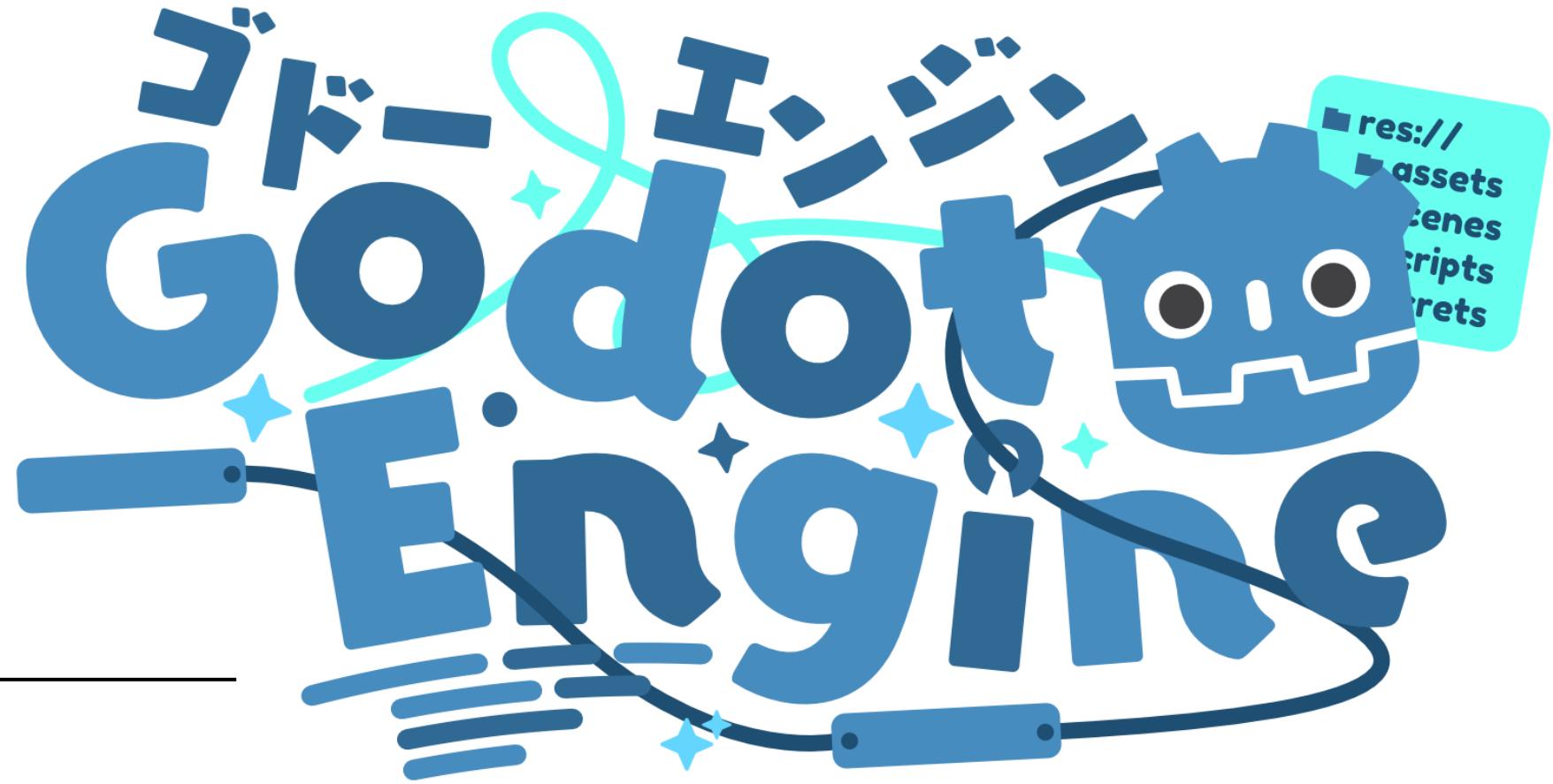


谢天

开源游戏引擎Godot初探

等待戈多



Disclaimer

声明



1. 分享内容使用的Godot版本基于4.2官方版
2. Opinions are my own, 我的观点不代表引擎组 or 公司的观点
3. 时间有限，只能分享一些我觉得有趣的方面

图片来于官方4.2版本的截图

Scene Node System

Editor

渲染

脚本

Final Thoughts

介绍

一句话概括

Godot 是一个 **易用的 开源的 轻量的** 游戏引擎

Godot怎么读

爱怎么读怎么读

French

go-do

English

GOD-oh

English-v2

g-DOUGH

Juan

go-do-t

[Waiting for Godot](#)

人们永远想要更多，永远等不到它的到来。

Different languages have different pronunciations for Godot and we find it beautiful.

<https://www.youtube.com/watch?v=p-up1YCRDmA>

介绍

游戏展示

游戏展示



土豆兄弟

2D、俯视角、Roguelike

游戏展示



Cassette Beasts

妖怪收集、回合制、
开放世界、RPG

https://store.steampowered.com/app/1321440/Cassette_Beast/

以及其他…

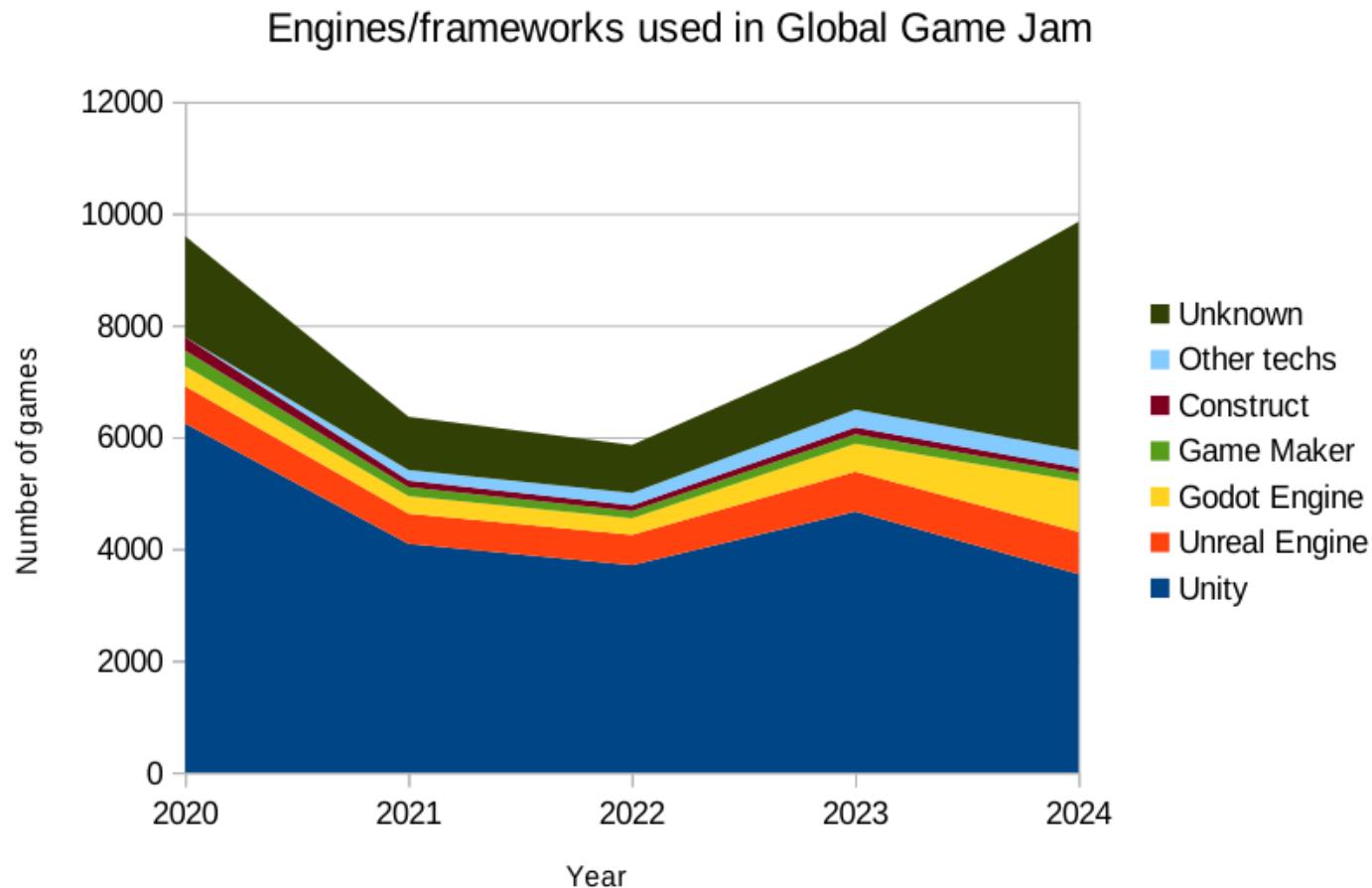


介绍

知名度

知名度

GGJ

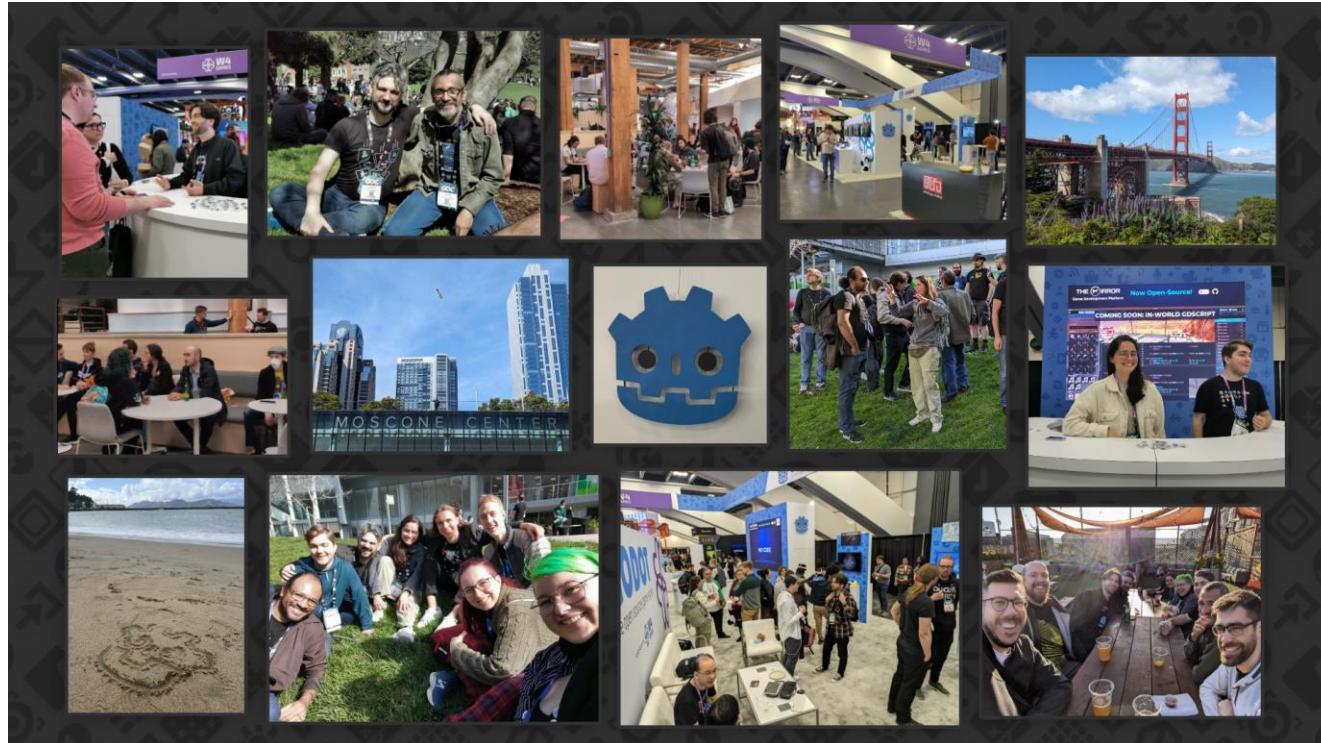


Rémi: “*Huge growth for Godot, almost double the amount of games (from 500 to 900+), and going from 8% to 16% "share" of the games for which data is provided.*”

<https://twitter.com/Akien/status/1751904118132683081>

知名度

GDC



<https://twitter.com/Akien/status/1751904118132683081>



jonbonazza 4:54 AM

Or rather, how did it go? Its over now, i believe



clayjohn 8:14 AM

It went really well. Godot is starting to become well known and a lot of potential partners are taking us seriously. Of course, we still have to convince them that partnering with us means giving us money without an expected ROI.

Godot is used at all levels of the industry now (a couple AAA game studios have switched to Godot for internal prototyping)



jonbonazza 8:15 AM

Ohhh cool to hear!



clayjohn 8:15 AM

It seems the biggest blockers to Godot adoption are no longer big technical issues, it is mostly ecosystem stuff (SDKs, file formats, asset store) and workflow papercuts (i.e. profiling tools, advanced settings etc.)



ajreckof 11:42 AM

that's really great to hear

知名度

OSS排名

Projects

Sort By: Commit Velocity | Community Votes | Fast Moving

Rank	Project	Open Core Companies	Score	Commits	Contributors	Started
1	LLVM Compiler	Modular AI	1000	503,964	5,089	2001
2	FreeBSD Operating System		998	950,313	2,703	1973
3	Linux Operating System	RedHat (IBM), SUSE, Rocky Linux	998 ▼1	1,251,891	26,556	2002
4	Rust Programming Language	Tembo	997	210,849	6,247	2010
5	OpenShift Container Management	RedHat (IBM), Microsoft, AWS / Amazon, IBM	997	151,530	3,061	2014
6	PyTorch Python Library, Machine Learning Framework	Modular AI, Yandex	994	69,055	4,283	2012
7	Zephyr Real Time Operating System (RTOS)		994	93,541	2,376	2014
8	ClickHouse Database	Yandex, ClickHouse, Aiven, Altinity, ChistaData	991	140,152	2,060	2008
9	Godot Game Engine	W4 Games	990 ▼2	76,444	3,856	2013

<https://ossrank.com/>

<https://ossinsight.io/analyze/godotengine/godot#people>

相当活跃的开发氛围



Contributors 2,469



[+ 2,455 contributors](#)

介绍

作为Godot灵魂的开源

开源

MIT License 的 FOSS (free and open source software)

免费

You are free to use Godot Engine, for any purpose

可魔改

You can study how Godot Engine works and change it

可分发

You can distribute unmodified and changed versions of Godot Engine, even commercially and under a different license (including proprietary). The only restriction to this freedom is that you need to distribute the copyright notice and license statement of Godot Engine whenever you redistribute it.

As if you made it yourself.

开源的保证

FOSS的优势

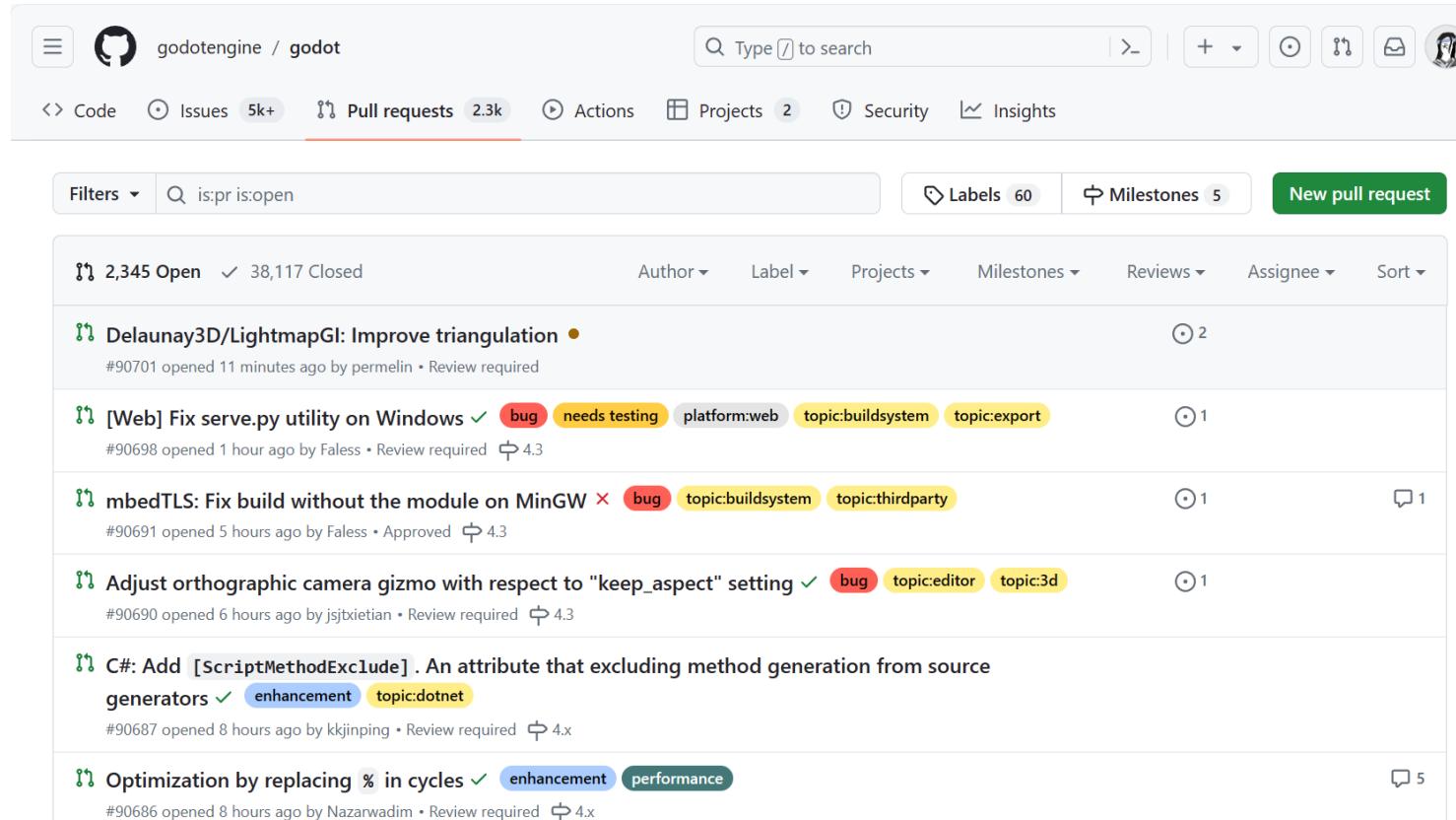
Godot can't be sold or purchased by any company.

It was built with 0% corporate greed and 100% passion. No surprises, no hidden fees, no strings attached.
The copyright to Godot's source is held collectively by every contributor.

