

Aspershine Texture Packer

User Guide & Documentation

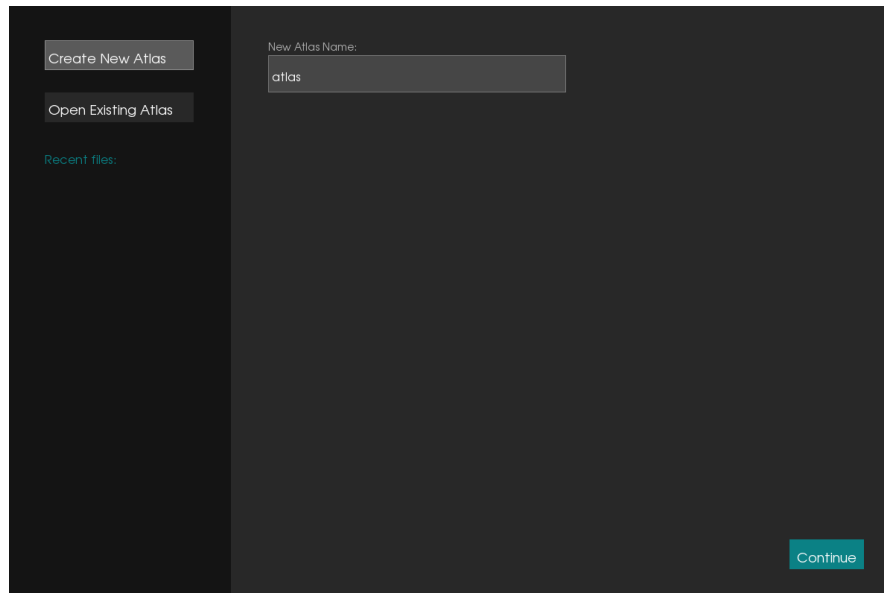
Aspershine is a light-weight texture packing application that allows the user to create atlases from images while.

The motivation behind this project comes from the significant performance increase of using atlases (packed textures) instead of individual textures for 2D rendering. Moving data back and forth between the RAM and a GPU's VRAM is slow, thus loading a whole atlas of textures is preferred. It's important that atlases do not exceed the maximum texture size of the GPU, otherwise splitting will occur resulting in diminished performance.

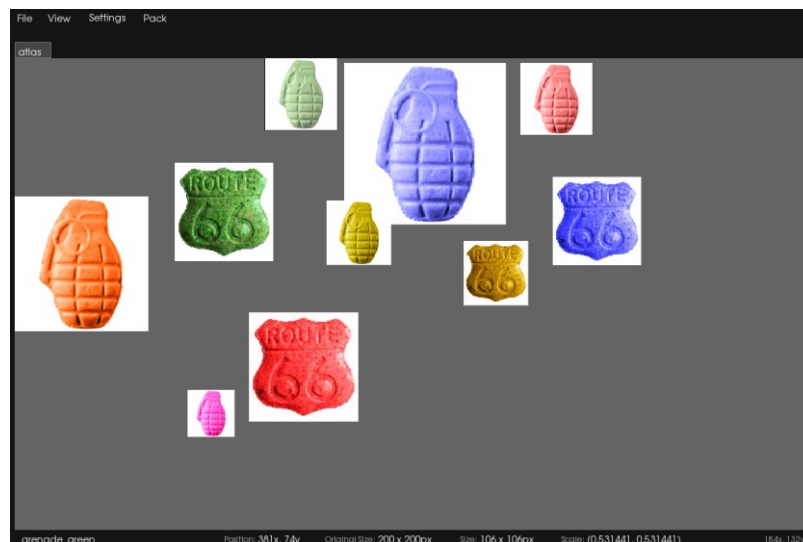
Although texture packing software exists for a long time, Aspershine has the unique feature of including a map file in the export result which drastically helps splitting the atlas back in to individual textures when rendering.

User Guide

When opening **Aspershine** you will be greeted by main menu, that allows you to **create a new** atlas from scratch or **open an existing** one. Recent projects are also shown for quick access.



In order to create a new atlas, all you have to do is give it a name and click **Continue**.



Whether you created a new atlas or opened an existing one, the editor will open. It is comprised of the following elements:

- File, View, Settings and Pack buttons
- List of opened tabs
- A canvas containing the images
- A details bar referring to the selected image



The **File Menu** allows you to **Save**, **Export** atlas as image, **Open** an existing atlas, **Return to the start menu** and **Close** the current atlas tab.

In the current version the **View Menu** only contains an option for marking the **overlapping images** on the screen.



In the **Settings Menu** the maximum atlas size can be selected as well as the free space left between images when packing.

The **Pack Button** rearranges the images so that as little as possible space is wasted, without resizing the images. We will return to this functionality later in the guide.

In the bottom section, the **name tag**, **position** on the canvas, **original size**, **scaled size** and **scale** of the selected image are presented. In order to select an image, simply left-click it.

grenade_pink	Position: 443x, 236y	Original Size: 200 x 200px	Size: 69 x 69px	Scale: (0.345192, 0.345192)
--------------	----------------------	----------------------------	-----------------	-----------------------------

Multiple images can be selected by also holding **Ctrl**. Selected images are treated as a whole when it comes to moving and resizing.

If you right-click anywhere on the canvas, the following menu will pop up. From here you can **add images**, **select all** images and **delete all** the selected images.



An image can also be deleted by right-clicking it and selecting **Delete**.

Selected images can be moved around the canvas and resized. When it comes to resizing, there are 4 precision levels:

- Scroll: $\times 0.1$ / $\times 1.1$
- Scroll + Wheel Press: $\times 0.01$ / $\times 1.01$
- Scroll + Ctrl: $\times 0.001$ / $\times 1.001$
- Scroll + Ctrl + Wheel Press: $\times 0.0005$ / $\times 1.0005$

After adding and resizing your images you can select a maximum atlas size and some margins and click **Pack**.



After packing you can still move the images around until you are satisfied with the result. Then, click **Export Image** in the **File Menu** and select a destination directory.

The application will generate an **image file** containing all your images together with a **map file**. The map file contains the **name tag**, **X coordinate**, **Y coordinate**, **width** and **height** respectively, separated by “:”. This file simplifies splitting the atlas into sprites later on.

```
grenade_blue:20:84:243:243
grenade_green:265:84:108:108
grenade_orange:20:329:166:166
grenade_pink:20:648:58:58
grenade_red:265:194:108:108
grenade_yellow:171:497:118:118
route66_blue:171:617:108:108
route66_green:20:497:149:149
route66_red:188:329:166:166
route66_yellow:281:617:98:98
```

Implementation details

Aspershine was built using C++11 and SFML: <https://www.sfml-dev.org>.

The texture packing algorithm is a slightly modified version genotech's old rectpack2D library: <https://github.com/TeamHypersomnia/rectpack2D/tree/legacy>, which helped by considerably speeding up the development of Aspershine. For the new rectpack2D library check: <https://github.com/TeamHypersomnia/rectpack2D>.

The Aspershine repository: <https://github.com/filip256/Aspershine>.