

Projeto Laboratorial

Jogo 3 em Linha



Descrição do Problema

Pretende-se realizar **uma aplicação que disponibilize o Jogo Três em Linha** para um só jogador.

O Jogo Três em Linha baseia-se num tabuleiro bidimensional que se vai preenchendo com peças que vão saindo aleatoriamente, tendo como objetivo conseguir juntar 3 peças iguais para formar uma linha horizontal. Apesar de na implementação gráfica solicitada, o tamanho do tabuleiro ser fixo (10 colunas x 8 linhas), a implementação do modelo deverá contemplar a possibilidade de a dimensão do tabuleiro poder ser alterada.

O Jogo Três em Linha terá vários temas para a interface gráfica (no mínimo Três: Halloween, Smiles, War) e contemplará duas formas de jogar (normal e rápida) e duas estratégias de pontuação. As regras do jogo encontram-se descritas no **Anexo A**.

A aplicação deverá ser concebida de forma a poder no futuro ser estendida para contemplar um conjunto variado de outros jogos do mesmo tipo, sendo no entanto durante este projeto apenas implementado o Jogo Três em Linha.

Existe um conjunto de funcionalidades adicionais ao jogo em si que a aplicação deverá disponibilizar, nomeadamente:

- 1) A aplicação deverá permitir registar as ações de um jogo em Ficheiro (Log). Este registo deve ser configurável. O utilizador deverá poder decidir o que quer registar: as jogadas, o tempo das jogadas, o tipo de jogo etc...
- 2) A aplicação deverá permitir registar jogadores, ficando associado a cada jogador os jogos por ele efetuados. É possível consultar o histórico de um jogador.
- 3) A aplicação deverá permitir consultar o ranking dos jogadores, por estratégia de pontuação.
- 4) A aplicação deverá permitir consultar as seguintes estatísticas em forma de gráficos:
 - a. Número de jogos jogados por cada um dos 10 melhores jogadores.
 - b. Tempo médio de jogo por jogador (dos 10 melhores Jogadores).
 - c. Pontuação Máxima por jogador (dos 10 melhores Jogadores).
 - d. Percentagem de jogos jogados em cada um dos tipos e variantes de pontuação. (War, Halloween, Smiles, etc..)
 - e. Percentagem de jogos jogados por tema escolhido (War, Halloween, Smiles, etc..)



Nota: Deverá ser possível para cada estatística obter os resultados apresentados em diferentes tipos de gráficos

- 5) A aplicação deverá permitir durante o Jogo fazer “undo” de uma jogada. Poderão ser executados undos sucessivos;
- 6) A aplicação deverá guardada em ficheiro toda a informação sobre os Jogadores e Jogos por estes efetuados, de forma a não se perder essa informação quando se interrompe a execução do programa.

Detalhes de implementação

A aplicação deverá ser implementada na linguagem Java segundo as boas práticas de POO, utilizando os Tipos Abstratos de Dados (TAD) lecionados e os Padrões de Software relevantes. A qualidade do projeto dependerá da qualidade das opções tomadas e implementadas nestas três vertentes.

Em termos de implementação, deverá obedecer-se ao seguinte:

- 1) O gerador de peças aleatórias, as linhas do tabuleiro, o histórico e o ranking deverão de ser modeladas recorrendo a tipo abstratos de dados (ver detalhes nas fases de entrega).
- 2) A aplicação deverá ser implementada segundo o modelo **MVC** e a interface gráfica é desenvolvida em **JavaFX**. O aspeto da visualização fica ao critério dos alunos, no entanto:
 - O Grafismo dos elementos que compõem o tabuleiro de jogo, devem a obedecer a um tema, o tema do jogo deverá ser escolhido em tempo de execução. Existem 3 temas obrigatórios: **Halloween, Smiles, War**
 - O utilizador deverá facilmente visualizar a pontuação obtida até ao momento no jogo.
 - O utilizador deverá poder de uma forma simples selecionar qual a linha em que quer fazer a jogada.
- 3) O aluno deverá usar de uma forma adequada os padrões de Software na resolução do problema apresentado (ver detalhes nas fases de entrega).
- 4) O código deverá ser documentado com recurso ao Javadoc, precedendo todo e qualquer método da descrição do seu propósito (exceção-se métodos construtores, seletores e modificadores). Comentários adicionais deverão ser colocados em trechos de código onde a compreensão por parte de terceiros possa ser menos óbvia.
- 5) Os testes unitários deverão ser realizados recorrendo à framework **JUnit**.

