Copyright notice

Authors notes

The purpose of this software is to give the modding community means to create and modify Sudden Strike II™ animations. It'll allow you to view and edit existing animation as well as create new from a number of images rendered by a 3D-animation program such as 3dsmax™.

To use this tool efficiently you'll have to do things a certain way. I recommend that you at least read and try the tutorials found in this document as they in just a few minutes time will give you an idea on how this tool should be used and what you can accomplish with it.

If you are going to render your own images in 3dsmax™, please read that section too as it provide essential information required for aniMaker to work.

I hope you'll enjoy this release and appreciate the effort I put into it by making some fine explosions and gunfire animations for Sudden Strike.

Enjoy, mzach.

Conventions used in this document

When you work an Sudden Strike II™ animation (.ani) from either an archive or a file, it will be referenced in this document as an animation. When you work with a number of 3D-rendered images (.tga), it'll be referenced as a sequence. Sudden Strike means either Sudden Strike II™ or Sudden Strike Resource War™ which animation wise is compatible.

About Sudden Strike II™ animations

A Sudden Strike II™ animation always consists of two files. The .ani contains the actual graphics as a number of RLE compressed frames while .col holds the palette information.

There are two kinds of animation types. The first uses semi-transparency and the second use palette-transparency. Pixels are either background (not drawn), 50% transparent or 100% opaque when working with semi-transparency. Palette-transparency on the other hand offers up to 256 levels of transparency but at the cost of color variety.

So when should you use what. Let the existing SS2 animations guide you. Examples of animations using semi-transparency is cursors, zeppelin's, flags and markers. Common for these animations is that they use transparency in a very limited way. Animations such as explosions, fire, gunfire and weather conditions depends heavily on transparency and therefore uses palette-transparency.

For in-dept information on the file structure of Sudden Strike II™ animations, visit this thread on strategy-wargames.com.

Setting up 3dsmax™

aniMaker will only import from 32bit (RGBA) Targa Graphic (.tga) images with non-premultiplied alpha and each frame should be rendered on a black background.

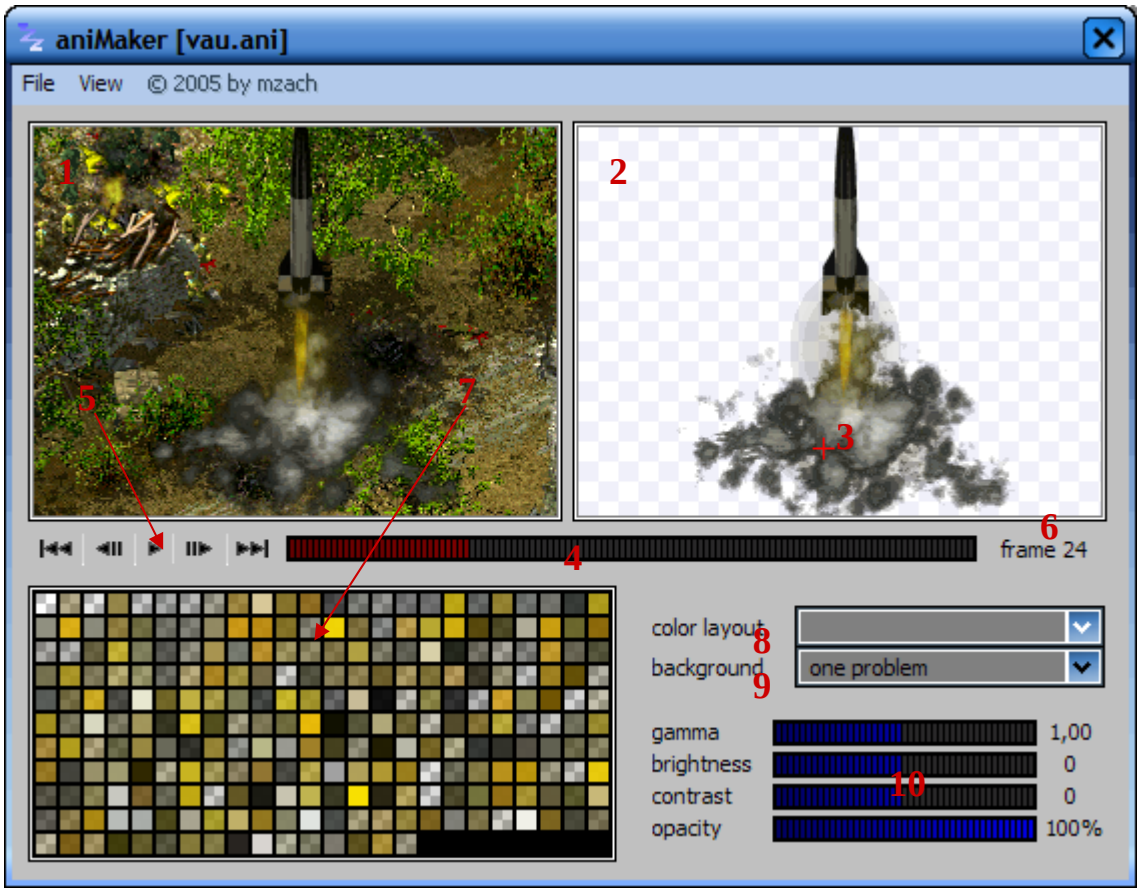
You need to instruct 3dsmax™ to render to environment. This is done in customize > preferences, go to the fane rendering and make sure that use environment alpha is checked.

