

**Faculdade de Engenharia da Universidade do Porto**



**Relatório de projecto**

**Sprite Assembler**

**João Carlos Macedo Flores dos Santos – 201106760**

**Wilson da Silva Oliveira - 201109281**

**Turma 5**

**Laboratório de Programação Orientada por Objectos**

**Mestrado Integrado em Engenharia Informática e Computação**

**10 de Junho de 2011**

# 1 Objectivos

## 1.1 Enunciado

*Sprite* no contexto de jogos é uma animação com um conjunto de imagens. Pretende-se desenvolver uma aplicação que permita criar um mapa de *sprites* (*spritemap*) a ser usado por um motor gráfico. A aplicação deverá permitir importar imagens por forma a criar uma matriz de imagens onde constarão os *frames* das *sprites* e gerar a imagem composta final (png), de onde o motor gráfico carregará as animações. A aplicação deverá igualmente exportar um ficheiro xml com informação relativa à configuração das várias animações constantes da matriz para que o motor gráfico saiba de onde as carregar convenientemente. A aplicação deverá ter um *preview* dos *sprites* para que o designer gráfico possa perceber se os *frames* estão corretos, fazendo variar a *frame rate*.

## 1.2 Nossa Interpretação

A aplicação foi desenvolvida em Java usando a biblioteca Swing para desenvolver a interface gráfica. Consiste numa janela principal com uma barra de menu com os menus *File*, *Edit Sprite* e *Help* e várias janelas auxiliares para a criar projectos, importar *sprites*, etc.

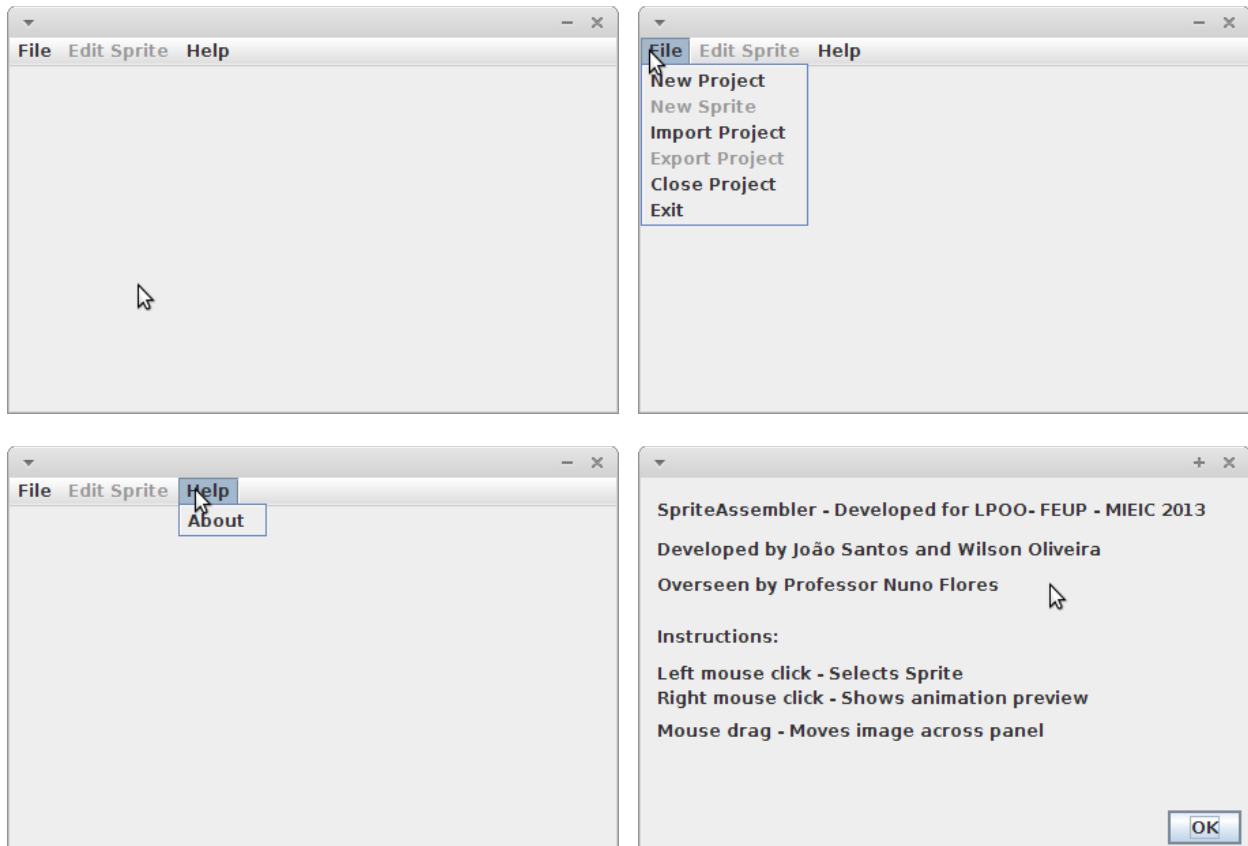
Os *spritemaps*, quando criados no programa, são interpretados como projectos que são constituídos por um ou mais *sprites*. Cada *sprite* pode conter uma ou mais imagens tem uma série de propriedades associadas (nome, *frame rate*, se pode ou não fazer *loop*). Os elementos de cada *sprite* podem ser arrastados pelo utilizador na janela principal de forma a obter o *spritemap* mais eficiente possível em termos de espaço. O programa automaticamente ajusta as imagens para a sua posição apropriada através de um algoritmo de *snapping*.

