Texture Atlas Generator

This cross-platform(Windows and Linux) command line texture atlas generator reads .png files from a given folder and outputs a texture atlas png file and its metadat json file.

Algorithm

This application uses *Binary Tree Bin Packing Algorithm, Dynamic Growing Rectangle* version. Instead of trying to guess the optimal width and height for packing all of our images into, we start with a small canvas, just big enough for the first image, and then grow the canvas towards to right side or to down side when there is not enough room for the next image. With this dynamic growing algorithm the unused area is very minimized.

http://codeincomplete.com/posts/bin-packing

Before packing, images are sorted by their max side, max(width, height) in descendent order, so the one who has largest side get packed first, this is a proven mechanism to achieve the most pleasing square-ish result and minimal whitespace.

Usage:

On Windows: atlas_generator.exe <image folder>
On Linux: ./atlas generator <image folder>

If image folder path contains space, please put the path in double.

Output:

The texture atlas png and its metadat json file will be generated in the working directory.



The generated ison file will have a format like this:

```
{
    "Metadata": [
             "name": "furniture bed.png",
             "x": 0,
             "y": 0,
             "width": 284,
             "height": 112
        },
             "name": "beach-chair-icon.png",
             "x": 0,
             "y": 112,
             "width": 256,
             "height": 256
        },
        //...
    ]
}
```

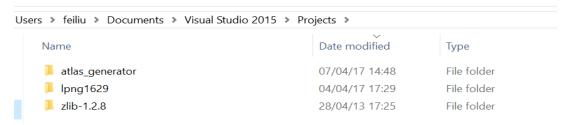
Building Instructions:

On Windows:

First compile the libping and zlib as Static Library for Visual Studio.

The steps I compiled libping and zlib as Static Library for Visual Studio 2015 x86 Release version:

- Download the source libpng-1.6.29.zip and zlib-1.2.8.zip
- Unzip in Visual Studio Projects folder ('atlas_generator' project also located here)
 Documents\Visual Studio 2015\Projects



• Start Visual Studio 2015, open the solution file vstudio.sln in Documents\Visual Studio 2015\Projects\lpng1629\projects\vstudio

Note: Click a source file in zlib project, be sure you can open it, the lpng1629 vstudio project bunds its zlib project to version zlib-1.2.8.

- Open the Solution vstudio's Properties. Set all of the fields shown to "Release".
- Open zlib project's Properties.
 - o Set the configuration type to Static library (.lib).
 - o Set the Runtime Library to Multi-Threaded (/MT).
 - o Set the Target Machine x86.
 - On the Librarian -> Command Line section copy the output file name: "C:\Users\feiliu\Documents\Visual Studio 2015\Projects\lipng1629\projects\vstudio\Release\zlib.lib".
- Open libping project's Properties.
 - O Set Additional Dependencies to the output file nameyou just copied: C:\Users\feiliu\Documents\Visual Studio 2015\Projects\lpng1629\projects\vstudio\Release\zlib.lib

- o Set the Link Library Dependencies to Yes.
- o Set the configuration type to Static library (.lib).
- o Set the Runtime Library to Multi-Threaded (/MT)..
- o Set the Target Machine x86.

Then, open 'atlas generator' project's Properties

- Add in Additional Include Directories:
 - ..\..\lpng1629; ..\thirdparty_common\include; thirdparty_win\include; src;
- Add in Additional Libraries Directories:
 - ..\..\lpng1629\projects\vstudio\Release;
- Add in Linker Input:

libpng16.lib; zlib.lib;

Finally build 'atlas_generator' project Release with x86.

On Linux:

Use system package manager to install libpng and zlib:

sudo apt-get install libpng12-dev

The zlib will be installed automatically when you install libpng12-dev.

Build 'atlas generator' project: On command line in 'UbuntuProject' folder run: make

Third Party Dependencies:

They are: libpng, zlib, dirent, and rapidjson.

- libpng, zlib is used for reading and writing png both for Windows and Linux.
- dirent is used on Windows for reading folder(Linux has 'dirent').

 It's a header only library, it's in folder <code>VisualStudioProject\thirdparty_win\include</code>.
- rapidjson is used for output json both for Windows and Linux. It's a header only library, it's in folder thirdparty_common\include.