# Report on Game Frameworks/Middleware

ID: S102946740

Name: Benjamin R Fekete

## Raylib (multi language framework)

Raylib is a C library and has multiple choices for being used on other coding languages, its purpose is to provide high level graphics functions for 2d and 3d games. The library is open source and is constantly updated by “Ray”. Raylib supports platforms like, Windows, Linux, OSX, Raspberry pi, Android and HTML. And supports language bindings including C#, Go, Lua, Python, Rust and many more. Raylib has multiple choices for graphics frame works to support multiple platforms such as, OpenGL, Direct3D, Vulkan and Metal.

A screenshot of a computer

Description automatically generatedRaylib is a great introduction to graphics frame works as it doesn’t require knowledge of advanced low level functions like buffers (used in DX12), it also provides game specific features like 3D models, animations, VR, Controller input and shaders.

Raylib is broken down into modules, which can be disabled/enabled when needed. This creates a very modular experience and increases the performance of your projects. Modules are broken down into categories such as, Core, rTectures, rText, rShapes, rModels, rAudio, rlgl.

Figure 1.1: Raylib Modules

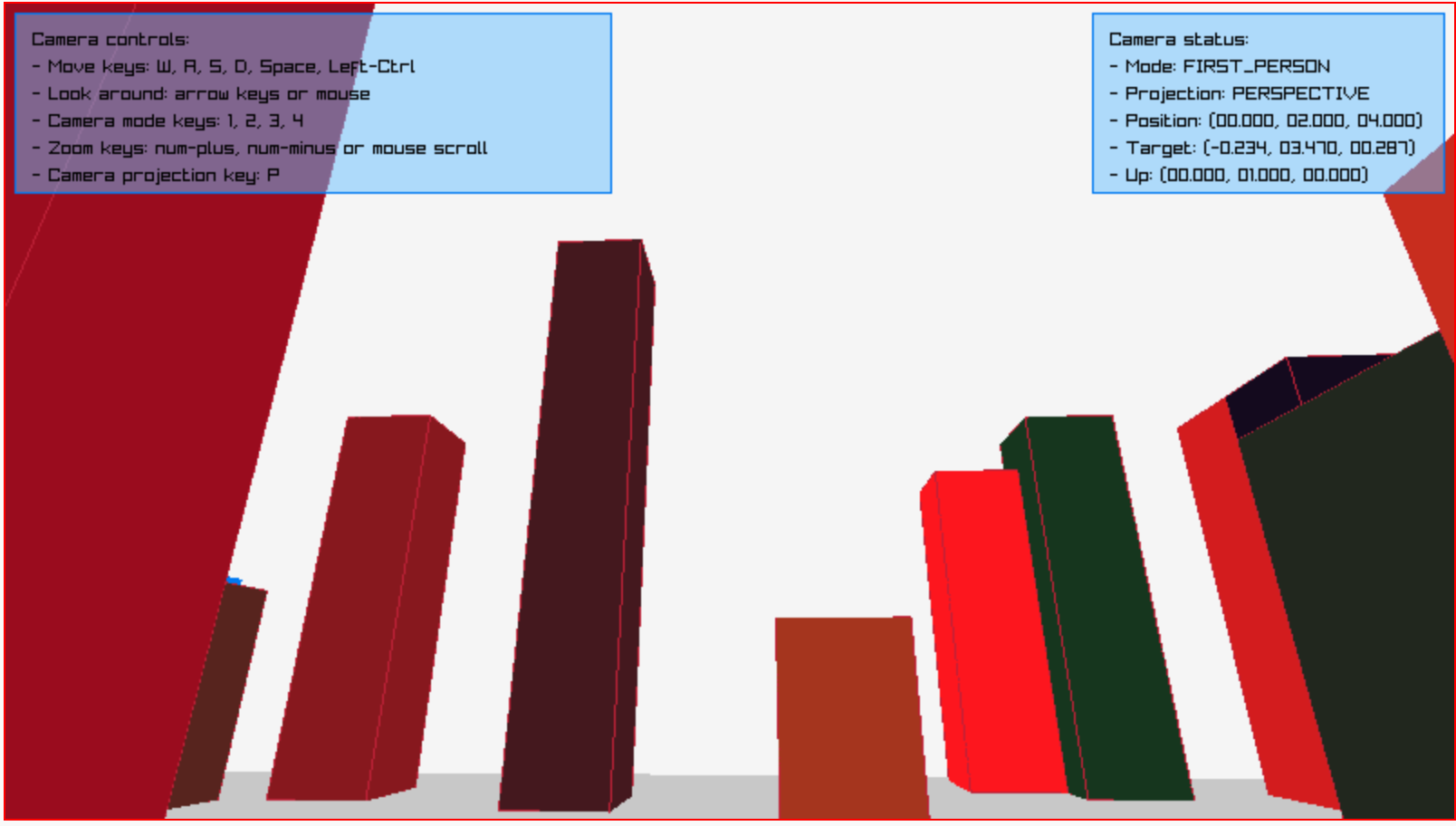


Figure 1.2: Example 3D rendering running on browser

Raylib is a powerful library requiring minimal setup and knowledge on how its used, it’s a good start for people who want to move from game engines and want to create thing tinkered to their needs.

