

# **Super-charging content production with Godot addons**

Building a pipeline to produce game  
content predictably.



 [/bitbraindev](https://discord.com/channels/@me/911111111111111111)

# @bitbrain

- working on a dwarven pixelart RPG! 💎🔨
- maintainer of **pandora** and **beehave** 🐝
- Godot = ❤️

# Godot's Design Philosophy

“ [...] new features from the core developers often focus on what will benefit the most users first.“

source:

[https://docs.godotengine.org/en/stable/getting\\_started/introduction/godot\\_design\\_philosophy.html](https://docs.godotengine.org/en/stable/getting_started/introduction/godot_design_philosophy.html)

# Terminology

- **addon** = third-party code and assets (including plugins)
- **plugin** = a Godot editor plugin (requires `plugin.cfg`)
- **extension** = extends Godot's core via C++ through the GDExtension interface (requires `*.gdextension`)
- **module** = compiled with Godot's core

# Plugins

Example: addons/dialogic/plugin.cfg

```
[plugin]

name="Dialogic"
description="Create dialogs, characters and scenes to display conversations in your Godot games.
https://github.com/coppolaemilio/dialogic"
author="Emi, Jowan Spooner, Exelia, and more!"
version="2.0-Alpha-10 (Godot 4.1.2)"
script="plugin.gd"
```

source:

<https://github.com/coppolaemilio/dialogic/blob/main/addons/dialogic/plugin.cfg>

# GDExtensions

Example: addons/fmod/fmod.gdextension

```
[configuration]
entry_symbol = "fmod_library_init"
compatibility_minimum = 4.1

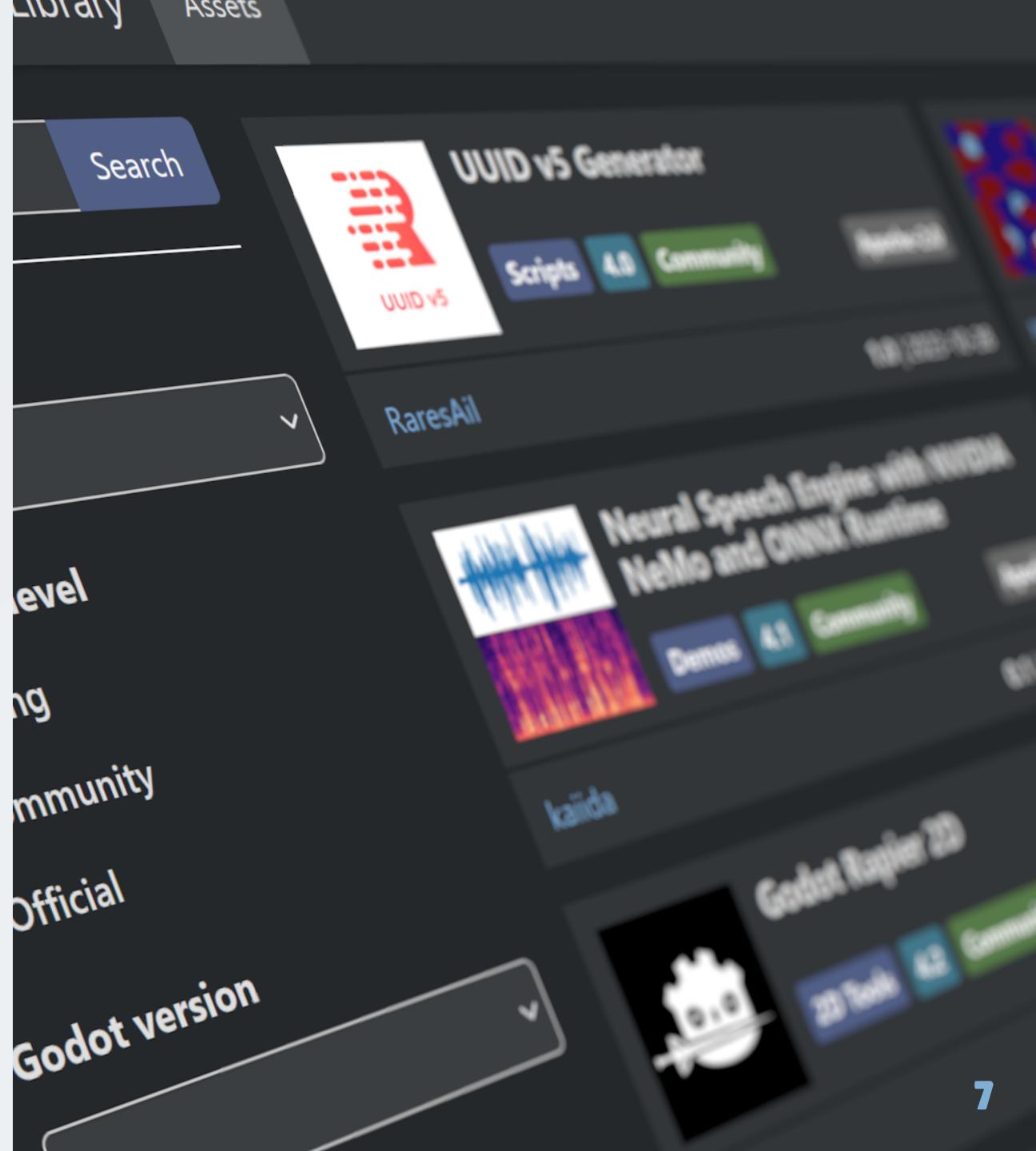
[libraries]
windows.editor.x86_64 = "res://addons/fmod/libs/windows/libGodotFmod.windows.editor.x86_64.dll"
windows.debug.x86_64 = "res://addons/fmod/libs/windows/libGodotFmod.windows.template_debug.x86_64.dll"
windows.release.x86_64 = "res://addons/fmod/libs/windows/libGodotFmod.windows.template_release.x86_64.dll"
```