Fases de entrega

O trabalho obedece a uma entrega final e duas entregas intermédias (**Milestone 1 e Milestone 2**). As entregas intermédias têm como objetivo garantir um desenvolvimento do trabalho de forma contínua. A não entrega das fases intermédias, não invalida a entrega final, mas implica a perda total da pontuação referente a estas fases (**20% + 20%**).

Os trabalhos desenvolvidos deverão ser submetidos no Moodle. Os trabalhos entregues no **Milestone 1** e no **Milestone 2** serão avaliados pelo professor de laboratório e discutidos em conjunto na **semana 12**.

A entrega final do projeto permite um atraso até 3 (três) dias, penalizado com 1 (um) valor por cada dia de atraso na nota final (0 a 20 valores).



Milestone 1

- **Objetivo:**
 - Especificar e implementar os tipos de dados abstratos (TADs) para posterior implementação do jogo 3 em Linha.
 - Definir o modelo de análise.
- **Data de entrega:** 16 de Novembro (até às 22:00).
- **Modo de entrega:** O trabalho desenvolvido é obrigatoriamente submetido no Moodle.
- **Conteúdo da entrega:**
 1. Um Documento PDF com a descrição dos TADs desenvolvidos.
 2. Um Documento PDF onde **seja apresentado o diagrama de classes do modelo de análise acompanhado com um breve texto explicativo das opções tomadas.**
 3. Um projeto Netbeans com o package: **pt.ips.pa.model.tads** que irá conter a implementação dos TADs abaixo indicados. O TAD LinhaTres deverá obrigatoriamente ser implementado usando uma estrutura dinâmica.

TAD	Descrição
ConjuntoAleatório	Um conjunto aleatório é uma coleção de itens não repetidos, em que não é possível adicionar ou retirar elementos, e que disponibiliza uma única operação: a operação de visualização aleatória de um item. Ou seja, se tivermos um conjunto aleatório com as cores <i>amarelo</i> , <i>verde</i> e <i>azul</i> , a operação de peek (visualização de um item) , devolve aleatoriamente uma das 3 cores.
LinhaTres	Uma LinhaTres é uma coleção de itens com um comportamento semelhante ao TAD Deque permitindo inserir e retirar elementos no início e no fim da fila. No entanto as operações de remover elementos, removem sempre 3 elementos de uma vez e as operações de getFirst e getLast , devolvem os 3 primeiros elementos e os 3 últimos respetivamente. O TAD LinhaTres tem a noção de capacidade máxima, gerando um erro quando essa é excedida. Disponibiliza ainda um iterador para percorrer a coleção.
Ranking	Um ranking é uma lista <u>ordenada</u> de elementos, em que os elementos são ordenados segundo um critério específico .
Histórico	O Histórico caracteriza-se por ser uma coleção de elementos, com tamanho limitado, onde é apenas possível colocar elementos não sendo possível retirá-los. Os elementos são guardados por ordem de entrada para o histórico e ficam associados a uma data (a data de entrada para o histórico). É possível percorrer sequencialmente todos os elementos, assim como saber quais os elementos de um determinado dia, ou dos últimos dias ou do mês corrente etc. Quando o número de elementos é atingido, os elementos mais antigos são descartados automaticamente.

O projeto apresentado deverá **obrigatoriamente** incluir os respetivos testes utilizando a **JUnit framework**.



Milestone 2

- **Objetivo:**
 - Definir o modelo de desenho.
 - Implementar o jogo.
 - Implementar as estatísticas
 - Implementar as funcionalidades: Undo, Log, Histórico
- **Data de entrega:** 14 de Dezembro (até às 22 horas)
- **Modo de entrega:** O trabalho desenvolvido deve ser obrigatoriamente submetido no Moodle.
- **Conteúdo da entrega:**
 1. Entrega de um documento em PDF com o diagrama das classes de desenho resultante da modelação do sistema a desenvolver e com uma listagem dos padrões de software utilizados, com apresentação da sua concretização na solução proposta. **Sugere-se** a implementação dos seguintes padrões (poderão optar por outros que julgarem mais adequados à solução por vocês proposta):

Padrão	Aplicação
Iterator	Para percorrer as coleções de elementos sequencialmente
Adapter	Para implementar algumas dos TADs
Memento ou Command	Para implementação do Undo no Jogo
Observer	Para Implementar as funcionalidade de Logger.
Strategy	Para implementação das diferentes estratégias de Pontuação. Para implementação do ranking dos jogadores por critérios distintos.

