

## Unidade 4

# Especificação do Trabalho 1

### **Nossos Objetivos nesta Unidade:**

- Conhecer a especificação do primeiro trabalho prático da disciplina. Saber como os trabalhos serão avaliados.

### **O Que fazer?**

Implementar o jogo: “Jardim das Flores\*\*”

### **Faça em grupo**

Faça esse trabalho preferencialmente em grupo. Você pode se organizar com os colegas, espontaneamente, em um grupo de até 4 pessoas. Vocês definem os grupos. A única restrição: precisam ser todos da mesma sala. Use o fórum livre da sua sala para negociar seu grupo com os colegas e, a partir de então, decidam a melhor forma de interagir e trabalhar em grupo. Informe o nome de todos os componentes do grupo ao propor a estratégia de desenvolvimento, e também ao entregar o trabalho.

Se por algum motivo você fizer questão de fazer sozinho, tudo bem, não é proibido. Mas, para os propósitos deste trabalho na disciplina, é melhor fazer em grupo.

Trabalhar em grupo não significa 4 pessoas olhando para o mesmo computador. Significa discutir com o grupo a melhor estratégia, dividir o trabalho, trabalhar de forma dividida porém integrada. Todos devem estar plenamente conscientes de como foi feito cada pedaço do trabalho.

Trabalhar em grupo significa também aprender uns com os outros. Receber ajuda, e fornecer ajuda.

### **Como fazer?**

Primeiramente discuta a estratégia de implementação com os colegas, tutores e professor. Há um fórum sobre isso no ambiente de interação. Procure se certificar de que você está escolhendo uma boa estratégia. Ao descrever a estratégia, defina os módulos do programa, as estruturas de armazenamento, se possível a divisão do trabalho. Faça um diagrama, se achar relevante. Mas seja bem objetivo. Usar mais do que uma página para descrever a estratégia de implementação, seria exagero.

## **Linguagem**

Implemente em C, C++ ou Pascal. Qualquer compilador dessas linguagens serve. Pode usar bibliotecas gráficas sim, naturalmente. Mas não necessariamente.

Não use bibliotecas prontas para estruturas de dados que você está aprendendo a implementar na disciplina (pilhas, filas, listas, etc.). Desenvolva você mesmo operações para manipular essas estruturas.

Procure utilizar estruturas de programação simples, ou seja, procure não usar artifícios muito peculiares e específicos de uma determinada linguagem de programação.

## **A interface gráfica é importante?**

Para os propósitos da disciplina, o mais importante é que você escolha uma boa estratégia de implementação, experimente, e pratique o desenvolvimento dessa forma. Você terá que implementar uma pilha de cartas, por exemplo. Como você vai manipular essa pilha é o mais importante.

Se você não souber utilizar uma interface gráfica, e não se interessar por aprender nesse momento, você pode implementar o jogo com uma interface mais simples, com as teclas de setas e com a tecla Enter, ou mesmo com uma interface de comandos textuais ou numéricos. Mas é óbvio que com uma interface visual o jogo fica muito mais legal de se jogar. É uma boa oportunidade para aprender a trabalhar com uma biblioteca gráfica, mouse, etc. Mas terá que aprender por conta própria, ou com os colegas. O propósito dessa disciplina não é ensinar uma linguagem de programação específica, nem biblioteca gráfica.

Uma alternativa interessante é um dos membros do grupo se dedicar mais intensamente à interface, ensinando os demais o que aprendeu, naturalmente.

Resumindo, com uma interface visual naturalmente vai ser melhor, vai ser até mais motivador para você. Mas, enfim, não é necessário. Pode implementar com uma interface qualquer.

Mas projete um bom jogo; crie seu próprio jogo; distribua aos seus amigos!

## **O que entregar?**

Código fonte, executável e documentação descrevendo a estratégia de implementação, e contendo também print-screens da execução de alguns testes.

Procure fazer um pacote de arquivos que possibilite aos professores e aos colegas descarregar e executar o seu jogo, sem a necessidade de configurar nada. Você sabe para onde mover um determinado arquivo que está faltando. Mas só você sabe disso. Por isso, facilite as coisas para quem vai descarregar e testar o seu jogo.

Ao descrever a estratégia de implementação, faça um desenho! Sim, um diagrama mostrando os módulos que utilizou, por exemplo. Se você projetou uma variação do jogo (ou seja, um jogo um pouco diferente do que encontramos nos nossos computadores), descreva bem qual é o seu jogo.

Inclua nas conclusões respostas para as perguntas: (a) Como foi a divisão do trabalho? (b) Qual foi a primeira estratégia escolhida para implementar o jogo? (c) Comparando a estratégia final com a inicial, que vantagens você pode apontar da final sobre a inicial? (d) O trabalho foi útil para você aprender alguma coisa? Que habilidades ele o ajudou a desenvolver?