source:

<https://github.com/utopia-rise/fmod-gdextension/blob/master/demo/addons/fmod/fmod.gdextension>

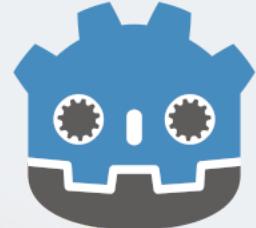
# Addon Ecosystem

Exploring the categories of Godot addons.

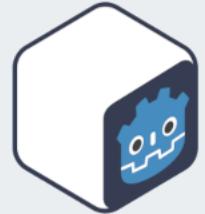


# Dialogic

fmod® GDExtension



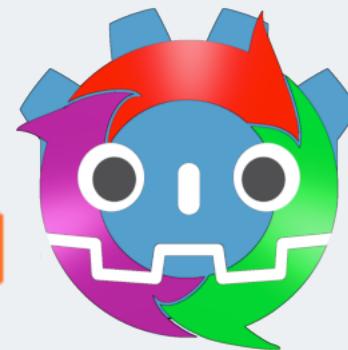
## GDShell



Pandora



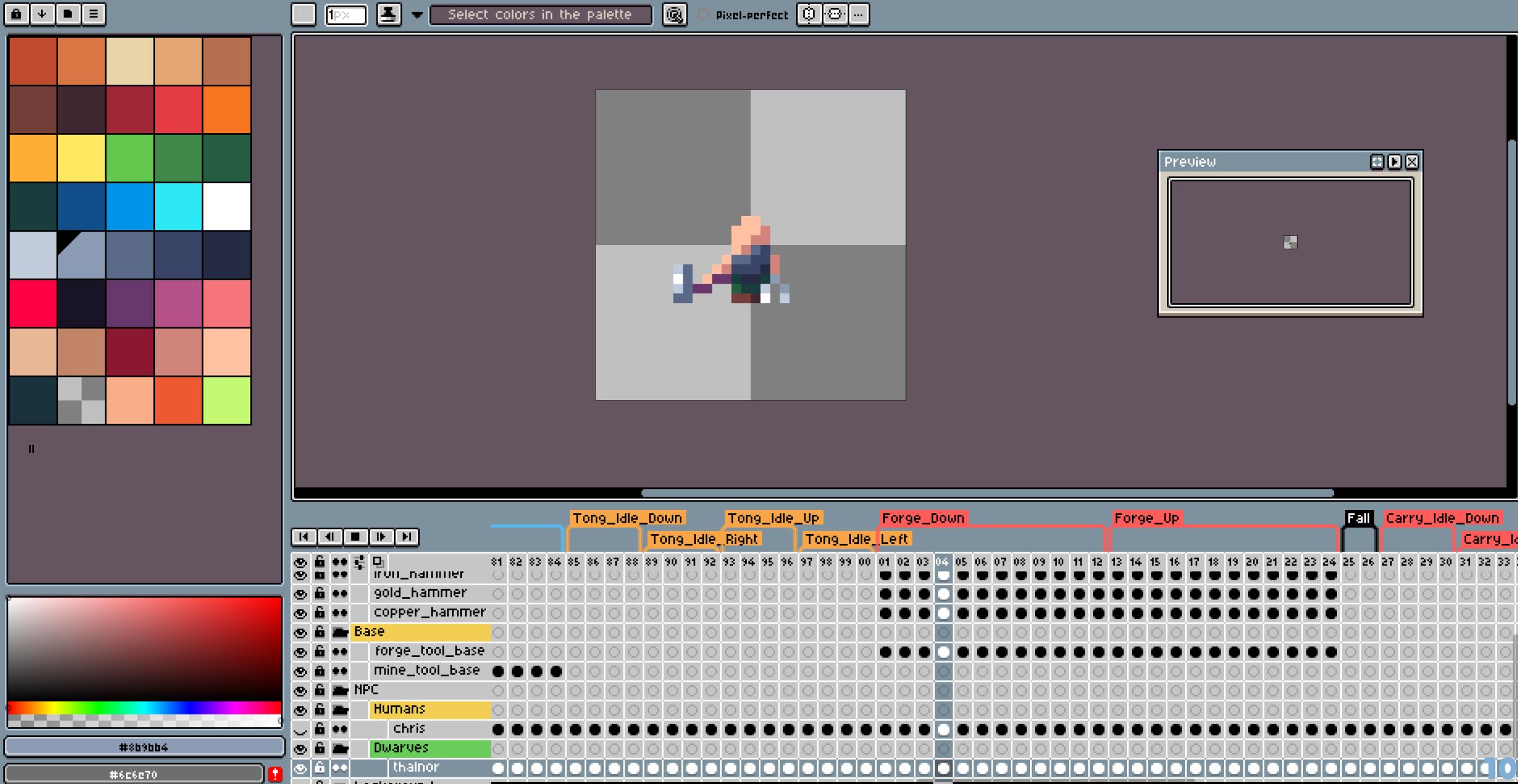
Beehave



# Workflow addons

Accelerate and  
automate common  
processes.



