

Projeto

Objetivo

Você deve utilizar os conhecimentos de orientação a objetos e Java para implementar um jogo computacional com as mesmas características do **Tetris**.

Especificações do Jogo

- 1) O jogo deve conter duas fases distintas, a ser escolhida no início pelo jogador, onde cada uma é caracterizada por um cenário diferente. Você é livre para definir o cenário como quiser, porém inclua obstáculos fixos em pelo menos uma das fases. Além disso, a grade do jogo deve conter no mínimo 10 x 18 elementos (largura x altura).
- 2) Você deve implementar as figuras geométricas (peças) e objetos de cenário, que são os elementos gráficos do jogo.
- 3) A tela do jogo deve conter, além dos elementos anteriores, a pontuação atual, a fase escolhida e a próxima figura geométrica que será colocada no cenário.
- 4) Você deve implementar uma funcionalidade que permita que o jogo seja salvo a qualquer momento, na situação em que estiver, para que seja carregado posteriormente. Para isso, utilize Serialização. A funcionalidade de salvar pode ser chamada através de uma combinação de teclas, como por exemplo "Ctrl+s". A funcionalidade de carregar um jogo salvo anteriormente só poderá estar disponível antes de iniciar um jogo. Pode ser também através de uma combinação de teclas ou um botão na interface gráfica.

Características dos Elementos do Jogo

Figuras geométricas

- Existem sete figuras geométricas (peças) no jogo, que devem ter igual chance de surgirem no cenário. Todas as peças devem ser criadas no topo do cenário, com alinhamento central, que automaticamente começam a descer com velocidade constante. Quando uma peça não puder mais descer (choque vertical com outra peça ou elemento do cenário), sua posição final é definida e uma nova peça é criada no cenário.
- As peças que podem aparecer no jogo são: **I, O, T, S, Z, J, L**. As letras representam o formato da peça (veja exemplos na figura adiante).
- As setas Esquerda e Direita do teclado movem a peça atual para os lados dentro do cenário.
- A seta Baixo do teclado faz com que a peça atual desça mais rapidamente pelo cenário.
- A tecla de Espaço do teclado gera uma rotação na peça atual. A Figura abaixo mostra como deve ser feita a rotação de cada peça. Observe quais grades deverão ser utilizadas para cada movimento de rotação. Caso a peça esteja em uma posição onde não é possível fazer a rotação (muito à direita ou à esquerda), a rotação não deve ser permitida.

J

L

O

S

7

T

Objetos de cenário

- São os elementos que definem o cenário, e podem ser elementos de fundo ou obstáculos para as peças.
- Os obstáculos são elementos fixos que impedem a passagem das peças. Esses elementos podem estar na base do cenário, de modo que a base não seja uma linha totalmente livre, bem como podem ser colocados flutuando pelo cenário. Esses elementos não existem na maioria das versões clássicas do jogo e seu intuito é dificultar um pouco o jogo. Quando uma peça se chocar verticalmente com um elemento deste, a peça é depositada em cima deste elemento fixo. Caso uma linha contendo um obstáculo seja preenchida, apenas os elementos de peças móveis devem ser eliminados da linha, mantendo sempre os elementos fixos.
- Os elementos de fundo servem apenas para desenhar o plano de fundo do cenário.
- Nem obstáculos nem plano de fundo podem se movimentar. Note porém que os elementos de fundo devem ser transponíveis enquanto os obstáculos não.
- Você é livre para definir a aparência destes objetos. Porém, preze pela harmonia e estética dos elementos do jogo.