2. Entrega de um projeto em NetBeans, com o código que implementa:

- O Jogo.
- A funcionalidade de Logger.
- A funcionalidade do Ranking de Jogadores
- A funcionalidade do Histórico do Jogador.
- A opção de Undo.
- Os cálculos para as estatísticas
- Persistência

O projeto apresentado deverá incluir os respetivos testes utilizando a JUnit framework.

Nota: Não se pretende nesta fase nenhuma forma de interface com o utilizador.

Entrega Final

- **Objetivo:** Desenvolver a interface gráfica que permita ao utilizador jogar o jogo selecionando o tema para o Jogo e aceder às restantes funcionalidades da aplicação.
- **Data de entrega:** 21 de Janeiro (até às 22 horas)
- **Modo de entrega:** O trabalho desenvolvido é obrigatoriamente submetido no Moodle.
- **Conteúdo da entrega:**
 1. Entrega de um documento em PDF com o diagrama das classes de desenho, a modelação final do sistema desenvolvido e a listagem dos padrões de software utilizados, com apresentação da sua concretização na solução proposta (**versão atualizada do entregue no milestone2**).
 2. Entrega de projeto NetBeans (JavaFX Application) totalmente funcional e com a interface gráfica desenvolvida em JavaFX. Sugere-se a aplicação do padrão de software Abstract Factory para implementação da troca de temas na interface gráfica. Critérios Gerais de Avaliação do Trabalho.



Critérios de Avaliação

Milestone1 (20% do total)	Pontos
Tads – especificação	40
Tads – implementação	120
Modelo de análise	40
TOTAL 1 (pontos)	200
Milestone 2(20% do total)	Pontos
Modelo de Desenho	40
Aplicação dos Padroes	60
Jogo	35
Undo	15
Historico	10
Ranking	10
Estatísticas	20
Persistencia	10
TOTAL 2 (pontos)	200
Avaliação Final – Global(60%)	Pontos
Implementação	
TADs	20
Uso de adequado dos padrões	20
Uso correto do padrão MVC	20
Qualidade geral da arquitetura	20
Funcionalidades / Utilização	
Jogo	30
Ranking	10
Undo	10
Historico	10
Logger	10
Estatísticas	10
Interface Gráfica /usabilidade	40
TOTAL 3	200
Bónus (a acrescentar à nota final antes da discussão)	Pontos
Aplicação adequada de outros padrões, ou aplicação noutros contextos	10
Relógio em modo contínuo	10

Cálculo da Nota do Projeto = ((Milestone1 * 0,2 + Milestone2 * 0,2 + Final * 0,6) + bónus) / 10.

Regras Adicionais

- O trabalho é desenvolvido individualmente, ou por grupos de dois alunos; Será disponibilizado um link no Moodle para a inscrição dos grupos de projeto.
- Projetos não funcionais serão severamente penalizados.
- As diferentes fases de entrega são feitas através do Moodle.
- É obrigatória a discussão oral do trabalho, que é realizada individualmente, dependendo a nota final do projeto do desempenho do aluno na discussão. As notas finais poderão ser por isso diferentes entre elementos do grupo. A falta à discussão oral resulta na classificação com 0 (zero) do aluno em questão.
- Projetos copiados total ou parcialmente serão cotados com 0 (zero), tanto o original como a cópia; Os projetos serão submetidos à plataforma MOSS para deteção de cópias.

(fim de enunciado. Versão 30/12/2015)



ANEXO A – Descrição do Jogo Três em Linha

Introdução

O Jogo proposto é uma versão simplificada do Jogo Halloween, que está disponível no seguinte sítio: <http://www.sheppardsoftware.com/braingames/pumpkins/pumpkins.htm>

O jogo baseia-se num tabuleiro bidimensional que se vai preenchendo com figuras que vão sendo aleatoriamente, tendo como objetivo conseguir juntar e eliminar peças. As peças são eliminadas quando se conseguirem juntar 3 figuras iguais para formar uma linha horizontal.

O jogo termina, quando num dos lados da linha não existirem mais espaços disponíveis para a próxima figura.

Como já foi referido, o grafismo dos elementos que compõem o tabuleiro de jogo, devem obedecer a um tema, o tema do jogo deverá ser escolhido em tempo de execução. Existem três temas obrigatórios: **Halloween**, **Smiles**, **War**. Nesta descrição vai ser utilizado o tema **Halloween**.

