Projeto De Programação 3

Bruno Mamoru
Universidade Federal de Santa Catarina
UFSC
Joinville, Brasil

Bruno Mamoru
Universidade Federal de Santa Catarina
UFSC
line 3-Joinville, Brasil

Resumo: Este arquivo é referente ao projeto de programação 3

Keywords—programação, jogo, regra, projeto;

I. INTRODUCAO

O projeto em questão tem como objetivo a implementação de um jogo de baralho na linguagem de programação C++, onde parte do trabalho fora discutida e implementada em conjunto com os colegas de classe e a parte final feita individualmente para cada jogo, incluindo interface gráfica que fora feita no "QT Creator", uma IDE sugerida pelo professor. Neste caso o jogo escolhido fora o truco, jogo onde podem jogar duas duplas em um total de quatro jogadores, ou apenas dois.

II. DESENVOLVIMENTO DO JOGO

Primeiramente discutiu-se em sala juntamente com o professor e integrantes da equipe o rumo inicial do projeto, definiu-se que o projeto em um total seria composto por seis classes genéricas, carta, baralho, jogador, mesa, regra e jogo, adotou-se como repositório de trabalho o "GitHub" e fora decidido que seria implementada em primeira instância a classe que definiria como seriam as cartas que comporiam o baralho. Então implementou-se métodos e atributos que seriam essenciais, como um inteiro que representa o valor e a sequência de naipes, que fora incluída em uma "enum class" (paus, espadas, copas e ouros).

Feito isso a tarefa seguinte foi determinar como as cartas implementadas seriam agrupadas em um baralho, e como seria o comportamento deste no jogo. Adotou-se primeiramente que o baralho seria um "vector" de cartas, no entanto tal solução apresentou alguns problemas, pois mostrou-se inviável retirar uma carta que situava-se em sua primeira posição. Como resolução do contra-tempo fora implementada uma saída sugerida por um integrante, que fora utilizar o método "pop_front" da classe "deque".

Além disso, resolveu-se definir o baralho como um template de carta, pois alguns integrantes do grupo implementaram de maneiras diferenciadas as cartas, deste modo deixou-se livre a escolha individual das características que uma carta deveria ter.

Em sequência discutiu-se possíveis formas de implementar o jogador, chegando a conclusão de que, como o baralho fora definido como template de carta o jogador também deveria ser, pois este teria um "vector" de cartas que chamou-se de "mão", também implementou-se métodos que permitiram o jogador a interagir com o jogo, por exemplo, retirar uma carta da mão do jogador, colocar uma carta na mão do jogador, consultar pontuação entre outros.

Sob o mesmo ponto de vista efetivou-se a discussão sobre a implementação da mesa, e chegou-se a conclusão de que ela deveria ser um template de baralho, pois esta teria como atributo um baralho, e definiu-se que a principal função da mesa seria manter e realizar a interação dos jogadores com os outros objetos presentes, como por exemplo o baralho, os montes, as cartas jogadas e também com outros jogadores. Para isso implementou-se alguns métodos básicos, como, o de criar um novo monte, tirar carta dos jogadores, entregar carta a jogadores, passar carta de um monte para outro, passar carta de um jogador para outro, entre outros. Posteriormente criou-se uma argumentação de como definir quais seriam as principais funções da regra e do jogo, e como seria a composição entre eles.

Por conseguinte definiu-se que que a principal função da regra seria reger o comportamento do jogo e dos outros objetos, e que o jogo seria aquele responsável por realizar todas as interações dos objetos presentes. Para tal, devido a necessidade de cada jogo individual possuir uma regra diferenciada, debateu-se sobre a possibilidade de fazer uma

derivada de regra para cada jogo, deste modo cada integrante da equipe poderia implementar sua própria regra tendo como base a regra genérica criada.

