

# ReadMe

## Overview

This is the ReadMe file for the STEEL's Sprites library. STEEL's Sprites is a free and open source ([MIT No Attribution](#) licensed) 2D sprite-based animation library to be used with images and animations produced via COLDSTEEL's Sprite Studio.

There are two implementations of this library, one in C, and one in Java. This has been done to hopefully reach the general audience of game developers, who use C, and for those who use Java for their game development. Both implementations are almost identical and this file applies to either (the same file is found in either repository).

The archives this file will be found alongside are source code, and the source code was tested and compiled as follows:

The C version was compiled with gcc version 9.2.0, and g++ version 9.2.0, and the Java version has been compiled under Java 17 and executed on HotSpot VM 64 bit.

STEEL's Sprites is a software library that can be used in conjunction with `.ctsa` files, which are produced from COLDSTEEL's Sprite Studio, to generate data needed to animate associated images. The library is architected to work with OpenGL, but its animation data is based on UV coordinates. And any system which uses UV coordinates will be compatible with STEEL's Sprites. In the worst case scenario, the UV coordinate values produced by the library can be converted into a usable format.

## What it *does*

STEEL's Sprites provides the means to load `.ctsa` files, create animation objects from them, and use functions and callbacks to run those animations.

## What it *doesn't*

STEEL's Sprites does not provide a way to load images. That is up to the user of the library. In the case of the C version, the library doesn't provide a way to read from disk, that also must be provided by the user (although it can use the standard library).

## Final Notes

- The C version *requires* `#include<stdint.h>` for interoperability with Java.
- Any changes and updates made to this library over time will *never* change the original API. Updates will only ever add new functions. When a function is added, it will have a number appended to its name, for example

```
public CTSAFile loadAnimation1(String filepath) {...
```

or

```
CS_CTSAContents* loadAnimation3(CSFile* filepath) {...
```

- In these examples, the number appended to the name of the function tells of its newness to the API. You should always prefer to use a function with a number to an unnumbered counterpart.
- You can find both the C and Java repositories online at:
  - For Java: <https://github.com/Chris-Brown15/COLDSTEELS-Sprites-Java>
  - For C: <https://github.com/Chris-Brown15/COLDSTEEL-s-Sprites-C>