Setting background color in 3dsmax is done in menu render > environment. This color is scene dependent, which means that it is saved with your .max scene file.

Also scene dependent is the output format. In the render > render menu, find the files… button and click it. Chose targa image file under filetype in the render output file dialog. Under filename, type something like explosion or whatever describes your animation. Then hit the save button.

In the targa image control dialog, make sure bits-per-pixel is 32 and that alpha split and pre-multiplied alpha is unchecked. Then hit ok and save your scene file again to keep the changes.

The interface



1. Frames displayed in this view is always drawn with the palette[7] shown below. Changes made this palette will immediately be reflected in this view.
2. When working with an animation, this view will represent the original and has its own copy of the original palette from which it is drawn. When working on an imported sequence of images, it will show the original 32bit RGBA image of a given frame.
3. The center maker is shown after you defined the center through view > set origin. In this menu you can also toggle weather or not it is shown at all. Its purpose is to indicate the ground center of a sequence.
4. The frame slider. You can use the mouse to select which frame to view using this.
5. Playcontrol should be kinda obvious.
6. Reflects the frame number currently shown.
7. The palette which 8bit indexed frames is drawn from in the left view[1].
8. When working with sequences, the color layout combobox is used to select a palette layout (distribution of colors and transparencies) before converting the 32bit RGBA images into 8bit images. The image right of the combobox processes all converts all frames with the selected color layout.
9. aniMarker parses a specified folder for images you can chose to display in the left view port. When working with sequences you also have an option to set left view port to transparent instead of a background.
10. The brightness, gamma, contrast and opacity controls to modify the palette.

