SCC0604 – Programação Orientada a Objetos

Projeto

Luiz Eduardo Virgilio da Silva
ICMC, USP

Proposta

- Consulte no Tidia o texto contendo a descrição do projeto
- Lá estão as informações mais atuais sobre o projeto
  - Inclui correções e alterações sobre a versão proposta inicialmente

Proposta

- Criar um jogo com as características do **PAC MAN**, utilizando Java e os conceitos e programação orientada a objetos vistos no curso
- O jogo deve conter 3 fases, cujo layout pode ser definido pelo grupo
  - Grade com mínimo de 15x15 elementos
- Elemento do jogo a serem implementados
  - Pacman
  - Fantasmas
  - Bolinhas comuns (Pac-Dots)
  - Bolinhas de poder (Power Pellets)
  - Frutas
  - Objetos do cenário

Proposta

- A tela principal do jogo também deve ter
  - Pontuação atual do Pacman
  - Quantidade de vidas restantes
  - Fase atual do jogo (1,2,3)

- Também deve ser implementada uma funcionalidade para salvar o jogo a qualquer momento, de modo que o jogo possa ser carregado ao iniciar o aplicativo do Pacman
  - Usar Serialização
  - Teclas de atalho ou botões

Definições

- Pacman
  - Elemento dirigido pelo usuário através teclado (setas e espaço)
  - Velocidade do Pacman é constante em todas as situações
  - Ao comer todas as bolinhas comuns (pac-dots) do cenário, próxima fase é chamada
  - Ao encostar em um fantasma, em condições normais, o Pacman morre
  - Ao comer uma bolinha de poder (power pellet), o Pacman pode comer os fantasmas
  - A cada 10.000 pontos, o Pacman ganha uma vida

Definições

- Fantasmas
  - São quatro fantasmas: Blinky (vermelho), Pinky (Rosa), Inky (Ciano), lyde (Laranja)
  - Cada fantasma tem uma regra de movimentação diferente (adiante). Porém, nenhum fantasma para de se movimentar durante o jogo e todos tem a mesma velocidade em condições normais.
  - Quando o Pacman come uma power pellet, todos os fantasmas mudam de estado: ficam azuis, mais lentos e mudam a direção do movimento
  - Ao comer um fantasma, o pacman ganha 200 pontos no primeiro, 400 no segundo, 800 no terceiro e 1600 no quarto (para um mesmo evento de poder)

Definições

- Fantasmas
  - Blinky (vermelho)
    - Persegue o Pacman. Movimentos baseados na direção atual do Pacman. Porém, adicione um componente aleatório na tomada de decisão para não ficar idêntico.
  - Pinky (Rosa)
    - Tenta encurralar o Pacam. Movimentos paralelos ao Pacman. Na outra direção é aleatório.
  - Inky (Ciano)
    - Movimenta-se aleatoriamente enquanto o Blinky está distante dele (dist > D). Quando Blinky se aproxima (dist < D) comporta-se igual ao Blinky.
  - Clyde (Laranja)
    - Igual ao Blinky quando está distante do Pacman (dist > D). Quando se aproxima (dist < D) passa a movimentar-se aleatoriamente.

Definições

- Bolinhas comuns (Pac-Dots)
  - São as bolinhas de pontuação que preenchem o cenário
  - A fase (cenário) muda quando o Pacman comer todos os elementos deste tipo
  - Cada bolinha vale 10 pontos

Definições

- Bolinhas de poder (Power pellets)
  - Elas dão poder de comer fantasmas ao Pacman
  - Faz com que o estado dos fantasmas mude
  - Quantidade fixa de elementos deste tipo por fase: quatro. Sua localização deve ser próxima aos cantos
  - Cada bolinha dessas vale 50 pontos, independente dos fantasmas que foram ou não capturados pelo Pacman