Quando o utilizador estiver satisfeito com a organização dos *sprites* e as suas propriedades pode então exportar o projecto. Quando isto acontecer a imagem apresentada no painel principal do programa é exportada para um ficheiro .png com o nome do projeto e é criado um ficheiro .xml com a informação do projecto para depois ser facilmente importada para o programa do utilizador.

## 2 Manual de utilização

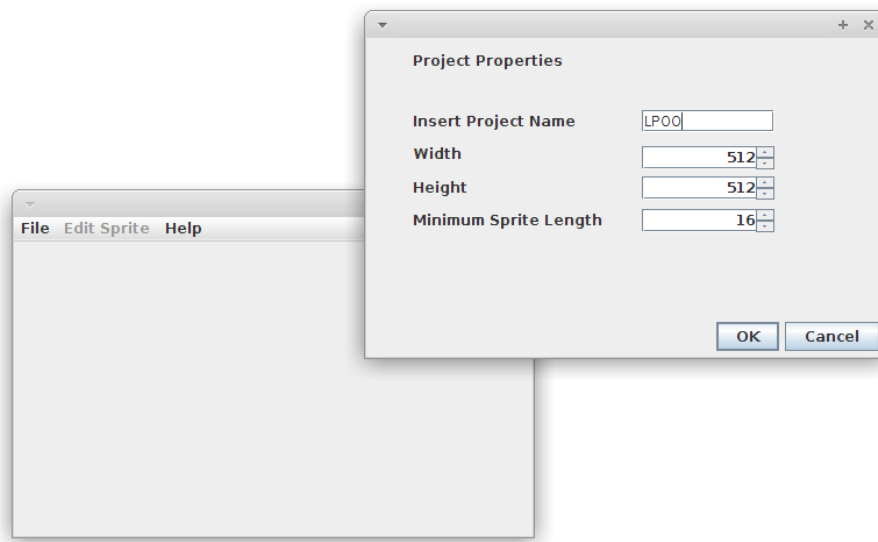
### 2.1 Abrindo a aplicação

Quando a aplicação é aberta é apresentada uma janela simples e vazia com 2 menus disponíveis: *File* e *Help*. O menu *About* apresenta informações e instruções básicas do projecto.

