



A complete decompilation of Retro Engine v4 and the menus from Sonic 1 and 2 (2013).

SUPPORT THE OFFICIAL RELEASE OF SONIC 1 & 2

- Without assets from the official releases, this decompilation will not run.
- You can get official releases of Sonic 1 & Sonic 2 from:
 - Windows
 - ★ Via Steam, from Sonic Origins
 - ★ Via the Epic Games Store, from Sonic Origins
 - iOS
 - ★ Sonic 1, Via the App Store
 - ★ Sonic 2, Via the App Store
 - ★ A tutorial for finding the game assets from the iOS versions can be found [here](#).
 - Android
 - ★ Sonic 1, Via Google Play
 - ★ Sonic 2, Via Google Play
 - ★ Sonic 1, Via Amazon
 - ★ Sonic 2, Via Amazon
 - ★ A tutorial for finding the game assets from the Android versions can be found [here](#).

Even if your platform isn't supported by the official releases, you **must** buy or officially download it for the assets (you don't need to run the official release, you just need the game assets).

If you want to transfer your save(s) from the official mobile version(s), the **Android pre-forever** file path is `Android/data/com.sega.sonic1` or `2/SGame.bin` (other versions may have different file paths). Copy that file into the decompilation's folder with the name `SData.bin`.

Additional Tweaks

- Added the built in script compiler from RSDKv5U.

-
- Added a built in mod loader and API, allowing to easily create and play mods with features such as save file redirection, custom achievements and XML GameConfig data.
 - Custom menu and networking system for Sonic 2 multiplayer, allowing anyone to host and join servers and play 2P VS.
 - Servers may be unreliable; this feature is more or less a proof of concept.
 - Egg Gauntlet Zone is playable in the Time Attack menu in Sonic 2, if you're using a version of the game that includes it.
 - There is now a `settings.ini` file that the game uses to load all settings, similar to Sonic Mania.
 - The dev menu can now be accessed from anywhere by pressing the `ESC` key if enabled in the config.
 - The `F12` pause, `F11` step over & fast forward debug features from Sonic Mania have all been ported and are enabled if `devMenu` is enabled in the config.
 - A number of additional dev menu debug features have been added:
 - `F1` will load the first scene in the Presentation stage list (usually the title screen).
 - `F2` and `F3` will load the previous and next scene in the current stage list.
 - `F5` will reload the current scene, as well as all assets and scripts.
 - `F8` and `F9` will visualize touch screen and object hitboxes.
 - `F10` will activate a palette overlay that shows the game's 8 internal palettes in real time.
 - Added the idle screen dimming feature from Sonic Mania Plus, as well as allowing the user to disable it or set how long it takes for the screen to dim.

How to Build

This project uses CMake, a versatile building system that supports many different compilers and platforms. You can download CMake here. **(Make sure to enable the feature to add CMake to the system PATH during the installation!)**

Get the source code

In order to clone the repository, you need to install Git, which you can get here.

Clone the repo **recursively**, using: `git clone --recursive https://github.com/Rubberduckycooly/Sonic-1-2-2013-Decompilation`

If you've already cloned the repo, run this command inside of the repository: `git submodule update --init --recursive`

Follow the build steps

Windows

To handle dependencies, you'll need to install Visual Studio Community (make sure to install the `Desktop development with C++` package during the installation) and vcpkg.

After installing those, run the following in Command Prompt (make sure to replace `[vcpkg root]` with the path to the vcpkg installation!): - `[vcpkg root]\vcpkg.exe install glew sdl2 libogg libvorbis --triplet=x64-windows-static` (If you're compiling a 32-bit build, replace `x64-windows-static` with `x86-windows-static`.)

Finally, follow the compilation steps below using `-DCMAKE_TOOLCHAIN_FILE=[vcpkg root]/scripts/buildsystems/vcpkg.cmake -DVCPKG_TARGET_TRIPLET=x64-windows-static -DCMAKE_PREFIX_PATH=[vcpkg root]/installed/x64-windows-static` as arguments for `cmake -B build`. - Make sure to replace each instance of `[vcpkg root]` with the path to the vcpkg installation! - If you're compiling a 32-bit build, replace each instance of `x64-windows-static` with `x86-windows-static`.

