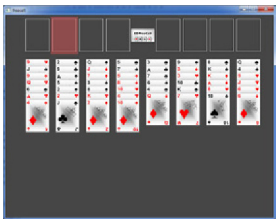
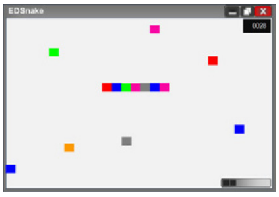
Apresentação

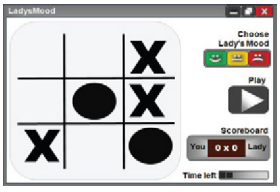


Conhece um jogo chamado ***FreeCell?*** É um jogo de paciência, muito conhecido, em que você precisa manipular as cartas de um baralho, com o objetivo de colocá-las em sequência. As cartas são armazenadas em Pilhas e só podem ser movimentadas de acordo com regras bem específicas. Como você desenvolveria um jogo como o FreeCell? Como faria para armazenar as cartas em Pilhas e garantir que elas sejam movimentadas de acordo com as regras do jogo?

E jogos do tipo ***Snake***, você já jogou? O personagem principal desse jogo é uma cobra composta por diversos pedacinhos coloridos. Ao se movimentar e comer alguma coisa, a cobra vai ganhando ou perdendo pedacinhos, crescendo ou diminuindo, e tornando o jogo cada vez mais difícil. É um jogo de habilidade. Se você fosse desenvolver um jogo do tipo *Snake*, como representaria no programa a cobra e cada um de seus pedacinhos? Como faria para retirar ou acrescentar os pedacinhos e para mantê-los na sequência correta?



Pense agora em um jogador humano disputando contra o computador. O jogo em si pode ser simples, como um ***Jogo da Velha***, por exemplo, mas não um jogo da velha qualquer; um jogo inteligente, no qual o computador escolhe a melhor jogada em função da jogada do adversário, sempre visando situações com as maiores chances de vitória. Como você implementaria essa inteligência em um jogo? Como faria o seu jogo prever todas as possíveis jogadas e depois escolher, conscientemente, a melhor opção?



Objetivos do livro

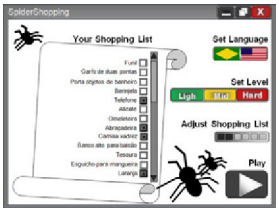
O livro *Estruturas de dados com jogos* tem por objetivo prepará-lo para implementar estruturas de dados para representação e armazenamento de conjuntos de informações em um programa. Conjuntos de informações como uma pilha de cartas, uma fila de espera, uma lista de passageiros ou de compras, por exemplo.

O foco do livro não são os jogos em si. A ideia é que você aprenda estruturas de dados — desenvolva habilidades sólidas de programação, enquanto cria alguns jogos. Os jogos tornarão seu crescimento mais divertido!

Os jogos e os capítulos

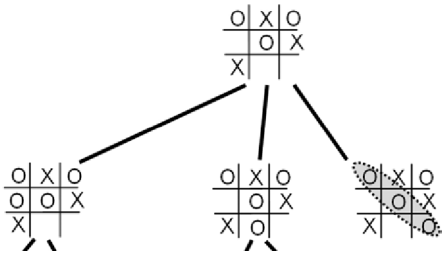
São quatro jogos para você desenvolver. Quatro desafios. O primeiro desafio é desenvolver uma adaptação do *FreeCell*. Para desenvolver um bom jogo, você estudará o Capítulo 1 e o Capítulo 2 de *Estruturas de dados com jogos*. Você vai conhecer uma estrutura de dados chamada **Pilha**. Você vai aprender como implementar uma Pilha de Cartas e como deixar seus jogos bem flexíveis. Assim será fácil ajustar o software do jogo a novas regras e a novas situações.

Seu segundo desafio é desenvolver um jogo do tipo *Snake*. Enquanto desenvolve seu próprio *Snake*, você estudará os Capítulos 3, 4 e 5, e conhecerá outra estrutura de dados chamada **Fila**. Você perceberá que uma Fila é muito útil para implementar um jogo como o *Snake*. Você irá comparar duas técnicas de implementação e escolher uma dessas técnicas para implementar sua própria Fila e seu próprio *Snake*.



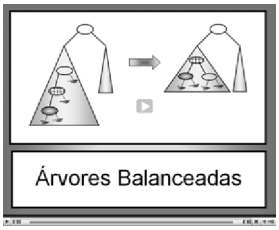
No terceiro desafio, você será apresentado a um jogo chamado ***Spider Shopping***. Seu objetivo será desenvolver uma adaptação desse jogo ou um jogo diferente que mantenha algumas das características do *Spider Shopping*. Para isso, estudará outro tipo de estrutura de dados: a **Lista Cadastral**. Nos Capítulos 6 e 7, você estudará diversas técnicas para implementar uma Lista Cadastral e usará sua criatividade para conceber seu próprio jogo — sua própria aplicação das Listas Cadastrais.

O ponto-chave do quarto desafio é construir a inteligência do jogo. Poderá ser o ***Jogo da Velha*** ou outro jogo, mas o jogo precisará prever as próximas jogadas, para então optar pela alternativa que ofereça maior chance de vitória. Estudando os Capítulos 8 e 9, você conhecerá uma estrutura de dados chamada **Árvore** e algumas de suas aplicações. Você perceberá que com uma Árvore poderá implementar a **Previsão de Jogadas** e a inteligência do seu jogo.