<https://godotengine.org/governance/>

社区驱动

Open development



The screenshot shows the GitHub interface for the Godot engine's GitHub repository. The top navigation bar includes links for Code, Issues (5k+), Pull requests (2.3k, currently selected), Actions, Projects (2), Security, and Insights. Below the navigation is a search bar and a toolbar with icons for creating new pull requests and managing filters. A filter bar at the top of the main content area shows 'is:pr is:open'. The main content displays a list of open pull requests:

- #90701: Delaunay3D/LightmapGI: Improve triangulation (2 reviews pending)
- #90698: [Web] Fix serve.py utility on Windows (1 review pending)
- #90691: mbedTLS: Fix build without the module on MinGW (1 review pending)
- #90690: Adjust orthographic camera gizmo with respect to "keep_aspect" setting (1 review pending)
- #90687: C#: Add [ScriptMethodExclude]. An attribute that excluding method generation from source generators (4 reviews pending)
- #90686: Optimization by replacing % in cycles (5 reviews pending)

Each pull request entry includes a link to the issue, the number of reviews pending, and the user who opened it.

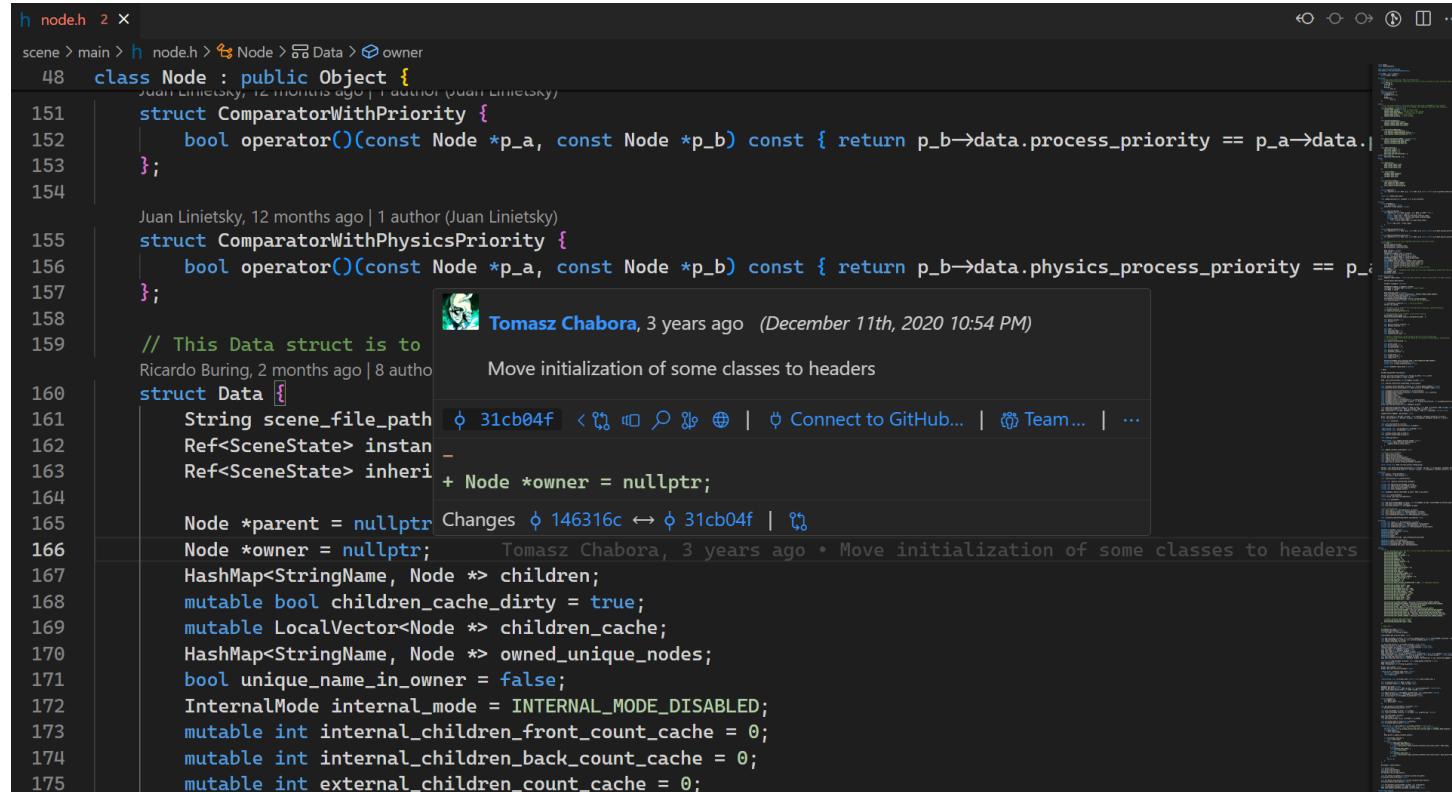
开发的可见性

All development (code that makes it into the engine, docs, website, etc) is made via Pull Requests, everyone is required to create them in order for their work to be included.

<https://github.com/godotengine/godot/issues>

社区驱动

Open development



The screenshot shows a GitHub pull request interface for a file named node.h. The code editor displays several classes and structures, including Node, ComparatorWithPriority, ComparatorWithPhysicsPriority, and Data. A specific commit by Tomasz Chabora is highlighted, showing a change in line 166 where the variable 'Node *owner' is initialized to nullptr. The commit message is "Move initialization of some classes to headers". The GitHub interface includes a sidebar with navigation links and a bottom bar with various icons.

优势

1. 公开透明，一切bug和feature都可见
2. 可追溯性，git blame马上就知道是谁写的代码了，非常适合学习如何实现某种功能
3. 因为这个可追溯性与pr的原子性，引擎本身修改起来也非常方便

<https://github.com/godotengine/godot/pulls>

社区驱动

Open discussion

Add ignorable columns to translation CSVs #8183

 Closed IntangibleMatter opened this issue on Oct 19, 2023 · 1 comment

IntangibleMatter commented on Oct 19, 2023

Describe the project you are working on
An RPG with lots of dialogue and many characters in complex conversations, featuring many characters at once.

Describe the problem or limitation you are having in your project
When I want to denote which character is speaking in a Translation file, either for my own use, or for the use of translators, there is no way to write a column which will be ignored in the CSV file when imported.

Describe the feature / enhancement and how it helps to overcome the problem or limitation
Make it so that csv locale columns that begin with an underscore are ignored when being imported into Godot. This means that you can add columns exclusively for those editing the files, without having to worry about weird invalid translation files being generated.

Describe how your proposal will work, with code, pseudo-code, mock-ups, and/or diagrams
No change to the engine API, just the ability to write files like this

```
key,en,_character
hey,Hey!,John
whatsup,what's up?,John
notmuch,Not much.,Jade
cool,"Cool, cool.",John
```

If this enhancement will not be used often, can it be worked around with a few lines of script?
This is a fundamental QOL thing for people working on translations. Additionally, it cannot be worked around without completely rewriting the CSV Translation importer.

New issue

Assignees
No one assigned

Labels
topic:core topic:import

Projects
None yet

Milestone
4.3

Development
No branches or pull requests

Notifications
 Customize
You're not receiving notifications from this thread.

3 participants


功能提案的可见性

Encourage maintainers, contributors, and community to open and discuss features and proposals in the Godot proposals repository.

More predictable development.

<https://github.com/godotengine/godot-proposals>

社区驱动

Community-minded



Juan Linietsky @reduzio · 2h

I opened a proposal for improving the usability of node and resource duplication in the [#GodotEngine](#) editor.

Feel free to leave feedback if this would be useful to you!

...

Priorities are set by project leadership and area maintainers based on the feedback of the community in bug reports, proposals, and discussions in the various community channels.

解决引擎用户的真实需求，用户可以表达意见参与讨论，一起塑造Godot的未来。

godotengine/godot-proposals

#9603 Node / Resource duplication usability improvements

reduz opened on April 26, 2024

Node / Resource duplication usability improvements · Issue #9603 · godotengin...

From github.com

<https://twitter.com/reduzio/status/1783846515309465689>

社区驱动

(非软广) Maintainer 中的唯一中国面孔

timothyqiu (LIVE) 年度大会员
Godot爱好者 afidian.net/@timothyqiu

关注 发消息

主页 动态 投稿 132 合集和列表 6 收藏 5 搜索视频、动态

代表作

- 基础框架 十分钟制作横版动作游戏 Basic Godot Project 10:25
- 制作《迷失岛2》游戏框架 | 面试题详解 | Godot游 01:52
- 【Godot】密室逃脱解谜游戏《节点》复刻版演示 03:20

充电 共111人充电中

公告

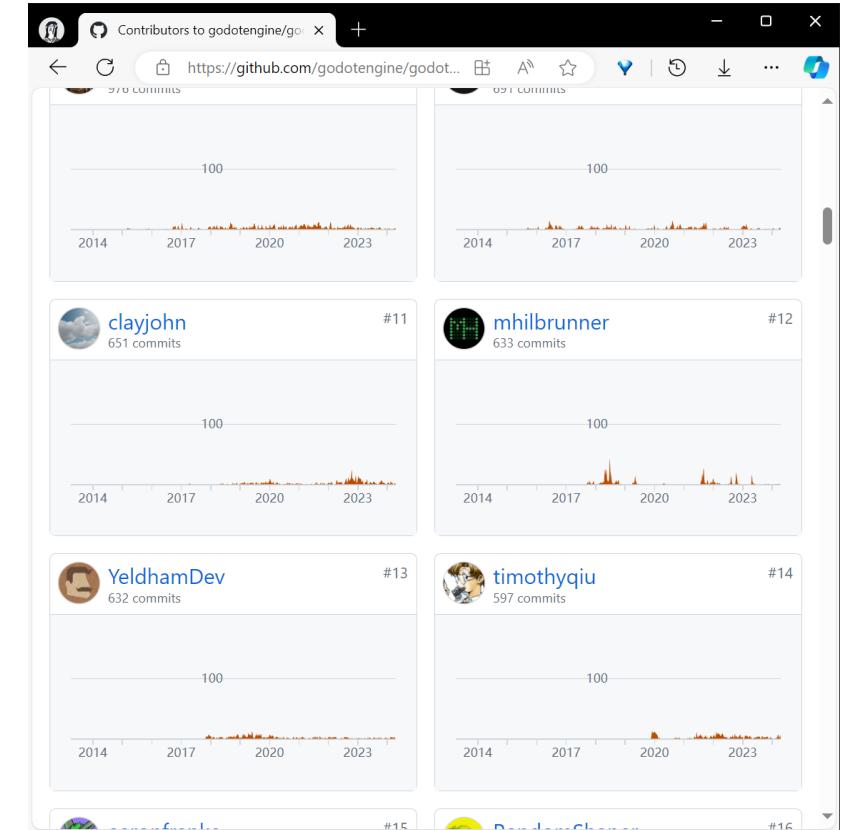
主要录一些 Godot 相关的教程，偶尔搬运，偶尔录一些无聊的游戏视频~
催更、支持我继续创作免费教程可以访问
<https://afidian.net/@timothyqiu>

直播间 关注直播间 1.3万

主播不在，关注后就能在动态收到开播的通知哦~
[前往TA的直播间 >](#)

合集·《勇者传说》Godot 4教程 30

- 基础框架 十分钟制作横版动作游戏 Basic Godot Project 10:25
- 相机的使用 Camera2D 09:44
- 节点 TileMap 的使用 TileSet & Terrain 17:12
- 实现视差背景 Parallax Background 15:56
- 运动控制 Coyote Time & Jump Buffering 11:57



<https://space.bilibili.com/7092/>

介绍与展示

Scene Node System

2

Editor

渲染

脚本

Final Thoughts

Scene Import

+ Filter: name, type,

Game

- Level
- InterfaceLayer
- PauseMenu

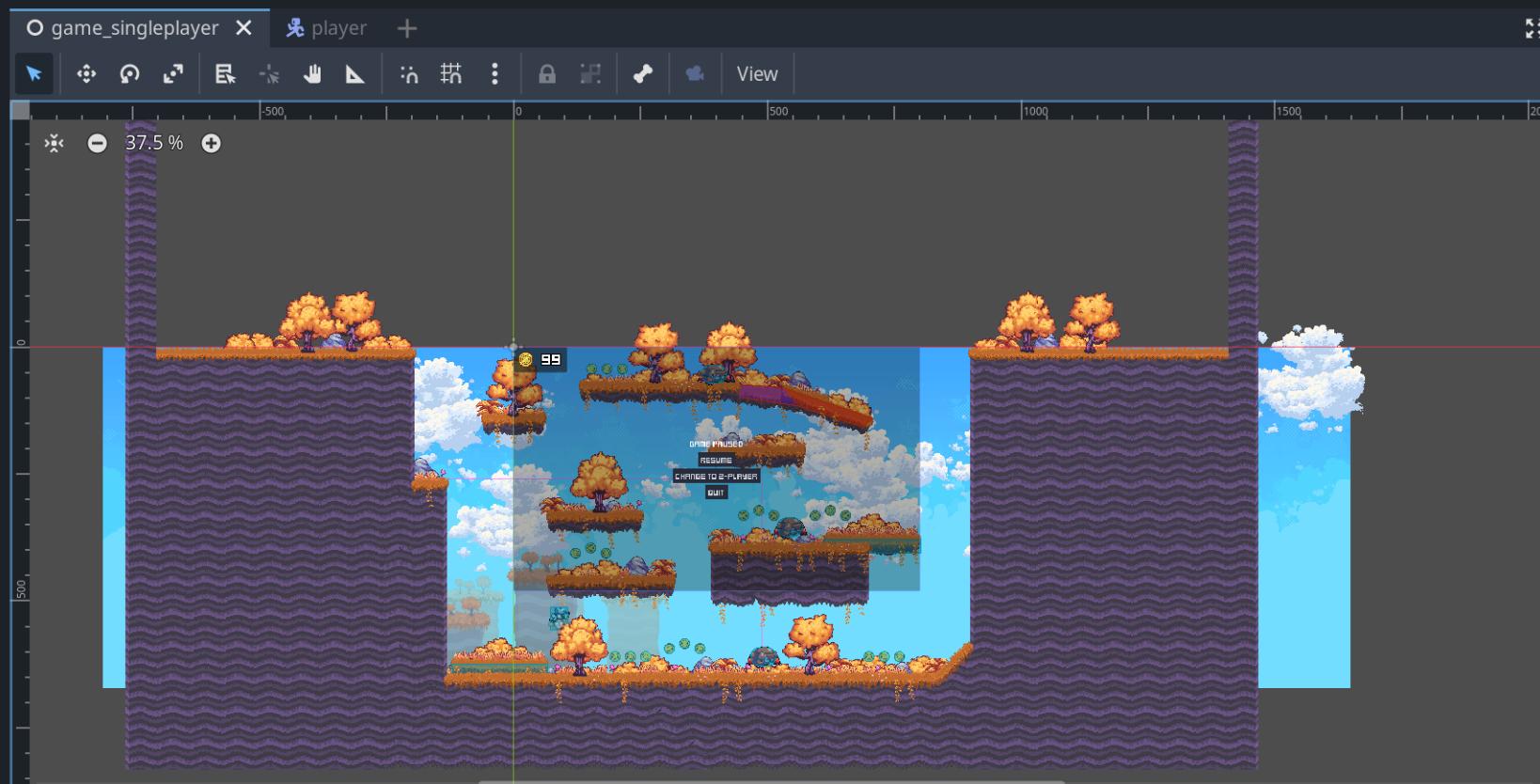
FileSystem

< > res://player/player.tscn

Filter Files

- gui
- level
- player
 - bullet.gd
 - bullet.tscn
 - bullet.webp
 - coin_pickup.wav
 - gun.gd
 - jump.wav
 - player.gd

player.tscn



Godot Engine v4.2.1.stable.official (c) 2007-present Juan Linietsky, Ariel Manzur & Godot Contributors.

- modules/gltf/register_types.cpp:63 - Blend file import is enabled in the project settings, but no Blender path is configured in the editor settings. Blend files will not be imported.

--- Debug adapter server started ---
--- GDScript language server started on port 6005 ---

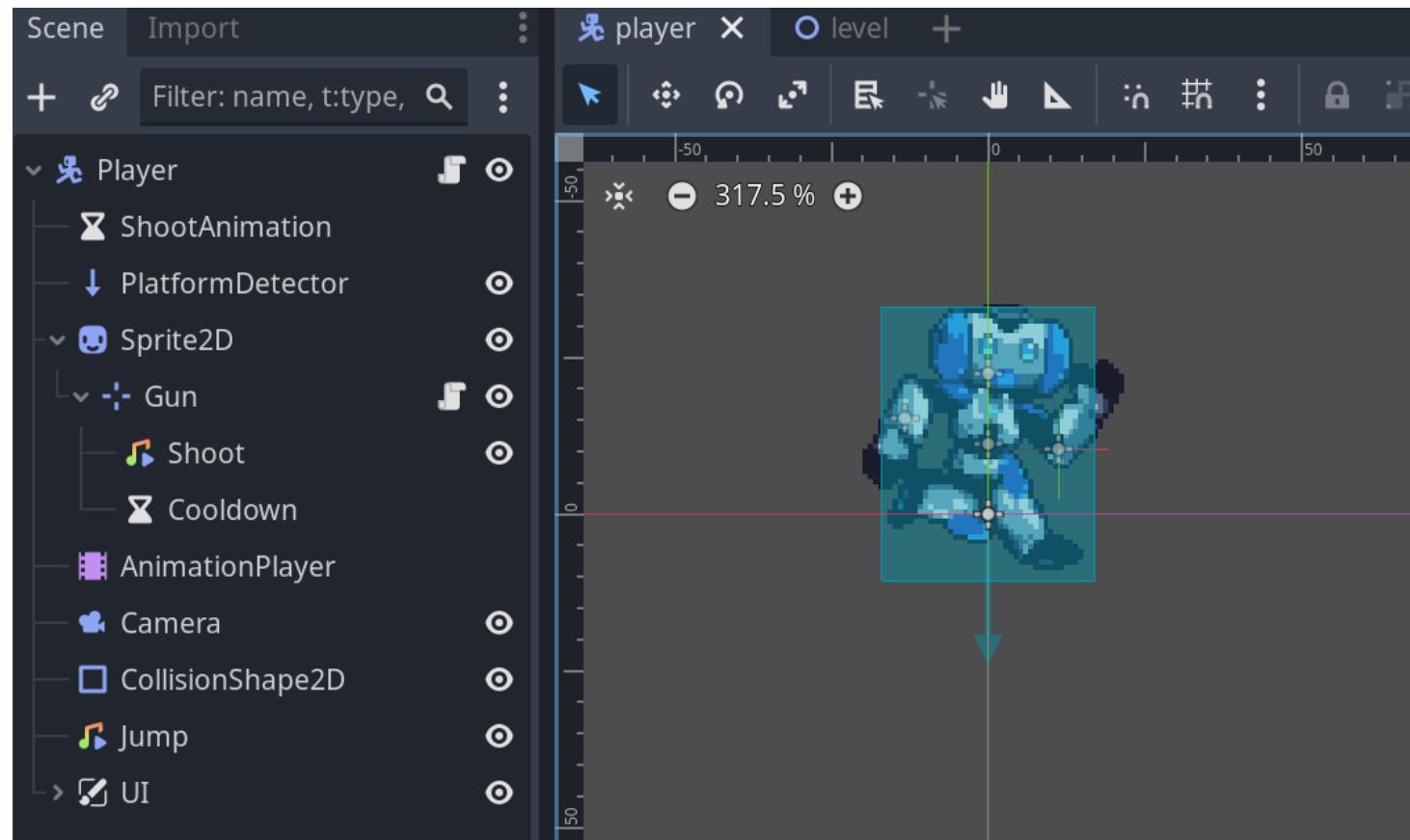
Filter Messages

Output Debugger Search Results Audio Animation Shader Editor

4.2.1.stable

Scene Node System

Node: Smallest building blocks



Node = 基础功能节点

1. 有名字
2. 有可被编辑的属性
3. 可以接受引擎的回调每帧Update
4. 可以被挂一个脚本拓展功能
5. 可以被当作其他Node的子Node (组合)

≈ 可以直接被挂到场景的Component

Its much cleaner to understand scenes at a glance.

