

Documentação de Projeto – Parte 1

CONOPS, Domínio do Problema, Especificação

Projeto: Embedded Snake

Autores: Gustavo Henrique Zeni
Lucas Perin Silva Leyser

Parte 1a – CONOPS

1 Introdução

O projeto consiste em criar uma versão, de sistema embarcado, do jogo “Snake” ou “Cobrinha” como é popularmente conhecido no Brasil, um dos primeiros jogos eletrônicos criados e que se tornou extremamente popular em 1998 ao ser lançado nos celulares da Nokia.

Considerando a crescente popularização dos jogos eletrônicos que vem acontecendo nos últimos anos, um sistema como esse pode representar uma boa perspectiva de lucro, principalmente se escalado com outros títulos já conhecidos.

2 Descrição do Sistema

O jogo consiste em uma linha, a cobra, que se movimenta ininterruptamente e tem sua direção alterada através dos direcionais, um ponto aparece em um lugar aleatório da tela e, caso o jogador consiga direcionar a cobra até aquele ponto, ela aumenta de comprimento e sua movimentação é acelerada, o jogo acaba quando a cobra cresce a ponto de ocupar toda a tela do jogo ou caso a ela colida com si mesma ou com uma das paredes do cenário.

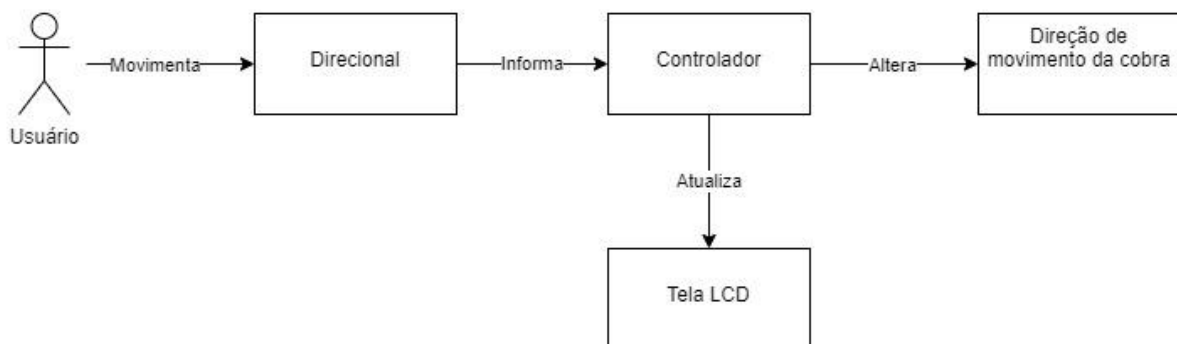


Diagrama com os principais elementos do sistema.

