

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METROPOLE DIGITAL

Projeto das disciplinas EDB/LP

2014.1

Bejeweled

1. Informações Iniciais

O projeto consiste no desenvolvimento, em C++ empregando a biblioteca SDL (<http://www.libsdl.org>), de um jogo conhecido como Bejeweled. O jogo será descrito mais adiante na Seção 3. O projeto deverá ser feito por um grupo de no máximo quatro alunos. A nota do projeto pode variar entre 0.0 e 10.0. A data de entrega, assim como o local de apresentação, será agendado.

Para desenvolver o jogo será necessário o suporte da biblioteca SDL. Uma apostila pode ser encontrada na página em <http://equipe.nce.ufrj.br/adriano/c/apostila/sdl/index.html>. Mais adiante iremos detalhar o que o aluno deverá preparar para a apresentação do trabalho.

2. Avaliação

Cada grupo terá **meia** hora para fazer sua apresentação. Nesta apresentação, os grupos deverão entregar o material pedido bem como demonstrar o funcionamento do jogo. Além disto, deverão estar aptos a responder questões sobre o desenvolvimento do jogo.

Importante: Não será dada uma única nota ao grupo. Cada componente do grupo receberá uma nota de acordo com seu desempenho durante esta apresentação.

O jogo será avaliado como um todo, ou seja, os requisitos não receberão pontuações individualmente. Dessa forma, a falta de um ou mais requisitos acarretará na perda de pontos, que poderá ser compensada (não totalmente, claro) através de outros componentes bem desenvolvidos.

Componentes adicionais serão muito bem vistos, desde que implementados de maneira racional.

Lembre-se de usar o bom senso para não transformar criatividade em bagunça.

3. Descrição e Objetivo

O jogo é conhecido como Bejeweled e se desenvolve em uma área quadrada e dividida em uma grade de tamanho $N \times N$. A Figura 1 ilustra uma possível tela deste jogo. Inicialmente esta grade é preenchida com diversos bitmaps de gemas preciosas (daí o nome inglês do jogo). Nesta tela inicial não pode haver nenhum grupo de três ou mais bitmaps iguais na vertical ou horizontal.

O jogador pode, a cada jogada, trocar dois bitmaps de posição na grade. **Importante, para esta troca poder ser realizada duas condições devem ser satisfeitas:**

- as células a serem trocadas devem ser vizinhas na vertical ou horizontal;
- após a troca, grupos de vizinhos com três ou mais bitmaps devem ser formados.

Os grupos de vizinhos devem ser retirados da grade deixando espaços vazios. As jóias nas posições acima dos espaços vazios caem para preencher as posições vazias. Uma vez que todas as jóias caíram, espaços vazios aparecem no topo da grade. Estes espaços vazios devem ser preenchidos por novos bitmaps sorteados entre os possíveis bitmaps. Se durante toda esta movimentação novos grupos de vizinhos forem formados, eles devem ser retirados e o processo de queda e substituição é repetido. Estes movimentos devem continuar até que não haja mais células da grade com vizinhos.



Figura 1: Exemplo de tela do Jogo

3.1. Controlando os Movimentos

Para trocar duas jóias vizinhas de posição, uma das possibilidades é o usuário marcar com o botão esquerdo do mouse as células das jóias que deseja trocar. O programa deve recusar trocas que não resultam na formação de grupos de três ou mais vizinhos.

3.2. Pontuação e Tempo

Cada quadrado de mesmo bitmap eliminado vale cinco pontos a mais para o jogador. O uso do botão dica (caso implementado) implica em custo. Cada dica deve custar 10 pontos.

3.3. Desfazer

Implemente a funcionalidade “desfazer” (undo). Para efeitos da mecânica do jogo, o undo pode tirar alguns pontos do jogador, mas, em certas situações ele vai querer fazer o undo mesmo perdendo pontos.

3.4. Término do Jogo

O jogo termina em duas (três) situações:

1. quando não há mais possibilidades de trocas. Portanto, o seu programa deve verificar a possibilidade de troca de bitmaps;
2. o jogador apertou a tecla escape;
3. o tempo de jogo acabou (opcional).

3.5. Na Internet

Este jogo pode ser encontrado na Internet nos seguintes sítios:

- bejeweled.1001jogos.pt
- jogosonline.uol.com.br/bejeweled-2_5944.html?

4. Telas

Nesta seção iremos apresentar as telas que compõem o jogo. As telas não são todas obrigatórias, a não ser quando indicado. No entanto, mais telas valem mais pontos na nota final do jogo.

4.1. Tela Inicial

O jogo deve possuir uma tela inicial, que deve ser apresentada ao jogador antes da tela com o jogo em si. A tela inicial deve pelo menos conter o nome do jogo, os nomes dos integrantes do grupo, e permitir a navegação para outras telas. Esta tela é obrigatória. Nesta tela também podem ser incluídas instruções de como jogar e possíveis opções para o jogo. A Figura 2 mostra um exemplo de tela inicial.



Figura 2: Exemplo de tela inicial do Jogo.

4.2. Tela do Jogo

A tela principal do jogo é obrigatória e deve conter pelo menos os seguintes itens:

- a grade com os bitmaps das jóias;
- o nome do jogo em destaque;
- um espaço para contagem dos pontos do jogador;
- um espaço para mostrar o tempo restante do jogo (caso o jogo tenha contagem de tempo);
- uma possível indicação da fase em que está o jogo (caso o jogo tenha diversas fases);
- (opcional) um botão dica para que o jogador possa pedir uma dica sobre que troca resultaria em um grupo de vizinhos.



