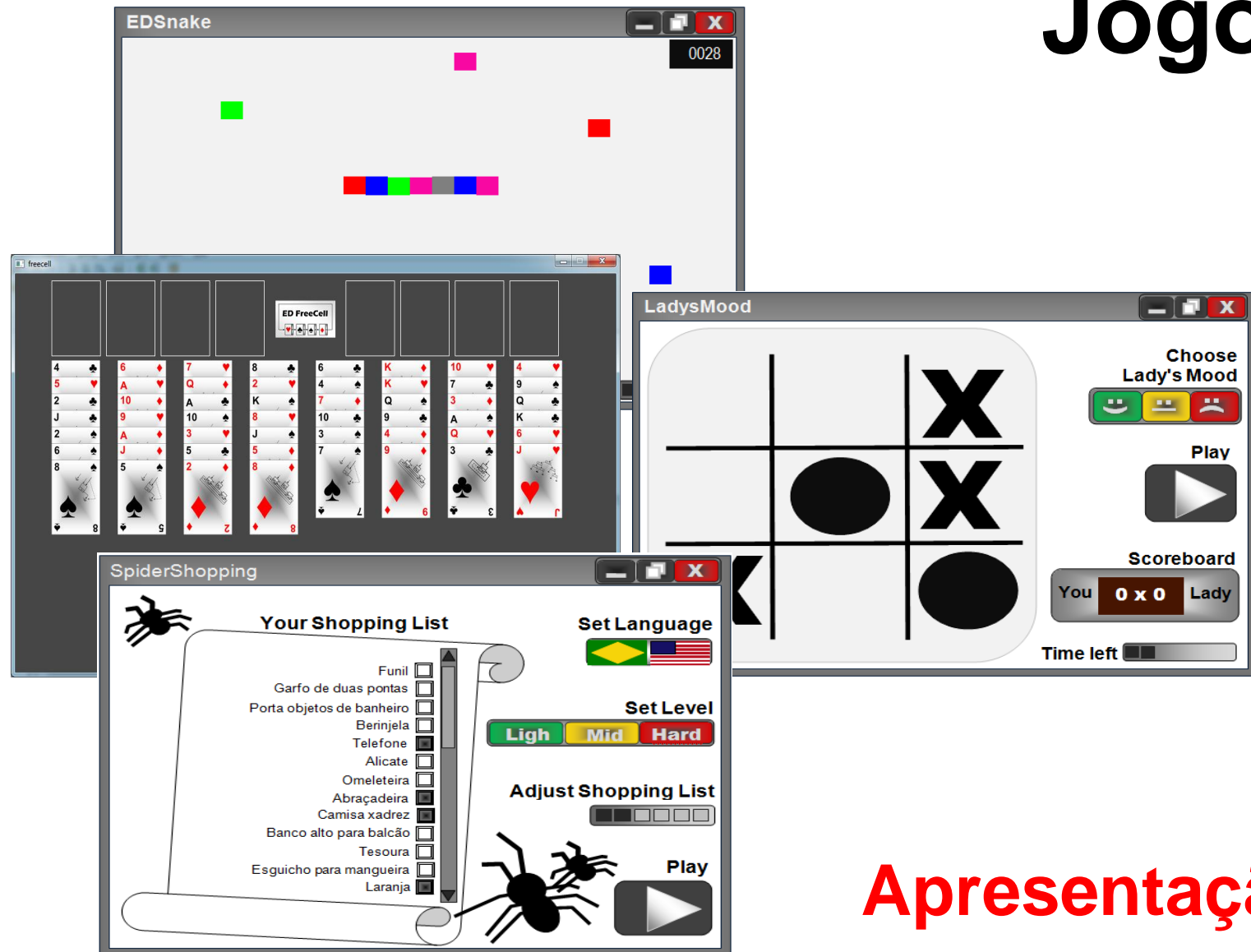


Estruturas de Dados com Jogos



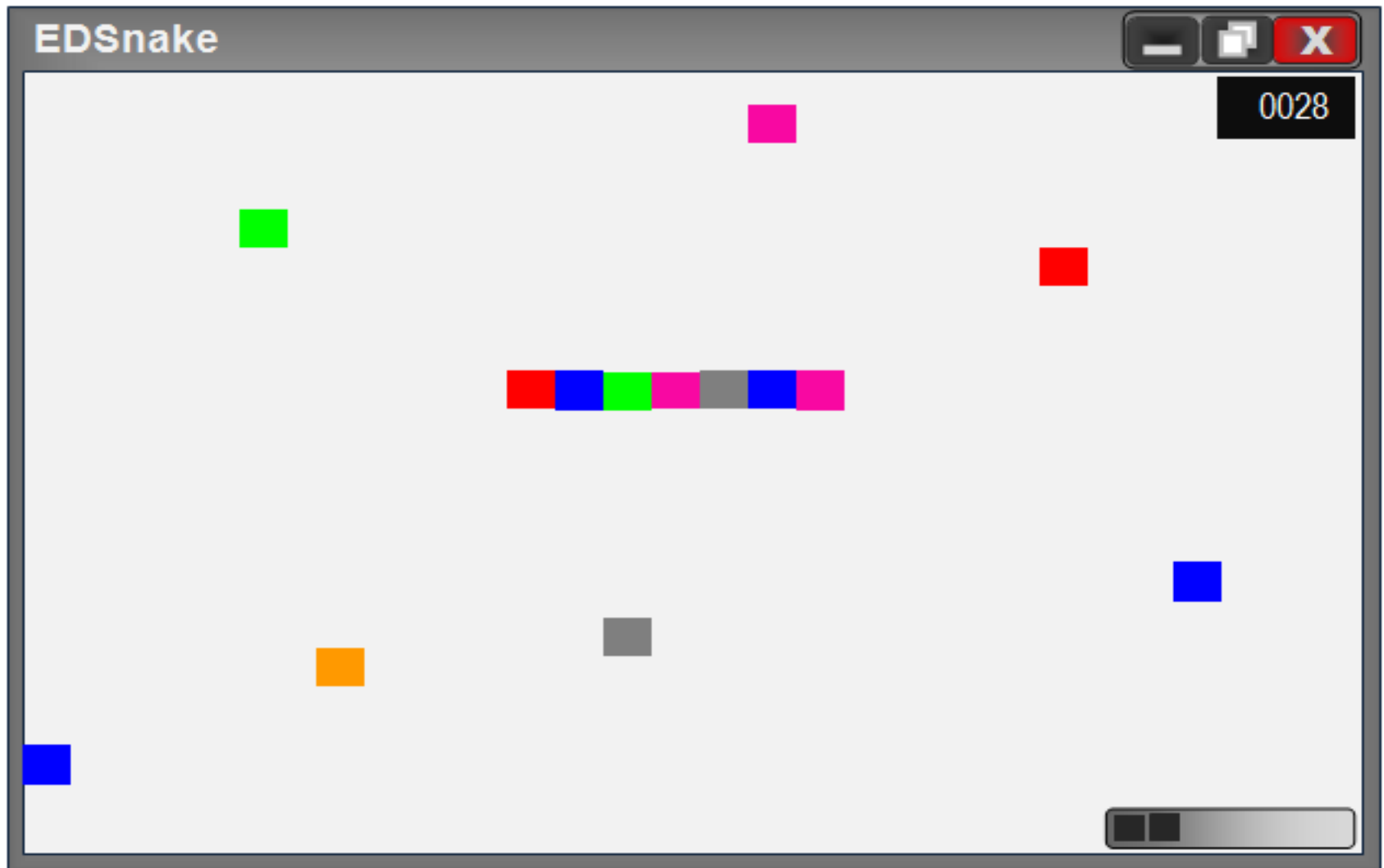
Apresentação

FreeCell



Como armazenar as cartas em pilhas, e implementar as regras do jogo?

Snake



Como retirar ou acrescentar os pedacinhos na cobra, e mantê-los na sequencia correta?

