# Animation Framework Comparison

This report is a summary of the features of various frameworks for making interactive, multimedia applications such as games. These are frameworks that I've looked at for personal and academic projects in the past. I've gone through documentation and compiled example projects for most of them to make an informed decision.

This document is divided into four categories: Java, C++, Visual and HTML5. The frameworks listed in the Java section are probably most relevant to Sysemia, given the Java-focused nature of our team. The C++ engines likely provide a greater level of low-level control, but are more painful to use than Java. Visual, editor-based frameworks similar to Unity would allow us to develop products the fastest, at the cost of a lack of flexibility. HTML5 engines would allow the flexibility (and restrictions) that web-based platforms bring.

Why not implement everything ourselves in OpenGL and SDL, or JOGL? This is a viable alternative and would allow us to only implement the features that we need. However, this approach would greatly increase the amount of time spent making the product, as well as the likelihood of bugs.

## Java-based

| Name | Programming language(s) | Platform | 2D/3D | Focus | Extensibility | License | Used commercially? | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Processing | Processing / Java | Windows, OSX, Linux, Android, HTML5 | Both | Visualisation / Interaction | High (Java) | GPL and LGPL | For interactive installations / exhibitions, yes. Less so for products / games. | Some performance issues |
| libGDX | Java | Windows, OSX, Linux, Android, iOS, HTML5 + webGL | Both | 2D games | Very high (Java) | Apache 2.0 | A great many commercial games have been published with it. | Heavily structured, but high performance. 3D support is pretty good. Supports many third-party tools. |
| jMonkey engine | Java | Windows, OSX, Linux, Android | 3D | 3D games | High (Java) | BSD | Some games published, not many. | Looks like the best solution for pure 3D. |

Processing uses its own simplified version of Java, which is great when using its IDE, but becomes cumbersome when integrating it into a Java project. 3D is possible, but not with the sophistication of a full-blown rendering engine. Collada support is possible, but we'd have to roll our own. Performance is generally acceptable, but is slower than game engines, which tend to be more heavily optimised. I had problems using its audio library in Linux, due to lack of pulseAudio support.

LibGDX is a widely-used game engine for mobile and cross-platform development. It supports more platforms than any other Java-based framework, including iOS and HTML5 with webGL. 3D support is great, allowing you to drop down into OpenGL ES 2.0 if needed. Has full Collada support via fbx-conv. In order to make cross-platform export easier, it does enforce a certain project structure. This is easy to extend, though, and does not force a specific design on you. HTML5 support uses GWT.

jMonkey could be a good choice for pure 3D applications. Has partial Collada support if you convert the Collada object first. Android support seems like a bit of a hack.

## C++ based

| Name | Programming language(s) | Platform | 2D/3D | Focus | Extensibility | License | Used commercially? | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cinder | C++ | Windows, OSX, iOS | Both | Visualisation / Interaction | High (C++) | BSD | For interactive installations / exhibitions, yes. Less so for products / games. | No Android support. |
| openFrameworks | C++ | Windows, OSX, Linux, Android, iOS | Both | Visualisation / Interaction | High (C++) | MIT | For interactive installations / exhibitions, yes. Less so for products / games. | Many examples published online |
| OGRE | C++ | Windows, OSX, Linux | 3D | 3D rendering | High (C++) | MIT | Yes, for many games. | Rendering engine only. No support for sound / input, etc. |
| Horde3D | C++ | Windows, OSX, Linux | 3D | 3D rendering | High (C++) | Eclipse Public License | Only a few games published. | Rendering engine only. No support for sound / input, etc. Aims to be an improvement on Ogre, but is less recently updated. |
| Irrlicht | C++ | Windows, OSX, Linux | 3D | 3D games | High (C++) | zlib | Yes, for many games. | More complete solution than Ogre. |
| GamePlay3D | C++ / Lua | Windows, OSX, Linux, Android, iOS | 3D | 3D games | High (C++) | Apache 2.0 | Only a few games published. | Backed by Blackberry |
| LÖVE | Lua | Windows, OSX, Linux | 2D | 2D games | Low (scripting) | zlib | Only a few games published. | Very easy to use for 2D games. |

Cinder and openFrameworks seem to be C++ alternatives to Processing. Cinder appears to be the most polished of the three, but has no Android support and appears to be heavily geared towards iOS.

