Sonic Rhythm Modding

The Informative Guide for Modders by Yours Truly

Sonic Rhythm is a mod to Sonic 3 A.I.R. that completely transforms the game into a rhythm game. For those who are looking to mod, I got great news for you. Not only is there support for it, but this document was also created to help you do so. Happy modding!

Table of Contents

Skinning

As someone who has played some rhythm games, I know the importance of skins in their respective communities, and I also know how useful a great skin can be. While I believe the options that are presented in Sonic Rhythm will allow many players to play without skins; I also believe that skinning is important to rhythm games to truly make a player's experience customizable and worthwhile.

Making a skin mod for Sonic Rhythm is quite easy to do. You may use the template skin located on the GitHub wiki as a base to do so. Then its simply a matter of replacing sprites with your very own.

Scripting

There is a script file located in the example skin template (or within Sonic Rhythm's script as `render-gameplay.lemon`) that can be modified for making your skin truly customized. It features a range of comments and clean code to make it easier to understand for non-scripters, though I would highly recommend learning scripting beforehand regardless.

All scripts should be self-explanatory; though if you need help understanding one of them do feel free to reach out to me or someone else who has made scripted skins before to ask.

Charting

Sonic Rhythm features a unique charting style that is not present in other Rhythm Games due to limitations involving the Oxygen engine. It involves manually placing each individual note within a .lemon script file.

You should have a firm grasp on hexadecimals, script modding within Oxygen Engine, and basic note pattern charting (jacks, jumps, streams, thrills, etc.) before beginning, as well as a calculator that could convert integer numbers to hexadecimal and back.
Additionally, for anyone who knows rhythm games, Sonic Rhythm runs on a frame-based system rather than a millisecond based one due to, again, limitations

Setting Up a Chart

Since all songs are currently hardcoded on the menu, you will have to replace one of the songs for charting. However, it is quite simple to do so. Here, I will example a replacement song over Endless Mine.

Go into `scripts/util/audioselection.lemon`. This is where the game selects which song to play and sets the BPM when needed as well.

Find Endless Mine and change the song.BPM to the BPM of the song you are charting (if you are unsure about what the BPM is, search it up or ask the original composers). Then change the "24" in Audio.playAudio into your own audio file (see Audio Modding in the modding handbook).

You may than go into `scripts/charts/endlessmine.lemon` and begin charting.

Notes & Events

setupNote(u8 type, s64 delay, bool isMine, u64 hitsound)

Creates a note that can be hit using the keyboard.

- Type tells the game which lane the note is on. It goes from 1 to 4, corresponding to each of the four lanes that notes can arrive on.
- Delay tells the game how late the note should come in. A delay of 0 for example tells the game to setup a note that occurs when the song begins on the same frame.
- isMine is currently under development and should not be used now. However, in the future, it creates a mine that hurts the player should they hit it.
- Hitsound gives the note a sound to play whenever the player hits it. You can even define your own sounds and have them play. If you do not want a hitsound to play, leave it blank as "". A hitsound does not play if they is turned off in the settings.

setupNote(u8 type, s64 delay, u64 hitsound)

Shortcut for setting up a note without a mine. Refer to *setupNote(u8 type, s64 delay, bool isMine, u64 hitsound)* for documentation.

setupLongNote(u8 type, s64 delay, u16 length, bool isMine)

Creates a long note that can be held with the keyboard.

- Type tells the game which lane the long note is on. It goes from 1 to 4, corresponding to each of the four lanes that notes can arrive on.
- Delay tells the game how late the long note should come in. A delay of 0 for example tells the game to setup a long note that occurs when the song begins on the same frame.
- Length tells the game how long your long note should be. There is no limit towards how long it can go but do keep in mind the Oxygen Engine’s 1024 sprite limit at any given moment.
- isMine is currently under development and should not be used now. However, in the future, it creates a minefield that hurts the player should they hold it.

setupLongNote(u8 type, s64 delay, u16 length)

Shortcut for setting up a long note without a mine. Refer to *setupLongNote(u8 type, s64 delay, u16 length, bool isMine)* for documentation.

eventNote(s64 delay, u64 eventType, u64 eventData)

Creates an event note that in the future will cause a multitude of effects that can make your chart more interesting or unique. Think of it as an invisible note that can cause effects and cannot be hit.

- Delay tells the game how late the long note should come in. A delay of 0 for example tells the game to setup a long note that occurs when the song begins on the same frame.
- eventType tells the game which event to run. The following events are currently available to use:
  - "End Song" – Fades out the music and ends the song roughly 300 frames later. All charts should have this somewhere near the end.
- eventData gives an event data to use. Read the events for more info on any specific event. If an event does not have data associated with it, it is completely fine to leave it as a 0 or something similar of nature.