Thank you for your support

Hello, and thanks for purchasing ATE, the Atlas Tile Editor. You have supported independent game development. Lost Astronaut Studios is a self-funded exploration of game development. As of 2017 it has yet to make a profit, so your support is greatly appreciated in offsetting the tremendous cost in time and resources required to make tools like this.

Since this is still in Early Access, you may see (WIP) indicating a feature is not yet ready or is incomplete.

Workspaces, Files and Folders:

When you create a new workspace, it will appear in a folder called “ATE” in your My Documents folder located in a path similar to this:
`C:/Users/You/Documents/ATE/workspaces/`

ATE is located in your local steam folder “ATE” and the sample workspaces are stored in:
`ATE/data/workspaces/`

The sample workspace is located in:
`ATE/data/workspaces/main/`

Resolution Limits

Note that the upward limits of atlas sizes in pixel dimensions are limited by hardware. For general knowledge, 4096 x 4096 was the maximum texture size standard in 2010. Today, it’s much larger (16384 x 16384 or larger) but that is true only for the newest hardware.

The exporter currently supports only up to 1024 x 1024 individual tile baking (though it may increase to 2048 x 2048 when possible). For low-resolution games, this is fine. However, you can certainly use high resolution source atlases to maximize the output quality of the exporter.

Workspace Folder Information

Each workspace folder has the following folders in it:

atlases/	Where to put your atlas sprite sheets
bumpmaps/	for keeping bump map textures*
films/	for keeping film frames*
mulligans/	for keeping mulligan textures*
music/	for keeping music files (FLAC)*
particles/	for keeping particle textures*
scripts/	for keeping Alacrity Scripts*
sounds/	for keeping <10 second 32 bit stereo WAV files (samples)*
textures/	for keeping texture maps*
tiles/	for keeping completed tiles*

(* this feature is not yet completed, but will be after Early Access is over)

In the workspace/main folder, a series of text files are used to store information about your atlases and atlas tile sets.

These text files are:

atlases.txt	Describes the atlas sprite sheets
atlastilesets.txt	Describes the atlas tilesets you've created
workspace.txt	Contains information about your workspace

Backing up regularly: notes on the Steam folder and the ATE folder, and using Git

It's kind of dangerous to keep your work inside the Steam folder. If you accidentally remove local content, your work may be lost. It's best to back up by zipping up your workspace folder on a regular basis, and keeping the ZIP files in My Documents. You can then share your workspace with other game developers on the same team, or unzip it next time you install ATE on a different device, or reinstall it on the same device.

Please note that uninstalling ATE from Steam does not remove the *My Documents/ATE* folder, so this is a safe place to keep your backup files. You should not keep any folders inside workspaces/ that are not a real workspace, however inside each individual workspace folder you are free to create new folders. ATE will actively recreate folders you delete in there, however.

You should ZIP up your workspace to back it up, and name the file something like workspace-dd-mm-yyyy.zip or however you would like to organize them. You also could use Git to store your work on a Git Server or on Github, so that you could roll back. However, you should always quit Atlas Tile Editor before rolling back changes from a git repository.

Using the Application

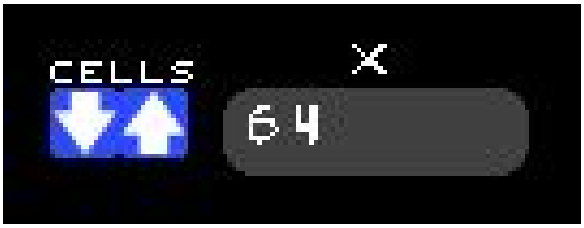
How to Quit: Generally you will want to use the “Go Back” buttons to return to the main menu. On the way back, you are given the option to “Save Workspace” and it is there so you can save tile changes. Some operations automatically do this for you, for example when you import a new atlas, your workspace is automatically saved after you have imported a new atlas. It’s a good idea to press the “Save Workspace” button when you encounter it, if you want to save your changes.

How to Quit immediately: Alt-F4 quits the editor immediately, without saving, on any screen. It will not prompt you if you have recent changes.

How to regain your mouse: Alt-TAB will exit the editor window. The editor “captures” the mouse cursor, so this is the only way out of the window unless you quit.

Known Issue, File Dialog Endless Loop: What happens if you get stuck in an “endless, repeating file dialog”? Because of various facets of the Windows operation system, you may find yourself stuck in a repeating loop of file dialogs. Even if you cancel, it doesn’t really cancel. This is because the main application window is no longer focused. Click the main application window, then you will see the dialog come to the foreground. Clicking cancel this time will exit it.

Numeric entry fields (like this one to the right) allow you to manipulate a number value. You can also type into the text entry field, or use the nearby up and down arrows. Sometimes it won’t let you type a number, you can only use the arrow buttons or the scroll wheel. By holding CTRL, SHIFT or ALT, or a COMBO of those keys, it will change the amount that the arrow buttons or scroll wheel adjusts the number. A small tooltip will appear when your mouse is over the buttons indicating how much it will change and if it will be positive or negative.



See the next page and subsequent section for information on each major screen in the application.

Using the Main Menu

After a brief loading sequence, the application will arrive at the main title screen.

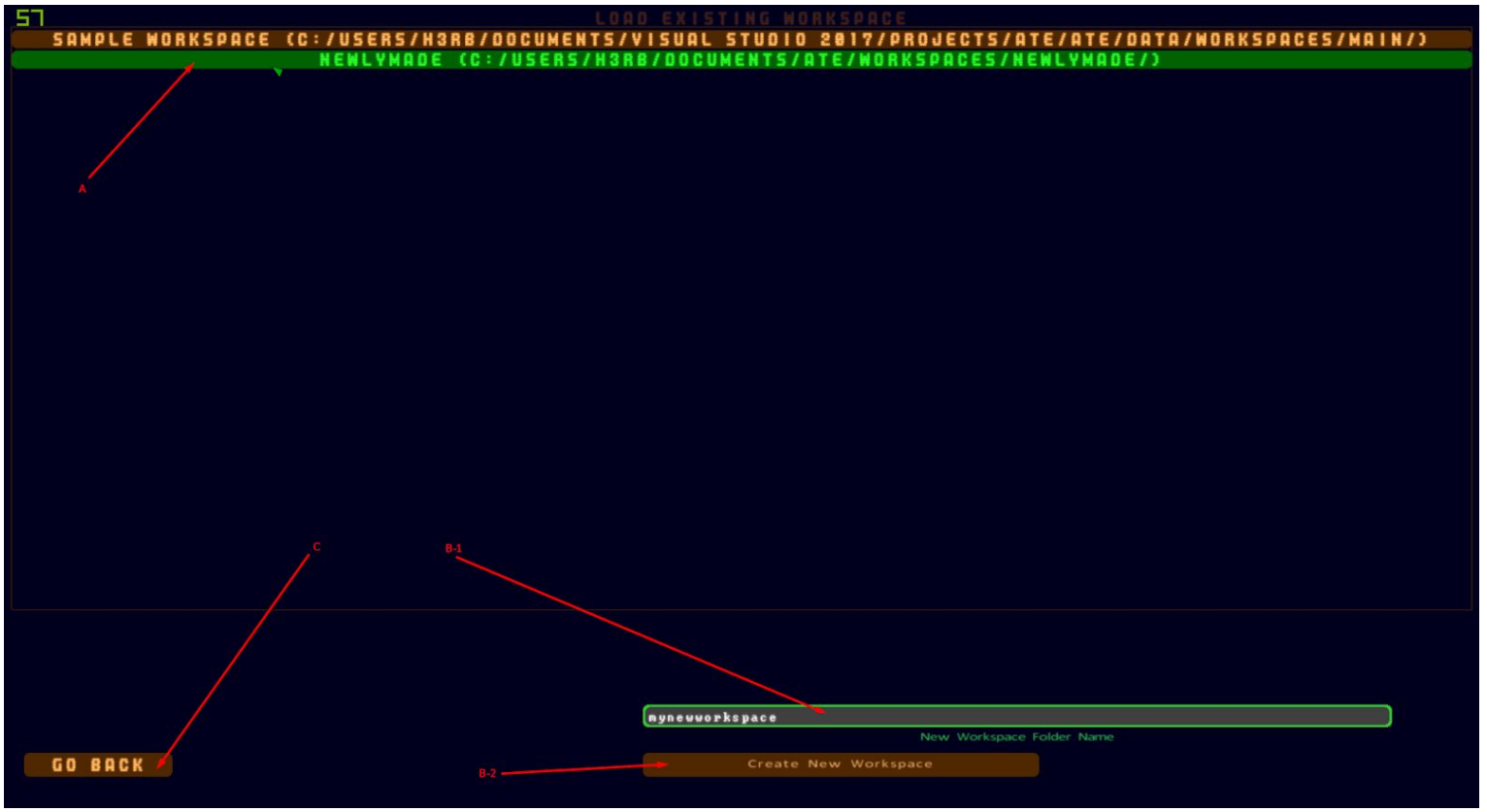
This is is the main menu of the application. Under “OPTIONS” are a few settings, but they may not be all that useful since they are for the game engine that Atlas Tile Editor (ATE) is built in.

Generally you will want to use “EDIT” to enter the main editing workflow, or “EXIT” to quit the application cleanly.