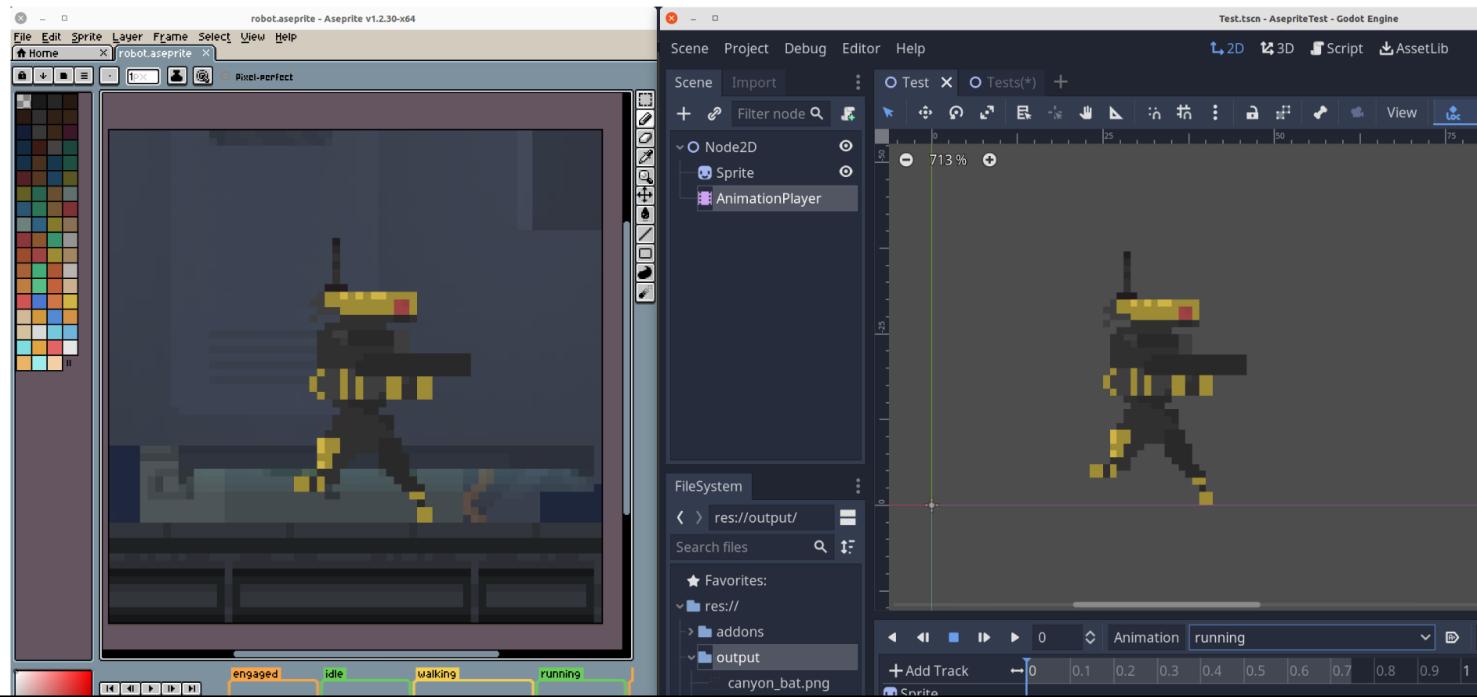


# Godot Aseprite Wizard (Godot 4) [🔗](#)



Godot plugin to help import Aseprite animations to AnimationPlayers, AnimatedSprites 2D/3D and SpriteFrames.

*This branch supports Godot 4. For Godot 3 docs and code check the [godot\\_3](#) branch. You can find more details about the differences between Godot 3 and Godot 4 on issue [#70](#).*