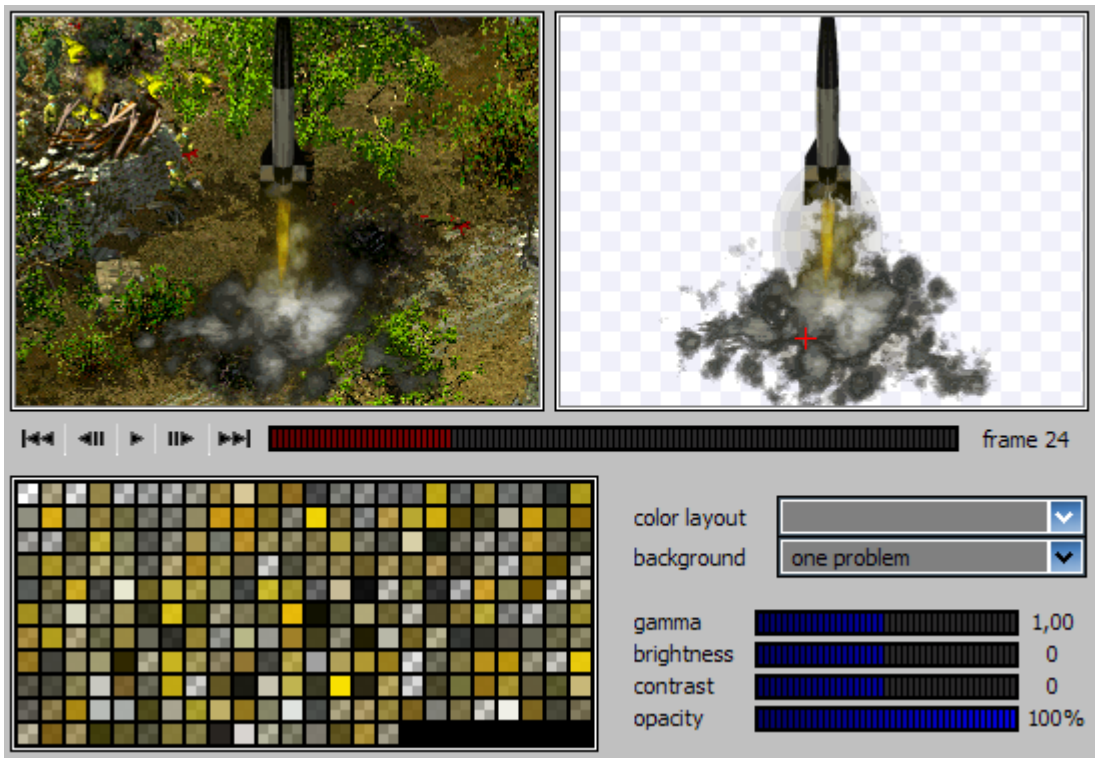
Modifying an existing animation first tutorial

Before going further we would like some Sudden Strike in-game screenshot to view animations on. Pressing Print Screen inside Sudden Strike will make it save the current view as scr00x.tga. If you don't have some already, make some with different scenarios.

Start aniMaker and go to view > path to screenshots or press Ctrl+B and select a path to where the screenshots are located. Typical location is the **game** or **run** folder of your Sudden Strike installation.

Go to file > get from archive or press Ctrl+Alt+L. Use the dialog box to navigate to your Sudden Strike folder where game_common.sue is located. (Typical in the same location as the screenshots mentioned above). Double-click on this archive and wait a few seconds while aniMaker searches it for animations.

You'll see a list of the 246 animations contained in this archive. You'll also see the file size and number of frames each animation has. Type v and you'll see that **vau.ani** is highlighted. Press Enter to load it into aniMaker.



You should now see the first frame of the animation. Try and navigate to frame 24. This can be done in several ways, either use the play controls, F10/F11 or the frame slider.

You'll notice that the animation in your view is positioned somewhat higher than in the above picture. Move the mouse to either view port and hold down the left mouse button. Now move you mouse around in the view. You will notice that you can move the animation around to get a better view.

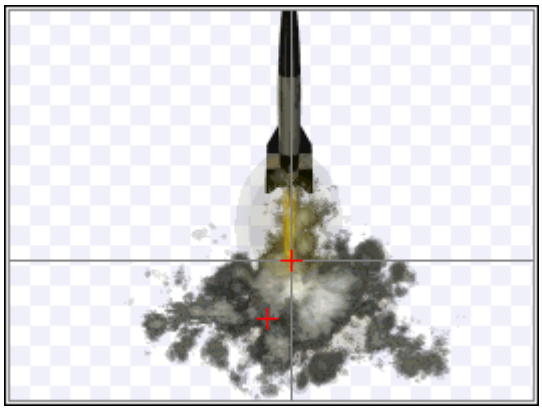
The position is at all times reflected in the other view also. Now move it down a little, so the lower part of the frame almost touches the lower boundary of the view as shown in the picture. The position don't need to be accurate as it only is for viewing purposes.

Now instead you press and hold the right mouse button in the left view. You can now move the background around to the extent of the background image to get a good view of the animation.