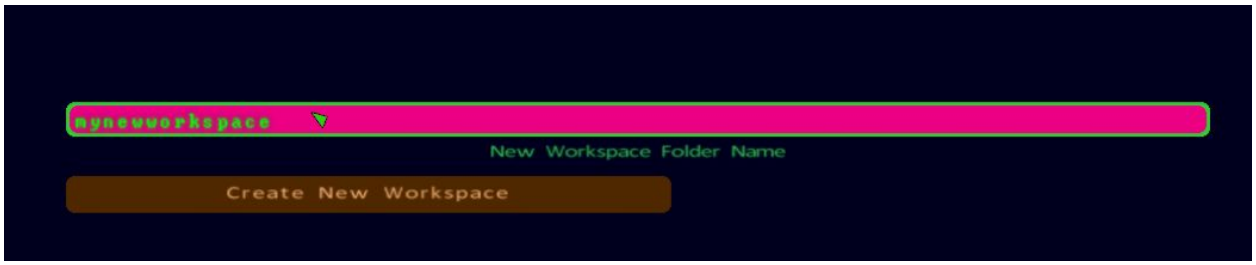
It all starts when you click “EDIT” from the main menu.

Workspace Selection Screen

The “EDIT” button leads to a new screen containing a list of your current workspaces. On this screen you can also initialize a new workspace. If you have unzipped a workspace into the workspace folder, it will appear here too after you have restarted the application.



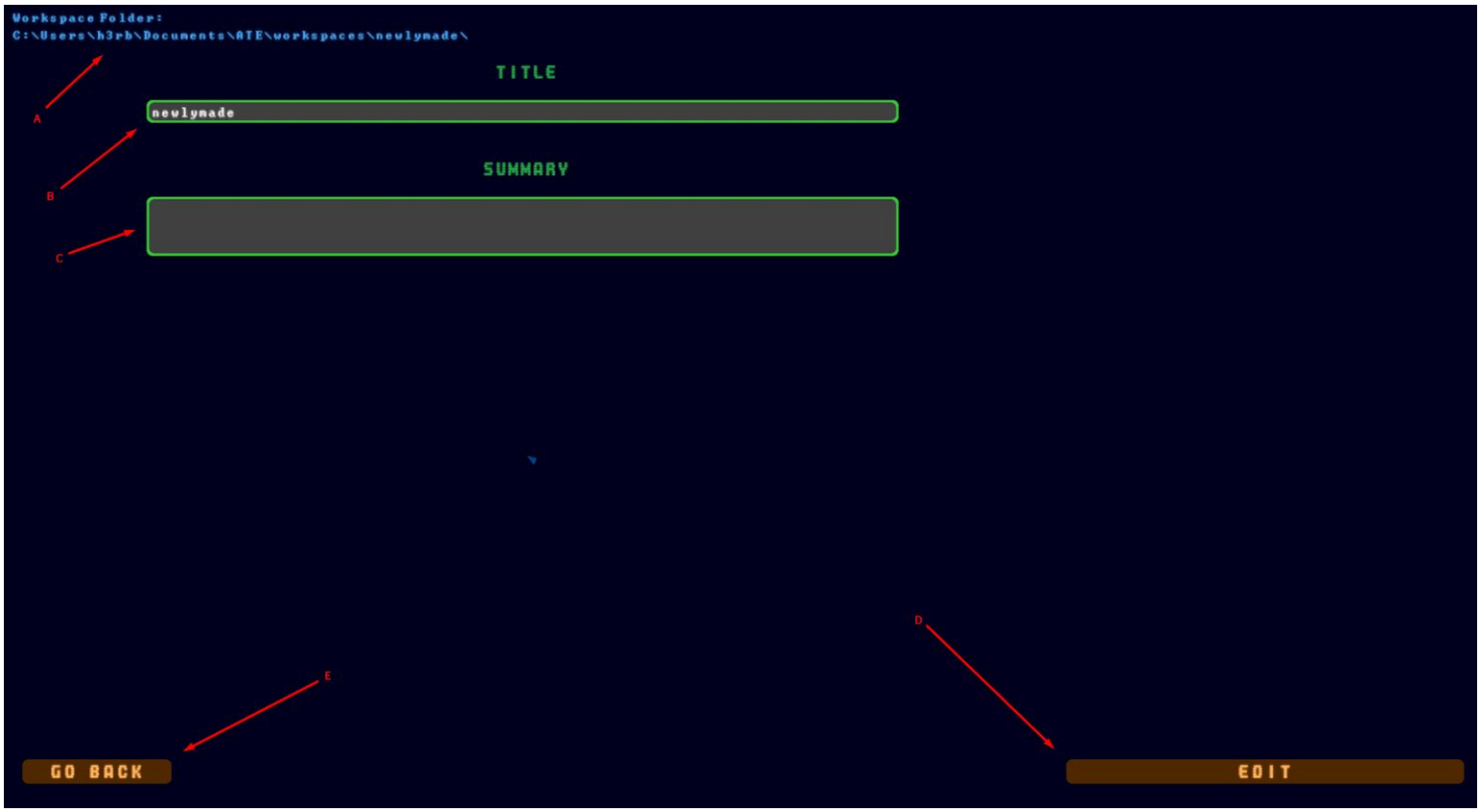
Callouts:
A: Select an existing workspace by clicking one of the items in this list. Workspaces you’ve made are here.
B - 1: Move the mouse over this text box and type on the keyboard to name a new workspace.
B - 2: Once you have entered a unique folder name, a button will appear. Clicking this button creates it.
C: This button lets you go back to the main menu



To create a new workspace: Mouse over the text in the lower right corner. Type a simple alpha-numeric name without spaces or special characters (numbers and letters only), this will name the workspaces/ folder. Once you type enough information, click the “Create New Workspace” button. This will also move on to the *Workspace Information Screen*.

Workspace Information Screen

When you first enter a workspace, you will see a screen that lets you name and describe your workspace.

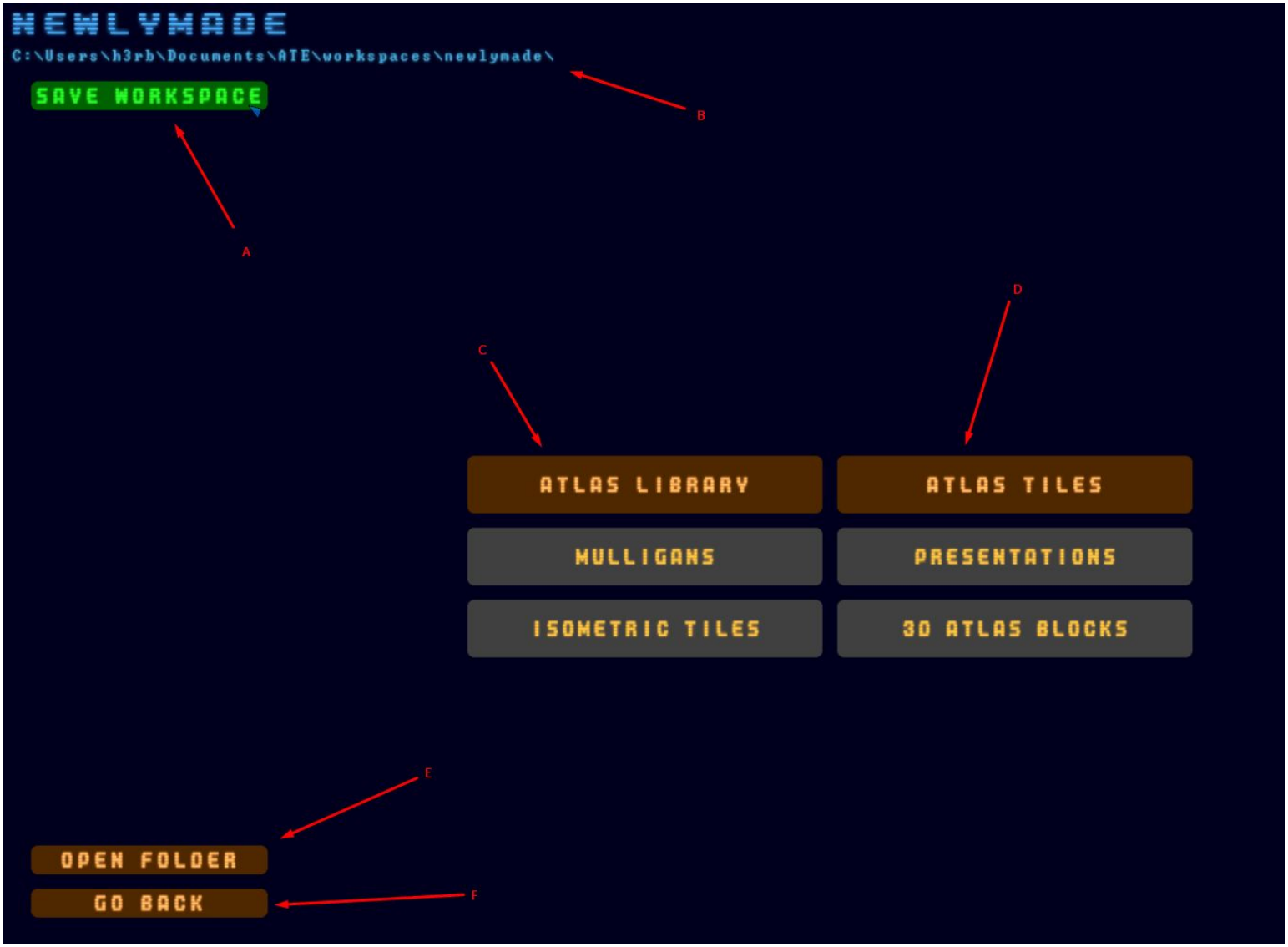


Callouts:
A: This is the folder your workspace is in.
B: This is the name of your workspace. It has nothing to do with the folder name, it is for reference. It shouldn't have any strange characters in it, just letters and numbers. This name is used to reference the atlas in JSON, and also when you are associating a "height map tile" with a "color map tile" (see Using Heightmaps section below for more detail.) To edit this text, hold your mouse over the textbox and use the keyboard.
C: You can write a short paragraph here to describe this workspace and include any important notes you feel are necessary to remember. To edit this text, hold your mouse over the textbox and use the keyboard.
D: Click this button to begin editing atlases, tiles, mulligans, presentations, etc.
E: The "GO BACK" button takes you back to the *Workspace Selection Screen*.