### **Critério de correção**

Fundamentalmente, podemos ter 3 tipos de resultados:

- **Nível 1: Mínimo:** O aluno se esforçou e procurou elaborar o trabalho segundo as orientações. Avançou, porém não conseguiu concluir, ou não entregou absolutamente tudo o que foi solicitado. Ainda assim, o trabalho pode ser avaliado com uma nota razoável (de 40% a 60%), dependendo do resultado apresentado.
- **Nível 2: Bom:** O aluno elaborou o trabalho segundo as orientações, entregou o que foi solicitado, no prazo, a documentação mostra alguns print-screens, o executável roda, etc. Mas é possível que a estratégia utilizada não seja ótima, é possível que a interface não seja das mais intuitivas, etc. O trabalho será avaliado com uma nota boa (de 60% a 80%).
- **Nível 3: Pra Bater Palmas:** Além de funcionar, a estratégia utilizada foi muito boa, a divisão do trabalho, a documentação, as conclusões, a interface, a criatividade, o empenho, tudo foi excelente, e o resultado é merecedor de aplausos (a nota será de 80% a 100%). Levará nota dez um jogo que rode, com interface legal (não necessariamente gráfica), operável sem a necessidade de leitura de quaisquer orientações, com estratégia de implementação adequada (respeitando conceito de tipos abstratos de dados, etc.), com documentação incluindo descrição do jogo, da implementação, screenshots, estratégia de implementação e conclusões com respostas as perguntas sugeridas.

Faça o seu melhor. Pra bater palmas, se possível. Mas se não for possível, faça o bom. Nunca deixe de fazer o mínimo.

### **Quando entregar?**

Veja no calendário / plano de ensino o prazo para entrega.

### **Onde e como entregar?**

No ambiente de interação.

### **Possíveis problemas**

1- não entregar no prazo; 2- Não entregar documentação/desenho das estruturas propostas; 3- Plágio; 4- Não incluir testes de execução (print screen), quando solicitado; 5- Estratégia de implantação escolhida incompatível com o conceito de tipos abstratos de dados (não usar arquivos separados para módulos independentes; alterar dados armazenados sem ser pelos operadores/métodos definidos; incluir nos métodos de armazenamento funções

que não sejam absolutamente pertinentes ao armazenamento e recuperação); 6- trabalho incompleto ou diferente do especificado; 7- erro nos algoritmos; 8- não entregar código fonte; 9- não entregar executável; 10- Não apresentar em aula (esse problema 10 é válido apenas para as turmas em que a apresentação oral do trabalho estiver prevista no calendário e no plano de ensino da turma).

### Como esse trabalho contribui para os objetivos da disciplina

Objetivo da Disciplina	Contribuição deste Trabalho
(c) Preparar os alunos para projetar e utilizar estruturas de dados através de sua funcionalidade, sem se preocupar com detalhes de implementação.	O critério de correção deste trabalho (que enfatiza a escolha de uma boa estratégia de implementação) e a possibilidade de trabalharem em grupo (divisão do trabalho, etc.) favorece o desenvolvimento de habilidades para projetar e utilizar estruturas de dados através de sua funcionalidade, sem se preocupar com detalhes de implementação.