Jogo da Velha

LadysMood

		X
	O	X
X		O

Choose Lady's Mood

Play

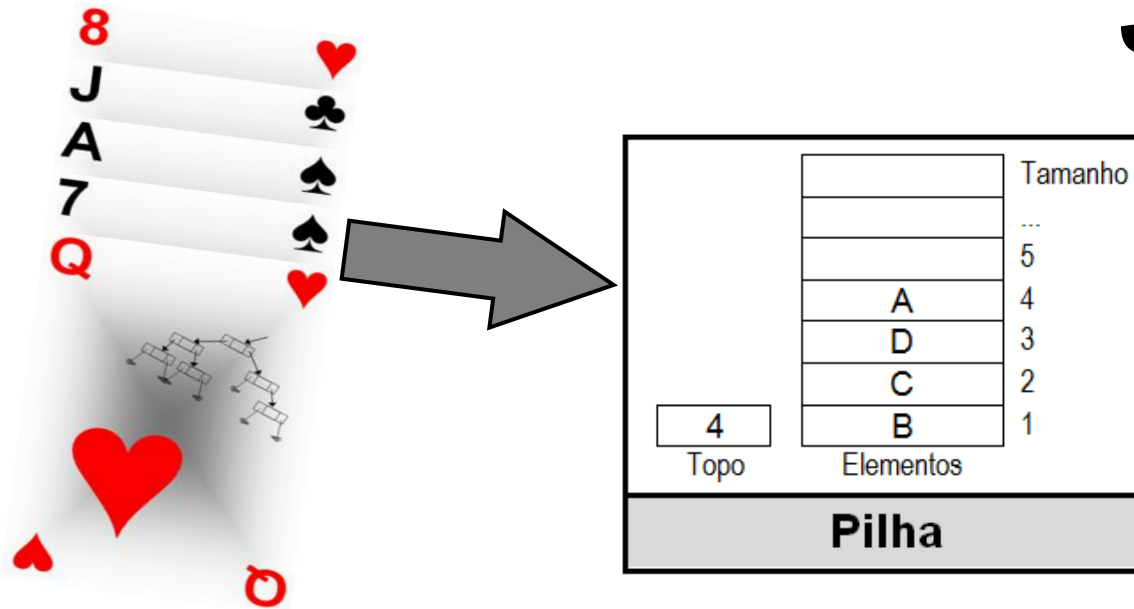
Scoreboard

You **0 x 0** Lady

Time left

Como prever todas as possíveis jogadas, e então escolher a melhor opção?

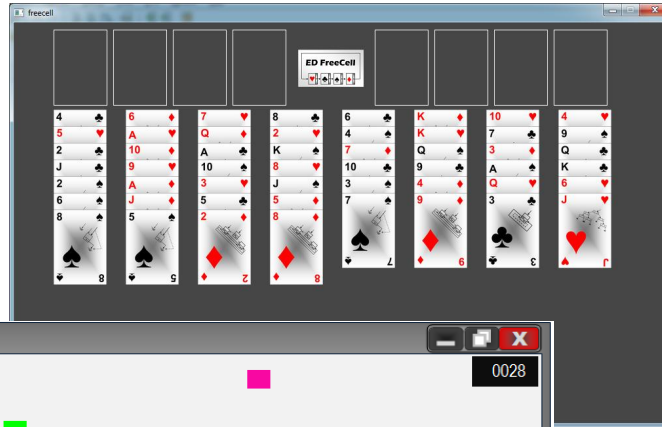
Estruturas de Dados com Jogos



Objetivos

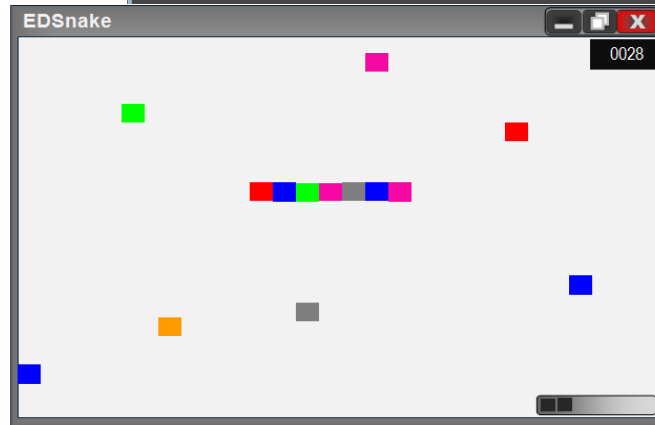
Preparar você para implementar **Estruturas de Dados** para representação e armazenamento de conjuntos de informações em um programa.