☝ *Backgrounds can be selected from the background combobox[9]. Here it will list all bitmaps (.bmp, .tga) found in the location you specified in view > path to screenshots.*

Go to the blue sliders[10] and move them around to view their effect on both the left view port and the color palette. The right view isn't affected by this and will at all time reflect the original material.

Press play or F8 and watch the frames being animated. You can use the blue sliders to affect the palette even when the animation is in playmode.



The palette modification should not be underestimated and is by far the easiest way to modify animations. The red marker in your right view is the center-of-origin of your animation. In case of the V2 rocket animation (vau.ani) it is where the rocket is mounted on the ramp.

You can very easily change this origin of any animation by using view > set origin, which will place you mouse in the middle of the right view. Now you just click where the new position is to be and all frames will be updated accordingly. To make more precise positioning possible you can zoom in and out by either using your mouse wheel or pageup/pagedown.

You can at any time save your animation again with the modified palette. You can either save it to a file or directly into an archive. Go to file > save as file or press Shift+S. Use the dialog to determine the name of the .ani and .col file you are about to export and the location where aniMaker should store it.

☝ *It'll require more than just generating the .ani and .col file to get an animation into Sudden Strike, but that is text file editing and beyond this release. aniMaker only creates the source material, the rest is up to you.*

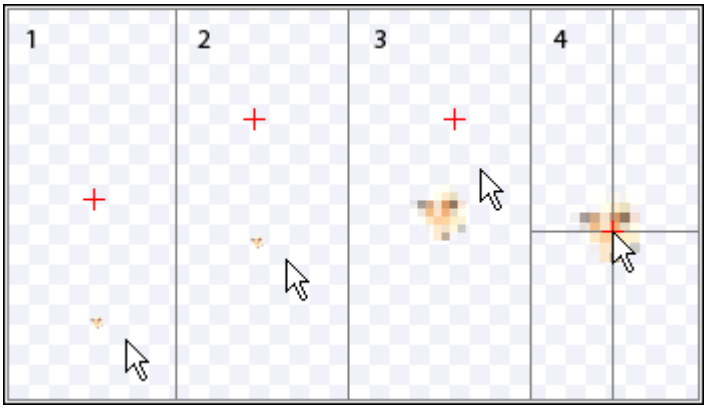
This concludes the first tutorial. Try and watch some other animations as well. You can also export an animation to images if you manually want to edit some or all the frames in another program.

In case you do export to images, these will be stored in the 32bit (RGBA) Targa Graphic (.tga) graphic file format.

Creating a new animation second tutorial

Start aniMaker. Go to file > get from images or press Ctrl+L. In the same place as the aniMaker executable is a folder named data. In this is a sequence called explosion which consists of 101 images named **explosion0000.tga** to **explosion0100.tga**. But they will only show up as one entry in the dialog, called "**explosion**". Load this sequence into aniMaker by double-clicking on it.

Now you should see the first frame of the explosion in both view ports. You'll notice that the center marker is misplaced in the right view. It remains in the center, whereas it in fact should point to where the explosion origins from—In this case in the center of frame 1.



With your left mouse button, move the frame 1 towards the center. Then use pageup or mousewheel to zoom in a bit. (shown in **1** to **3**)

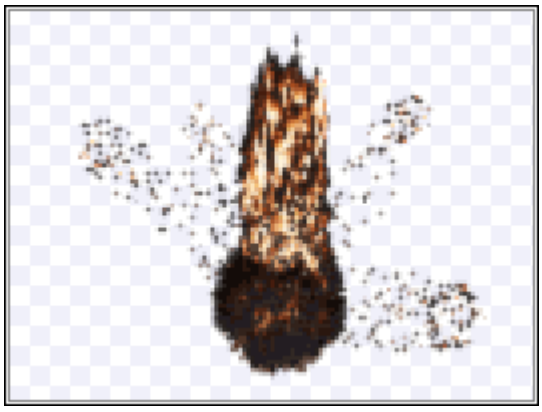
Go to view > set origin or press F7. This will position your cursor in middle of the right view port. Place the cross as show in **4** and click to confirm.