OGRE and Horde3D are the most advanced open-source rendering engines available. Horde3D claims to be next-generation, and more advanced than OGRE, though this is hard to verify. Development on it seems to have stalled. It's worth noting that these aren't game engines. They don't handle audio, input, physics, etc - only graphics.

Irrlicht is a full-featured game engine, perhaps the most graphically sophisticated of the open source ones, with a rendering engine comparable to OGRE. Several commercial games have been made with it.

GamePlay3D looks very nicely designed, and has the commercial backing of Blackberry. It has the widest platform support of all the C++ 3D game engines. Some advanced 3D features are yet to be implemented, though.

LÖVE is a very popular Lua-based 2D game engine that I've used a lot before and enjoyed. For 2D-only, I would highly recommend it.

## Visual, editor-based

| Name | Programming language(s) | Platform | 2D/3D | Focus | Extensibility | License | Used commercially? | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Unity | C# | Windows, OSX, Linux, Android, iOS, HTML5 | Both | 3D games | Low (scripting) | Proprietary | Very extensively, for games and simulations. | Industry standard. Allows games to be made very quickly. |
| Godot | C++ / GDScript | Windows, OSX, Linux, Android, iOS, HTML5 + webGL | Both | 2D + 3D games | Low (scripting) | MIT | Yes, for many games. | Open source Unity alternative. Excellent Blender integration. Very fast workflow. |
| Polycode | C++ / Lua | Windows, OSX, Linux | Both | Visualisation / Interaction | High (C++) | MIT | Not yet | Open source Unity alternative. Still a work in progress. |
| Torque 3D | C++ | Windows, OSX, Linux | Both | 3D games | High (C++) | MIT | Yes, for many games. | Open source Unity alternative. |

Unity is a widely-used industry standard game engine. Its popularity comes from the fact that it enables non-programmers to make games extremely quickly. Even seasoned games developers are turning to it because of the extreme speed of development it enables. The only problem is its proprietary licensing and lack of extensibility.

The most promising open-source alternative is the Godot engine. It was developed in-house by a major games studio, then released as open source. It has much of the ease of use and features of Unity, though a bit less extensive. However, it uses its own scripting language, GDScript, which is syntactically very similar to Python. Blender-oriented workflows are very fast: you can create a whole scene in Blender and export it to Godot with its structure intact.

Torque3D is also a good alternative, though I don't have much experience with it.

## HTML5 engines

| Name | Programming language(s) | Platform | 2D/3D | Focus | Extensibility | License | Used commercially? | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Phaser | Javascript | HTML5 + WebGL | 2D | 2D games | Medium (Javascript) | MIT | Yes, for many games. | Most promising HTML5 2D game engine |
| PlayCanvas | Javascript | HTML5 + WebGL | 3D | 3D games | Medium (Javascript) | MIT | Not yet | Has an editor, but it's browser based and closed. |

HTML5/WebGL based engines support all platforms, in theory. Free tools like CocoonJS package the game for deployment to multiple platforms. One thing to note is that iOS only supports WebGL from version 8, which is yet to be released. While these frameworks can't be easily extended with Java, etc, they would make it easy to embed applications seamlessly inside web pages.

I used Phaser for a 2D game at a Bristol game jam. It has a lot of features, and is very easy to use. It's limited to just 2D stuff, though.

PlayCanvas looks promising for 3D/WebGL stuff. It's both open source and backed by a company. It also has the advantage of having a Unity-like visual editing environment, though this is browser-based, and you have to pay for private projects. The editor itself is closed-source.

## Summary

LibGDX appears to be the best general-purpose framework for interactive applications. 2D and 3D support are both excellent, it is Java-based and it supports all desktop and mobile platforms with minimal effort. It even supports HTML5 and WebGL. I recommend it as the most flexible choice out of all of the examined frameworks.