Pilhas



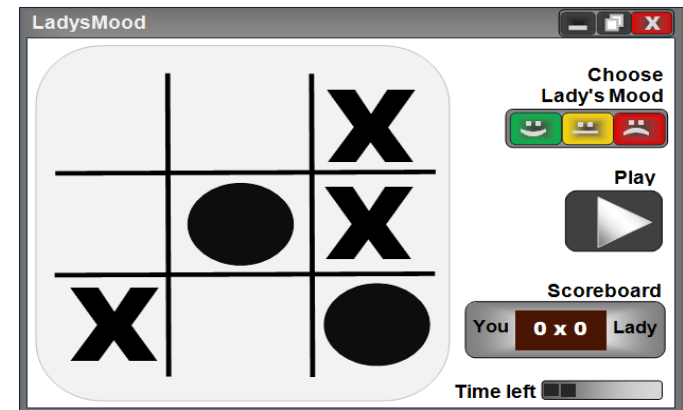
4 Jogos

Filas



4 Desafios

Listas



Árvores



Parte I - Pilhas

Desafio 1: Desenvolver uma Adaptação do Jogo *FreeCell*

Capítulo 1: Tipos Abstratos de Dados

Capítulo 2: Pilhas, com Alocação Sequencial e Estática



Parte II - Filas

Desafio 2: Desenvolver uma Adaptação do Jogo *Snake*

Capítulo 3: Filas, com Alocação Sequencial e Estática

Capítulo 4: Listas Encadeadas

Capítulo 5: Listas Encadeadas com Alocação Dinâmica

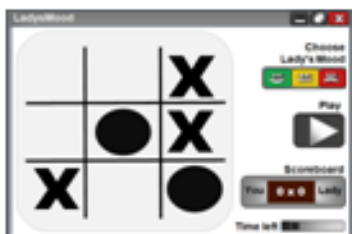


Parte III - Listas Cadastrais

Desafio 3: Desenvolver um Jogo que use Listas de Elementos

Capítulo 6: Listas Cadastrais

Capítulo 7: Generalização de Listas Encadeadas







Parte IV - Árvores

Desafio 4: Desenvolver um Game com Previsão de Jogadas

Capítulo 8: Árvores

Capítulo 9: Árvores Balanceadas

Materiais Complementares

			
Videos	Animações	Tutorial de Programação Gráfica	Banco de Jogos

<http://edcomjogos.dc.ufscar.br>

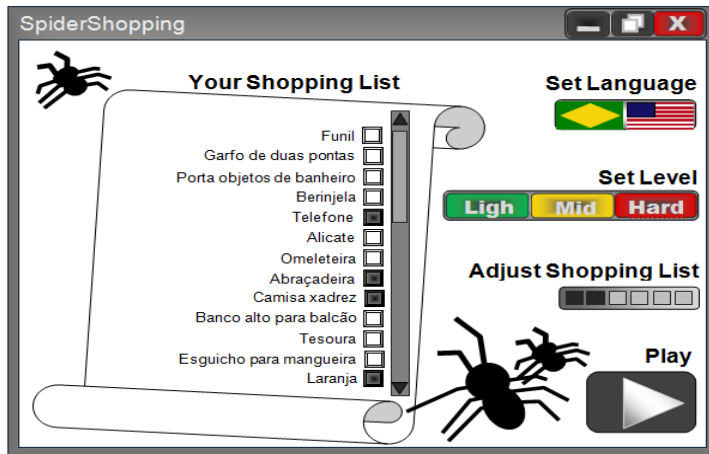
Notação Conceitual

```
PegaOPrimeiro( L1, X, TemElemento );  
  
Enquanto TemElemento == Verdadeiro  
Faça  
    {  
        Se EstaNaLista(L2, X)  
        Então Insere (L3, X, Ok);  
        PegaOPróximo( L1, X,  
TemElemento );  
    };
```