Scene Node System

Node表达了功能

Control

UI相关的节点，例如Button, Label, TreeView, ItemList, BoxContainer等

Node2D

2D游戏会用到的节点，例如:
Sprite2D, Camera2D, CollisionShape2D, Skeleton2D, TileMap, VisibleOnScreenNotifier2D等

Node3D

3D游戏会用到的节点，除了2D Node的3D版，还包括：
MeshInstance3D, LightmapGI, Decal, GPUParticle3D, OccluderInstance3D, RayCast3D等

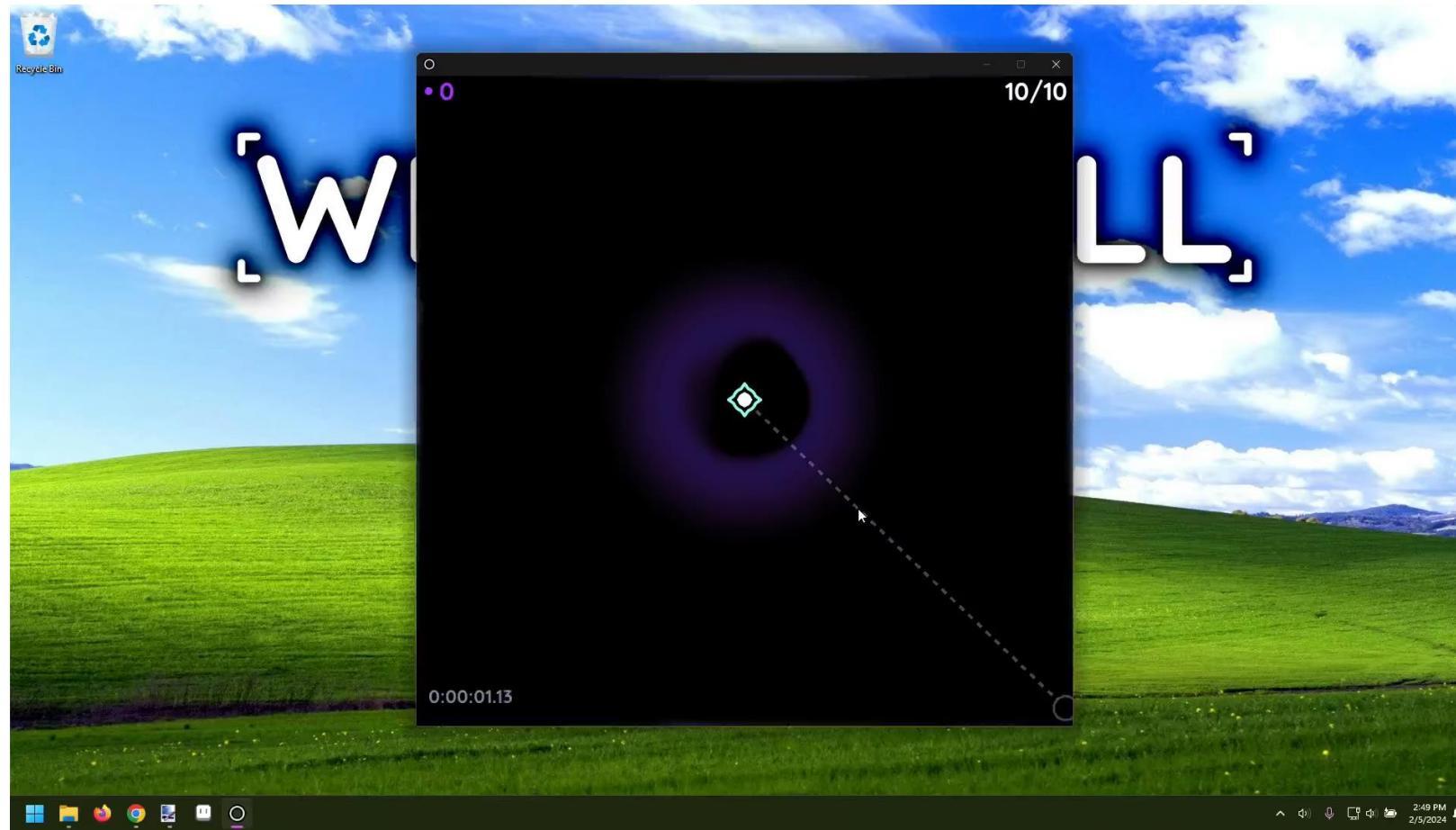
其他

其他节点，例如Viewport, Timer, HttpRequest, MultiPlayer, WorldEnvironment, ResourcePreloader等

Node作为引擎提供的功能组件基类：Godot provides an extensive library of base node types you can combine and extend to build more powerful ones. 2D, 3D, or user interface, you will do most things with these nodes.

Scene Node System

Demo: WindowKill



Window也是Node

When you load a scene in Godot, the engine will place it under the **root** (Window) node.

If you want to use multiple windows in your game, instantiate more of these and put children nodes to it.

Scene Node System

此外，脚本也可以继承Node

```
# item.gd

@icon("res://interface/icons/item.png")
class_name Item
extends Node

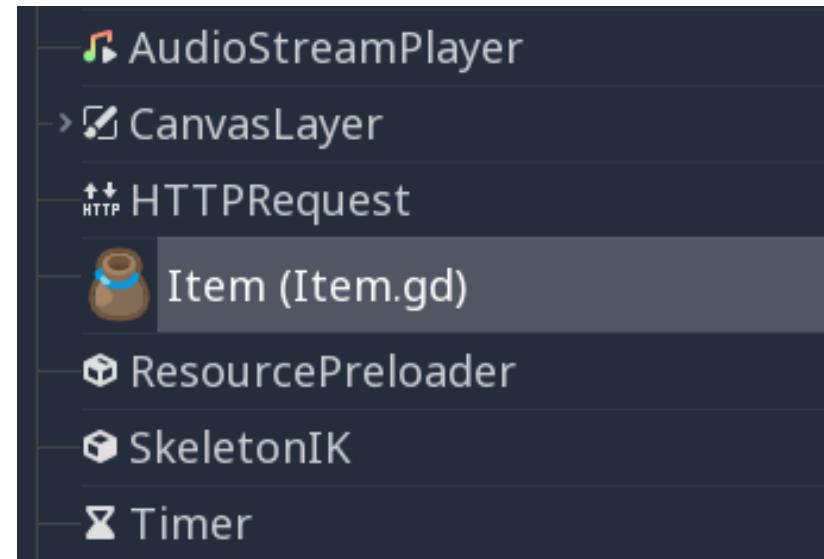
var s: String

func _ready():
    s = "I'm an item"

func _process(delta):
    print(s)
```

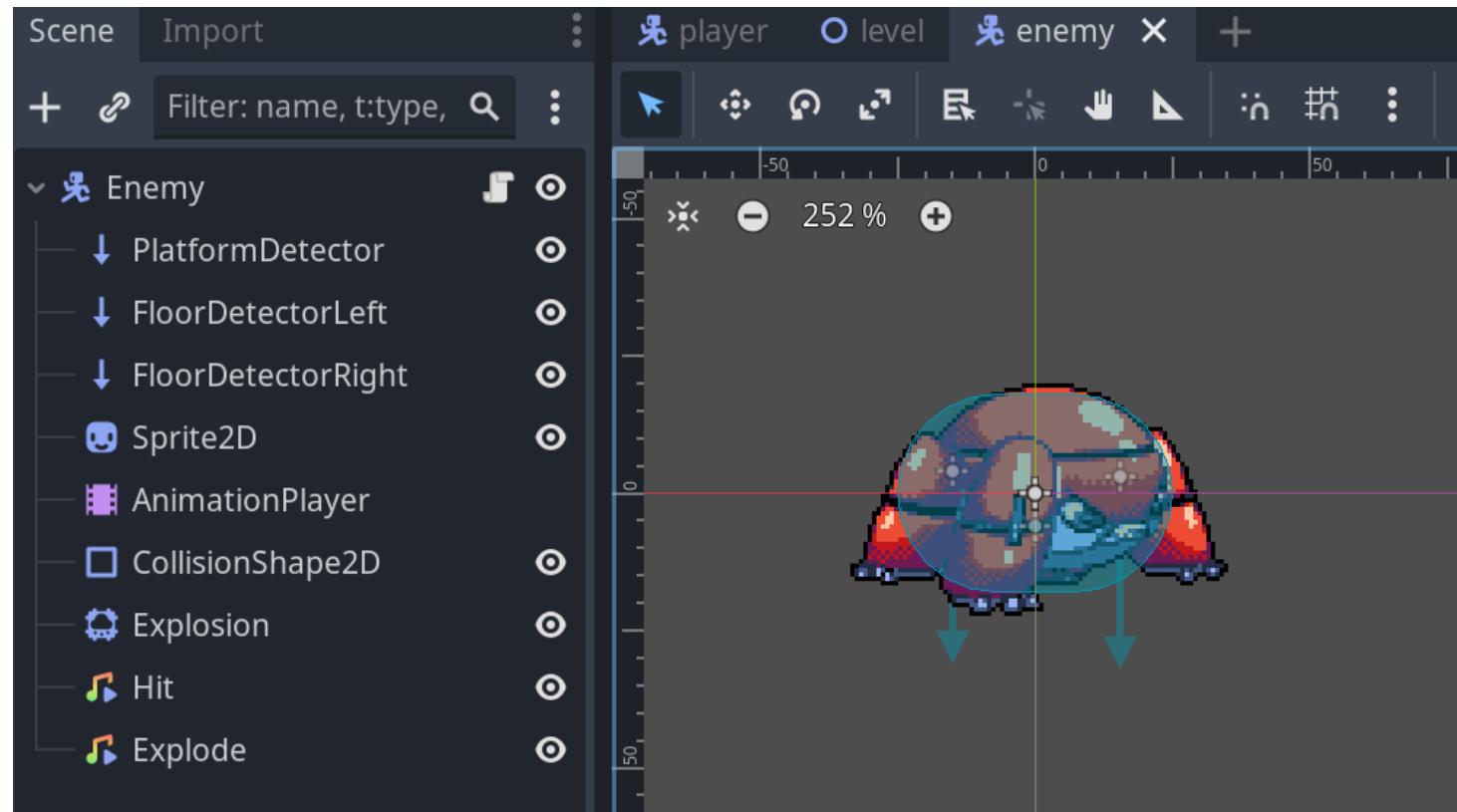
继承带来的魅力

Your class will then appear with its new icon in the editor :



Scene Node System

A scene is composed of one or more nodes.



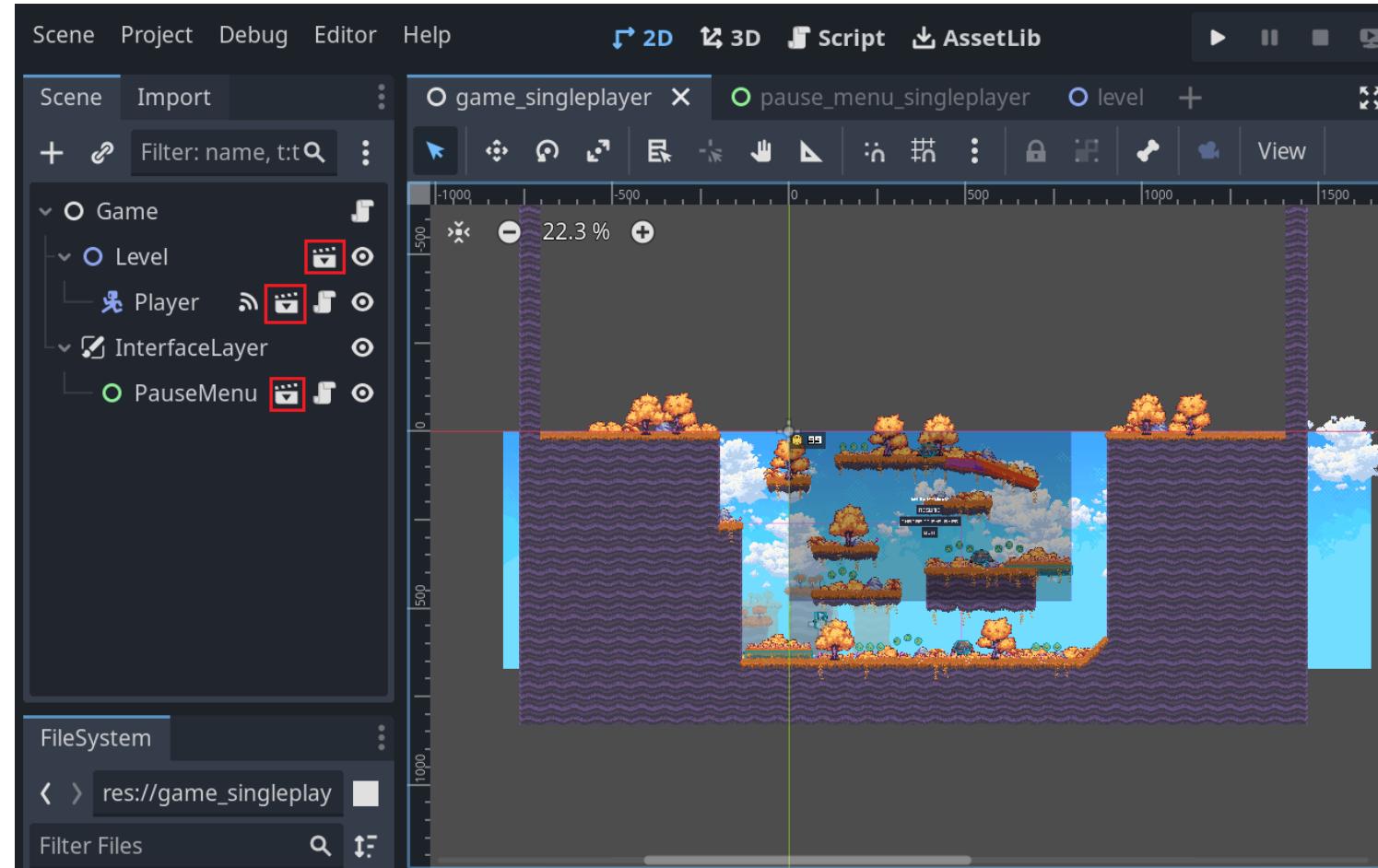
Scene作为抽象

Scenes are **reusable**, **instantiable**, and **inheritable** groups of nodes. A composition of nodes in the form of a tree. Scene has a single root node.

A Godot scene could be a Weapon, a Character, an Item, a Door, a Level, part of a level...

Scene Node System

Godot lets you compose or aggregate scenes.



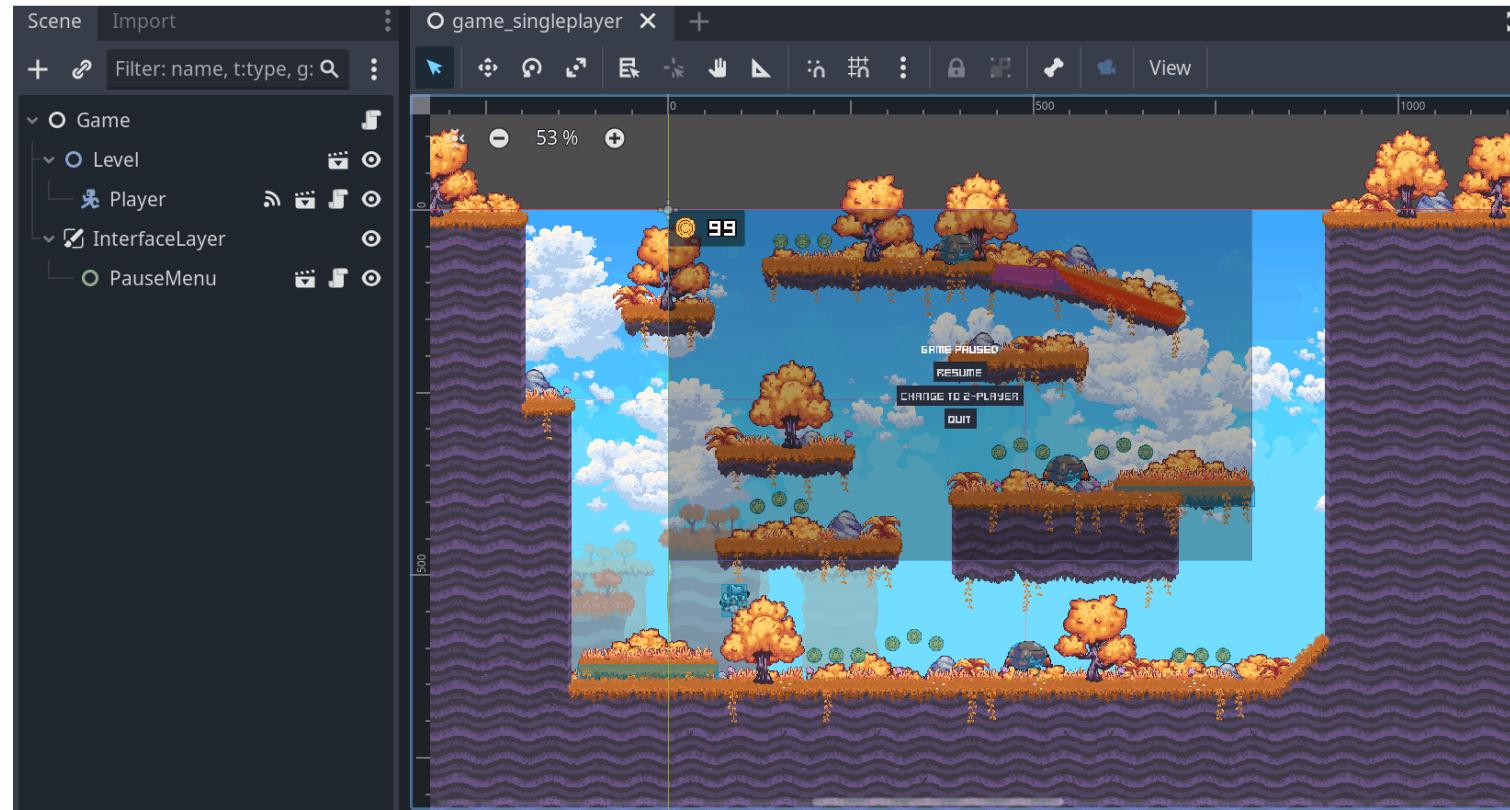
Scene是可组合的

该Demo的主场景由三个子Scene组成

红框标注的icon代表了这是一个Scene
点击可进入对应的子Scene

Scene Node System

Godot lets you compose or aggregate scenes.

