Timpart Spriter Test Suite

Version 0.1

16th November 2012

Introduction

This collection of animations go through the features of Spriter's SCML file format one feature at a time. Plugin authors can use the test suite to gain additional confidence that their implementation is behaving correctly. BrashMonkey's SCML specification has many powerful features, which can seem overwhelming as a whole. The author hopes this test suite will help break down the specification into manageable chunks and allow plugin authors to feel they are making concrete progress with their implementations.

Descriptions are given for each test of how it should look and what to watch out for. Another way to see how an animation should look is to load it into Spriter then run it. In some cases there are extra animations available, with Fail in the name which simulate some typical problems that may be encountered. The Fail animations are not intended to be used as tests with a plugin – they have no hope of working correctly!

Each test has been individually crafted, and in some cases manipulated outside of Spriter to obtain the exact test condition wanted. Although you can load them into Spriter, never save one of the tests from Spriter, as Spriter may have cleaned up or optimised things and the saved version may differ in some important aspect from the original.

The author can be contacted via BraskMonkey's Spriter forums for comments and suggestions about this test suite. He regrets he is unable to help with advice on implementing SCML or debugging software. To avoid confusion do not distribute modified versions of this test suite. Please obtain only directly the source given in the Brashmonket Spriter forums.

This test suite is expected to evolve with time and the initial release is very limited in scope. To avoid confusion if you mention in your literature that your plugin passes tests in this suite you must mention which version of the suite was used (see first page). You should also state which tests were passed and which failed. You can refer to individual sections within the suite or the whole suite if the results were the same. E.g. The plugin passes all of Timpart Spriter Test Suite version 1.5 except for the Pixel Art Mode section.

Legal Matters

The author has produced this Work as a private individual and it is not associated with his employer or BrashMonkey.

“Spriter”, “Spriter Character Markup Language” and “SCML” are the property of BrashMonkey and no infringment of their intellectual property is intended. Trade marks are acknowledged.

A test suite cannot prove a software product to be free of defects and no one must place any reliance on a software product passing the tests in this Work as any indication it is free of defects or fit for any purpose nor may they represent a software product as such.

This Work does not claim to give a complete or accurate diagnosis of faults in software products. A particular software product may have a fault mentioned but not exhibit the behaviour mentioned, or may have a fault not mentioned.

Users of this Work must not state or suggest the the author of the test suite endorses a software product.

Unless required by applicable law or agreed to in writing, the author provides the Work on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with your usage of the Work.

Any Product produced by a third party and tested (by any person) using this Work is the sole responsibility of that third party. The author does not provide the Product with warranties of any kind or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Product and assume any risks associated with your usage of the Product.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the author be liable to you for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the author has been advised of the possibility of such damages.

Versions

0.1 16th November 2012

Initial two Basic tests: Position and Rotation

The Tests

The tests are divided up into sections:

Basic

Basic

The Basic tests are concerned with putting sprites on display, moving them without tweening and overriding their attributes.

Most sprites have dimensions that are a power of two, to make it easier for developers starting on platforms that only readily support such sprites. In later sections the sprites can have any dimensions.

1) Position
2) Rotation

Basic

1. Position

This test checks that x and y coordinates are correctly used and that an object can be positioned with both positive and negative coordinates relative to the origin. It also checks for correct usage of the pivot point which is used to describe what place should be positioned at the x,y coordinates. No rotation takes place in this test.

Expected Result

A square moves along a square path around the origin in a clockwise direction. The origin is marked by a small dot. The large square should start and finish diagonally above and to the right of the origin with two opposite corners on a line passing through the origin.

Check that the square moves down, then left, then up, then right back to the original position where it remains for half a second.. There are brief changes of pivot point in the middle of the bottom, left and top sides of the path. These do not affect the smooth movement of the square if pivot points are used correctly.

0 ms

1000ms

2000ms

3000ms

4000 - 4500ms

Unexpected Results

If the first move of the square is up rather than down then y axis is probably reversed and the square may start like this:

Y-axis reversed 0ms

The SCML coordinate system has increasing y values at the top of the display area. If you have this problem it is strongly recommended that you resolve it before diagnosing other problem.