After you click "EDIT" on the *Workspace Information Screen (above)*, you will arrive at a main navigation screen for accessing the various features of the Atlas Tile Editor (ATE) (this application contains multiple editors that allow you to create particle generators, 3D tiles, isometric tiles, atlas tiles, spritesheets, and more).

Once you've selected your newly created workspace, you may at this point want to add an atlas to the workspace. Keep reading to learn about that process.

Main Editor Navigation Menu



Callouts:

A: The "SAVE WORKSPACE" button saves the workspace and all of its atlas, tileset and tile information (along with anything else). While the saving operation is underway you will see the message "SAVING..." appear near the button. The application will appear the freeze for a brief time, this is normal.

B: This is the name of your workspace and its folder path.

C: The "ATLAS LIBRARY" button leads you to the *Atlas Library Browser* and related utility functions for handling spritesheet/atlas importing, exporting, review and texture manipulation like *Image Slicing*, and *Import and Assemble* operations, among others.

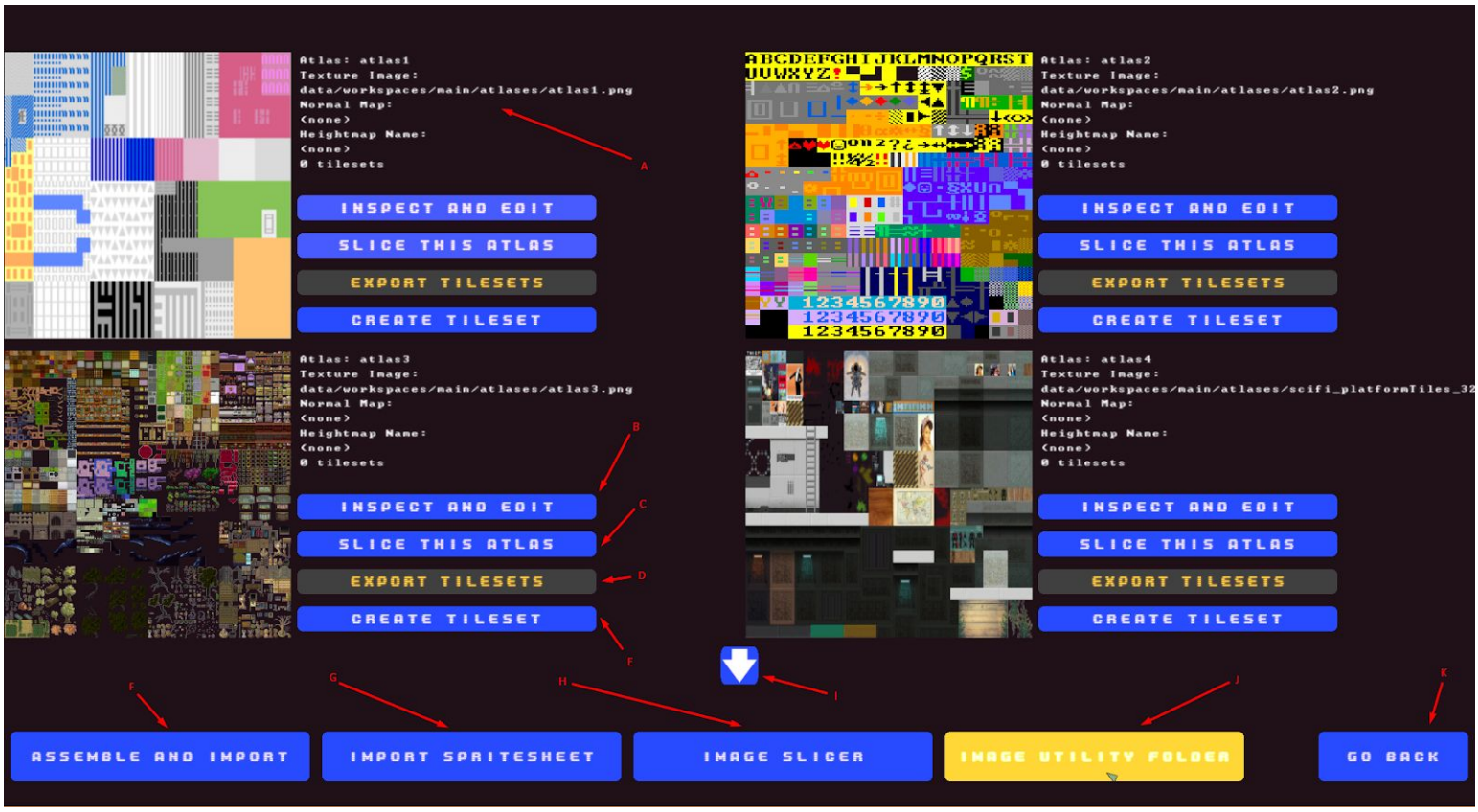
D: The "ATLAS TILES" button leads you to all of your tilesets for quick editing of tilesets.

E: The "OPEN FOLDER" button will open Windows Explorer and navigate to this Workspace's folder so you can access these files. Be careful not to modify these files while ATE is running, as ATE could overwrite your changes or lose important data. It's OK to ZIP them up, however.

F: The "GO BACK" button will take you back to the *Workspace Information Screen*.

Atlas Library Browser

Click “Atlas Browser” on the *Main Editor Navigation Menu* to begin importing and assembling sprite sheets from content you have provided or downloaded from third-party websites like OpenGameArt.org



Callouts:

A: This is a list of atlases stored in this workspace. Some information is displayed next to the Atlas Color Map Texture. Four buttons also appear for each atlas. Using the *mouse scroll wheel* will allow you to quickly navigate this list.

B: “INSPECT AND EDIT” will bring up the Atlas Inspector Screen allowing you to take a look at the settings of a particular atlas. It is on this screen that you can associate an atlas with a Height Map or a Normal Map.

C: “SLICE THIS ATLAS” will enter the Image Slicer with the current atlas texture selected. This will allow you to slice the atlas. If the atlas has a Height Map or a Normal Map associated with it, the Image Slicer will also slice those two textures in addition to the Color Map Texture. This feature is only available by clicking this button, however you can slice single images using the Image Slicer button mentioned below.

D: “EXPORT TILESETS” takes you to the *Atlas Multi-Tileset Exporter Panel*, a screen that allows you to export multiple tilesets either by exporting each individual tile from the sets, or by exporting multiple tilesets using the ATE format or the JSON export format, to a target path.

E: “CREATE TILESET” takes you to a special browser, similar to the “ATLAS TILES” option on the *Main Editor Navigation Menu*, but focused on just this atlas and filtered to show only this atlas’s usage.

F: “ASSEMBLE AND IMPORT” takes you to the special Atlas Assembler Panel, allowing you to combine images from multiple sources into a single Atlas, and then import the resultant atlas into your Atlas Library (optionally).

G: “IMPORT SPRITESHEET” takes you to the Spritesheet Importer Panel, allowing you to import an existing spritesheet or filmstrip image as an atlas to use as source material for new tilesets.

H: "IMAGE SLICER" takes you to the Image Slicer Panel, which allows you to simply load up and slice an image into many images. You can load any PNG or JPG here, and then adjust the settings and export. See a section below on how to use this screen.
I: ARROW LIST SCROLL BUTTON(S) - The large down arrow (pictured) is another option to scroll the list of atlases. Since only four appear on the screen at any time, you may have dozens that you need to scroll through. Alternatively you can use the scroll wheel. When the list is at the beginning, no "up" arrow appears. When the list is at the end, no "down" arrow appears. (not pictured)
J: "IMAGE UTILITY FOLDER" is a button that opens a README.TXT file and the Image Utility Folder that comes with Atlas Tile Editor (ATE). The Image Utility Folder contains several third-party freeware or open source utilities for gluing, splitting and manipulating images. If you go back one folder from this folder, you will see ATE-Docs.pdf and be able to read this manual (also available from the STEAM menu in the STEAM library)
K: "GO BACK" will take you back to the *Main Editor Navigation Menu*.

Importing a New Atlas: First you must provide a series of "tiles" or "brushes" (an atlas spritesheet) to use in the Tile Editor. Only one Atlas is used per set of tiles, but you can create multiple sets from a single atlas. This is whole premise of atlas tiles in a game engine: this decreases texture swaps and lets you create a series of tiles from a single source image by using an artistic method known as collage. A smart 2D engine would "bucket" tiles by their atlas setting to fully achieve optimal performance when using Atlas Tilesets natively. Otherwise, it just serves as a way to organize your resulting tiles under its source graphics.

Click the "Import and Assemble" button if you want to "glue" a series of images together into a single sprite sheet.

Click the "Import Spritesheet" button if you want to use an already glued series of images in a spritesheet or filmstrip image.

Click "Export" on the bottom of the screen to export multiple tilesets in one batch. See "Exporting Tilesets" section below.

Using the Assemble and Import Panel (Atlas Assembler)

The Assemble and Import Panel lets you take multiple images and combine them into a single sprite sheet. It is the first step if you have multiple images (for example images exported from layers in Photoshop) - you will use the top left area to select images to add to the big list on the left of this screen. This is the most complicated panel to import with, but it offers several powerful ways of importing and assembling multiple images.

Callouts:
A: This button will open a dialog that allows you to select a folder and import all PNG or JPG images in that folder.
B: Toggle this switch "on" (a dot will appear in the rectangle) to indicate you want to recursively search every folder within that folder. (WIP)
C: Click this button to open a dialog that will allow you to navigate to a folder and then use CTRL- or SHIFT- click to select one or multiple files in a single location.
D: Any folders or files selected will be added to this list. The list is IN ORDER, so once the final sprite sheet is made, it will be made from top left to bottom right in the same order it appears here.