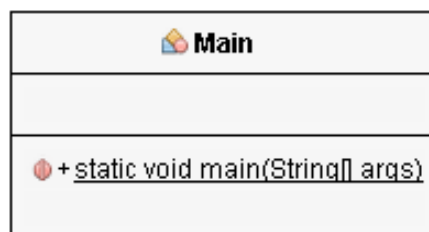
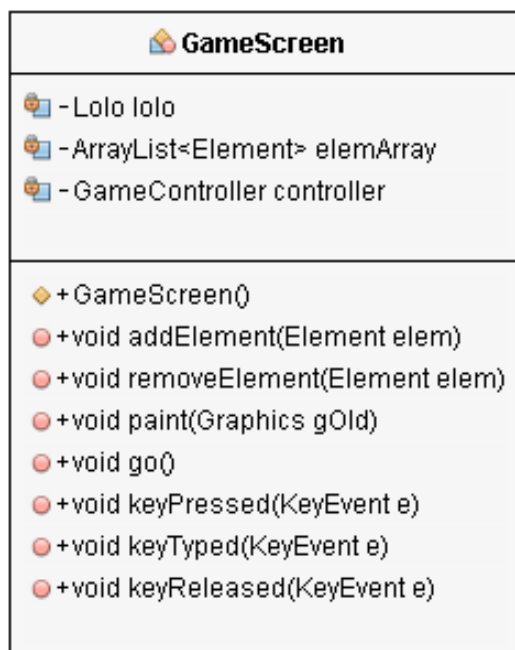
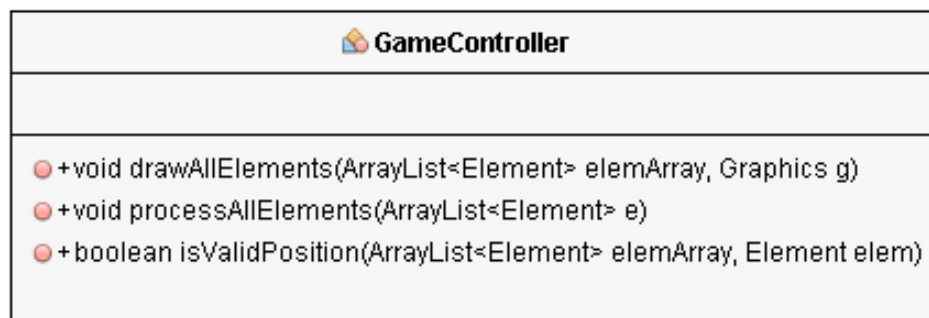
Framework para Desenvolvimento do Projeto

No Moodle está disponível um framework (conjunto de classes) sobre o qual o projeto pode ser desenvolvido. Nele já estão implementadas várias funcionalidades que facilitarão o desenvolvimento.






A tela do jogo foi definida como uma grade de $N \times N$ elementos, sendo que cada elemento da grade poderá conter um objeto do cenário. Cada peça irá ocupar 4 elementos da grade ao mesmo tempo e o movimento das peças é feita em posições inteiras na grade. As peças só poderão se movimentar quando não houver choque entre elementos (entre objetos definidos como não transponíveis).
















Abaixo está o diagrama de classes simplificado do conjunto de classes disponibilizado. Você pode alterar a estrutura destas classes como quiser, se achar conveniente.






