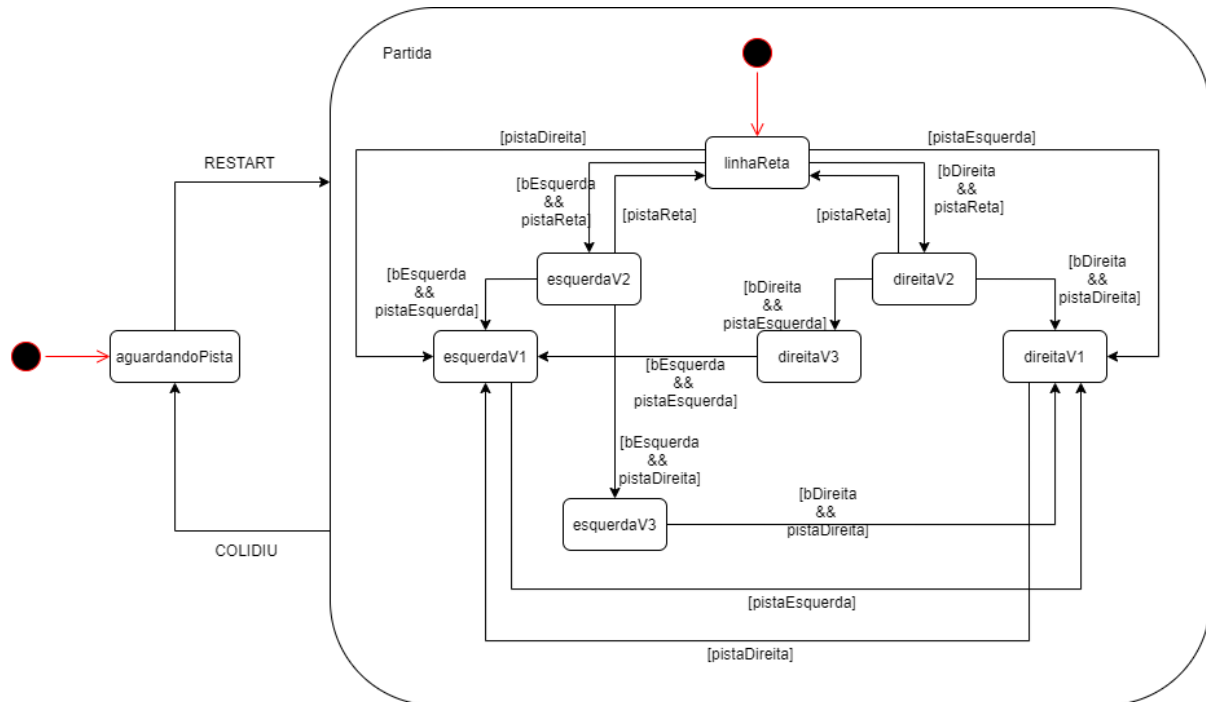


Figura 9 - Máquina de Estados: Carro do Jogador

Por último temos o funcionamento do carro do jogador. Ao início do jogo ele é gerado e fica aguardando. Caso ocorra um início (ou reinício) ele estará inicialmente em linha reta. A partir daí existem 7 estados possíveis para o carro do jogador. Caso a pista esteja reta e ele não pressione nenhum botão, seguirá em linha reta. Caso a pista faça uma curva e ele não pressione nenhum botão, será levemente empurrado na direção contrária. Caso a pista faça uma curva e ele pressione o botão correto, será levemente puxado na direção da curva. Caso a pista faça uma curva e ele pressione o botão errado, será bruscamente empurrado na direção contrária. Por último, caso a pista seja reta e ele pressione um botão, ocorrerá um deslizamento moderado na direção escolhida. Em caso de colisão, o carro do jogador retorna para o estado inicial, aguardando uma nova partida.

## Parte 2b – Estudo da Plataforma

### 1 Timer

A primeira questão a ser destacada é que o Enduro opera através do controle de timers. Para um desenvolvimento bem sucedido, é importante ter pleno domínio dos timers, para



poder dispará-lo, bem como acrescentar tempo adicional ao mesmo e interromper a operação caso esse chegue a zero.

## 2 LCD

---

Por se tratar de um jogo com interface gráfica, outro aspecto vital para o desenvolvimento do Enduro é o estudo e entendimento do LCD. É através deste dispositivo que todo o jogo se desenrola, permitindo ao jogador enxergar o próprio carro, bem como o trajeto da pista e seus competidores.

## 3 Touch Screen

---

De forma análoga, é através do touch screen que o usuário terá acesso aos controles do jogo. Tanto as opções de mudar o trajeto, quanto o próprio reinício do jogo ficam comprometidos caso o Touch Screen não seja utilizado corretamente.

## 4 Interface gráfica

---

O controle preciso das informações presentes na tela é o que faz com que o jogo aconteça. Posicionamento dos elementos na tela, taxa de atualização de quadros, controle e escolha dos sprites adequados, o Enduro só funcionará corretamente caso todos esses aspectos sejam aplicados conforme se deve.

## 5 Colisões

---

Um sistema de colisões mal feito pode acabar com a experiência de um jogo ativo como o Enduro. Portanto um estudo sobre qual o melhor método de aplicação desses conceitos deve ser seguido de forma a deixar a experiência o melhor possível. Isso é especialmente vital numa aplicação que tem o jogo atrelado à sobrevivência do jogador, que não pode colidir com outros elementos do cenário e outros carros.

## 6 Arquivo

---

Um último aspecto importante é de conseguir armazenar o valor do melhor tempo feito por outro jogador. Pelo Enduro ser um jogo de bater recordes, a experiência fica bastante prejudicada caso esse valor seja perdido ou resetado toda vez que o jogo se reiniciar. É importante salvar essa informação entre uma sessão e outra de uso do sistema, mesmo após o desligamento da plataforma.

## 7 GUIX

---

Para o setup do ambiente, será utilizado o tutorial Hello World disponível no site da Renesas, que auxilia na criação de um ambiente base que pode ser posteriormente alterado. O GUIX possui uma série de funções já implementadas que facilitarão muito o trabalho de desenvolvimento da interface gráfica do projeto. Sem ele, seria impossível desenvolver o Enduro em tempo hábil para o final da disciplina. De forma semelhante, o GUIX também será o coração do desenvolvimento, uma vez que um jogo é um tipo de implementação que depende muito da interação com o usuário, e essa interação será feita por meio dos botões touch na tela. No site <https://docs.microsoft.com/pt-br/azure/rtos/guix/chapter-4> temos uma lista com os serviços já implementados pelo guix, que certamente serão utilizados no projeto.