No entanto, tal decisão gerou certos contratempos, um deles foi a necessidade do jogo possuir um ponteiro inteligente da regra base, para que este pudesse receber as regras derivadas e como consequência disto, os métodos utilizados pelo ponteiro deveriam estar necessariamente declarados também na regra base, e não somente na regra derivada. Solucionou-se o problema fazendo uso de métodos virtuais que poderiam ser sobre-escritos pela regra derivada de acordo com a necessidade individual de cada integrante da equipe. Feito isso, cada participante da equipe iniciou a implementação individual do jogo escolhido levando em consideração as classes bases implementadas. Neste caso o jogo escolhido fora o truco.

A. Regras do jogo

1) A Sequência de força das cartas no truco em ordem decrescente é 3,2,A,K,J,Q,7,6,5,4.

2) O truco é baseado em um sistema de pontuação por turno.

3) Cada turno consiste em um máximo de três rodadas

4) Cada rodada é caracterizada pelo ciclo completo de jogadas, ou seja, uma rodada termina quando todos os jogadores da mesa jogarem uma vez.

5) Na primeira rodada da partida o jogador que iniciará jogando é decidido aleatoriamente.

6) Após o primeiro turno, o jogador que iniciará jogando será escolhido em sentido horário.

7) Após a primeira rodada do turno, o jogador que começará jogando é aquele que ganhou a última rodada, em caso de empate, o jogador que começará será o mesmo que começou na rodada anterior.

8) Cada turno vale 1 ponto ou múltiplos de 3 até um máximo de 12

9) Vence o turno o jogador ou dupla que ganhar duas rodadas primeiro, ou em casa de empate na primeira, o jogador ou dupla que levar a segunda rodada vence. Em caso de empate das três rodadas ninguém ganha.

10) Em sua vez o jogador tem direito de pedir "Truco", se os adversários aceitarem o turno passará a valer 3, 6, 9 e 12. Caso o adversário recuse o jogador que pediu truco ganha em pontos o valor do turno anterior.

11) Quando um jogador ou dupla chega a 11 pontos, este tem direito à mão de 11, se em dupla poderá ver a mão do parceiro, e decidir se irá jogar a partida ou não. Caso o jogador recuse, o adversário receberá um ponto, caso aceite e perca, o adversário receberá 3 pontos.

12) e ambos os jogadores/duplas atingirem mão de 11, teremos a mão de ferro ou rodada às cegas, onde nenhum jogador poderá olhar suas cartas.

13) O vira será a carta virada após a distribuição de cartas, a carta mais forte do turno será a próxima da

III. IMPLEMENTAÇÃO DO TRUCO

Primeiramente procurou-se definir qual seria o método para comparar a força das cartas de acordo com a regra do truco, algumas sugestões foram levantadas, como por exemplo ter uma herança de carta e fazer override de operador igual, maior e menor da classe carta. No entanto tal método dificultaria na comparação das manilhas, pois precisaria ter informações sobre a carta que fora virada, por esta razão, implementou-se um função que compararia as cartas, e que receberia como parâmetro a carta a ser comparada, a carta com a qual comparar e o vira.

Considerando que no construtor do baralho quando passados o número de cartas de um baralho de truco, quarenta cartas, o baralho somente possuiria cartas de um a dez, para implementar o método de comparar as cartas definiu-se que a carta "A" seria representada pelo número um, a carta "K" pelo número dez , a carta "J" pelo número nove e a carta "Q" pelo número oito, enquanto as outras seriam representadas pelo seu respectivo número.

Feito isso, dividiu-se o método compara em métodos menores que realizaram cada uma uma ação específica, como

regra compara manilhas, regra compara valores, e implementou-se na classe da regra truco, que seria a responsável por comparar as cartas.

Entretanto percebeu-se que para poder utilizar o método compara declarado na derivada de regra, regra truco, precisava declará-la em primeiro lugar na regra base, por este motivo modificou-se a regra base adicionando um método para comparar cartas virtual.

Em seguida precisou-se de um método para ter a informação de que jogador havia ganho a rodada, para isso implementou-se um método novamente na regra truco, pois esta também controlaria quem ganha as rodadas.

