**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**

# Objectivos

## 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 animaçoes 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*.

## 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.

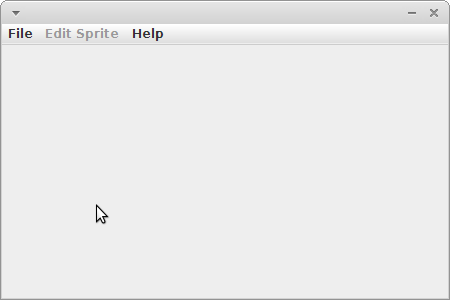
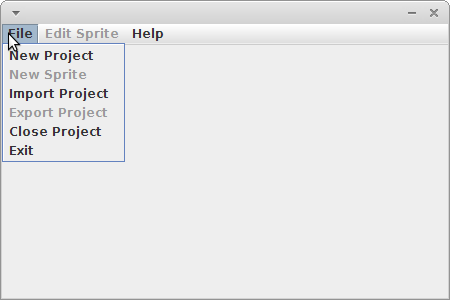
Os *spritemaps*, quando criados no programa, são interpretados como projectos que são constituídos por um ou mais *sprites*. Cada *sprite* pode conter uma ou mais imagens tem uma série de propriedades associadas (nome, *frame rate*, se pode ou não fazer *loop*). Os elementos de cada *sprite* podem ser 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*.

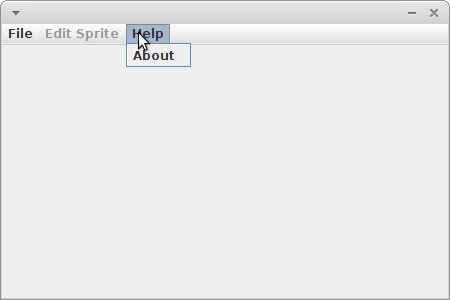
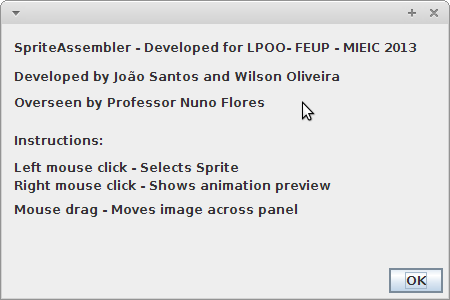
Quando o utilizador estiver satisfeito com a organização dos *sprites* e as suas propriedades pode então exportar o projecto. Quando isto acontecer a imagem apresentada no painel principal do programa é exportada para um ficheiro png com o nome do projeto e é criado um ficheiro xml com a informação do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

# Manual de utilização

## Abrindo a aplicação

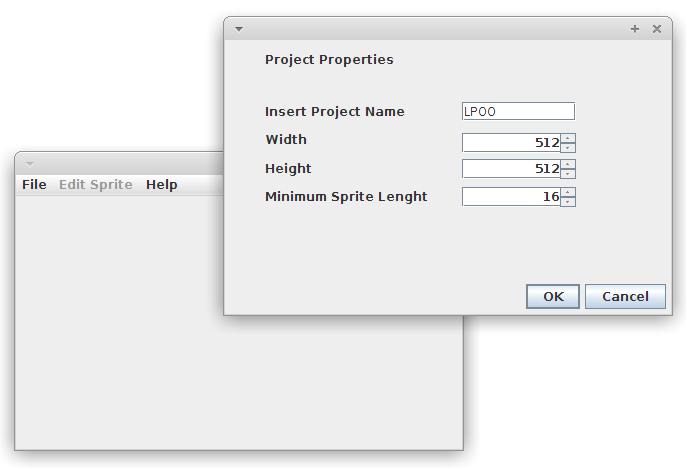
Quando a aplicação é aberta é apresentada uma janela simples e vazia com 2 menus disponíveis: *File* e *Help*. O menu *About* apresenta informações e instruções básicas do projecto.

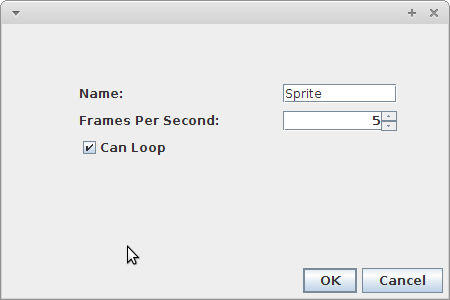
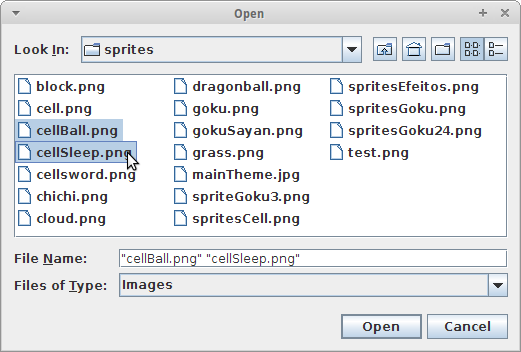
 

## Criando novo projeto

Quando é escolhida a opção *New Project*  é apresentada uma nova janela para a configuração do mesmo. O campo *Minimum Sprite Lenght* 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 Lenght*.