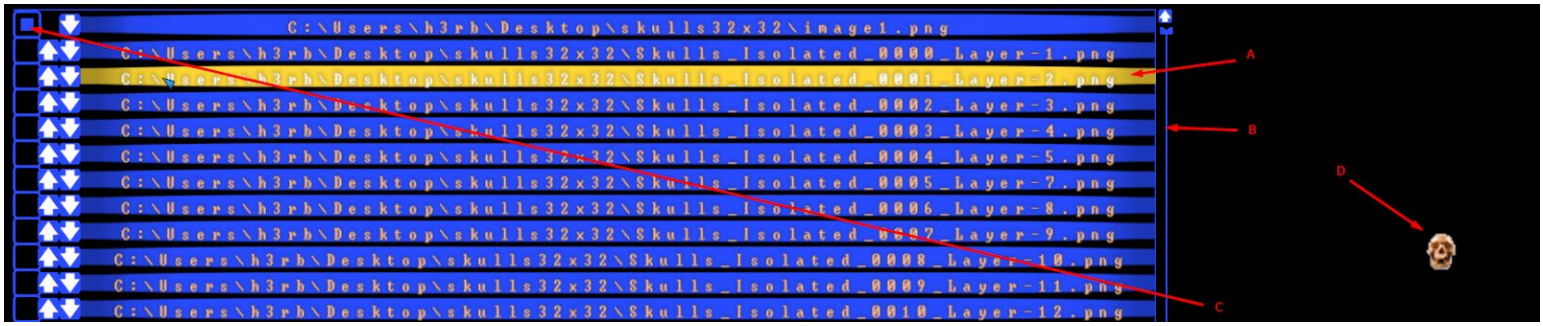
On the left hand side, in the middle of the screen, you will see at first an empty box. As you add files to it, it will fill up with paths and filenames. This is the stack of images you are going to assemble into an Atlas Spritesheet.

The big list o' files in the middle of the screen allows you to preview what files will be assembled in your sprite sheet.

Once you've used the buttons "Import All From Folder" or "Import Multiple Files", this list will fill up.

If a file does not appear in the list but you selected it, it means that something was wrong with the file -- like it wasn't an image or the image didn't load properly.

More about the “big list o’ files” on the Atlas Assembler panel:



Callouts:
A: Click the filename to inspect the file.
B: Click the scrollbar to scroll the list, or you can use your mouse scroll wheel.
C: Click the selection box next to a file to select it.
D: This is where your image will be displayed (along with its dimensions) if you click it in the list.

Callouts:
A: At the left side of each file in the list, there is a selection toggle, and next to this toggle is an up arrow and a down arrow. The up and down arrows allow you to change the order of items in the list. The order matters as the first one will be assembled into the spritesheet first.
B: The “Remove Duplicate Files” button will automatically detect and remove duplicated files from the list. The first occurrence of the file in the list will stay. If a file is unique, it will remain in the list.
C: The “Drop Selected Files” button will remove any selected items from the list.
D: The “Clear File List” button empties the list entirely.

Once you have the precise list of files, you will want to then focus on the other settings to make sure you are outputting everything correctly.

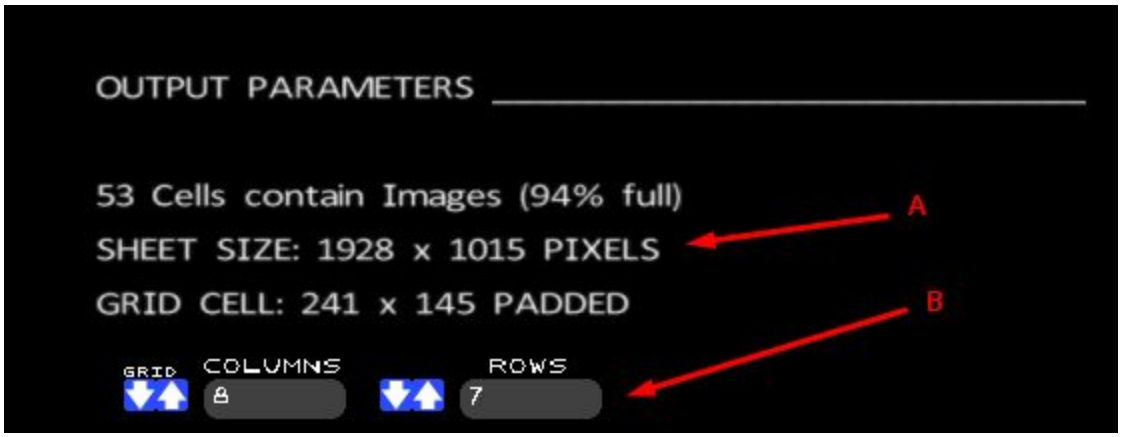
The next logical step is to visit the lower left area of the screen where you can see the ATLAS NAME and ATLAS IMAGE FILENAME. These two settings allow you to name the atlas, as well as name the file, distinctly or by using “Automatic Filename Generation”. Unlike the other text areas in the application, you can click and type like you would in most other applications - you do not need to hold your mouse cursor over the item. If the text box appears with a red hue, that is the text box you will be typing into. When imported, the file will be saved in your Workspace Folder.

You won't be able to click "Assemble and Import" until all of the settings are correct. You'll need a proper name for the atlas (and you need to make sure it is unique to this workspace). You'll also need to take a look at some of these other settings explained below:

Callouts:
A: CELLS - Cell Size in Pixels - you can use your mouse scroll wheel to adjust these numbers when you have your mouse placed over the numeric text entry. You can also type into the text entry field, or use the nearby up and down arrows. By holding CTRL, SHIFT or ALT, or a COMBO, it will change the amount that the arrow buttons or scroll wheel adjusts the number.
B: PADDING - Padding Between Cells in Pixels - this number is by default 1 x 1 and allows you to manipulate how much padding space is used between each cell. You can set it down to 0 x 0 if you don't want any padding, or as thick as you'd like. Having a 1 pixel border around each cell helps alleviate various bleed problems associated with texture atlases in OpenGL. The padding area will be filled with "Clear" RGBA (0,0,0,0)
C: SHEET MAX - Sheet maximum size in Pixels - this number is by default equal to your GPU's maximum texture resolution. It is a hardware dependant number. This max is the absolute largest size of image supported by Atlas Tile Editor (ATE) and can be adjusted only by purchasing newer GPUs. It will not be the same size on every computer. See the above section "Resolution Limits" for more information about this. Keep in mind 4096x4096 is generally big enough for most pixelated 2D games, and even many higher resolution tilesets for games that are Full HD or even UHD. It does not reflect directly the maximum LOD possible with texture filtering. If the "Allow Sheet Trimming" (D) is turned on, your sheets will not be this size except if filled to the brim. If "Allow Sheet Trimming" (D) is turned off, your sheets will equal this size, so you may want to adjust it to be a smaller number.
D: "Allow Sheet Trimming" toggle turns off the automatic size calculation done so that you can export sheets of a particular size, either cropping or filling sheets that, when assembled, do not match the value of "SHEET MAX" (C)
E: "Upscale to Cell" when turned on reveals some scaling options (see below in next callouts) - this activates the ability for Atlas Tile Editor (ATE) to scale up images that are smaller than the cell size so that they fit the cell precisely.
F: "Cell = Largest" means the sheet's CELLS dimensions will be calculated automatically (ignoring what you've entered for "CELLS" (A), and that it will be set equal to the largest image in the big o' file list.

When you turn on "Upscale to Cell", a second option "Bilinear Upscale" is offered. When this is toggled off, Nearest Neighbor (pixel-preserved scaling, without antialiasing, generally great for pixel art) is used. When this is toggled on, Bilinear Filtering is used (similar to the effect in scaling in photoshop, where an attempt at averaging creates antialiasing to make the image appear blurry, as an attempt to fill in the missing data.)

As you begin to adjust settings, one area of the screen should be double checked (right above the Assemble and Import button in the lower right):



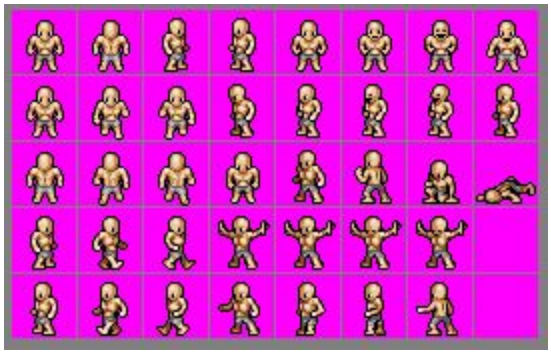
Callouts:
A: Output parameter information. This is the actual dimensions of the output sheet size and its grid cell calculation. This data changes as you manipulate the various features available on the Atlas Assembler panel. The first line tells you how many images are going to fill cells. The second and third line tell you about the overall output image size (the "SHEET SIZE") and the individual grid cell size, in pixels, with cell padding taking into account. In this case, 53 cells are being exported as a single sheet, and that sheet is 94% "full" -- meaning 6% is unused waste memory and will be filled with the special transparent "clear" color. This is a pretty efficient sheet. It is possible to fill 100% of a sheet only if all of the input images are of the same size and the number of input images can easily be arranged as a square or rectangle. You can always fill in that wasted area by editing the resulting sheet file image in an image editor and putting new content there.
B: The "GRID" dimensions (values for COLUMNS and ROWS) shows you how many columns and rows of cells are used to fill the sheet. If you decrease theses numbers such that, when multiplied, the resulting number (total cells) is less than the necessary cells, you'll see a "full" value of > 100%. Atlas Tile Editor will still let you import the sheet, but you should be aware that not all of the cells will be available in the assembled sheet, because your grid was too small. You can then increase or decrease the COLUMNS and ROWS values until you get the size and shape you want.