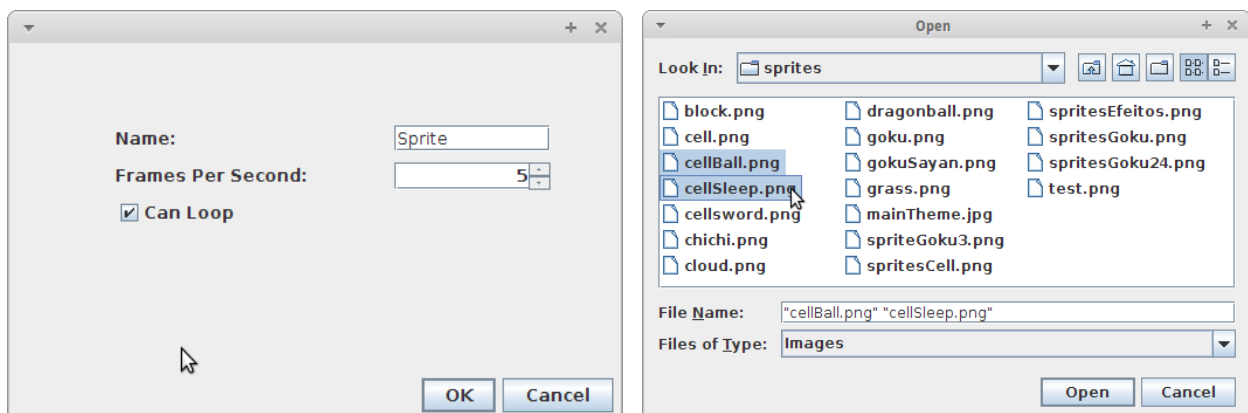


## 2.2 Criando novo projeto

Quando é escolhida a opção *New Project* é apresentada uma nova janela para a configuração do mesmo. O campo *Minimum Sprite Length* deverá ser preenchido com a largura/altura da imagem mais pequena a ser utilizada nesse projecto. Quando alterado qualquer um dos valores numéricos nesta janela é feito automaticamente um ajuste para que o tamanho da janela seja um múltiplo do *Minimum Sprite Length*.



Depois de pressionado *OK* a janela principal será redimensionada para o tamanho especificado pelo utilizador e será apresentada uma nova janela onde o utilizador poderá importar o primeiro *sprite* do projeto. Aí são editadas as propriedades do *sprite* e quando *OK* é pressionado é aberta uma janela onde o utilizador pode seleccionar as imagens pretendidas do seu computador.



## 2.3 Após importar um *sprite*

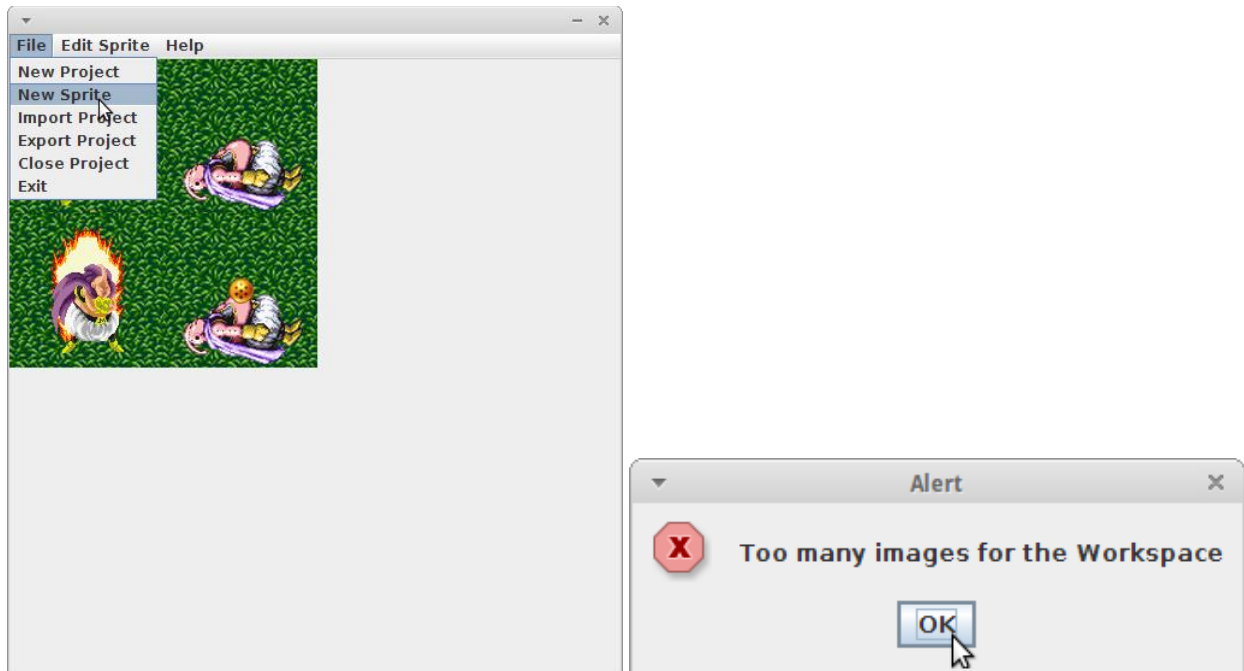
Depois de importar um *sprite* as imagens deste serão inicialmente dispostas de forma a ocupar da melhor forma o espaço disponível. O utilizador poderá então mover as imagens para onde pretende e se as tentar mover para um sítio inválido a imagem selecionada tornar-se-á vermelha para indicar o erro do utilizador. Caso seja largada num sítio inválido será reposicionada na sua última posição válida.



