

PROJECTO Nº2

1. INTRODUÇÃO

Neste projecto os estudantes terão oportunidade de pôr em prática os conhecimentos da programação procedimental e da programação baseada em objectos que foram adquirindo ao longo da unidade curricular de Programação. Embora sejam dadas algumas indicações sobre o desenvolvimento do projecto, pretende-se que os estudantes sejam capazes de tomar decisões relativas à estrutura da solução que vierem a implementar.

O projecto divide-se em duas fases: concepção e desenvolvimento. Na primeira fase (concepção), pretende-se que seja realizada uma análise do problema e produzido um esboço da solução (diagrama de classes). A segunda fase consiste na implementação do projecto.

Pretende-se, com este projecto, implementar um jogo de Dominó "All Fives".

2. DESCRIÇÃO DO JOGO

O Dominó "All Fives" é um jogo que se joga com peças rectangulares divididas em duas partes quadrangulares, contendo cada uma um número de 0 a 6. Há portanto um total de 28 peças.

Aos jogadores (entre 2 e 4) são inicialmente distribuídas 7 peças. As que sobram, se as houver, ficam no "monte". O objectivo não é apenas ficar sem nenhuma peça, mas também acumular pontos.

Começa a jogar, colocando uma peça na mesa de jogo, o jogador que tiver a peça dupla (6-6, 5-5, ..., 0-0) de maior valor ou, no caso de não haver nenhuma em jogo, o jogador que tiver a peça de valor mais alto.

Ao jogar, o jogador deve colocar uma das suas peças de forma adjacente a outra peça da mesa que tenha um número em comum que esteja livre (isto é, sem nenhuma peça adjacente). No caso particular das peças duplas, estas devem ser colocadas de forma perpendicular às restantes peças. A primeira peça dupla jogada, após ter já duas peças adjacentes, pode ser continuada em qualquer direcção. Haverá portanto, em cada passo, quatro locais para colocar uma nova peça na mesa, excepto enquanto a primeira peça dupla jogada não tiver ainda duas peças adjacentes (neste caso haverá apenas dois locais possíveis).

Sempre que os pontos do lado livre das peças a partir das quais se pode jogar somarem um múltiplo de 5, o jogador pontua essa soma. No caso particular de uma peça dupla, contam os pontos de ambos os lados da peça. Cada lado da primeira peça dupla jogada continua a contar até que seja colocada uma peça adjacente a esse lado (ver figuras em baixo).

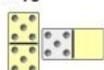
No caso de o jogador não ter nenhuma peça jogável, deve ir ao monte "comprar" novas peças, até obter uma peça jogável ou não haver mais peças para comprar, caso em que o jogador passará a vez.

As figuras seguintes mostram uma sequência de jogo possível, bem como os pontos acumulados.

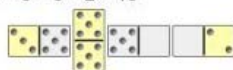
a) $5 + 5 = 10$



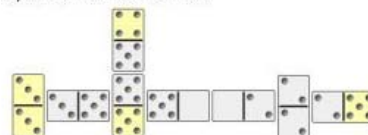
b) $5 + 5 + 0 = 10$



c) $5 + 5 + 3 + 2 = 15$



d) $5 + 4 + 3 + 3 + 5 = 20$



No final, o primeiro jogador que ficar sem peças na mão fica com os pontos correspondentes à soma dos números das peças que ficaram nas mãos dos restantes jogadores, arredondados ao múltiplo de 5 mais próximo.

Mais informações sobre o jogo, incluindo um glossário de termos ingleses, poderão ser consultadas, por exemplo, em <http://www.dominio-games.com/dominio-rules/allfives-rules.html>.

3. IMPLEMENTAÇÃO BÁSICA (15 valores)

Pretende-se desenvolver um programa que permita jogar Dominó "All Fives", com todas as regras descritas acima.

Notas sobre o desenvolvimento

Tudo o que não estiver especificado no enunciado poderá ser livremente especificado pelos membros do grupo de trabalho, devendo as especificações adicionais ser indicadas no relatório a elaborar.

A estrutura da solução a desenvolver (classes e seus métodos) é definida pelo grupo de trabalho. Deve ser incluído no relatório um diagrama de classes da solução desenvolvida.

Sugere-se a criação de classes para:

- *Bone*: uma peça de dominó.
- *Player*: um jogador.
- *Boneyard*: o monte de peças.
- *Board*: a mesa de jogo onde vão sendo colocadas as peças jogadas.