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 precionado éaberta uma janela onde o utilizador pode seleccionar as imagens pretendidas do seu computador.

## 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.

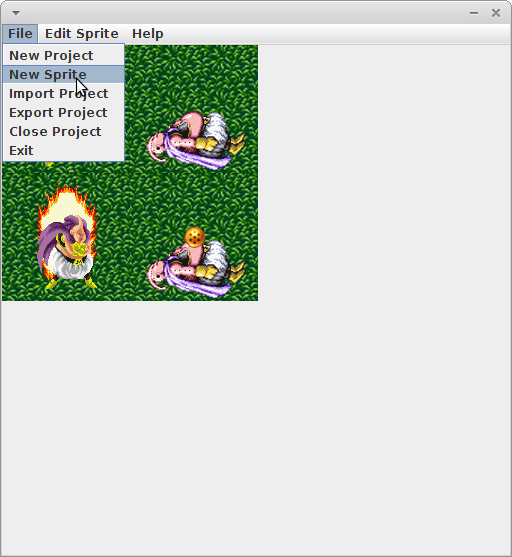
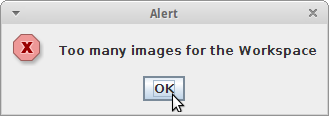
 

*Exemplo de tentativa de posicionar a imagem num lugar inválido*

## 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.

## Pré-Visualizar *sprites*

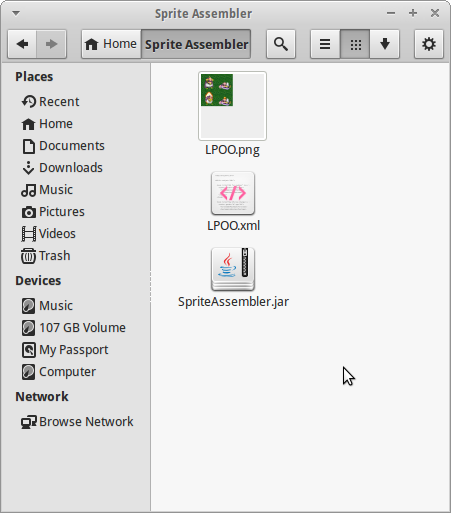
A qualquer momento o utilizador poderá pré-visualizar o resultado da sua *sprite* simplesmente pressionando com o botão direito do rato em cima da *sprite*.

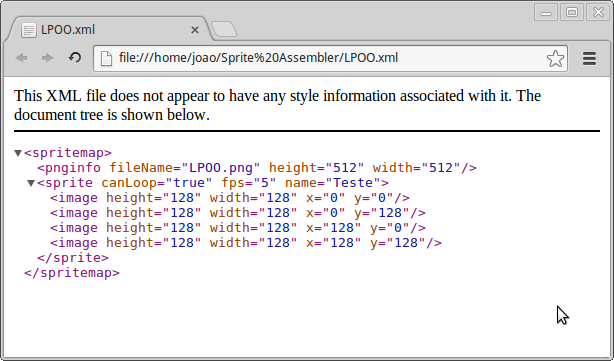
## Editar e apagar *sprites*

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.

## 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*

# Concepção

## Estrutura de *packages*

*Diagrama UML de visão geral da estrutura de packages da aplicação (com packages e dependências).*

*Descrição sumária de cada package (responsabilidades e colaborações), possivelmente na forma de uma tabela.*

*Identificação (possivelmente numa tabela) das bibliotecas foram utilizadas (por exemplo, collections, reflection, rmi, etc., e com que finalidade.*

## Estrutura de classes

*Um ou mais diagramas de classes UML (por package).*

*Descrição sumária das responsabilidades de cada classe (por exemplo na forma de tabela).*

*Opcionalmente, diagramas de objectos para exemplificar instanciação.*

## Comportamento

*Um ou mais diagramas de sequência ou de estados descrevendo os principais comportamentos da aplicação (nomeadamente, o ciclo principal de execução da aplicação).*

## Principais problemas e soluções (padrões) de desenho

*Indicar os padrões de desenho (Singleton, Composite, etc.) e arquitecturais (MVC) utilizados e os respectivos problemas que aqueles resolveram.*

*Indicar outros problemas e soluções relevantes.*

# Testes

*Tabela com uma descrição sumária dos objectivos dos testes implementados em JUnit.*

# Conclusões

*Reflectir sobre o cumprimento dos objectivos.*

*Listar pontos de possível melhoria futura.*

# 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/*