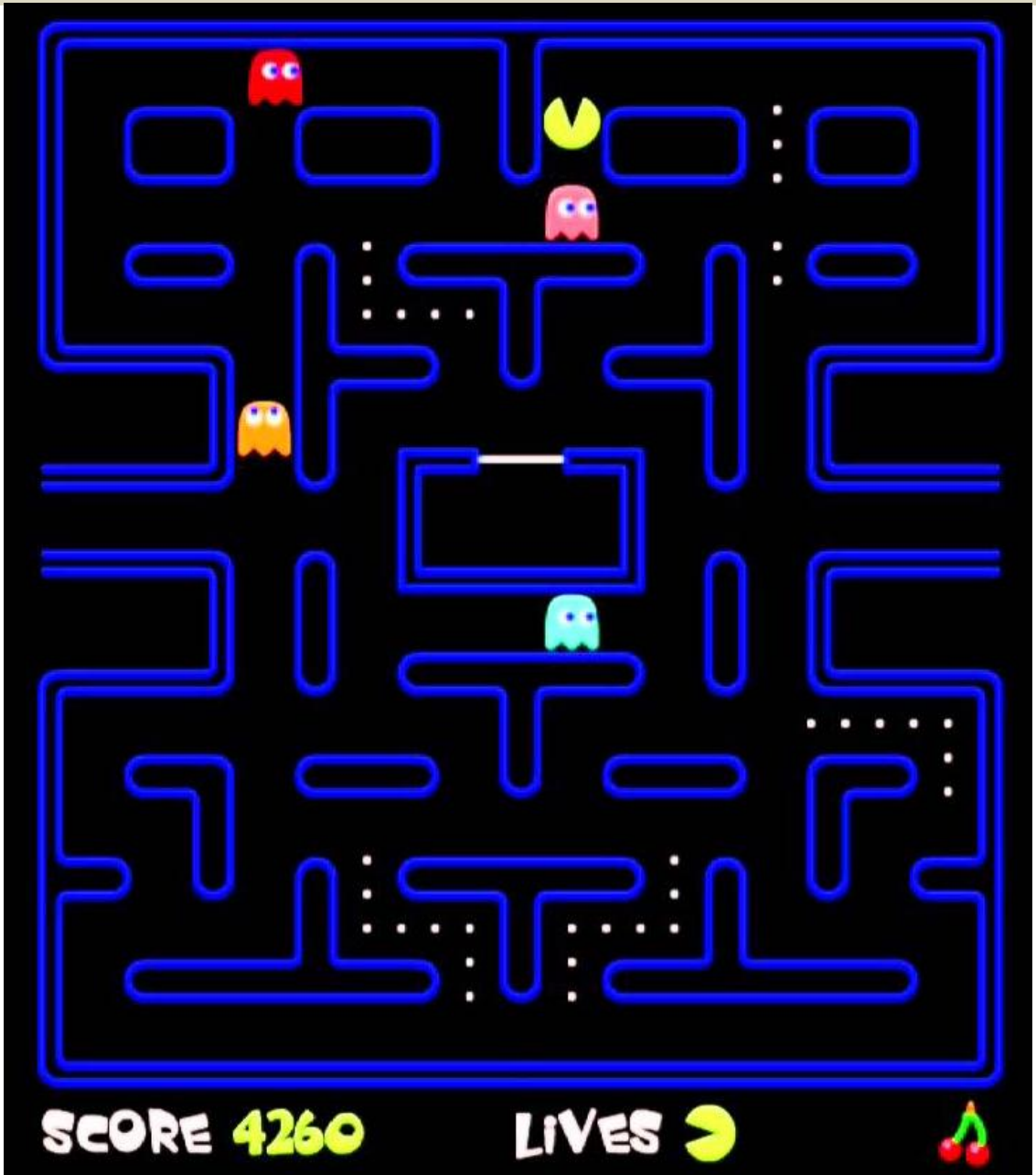
Definições

- Frutas
  - No PAC-MAN original, várias frutas podem aparecer no cenário
  - Vocês devem implementar apenas a cereja e o morango
  - Cerejas valem 100 pontos e morangos valem 300
  - O aparecimento das frutas acontece de tempos em tempos, em posição aleatória no cenário (em uma posição que o Pacman possa passar)
  - A constante de tempo da cereja será 30 segundos e do morango será 45