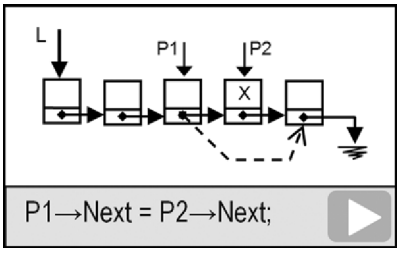


Nos quatro desafios, você terá a oportunidade de adaptar o jogo sugerido e criar seu próprio jogo: um jogo com personalidade própria; um jogo com a sua cara! Você poderá até propor jogos totalmente novos. Mas, para que você desenvolva as habilidades pretendidas, os jogos terão que manter algumas das características de cada desafio. Em essência, os quatro jogos precisarão ser aplicações de quatro estruturas de dados fundamentais que estudaremos ao longo deste livro: Pilhas, Filas, Listas Cadastrais e Árvores.

Materiais Complementares

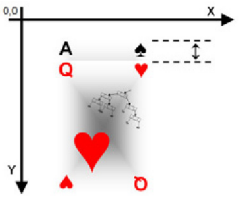


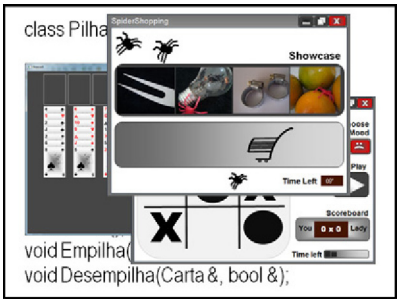
Para facilitar sua compreensão de alguns dos conceitos e algoritmos, você poderá assistir a **vídeos** com explicações e exemplos. Além dos vídeos, os Materiais Complementares de *Estruturas de dados com jogos* contêm algumas **animações**. As animações lhe mostrarão graficamente a execução de um trecho de programa.



Você poderá interagir com as animações, avançando a execução do programa passo a passo enquanto observa uma representação visual do que está acontecendo. O papel dos vídeos e das animações é complementar a leitura, mas nunca substituí-la. Estude o texto, faça os exercícios e complemente seu estudo com os vídeos e animações dos Materiais Complementares.

Se você ainda não tiver prática em programação utilizando interfaces gráficas, poderá consultar nos Materiais Complementares um **Tutorial de Programação Gráfica**. Passo a passo, o tutorial lhe ajudará a instalar uma biblioteca gráfica e a desenvolver seu primeiro jogo. Estude o tutorial em paralelo ao estudo das estruturas de dados. Coloque um foco em seu estudo: viabilizar o desenvolvimento de seus quatro jogos. Será mais produtivo e mais divertido aprender assim. A biblioteca gráfica utilizada no tutorial é apenas uma sugestão; você poderá utilizar a ferramenta gráfica de sua preferência.





Você encontrará também, nos Materiais Complementares, um **Banco de Jogos**. São **jogos-exemplo**, desenvolvidos por pessoas como você, que aceitaram o desafio de aprender estruturas de dados de um modo bem divertido. E que tal **você** desenvolver jogos legais e disponibilizá-los nesse banco de jogos? Que tal tornar seus jogos públicos e conhecidos? Que tal participar de uma **competição de jogos**?

Acesso aos Materiais Complementares

Os Materiais Complementares estarão à sua disposição a qualquer tempo. Consulte-os quando quiser a partir do link a seguir:

http://www.elsevier.com.br/edcomjogos

Você também encontrará, ao longo do texto, **ícones** representando os **vídeos**, as **animações**, o **Tutorial de Programação Gráfica** e os **jogos-exemplo**. Quando encontrar um desses ícones, significa que existem materiais complementares pertinentes ao assunto que você está estudando e que esse será um bom momento para você consultar esses materiais, caso desejar. Dependendo de qual versão do livro estiver utilizando, você poderá inclusive acessar os Materiais Complementares clicando nos ícones ou fazendo a leitura do *QR-Code* associado ao link.

Notação conceitual

Os algoritmos que estudaremos serão apresentados e discutidos em **notação conceitual**, sem construções de uma linguagem de programação específica. Isso permite que os conceitos sejam compreendidos em sua essência e implementados em qualquer linguagem. Para exemplificar a implementação dos conceitos em uma linguagem de programação, em alguns momentos-chave são apresentados **códigos em C e em C++.** Mas é perfeitamente possível utilizar este livro como referência e implementar os algoritmos em qualquer outra linguagem de programação.

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

A chave para um bom aproveitamento: praticar

Não tenha como meta apenas *conhecer* ou *entender.* Isso não é suficiente. Tenha como meta *desenvolver habilidades* para projetar estruturas de armazenamento de dados, para implementar essas estruturas e para utilizá-las na prática, seja no mundo dos games, seja em outro contexto. E para realmente desenvolver essas habilidades, você precisará fazer uma coisa: praticar.

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

Pratique! Pratique muito! Você encontrará muitos exercícios neste livro e soluções para boa parte deles. Faça os exercícios! Proponha uma primeira solução, erre, faça de novo, erre menos, faça de novo e então acerte! Só consulte as soluções fornecidas após propor sua própria solução ou, pelo menos, após tentar exaustivamente.

Encare o desafio!

Aprenda estruturas de dados. Aprenda pra valer! Em paralelo a isso, desenvolva seus jogos! Dê personalidade própria a eles! Torne-os divertidos! Projete uma interface legal! Mostre os jogos para seus amigos! Participe de uma competição de jogos! Envolva seus colegas; desenvolva jogos junto com eles; cresça junto com eles! Divulgue seus jogos na internet! Faça seus jogos bombar!

Aprender a programar pode ser divertido!

http://www.elsevier.com.br/edcomjogos