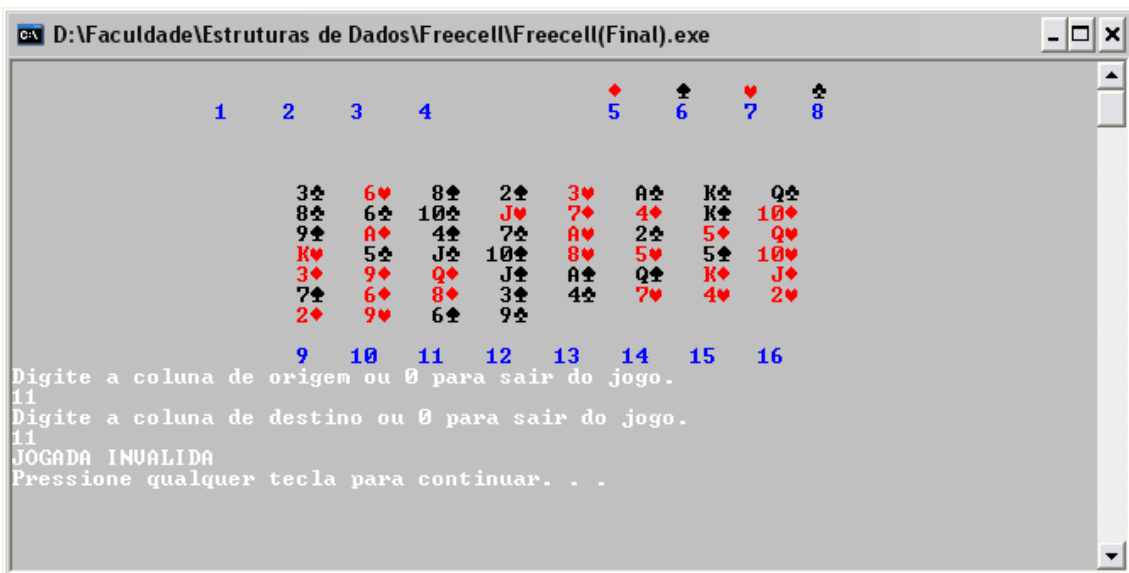
### Anexo 1: Exemplos de Interface



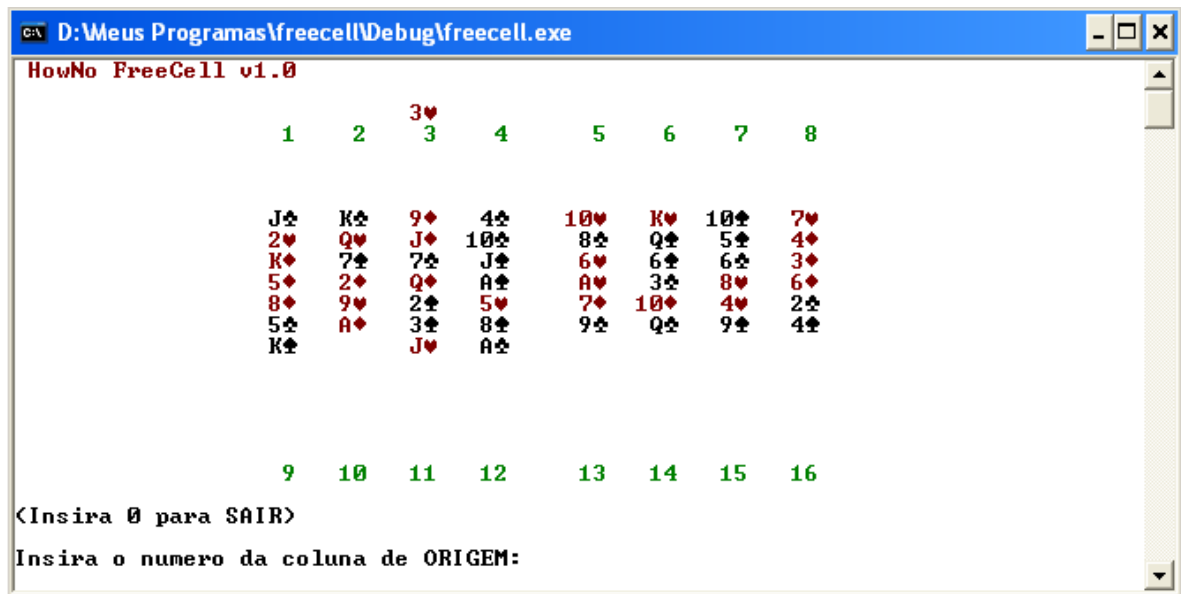
Tela de jogo desenvolvido por André Lourenço, Carlos Santos e Renato Costa, alunos da disciplina Estruturas de Dados, UFSCar, 2008.



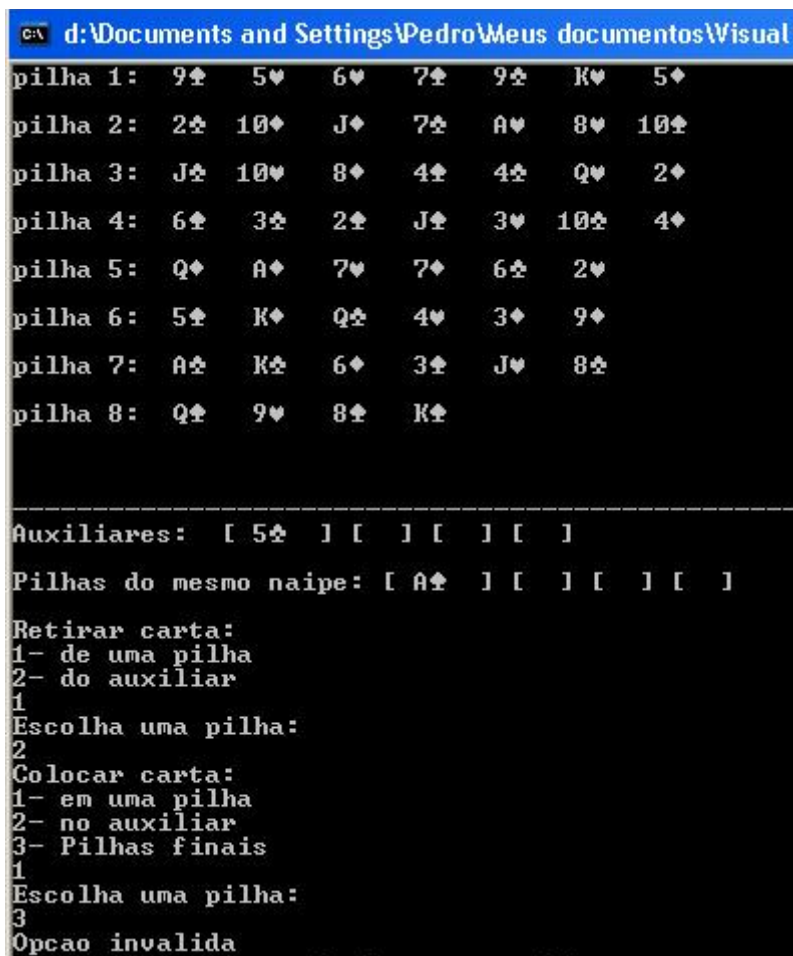
Tela de jogo desenvolvido por André Souza, Guilherme Rabello e Lucas Branisso, alunos da disciplina Estruturas de Dados, UFSCar, 2008.