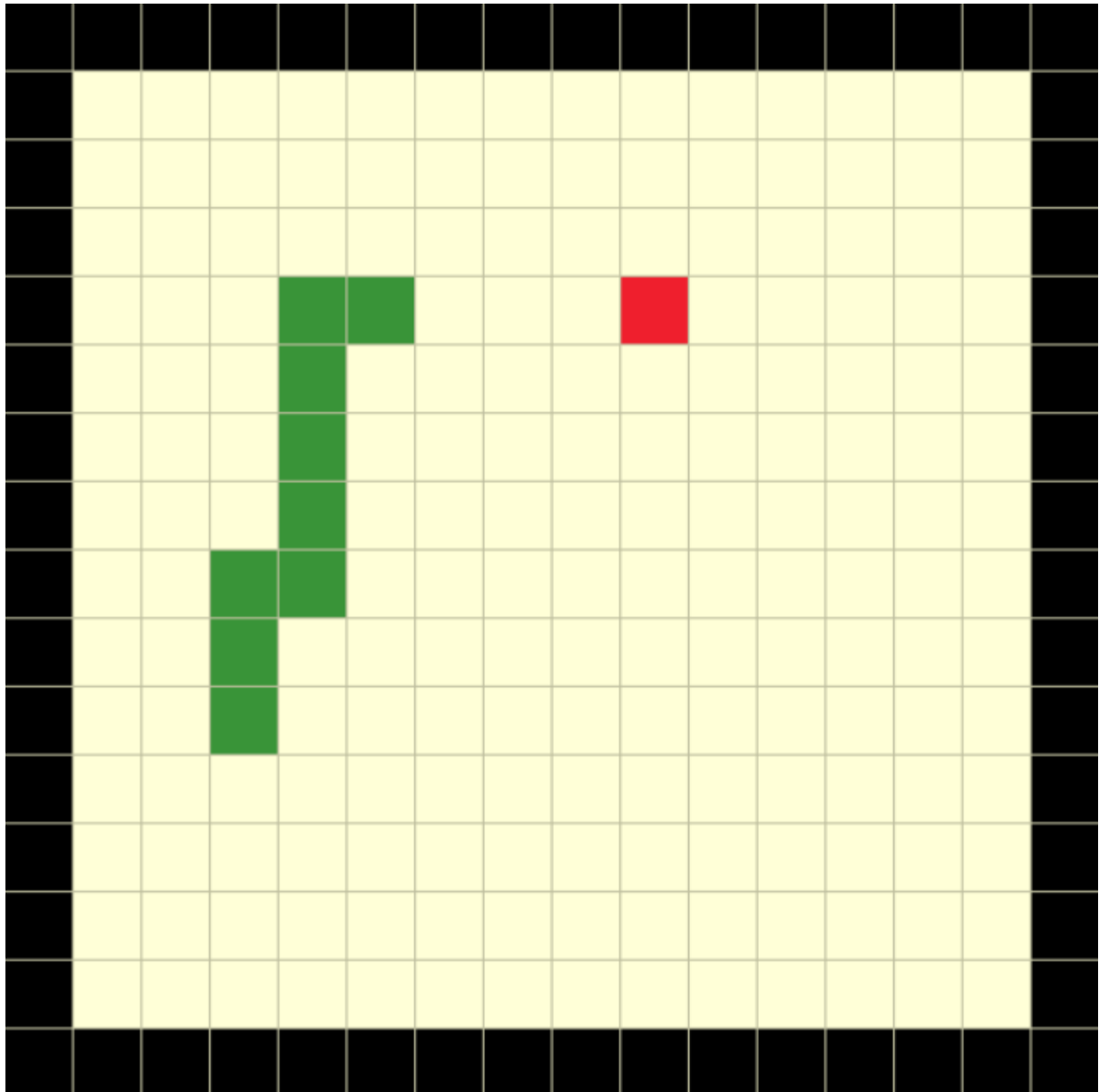
3 Interface com o Usuário

Toda interação com o usuário final será feita através do *BoosterPack* Educacional MKII, que conta com uma tela LCD, onde o jogo será exibido, um joystick que controlará a direção da cobra e um botão responsável por iniciar e pausar o jogo, além de outros sensores e atuadores, que não serão utilizados nesse sistema.



BoosterPack Educacional MKII.

O cenário do jogo será um tabuleiro de 16x16 posições, as primeiras e últimas linhas e colunas demarcaram as paredes, restando uma área de 14x14 para a movimentação da cobra que terá 5 posições de comprimento no início do jogo.



Prévia da Interface do Jogo.

4 Identificação dos Stakeholders

Os Stakeholders do projeto são:

- Membros da equipe de desenvolvimento.
- Desenvolvedor da placa Tiva.
- Desenvolvedor do kit educacional.
- Criador do jogo *Snake*.
- Interessados em jogos e sistemas embarcados.
- Equipes de manutenção de sistemas embarcados.
- Equipes de suporte de sistemas embarcados.
- Varejistas de artigos eletrônicos.
- Instituições de ensino em áreas de tecnologia.
- Instituições de coleta de resíduos eletrônicos.

5 Necessidades de Stakeholders

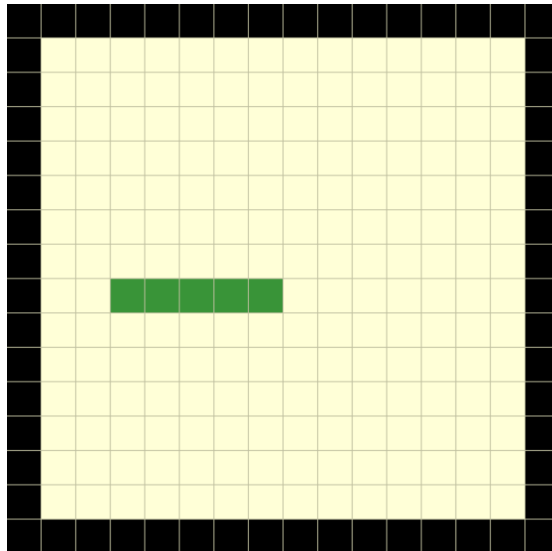
Podemos listar as necessidades dos Stakeholders como:

- **Equipe de desenvolvimento** - Implementar o jogo Snake de maneira satisfatória.
- **Equipe de desenvolvimento** - Integrar o software do jogo com o sistema embarcado.
- **Equipe de desenvolvimento** - Integrar o sistema embarcado com o kit educacional.
- **Equipe de desenvolvimento** - Obter lucro com a venda do sistema.
- **Desenvolvedores da placa Tiva** - Aumento nas vendas da placa.
- **Desenvolvedores do kit educacional** - Aumento nas vendas do kit educacional.
- **Criador do jogo Snake** - Participação nos lucros obtidos com o sistema.
- **Compradores** - Diversão com o jogo.
- **Equipes de suporte e manutenção** - Ganho de clientes com as vendas do sistema.
- **Varejistas de artigos eletrônicos** - Obtenção de lucro com as vendas do sistema.
- **Instituições de ensino** - Utilização do sistema para o aprendizado de alunos.
- **Instituições de coleta de resíduos eletrônicos** - Oportunidade para o aumento nas arrecadações de lixo eletrônico.

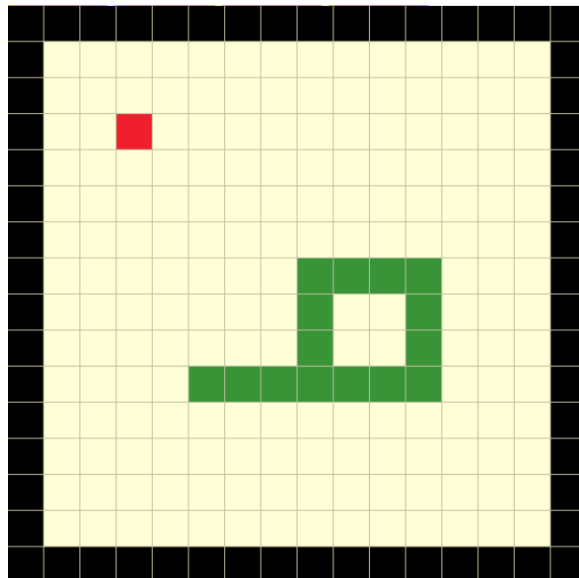
6 Cenários de Operação

O sistema foi projetado para rodar uma versão completa do jogo Snake – cobra. Ele começa na tela inicial, onde o usuário pode iniciar o jogo. Ao iniciar, o texto da tela se apaga e é apenas possível visualizar uma linha (a cobra) e um ponto (a comida). A cobra inicia numa posição fixa e a comida também, parados. Assim que o usuário movimenta o analógico, a cobra começa a se mover na direção apontada e o jogo começa. O objetivo é fazer com que a cobra colete o maior número de comidas possível sem colidir com as paredes ou com o próprio corpo. Após colidir com a comida, o tamanho da cobra aumenta em 1 e a comida reaparece em um lugar aleatório da tela.

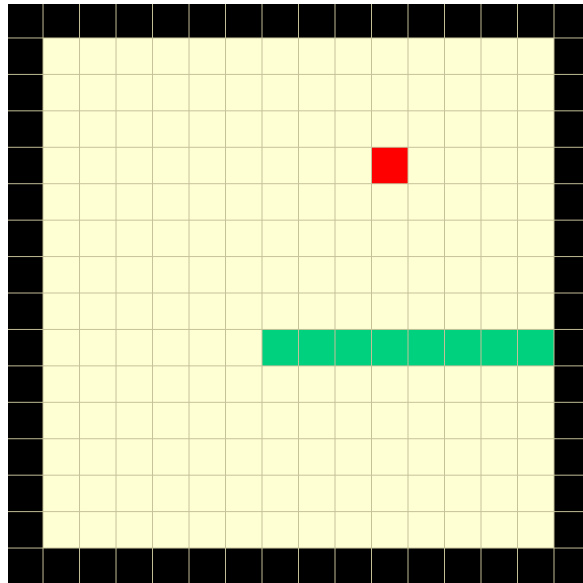
Tratando-se de um jogo simples, os cenários previstos são, em sua maioria, os estágios finais de cada partida, exemplificados abaixo. Além destes, existem apenas as situações em que o movimento da cobra encontra a comida, e seu tamanho aumenta, e quando a comida não é encontrada, apenas movimentando a cobra.