You have now set the origin of your sequence. This is all that is needed before you can save it as an Sudden Strike animation file. Use file > save to file or file > save to archive in order to save it.

Optimizing colorreduction third tutorial

You should now be where the second tutorial left you, if not repeat it before continue. You do not, however need to save as instructed above.

Navigate to a frame 27 and use zoom and move to get a good enlarged view of the frame.

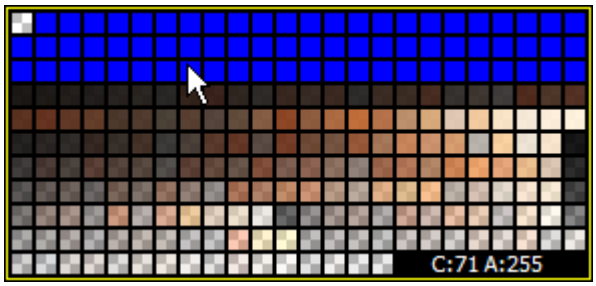


Right-click your mouse on the palette. You'll notice four things; a section of the palette is highlighted in blue, the palette's frame has changed to yellow and the color layout combobox changed from "default" to "new" and the frameslider goes green.

You have entered what I call the ***sandbox***. In this mode you are able to configure the color reduction scheme manually.

When you move your mouse across the palette the highlighted area changes to show you how which color entries are assigned to a certain transparency level.

In the lower right corner of the palette you'll see a **C** value and an **A** value. The **C** value tells you the number of highlighted colors, while the **A** value informs you of the selections transparency value (alpha). An **A** value of 255 means 100% opaque, while 0 means completely transparent.



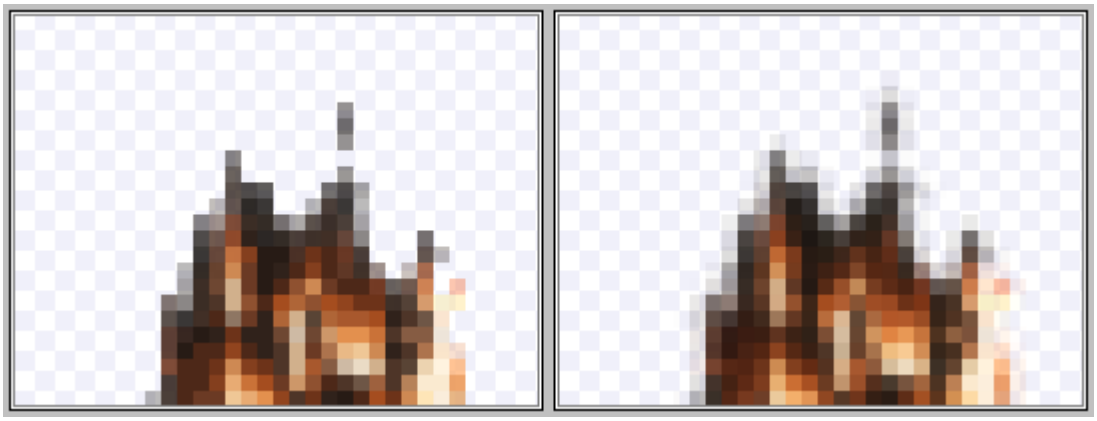
The first entry is reserved for complete transparency and cannot be assigned.

☝ *The current color scheme is in fact a new copy of the default scheme. Every time you enter the sandbox, the current scheme is copied into the scheme called "new".*

Now press F6 to clear the layout. You'll see that the palette is cleared and the image in the left view disappeared, as the left view at all time is drawn using the palette. (no palette, no picture)

Position your cursor on the first entry, press and hold your left mouse button and drag a new selection as show in the right figure. Release the mouse button and you selection will go blue. The **A** value says 255 and every pixel with a transparency value of 255 is drawn in the left view.

Move to start of line 3 and allocate two lines more (same procedure as above). Now the **A** value says 235, and every pixel with an alpha value between 233 and 254 (both included) is added to the picture above. Every time you allocate a new area, the **A** value will decrease by 20.