## Contributors 10



## Languages

- GDScript 100.0%



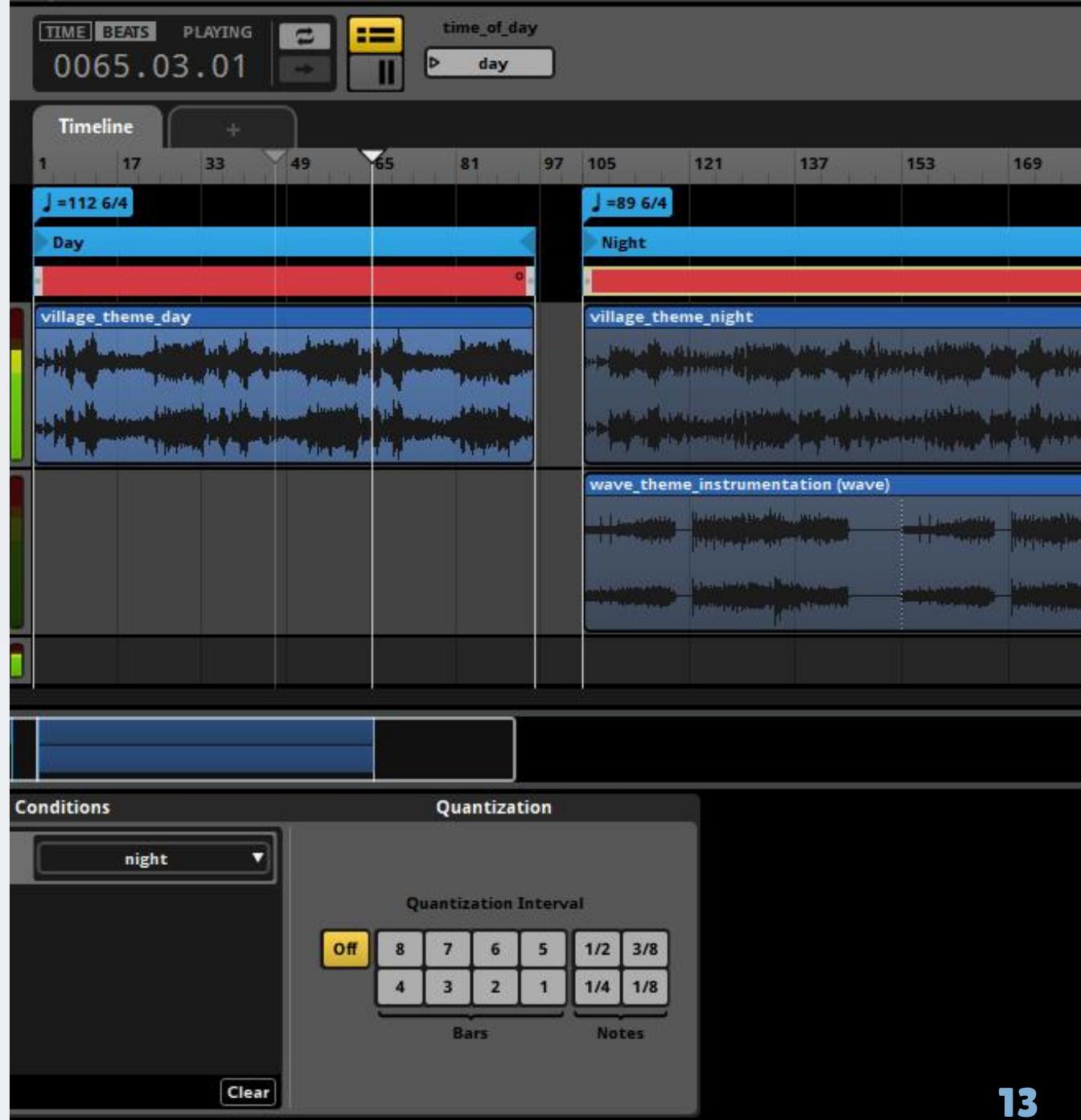
# Integration addons

Connect any tool directly into  
Godot Engine.

# FMOD GDExtension

[utopia-rise/fmod-gdextension](#)

[alessandrofama/fmod-for-godot](#)





Godot extension that integrates the Jolt physics engine.

[godot-jolt/godot-jolt](#)

- works with `CharacterBody3D` and other familiar Godot nodes out of the box (drop-in replacement)

# Feature extending addons

- show example of feature extending addons
- add image

# Template repositories

- show example of template repo addon
- add image

# Addon discovery

- Official: <https://godotengine.org/asset-library>
- Useful: <https://github.com/godotengine/awesome-godot>
- Goldmine: <https://github.com/search?q=godot%2Baddon>
- Supportive: <https://itch.io/search?q=godot%2Baddon>

# Content Pipelines

Building reliable workflows that produce content for your game.

# Task Complexity

S	M	L	XL
fix translation	add new language	add voicelines	dialogue system
change color of sword	update sword animation	add new weapon type	itemization system
fix sound timing	add new sound effect	dynamic sound playback	integrating FMOD
fix level collision	rework existing level	add new level	procedural level generation

# Identify the bottleneck

- Is the task a one-off?
- Does finishing the task create new work after?
- How much is human error a factor?
- Is it a common problem a lot of people solved before?