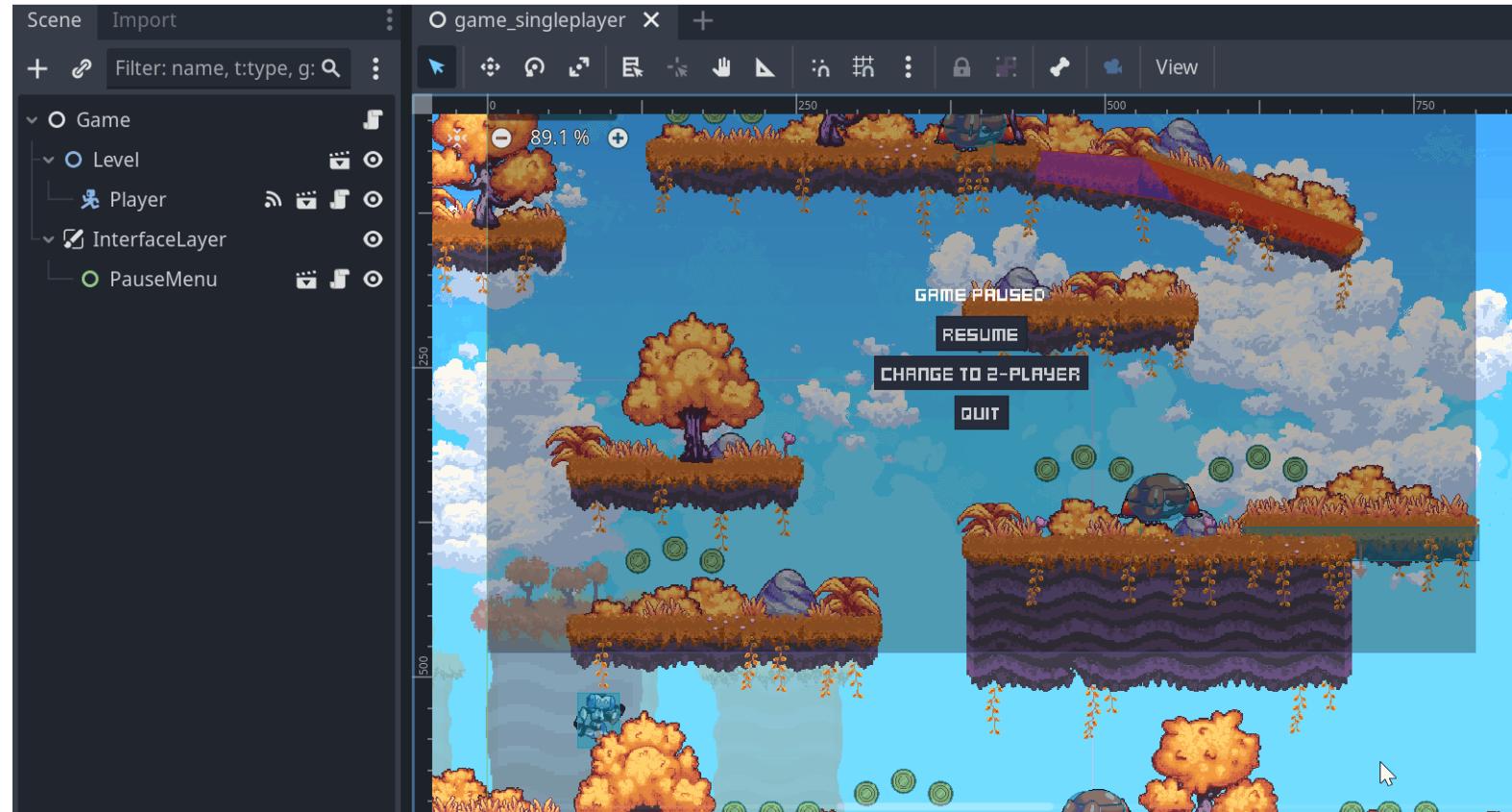


可以使用顶部tab快速来回切换编辑场景

Notice how nodes and scenes look the same in the editor. When you save a tree of nodes as a scene, it then shows as a single node, with its internal structure hidden in the editor.

Scene Node System

Scenes can be nested into other scenes

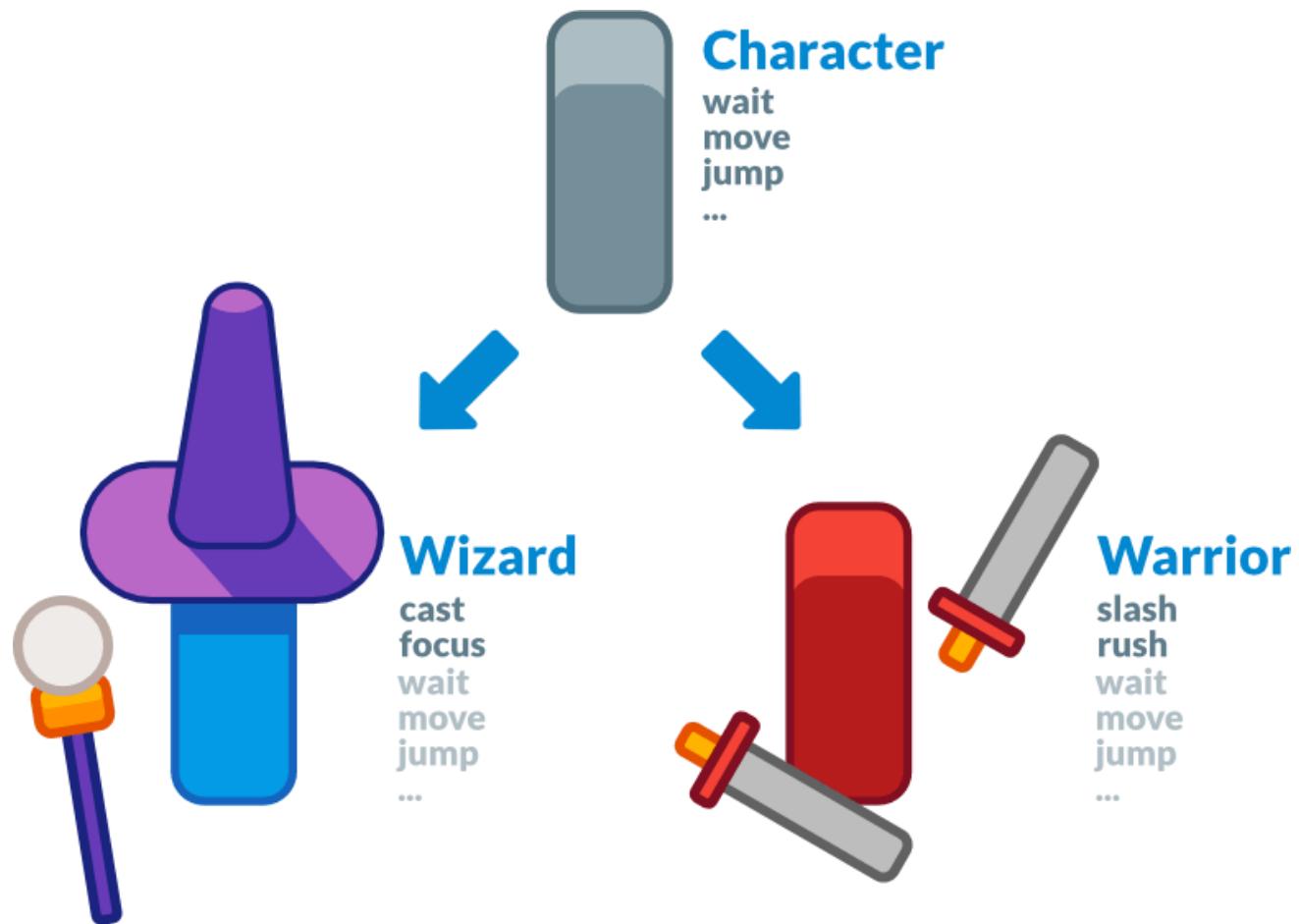


Scene ≈ Unity Prefab + Scene

Scene可以无穷嵌套 (组合)

Scene Node System

On top of that, you can inherit from any scene.



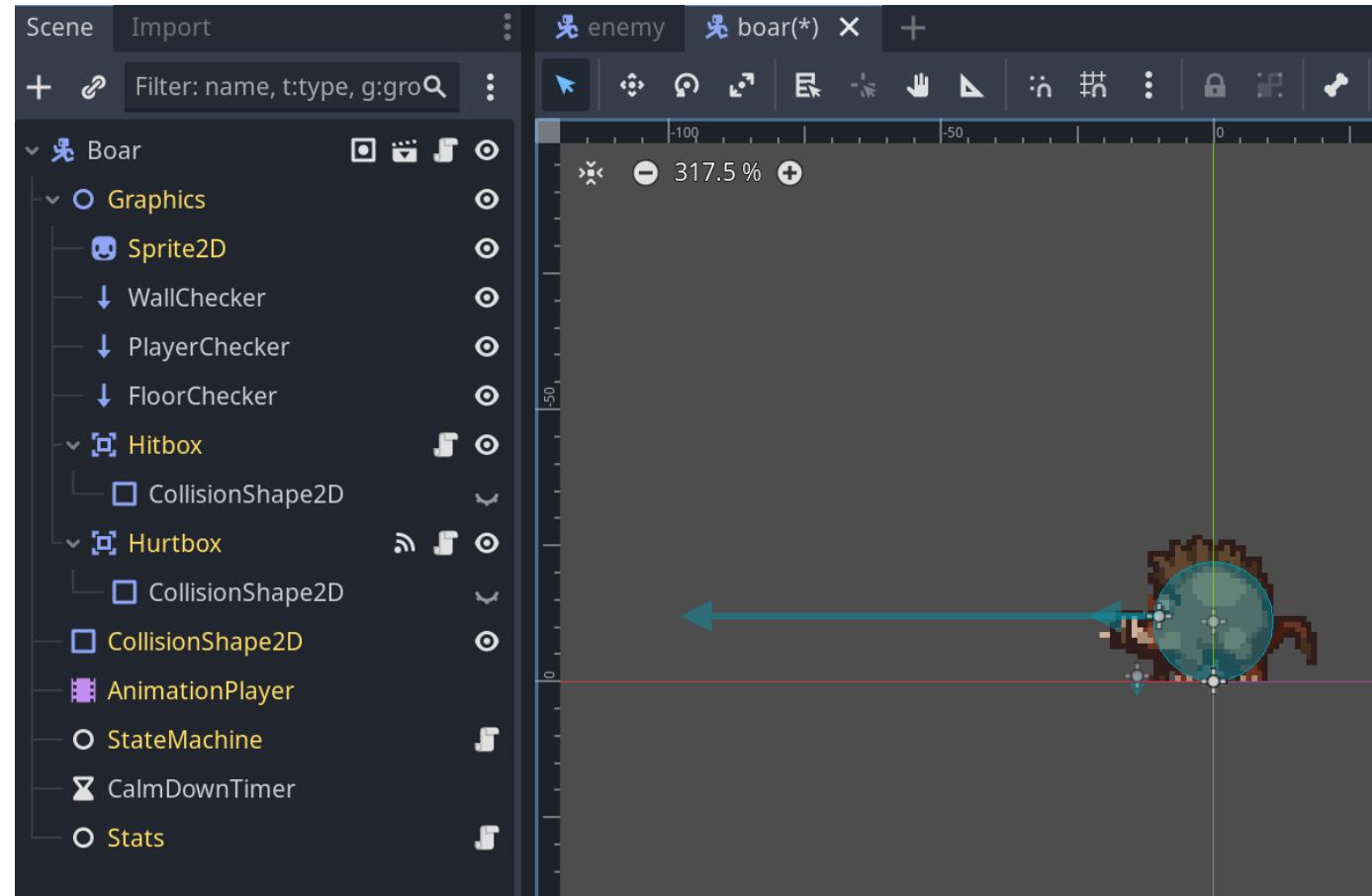
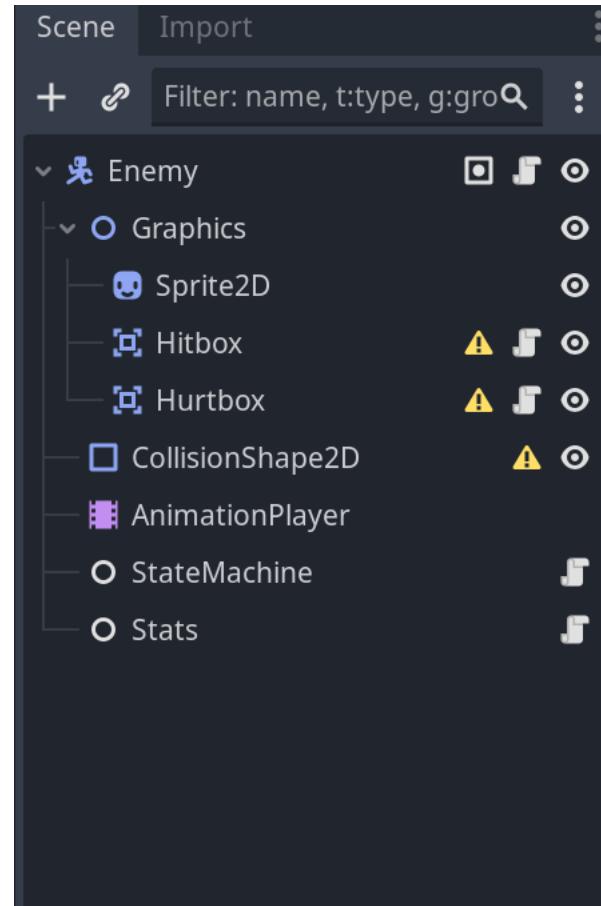
继承

Scene works like a class in pure code, except you're free to design it by using the editor, using only the code, or mixing and matching the two.

Unreal Blueprint class / Unity Prefab variant

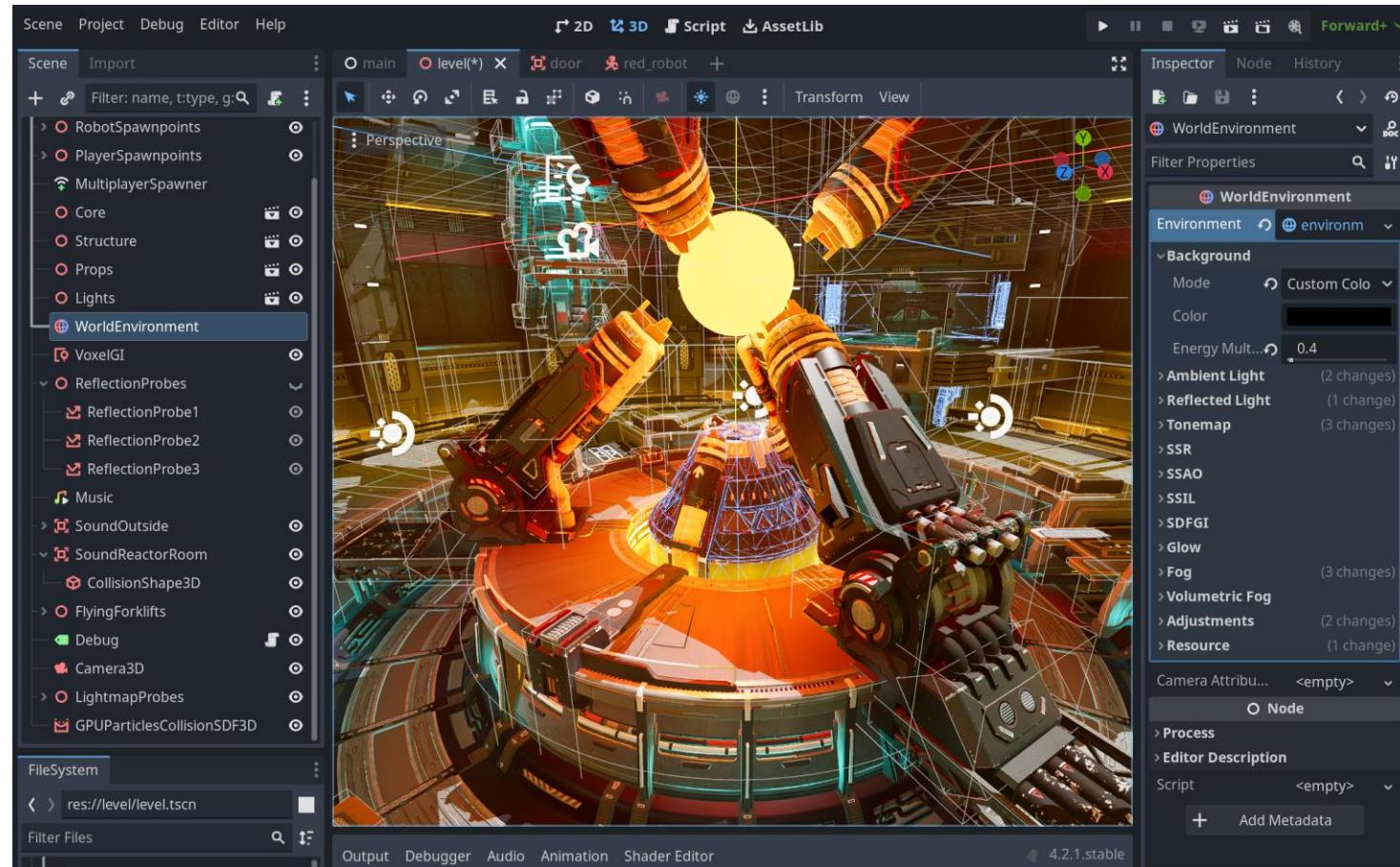
Scene Node System

Demo: 继承Enemy的野猪



Scene Node System

There are no "global settings" per scenes.