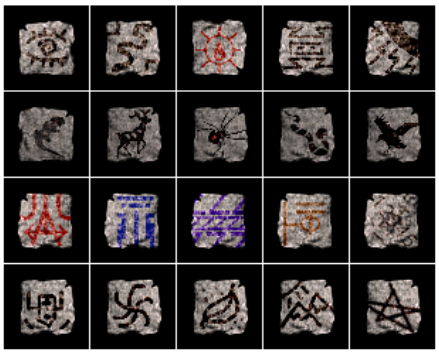
Other features of the Atlas Assembler panel includes these source image manipulation options that filter incoming pixel data in specific ways.

These two toggles in the top right allow you to have Atlas Tile Editor (ATE)'s Atlas Assembler to interpret "Black" as "Clear" or "Magenta" as "Clear" -- below is a color table explaining these colors and their meanings:

Black	Red = 0	Green = 0	Blue = 0	Alpha = 255	Opaque
Magenta	Red = 255	Green = 0	Blue = 255	Alpha = 255	Opaque
Clear	Red = 0	Green = 0	Blue = 0	Alpha = 0	Transparent

Have you ever seen game sprite sheets that resemble this?





The purple color you see is “Red 255, Blue 255” or “Magenta” - the option *Purple2Clear* will tell Atlas Tile Editor (ATE) to import these images in the Atlas Assembler and replace the “Magenta” with the “Clear/Transparent” color RGBA(0,0,0,0) -- the same is true for *Black2Clear*, which converts “black” to transparent. It can also be applied to the edges of anti-aliased sprites, allowing “Additive” sprites to become “Transparent” sprites, see below:

	
Additive Sprite = Colored Light on Black	Transparent = Colored Light on Clear (Black2Clear)

A “Colored Light on Clear” (RGBA 0,0,0,0) can be used with either ADDITIVE blending or with TRANSPARENT blending, for different effects, whereas an additive sprite can only be used with ADDITIVE blending. This is also known as “Remove Matte...” in Greenfish Icon Editor Pro (included in the Image Utility Folder).

Finally, there are two additional toggles on the *Atlas Assembler* panel:

1) By activating the toggle labeled “Mirrored Atlas”, you are telling Atlas Tile Editor (ATE) that the atlas is “4-way mirrorable”, meaning that the *Tile Editor View* (see section below) will permit artists to flip the tile brushes horizontally, vertically or both ways, resulting in four times the number of brush options.

2) By activating the toggle labeled “Nearest-Neighbor Atlas”, you are telling Atlas Tile Editor (ATE) that the resulting Atlas is going to respect the Nearest Neighbor option for loading the texture. This means whenever it appears in the editor, it will use Nearest Neighbor instead of Bilinear Filtering to scale the textures when applying the textures to any tiles in any tilesets using this atlas.

Finalizing your Atlas Assembly

Once you have all of your settings like you want, click the “Assemble and Import” button and it will go to a preview page where you can inspect the assembled atlas before importing it into your atlas library.

Note that it will overwrite any atlases with the same filename, in the same workspace, so be careful!

See the next page for what happens after you click the “Assemble and Import” button.

Callouts:

A: Mouse over this area to inspect the transparency of the atlas.

B: This is the Sheet Grid Inspector. Click one of the items to inspect the individual cell for any overages.

C: Use the arrow keys to move the selection around. A preview will appear to the right of the grid.

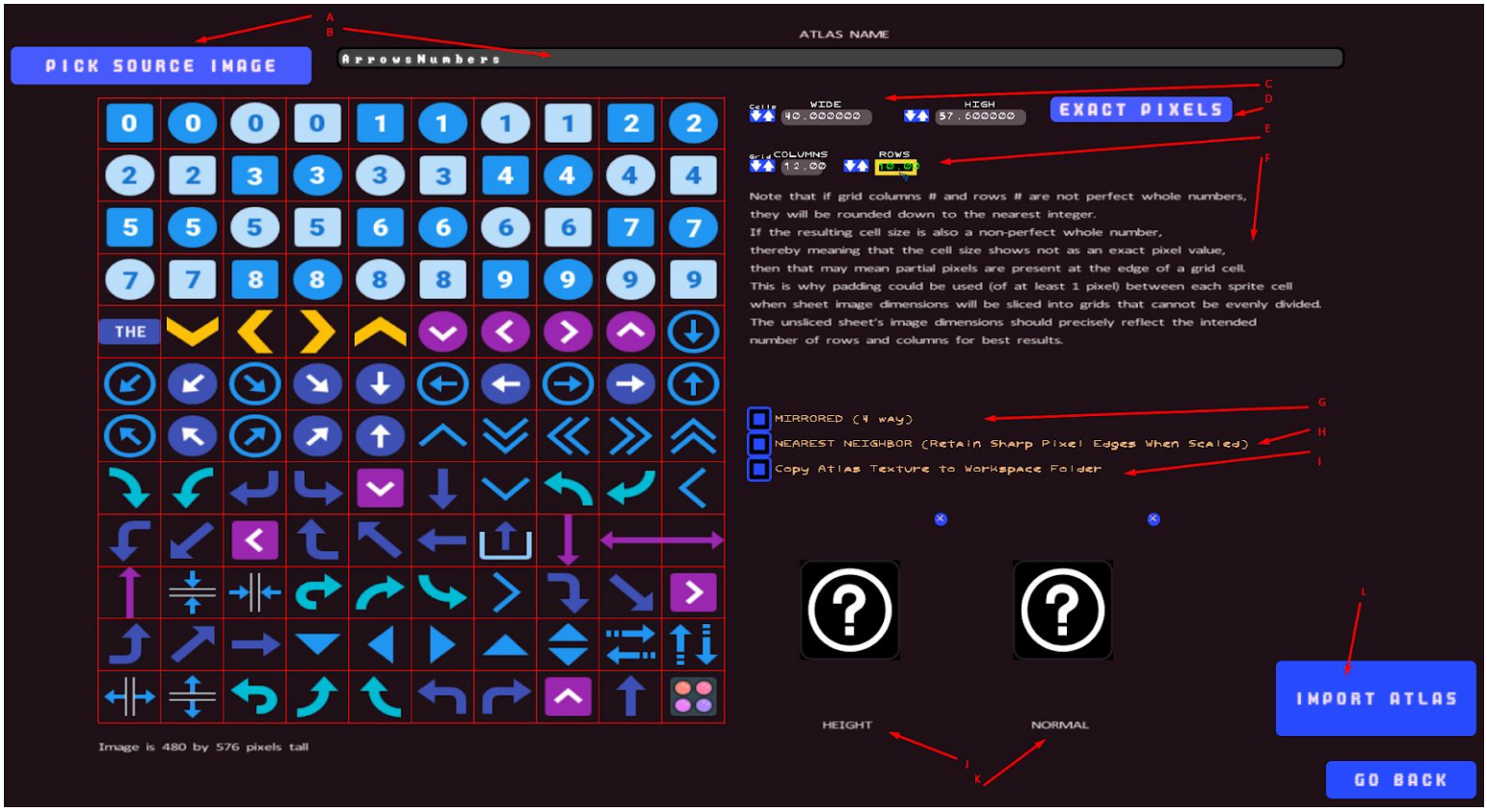
D: This is an optional way to export an additional copy of the assembled texture, without importing it into the Atlas Library. Set the filename/path in the "EXPORT PNG FILENAME" textbox (just typing on this screen will modify it).

E: Click the "EXPORT PNG..." button to export the PNG to the target filename.

F: Click this button to import your atlas into the Atlas Library and return to the *Atlas Assembler*.

There is also a button labeled "GO BACK" in the bottom right which will cancel out of this screen and return to the Atlas Assembler panel.

Using the Import Spritesheet Panel (Spritesheet Importer)



Callouts:

A: Click here to open a dialog and select a single image from a folder anywhere on your computer.

B: The ATLAS NAME is used by the library to reference it. It should be unique within all workspaces. The file's name will appear here as a suggestion for the Atlas Name.

C: "Cells" value is how you can set individual cell heights and widths (in pixels). This number is recalculated when Grid COLUMNS and ROWS (E) is changed.

D: Click the "EXACT PIXELS" button to force the "Cells" (C) to be exact whole numbers (integers). This will also change the value fo Grid COLUMNS and ROWS (E).

E: "Grid" COLUMNS and ROWS are the settings for manipulating the grid that will be interpreted by the Atlas Tile Editor (ATE) as the "edge" of a single cell.

F: This area contains text that explains certain nuances of your settings when the "Grid" and "Cells" values don't add up to whole numbers (integers). Bottom line is that within ATE and using ATE's native formats, everything is fine, but you might see slight bleeding if your input atlas doesn't have padding. If "Grid" and "Cells" are in fact integers, this text doesn't matter and therefore disappears.

G: MIRRORED (4-Way) tells Atlas Tile Editor (ATE) that artists using this Atlas can "flip" and "invert" as well as "flip-invert" grid-placements in the *Tile Editor Screen*.

H: NEAREST NEIGHBOR toggle tells Atlas Tile Editor (ATE) to load this texture in "Nearest Neighbor" scaling mode, instead of "Bilinear Filtering" mode. "Nearest Neighbor" is good for pixel art as it preserves hard edges.

I: "Copy Atlas Texture to Workspace Folder" will do what it describes, and is the recommended setting. The input texture will be copied to the atlas texture. You can turn this off if you've selected a texture inside your workspace folder already.