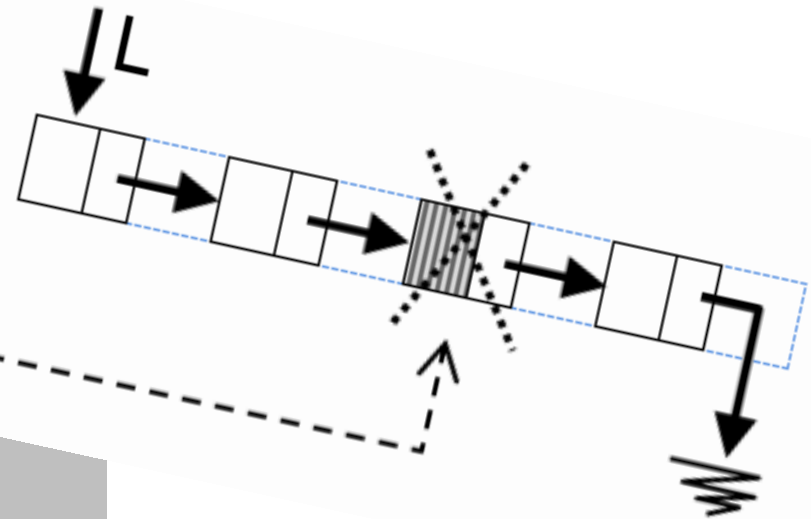
Códigos em C e em C++

```
struct Node {  
    char Info;  
    struct Node *Next; };  
typedef struct Node *NodePtr;  
NodePtr P;  
P = new Node;
```

Chave para um Bom Aproveitamento: Praticar!

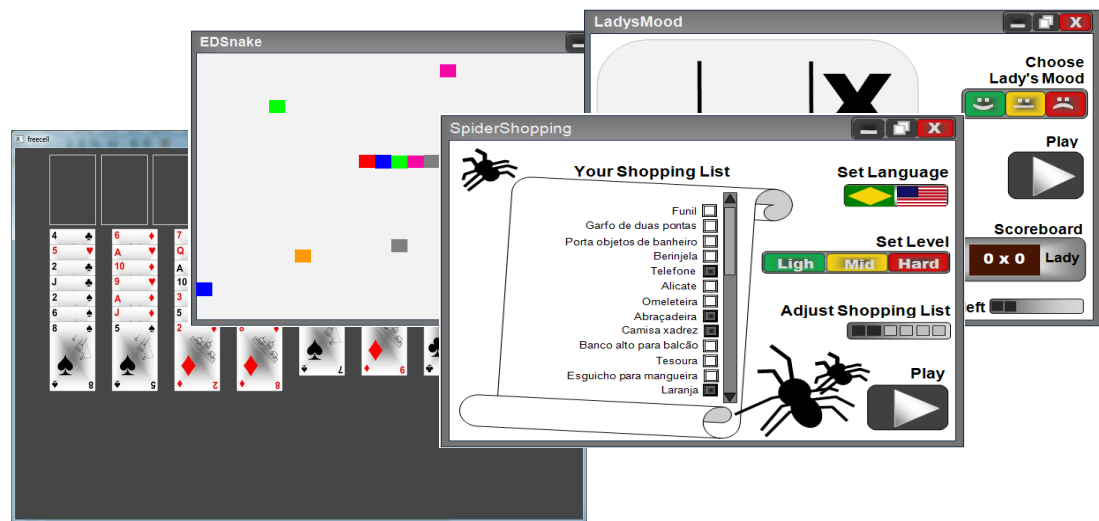


Xxxxxx XX xxx
yYY yyy YY
xXwz Ywx xXwz
ZZx XwX Zzz



```
struct Node {  
    char Info;  
    struct Node *Next; };  
typedef struct Node *NodePtr;  
NodePtr P;  
P = new Node;
```

Encare o Desafio!



Desenvolva jogos divertidos! Projete uma interface legal! Mostre para seus amigos! Divulgue seus Jogos! Participe de uma competição de jogos! Faça seus jogos bombar!

Aprenda pra valer! Envolve seus colegas; cresça junto com eles!

Aprender a programar pode ser divertido!

Estruturas de Dados com Jogos

Aprender a programar pode ser divertido!