If the big square is too far above the origin, and makes sudden eratic movements left and/or down half way along each side except the right hand one then perhaps the sprite is being positioned so that bottom left corner is at stated position all the time. It should be positioned so that the point identified by multiplying the pivot_x and pivot_y ratio by the side lengths of the sprite is positioned at x y:

Origin of sprite bottom left 0ms

If the square seems relatively unaffected but has a V shaped dip on the lower and upper part of the path, it is possible that your software has got the y-pivot point inverted. \subtractinhit from 1 might help. Start position similar to that above

Notes

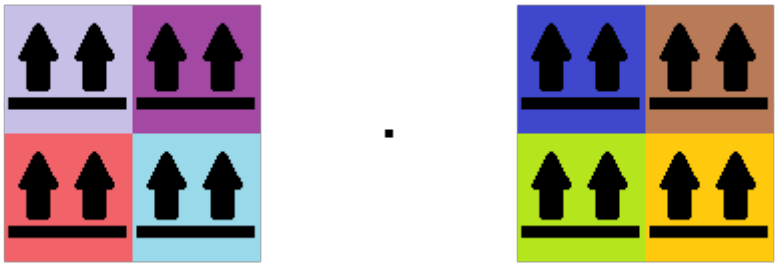
The movement is untweened and animated at twenty frames per second. The animation does not loop.

2. Rotation

This animation does not actually move, instead it has a single frame containing eight large squares, and a small square indicating the origin. Various pivot positions and rotations are used.

Expected Result

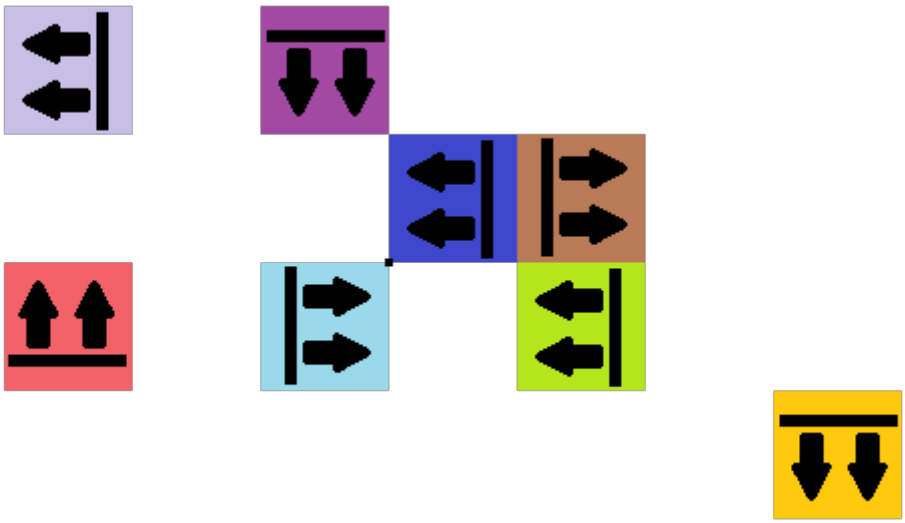
The squares have been rotated in various ways so that they are arranged to form two larger squares. Each individual square has a “This way up” symbol in it which should all point upwards.



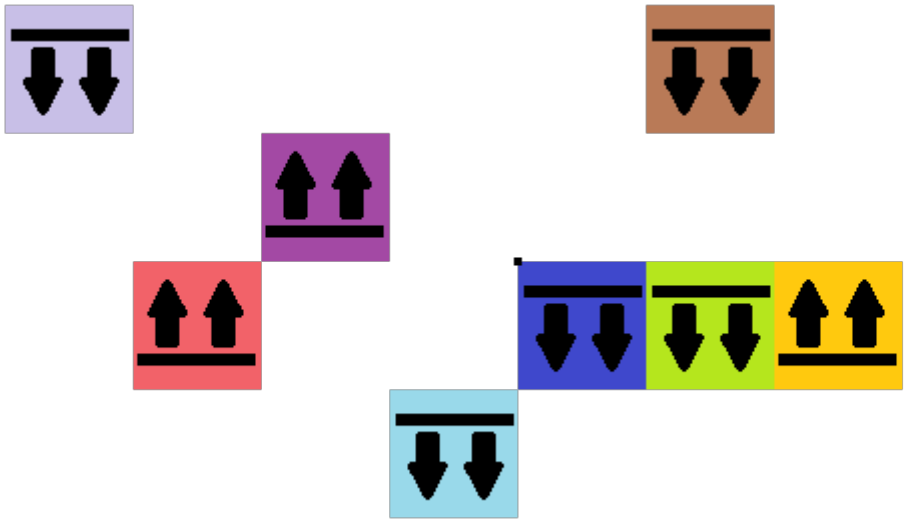
Unexpected Results

The following diagrams may help work out what has gone wrong. The first diagram shows the positions of the squares before rotation. The pink square, bottom left of the left hand group, does not move at all. The green square, bottom left of the right hand group, should only rotate about its centre. The other squares rotate about one of their corners, with the left hand group all using the bottom left corner, and the right hand group the other three possibilities.

If no rotation takes place the squares remain in their original orientations:



If the rotations are made clockwise, instead of anti-clockwise this is the result:



End of Document