J: Associate a height map version of your sprite sheet with its color map (the main texture). If you do this, then later

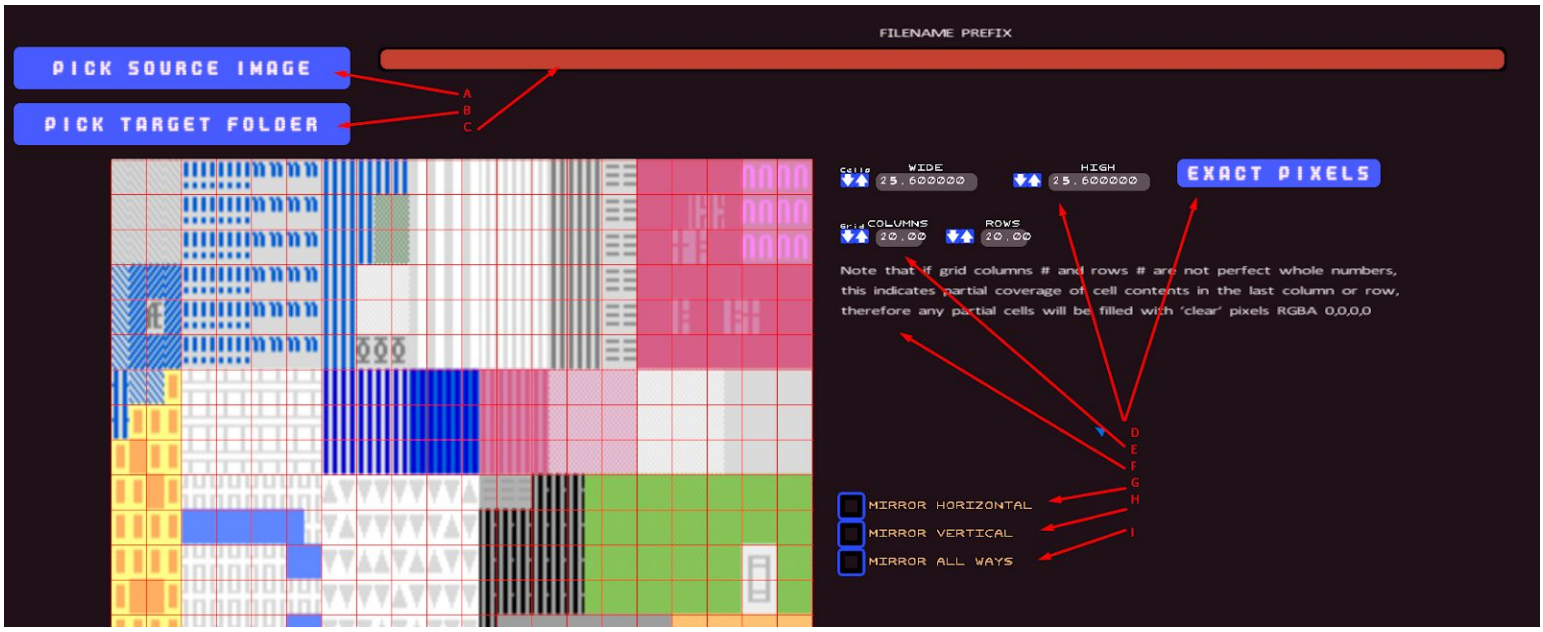
“Slice Atlas”, the height map will also be sliced. The main reason you want to associate such a texture is to provide a way for the game engine using this tilemap to gather height map texture data from a baked texture sheet, rather than from a special height map tile.

K: Associate a normal map version of your sprite sheet with its color map (the main texture). If you do this, then later “Slice Atlas”, the normal map will also be sliced. The main reason you want to associate such a texture is to provide a way for the game engine using this tilemap to gather normal map texture data from a baked texture sheet, rather than from a special normal map tile.

L: “IMPORT ATLAS” button instantly imports the information and appends a new atlas to your Atlas Library, and returns you to the *Atlas Library Browser*, then scrolls the library to your newly added atlas.

Using the Image Slicer Panel (Image Slicer)

The Image Slicer allows you to cut images up using a grid and save each cell independently. You can do this if you want to cut up an atlas into individual pieces.



Callouts:
A: “PICK SOURCE IMAGE” is a button that leads you to a dialog that allows you to pick a source image.
B: “PICK TARGET FOLDER” is a button that allows you to Make a New Folder, or Pick an Existing Folder, where images will be stored. If you don’t press this button, you won’t be able to click the “SLICE NOW” button because it will be invisible. Once you have picked a target folder, the button will appear.
C: “FILENAME PREFIX” text area (whenever you type, you are editing here, on this panel) allows you to add a prefix before the auto-generated filenames. Filenames are generated like PREFIX<CELLX>_<CELLY>.PNG
D: The “Cells” values for WIDE and HIGH allow you to set the Pixel Value. If the values here aren’t whole numbers, you can force them to be by clicking “EXACT PIXELS”
E: The “Grid” values for COLUMNS and ROWS allow you to determine where the image will be sliced, simply by evenly splitting up the file into a certain number of COLUMNS and ROWS.
F: This text explains nuances of the export after slicing, if the values for “Cells” (D) and “Grid” (E) don’t equal whole numbers. The text disappears when it is not longer relevant to your current export settings.
G: “MIRROR HORIZONTAL” will export an additional image file for each cell named PREFIX<X>_<Y>_H.PNG and it will be horizontally reversed.
H: “MIRROR VERTICAL” will export an additional image file for each cell PREFIX<X>_<Y>_V.PNG and it will be vertically reversed.
I: “MIRROR ALL WAYS” will export an additional image file for each cell PREFIX<X>_<Y>_HV.PNG and it will be both horizontally and vertically reversed.

Note that the action button “SLICE NOW” will appear in the lower right once you’ve set up an export folder.
Additionally, there is a “GO BACK” button that appears in the lower right if you wish to leave this panel.

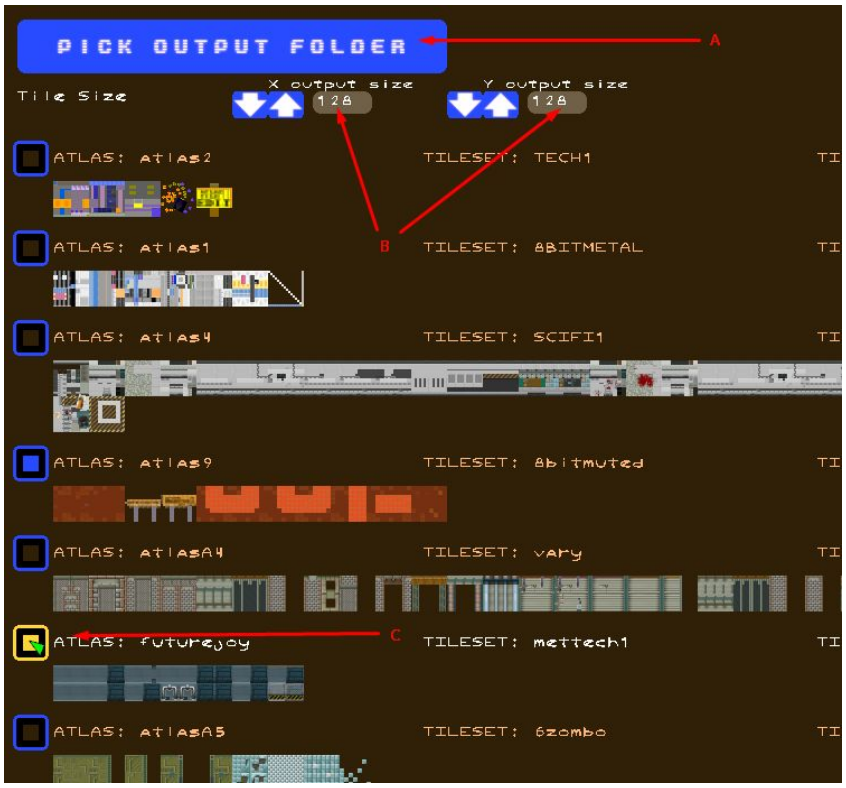
Exporting Tilesets

Atlas Tile Editor (ATE) allows you to export tilesets or even multiple tilesets in a single batch process. Files are exported to a folder of your choice, and while the process can take a while, it tells you exactly how far along it is in the process and allows you to abort the process if it is producing undesirable results.

In the Atlas Library Browser, click “Export” or “Export This Tileset” buttons to reach the Export Tilesets panel. This will bake and export your tilesets to a folder as a series of PNGs carefully named so that they can be used in another game engine. You can also use this process to export baked tiles, then use the “Assemble and Import” feature to recombine them into a single atlas to then reuse in the collage creation process described in a later section.

Callouts:

A) Click here to pick the folder you want to export the files to.
B) Adjust these numbers to set the output size of the tiles, up to 1024x1024 per tile.
C) Click the toggles next to the ATLAS names to select the tilesets you wish to export.



You can use the scroll wheel to go up and down if your tileset lists is longer than a single screen. Only tilesets in the current workspace will be displayed.

Once you have selected some tilesets, you will want to modify the size of the tile output dimensions (resolution of each tile can be up to 1024 x 1024) and of course the bigger the tile size the longer it is going to take to export.

You can also set the “padding” values, which add a border WITHIN the tile size. So if you set it to 1x1, it will be a border of 1 along the vertical and 1 along the horizontal edges of your tile, meaning the tile will take up exactly 2 less pixels in each dimension, but the file size will remain the size you have set.

In the lower right corner of the screen is an area labeled “FILENAME PREFIX OPTIONS”, and this lets you modify the filename formatting that will be auto-generated when the tiles are exported. Note that during the export operation, any existing files of the same name WILL BE OVERWRITTEN. Also note that this area of the screen tells you some basic facts about your impending export.

Click “EXPORT NOW” once you have selected a folder and chosen some tilesets. Bon voyage!

Creating and Editing Individual Tiles on the Tile Editor Screen

When you are back on the main navigation screen, you will see a button labeled “Atlas Tiles” -- Click this to edit atlas tilesets.