Exemplo de tentativa de posicionar a imagem num lugar inválido

## 2.4 Adicionar *sprites* subsequentes

A qualquer altura o utilizador poderá adicionar mais *sprites* ao programa (desde que o projeto esteja criado). A adição é provisionada da mesma forma que quando o projeto é criado (ver ponto 2.2) e a função pode ser acedida através do menu *File* -> *New Sprite*. Caso não haja espaço no programa para um novo *sprite* ser adicionado será mostrada uma mensagem de erro.



## 2.5 Pré-Visualizar *sprites*

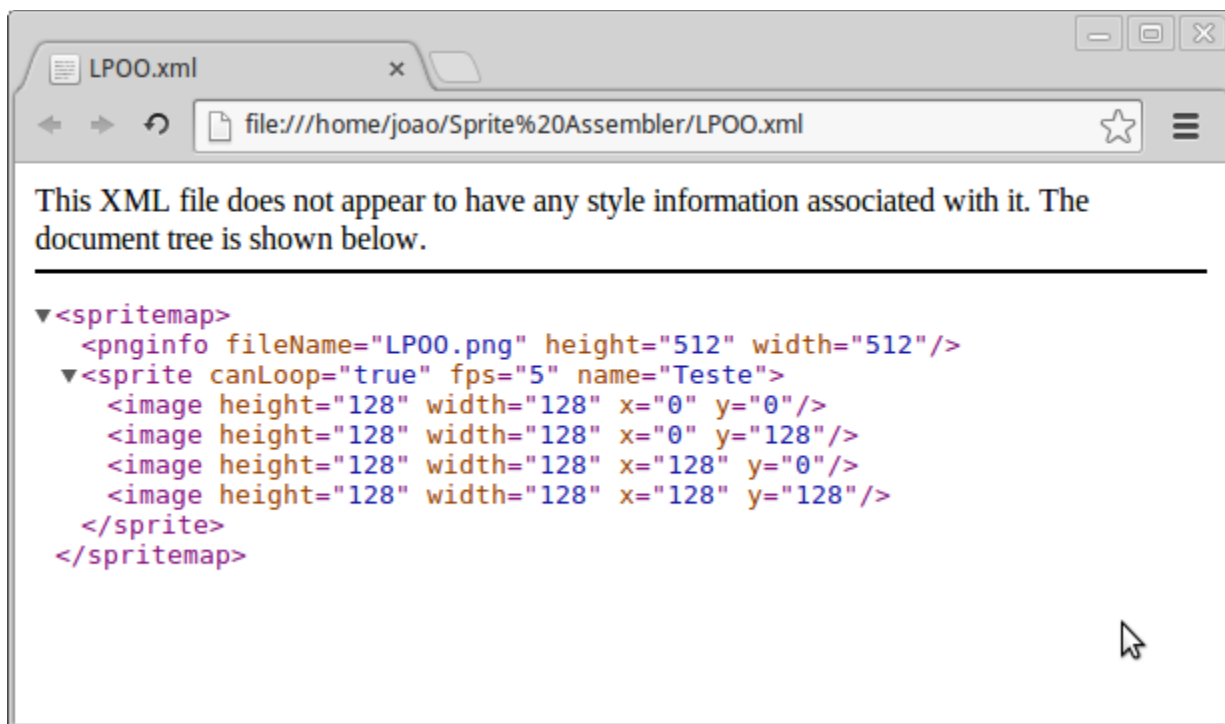
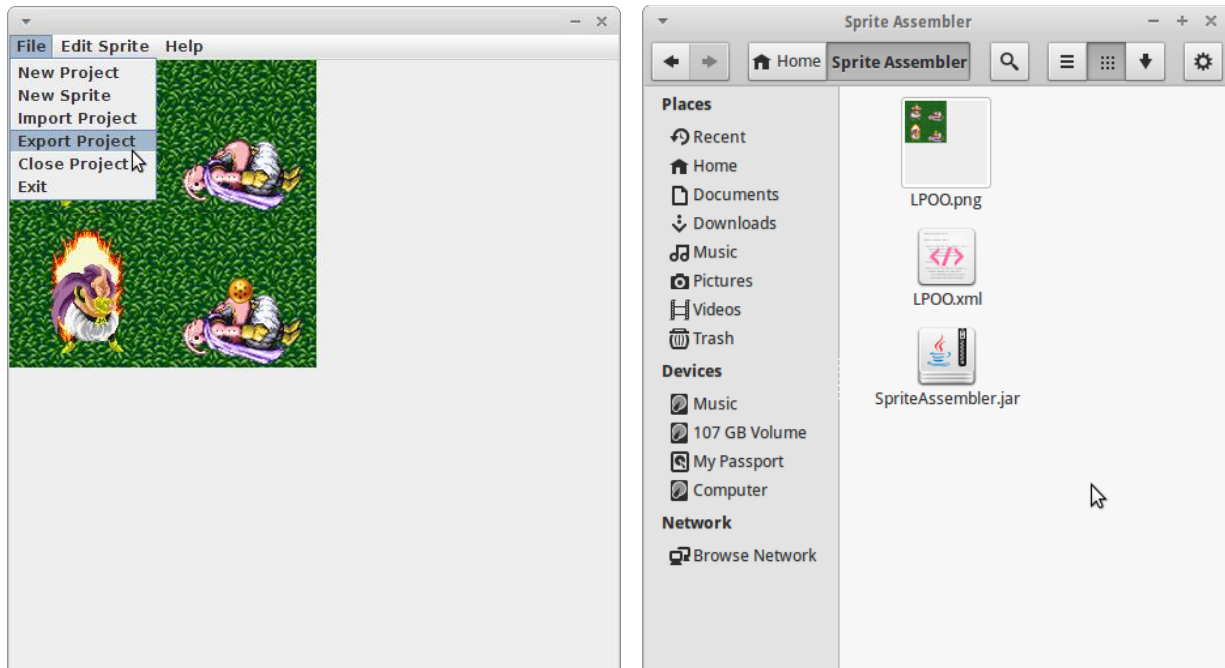
A qualquer momento o utilizador poderá pré-visualizar o resultado da sua *sprite* simplesmente pressionando com o botão direito do rato em cima da *sprite*.

## 2.6 Editar e apagar *sprites*

Após ser inserida, uma *sprite* poderá ser modificada ou apagada recorrendo ao menu *Edit Sprite*. Este menu torna-se disponível depois de clicar na *sprite* desejada.