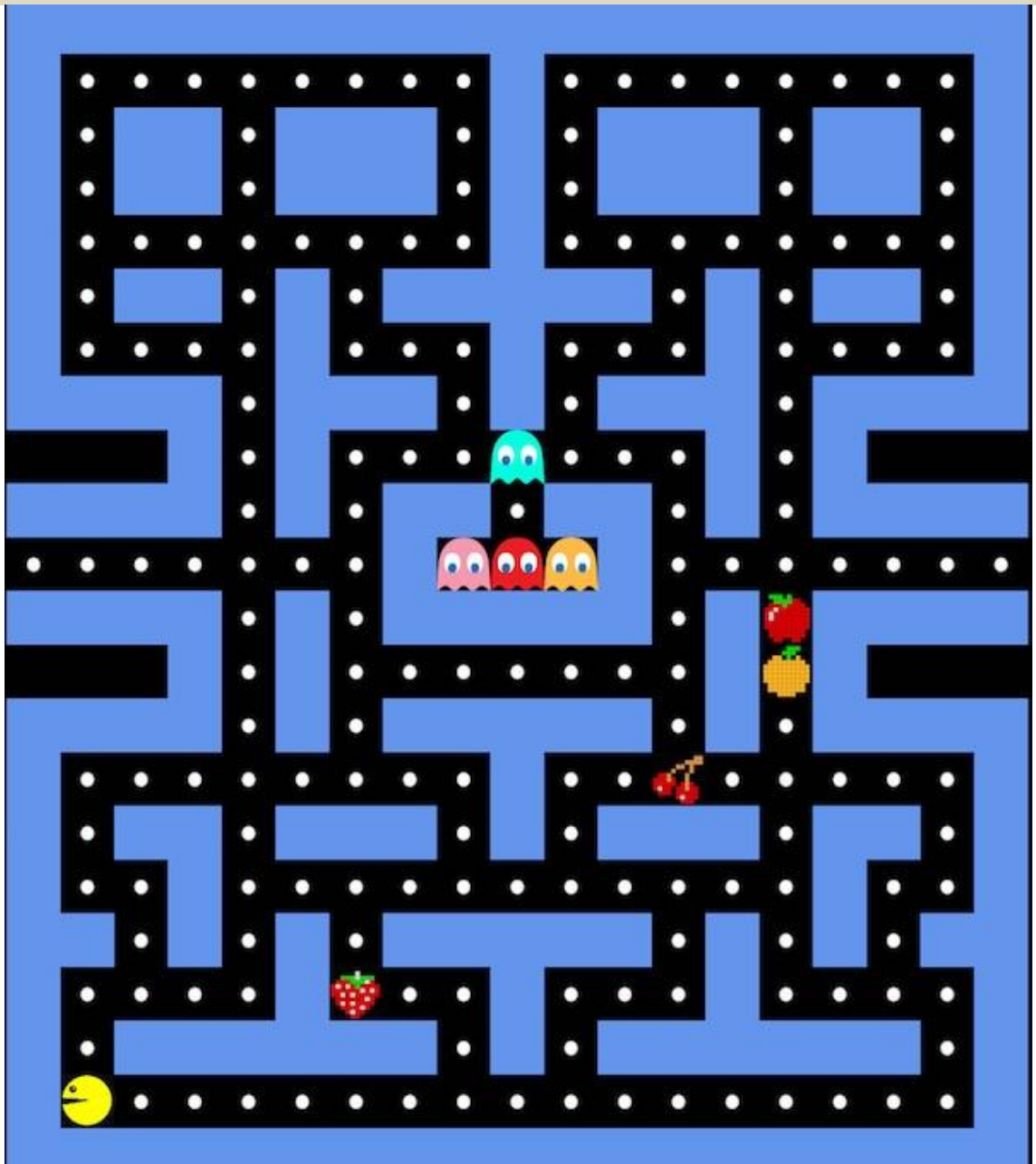
Definições

- Objetos do cenário
  - São os objetos de fundo, que definem os caminhos onde o Pacman e os fantasmas podem andar
  - Nenhum desses objetos pode se movimentar, porém alguns são transponíveis (apenas fundo) e outros não (paredes)
  - Defina esses objetos de forma que o cenário fique bonito e de fácil identificação dos caminhos

Exemplos



Exemplos



Projeto em Java

- Será fornecido um framework em Java (componentes de software) sob o qual o jogo deve ser desenvolvido
  - Atualização de interface gráfica usando Threads
  - Estrutura de classes pré-definidas

- Os grupos deverão implementar
  - Os objetos que o jogo deve conter
  - A maneira como objeto irá se comportar
  - Particularidades da comunicação entre os objetos

Projeto em Java

- A tela do jogo é dividida em forma de grade quadrada
  - Cada elemento do jogo ficará em uma célula da grade
- A movimentação pela grade é feita em passos menores, dando um efeito mais realista
- Contudo, os elementos tem todos o mesmo tamanho e só podem caminhar desde que não choque com outro elemento
  - Objetos transponíveis ou não