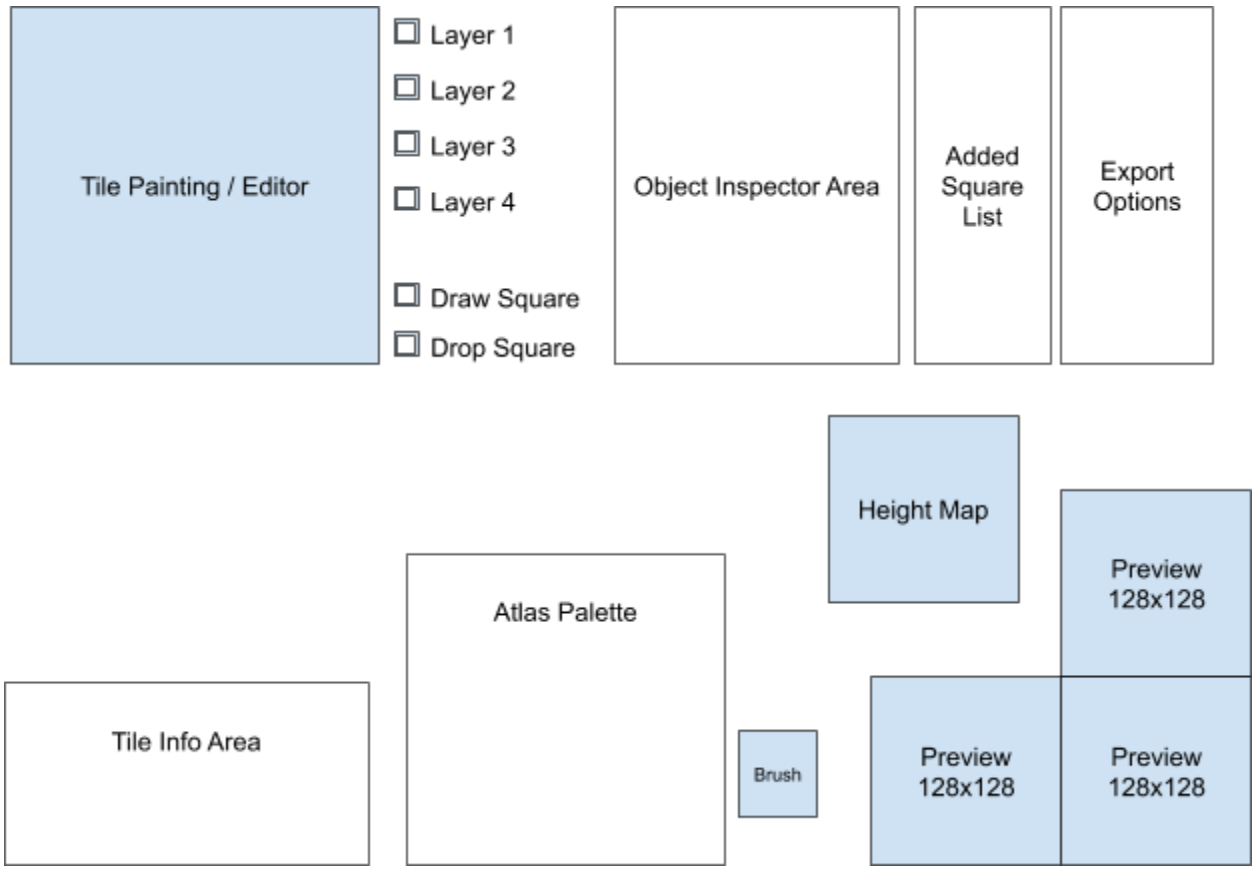
When you click the “+ TILESET” button, you then immediately pick the atlas for creating the individual tiles. There is only one atlas per tileset, but you can have multiple tilesets that use the same atlas.

Once you’ve selected the atlas, make sure to give your atlas tileset a good name that is unique and follows the “alphanumeric only” convention. This helps when looking up tiles in a game, though not all game engines will require this, and it’s relevant only to games that are going to be using the native ATE format, or a JSON format, as this name will be used to identify the tilesets when exporting for HTML.

Now you should be on a screen that has a “+ TILE” button. Clicking this will create a new tile.

If you have already created a tile, LEFT-CLICK the tile to edit it. RIGHT-CLICK the tile to duplicate it, adding it to the tileset. This allows you to create one or more variations of previously designed work.

Once on the *Tile Editor Screen*, you have these major features:



Layered Grid Drawing: To select what to draw, click in the Atlas Palette. This will set your active Brush.
LEFT CLICK on the Tile Painting / Editor will draw in the grid. When you select a layer, you draw on that layer.

- ❏ Layer 1 - 2 x 2 grid
- ❏ Layer 2 - 4 x 4 grid
- ❏ Layer 3 - 8 x 8 grid
- ❏ Layer 4 - 16 x 16 grid

Once a layer is selected, you will draw on that grid based on the currently selected. Layer 1 is in “back”, layer 4 is in “front” -- this way you can draw most reusable tiles.

RIGHT CLICK to remove something you drew on the grid. LEFT CLICK to draw something. Tinier tiles are drawn in front of the larger ones.

Freehand Drawing: Draw / Drop Square Tool and the Object Inspector and Added Square List

If you toggle the box ON next to “Draw Square”, when you click + drag you will draw a square that can be rotated. You can fine tune the rotation and position of the last drawn or selected square in the “Object Inspector Area” and you can select others in the Added Square List, which shows a particular layer’s list of squares (or, when no layers are selected, then the top list of squares).

If you want to insert many of the same size and rotation, you can “paint” with a square by toggling the box ON next to “Drop Square”, and the last dropped square’s rotation and size will be used.

- If you are using the Draw/Drop square tool, it will draw behind the current layer.
- If you deselect all layers, the Draw/Drop square tool will draw on top of the entire stack.

Tile Info Area: In this area you can “Save” which saves all tilesets and tiles in them. You can also name the tile by moving the mouse over the text box, then typing. It’s again best to use alphanumeric names only. Each tile should have a unique name. A name is autogenerated for you.

Export Options

The “Bake to PNG” button appears here.
In the future “Bake to CSS/JSON” may also appear here.
“Bake Tileset” appears here.

Pick Height Map

When you click the height map button (a tile-sized button), you should navigate to a special height map atlas tileset in your workspace. Once you find the height map, you can then click “Select this Height Map and Return to Editor” -- it’s best not to modify the height map during the selection process, unless you’ve made a copy, as other tiles might be using it.

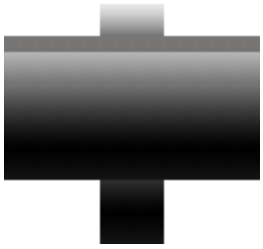
This is not the same as associating a “height map” with a “color map” at the Atlas level. You would not need to Pick a Heightmap for tiles that use an atlas with the additional “height map” or “normal” map” associations, as they come “baked” -- however, you can use this to generate custom height maps for each tile instead if you want, and it is up to the game implementer to determine how to use the additional data.

Append A: About Height Maps and Normal Maps

A sample one has been provided for you. You can copy that atlas from the “main” sample workspace, or, you can create your own. The following atlas appears in the ATE/data/workspaces/main/atlases/ folder as “heightmap.png” and has a cell size of 8x8.

The goal of the height map atlas is to a provide a variety of topographies to describe a heightmap for using bump or normal mapping in a game. Your goal is to match the height map to the color map as precisely as possible. This is not the same as “associating a height map” but rather is a special atlas solely for creating custom tile height maps for each tile. It’s hard to describe but if you look into the height map atlas provided in the sample project, you can see reusable height maps that can be associated with tiles in other tilesets. You can also create your own heightmap source atlas and make custom height map tiles yourself as needed.

Using this feature properly, you can describe height maps to create normal maps for your tiles, and one possible style should something like this:

		
“Color Map” texture (standard tile)	Custom matching “Height Map” tile	Resulting Normal Map conversion

Because height maps are up to interpretation -- as the source art in the color maps is also up to the artist's desires and the requirements of the game itself -- meaning not only plot and content but also how the game engine technically works -- it is not possible to definitively explain the best use of height maps as there may be several different approaches that are all equally valid and produce differing results. The style of heightmap matches your intended look.

Exporting the height map: When you "Bake to PNG" you will see this heightmap exported alongside the color map. What you won't see is the one that is associated with the Atlas, because you would use that one when drawing from the JSON or ATE format for each tile, and wouldn't need an associated height map (usually), though it is better to make your own custom height map for each tile, using a height map atlas.

Note: It's also recommended that, in the game engine, you overwrite the original tile 10% opacity over the height map before converting it to a normal map, but the height map portion will be converted to a normal map without this added detail. (In the future, there may be an option to automatically do this.) Doing this increases the chance for detail, but is not mathematically accurate, though it generally looks OK. It's not required, however.

Resulting "game world" with lighting added, using the height map textures to generate normal maps in real time.

Appendix B: Manually Importing New Sprite Sheet Atlases and Understanding the "ATE" format

Atlas Description Files (atlases.txt)

You can manually add an atlas by copying the file to the atlases/ folder in the workspace, and then add the following block to the *atlases.txt* file:

```
atlas {
name atlas2
texture "data/workspaces/main/atlases/atlas2.png"
grid
mirrored
nearest
rows 20
columns 20
dontGenElements
}
```

If the atlases.txt file does not exist, you can merely create it using notepad.exe or equivalent text editor. Change the highlighted areas to adjust to your settings in your sprite sheet. You can do this multiple times, simply by adding multiple blocks to the file.

Simply change the rows and columns numbers based on your grid-style sprite sheet. You can name your atlas, but it must have no spaces, commas, equals or other special symbols (alphanumeric only).

- The keyword "mirrored" allows you to automatically mirror your atlas 4 ways.
- The keyword "grid" is required for rows and columns to work. Otherwise, individual sections of the atlas sprite sheet must be selected using a rectangle, however it's best not to do that by hand. Instead use the importer.
- The keyword "dontGenElements" is an optimization keyword for not generating individual elements for each sprite selection. It's best to just leave it in.
- The keyword "nearest" tells the atlas loader to use nearest neighbor for enlargement. Omitting this keyword will attempt bicubic resizing, which can result in blurring.