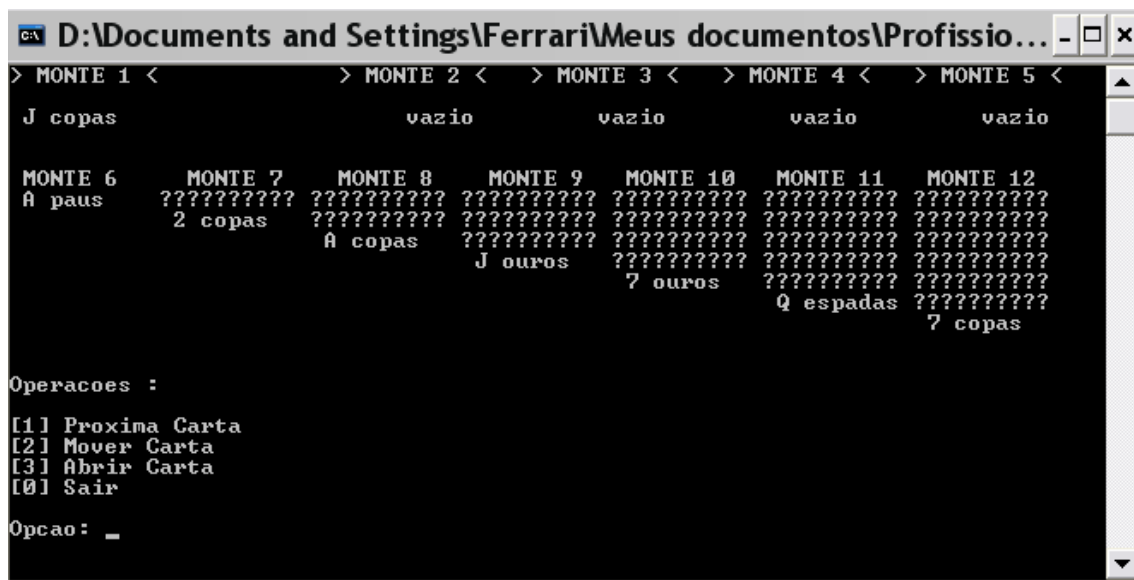
Tela de jogo desenvolvido por Antonio Carvalho Candal e Danilo Amaral de Oliveira, alunos da disciplina Estruturas de Dados, UFSCar, 2008.



HowNo FreeCell, desenvolvido por Flavio Tomazella de Oliveira, Halison Alan Fernandes, Miguel Mêndola Antonio, Wendell Fernandes Carvalho, alunos da disciplina Estruturas de Dados, UFSCar, 2008.

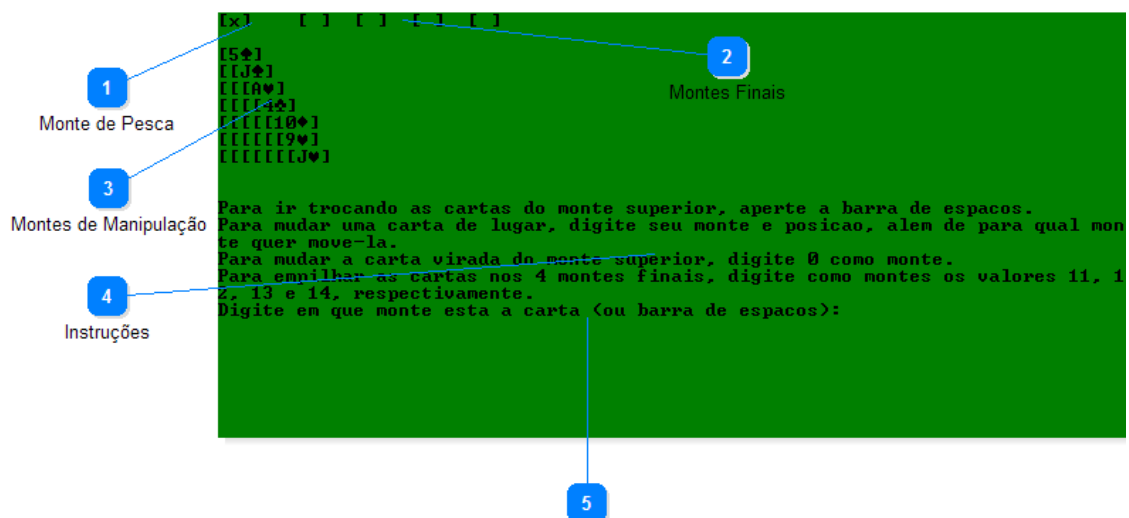


Tela de jogo desenvolvido por Alexandre Osada, Caio Augusto da cruz e Pedro Barabanov de Assis, alunos da disciplina Estruturas de Dados, UFSCar, 2008.

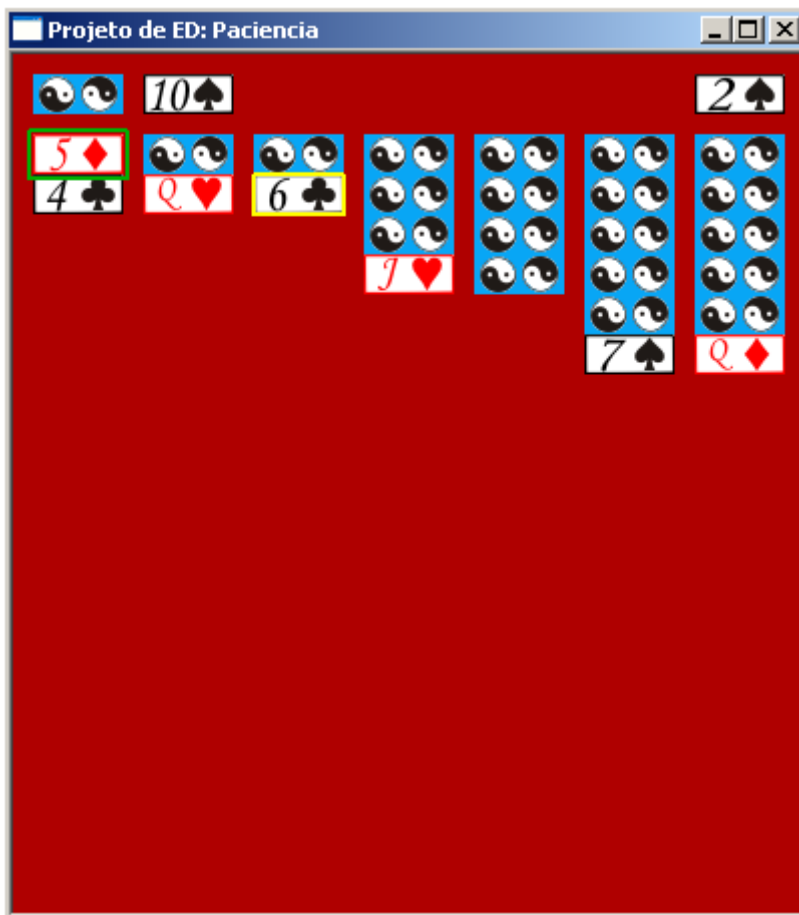


Tela de jogo desenvolvido por Martin Haruo Hori e Caio Luthi Fujii, alunos da disciplina Estruturas de Dados, UFSCar, 2008.