## 2.7 Exportar o Projeto

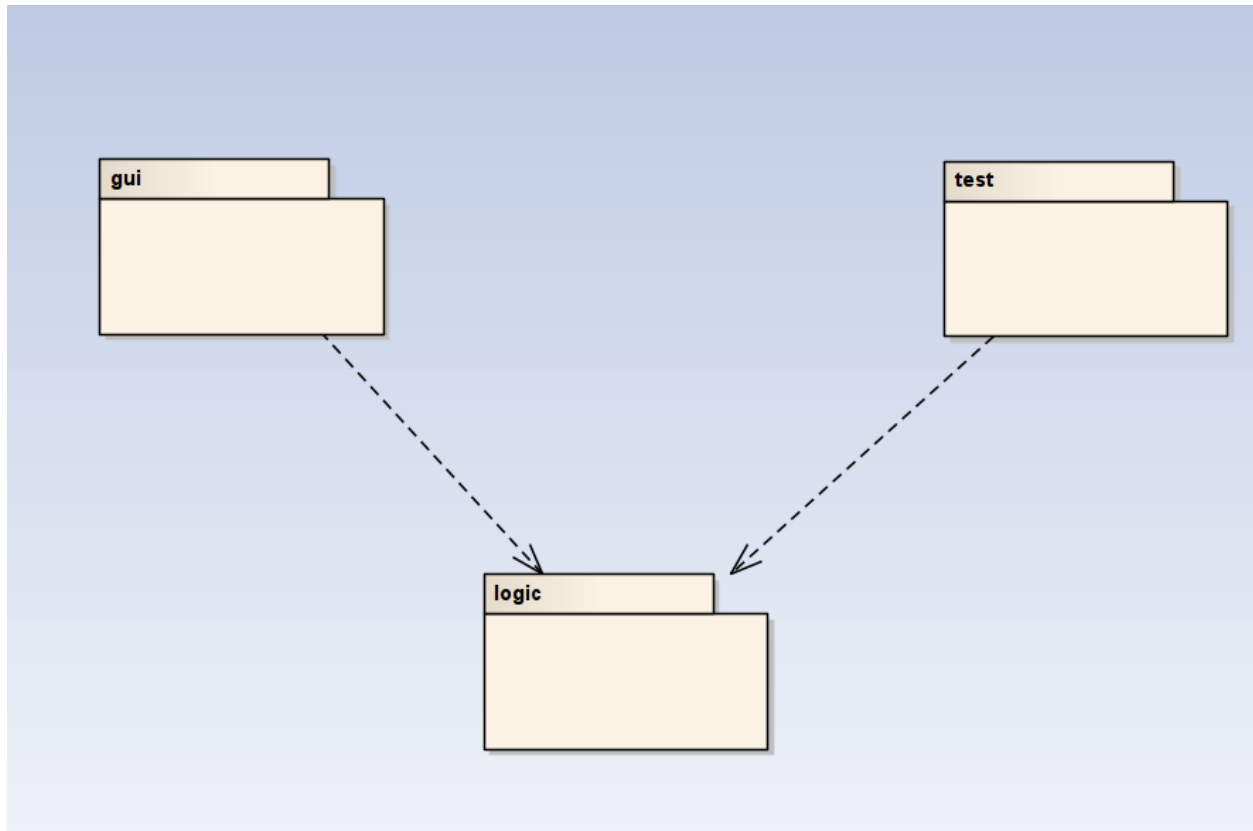
O projecto poderá ser exportado a qualquer momento recorrendo à opção *Export Project* no menu *File*. Os ficheiros resultantes são colocados na pasta onde reside o executável do programa.



Ficheiro xml resultante do projecto de exemplo

## 3 Conceção

### 3.1 Estrutura de *packages*



#### 3.1.1 Package gui

Este package contém todos os elementos da interface com o utilizador bem como algumas funções relativas ao snapping de imagens.

#### 3.1.2 Package logic

Este package contém as classes que gerem o funcionamento geral do programa e os seus componentes, para além das funções de snapping, exportar, etc.

