

## RELATÓRIO FINAL – PROJETO DOMINÓ

**Desenvolvido pelo Grupo JIPP**

Igor Costa - RA00336423

João Avila Harduin - RA00332195

Pedro Evaristo de Oliveira - RA00274677

Priscilla de Souza Jardim - RA00333990

### 1)Manual de operação do jogo Dominó JIPP:

#### - Menu Principal

```
===== DOMINO JIPP =====  
  
1.Iniciar o jogo(2 jogadores)  
2.Iniciar o jogo(contra o computador)  
3.Regras gerais do jogo  
4.Gravar partida  
5.Recuperar partida gravada  
6.Sair  
  
Opcao:
```

O jogo inicia com a apresentação do menu principal composto por 6 opções, listadas em ordem crescente, com diferentes funções cada uma.

## - Regras do jogo

```
===REGRAS DO DOMINO JIPP===

Regra1: Cada jogador começa com 7 pecas aleatorias, podendo comprar quantas pecas quiser no seu turno.

Regra2: O jogo inicia com o jogador que tiver a peca [6|6]. Caso ninguem tenha a [6|6] entao calcula-se outra primeira peca que e' jogada e o primeiro turno sera do jogador que nao tinha a primeira peca.

Regra3: Na sua vez cada jogador deve colocar uma das pecas em uma das extremidades da mesa, de forma que os lados coincidam.

Regra4: Quando o jogador coloca sua peca sobre a mesa, seu turno acaba e passa para o proximo.

Regra5: Caso um jogador nao possua uma peca compativel para jogar, deve comprar do monte ate encontrar uma peca ou ate as pecas do monte acabarem. So quando atendidas essas condicoes o jogador pode passar a vez(passar o turno sem jogar uma peca).

Regra6: O jogo pode acabar de duas maneiras, porem nos dois casos o deposito de pecas deve estar vazio. A primeira e' se um jogador colocar sua ultima peca na mesa, essa acao e' chamada de bater e ele vence o jogo. A segunda sao casos onde nenhum dos jogadores pode continuar a partida, acontece quando ninguem tem uma peca possivel de ser lancada na mesa, nesse momento a partida esta fechada. Vence aquele que tiver menos pecas e caso empatem, vence aquele com menos pontos(soma dos lados das pecas).
```

Antes de se iniciar o jogo recomenda-se a leitura das regras do mesmo. Esta tela aparecerá caso a opção digitada no menu inicial for “3” e será composta pelas 6 regras principais do dominó, muito bem explicadas para o usuário.

## - Jogando

```
O primeiro lance foi do jogador 1 ← Anúncio da primeira jogada (automática)

=====
MESA: [6|6] ← Composição atual da mesa
=====

Peças disponíveis para o jogador
↓
JOGADOR 2: 1.[0|5] 2.[1|6] 3.[1|4] 4.[2|2] 5.[2|6] 6.[0|3] 7.[0|1]
J - Jogar (possiveis: 6 ou 6)
C - Comprar
P - Passar
S - Sair da partida
Opcao: ← Menu de ações do jogador em sua vez de jogar
```

estão em sua mão (mostradas logo acima do menu de ações).

## Escolha o lado da mesa (E/D):

Para as vezes em que os dois lados da mesa são iguais, ou então a peça escolhida pelo jogador se encaixa em ambos os lados, é perguntado onde o jogador deseja depositar a sua peça.

## - Mensagens de auxílio

Algumas mensagens aparecem ao longo do jogo para melhor guiar o usuário. São elas:

### **Opcao Invalida!**

\* Resposta digitada no Menu Principal ou no Menu de Jogar não é compatível com as opções disponíveis.

### **Peca incompativel! Tente novamente ou compre uma peca**

\* A peça escolhida para depositar na mesa não se encaixa em nenhum dos lados dessa.

### **Passagem bloqueada. O deposito nao esta vazio. Voce deve comprar**

\* Tentar passar a jogada mesmo com peças disponíveis para a compra

### **Passagem bloqueada. Faca sua jogada!**

\* Tentar passar a jogada tendo peças que se encaixam no jogo (lado direito ou esquerdo).

### **Nada a comprar**

\* Tentar comprar uma peça estando o depósito vazio.

### **Peca invalida! Voce nao possui essa peca. Tente novamente**

\* Escolher uma peça que não faz parte das opções apresentadas. Por ex: escolher a peça de número 8, sendo que o jogador possui apenas 5 peças em sua mão.

## - Fim do jogo

No jogo de dominó não há empate, e as regras para a determinação de um vencedor estão bem explicitadas na opção 3 do menu principal. Ainda assim, há 2 maneiras de se encerrar um jogo, que são bem explicadas quando o mesmo acaba.

```
===Resultado da partida===
```

```
O Jogador 1 jogou sua ultima peca na mesa e o monte esta' vazio!
```

```
===Jogador 1 bateu!===
```

\* Um jogador ficou sem peças ao fazer sua última jogada e não há mais peças para comprar.

```
Os dois jogadores passaram a vez em seguida!
```

\* Os dois jogadores não possuem mais peças que se encaixam na mesa e o depósito está vazio, portanto passam, ambos, de vez.