#### Tela inicial e regras

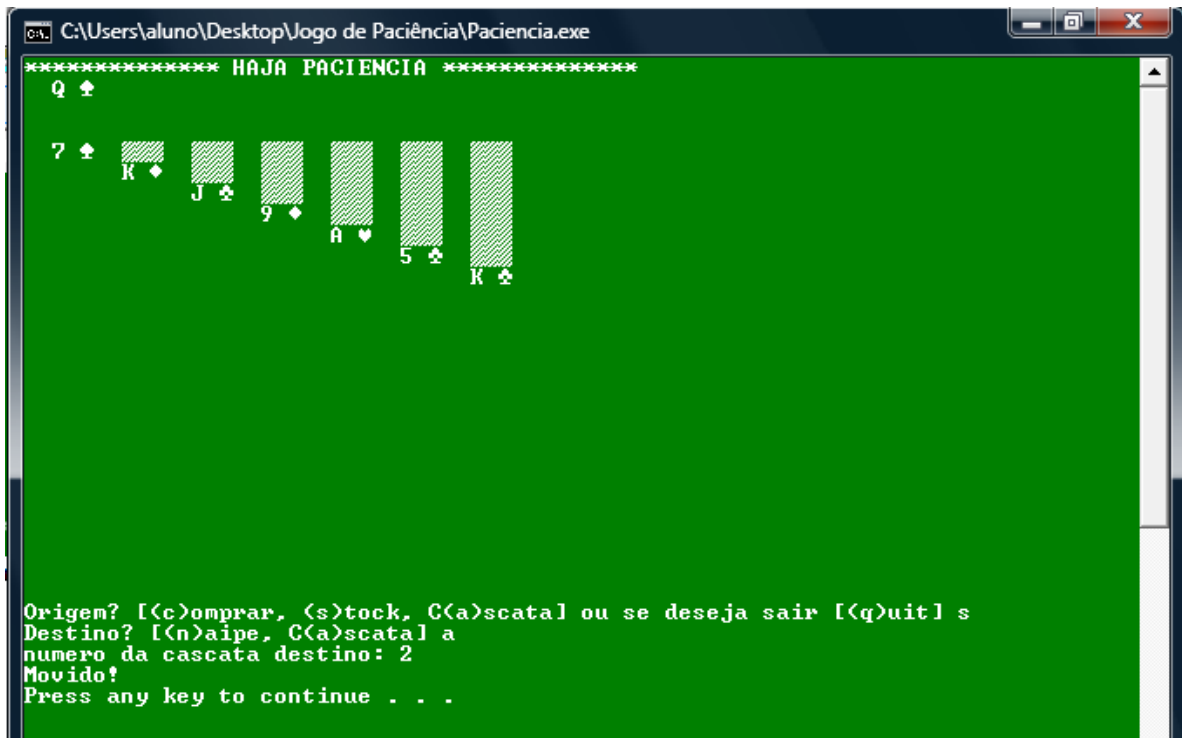


Tutorial de jogo desenvolvido por Carlos Toledo e Paulo Fernando Gimenez de Fávori, alunos da disciplina Estruturas de Dados, UFSCar, 2008.

