

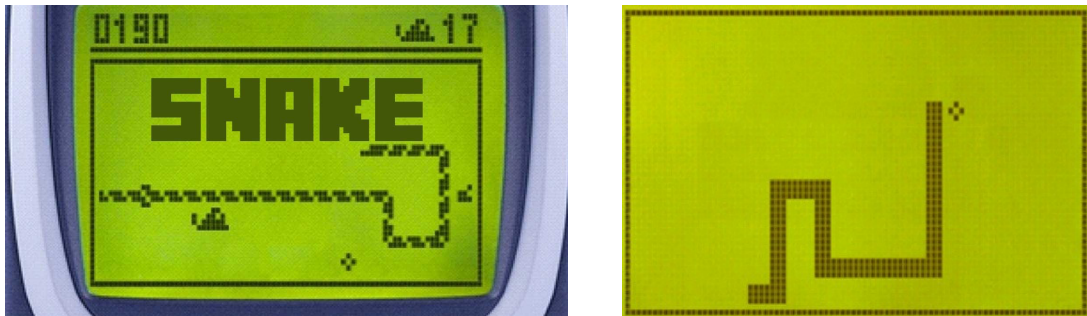


### **Primeiro Trabalho**

#### **Motivação:**

Jogos eletrônicos costumam demandar por estruturas de dados eficientes e robustas em várias de suas etapas. Mesmo jogos simples podem requerer estruturas de dados apropriadas para que o roteiro do jogo possa ser implementado de forma eficiente.

Um jogo bastante simples que pode demonstrar como boas escolhas na estrutura de dados podem simplificar a programação é o famoso *Snake* [5]. Como podemos ver na Figura 1, o jogo é bastante simples: o jogador “guia” uma cobrinha que se alimenta de elementos colocados na tela. Cada vez que ela come um alimento seu comprimento aumenta com um novo segmento. A cobra “morre” caso ela “coma a si mesma”, ou seja, sua cabeça intercepte seu corpo, ou se ela bater nas paredes que delimitam a tela.



*Figura 1 - Telas do jogo Snakes Clássico.*

#### **Objetivos do trabalho:**

O objetivo principal do trabalho é aplicar os conceitos de Tipos Abstratos de Dados (TAD) Lineares [1][2][3] apresentados em sala de aula, no contexto de um problema real e prático.

Os alunos deverão analisar o problema proposto, e definir qual o melhor TAD a ser utilizado na resolução do problema. Mais de um TAD pode ser necessário para que se possa atingir a solução do problema.

Escolhidos os TADs, sua implementação deverá ser integrada ao problema e uma solução computacional codificada em linguagem C.

## O Problema:

O problema é desenvolver uma versão de jogo eletrônico para PC, baseado no jogo *Snake*, que chamaremos de **Cobra**. Nesse jogo teremos também uma cobra que cresce conforme se alimenta. No início do jogo existe apenas a cabeça da cobra, que se movimenta de forma automática. A direção do movimento é controlada pelo usuário, utilizando as setas do teclado. Cada vez que a cobra se alimenta, seu corpo cresce em um segmento. Caso a cabeça da cobra cruze o seu corpo a cobra morre.

Para tornar o jogo mais “emocionante” (pelo menos do ponto de vista da disciplina de Estrutura de Dados), algumas regras novas serão consideradas:

1. Alimentos serão bolinhas coloridas, cada um com um valor energético (pontos) diferente;
2. Os alimentos serão colocados na tela, em grupos de 3, e em uma determinada ordem (aleatória mas memorizada). O tamanho dos grupos vai crescendo ao longo do jogo (o céu é o limite);
3. A distribuição dos grupos de alimentos deve ser feita de forma que o usuário possa perceber a ordem em que eles foram colocados;
4. Quando a cobra comer um alimento, um novo segmento é incorporado ao seu corpo (mantendo a cor do alimento);
5. Caso o usuário consiga comer os alimentos em uma certa ordem, exatamente a ordem inversa ao que eles foram colocados na tela, a sequencia equivalente de segmentos criados na cobra é removida do seu corpo. Caso a ordem não seja respeitada, os segmentos ficam integrados ao corpo da cobra até o final do jogo;
6. A cobra não morre ao tocar nas bordas da tela, mas passa para o outro lado da tela (diferente do jogo original).

## A Implementação:

A implementação deve ser desenvolvida em linguagem C ANSI (independente de qualquer IDE ou SO) e tomar por base o código fonte base fornecido pelo professor. Esse código fonte está dividido em alguns módulos, a saber:

- **EstruturasDeDados.h**  
Descrição das estruturas de dados a serem utilizadas nesse trabalho;
- **winGL.\***  
Rotinas responsáveis pelo controle das janelas e dos desenhos;
- **trabalho.\***  
Programa principal e rotinas de tratamento de eventos de teclado e desenho.

As rotinas que você deve codificar estarão nos módulos **cobra.\***, **trabalho.\*** e **EstruturaDeDados.h**.

Para que você possa utilizar a biblioteca gráfica **glut**, você deve baixar os arquivos \*.h e **libs** e instala-los/copia-los para os diretórios do seu compilador. Esse processo e o arquivo dessa **lib** para download podem ser encontrados em [4].

Você deverá definir seu TAD baseado na estrutura de dados inicial, definida no módulo **EstruturasDeDados.h**. **Não é permitido modificar essa estrutura. Caso ache necessário, consulte o professor para saber se sua modificação é válida.**

Os trabalhos deverão ser desenvolvidos individualmente. O código fonte gerado deve ser comentado e legível. Acompanhando o código fonte um breve relatório técnico deve ser entregue, descrevendo as estruturas de dados utilizadas, uma justificativa para seu uso e quais testes foram realizados para validar o funcionamento do programa (descrição, objetivo e resultado).

#### **A Entrega:**

O trabalho deverá ser submetido via *Moodle*, respeitando a data e hora limite para entrega. Em caso de atraso, será aplicado um fator de penalização de 1,0 ponto por dia de atraso. Qualquer problema de arquivos corrompidos ou similar o trabalho será considerado não entregue. Portanto, verifique bem o que for entregar.

Os arquivos devem ser enviados seguindo o seguinte padrão: arquivo compactado (zip, rar, tgz ou gzip apenas) contendo um diretório com o nome do(s) aluno(s) e seu arquivos. Arquivos fora desse padrão sofrerão penalização de 0,5 ponto na nota final. Códigos com erros de compilação ou qualquer outra pendência que impeça a compilação não serão avaliados.

Uma aula será destinada a apresentação dos trabalhos, onde cada aluno mostrará ao professor o que foi feito.

A cooperação entre alunos e grupos é considerada salutar. No entanto, trabalhos com alto grau de similaridade serão tratados como “plágio”, o que resultará em avaliação **zero** para todos os envolvidos.

Qualquer dúvida adicional, evite problemas: não presuma nada, procure o professor para esclarecimentos.

#### **Referencias Bibliográficas:**

- [1] Ziviani, Nivio. **Projeto de Algoritmos: com Implementações em Pascal e C**. Vol. 2. Thomson, 2004.
- [2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. **Introdução a algoritmos**. 2001.
- [3] Sedgewick, Robert. **Algorithms in C++**. Vol. 2. Pearson Education India, 2003.
- [4] Martin Payne, *Using freeglut or GLUT with MinGW*, seção "**Setting Up GLUT for Win32 With MinGW**", disponível em:  
<http://www.transmissionzero.co.uk/computing/using-glut-with-mingw/>.
- [5] Wikipedia. **Snake (Video Game)**. disponível em:  
[https://en.wikipedia.org/wiki/Snake\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Snake_(video_game))