Para o próximo passo implementou-se um método que retornaria o jogador que ganhou o turno, também na regra truco, mas para tal necessitou-se de um atributo na regra truco que guardaria o resultado das rodadas jogadas, portanto criou-se um "vector" de "unsigned int" ou "std::size_t" que guardaria ou o resto da divisão da posição do jogador por dois (0 ou 1), que indicaria qual equipe ganhou a rodada ou armazenaria o número dois caso a rodada terminou em empate.

Então novamente modificou-se a regra base, adicionando métodos virtuais para que os outros integrantes pudessem também sobrescrever o método para seu uso.

Além disso foi também necessário métodos que gerenciassem o pedido de truco dos jogadores e o valor da pontuação que o turno estava valendo, em função disso implementou-se um método na regra truco que gerenciaria o valor da pontuação de acordo com as condições que o jogo truco recebia, e implementou-se um método no jogo que permite que os jogadores peçam truco.

Necessitou-se também de métodos que informam ao usuário quando seria sua vez de realizar uma jogada, para isso implementou-se um método que esperaria a entrada do usuário para realizar uma ação, com esse objetivo definiu-se que a posição do jogador que representaria o usuário seria sempre a posição zero, deste modo utilizou-se métodos implementados previamente na classe base , que informam a posição do jogador atual, e métodos implementados posteriormente na regra truco, que retornam qual seria o jogador em que começaria a rodada e qual seria o jogador em que terminaria a rodada, para estipular o momento em que o jogo requisitaria uma ação do usuário.

Tendo em mente que o jogo implementado teria como outros jogadores o próprio programa, implementou-se métodos que simulam as jogadas de um jogador. Para isto utilizou-se também métodos implementados anteriormente no jogo base, e também escreveu-se novos métodos para simular jogadas, como por exemplo aceitar ou correr de um truco.

IV. INTERFACE GRAFICA

Após a implementação do truco, utilizou-se o "QT Creator" para implementar a interface gráfica.

Quando iniciado o jogo, a interface solicitará ao usuário o qual será a ação desejada, podendo esta ser "jogar", "help" e "quit",.

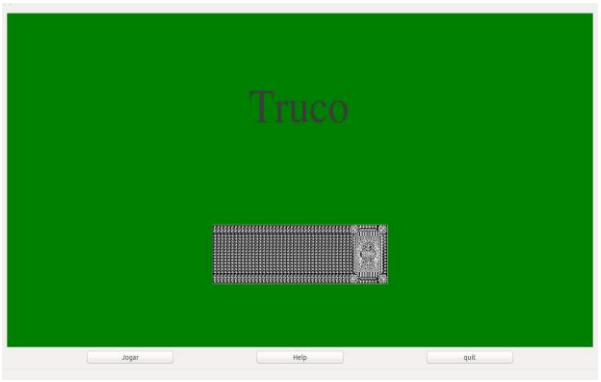

Fig.1. Exemplo de tela inicial

Se a ação desejada for "help", abrirá então uma tela de diálogo que informará ao usuário do jogo as regras do mesmo.

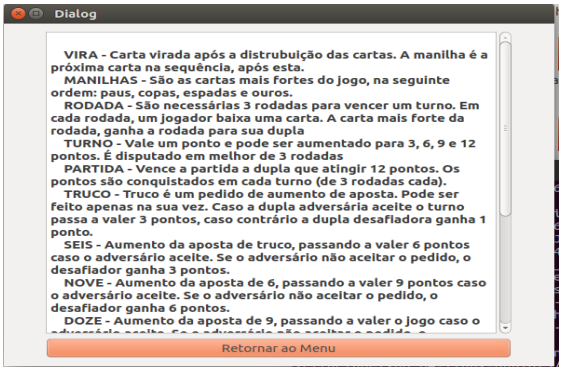

Fig 2. Exemplo de tela help

Se a ação for jogar, apresentará ao usuário uma tela para inserir o nome do jogador.