Mecânica geral do Jogo

A Figura 1 representa o terreno básico de jogo. Este possui a informação acerca das figuras aleatórias (assinaladas como **Próxima figura**) que representam as próximas figuras/peças a poderem ser utilizadas no terreno do jogo. O terreno do jogo aparece assinalado com cor vermelha e representa uma matriz **8x10**, que por sua vez possui com um conjunto de figuras, também geradas aleatoriamente (assinaladas com cor laranja) que representam o conjunto inicial de figuras.

A barra de seleção de linha (assinalada com fundo azul) possui as duas figuras atuais (uma do lado direito e outra do lado esquerdo) que irão ser colocadas na linha selecionada.

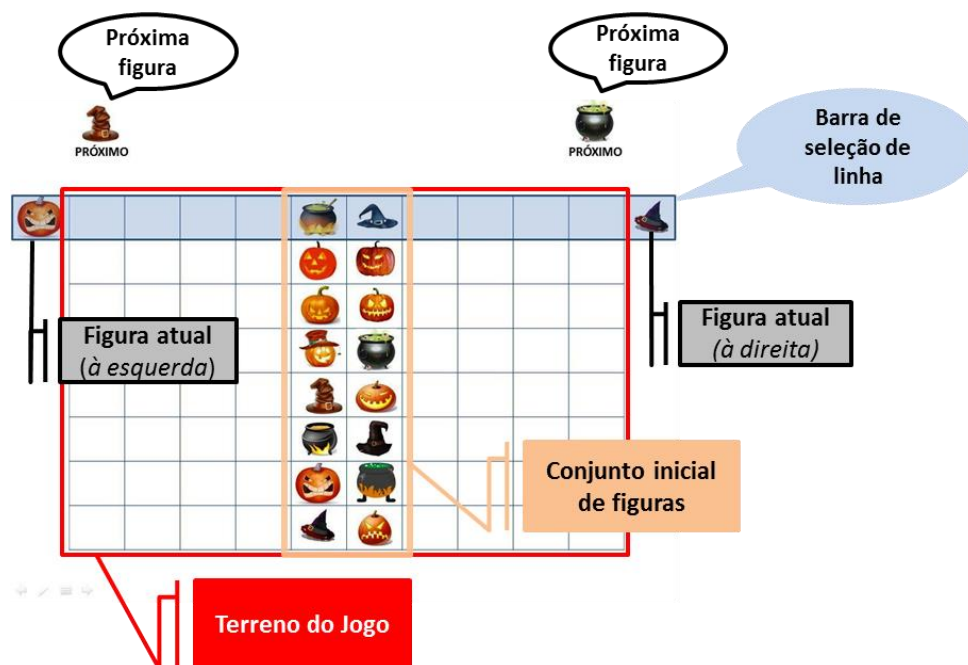


Figura 1: Terreno básico do jogo



A Figura 2 representa a linha escolhida para colocar as duas figuras atuais (representadas na Figura 1).



Figura 2: Escolha de uma linha

As Figuras 3, 4 e 5 representam uma sequência de jogadas que terminam num, **Três em Linha** como se pode verificar na Figura 6 (três figuras iguais).



Figura 3: Sequência de jogadas



Figura 4: Sequência de jogadas

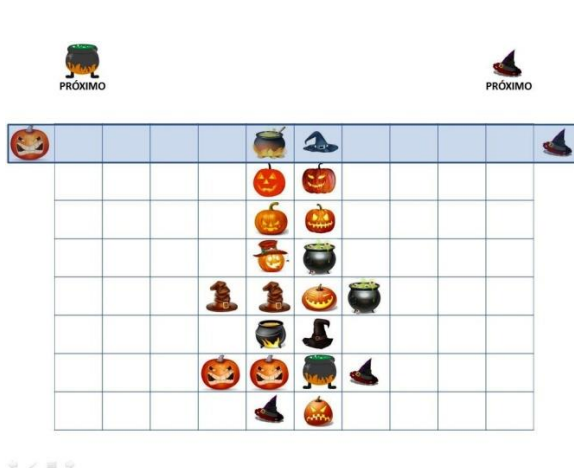


Figura 5: Sequência de jogadas



Figura 6: Três em linha



Sempre que o jogo deteta três figuras iguais em linha elimina-as da respetiva linha.