## Controlling the 2D camera with views (SFML / Learn / 2.6 Tutorials)SFML (Graphics API)

SFML is a multi platform library with modules that handle, graphics, windows, system, audio and networking. SFML supports bindings in C#, Java, Go, Python and Rust but is primarily a c++ library. Just like Raylib, SFML is simple to use, removing boilerplate code and advanced features by simplifying them. The graphics make use of OpenGL, providing efficient hardware acceleration.

SFML is designed to make games but can also be designed for multimedia applications as it provides | Figure 2.1: multiple view port game example the essentials for people to create any app they need. SFML also offers modularity by allowing developers to remove modules that aren’t needed.

However, Comparing to Raylib this API doesn’t support nearly as my platforms or language bindings, and isn’t as easy to pickup as Raylib. And SFML doesn’t have nearly as many features for games as Raylib does (due to raylib being focused on games development), not having features such as 3d models, textures, physics and cameras. SFML has no built in 3d functionality too, so making 3d game with it requires a lot more time and effort.

SFML does provide a faster experience, with less bloated features it doesn’t need to compile as much and allows developers more freedom to create features they want.

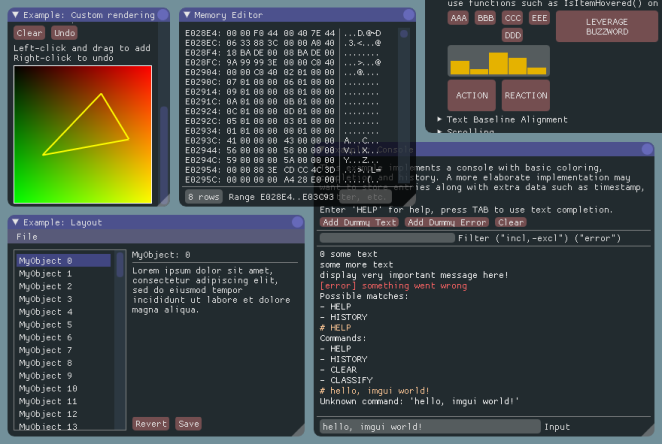
SFML also provides a low level GUI build to show performance, memory management and other tools to help developers with debugging.

Figure 2.2: GUI Example

## Cocos2d-x

Cocos2d-x is an open source lightweight mobile development library written in c++, interfacing with lua and java script. This library is purely 2d much like SFML. Cocos2d-x is also a dedicated games library providing features like scenes, sprites, actions, director, event handling, physics, particles, audio and tile maps, perfect for 2d games.

Scenes work much like worlds in unreal engine, where you create an instanced object that is used as a container for objects, camera placement and other functionality.

Cocos2d-x provides support for platforms like IOS, Android, MacOS and linux, this is done using CMake for cross platform compilation that can be integrated through major IDE’s such as Visual Studio and Xcode.

Cocos2d-x gets an advantage over other libraries with its support for mobile devices (both android and IOS), unlike Raylib and SFML. It also has more intuitive features that makes it easier to make games with compared to SFML like tile sheets for sprite animations and level building.

Cocos2D-x is useful for more advanced game development compared to raylib as it has features like a built in director to manage game loops, sprite, scene and node systems, particles and advanced UI systems, making Cocos2d-x more feature rich then raylib at the sacrifice of simplicity. But doesn’t support nearly as many platforms as raylib.

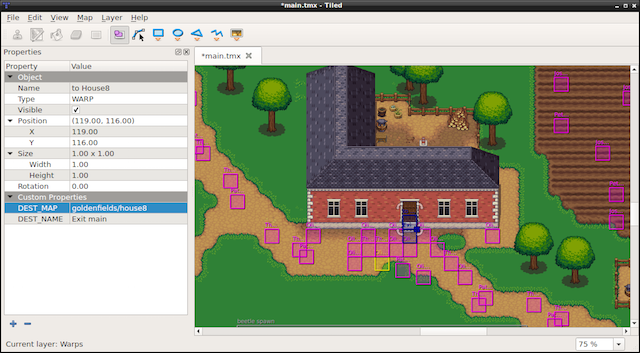


Figure 3.1: scene and node builder UI