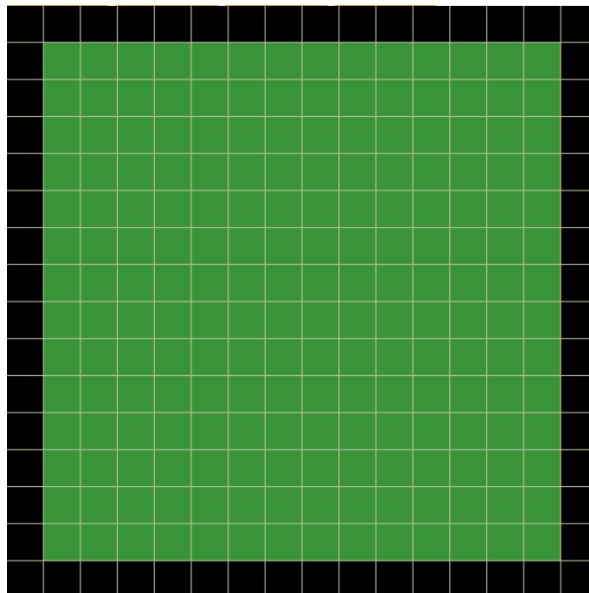
Primeiro instante, onde a cobra tem tamanho 5.



Cenário de “Game Over” por colisão com o próprio corpo.



Cenário de “Game Over” por colisão com uma das paredes do cenário.



Fim de jogo por tamanho máximo.

Parte 1b – Domínio do Problema

1 Responsividade aos Comandos

Para uma experiência condizente com o esperado, é necessário que o sistema reaja corretamente a todos os comandos dados pelo usuário, a leitura dos movimentos do direcional e do pressionar do botão de pausa devem ser realizados com cuidado para que nenhum deles seja interpretado erroneamente, mas também rápido o bastante para evitar que, devido a demora no processamento de um comando, outro comando seja ignorado pelo sistema. Dessa forma, garantimos que o usuário não perderá uma partida por conta do mau funcionamento do sistema.

2 Interfaceamento com o Display LCD

Como todos os comandos dados pelo usuário são uma resposta dele ao que está sendo exibido na tela, o estado interno do jogo deve ser reproduzido fielmente em todos os momentos, principalmente a velocidade com que a tela é atualizada, pois pode impactar muito a experiência do usuário com o sistema como um todo. Mais uma vez, pensando na satisfação do usuário final, o mau funcionamento do sistema não pode ser o responsável pelo término de uma partida.

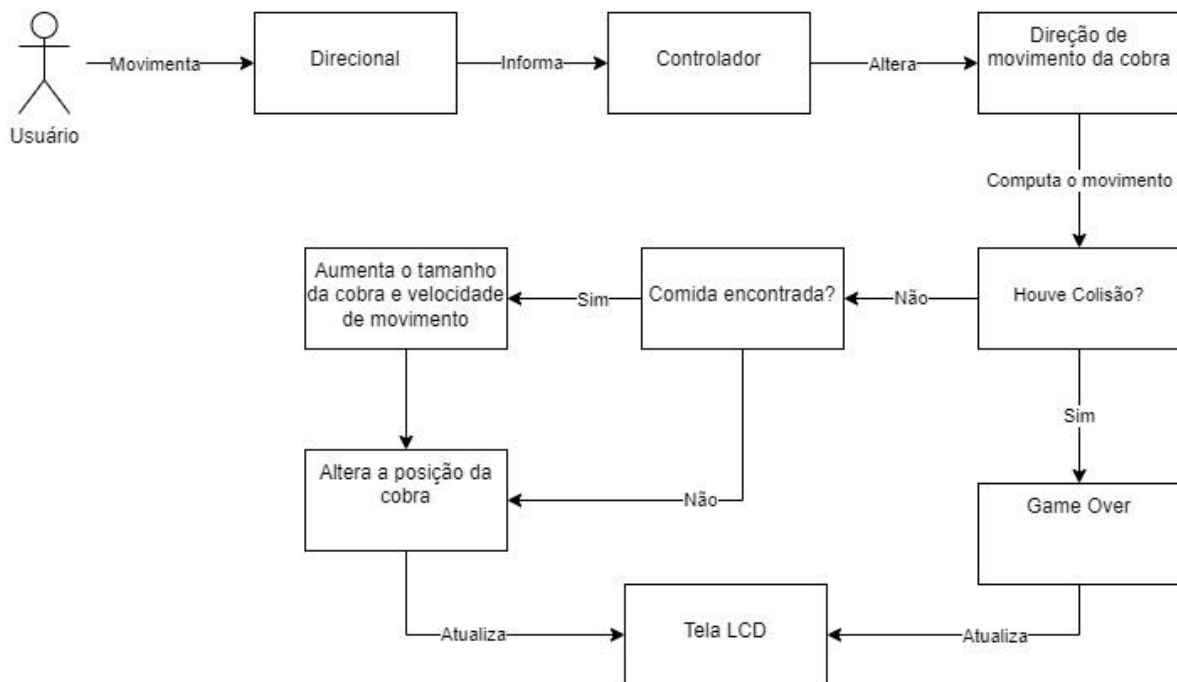
Parte 1c – Especificação

1 Introdução

Nesta etapa do documento, a arquitetura e os requisitos do sistema serão apresentados.

