

Spriter Beta XNA

QuickStart

This is an example implementation of using Spriter characters in the XNA framework.

If you want to see the library in action, run SpriterBetaXNA program located in the bin directory.

Note: The sample program requires that the XNA 4.0 runtime framework to be installed. If you do not have this installed on your machine, you can download it from Microsoft at:

<http://www.microsoft.com/download/en/details.aspx?id=20914>

When loaded, you should see the BetaExampleHero standing in the middle of the screen. To switch between the animation sequences, press the spacebar (or the A button on your gamepad). To exit, press the escape key or the back button on the gamepad.

Now that you have seen the example, you probably would like to use this in your own games! To do this, you will need to add the SpriterXNARuntime library to your project, and use the custom content pipeline to convert the Spriter characters into a format that XNA can use.

Compiling and running the example solution

Open the solution file in MSVC2010. You will need to have the XNA 4.0 framework (refresh) installed. Build the projects by pressing F6. Then, press F5 to run the debug version.

There are four projects in the solution:

SpriterBetaPipelineExtension – this project is the custom content extension. It is responsible for converting the .SCML files you add to your solution into a compiled .xnb file suitable for deployment. No actual code from this project will be used in your game, it is only used at build time.

SpriterBetaRuntime – this project is the runtime library that handles the Spriter character display and animation in your game project. You will need to ensure your project has a reference to this library.

SpriterBetaXNA –this is a *very* simple example that uses the SpriterBetaRuntime library. This loads and displays a single Spriter Character. This is the example program from QuickStart.

SpriterBetaXNA Content (Content) – this is the content reference project used by SpriterBetaXNA.

Using the library

The easiest way to use the Spriter Beta XNA runtime library in your own games is simply to delete the existing SpriterBetaXNA project within the solution and add your own new project (alternatively use the demo as a base for your own project). If instead, you create a new project, you will need to manually add the SpriterBetaRuntime library as a reference.

To add new Spriter characters, first add the SCML file into the Content project. Make sure that the Content Importer is set to “Spriter Beta Importer” and the Content Processor is set to “Spriter Beta Animation”. This ensures that the custom content pipeline required by the library will run properly over the Spriter file.

Second, you will need to copy the sprite image folders into the content directory so that the content processor can find them.

Finally, build the project and your new characters should be available for loading.

Implementation Notes

This project implements a custom content pipeline. The pipeline extension reads the specified .SCML file and compiles it into a format suitable for the XNA libraries. The pipeline does a number of processing steps. First, it assembles all of the referenced sprites into a single texture atlas (also known as a sprite sheet). This means that for a single character only a single Texture2D object will need to be loaded. The sprite packing routines are taken from the Microsoft XNA sample project “SpriteSheetSample”. The packing routines are basic, and not terribly optimal. Future versions of the library may implement more efficient texture atlas generation.

Note: At the moment, all of the sprites must fit into a single texture, which might cause problems with characters that have a large number of sprites. Future versions of the library may address this by allowing a single Spriter character to access multiple sprite sheets.

When processing the .SCML object, a number of conversions are performed. For each sprite, the sprite’s color and opacity are combined into a single XNA Color structure (each channel ranging from 0-255). Rotations are converted to radians and mapped into the range of $-\pi$ to $+\pi$. The rotation itself has also been negated, to align XNA rotation direction with Spriter’s rotation direction. Sprite size is converted to a scalar value of the original image file. Finally, nearly all of the string references are replaced with indices into lists. This makes for faster playback, but can make debugging more difficult.

Upon completion of the content processing, a single .xnb file will have been created, containing the sprite sheet, the animations, and all of the frame information required by the library.

The custom content processor makes many assumptions. First, it assumes a .SCML file compatible with the Beta version of Spriter, containing a single character. It will only read in the animation keyframes and sequences, skipping additional information in the file. (i.e. it will not handle tweening or other advanced features). The file format is guaranteed to change by version 1.0, at which point the SpriterBetaXNA libraries will need to be updated.

Have fun!

Mark Schmelzenbach, April 9, 2012.