Também é possível eliminar uma linha completa. Para que tal aconteça tem de ser gerada uma figura considerada especial.

As Figuras 7 e 8 representam uma sequência de jogadas que utilizam uma figura especial para eliminar uma determinada linha, como se pode verificar na Figura 9 (linha completamente eliminada).

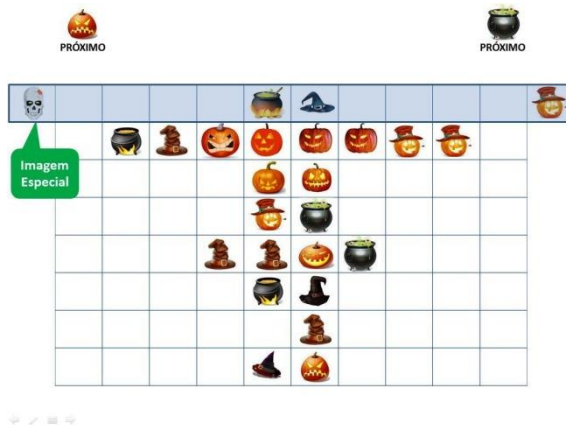


Figura 7: Sequência de jogadas

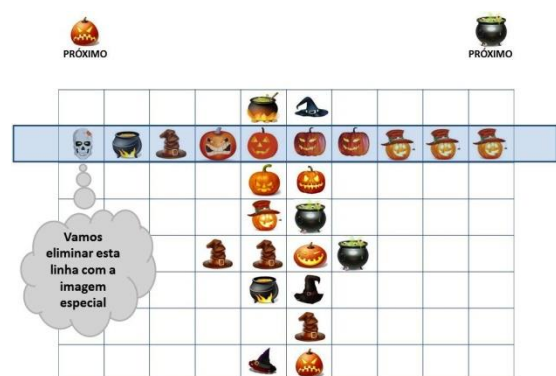


Figura 8: Sequência de jogadas

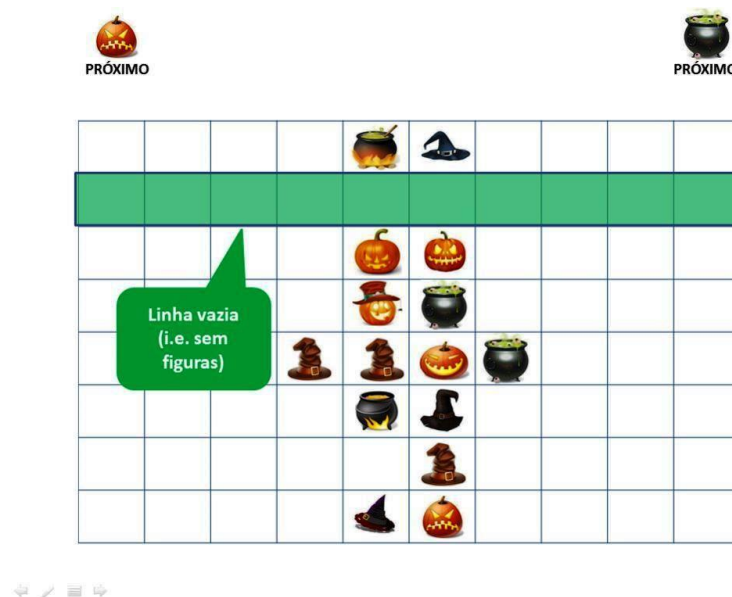


Figura 9: Linha completamente eliminada

Quando um dos lados de uma linha ficar totalmente preenchido o jogo termina.

As Figuras 10 e 11 representam uma sequência de jogadas que resulta no **Fim de um jogo**, Figura 12, porque na linha assinalada com fundo de cor vermelha (não existe espaço disponível para a próxima figura no lado esquerdo dessa linha).



Figura 10: Sequência de jogadas



Figura 11: Sequência de jogadas



Figura 12: Fim de jogo

Pontuação

Existem duas versões de Pontuação:

- **Base** – 20 pontos por cada peça destruída.
- **Corrida** – Começa-se o jogo com 1000 pontos, e ganha-se 10 pontos por cada peça destruída e perde-se um ponto por cada 10 segundos gastos.

Também deve ser considerado que o jogo pode ter duas variantes, uma normal e uma rápida. Na versão **rápida** o jogo tem uma limitação máxima de 20 jogadas.