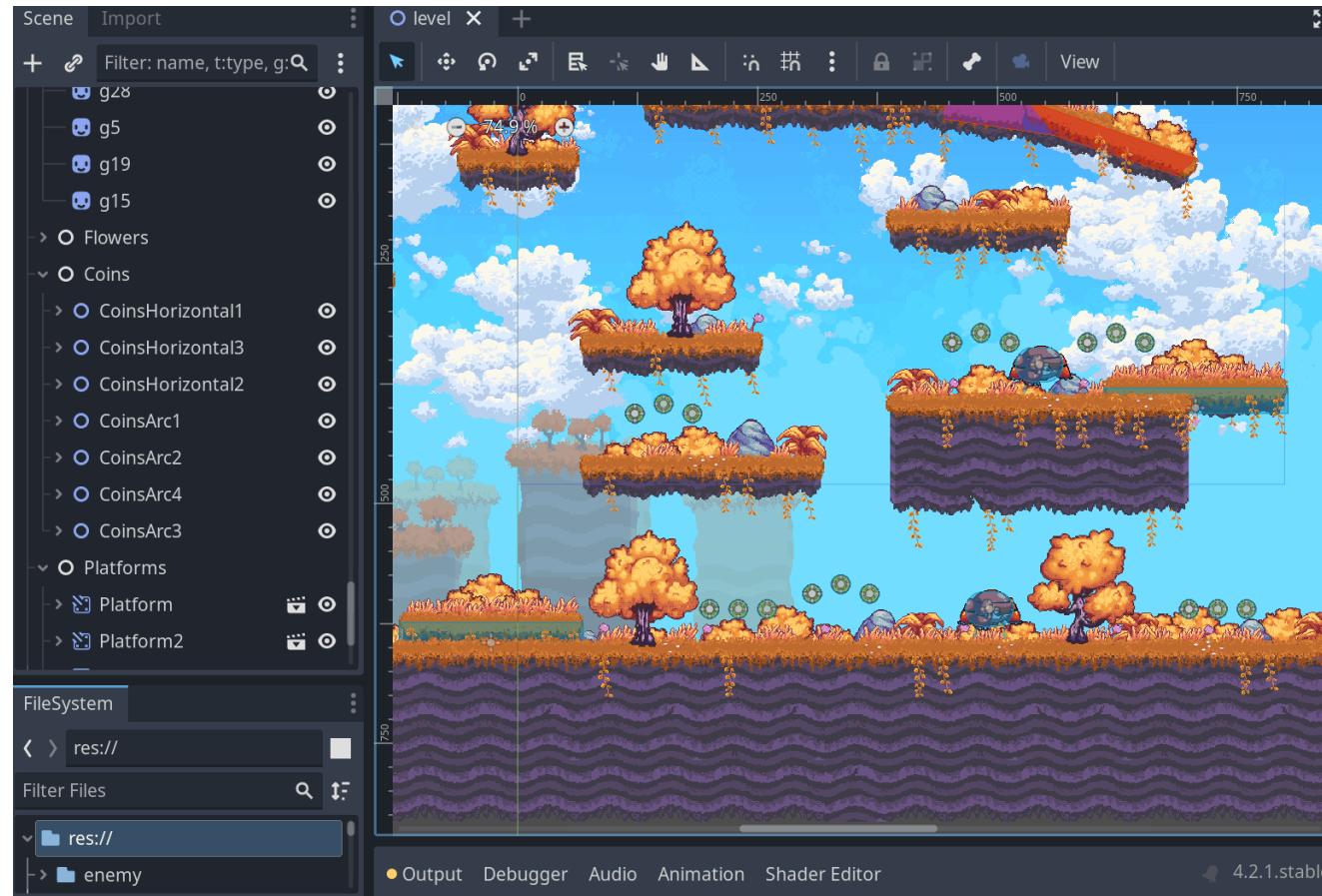


Scene setting 也走Node

Things that in other engines that belong to a scene, such as lightmaps, navigation, environment, etc. In Godot is still just nodes, allowing to mix and match anything as you want..

Scene Node System

Make a new scene is easy

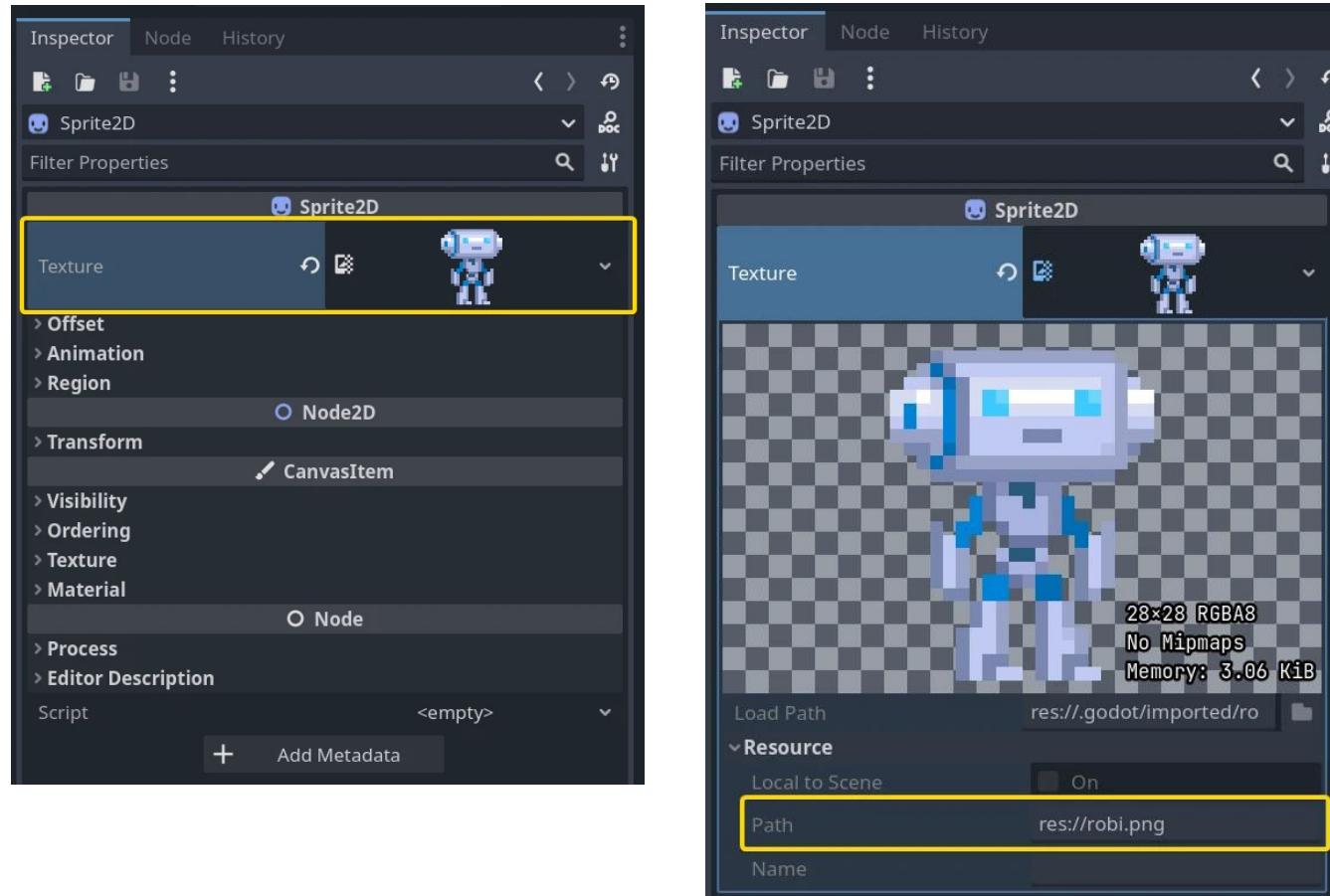


完善的相关功能

- Save branch as scene
- Editable Children
- Make Local
- Load as PlaceHolder

Scene Node System

Resource: data containers



RefCount的Resource

- Anything Godot saves or loads from disk is a resource. Be it a PackedScene (a .tscn or an .scn file), an image, a script...
- Data only exists only once in the memory
- Resources can be saved to disk, or be contained inside another Resource
- Resources are refcounted, freed when no longer used, no GC
- Readable, VCS friendly text format or efficient binary format

Scene Node System

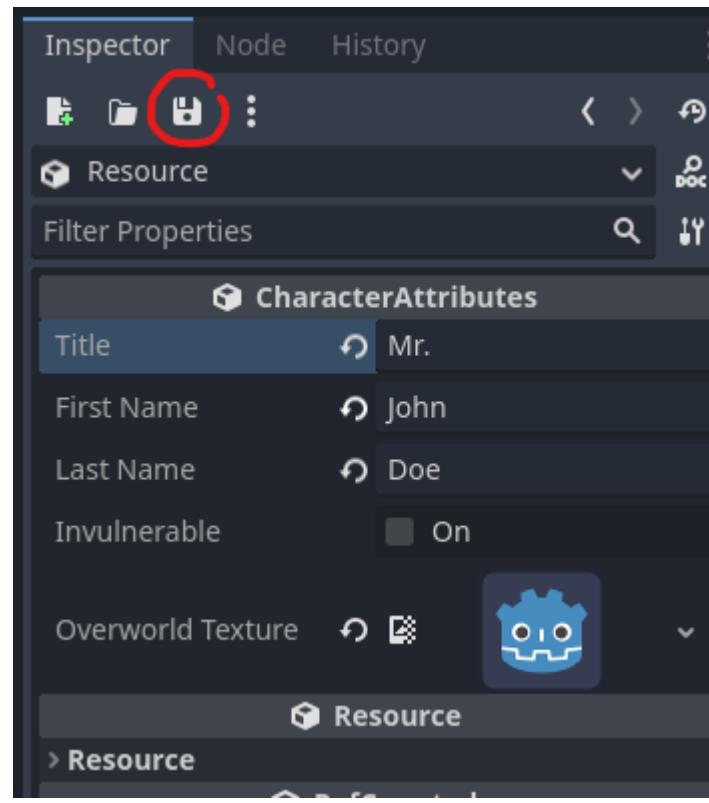
自定义Resource

```
# character_attributes.gd
extends Resource
class_name CharacterAttributes

@export var title: String = ""
@export var first_name: String = "Person"
@export var last_name: String = "McHuman"
@export var invulnerable: bool = false
@export var overworld_texture: Texture

func has_title() -> bool:
    return (title != "")

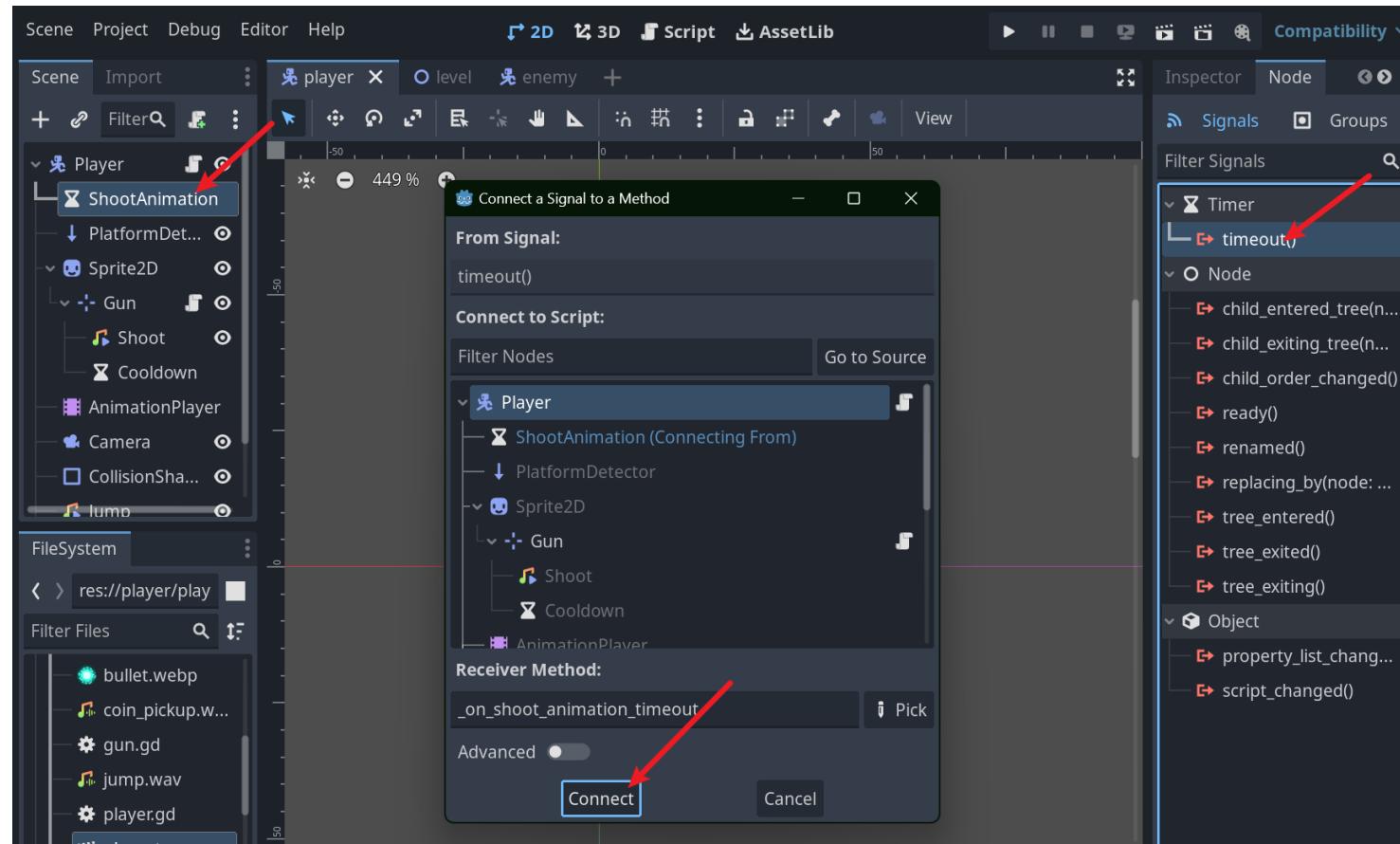
func get_full_name() -> String:
    if (title == ""):
        return "%s %s" % [first_name, last_name]
    return "%s %s %s" % [title, first_name, last_name]
```



≈ ScriptableObjects

Scene Node System

Signal



事件系统

Signals are a delegation mechanism built into Godot that allows one game object to react to a change in another without them referencing one another.

Scene Node System

Godot's Node and Scene system gives you both power and flexibility to create anything.



Juan Linietsky
@reduzio

Advise for Unity users moving to Godot:

Entities: Nodes

Components: Nodes

Scene settings: Nodes

Navigation: Nodes

Lighthmaps: Nodes

Viewport: Nodes

Behaviours: Node+Script

Prefab: Scenes

Scene composing: Scenes

Scriptable objs: Resources

Almost all is a Node, Scene or Resource..



Juan Linietsky @reduzio · Sep 14, 2023

A lot of really complex subsystems in Unity, in Godot they are expressed more naturally and intuitively. As example, threaded loading just open a thread and load resources. Resources are refcounted and unique, etc...

3

18

418

21K

...



Juan Linietsky @reduzio · Sep 14, 2023

Godot uses also a different philosophy to help keep the engine simple, yet powerful. Where other technologies focus on complex, flexible features, Godot does:

- Common use case: Tailored feature
- Corner cases: Best possible level access to build feature yourself as you wish

1

17

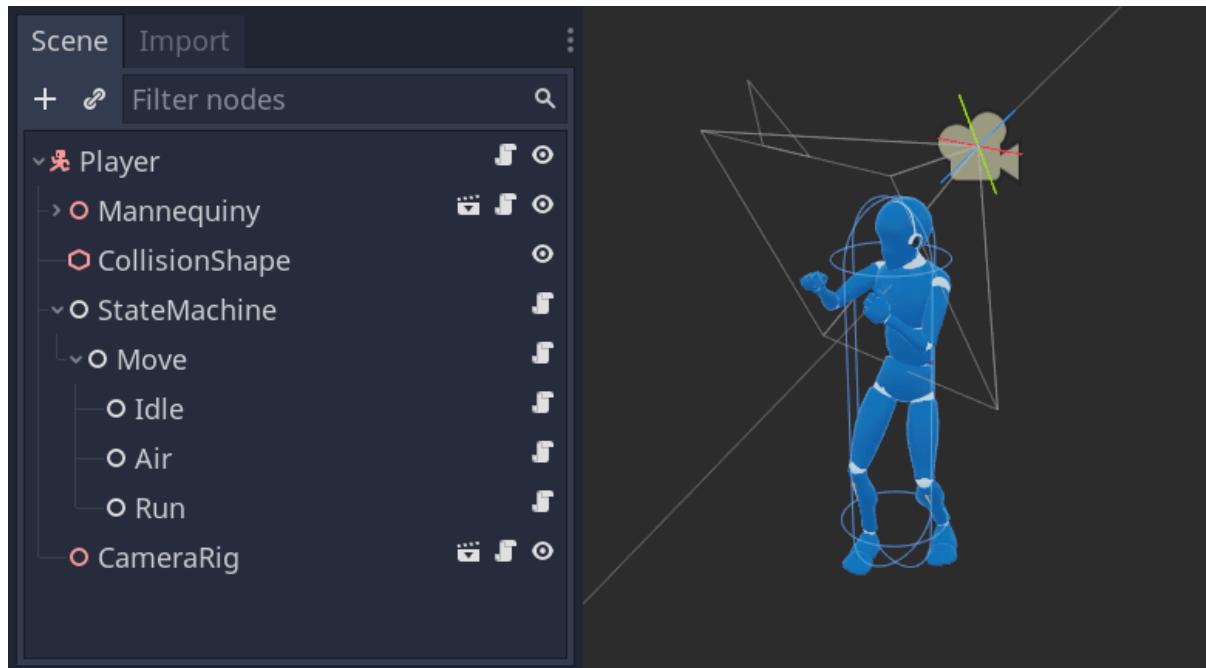
338

23K

...

Scene Node System

Godot的骨架



https://docs.godotengine.org/en/stable/getting_started/step_by_step/nodes_and_scenes.html

1. In General, a game is a tree of nodes that you group together into scenes. Imagine Godot "Scene Tree" as "A tree of components without entities."
2. Everything in Godot is a Node. Games are SceneTrees with a root Viewport node. The Godot Editor is one big fancy EditorNode. Everything is a Node.
3. A scene can be a character, a weapon, a menu in the user interface, a single house, an entire level, or anything you can think of. Godot's scenes are flexible; they fill the role of both prefabs and scenes in some other game engines.