Pacote *control*



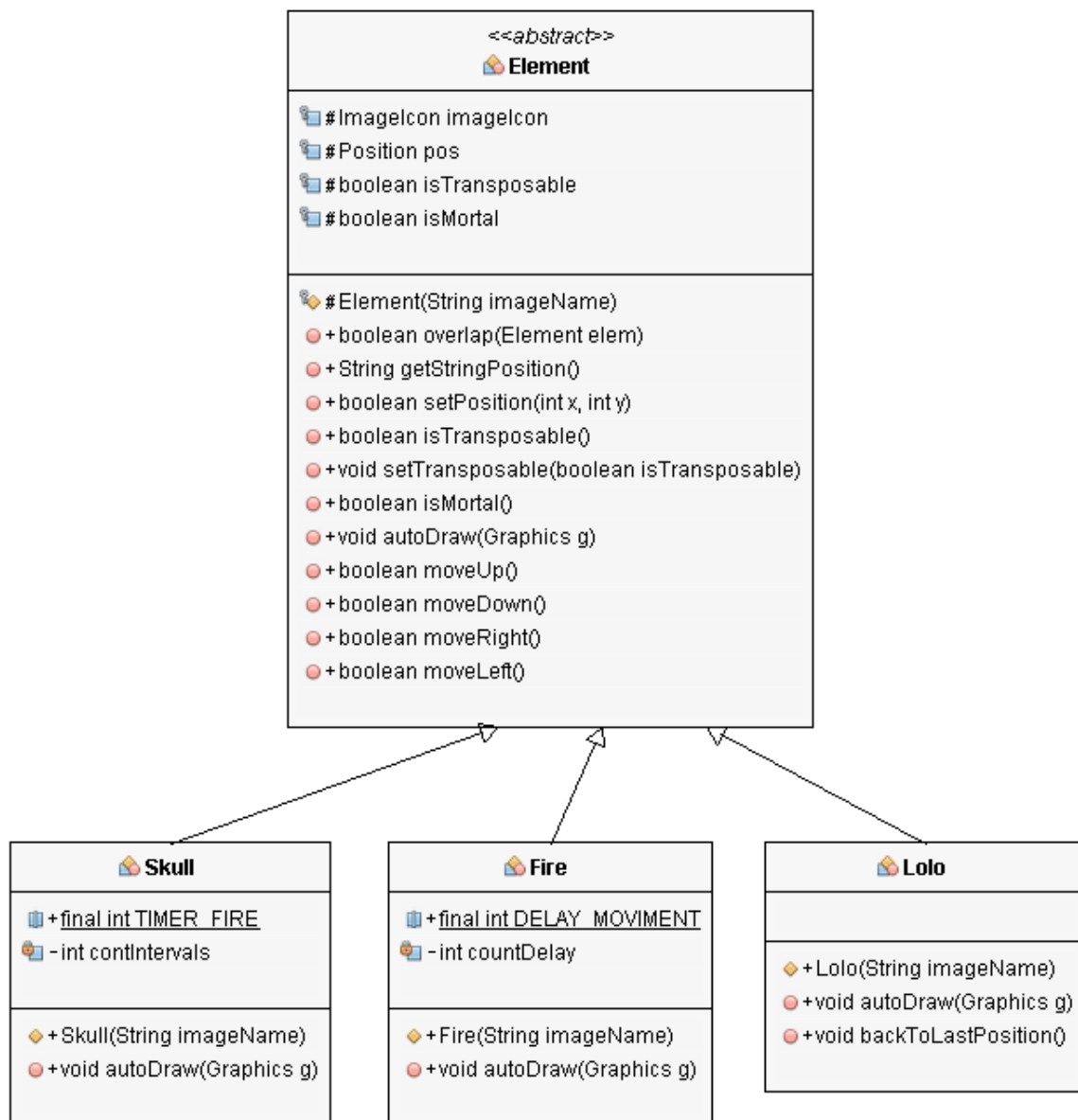
Pacote *utils*

 Drawing
 ~ <u>static GameScreen screen</u>
<ul style="list-style-type: none"> + <u>static GameScreen getGameScreen()</u> + <u>static void setGameScreen(GameScreen newScreen)</u> + <u>static void draw(Graphics g, ImageIcon imagen, double y, double x)</u>

 Position
<ul style="list-style-type: none"> + <u>final int WALK_STEP_DEC_PLACES</u> - int x - int y - int previousX - int previousY
<ul style="list-style-type: none"> + Position(int x, int y) + boolean setPosition(int x, int y) + int getX() + int getY() + boolean comeBack() + boolean moveUp() + boolean moveDown() + boolean moveRight() + boolean moveLeft()

 Consts
<ul style="list-style-type: none"> + <u>final int CELL_SIZE</u> + <u>final int NUM_CELLS</u> + <u>final String IMG_PATH</u> + <u>final int DELAY_SCREEN_UPDATE</u>

Pacote *elements*



Classes Obrigatórias

- 1) O projeto tem uma classe abstrata *Element*, da qual devem ser derivados todos os objetos do jogo. Crie classes para as peças, herdando da classe *Element*. Crie também uma classe *BackgroundElement* para representar os elementos do cenário. Analise como deve ser a estrutura dessas classes para que se aproveite bem os conceitos de POO. Se achar necessário, você pode alterar a estrutura da classe *Element*.
- 2) Crie uma classe *Stage* para representar uma fase. Ela deve conter os objetos do cenário e os métodos que achar necessário.

Dicas

- 1) Você pode utilizar screenshots do próprio jogo e recortá-los em algum editor de imagens (paint, gimp, photoshop, ...) para obter as imagens do jogo, ou obtê-las diretamente na internet.
- 2) Estude com calma a relação entre as classe no framework fornecido. O bom entendimento facilita a criação do seu projeto.
- 3) Teste cada nova funcionalidade. É muito mais difícil identificar erros depois de inserir muitas funcionalidades.
- 4) Seja criativo! Tente colocar mais funcionalidades além das exigidas, tais como animações, sons, etc.

Composição dos Grupos

O trabalho deve ser feito em grupos de 2 a 4 pessoas. Informar por email até o dia 01/04/2018 os componentes dos grupos.

Data limite para entrega: 24/06/2018.

Local de entrega: Moodle.

Forma de entrega:

Um arquivo zip contendo

- 1) Projeto do Netbeans (ou em outro formato, com instruções de compilação)
- 2) Arquivo PDF contendo:
 - Nome e número USP de cada integrante do grupo.
 - Quaisquer esclarecimentos que acharem necessários.
- 3) Documentação do Projeto (*Javadoc*)

A discussão de ideias entre colegas é estimulada, porém a solução do trabalho é individual (do grupo). A identificação de plágio acarretará em nota zero para todos os envolvidos.