Luke Monaghan **C# Sharp Tool-Sprite Map Generator**

Design Specifications

Designing the Sprite Mapping Tool took a lot of trial and error, both with the Graphical User Interface (GUI), the use of C# and Windows Presentation Foundation (WPF). I have used the WPF elements of Canvas, Image, ListView, Label, Border, Textbox, Checkbox, Menu and Menu items. Use of these elements has been used over three windows, the first being the main tools windows the second being the settings pop up window and the final being the about window. Separation of these has been done to maximize screen space for the main tool. More information on the actual elements can be found below in the user guide.

UML

A UML class diagram has been included in the source files names "ClassDiagram.cd".
If you do not have access to these files, download them from my github.
https://github.com/lukemonaghan/SpriteMapGenerator

Data Structures

The data structure I used is a simple class. This class holds only three variables and a constructor. This is used by pushing a new ListViewItem in to the items of the ListView's I use to hold the loaded sprites and data attached to them.

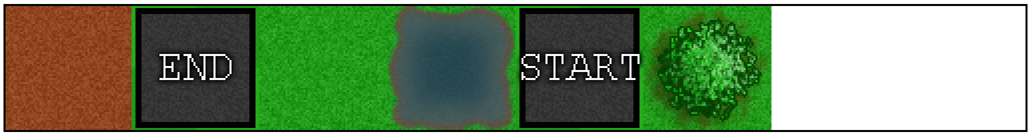
Algorithms

I have only used a simple wrap algorithm on loading of images. When the sprite position and width are greater than the maximum width of the canvas it will wrap down to the next available spot.

XML Design Specifications

My XML file is generated either using UV or Positional Coordinates. This gives the user some options as to how they would export the final product. They both have a standard namespace “SpriteMap”, which contains a namespace “Images”. The Images namespace has attributes of Type being either “Pos” or “UV” a count of the amount of images in the file, an URI to the Image.png, a width and height.The next namespace “Image” is used to contain each individual images data. These attributes do change depending on what type of export was picked. For a Positional export it will contain an id, Name, x, y, width and height. For an UV export it will contain an id, Name, uMin, uMax, vMin and vMax.

The following image is used in the below XML examples.



Positional XML file will look like

```
<?xml version="1.0" encoding="utf-8"?>
<SpriteMap>
  <Images Type="Pos" count="6" Image="Image.png" Width="512" Height="64">
    <Image id="0" Name="Dirt" x="0" y="0" width="64" height="64" />
    <Image id="1" Name="End" x="64" y="0" width="64" height="64" />
    <Image id="2" Name="Floor" x="128" y="0" width="64" height="64" />
    <Image id="3" Name="Puddle" x="192" y="0" width="64" height="64" />
    <Image id="4" Name="Start" x="256" y="0" width="64" height="64" />
    <Image id="5" Name="Tree" x="320" y="0" width="64" height="64" />
  </Images>
</SpriteMap>
```

UV XML file will look like

```
<?xml version="1.0" encoding="utf-8"?>
<SpriteMap>
  <Images Type="UV" count="6" Image="Image.png" Width="512" Height="64">
    <Image id="0" Name="Dirt" uMin="0" uMax="0.125" vMin="0" vMax="1" />
    <Image id="1" Name="End" uMin="0.125" uMax="0.25" vMin="0" vMax="1" />
    <Image id="2" Name="Floor" uMin="0.25" uMax="0.375" vMin="0" vMax="1" />
    <Image id="3" Name="Puddle" uMin="0.375" uMax="0.5" vMin="0" vMax="1" />
    <Image id="4" Name="Start" uMin="0.5" uMax="0.625" vMin="0" vMax="1" />
    <Image id="5" Name="Tree" uMin="0.625" uMax="0.75" vMin="0" vMax="1" />
  </Images>
</SpriteMap>
```

Tests and GUI documentation

Tests

- Loading same sized images
- Loading images of different sizes
- Loading multiple Images
- Loading single images
- Exporting PNG
- Exporting XML
- Moving images
- Moving canvas
- Adding loaded images to canvas
- Snapping of images on canvas
- Adjusting snapping of grid
- Adjusting Size of canvas
- Adjusting max sizes of canvas

The above tests all work to at least minimal functionality. Some need more work and fixing such as Moving canvas which snaps to its original position before moving again. More testing will be used in time when I revisit this project.

Resolution

Besides fixing a few minor logic errors the project is working as expected. General use of the program itself could also be revamped to create a more user friendly experience.

User Guide

After opening the program a window will appear, this window will contain two list, a canvas and a menu system. The menu system at the top of the window holds the menu items of File, Export and Help. These will be your main controls of this program. Clicking one of the three will present you with a sub menu.

File will have the options of Load, Save, Import Image, New, Clear, Settings and Exit.
Export will have XML Pos and XML UV.
Help will contain About.

Importing images is as simple as Clicking File, Clicking Import Image, Going to the directory of the image you wish to import and clicking open. This will add the image to the canvas and import it to the Images List for further use.

Exporting a PNG image with a XML is as follows. Click the menu option Export, Click either the XML Pos or XML UV, the first will export relative Positional coordinates to the top left of the canvas, UV will export UV coordinates relative to the top left. The use of Positional is left only for legacy or hardcoding methods. You will almost always want to export as UV.

Changing of settings is done by clicking File, Settings. You will be presented by a pop up window holding options to change and activate snapping of images, the current width and height of the canvas and the maximum width and height of the canvas. These will usually not be used unless you want a specific sized image.

You are able to move the canvas by holding right click inside the canvas area and moving the mouse. You are able to move the images inside the canvas by holding left click on top of an image and moving the mouse. This will have snapping applied if it has been set and turned on.