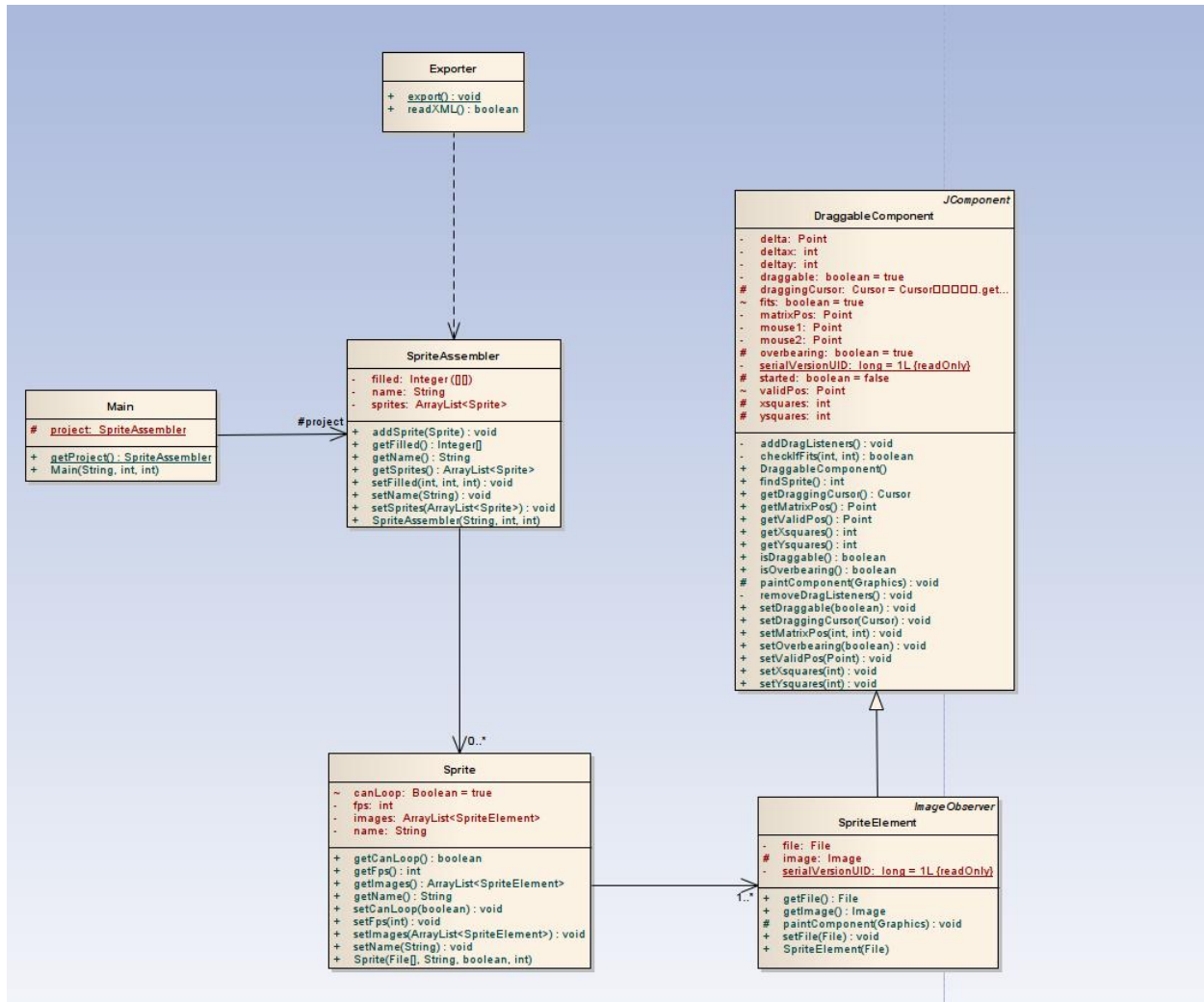
#### 3.1.3 Package test

Este package contém todos os testes unitários e as suas dependências.