### - Jogando contra o computador

Jogar contra o computador não apresenta mudanças reais na interação do usuário com suas peças e suas opções de jogada. A principal diferença será a de que a máquina fará sua jogada automaticamente, e mensagens servirão de guia para o usuário descrevendo essas jogadas do computador.

```
O jogador 2 e' o computador.
```

```
O computador jogou a peca [2|5].
```

```
O Computador comprou!
```

```
O Computador passou a vez
```

### **- Gravando e recuperando um jogo salvo**

O jogo ainda possui o recurso de salvar um jogo para retomá-lo novamente em um momento posterior, através das opções 4 e 5 do menu principal, respectivamente.

**Nenhum jogo foi iniciado para ser gravado.**

\* Caso não exista um jogo para ser salvo, será mostrada a mensagem acima.

**Gravados os arquivos CAD\_DOMINO, CAD\_MESA e CAD\_JOGO**

\* Se a gravação for feita com sucesso, esta mensagem aparecerá na tela.

**Jogo terminado nao pode ser gravado.**

\* Caso o usuário tente gravar um jogo que já terminou, será mostrada a mensagem acima.

**retornando ao jogo recuperado**

\* E por último, quando se desejar retomar um jogo salvo, o recado acima mostrará que tudo ocorreu corretamente.

Essa foi uma visão geral de como funcionará o Jogo Dominó JIPP nas mãos do usuário. Obviamente, algumas experiências e descobertas ocorrerão somente no momento em que o jogo for desfrutado. Assim, desejamos a todos um bom jogo!

## **2)Organização do código em termos de módulos, funções, procedimentos e estruturas de dados utilizadas:**

O código foi organizado seguindo o padrão MVC: Model - View - Controller. Totalizando 7 arquivos, são eles: projeto, model, view e controller(arquivo.cpp) com cada um deles contando com seu respectivo arquivo de cabeçalho menos o projeto. Para compilação e execução do programa deve-se usar o arquivo “projeto”, pois nele estão incluídos todos os outros em forma de bibliotecas pessoais(bibliotecas com aspas).

### **- Os arquivos de implementação(arquivo .cpp):**

No “mode”l, é onde estão as **estruturas de dados utilizadas**. Sendo elas o tipo “struct” e vetores, usados para criação das três estruturas usadas no programa, a estrutura “Mesa”, “Peca” e “Jogo”. A estrutura “Mesa” é usada para criar o vetor “mesa[28]” que guarda os dois lados(direito e esquerdo) de cada uma das 28 peças que podem ser jogadas na mesa. A estrutura “Peca” definida como “tipo\_peca” é usada para criar o vetor estrutura “peca[28]” que é usado para guardar as 28 peças do jogo, seus lados e status, os status sendo ‘0’ para depósito, ‘1’ para em posse jogador 1, ‘2’ para em posse do jogador 2 e finalmente ‘M’ para jogada na mesa, além disso é usada para criar uma variável estrutura “aux” usada na função de embaralhamento das peças. A estrutura “Jogo” é usada para criar uma variável estrutura chamada “sitJogo”(situação do jogo), nela encontram-se todas as variáveis globais obrigatórias de serem guardadas em caso de o usuário optar por salvar o jogo, assim como variáveis globais úteis de serem salvas. Essas variáveis globais e suas funções no programa são explicadas no arquivo de interface do “model”.

No “controller”, onde acontece a maior parte do programa, são realizados todos os tipos de procedimentos menos os de interação com o usuário e onde é formada a lógica do programa. Ele controla o “view” e o “view” está incluído nele como biblioteca, portanto no controller pode-se encontrar a execução de todas as funções definidas no “view”. Também no “controller” encontra-se a inicialização das variáveis globais e o uso extensivo delas, além disso nele existem duas funções bem diferentes das outras em termos de procedimentos mas que são muito úteis e usam todas as estruturas criadas no “model”, as funções de salvar e recuperar um jogo.

O “view”, ao contrário do “controller”, só é usado para interações com o usuário, ou seja, para funções que ao menos possuem procedimentos que interagem com esse. Nele encontram-se todas as funções que possuem “printf” e/ou “scanf”.

No projeto, estão incluídos como bibliotecas o “controller” e o “model” assim, portanto, inclui todo o programa. Nele é onde encontra-se a função “main” do programa, ela executa a função de criar peças e a função de iniciar o jogo, as duas estando definidas no “controller”. Assim dando início a um desencadear de chamadas de função até o término dessas.

## **- Os arquivos de interface(arquivo .h):**

Como esclarecido anteriormente, cada um dos arquivos de implementação possui incluído como biblioteca um arquivo de interface com exceção do projeto. No “controller” e no “view” encontra-se o protótipo de cada das funções presentes nos seus respectivos arquivos de implementação, além disso uma coleção de comentários que explica brevemente o que cada função faz, ou seja, se existente quais as entradas e saídas de cada função, a

relação entre o que entrou e saiu e eventualmente as transformações realizadas sobre o que entrou.

No “model”, é onde estão localizadas as declarações de todas as variáveis globais(aquelas que são declaradas fora de uma função), além de comentários sobre a finalidade dessas variáveis no programa e o que elas guardam.

As bibliotecas ANSI C(bibliotecas <>) utilizadas no programa são, <stdio.h>, <stdlib.h>, <ctype.h> e <time.h> .

Sobre os comentários, também é possível achá-los nos arquivos de implementação, de forma mais extensa, em algumas partes do “controller”. Onde é oportuno detalhar os procedimentos mais complexos prezando o entendimento do leitor, além de explicações de chamadas de função e atribuição de valores à variáveis em locais específicos.