介绍与展示

Scene Node System

Editor

渲染

脚本

Final Thoughts

Editor

初见印象：优化到极致的引擎包体



包体小的原因

1. Godot aim to keep its core feature set small
2. Development rules: Use case based development and consensus is required to move forward
3. Godot are very picky with what goes in, and prefer smaller libraries (single header ones are favorite). It will only bundle something larger if there is no other choice.

https://docs.godotengine.org/en/stable/contributing/development/best_practices_for_engine_contributors.html

包体越小， accessibility越好。

Scene Import

+ Filter: name, t:t:y

Remote Local

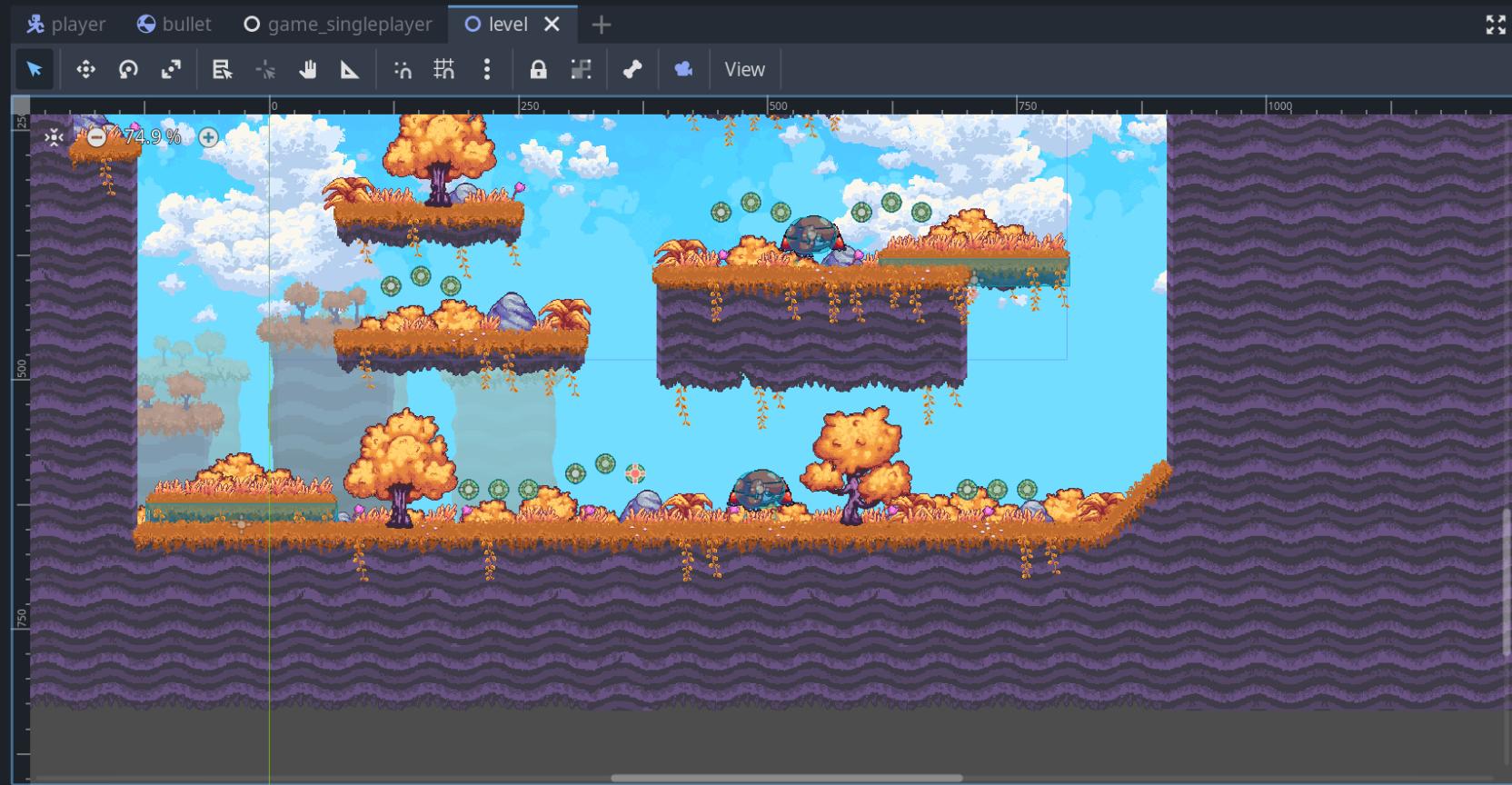
- > O CoinsHorizontal3
- o CoinsHorizontal2
 - > Coin
 - > Coin2
 - > Coin3
- > O CoinsArc1
 - > Coin
 - > Coin2
 - > Coin3
- > O CoinsArc2
 - > Coin

FileSystem

< > res://game_singleplayer.tscn

Filter Files

- gun.gd
- jump.wav
- player.gd
- player.tscn
- robot.webp
- shoot.wav
- default_bus_layout.tres
- game.gd
- game_singleplayer.tscn
- game_splitscreen.gd
- game_splitscreen.tscn
- icon.webp



Godot Engine v4.2.1.stable.official.b09f793f5 - <https://godotengine.org>
OpenGL API 3.3.0 NVIDIA 536.19 - Compatibility - Using Device: NVIDIA - NVIDIA GeForce RTX 3060

Move CanvasItem "T5" to (231, 583)
Scene Undo: Move CanvasItem "T5" to (231, 583)

Filter Messages

Output Debugger Search Results Audio Animation Shader Editor

4.2.1.stable

Inspector Node History

File Filter Properties

Coin3

Area2D

Monitoring On
Monitorable On
Priority 0

> Gravity
> Linear Damp
> Angular Damp
> Audio Bus
 CollisionObject2D
Disable Mode Remove

> Collision

> Input

 Node2D

> Transform

Position x 60 px

y 0 px

Rotation 0 °

Scale x 1

y 1

Skew 0 °

CanvasItem

> Visibility

> Ordering

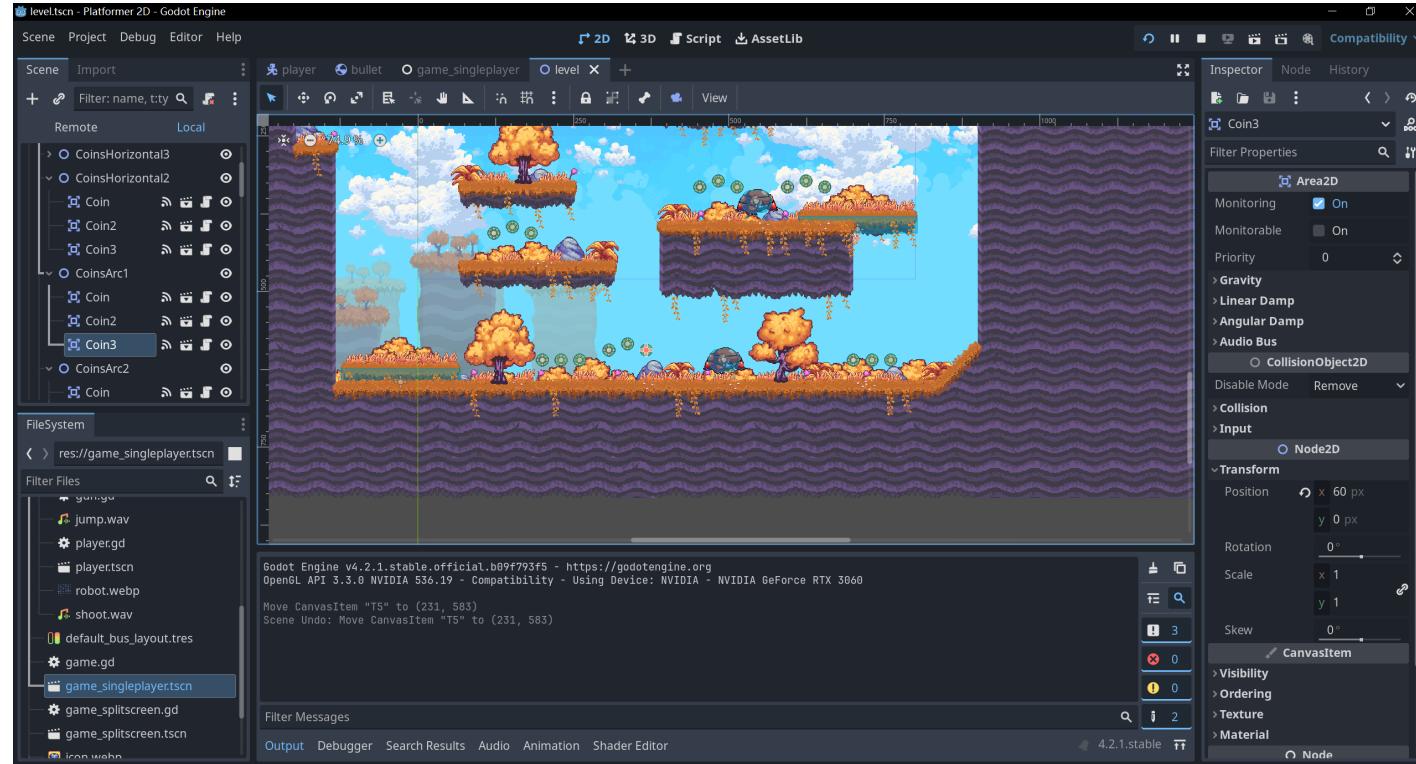
> Texture

> Material

Node

Editor

功能

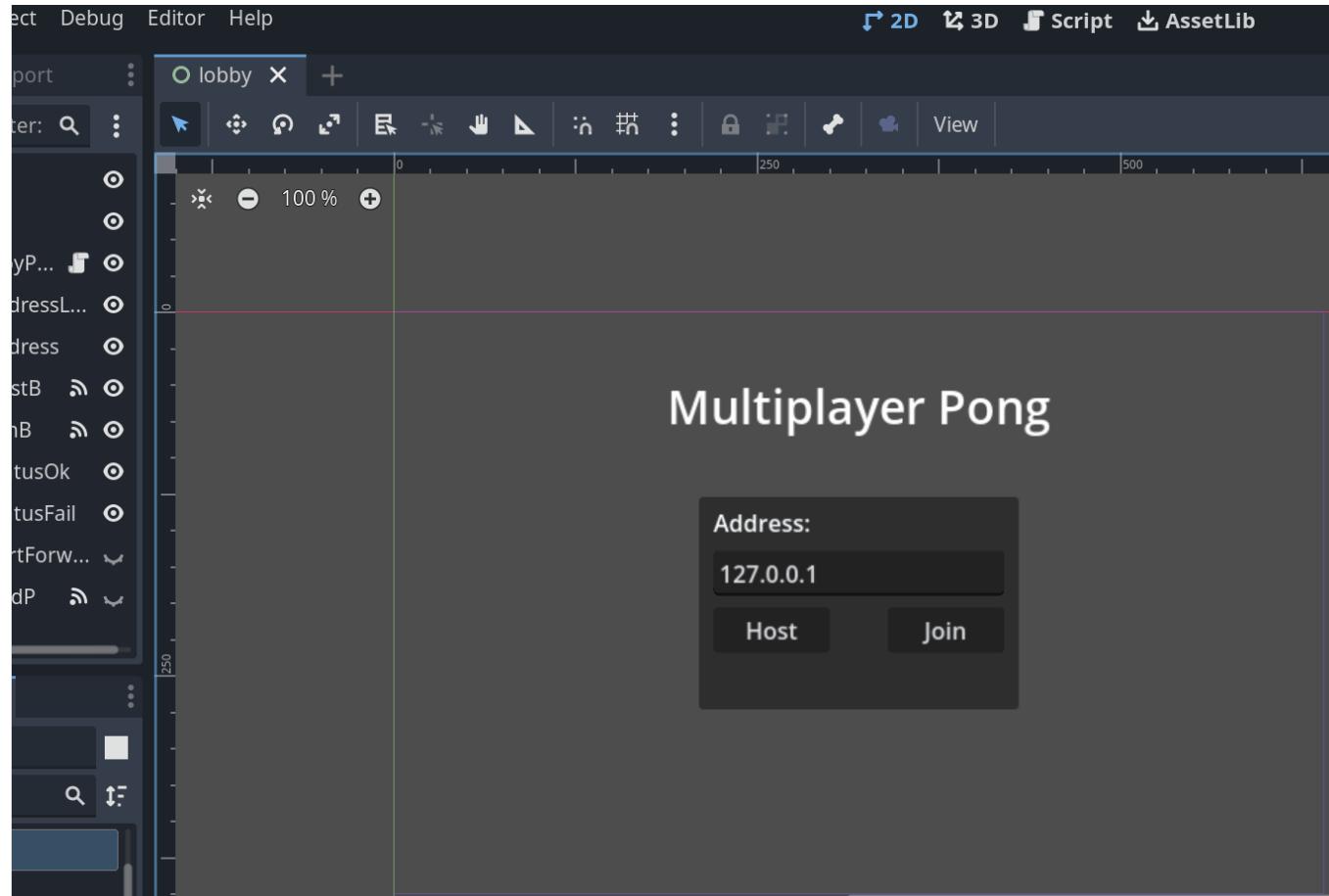


一些特点

1. 有专门的2D和3D编辑界面，2D编辑器的基础单位是像素 (对于3D游戏，2D编辑器用来做UI)
2. 完善的2D工具链，如 TileSet TileMap编辑器
3. 内置的脚本编辑器来写gdscript
4. 只有编辑界面，没有Game窗口
5. Godot编辑器也是Godot游戏，所以Godot编辑器也可以在安卓、web上使用

Editor

editor和runtime进程分离：天生支持游戏多开



只需要一个exe

启动Editor编辑该project:

```
godot -e --path path_to_your_project
```

运行该project:

```
godot --path path_to_your_project
```

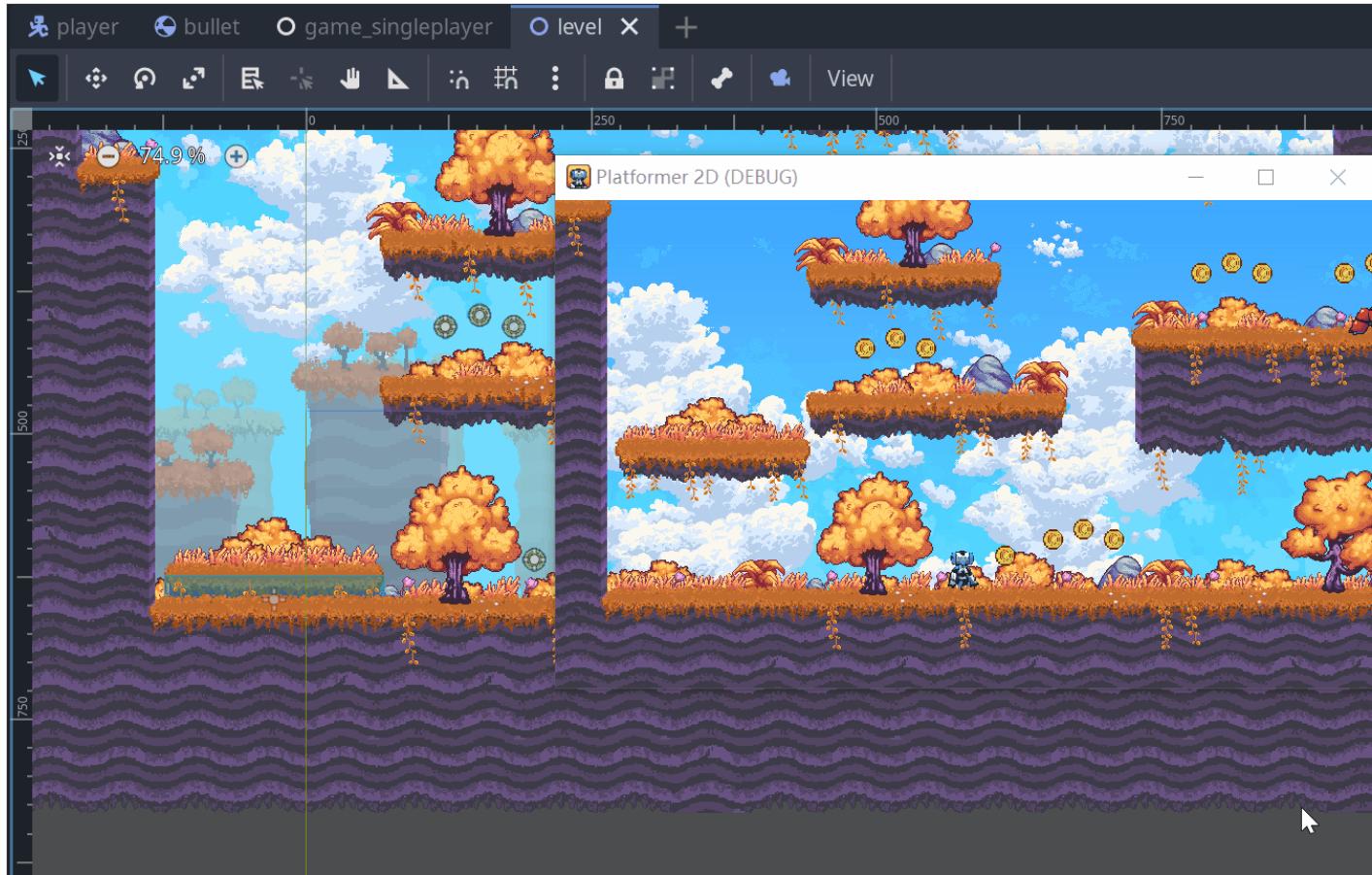
跑测试、CI 等:

```
godot -headless --path path_to_your_project
```

https://github.com/godotengine/godot-demo-projects/tree/master/networking/multiplayer_pong

Editor

Scene 支持 live editing



进程分离的优势

- Running the project and closing it is fast
- Changes done to the editor take effect immediately in the game and are not lost
- The editor is more stable because the game runs in a separate process

劣势

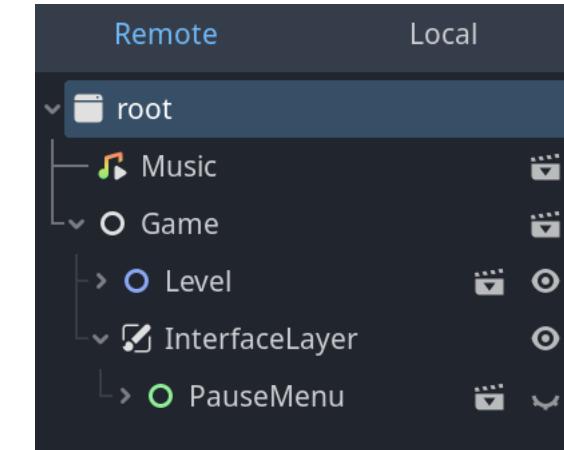
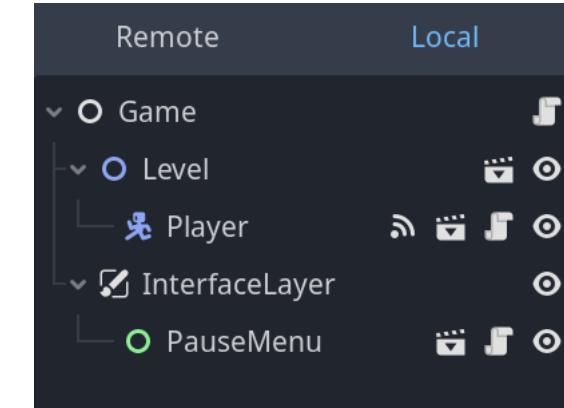
- The running game can not be conveniently explored from different angle

