Music Manager.

A plugin for Godot that allows for easy music implementation. The plugin UI is a DAW like interface with a single tempo measured in crotchets/quarter notes and time signature track and an optional amount of audio tracks with volume and bus control.

The manager has a workflow of [Level 1 > Stage 1, Stage 2, etc] [Level 2 > Stage 1, Stage 2, etc] e.g. Level – Boss fight, Stage – Phase2. The stages hold 3 pieces of data – start time, end time and selected tracks.

The interface has a two keyboard shortcuts:
Save is ctrl + S however the plugin saves automatically with the resource_saved signal. Data is saved to a custom resource in the SaveData directory.
To duplicate a sound clip is shift + D (couldn't use ctrl + D as the plugin doesn't override the other docks).

Left clicking a level, track controls, clip or stage will show a delete option.

Scripting.

For scripting, there are 4 methods to make it so one line of code can switch between music. All these methods are void functions.

set_level_and_play(String: level name, String: stage name, float: volume fade in seconds, enum value INSTANT, ON_BEAT, ON_BAR or ON_LOOP)
Sets the level and plays stage. This should be the first method to be called to start playing music instantly or to quickly change the level without stopping.

play_stage(String: stage name, enum value INSTANT, ON_BEAT, ON_BAR or ON_LOOP, float: volume fade in seconds, bool: overdub/layering will add and remove tracks whilst playing if the time selection is the same as the last stage)

set_level(String: level name) Only use this while music is stopped to get a level ready, it's slightly quicker than using set_level_and_play().

stop(float: volume fade in seconds)

There's also the 'playing' property that can be set directly to pause and play the music.

Example:
```
Music.set_level_and_play("All Menus", "Main Theme")
```

```
if go_to_map_selection:
    Music.play_stage("Map Selection", Music.ON_BEAT)
```

Signals.

For syncing the game with music there are the next_beat, next_bar, on_loop and clip_playing signals.

next_beat(beat number from stage start)
next_bar(measure number from beginning)
on_loop
clip_playing(sound clip name, track number starting from 0) this calls as the sound clip starts playing.

Example:

```
Music.connect("next_beat", self, "do_something")

func do_something(beat):
    #use fmod (float modulus) or % to find beats.
    #assuming in common time, 4/4.
    if beat % 4 == 0:
        #First beat of bar, same time as on_bar with beat value instead of measure parameter.
    elif fmod(beat + 1, 2):
        #Off beats
```