"ATE" tileset format (atlastilesets.txt)

...

Customizing Your Window: Command line options

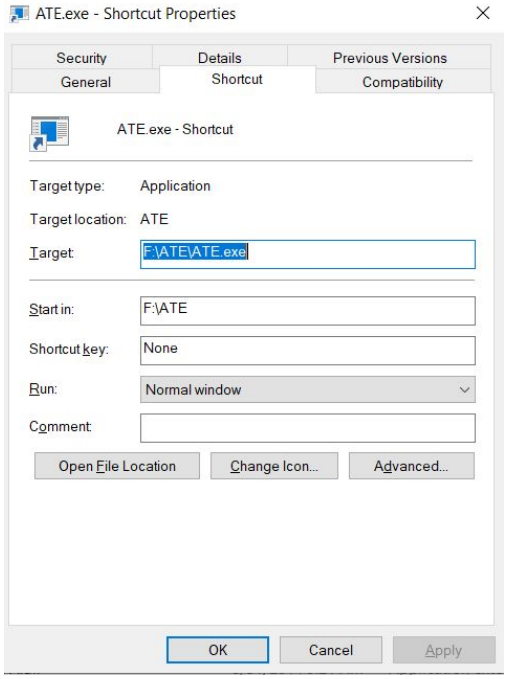
Did you know you can run ATE at a variety of resolutions in Fullscreen or Windowed mode?

Method 1: Windows Shortcut

First, locate the ATE folder. It's near the "utility folder" (a button inside ATE) or you can "Browse Local Files"

Once inside you can create a Shortcut to ATE.exe and rename it something memorable. You can put this on your desktop, too, and make copies of it for your other menus, or "Pin to .." start or taskbar.

Then go to "Properties" for the shortcut.



Then, in the Command Line Options area, you can add custom options after the "Target" ... Just add a space and then add a list of commands separated by a space each. They all begin with "-" (minus sign)

You can create a custom command line option by choosing the following, by the way you can also do this inside STEAM:

Method 2: Inside STEAM

If you right click the application in the STEAM Client, you can use the Advanced Launch Options to enter one or more options from the list below. Please note that all of these options provided as command line parameters to ATE.exe should precede with a single minus sign (-). So, "-windowed" works but "window" won't work without the minus sign. Enter the command line options into the Steam client "ADVANCED LAUNCH OPTIONS" textbox.

For example, to load in 720p mode in a window,

-windowed -720p

Order does not matter, so this will also work:

-720p -windowed

Window Options

-window	Disables use of full screen when creating the OpenGL application window.
-windowed	Same as above.
-borderless	Create a borderless full screen window instead of GPU mode full screen
-bordered	Create a bordered full screen window instead of GPU mode full screen
-resizable	Turns on WIN_RESIZABLE, which usually sucks.
-recenter	Attempts to use alternative window centering at 0,0
-force1080p	Forces use of 1080p

Engine Options

-nomipmaps	Forces mipmaps OFF (why?)
-8bit	Forces bit depth to 8 - may not work right
-16bit	Forces bit depth to 16 - may not work right
-24bit	Forces bit depth to 24 - may not work right
-nolimit	Do not limit the frame rate (will speed up time on some systems, otherwise the frame sleep will fill frames that don't take up 17ms at 60FPSz

Input Options

-trapdoor	Makes “ESC” quit IMMEDIATELY
-lgbuttons	Enables support for super old “Logitech Buttons” input devices like the 18 button mouse (no support)
-resetbutton	Activates a script reset button that has no value in ATE but is used in Fringes of the Empire
-nologitech	Removes (ignores) support for Logitech Kit v1.1 and “older” Logitech devices.
-capturemouse	If set, the mouse button click does not activate the other window.
-mouseescapes	Allows the mouse to leave the window. Might not work right!
The following are Legacy Windows Input Options:	
-nolegacyinput	If RIDEV_NOLEGACY is set for a mouse or a keyboard, the system does not generate any legacy message for that device for the application -- note this feature may be turned off.
-nohotkeys	If set, the application-defined keyboard device hotkeys are not handled. However, the system hotkeys; for example, ALT+TAB and CTRL+ALT+DEL, are still handled. By default, all keyboard hotkeys are handled. RIDEV_NOHOTKEYS can be specified even if RIDEV_NOLEGACY is not specified -- note this feature may be turned off.
-appkeys	If set, the application command keys are handled. RIDEV_APPKEYS can be specified only if RIDEV_NOLEGACY is specified for a keyboard device. -- note this feature may be turned off.
-finetouch	Activates “fine grain touch” forcing TWF_FINETOUCH option by command line. (legacy windows input)
-wantpalm	forcing TWF_WANTPALM option by command line. (legacy windows input)
-wm_gesture	Attempts to call RegisterTouchWindow enabling “Wintouch” gesturing

Custom Window Size

-resolution X Y where X is the width and Y is the height, example:

-resolution 1234 567
OR
-x=1234 -y=567

Keep in mind that the use of this feature in fullscreen may have adverse effects.

Custom Window Offset

You can also move the window if it isn't showing up in the right place. To add to the "x offset" and "y offset" use:

-xofs=123 -yofs=456

If you don't provide an xofs or a yofs, these values will be 0. Keep in mind that the use of this feature in fullscreen may have adverse effects.

There are also a number of specific resolutions supported by command line option:

Specific Resolutions

-320x200 -720x480 -1152x768 -1280x1024 -1600x1200 -320x240
-854x480 -1280x720 -1366x768 -1768x992

-320x400 -768x576 -1280x768 -1400x1050 -1920x1080 -320x486
-800x600 -1280x800
-1440x900 -480x320 -1024x600 -1280x854 -1440x960 -640x480
-1024x768 -1280x960 -1680x1050

HD Resolutions by Trade Name

-hd720
-hd720p
-720p

Sets 1280x768 (HD720p) as the resolution.

-hd1080
-1080i
-1080p
-1080

Sets 1920x1080 (HD 1080i) as the resolution

Specialty Resolutions

-cga	Sets 320x200 (CGA) as the resolution
-tv	Sets 320x400 (Classic scanline TV) as the resolution
-qvga	Sets 320x240 (QVGA) as the resolution
-320x243	Sets 320x243 (pseudo NTSC) as the resolution
-ntsc	Sets 320x486 (NTSC non-interlaced) as the resolution
-pal	Sets 768x576 (PAL) as the resolution
-vga	Sets 640x480 (VGA) as the resolution
-800x480	Sets 800x480 (WVGA) as the resolution
-wvga	Sets 854x480 (WVGA) as the resolution
-svga	Sets 800x600 (SVGA) as the resolution
-wsvga	Sets 1024x600 (WSVGA) as the resolution
-xga	Sets 1024x768 (XGA) as the resolution
-wxga	Sets 1280x768 (WXGA) as the resolution
-sxga	Sets 1280x1024 (SXGA) as the resolution
-sxga+	Sets 1400x1050 (SXGA+) as the resolution
-wsxga+	Sets 1680x1050 (WSXGA+) as the resolution
-uxga	Sets 1600x1200 (UXGA) as the resolution
-qxga	Sets 2048x1536 (QXGA) as the resolution
-cinema	Sets 2560x1440 (Apple Cinema T) as the resolution
-wqxga	Sets 2560x1600 (WQXGA) as the resolution
-qsxga	Sets 2560x2048 (WSXGA) as the resolution

Support for Matrox Dual- and TripleHead2Go

-32godual1	1600x600 DualHead2Go Dual Mode 1
-32godual2	2048x768 DualHead2Go Dual Mode 2
-32godual3	2560x720 DualHead2Go Dual Mode 3
-32godual4	2560x768 DualHead2Go Dual Mode 4
-32godual5	2560x800 DualHead2Go Dual Mode 5
-32godual6	2560x800 DualHead2Go Dual Mode 6
-32godual7	2560x960 DualHead2Go Dual Mode 7
-32godual8	2560x1024 DualHead2Go Dual Mode 8
-32godual9	2720x768 DualHead2Go Dual Mode 9
-32godual10	2800x1050 DualHead2Go Dual Mode 10
-32godual11	3200x900 DualHead2Go Dual Mode 11
-32godual12	3200x1200 DualHead2Go Dual Mode 12
-32godual13	3360x1050 DualHead2Go Dual Mode 13
-32godual14	3840x1080 DualHead2Go Dual Mode 14
-32godual15	3840x1200 DualHead2Go Dual Mode 15
-32godual16	4096x1152 DualHead2Go Dual Mode 16
-32godual17	4096x1152 DualHead2Go Dual Mode 17
-32gotri1	1920x480 TripleHead2Go Triple Mode 1
-32gotri2	2400x480 TripleHead2Go Triple Mode 2
-32gotri3	2400x600 TripleHead2Go Triple Mode 3
-32gotri4	3072x768 TripleHead2Go Triple Mode 4
-32gotri5	3840x720 TripleHead2Go Triple Mode 5

-32gotri6	3840x768 TripleHead2Go Triple Mode 6
-32gotri7	3840x800 TripleHead2Go Triple Mode 7
-32gotri8	3840x960 TripleHead2Go Triple Mode 8
-32gotri9	3840x1024 TripleHead2Go Triple Mode 9
-32gotri10	4080x768 TripleHead2Go Triple Mode 10
-32gotri11	4098x768 TripleHead2Go Triple Mode 11
-32gotri12	4200x1050 TripleHead2Go Triple Mode 12
-32gotri13	4320x900 TripleHead2Go Triple Mode 13
-32gotri14	4800x900 TripleHead2Go Triple Mode 14
-32gotri15	5040x1050 TripleHead2Go Triple Mode 15
-32gotri16	5760x1080 TripleHead2Go Triple Mode 16