# Production Point Principle

by ❤️ HeartBeast



# Accellarate Prototyping

Image: An artist's canvas with various game elements quickly added.

Engagement: "Can anyone share an experience where an addon saved them significant time during prototyping?"

Key Points:

Highlight the strengths of addons: Rapid development, quick feature integrations, testing ideas swiftly.

# Accellarate Production

Image: An artist's canvas with various game elements quickly added.

Engagement: "Can anyone share an experience where an addon saved them significant time during prototyping?"

Key Points:

Highlight the strengths of addons: Rapid development, quick feature integrations, testing ideas swiftly.

# There is no silver bullet

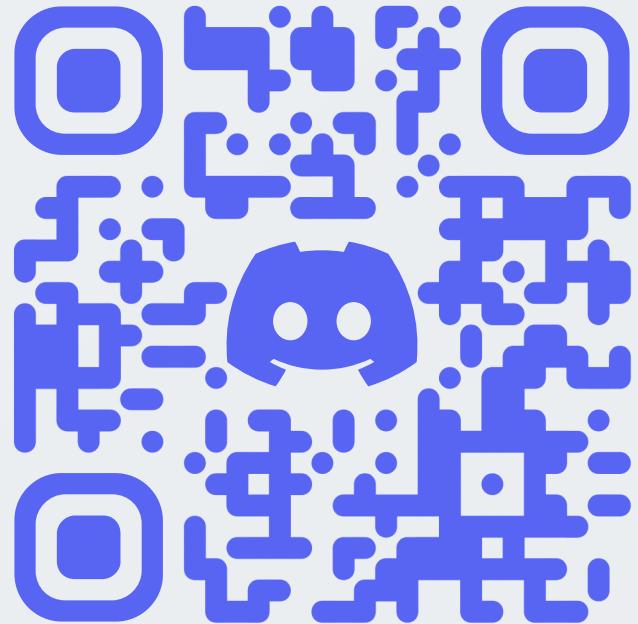
- the success of your game may be dependent on the quality of the addon
- building it yourself means you can own the design and architecture of it fully
- addons may get out of date and you will spend time trying to patch them yourself in case Godot updates
- debugging code that is not yours is often not fun and wastes a lot of time
- how can you be sure the addon code is performant and does what it says it does? What guarantees does the addon give you?

# Properties of a good addon

A **good** addon should be:

- useful
- well-documented
- tested
- **maintained or archived**
- compatible

# Questions?



**[youtube.com/@bitbraindev](https://youtube.com/@bitbraindev)**