2 Estrutura do Sistema

Considerando o jogo aqui implementado, a principal interação que o usuário terá com o sistema é através do direcional que controla a direção de movimento da cobra, um funcionamento geral de como o sistema interpreta esses comandos pode ser visto no diagrama abaixo.



3 Especificação Funcional

RF01: O sistema deve receber e reagir a toda movimentação do direcional enquanto o jogo é executado.

RF02: O sistema deve pausar o jogo quando o botão de pause for pressionado.

RF02.1: O sistema deve exibir uma mensagem de “Pause” sempre que o jogo for pausado.

RF03: O sistema deve atualizar a tela a cada movimentação da cobra.

RF04: O sistema deve aumentar a velocidade de movimentação da cobra em 5% do valor atual a cada vez que a cobra colidir com uma “comida”.

RF05: O sistema deve aumentar o tamanho da cobra, em uma unidade, a cada vez que a cobra colidir com uma “comida”.

RF06: O sistema deve terminar o jogo caso a cabeça da cobra colida com alguma parte do corpo.

RF07: O sistema deve terminar o jogo caso a cabeça da cobra colida com alguma posição das paredes do cenário.

RF08: O sistema deve exibir uma mensagem de “Game Over” sempre que o jogo terminar por colisão.

RF09: O sistema deve terminar o jogo caso a cobra cresça a ponto de ocupar toda a área jogável do cenário, 256 posições de comprimento.

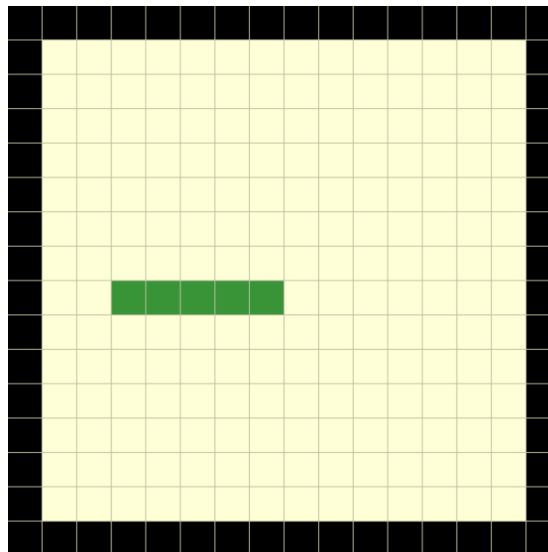
RF10: O sistema deve exibir uma mensagem de “You Won” sempre que o jogo terminar por tamanho máximo da cobra.

RF11: O sistema deve exibir o tamanho final da cobra, abaixo da mensagem de término, sempre que o jogo terminar.

RF12: O sistema deve colocar a comida em uma posição aleatória do cenário, desde que não faça parte das paredes ou do corpo da cobra.

RF13: O sistema deve iniciar o jogo caso o botão de *pause* seja pressionado na tela de início.

RF14: O sistema deve inicializar o jogo com a cobra de tamanho 5, cabeça na posição 7x7 e o corpo na horizontal, como na imagem abaixo.



4 Especificação Não Funcional

RNF01: O sistema será desenvolvido utilizando a placa Tiva TM4C1294NCPDT.

RNF02: O usuário deverá interagir com o jogo utilizando o kit Boosterpack Educacional Mk-II.

RNF03: A visualização do jogo será através do LCD 128x128 do kit Boosterpack Educacional Mk-II.

RNF04: O jogo terá uma taxa de atualização que varia com a velocidade de movimento da cobra, não excedendo o limite do LCD de 20 *frames per second*.

RNF05: O sistema recebe apenas comandos na horizontal ou vertical, movimentos na diagonal são interpretados como a direção que for detectada primeiro.

5 Restrições

RT01: O sistema será programado em linguagem C/C++ e/ou Assembly.

RT02: O sistema será desenvolvido na plataforma IAR Embedded Workbench for ARM.

RT02: O sistema não salvará informações ou progresso referente às partidas passadas.