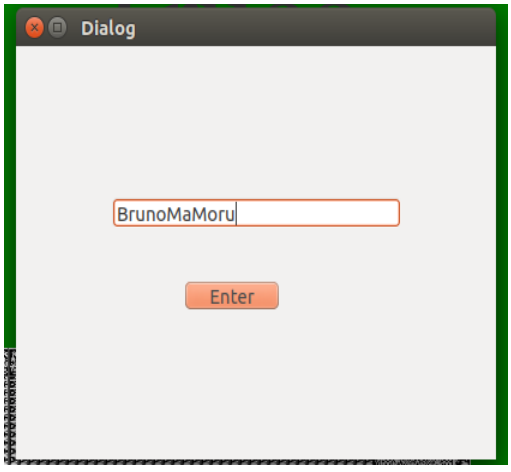

Fig.3. Exemplo de tela de inserção de nome

Utilizou-se imagens de cartas encontradas na internet para mostrar ao usuário suas cartas e as jogadas pelos jogadores, tendo como base o exemplo mostrado em sala de aula pelo professor.

A enumeração das cartas foi dada como citada anteriormente no desenvolvimento do jogo.

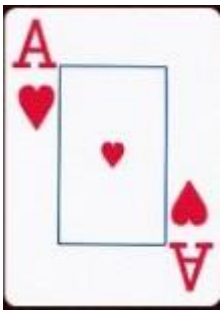

Fig.4 .Exemplo de carta utilizada na interface

Para realizar a interação do usuário definiu-se que quando a carta fosse clicada com o botão esquerdo do mouse , ou seja selecionada, pelo usuária esta se moveria uma determinada altura no eixo y, indicando ao usuário que a carta fora selecionada

Fig.5. Exemplo de carta selecionada

Da mesma forma, quando o jogador deseja-se virar a carta para escondê-la, este poderia clicar na carta com o botão direito do mouse, e esta viraria, mostrando hora a face, hora a parte de trás da carta.

Fig.6.. Exemplo de carta virada

O nome dos jogadores e as suas respectivas pontuações é mostrada na lateral da tela de jogo, toda rodada que o jogador pontuar a pontuação será atualizada.

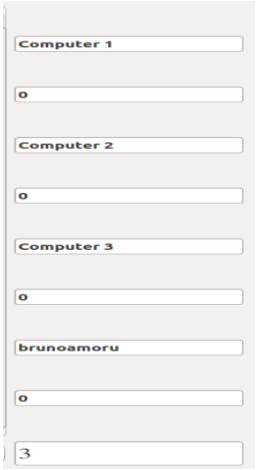

Fig.7 Exemplo da tela de pontuação

A ação do usuária é pega através dos botões da interface, enquanto não há nenhuma entrada o programa continua em espera, ou em loop.

Implementou-se também telas especiais para a mão de 11 e outros eventos, como por exemplo, a mão de ferro, ou rodada às cegas. Quando em mão de 11, o usuário poderá ver as suas cartas, as cartas do seu parceiro e o vira, podendo então decidir jogar ou correr.

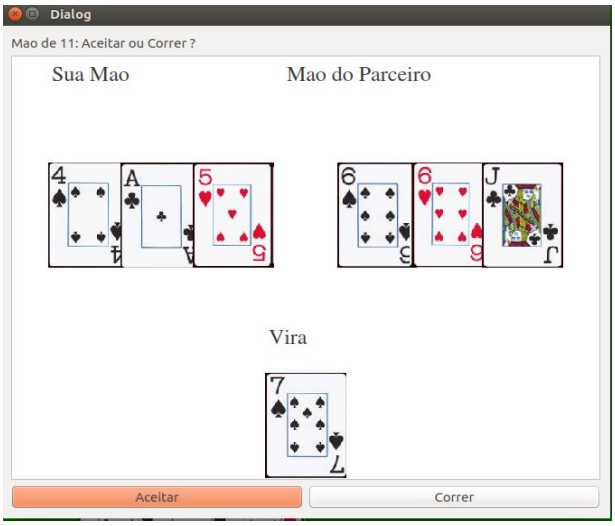

Fig.8.. Exemplo da tela da mão de 11

E quando em mão de ferro, ou rodada às cegas, o jogador não poderá selecionar nem virar as cartas de sua mão, estando todas elas viradas para baixo.

Por fim, quando um dos jogadores chegar a 12 ou mais pontos, a interface apresentará o nome dos jogadores que ganharam a partida.