4. VALORIZAÇÃO (3 valores)

Pretende-se desenvolver um jogador de Dominó automático. Este jogador deve ser capaz de jogar de forma inteligente, utilizando heurísticas simples. Como exemplos: é preferível jogar peças que permitam arrecadar pontos, e quantos mais melhor; é bom livrar-se das peças duplas o mais cedo possível; de acordo com a informação disponível (peças na mão e peças na mesa), pode-se inferir sobre as possibilidades de desenvolvimento do jogo (jogadas possíveis dos adversários e próprias).

5. RELATÓRIO (2 valores)

Deve ser elaborado um pequeno relatório seguindo a estrutura indicada em anexo.

6. ENTREGA DO PROJECTO

Conforme descrito no início deste documento, o projecto divide-se em duas fases.

6.1. Entrega intermédia

A primeira consiste na concepção da solução através da construção de um **diagrama de classes**. Deverá ser entregue, via Moodle, até às **23:55 horas do dia 29/04/2011**, um documento **PDF**, de nome **TxxGyy.pdf** (onde **xx** e **yy** representam o número da turma e o número do grupo de trabalho; por exemplo, T02G07, para o grupo 7 da turma 2). Este documento, com um máximo de 3 páginas, deve conter o diagrama de classes e uma breve descrição de cada classe proposta e seus métodos.

6.2. Entrega final

A segunda fase consiste na implementação do projecto, que deve ser submetido até às **23:55 horas do dia 21/05/2011**, tendo em conta as regras gerais publicadas na página da unidade curricular, no Moodle da FEUP.

ANEXO

CONSIDERAÇÕES GERAIS SOBRE A ELABORAÇÃO DO RELATÓRIO

- O relatório é um documento técnico que tem como objectivo a documentação do trabalho realizado, pelo que deve utilizar terminologia adequada. Deve ter boa apresentação e ser de leitura fácil, e não deve ser uma cópia do guião do trabalho.
- O relatório deve ter um máximo de 5 páginas, sem contar com a capa e com os apêndices.
- Usar os seguintes tipos de letra e espaçamentos entre linhas:
 - Títulos de secções e subsecções: Arial bold, com tamanho entre 14 e 10, com numeração.
 - Texto do corpo do relatório: Times New Roman, tamanho 10, com espaçamento simples dentro de cada parágrafo do corpo do relatório e ½ linha de espaço antes de cada parágrafo.

ESTRUTURA DO RELATÓRIO

Capa

Na capa deve constar o nome do curso, o nome da unidade curricular, o título do projecto, o nome dos autores, a turma e o grupo a que pertencem, bem como a data de entrega.

1. Introdução (máx. ½ página)

Descrever sumariamente o que é o trabalho, e que funcionalidades foram implementadas (identificar também as funcionalidades pedidas mas não implementadas, bem como funcionalidades não solicitadas que tenham sido acrescentadas como forma de valorização).

2. Concepção e Implementação da Solução (máx. 4 páginas)

2.1. Estrutura de classes

Descrever as classes de C++ que foram definidas e as relações entre elas (recorrendo a um diagrama de classes). Justificar as principais opções tomadas em relação à organização das classes.

2.2. Implementação das classes e algoritmos utilizados

Descrever e justificar as principais opções de implementação tomadas.

Descrever os algoritmos usados na implementação das funções do programa, sempre que eles tiverem uma complexidade ou originalidade que o justifique.

Caso tenham surgido, durante a implementação, algumas dificuldades dignas de relevo, descrevê-las e explicar como foram resolvidas.

3. Conclusão (máx. ½ página)

Apresentar as conclusões consideradas convenientes.

Referir, por exemplo, em que medida é que os objectivos traçados no guião foram cumpridos, as limitações conhecidas do programa (se existirem), quais os principais ensinamentos retirados da realização deste trabalho, como é que o programa poderia ser melhorado, etc.

4. Bibliografia

Indicar a bibliografia ou URL's consultados que tenham sido relevantes para realização do trabalho.

A citação de um livro deve incluir: o nome do(s) autor(es), o título, a editora e a data/ano de publicação. Exemplo:

- P. Deitel, H. Deitel: "C++ How to Program", Pearson, 2010.

A citação de um URL deve incluir, se possível: o nome do(s) autor(es), o título, o local de publicação, a data/ano de publicação, o URL e a data em que foi consultado. Exemplo:

- Anónimo, "Citation Formatting Guidelines for Engineers", Bevier Engineering Library – University of Pittsburgh, <http://www.library.pitt.edu/guides/citing/engineering.html> (consultado em 2011-04-05)

5. Apêndices

Descrever, sucintamente, os testes efectuados para verificar o bom funcionamento do programa.

Apresentar capturas de ecrã do resultado de alguns testes e de uma ou mais sessões de utilização da versão final do programa.

Incluir também quaisquer ficheiros de configuração que tenham sido utilizados nessas sessões.