

# Estudo de Caso - Jogo *Snake*



## Objetivos pedagógicos

1. Avaliar a capacidade do aluno em compreender a codificação de uma aplicação funcional completa em Javascript
2. Avaliar a capacidade do aluno em estender esta codificação com novas funcionalidades, mantendo a fidelidade ao paradigma

## Descrição da aplicação

A aplicação-base para o trabalho é uma implementação em Javascript do tradicional jogo *Snake*. A codificação segue o Paradigma Funcional.

O jogo consiste de uma "cobra", controlada pelo usuário através do teclado, que se movimenta sem interrupção em quatro possíveis direções: Norte, Sul, Leste, Oeste. Existe uma "maçã" que aparece aleatoriamente na tela e, ao comê-la, a cobra cresce de tamanho.

Esta implementação apresenta duas versões de interface: (i) *console* e (ii) *web* (com HTML Canvas).

- A versão *console* está organizada em 3 arquivos: `base.js`, `snake.js` e `cli.js`. Para executar o jogo, basta abrir um `terminal` (pode ser no próprio VSCode) e digitar o comando `node cli + ENTER`.

- A versão *web* também está organizada em 3 arquivos: `base.js` , `snake.js` e `web.js` (além, naturalmente, dos arquivos HTML e CSS para a interface: `web.html` e `web.css` ). Para executar o jogo, basta clicar duas vezes no arquivo `web.html` .

## ARQUIVO-BASE

# O que deverá ser feito?

Primeiramente, a equipe deverá decidir a versão que deseja trabalhar: *console* ou *web*.

Então, de posse dos arquivos respectivos, a equipe deverá cumprir 3 tarefas, conforme descrito a seguir.

1. A fim de provar a capacidade de compreensão da lógica algorítmica utilizada para implementar o jogo, a equipe deverá adicionar **comentários explicativos detalhados** para todos os trechos codificados.
2. A fim de provar o entendimento sobre aspectos/princípios da Programação Funcional estudados, a equipe deverá **selecionar funções codificadas que sirvam para ilustrar aquele aspecto/princípio em questão**. Caso não seja possível identificar um determinado aspecto/princípio, a equipe deverá mostrar (em comentário no código) como aquele trecho ou função poderia ser modificado a fim de ilustrar o aspecto/princípio adequadamente.
3. A fim de mostrar capacidade de programação funcional criativa, a equipe deve **propor e implementar novas dinâmicas de jogo**. Algumas sugestões são:
  - Acrescentar um contador de pontos (ex: diretamente proporcional ao tempo de sobrevivência e inversamente proporcional ao número de vezes em que as teclas de movimentação foram pressionadas)
  - Acrescentar outra cobra concorrente com movimentação aleatória (com risco de morte ao choque)
  - Modificar para que paredes passem a ser sólidas com morte ao choque
  - Acrescentar mais maçãs
  - Acrescentar veneno além de maçãs
  - ...

# Como apresentar a solução?

A equipe deverá manter todos os arquivos de sua solução dentro de uma pasta chamada `snake` e compactar esta pasta em `snake.zip` . A seguir, **um ÚNICO INTEGRANTE** da equipe **deverá realizar upload no CLASSROOM** desse arquivo zip.

# Pontuação

1. **[até 3pts]** Quantidade e profundidade dos comentários explicativos
2. **[até 3pts]** Quantidade de aspectos/princípios ilustrados
3. **[até 4pts]** Quantidade (e criatividade) das novas dinâmicas de jogo implementadas corretamente
4. **[até 1pt EXTRA]** Qualidade da demonstração do trabalho realizada no dia e horário marcados para a respectiva equipe e desempenho na arguição na ocasião da apresentação. **ATENÇÃO:** será realizado sorteio dos integrantes responsáveis pela demonstração do trabalho e pela arguição. Se os integrantes não estiverem presentes ou tiverem mal desempenho, a equipe perderá a chance da pontuação extra.

⚠ **Nunca é demais lembrar que, na implementação das novas dinâmicas de jogo, o uso de construções que firam o paradigma funcional estão sujeitas a penalizações severas (exceção é feita ao controle do Estado de jogo que, de fato, trata-se de um elemento mutável e, assim, deve permanecer).**

## PRAZO

**TODAS AS EQUIPES** devem fazer o upload do arquivo `snake.zip` até às **14h59 do dia 24/05/2022**.

⚠ **Envios de arquivo realizados diretamente por e-mail, sob qualquer justificativa, serão sumariamente IGNORADOS. Os envios devem ser feitos EXCLUSIVAMENTE pelo Google Classroom da turma.**

## Duas soluções disponíveis

Os dois arquivos compactados a seguir, disponíveis para download são soluções para o trabalho proposto. As soluções foram desenvolvidas por duas equipes diferentes de uma das turmas da disciplina de Programação Funcional do DCOMP/UFS no período 2021.2.

Recomenda-se tentar resolver o problema antes de visitar essas soluções. Ainda que tome-se conhecimento das soluções disponíveis, recomenda-se realizar modificações e testes nas variadas funções para experimentar e treinar alternativas interessantes.

**SOLUÇÃO 1: versão *Web*** 

**SOLUÇÃO 2: versão *console*** 