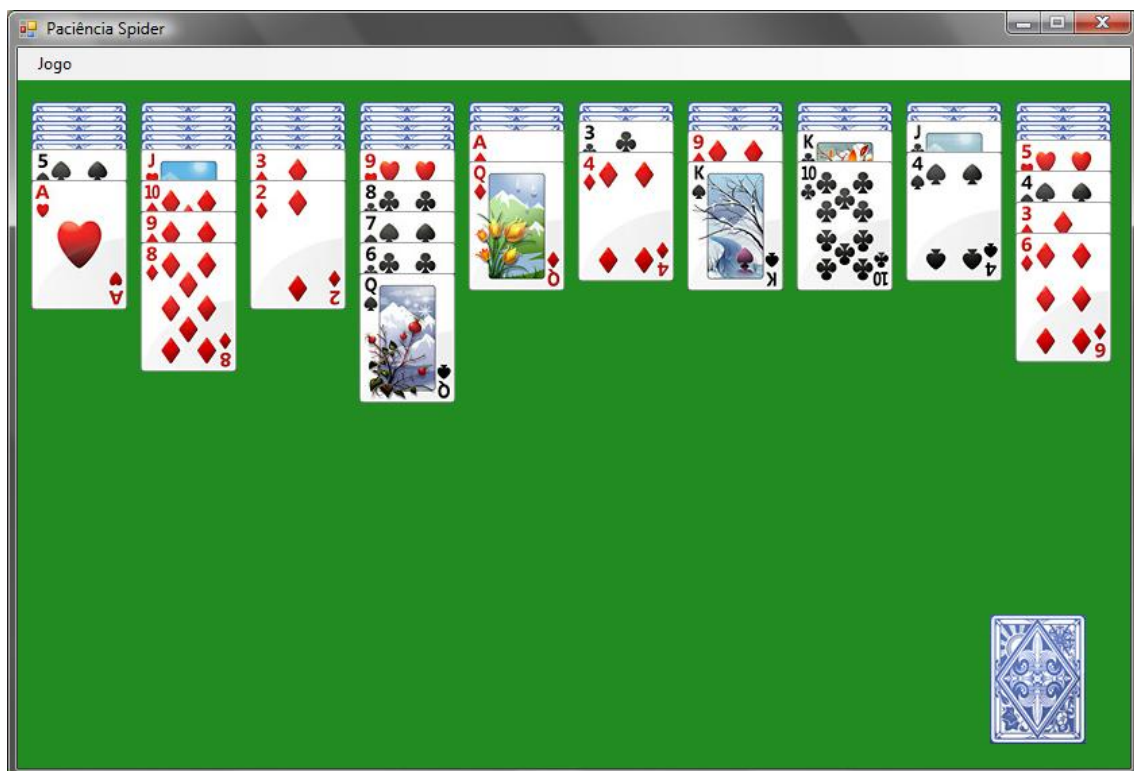


Paciência (interface com setas e enter) desenvolvido por Janito Vaqueiro Ferreira Filho, Thiago de Campos Moraes, Talles Villalva Caceta e Luy Dias Malpighi, alunos da disciplina Estruturas de Dados, UFSCar, 2008.