Try and zoom in on the top of frame 27 to get a better view of the transparency. Always remember that the right view represents the real 32bit RGBA image. Our goal is to get a near match between the two.

Try and allocate one line at a time until you **A** value reaches 155. You'll notice that the colors become more and more transparent and that the image in the left view comes closer to match the right view.

Continue down to an **A** value of 75, but now you'll only allocate 12 entries (a ½ line) at a time.

Using pageup/pagedown or the mousewheel you can lower or raise the **A** value of a highlighted section. Point to the section with the **A** value 155 and raise its value one notch. You'll see that the value is raised with 10. By holding down Control while manipulating the value will narrow that to an increase of 1.

Raise the value to 175. As you reach 175 you'll see your current section and the section above goes red. This indicates that they have the same **A** value and this is not allowed. Lower the value to 135 and the section below goes red indicating that you again is violating the rule. Correct this by returning the value to 155.

Allocate the last sixteen values as you please, just keep the lowest alpha value above zero. As you'll see, it can be quite hard to distinguish between the colors as the sections become very transparent. With this in mind you can allocate less and less palette entries per section as they become more and more transparent. 2-3 colors should be sufficient for the section with the lowest alpha value.

Also, keep in mind that these are animated frames and not stills, so you won't even notice pixels with very low alpha values. Every pixel with alpha values below the lowest A-section value is discharged.

Save your palette layout by pressing F5. Don't call it "new" as it eventually will be overridden next time you enter the sandbox mode and don't call it "default" either as overwriting the default layout isn't allowed.

Now try and switch between the "default" layout and the one you just saved using the color layout combobox. Watch the left view as you switch between them to see which color scheme works best for you.

While playing in the sandbox, you can manipulate the views in terms of moving and zooming. You can also view your current scheme on a different frame using the frameslider.

The main difference between being in the sandbox and outside it, is that all your actions is only applied to the frame in view when in sandbox mode.

Leave the sandbox by right-click the palette. aniMaker now performs color reduction on all frames according to your chosen palette layout.

Zoom out, center the view (F2) and set a background and watch the animation being played F8. Correct the palette brightness, contrast and gamma as needed and save you animation.

This concludes the final tutorial. Should you have questions, then direct them to the aniMaker thread at www.strategy-wargames.com.

Hotkey assignments

F1 Opens the manual (this document)
F2 Center view
F3 Reset sliders
F4 Perform color reduction on all frames (exits sandbox mode)
F5 Save current color layout
F6 Clear palette layout
F7 Set center-of-origin
F8 Play animation
F9 Go to first frame
F10 Go to previous frame
F11 Go to next frame
F12 Go to last frame

PAGEUP Mouse Wheel Up
PAGEDOWN Mouse Wheel Down

With SHIFT pressed:
F2 Center view on current frame
F6 Delete current palette layout

L Load animation from file
S Save animation to file

With CTRL pressed:
L Load sequence
S Save as sequence

With CTRL + ALT pressed:
L Load animation from archive
S Save animation to archive

The tutorial material

The 3dsmax™ explosion scene included in this release is made by Storm Cell. It require the AfterBurn 3.2 plugin for 3dsmax to work. In case you didn't have it I pre-rendered the 100 frames from this scene for you. A special thanks to Storm Cell for providing this material.

Troubleshooting

This release is not flawless and I haven't had enough free time to eliminate every bug. But that said I think aniMaker has turned out rather well given the time in hand. If you should encounter strange behaviour in the user interface or even crash, please let me know. Post a reasonable description on how to recreate the error you experience, in the aniMaker thread of www.strategy-wargames.com.

It has been developed and tested within a Windows XP environment. I haven't had the time to test it much in other environments than XP, so it might work and it might not. In case of problems it is important for me to know which Windows version you use and the amount of memory you have installed.

CRASHES
The program may demand a massive amount of memory depending on the animations you ask it to handle. I haven't had the time to build-in safety precaution for low memory issues, but you might be able to prevent some problems from happening by minimizing and then restore aniMaker as this forces Windows to release a lot of pre-allocated memory.

End of document