Linux

Install the following dependencies: then follow the compilation steps below: - **pacman (Arch):** `sudo pacman -S base-devel cmake glew sdl2 libogg libvorbis` - **apt (Debian/Ubuntu):** `sudo apt install build-essential cmake libglew-dev libglfw3-dev libsdl2-dev libogg-dev libvorbis-dev` - **rpm (Fedora):** `sudo dnf install make gcc cmake glew-devel glfw-devel sdl2-devel libogg-devel libvorbis-devel zlib-devel` - **apk (Alpine/PostmarketOS)** `sudo apk add build-base cmake glew-dev glfw-dev sdl2-dev libogg-dev libvorbis-dev` - Your favorite package manager here, make a pull request

Android

Follow the android build instructions here.

Compiling

Compiling is as simple as typing the following in the root repository directory:

```
1 cmake -B build
2 cmake --build build --config release
```

The resulting build will be located somewhere in `build/` depending on your system.

The following cmake arguments are available when compiling: - Use these on the first `cmake -B build` step like so: `cmake -B build -DRETRO_DISABLE_PLUS=on`

RSDKv4 flags

- `RETRO_REVISION`: What revision to compile for. Takes an integer, defaults to 3 (Origins).
- `RETRO_DISABLE_PLUS`: Whether or not to disable the Plus DLC. Takes a boolean (on/off): build with `on` when compiling for distribution. Defaults to `off`.
- `RETRO_FORCE_CASE_INSENSITIVE`: Forces case insensitivity when loading files. Takes a boolean, defaults to `off`.
- `RETRO_MOD_LOADER`: Enables or disables the mod loader. Takes a boolean, defaults to `on`.
- `RETRO_NETWORKING`: Enables or disables networking features used for Sonic 2's 2P VS mode. Takes a boolean, defaults to `on`.
- `RETRO_USE_HW_RENDER`: Enables the Hardware Renderer used by the main menu and touch controls UI. Takes a boolean, defaults to `on`.
- `RETRO_ORIGINAL_CODE`: Removes any custom code. *A playable game will not be built with this enabled.* Takes a boolean, defaults to `off`.
- `RETRO_SDL_VERSION`: *Only change this if you know what you're doing.* Switches between using SDL1 or SDL2. Takes an integer of either 1 or 2, defaults to 2.

Unofficial Branches

Follow the installation instructions in the readme of each branch. * For the **PlayStation Vita**, go to Xeeynamo's fork. * For the **Nintendo Switch**, go to heyjoeway's fork. * For the **Nintendo 3DS**, go to JeffRuLz's fork. * A New Nintendo 3DS is required for the games to run smoothly. * To play it on the web using **Wasm**, go to mattConn's fork.

Because these branches are unofficial, we can't provide support for them and they may not be up-to-date.

Other Platforms

Currently the only supported platforms are the ones listed above, however the backend uses libogg, libvorbis & SDL2 to power it (as well as tinyxml2 for the mod API and asio for networking), so the codebase is very multiplatform. If you're able to, you can clone this repo and port it to a platform not on the list.

Server

The multiplayer server requires Python 3.8 or later. You can download Python [here](#). To use the server, open Command Prompt in the folder `Server.py` is located in, then run the command `py -3 Server .py [local IPv4 address] [port] debug`. You can find your local IPv4 address using the command `ipconfig`. Note that the C++ server found in the [Server](#) folder has been deprecated and no longer works. It has been kept in the repo for reference purposes.

FAQ

You can find the FAQ [here](#).

Special Thanks

- st×tic for helping me fix bugs, tweaking up my sometimes sloppy code and generally being really helpful and fun to work with on this project.
- The Weigman for creating the header you see up here along with similar assets.
- Everyone in the Retro Engine Modding Server for being supportive of me and for giving me a place to show off these things that I've found.

Contact:

Join the Retro Engine Modding Discord Server for any extra questions you may need to know about the decompilation or modding it.