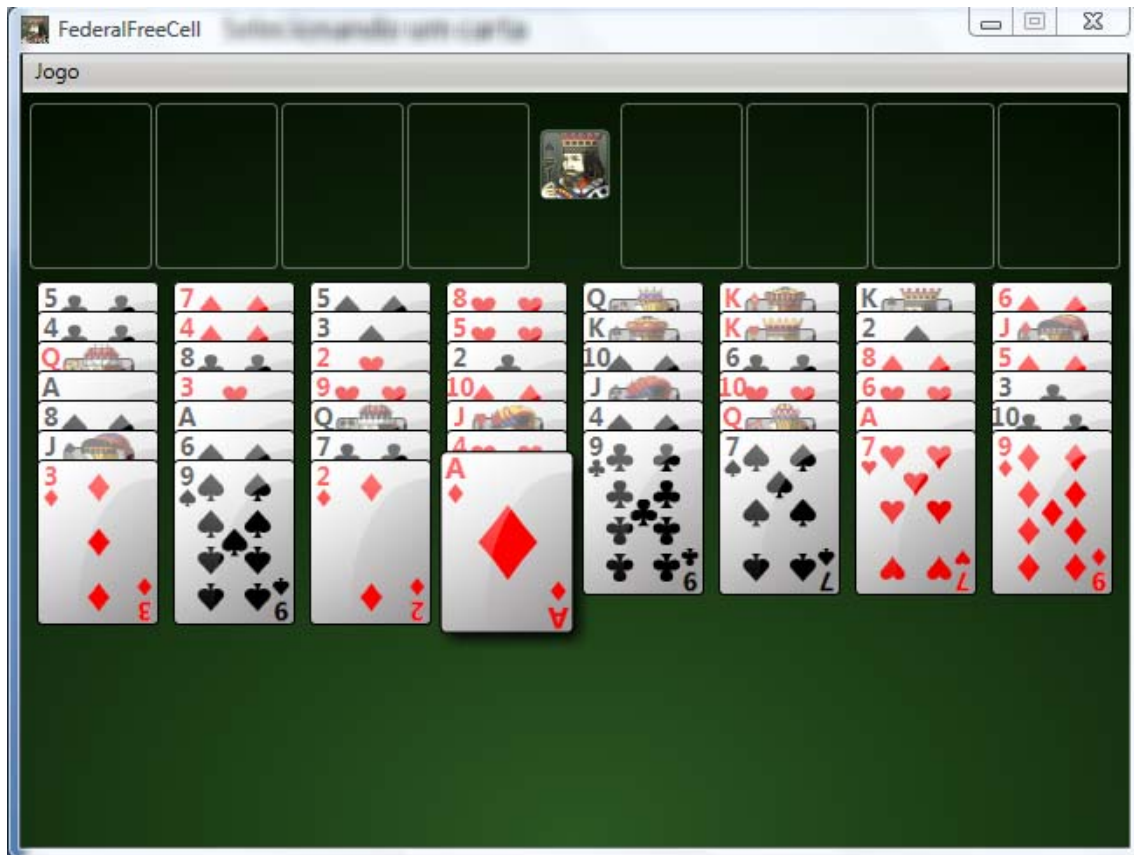




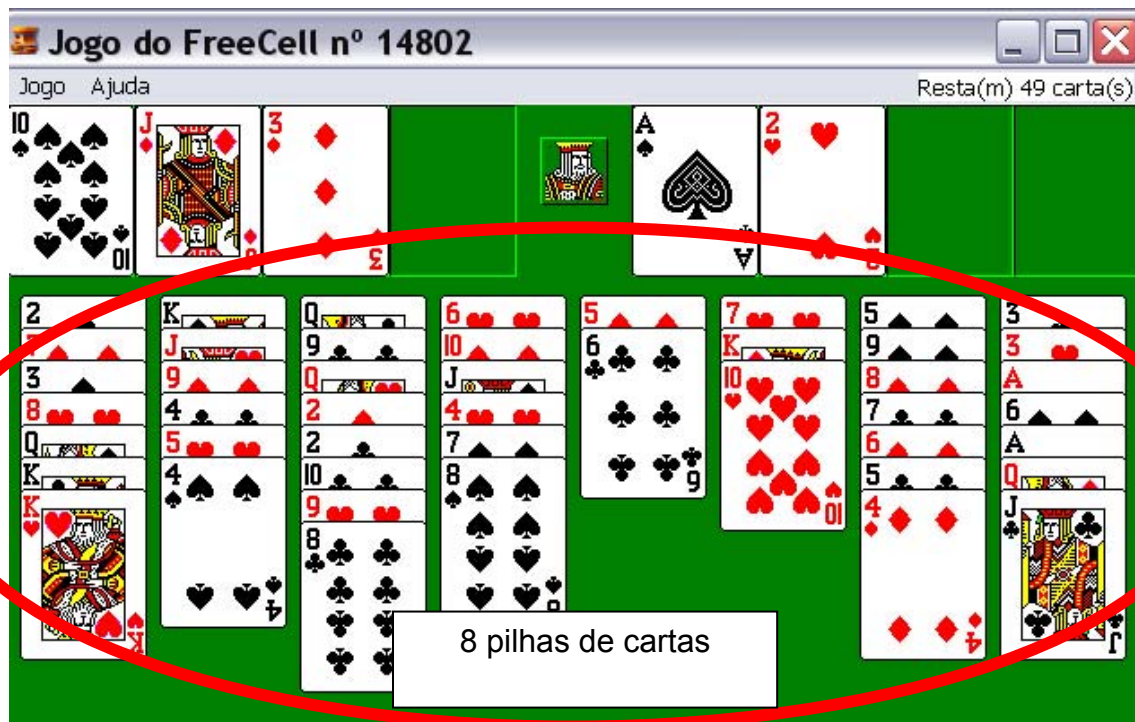
**Haja Paciência – desenvolvido por Guilherme Cartacho Pedroso e Mikio Yokoi Ikeda, alunos da disciplina Estruturas de Dados, UFSCar, 2008.**



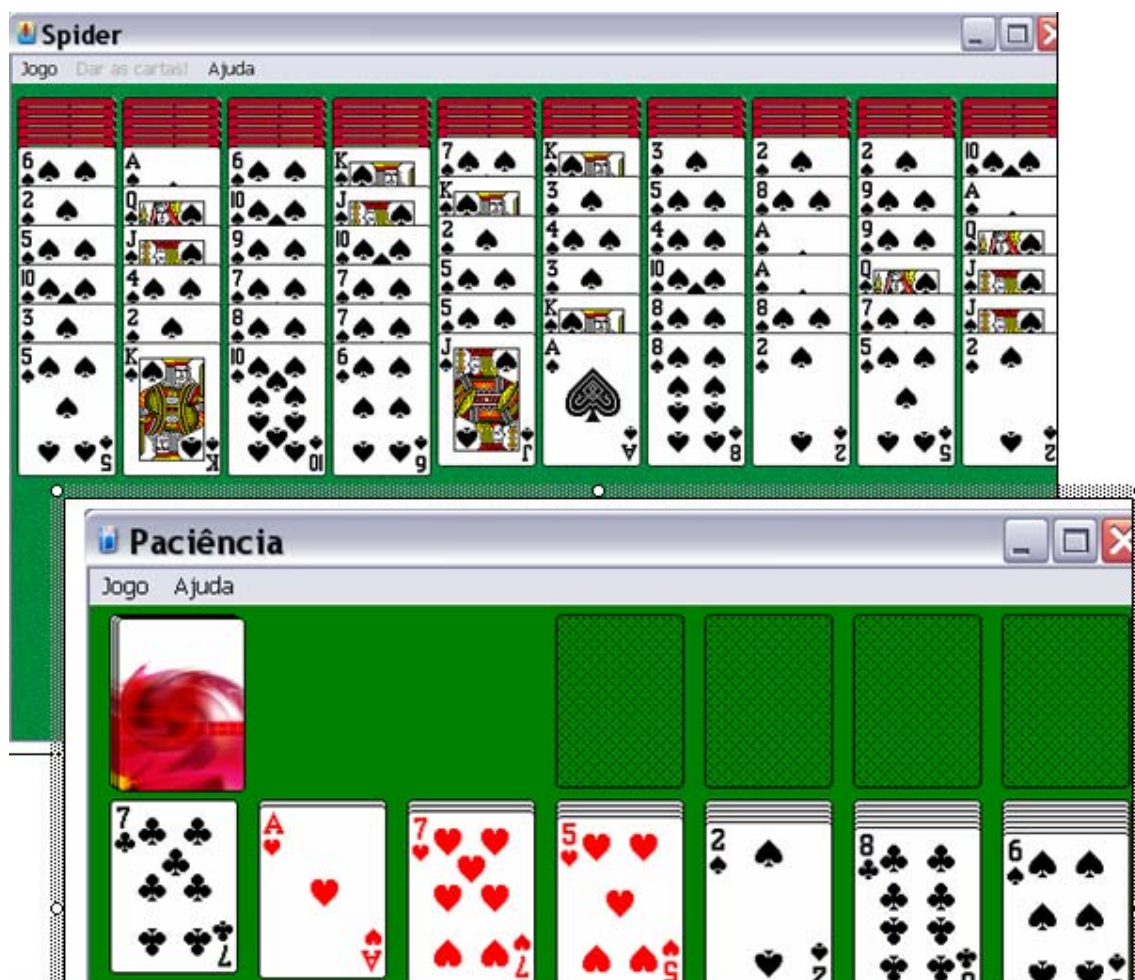
**Spider – desenvolvido por Douglas de Souza Santana, Douglas Oliveira de Freitas, Lucas de Sio Rosa, alunos da disciplina Estruturas de Dados, UFSCar, 2008.**