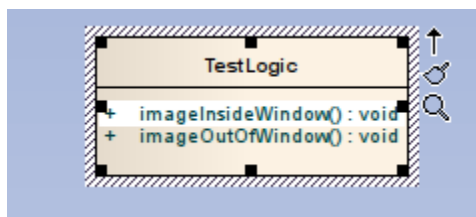


## 3.2 Estrutura de classes

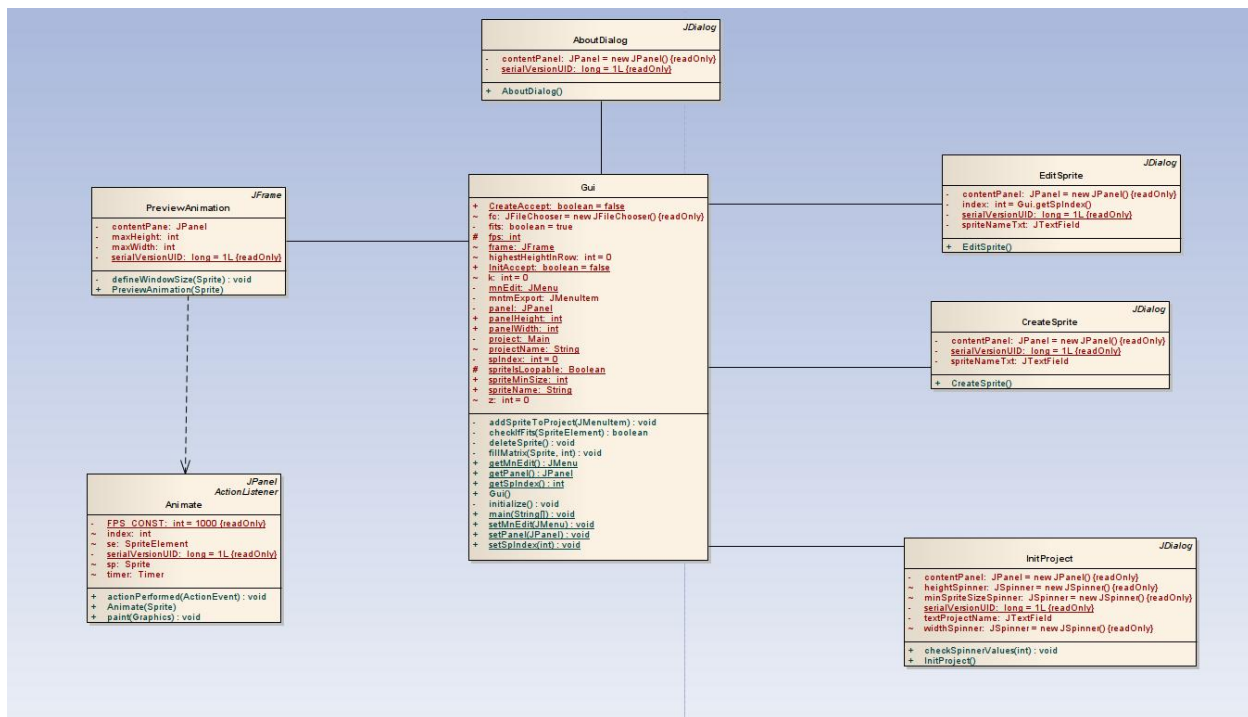
### 3.2.1 Diagrama de classes do package logic



### 3.2.2 Diagrama de classes do package test



### 3.2.3 Diagrama de classes do package gui



## 3.3 Principais problemas

A parte mais difícil do programa foi fazer um algoritmo de snapping eficiente e funcional, algo que só conseguimos obter com a 3ª iteração do algoritmo.

## 4 Testes

Só foram elaborados 2 testes unitários devido a constrangimentos de tempo e a relativa pouca utilidade destes no nosso programa.

### 4.1 ImageOutOfWindow

Este teste verifica se quando a imagem é colocada fora da janela o programa detecta esta situação. Se não a detetasse ia atirar uma exceção *Out of Range* porque iria tentar aceder a uma posição inválida no vector que verifica se um espaço do painel está ocupado.

### 4.2 ImageInsideWindow

Este teste é a versão oposta do anterior e verifica se uma imagem está dentro da janela do programa para poder ser colocada onde se situa o rato.

## 5 Conclusões

*Concluído o projecto, gostaríamos de ter tido mais tempo para implementar a função de guardar e importar projectos e para reestruturar melhor o código.*

*Para além disso estamos satisfeitos com o nosso desempenho e aprendemos bastante sobre interfaces com o utilizador, como gerir o input do rato, etc.*

*Queríamos também agradecer ao Professor Nuno Flores por nos disponibilizar auxílio sempre que precisamos e nos ter dado a oportunidade de fazer um projecto diferente.*

## 6 Referências

*-Draggable components:*

*<http://www.codeproject.com/Articles/116088/Draggable-Components-in-Java-Swing>*

*-How to export a panel to png:*

*<http://stackoverflow.com/questions/5655908/export-jpanel-graphics-to-png-or-gif-or-jpg>*

*-How to create xml in Java:*

*<http://www.mkyong.com/java/how-to-create-xml-file-in-java-dom/>*