Figura 3: Exemplo de tela de jogo em andamento.

4.3. Tela de Instruções

Esta tela (Figura 4) é opcional e apresenta as instruções de como se joga Bejeweled.

A tela final deve permitir ao jogador poder jogar mais uma vez. A Figura 6 mostra um exemplo de tela final.



Figura 6: Exemplo de tela final do Jogo.

5. Estrutura do Jogo

O jogo pode ter apenas um nível. Níveis adicionais podem ser incluídos e contarão mais pontos na nota final do projeto.

O jogo deverá ter alguma estrutura global de fluxo (telas e/ou menus). Um jogo que já começa dentro de uma fase, e termina (“fecha”) assim que se chega ao final não é muito amistoso.

Uma sugestão de configuração para a estrutura é a seguinte:

Introdução → Menu → Jogo → Finalização

6. Extras

O grupo que seguir e implementar os requisitos de maneira correta receberá uma nota razoável.

Os grupos estão livres para adicionar outros componentes ao jogo para que recebam notas melhores. Porém, antes de fazer alterações radicais em relação ao que foi pedido, leia a seção 2.

Alguns possíveis componentes adicionais:

- Pode-se fazer o jogo ser composto por diversas fases com graus crescentes de dificuldades.
- Pode-se implementar o jogo por tempo, ou seja o turno tem um tempo definido.

- Pode-se implementar o conceito de bonus, ou seja quando o jogador faz um certo número de pontos ele recebe alguma vantagem.
- Pode-se implementar o conceito de jóias especiais que permitem ganhar mais pontos.
- O programa pode manter uma lista com os nomes e a pontuação dos n melhores jogadores. Ao final de cada jogo, caso o jogador tenha obtido pontos suficientes para entrar na lista, o programa deve pedir o nome do jogador e colocá-lo na lista, removendo o último colocado. O jogador deve poder consultar esta lista.
- Implementar o botão dica, conforme mencionado antes.

7. Resultados

No dia da apresentação o grupo deve apresentar os seguintes itens:

Nome: Quem sabe um texto de propaganda! Ou seja, um folheto promocional incluindo a escolha de um nome para o jogo e a companhia que irá vender o produto. Um frase justificando por que o seu produto é bom.

Programa: O aluno deverá mostrar o programa funcionando satisfatoriamente. Alguns dos itens pedidos poderão não ter sido implementados. A nota levará em conta o esforço do grupo em entregar parte do projeto funcionando. Erros podem acontecer, mas não devem comprometer o funcionamento geral do programa.

Fonte(s): Deverá(ão) ser entregue(s), na hora da apresentação, em meio óptico ou magnético o(s) fonte(s) do programa para posterior avaliação. Deverá vir junto com o(s) fonte(s) um arquivo README.TXT contendo instruções de como se pode obter o executável a partir deste(s) fonte(s). Makefiles também são uma boa opção. Observação Importante: Fontes que sejam bem estruturados e legíveis, ou seja, estejam bem identados, bem organizados, separados por arquivos, usem definição de constantes, funções, loops etc serão muito bem vistos. E não deixa de ser bom ter seu código bem organizado caso uma edição emergencial seja necessária no momento da apresentação.

Manual de Usuário: Como se trata de um jogo, é interessante preparar um manual que mostre como instalar, executar e jogar o seu jogo. Procure saber como são feitos os manuais de jogos e prepare alguma coisa similar. Claro que devem ser manuais para jogos de qualidade.

Manual de Programador: Um manual que descreva como o seu programa foi construído. Não é necessário detalhar demais. Uma explicação resumida do funcionamento geral do seu programa. Pode estar incluído no README.

Meio Optico ou Magnético: Em meio magnético ou óptico os fontes, manuais e executáveis.

8. Dúvidas e Dicas

Se durante o desenvolvimento do projeto tiver alguma dúvida sobre o jogo tente três possibilidades. Primeiro leia este documento, ou novamente ou pela primeira vez. Segundo procure na internet versões do jogo e procure entendê-las. Esta procura pode resolver o seu problema. Por último pergunte ao professor. Se a sua dúvida for interessante ela e a resposta serão acrescentadas neste documento.

Procure planejar o desenvolvimento do projeto, dividindo as tarefas e organizando como as partes vão se conectar. Conte as semanas que faltam para a entrega e faça um planejamento de atividades até a data. Não adianta chegar com cara de sono no dia da entrega do trabalho. Isto não é sinal de muito trabalho mas sim de falta de planejamento.

Procure manter versões funcionando do projeto. Caso alguma idéia não funcione terá um porto seguro para onde voltar. Para manter versões antigas use o que for mais fácil, ou seja, o que você souber usar: arquivos separados, GIT, SVN etc.

Procure seguir um ciclo de pequenas modificações e testes de funcionamento. Não tente grandes mudanças porque poderá perder o controle e dificultará a descoberta do erro.

Tente a cada semana gerar uma versão que possa ser mostrada em funcionamento. Não é o programa completo mas parte dele, mesmo que pequena. Procure acrescentar uma funcionalidade de cada vez.