

Relatório do Projeto Final - MC322: Super Trunfo Virtual com Cartas de Cães Fofinhos

1. Introdução

Neste projeto, desenvolvemos uma versão virtual do jogo Super Trunfo utilizando cartas com imagens de cães fofinhos. O objetivo foi combinar conceitos aprendidos em sala de aula e conhecimentos pessoais para criar uma interface gráfica intuitiva, dinâmica e interativa para o jogador.

2. Funcionalidades

Uma funcionalidade destacada do jogo é a possibilidade de escolha da dificuldade. Os métodos que controlam essa funcionalidade são `escolherCategoriaDadoProb` e `escolheCategoria`, presentes na classe `JogadorMaquina`. Dessa forma, pessoas de qualquer idade podem se divertir jogando Super Trunfo e se desafiar ao mesmo tempo.

Os métodos mencionados foram desenvolvidos para que, dependendo da categoria de dificuldade, o computador escolhesse a carta que mais se distancia da média dos atributos com probabilidades de 100%, 70%, 45% e 30%, conforme a dificuldade escolhida. A escolha aleatória é feita com a probabilidade complementar em cada modo. As médias foram calculadas pelo método `instanciarMedia` da classe abstrata `Baralho` e salvas como atributos estáticos dessa mesma classe.

Para atender ao requisito de utilizar duas classes abstratas e uma interface, definimos as classes `Jogador` e `Baralho` como abstratas, pois não queremos instanciá-las diretamente. Em vez disso, as classes `JogadorReal` e `JogadorMaquina` estendem `Jogador`, enquanto `BaralhoMao` e `BaralhoGeral` estendem `Baralho` e possuem características distintas.

A interface `I_Arquivo` foi implementada conforme os laboratórios realizados em aula. Esta interface é implementada pela classe `LerCarta`, que lê o arquivo XML contendo as informações das cartas do Super Trunfo.

3.3 Interface Gráfica (GUI)

A GUI do jogo facilita a interação entre o usuário e a lógica do jogo utilizando JavaFX. A classe `GuiApplication` inicia o aplicativo e carrega a primeira tela, definida em `main.fxml`, permitindo ao usuário selecionar a dificuldade e iniciar o jogo. A classe `GuiController` gerencia as interações do usuário e a lógica do jogo.

Quando o jogo começa, a tela muda para a interface do jogo definida em `game.fxml`, que exibe informações sobre as cartas do jogador e da máquina. O `GuiController` atualiza a interface com base no estado do jogo e controla os turnos dos jogadores. Ao final do jogo, a interface muda para a tela de fim de jogo, definida em `fimDeJogo.fxml`, que exibe uma mensagem de vitória ou derrota. O `GuiController` gerencia essa transição, garantindo uma experiência de usuário fluida.

3.4 Tratamento de Exceções

Utilizamos o método `try-catch` para tratar exceções nos seguintes métodos:

- `lerArquivo` da classe `LerCarta`, conforme o laboratório 4, onde pode ocorrer um erro na leitura do arquivo.
- No método `handleStartGame`, o bloco `try-catch` gerencia a transição para a tela do jogo quando o usuário seleciona uma dificuldade e inicia o jogo. Dentro do bloco `try`, o código carrega o arquivo `game.fxml` usando `FXMLLoader`, criando a interface da próxima tela. Em seguida, ele obtém o controlador da nova tela (`GuiController`), passa a instância do jogo para esse controlador, e configura e exibe a nova cena no palco atual.
- Na classe `GuiApplication`, o bloco `try-catch` é utilizado dentro do método `start`, chamado quando a aplicação JavaFX é iniciada. No bloco `try`, o código carrega o arquivo `main.fxml` usando `FXMLLoader.load`, criando a interface inicial da aplicação. Uma nova cena é criada com essa interface e configurada no palco principal (`primaryStage`). Finalmente, o palco é exibido com a chamada `primaryStage.show()`.
- Em ambos os casos, o bloco `catch` captura exceções do tipo `Exception` que podem ocorrer durante o carregamento do arquivo FXML ou a configuração da cena. Se uma exceção for lançada, a pilha de rastreamento da exceção é impressa para ajudar na depuração. Isso assegura que o aplicativo possa lidar com problemas ao carregar a interface inicial de forma controlada e informativa, permitindo que os desenvolvedores identifiquem e corrijam possíveis erros na configuração da interface gráfica.

3.5 Leitura de Arquivos

Optamos por fazer a leitura de um arquivo XML. O método `lerArquivo` da classe `LerCarta` traz as informações das cartas para o jogo.

3.6 Design Patterns

Implementamos alguns Design Patterns:

- A classe `Jogo` foi implementada como Singleton.
- A classe `BaralhoMao` foi implementada seguindo o padrão MVC.

4. Conclusão

O projeto final do curso MC322 resultou em uma versão virtual divertida e desafiadora do jogo Super Trunfo com cartas de cães fofinhos. Conseguimos aplicar diversos conceitos de programação orientada a objetos, como o uso de classes abstratas, interfaces, tratamento de exceções, leitura de arquivos e padrões de design. Além disso, a criação de uma interface gráfica com JavaFX proporcionou uma experiência de usuário fluida e intuitiva. Este projeto não apenas reforçou os conhecimentos adquiridos durante o curso, mas também nos permitiu desenvolver habilidades práticas valiosas.

5. Referências

- Documentação JavaFX
 - Notas de aula do curso MC322
 - Exemplos de laboratório fornecidos durante o curso
-