Solution

- Live editing
- Remote and local scene tree
- Project camera override

Editor

Remote 和 Local SceneTree



Editor

资源处理



名称	修改日期	类型	大小
enemy.gd	2024/3/26 10:01	GD 文件	2 KB
enemy.tscn	2024/4/16 0:23	TSCN 文件	9 KB
enemy.webp	2024/3/26 10:01	WEBP 文件	10 KB
enemy.webp.import	2024/4/14 22:26	IMPORT 文件	1 KB
explode.wav	2024/3/26 10:01	WAV 文件	75 KB
explode.wav.import	2024/4/14 22:26	IMPORT 文件	1 KB
hit.wav	2024/3/26 10:01	WAV 文件	71 KB
hit.wav.import	2024/4/14 22:26	IMPORT 文件	1 KB

只有非native的资源才会有导入选项 (import options)，Godot自己的格式 (`.tscn`、`.scn`、`.tres`、`.res`) 没有导入选项
.import类似unity的meta，导入处理完的文件在 .godot/imported 下

Editor

老大难的UID问题

```
1 @uid("uid//3us27dknw8sh1") # auto generated, do not modify
2 class_name Game extends Node
3
4 @onready var _pause_menu := $InterfaceLayer/PauseMenu as PauseMenu
5
6 func _unhandled_input(event: InputEvent) -> void:
7   if event.is_action_pressed(&"toggle_fullscreen"):
8     var mode := DisplayServer.window_get_mode()
9     if mode == DisplayServer.WINDOW_MODE_FULLSCREEN or \
```

Crash when trying to edit a 3D model with an external bone map assigned that was renamed #90014

[Open](#) LunaCapra opened this issue 3 weeks ago · 0 comments

The screenshot shows a GitHub issue page for a Godot engine bug. At the top is a snippet of Godot script:

```
1 @uid("uid//3us27dknw8sh1") # auto generated, do not modify
2 class_name Game extends Node
3
4 @onready var _pause_menu := $InterfaceLayer/PauseMenu as PauseMenu
5
6 func _unhandled_input(event: InputEvent) -> void:
7   if event.is_action_pressed(&"toggle_fullscreen"):
8     var mode := DisplayServer.window_get_mode()
9     if mode == DisplayServer.WINDOW_MODE_FULLSCREEN or \
```

Below the script is a detailed view of the GitHub issue card:

- Tested versions**: v4.3.dev5.official [[b9f70e9](#)], v4.2.1.stable.mono.official [[b09f793](#)]
- System information**: Windows 10.0.22631 - Vulkan (Forward+) - dedicated NVIDIA GeForce RTX 3070 (NVIDIA; 31.0.15.4633) - AMD Ryzen 9 5900X 12-Core Processor (24 Threads)
- Issue description**: (empty)
- Assignees**: No one assigned
- Labels**: bug, crash, topic:animation
- Projects**: None yet
- Milestone**: No milestone

因为：

1. 脚本是resource，所以没有.import文件
2. 脚本本身只有文本，不像自定其他格式可以加uid

所以：想索引到脚本只能靠路径+文件名。

想加uid但是太丑了，被回退

<https://github.com/godotengine/godot/pull/67132>

更严重的会引起crash

<https://github.com/godotengine/godot/issues/90014>

Editor

场景的格式: tscn

```
[gd_scene load_steps=4 format=3 uid="uid://cecaux1sm7mo0"]

[sub_resource type="SphereShape3D" id="SphereShape3D_tj6p1"]

[sub_resource type="SphereMesh" id="SphereMesh_4w3ye"]

[sub_resource type="StandardMaterial3D" id="StandardMaterial3D_k54se"]
albedo_color = Color(1, 0.639216, 0.309804, 1)

[node name="Ball" type="RigidBody3D"]

[node name="CollisionShape3D" type="CollisionShape3D" parent=".."]
shape = SubResource("SphereShape3D_tj6p1")

[node name="MeshInstance3D" type="MeshInstance3D" parent=".."]
mesh = SubResource("SphereMesh_4w3ye")
surface_material_override/0 = SubResource("StandardMaterial3D_k54se")

[node name="OmniLight3D" type="OmniLight3D" parent=".."]
light_color = Color(1, 0.698039, 0.321569, 1)
omni_range = 10.0

[node name="Camera3D" type="Camera3D" parent=".."]
transform = Transform3D(1, 0, 0, 0, 0.939693, 0.34202, 0, -0.34202, 0.939693, 0, 1, 3)
```

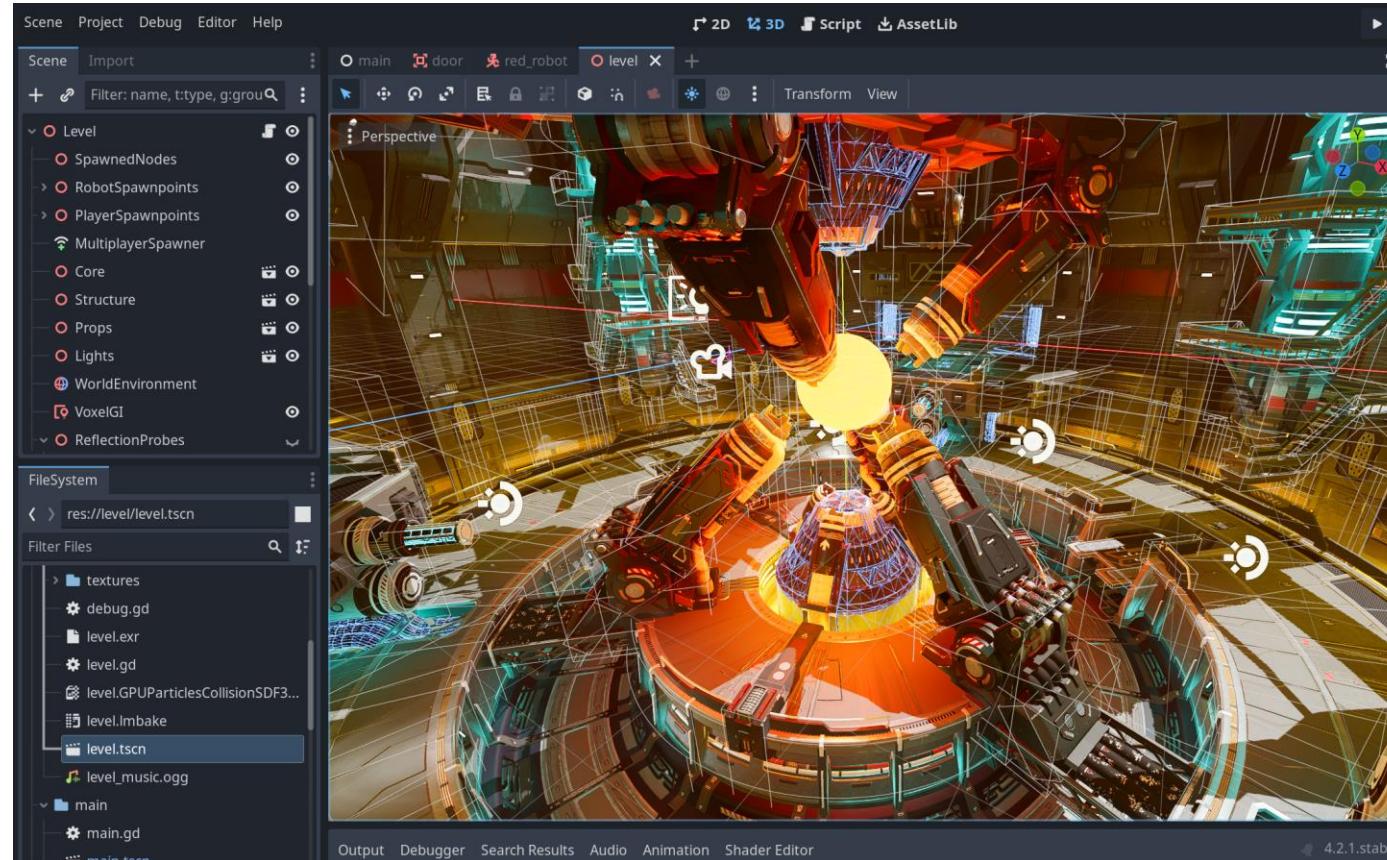
Git友好

“Holy shit. Much better than the Unity YAML stuff which made us contemplate if performance was actually useful (it is not, fight me). Gone are the days of hellish prefab merges. Still, if there are a lot of changes in a scene you can get into a bit of trouble. Just a bit though.”

<https://caseyyano.com/on-evaluating-godot-b35ea86e8cf4>

Editor

极致的打包速度



项目文件夹大小：466M
全量导出：7s
改一行脚本后增量导出：2s
改一个level物体后增量导出：2s

名称	类型	大小
test.exe	应用程序	67,982 KB
test.pck	PCK 文件	462,438 KB

<https://github.com/godotengine/tps-demo>

Editor

Why

```
.  
├── .godot  
│   ├── exported  
│   │   └── 133200997  
│   │       └── export-78c237d4bfdb4e1d02e0b5f38ddfd8bd-scene.scn  
│   ├── global_script_class_cache.cfg  
│   ├── imported  
│   │   ├── map_data.png-ce840618f399a990343bfc7298195a13.ctex  
│   │   ├── music.ogg-fa883da45ae49695a3d022f64e60aee2.oggvorbisstr  
│   │   └── sprite.png-7958af25f91bb9dbae43f35388f8e840.ctex  
│   └── uid_cache.bin  
└── client  
    ├── music.ogg.import  
    └── sprite.png.import  
└── server  
    └── map_data.png.import  
└── test  
    └── scene.gd  
└── unused  
    └── development_test.gd  
└── project.binary  
└── scene.gd  
└── scene.tscn.remap
```

简单的PCK格式

1. 引擎部分：Export template
2. 脚本：直接原样塞进pck文件中 (4.3开始支持token化并压缩)
3. 资源：直接把处理过的资源转二进制然后丢进pck打包(PCK文件开头记录各个资源的路径、offset和size，后续是blob data, 默认不压缩)

介绍与展示

Scene Node System

Editor

渲染

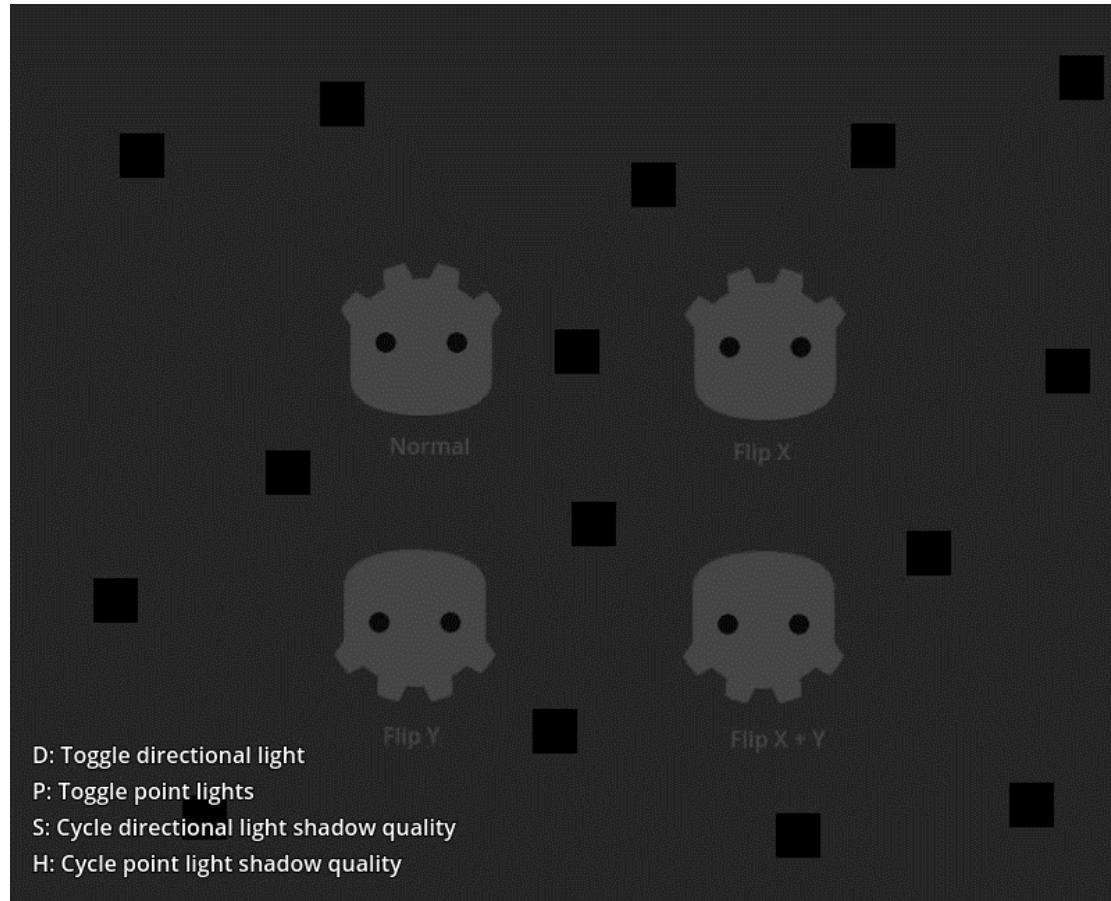
4

脚本

Final Thoughts

渲染

2D渲染



丰富的2D渲染支持

- CanvasModulate (to darken the rest of the scene)
- PointLight2D (for omnidirectional or spot lights)
- DirectionalLight2D (for sunlight or moonlight)
- LightOccluder2D (for light shadow casters)
- GPUParticles2D and CPUParticles2D
- 2D MSAA Support
- Other 2D nodes that receive lighting, such as Sprite2D or TileMap.



GODOT
Game engine

渲染

3种预设的渲染方法

Forward plus

A forward renderer that uses a clustered approach to lighting (vk & dx12).
This approach can greatly speed up rendering performance on desktop hardware.

Mobile

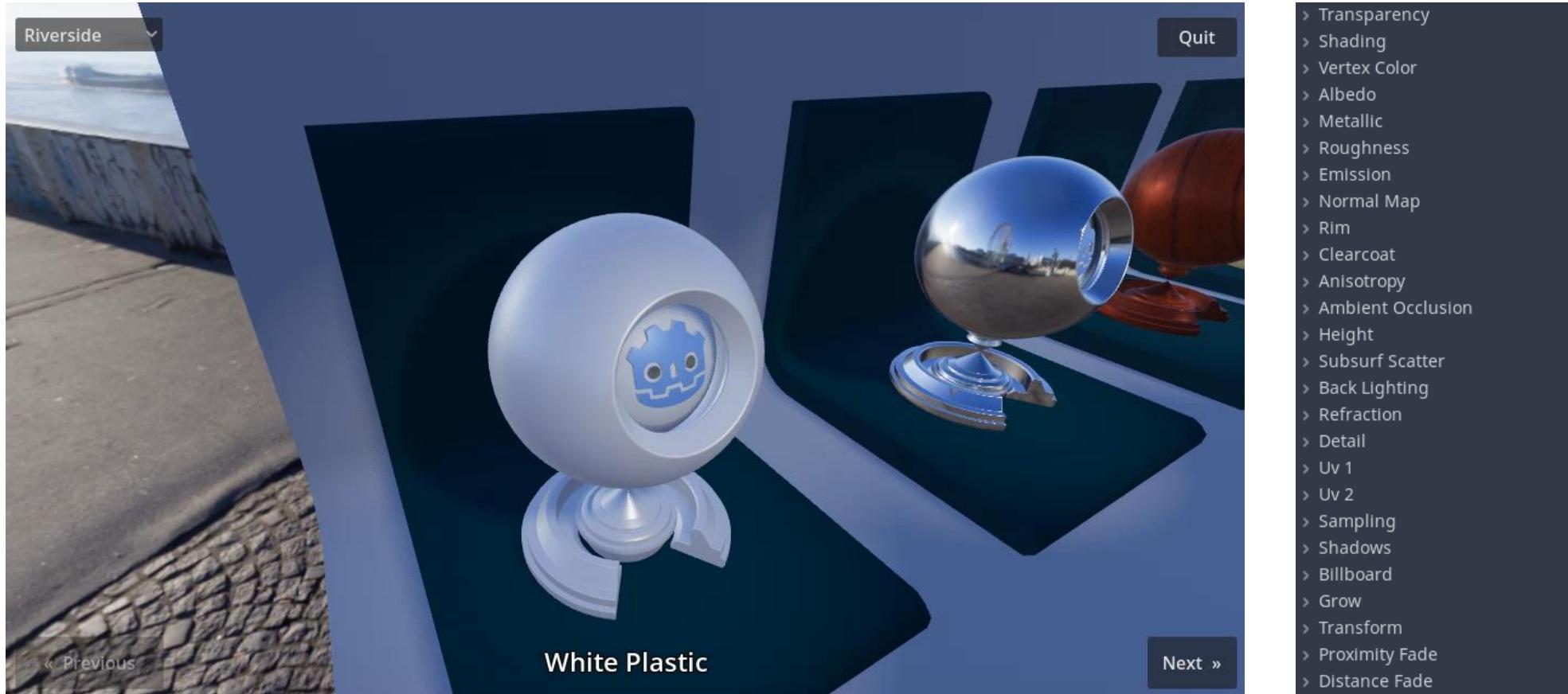
A forward renderer that uses a traditional single-pass approach to lighting (vk).
This rendering method is optimized to perform well on mobile GPUs.

Compatibility

A traditional (non-clustered) forward renderer (opengl 3).
For old GPUs that don't have Vulkan support and optimized for older and lower-end devices.