**Federal FreeCell – desenvolvido por Carlos Eduardo Barbosa, Ronaldo Akio Miyoshi, e Marco Diniz Garcia Gomes, alunos da disciplina Estruturas de Dados, UFSCar, 2008.**



FreeCell (Windows)



Paciência e Spider (fonte: Windows)

## Anexo 2: Tutorial FreeCell

Para vencer o jogo, é preciso empilhar as cartas, classificadas por naipe (copas - ♥, espadas - ♠, paus - ♣, e ouros - ♦), e na sequência (A, 2, 3, 4, 5, 6, 7, 8, 9, 10 J, Q, K), nas quatro pilhas localizadas no canto superior direito.



FreeCell desenvolvido por Aline Manera, Caroline Gracioso, Michel Moura e Tatiane Ferreira, alunos da disciplina Estruturas de Dados, UFSCar, em 2007.

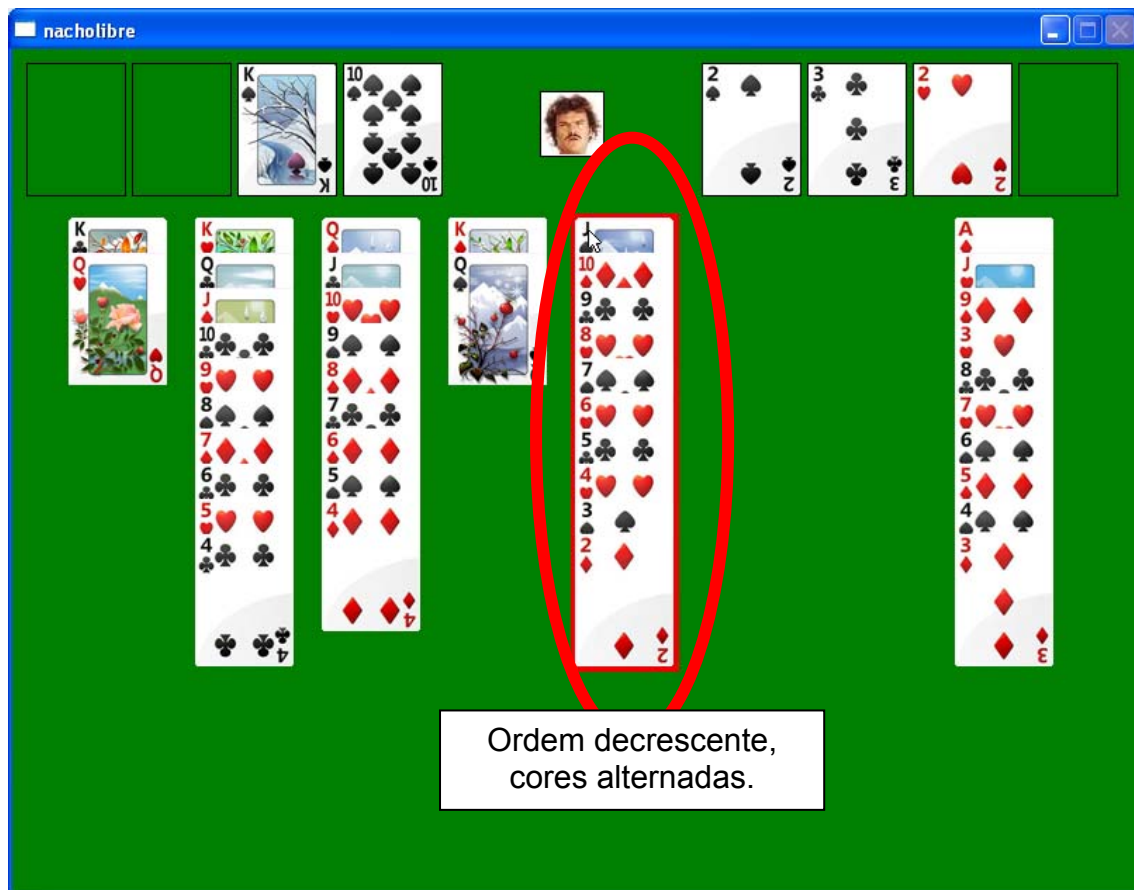


**Tela final do FreeCell desenvolvido por Aline Manera, Caroline Gracioso, Michel Moura e Tatiane Ferreira, alunos da disciplina Estruturas de Dados, UFSCar, em 2007.**

Só é possível retirar a carta que está no topo de uma pilha de cartas. Também não é possível inserir uma carta em uma pilha a não ser no topo da pilha. Os espaços no canto superior esquerdo podem ser utilizados para colocar cartas temporariamente, auxiliando a movimentação.

Nas oito pilhas de cartas ao centro da tela, só é possível acrescentar cartas em ordem decrescente, e com cores alternadas.





**NachoLibre, variação do FreeCell, desenvolvido por Luiz Nogueira, Mauro Anjo e Thiago Campmany, alunos da disciplina Estruturas de Dados, UFSCar, em 2007.**

Se você ainda não conhece o jogo, tente jogar algumas vezes para se familiarizar com as regras. No seu computador deve ter esse jogo.