Diagrama de Classes

- Pacote control

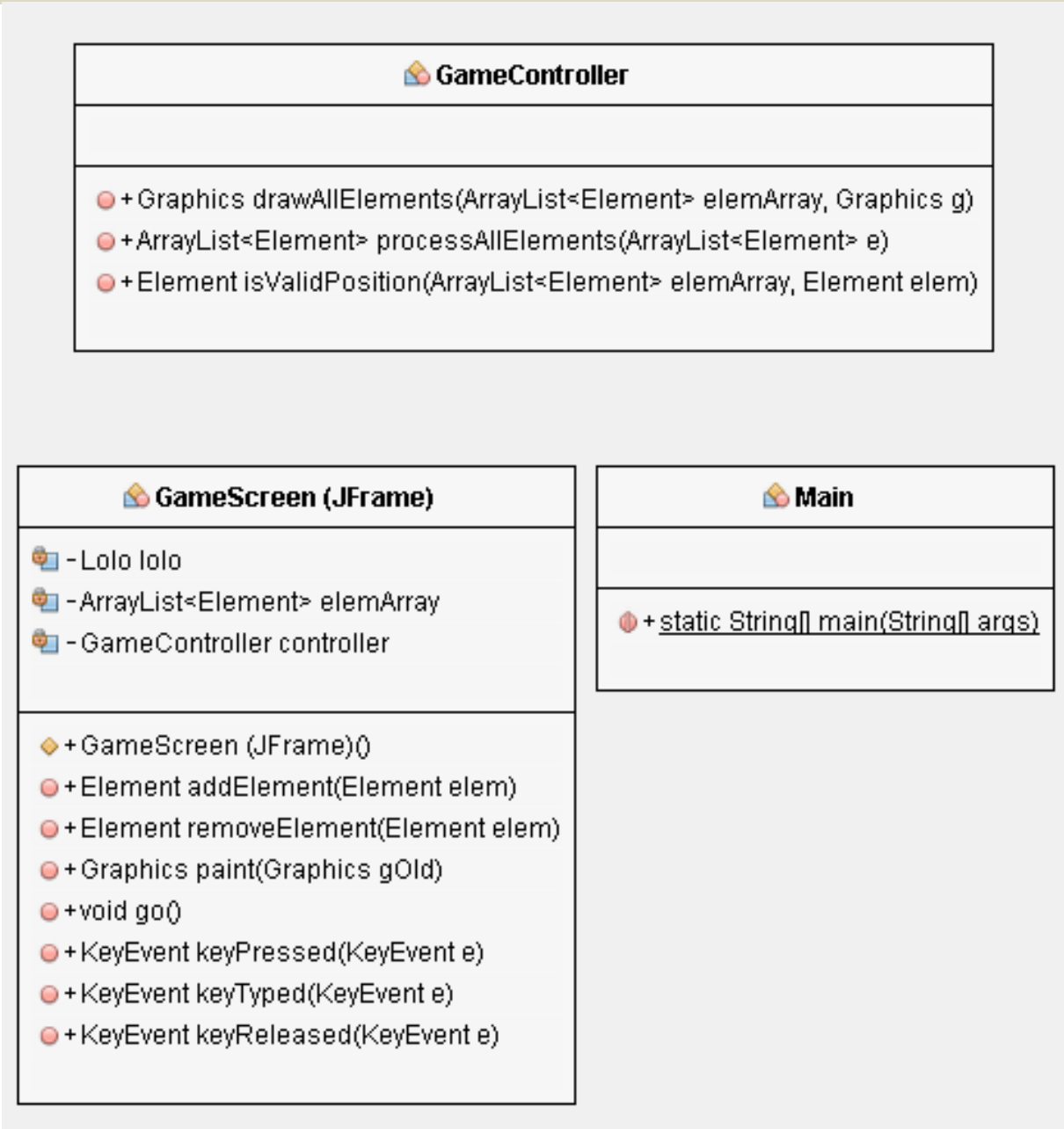


Diagrama de Classes

- Pacote utils

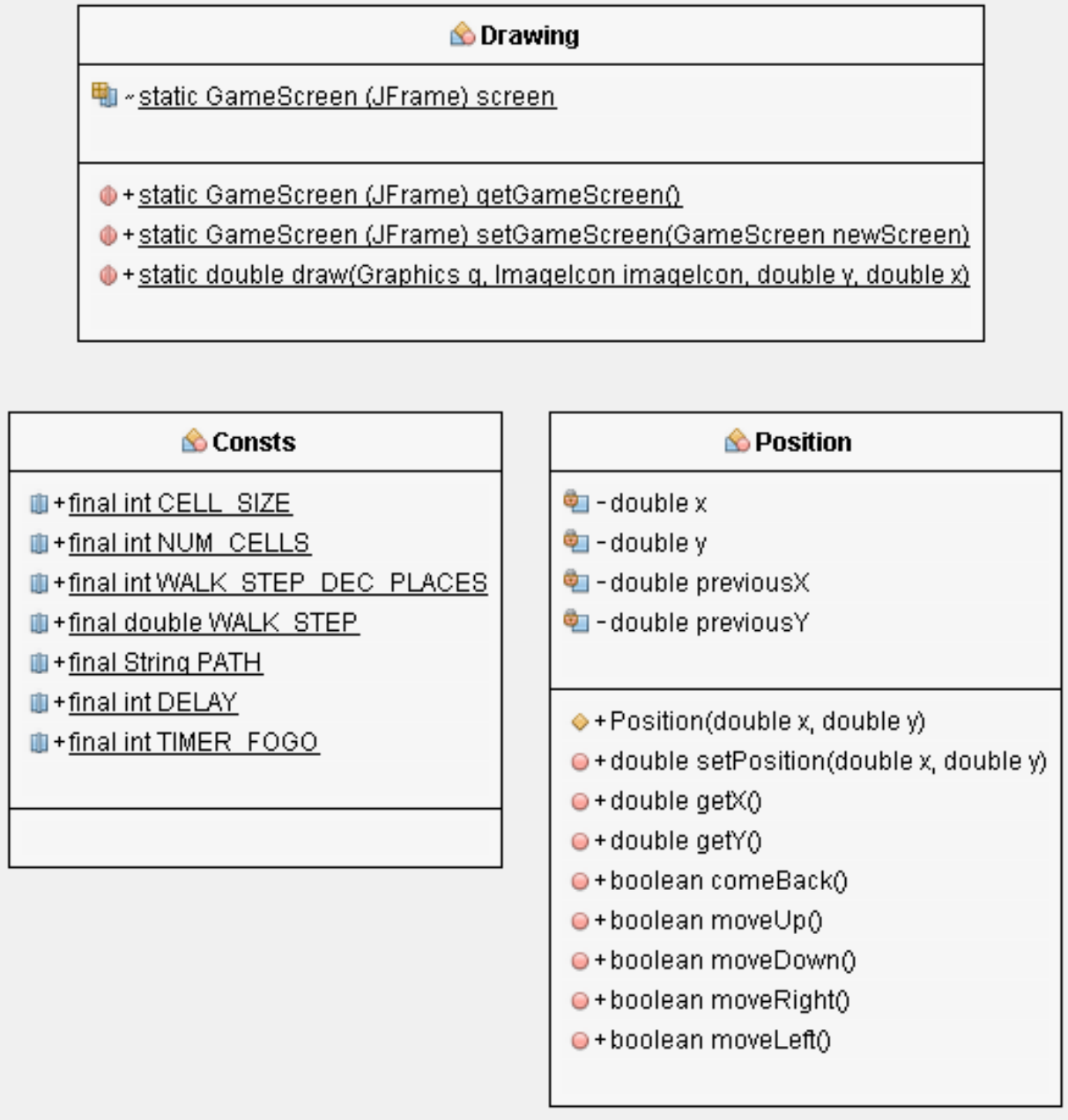


Diagrama de Classes

- Pacote elements

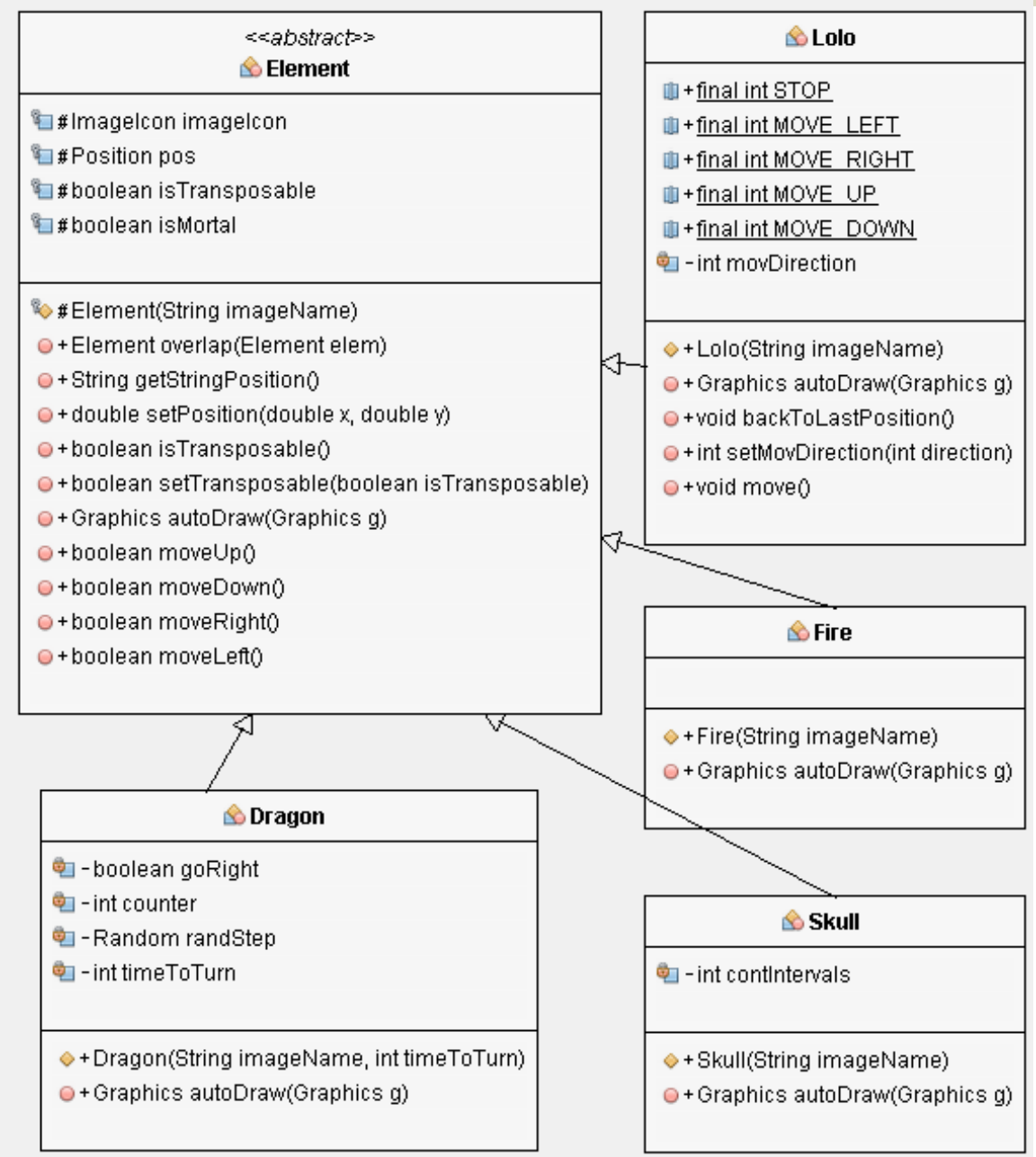


Diagrama de Classes

- Implementação obrigatória
  - Classes que herdam da classe Element
    - Pacman e fantasmas
    - BackgroundElement
    - Pense na melhor forma de implementar a estrutura de classes usando Element (herança, polimorfismo...)
    - Se achar necessário, você pode alterar a estrutura da clasee Element
  - Classe Stage
    - Representa um cenário
    - Contem os objetos de um cenário

Funcionando...

Informações

- Os grupos podem alterar as classes do framework fornecido a vontade
  - É apenas um pontapé inicial

- Dicas
  - Você pode utilizar screenshots do próprio jogo e recortá-los em algum editor de imagens (paint, gedit, photoshop, ...) para obter as imagens do jogo, ou obtê-las na internet.
  - Estude com calma a relação entre as classe. O bom entendimento facilita a criação do seu projeto
  - Teste cada nova funcionalidade. É muito mais difícil identificar erros depois de inserir muitas funcionalidades

Informações

- Grupos
  - 2 a 4 alunos
  - Informar por email os componentes do grupo até dia 26/09/17

- Seja criativo! Tente colocar mais funcionalidades do que o exigido
  - Animações (objetos, transições entre as fases, ...)
  - Sons
  - ...

Informações

- Entrega
  - Até 10/12/17
  - Projeto do NetBeans (ou outro formato, com instruções)
  - PDF contendo
    - Nome e numero USP dos integrantes
    - Diagrama de classes simplificado: heranças, implementações, principais atributos e métodos de cada classe
    - Outros esclarecimentos (se necessário)

Dúvidas?