渲染

材质：StandardMaterial3D

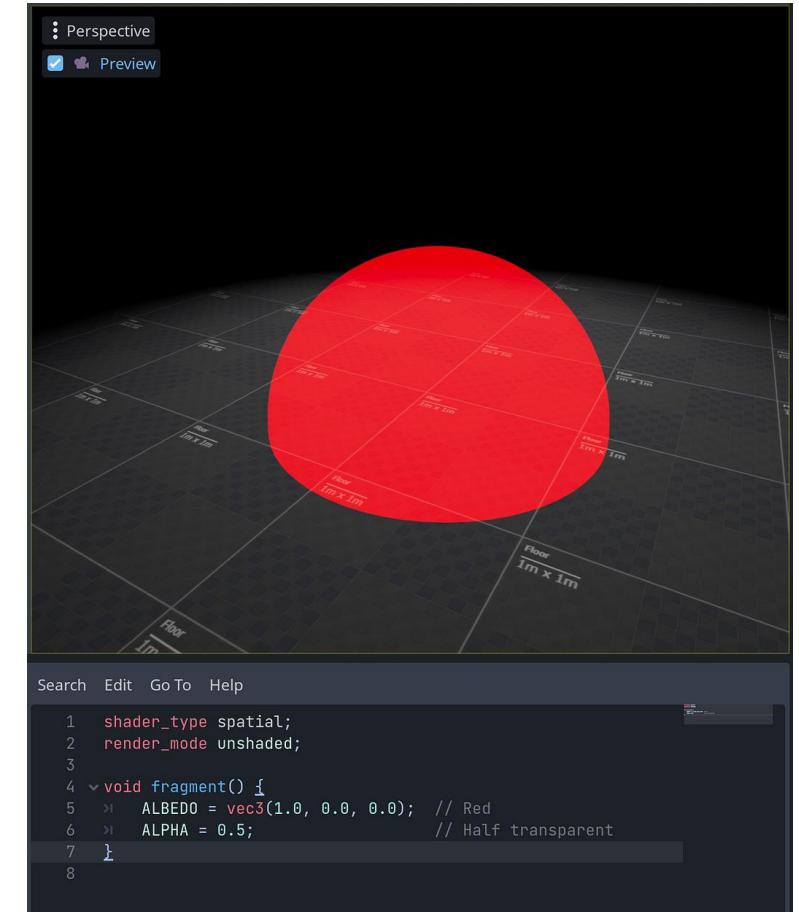
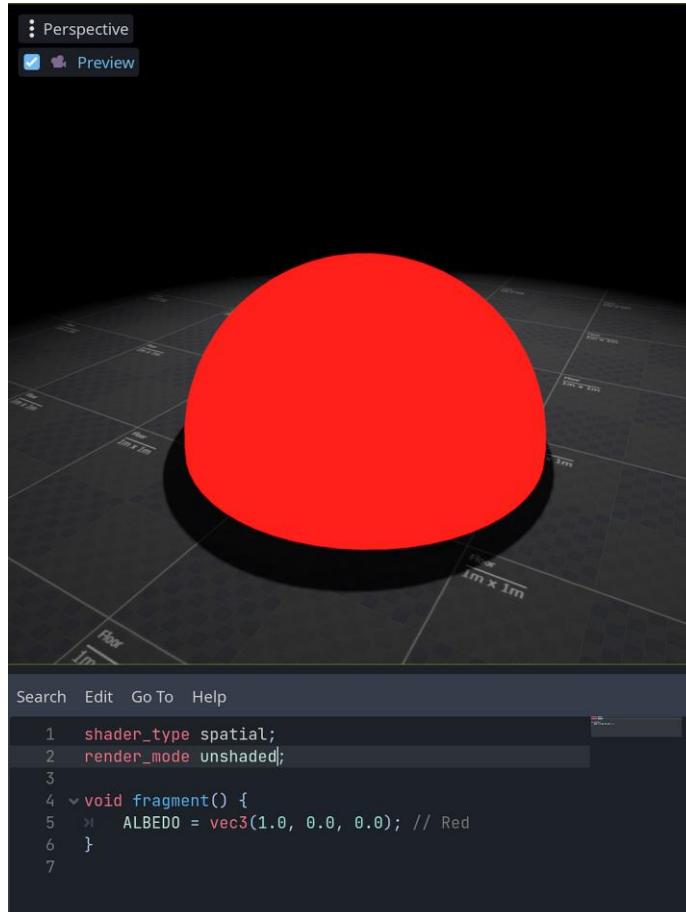
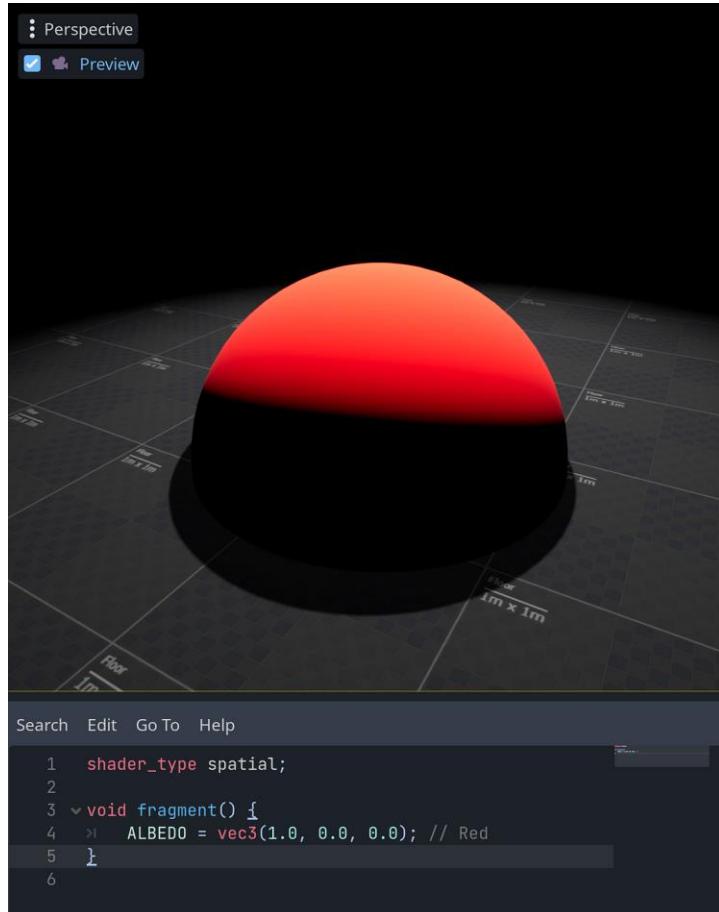


A PBR material Can be converted to shader code if additional functionality is needed.

https://github.com/godotengine/godot-demo-projects/tree/master/3d/material_testers

渲染

材质：ShaderMaterial



渲染

Shader

```
void fragment() {  
    METALLIC = 0.0;  
    ROUGHNESS = 0.01;  
    ALBEDO = vec3(0.1, 0.3, 0.5);  
}
```

Fragment Shader

The standard use of the fragment function is to set up material properties used to calculate lighting.

```
void light() {  
    DIFFUSE_LIGHT += clamp(dot(NORMAL, LIGHT), 0.0, 1.0) * ATTENUATION * LIGHT_COLOR;  
}
```

Light Shader

Light() exists as a function called inside the fragment(). The light function is called for every light in every pixel.

渲染

How shader works

```
386     float roughness = 1.0;  
387  
388     mat4 modelview = scene_data.view_matrix * model_matrix;  
389     mat3 modelview_normal = mat3(scene_data.view_matrix) * model_normal_matrix;  
390     mat4 read_view_matrix = scene_data.view_matrix;  
391     vec2 read_viewport_size = scene_data.viewport_size;  
392  
393     {  
394 #CODE : VERTEX      Juan Linietsky, 3 years ago • Refactor GLSL shader compilation ...  
395     }  
396
```

Core Shader

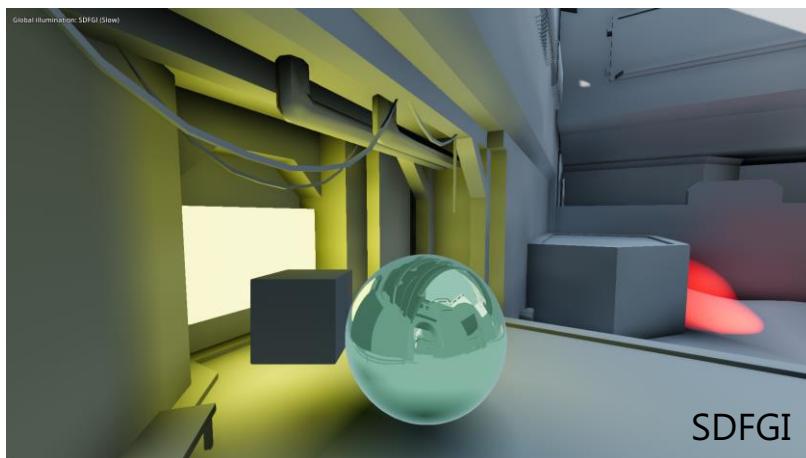
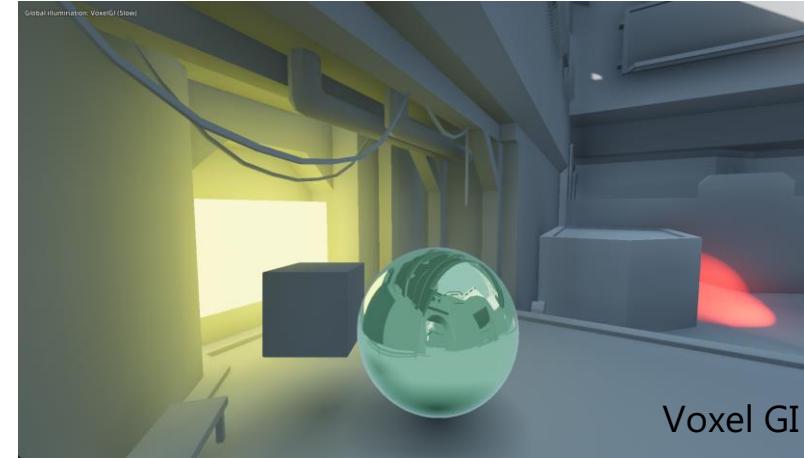
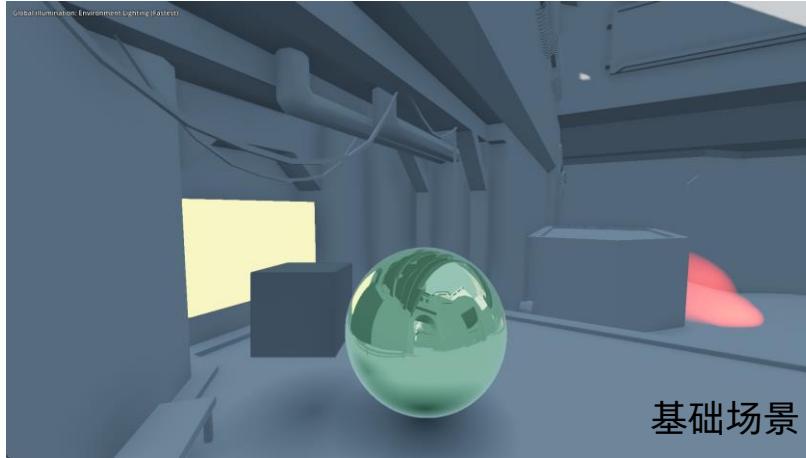
向Core shader (GLSL) 模板里填充经过预处理的用户shader代码，每个渲染方法都有自己的Core shader

- 让用户少写模板代码
- 切换渲染方法时候尽量无痛

```
991     vec3 normal = N;  
992     vec3 light = L;  
993     vec3 view = V;  
994  
995     /* clang-format off */  
996  
997 #CODE : LIGHT      Clay John, 24 months ago  
998  
999     /* clang-format on */  
1000
```

渲染

全局光照



开箱即用的GI

- Voxel GI
- SDFGI
- LightmapGI
- SSIL

https://github.com/godotengine/godot-demo-projects/tree/master/3d/global_illumination

渲染

无穷无尽的性能优化

Announcing a collaboration with Google and The Forge

By:  Clay John

14 December 2023

News

We are excited to announce that we have partnered with [Google](#) and [The Forge](#) to bring some helpful performance optimizations to our Vulkan mobile backend. This will primarily benefit users targeting Vulkan-capable mobile devices.

Google is committed to enhancing the Android gaming ecosystem by ensuring that Vulkan is well supported across many games, game engines, and devices. Lucky for us, this means that they have decided to help us ensure that our Vulkan mobile renderer is as efficient as possible.

得道多助

Google 和 The Forge 来帮忙优化vulkan的性能

<https://github.com/godotengine/godot/pull/90284>

<https://godotengine.org/article/collaboration-with-google-forge-2023/>

<https://godotengine.org/article/update-on-google-forge-2024/>

Pixel 7

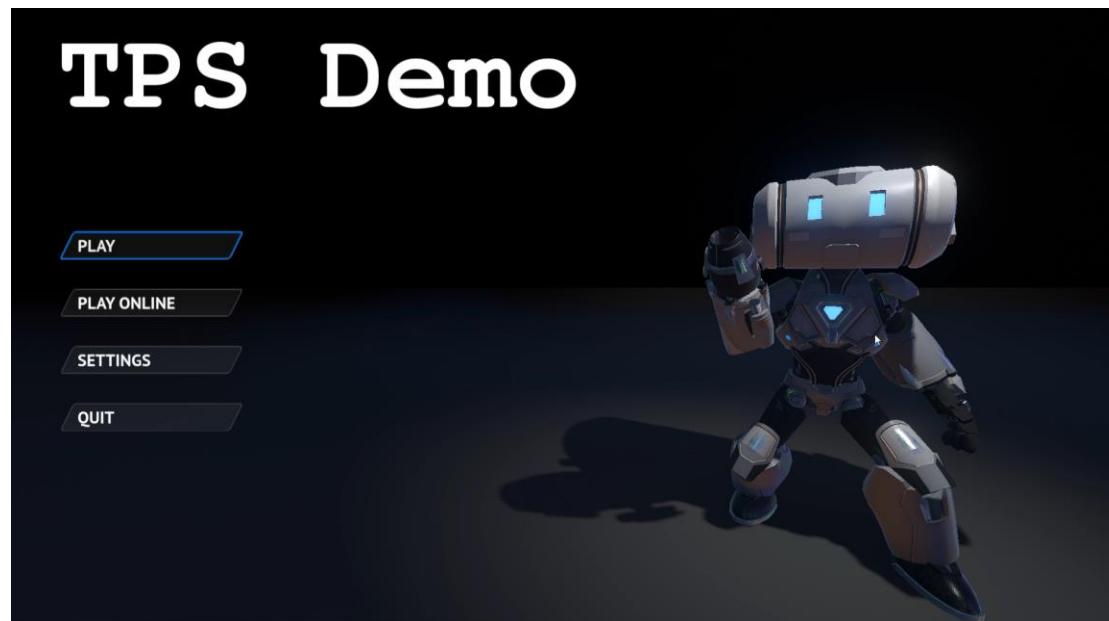
	Sun Temple OPT	Sun Temple (UNOPT)	Third Person Shooter (OPT)	Third Person Shooter (UNOPT)
Frame time	26.1 ms	29.1 ms	36.4 ms	37.1 ms
GPU frame time	25.81	28.60 ms	35.03 ms	35.81 ms
CPU frame time	4.85 ms	5.01 ms	3.67 ms	3.53 ms
Vk Memory	2487.53 MB	2488.47 MB	554.64 MB	555.93 MB

渲染

More

1. Acyclic Command Graph for Rendering : <https://github.com/godotengine/godot/pull/84976>
2. HDDAGI : <https://github.com/godotengine/godot/pull/86267>
3. Ubershaders and pipeline pre-compilation : <https://github.com/godotengine/godot/pull/90400>
4. More here : <https://godotengine.org/article/rendering-priorities-january-2024/>

例如Uber Shader



介绍与展示

Scene Node System

Editor

渲染

脚本

Final Thoughts

脚本

一目了然

```
class_name my_node
extends Node

var count : int

func _ready():
    count = 0

func _process(delta):
    count = count + 1
    print(count)
```



```
using UnityEngine;
using System.Collections;

public class MyComponent : MonoBehaviour
{
    int Count;

    // Use this for initialization.
    void Start ()
    {
        Count = 0;
    }

    // Update is called once per frame.
    void Update ()
    {
        Count = Count + 1;
        Debug.Log(Count);
    }
}
```



```
#pragma once
#include "GameFramework/Actor.h"
#include "MyActor.generated.h"

UCLASS()
class AMyActor : public AActor
{
    GENERATED_BODY()

    int Count;

    // Sets default values for this actor's properties.
    AMyActor()
    {
        // Allows Tick() to be called
        PrimaryActorTick.bCanEverTick = true;
    }

    // Called when the game starts or when spawned.
    void BeginPlay()
    {
        Super::BeginPlay();
        Count = 0;
    }

    // Called every frame.
    void Tick(float DeltaSeconds)
    {
        Super::Tick(DeltaSeconds);
        Count = Count + 1;
        GLog->Log(FString::FromInt(Count));
    }
};
```



脚本

概览

```
# Everything after "#" is a comment.  
# A file is a class!  
  
# (optional) icon to show in the editor dialogs:  
@icon("res://path/to/optional/icon.svg")  
  
# (optional) class definition:  
class_name MyClass  
# Inheritance:  
extends BaseClass  
  
# Member variables.  
var a = 5  
var arr = [1, 2, 3]  
var dict = {"key": "value", 2: 3}  
var typed_var: int  
var inferred_type := "String"  
  
# Enums.  
enum Named {THING_1, THING_2, ANOTHER_THING = -1}  
  
# Built-in vector types.  
var v2 = Vector2(1, 2)  
var v3 = Vector3(1, 2, 3)
```

```
# Functions.  
func some_function(param1, param2, param3):  
    if param1 < 5:  
        print(param1)  
    else:  
        print("Fail!")  
  
    for i in range(20):  
        print(i)  
  
    match param3:  
        3:  
            print("param3 is 3!")  
        _:  
            print("param3 is not 3!")  
  
    var local_var = param1 + 3  
    return local_var  
  
# Functions override functions with the same name  
func something(p1, p2):  
    super(p1, p2)  
  
# Inner class  
class Something:  
    var a = 10
```

GDScript

A high-level, object-oriented, imperative, and gradually typed programming language built for Godot.

Scripts customize and extend the behavior of Objects (Nodes).

脚本

迭代快



真“脚本”

1. No build times
2. Live script reloading

脚本

整合性好

The screenshot shows the Godot Engine Editor interface. On the left, the FileSystem browser displays a folder structure under 'res://level/'. It contains a 'background' folder which holds three files: 'cloud_1.webp', 'cloud_2.webp', and 'cloud_3.webp'. On the right, the main window shows a GDScript script named 'player.gd'. The script starts with 'class_name Player extends CharacterBody2D'. The fourth line contains the code '@onready var sprite :=', which is highlighted in red, indicating a syntax error. A tooltip below the line says 'Error at (4, 24): Expected expression for variable initial value after "=".' The bottom status bar provides information about the Godot engine version and some module imports.

```
1 class_name Player extends CharacterBody2D
2
3
4 @onready var sprite :=
5
6
7
8
9
10
11
12
13
14
15
16
< Error at (4, 24): Expected expression for variable initial value after "=".

Godot Engine v4.2.1.stable.official (c) 2007-present Juan Linietsky, Ariel Manzur & Godot Contributors.
● modules/gltf/register_types.cpp:63 - Blend file import is enabled in the project settings, but no Blender path is configured in the
editor settings. Blend files will not be imported.
--- Debug adapter server started ---
--- GDScript language server started on port 6005 ---
```

编辑器“知道”游戏场景

1. 自动补全路径 : Autocomplete can complete code & game data (node paths, file paths, animation names in a player, etc))
2. 拖文件进去自动生成读取资源的代码

脚本

整合性好

The screenshot shows the Godot Engine's integrated development environment (IDE). The top half is a code editor with tabs for 'game_singleplayer' and 'game.gd'. The code in 'game.gd' is:

```
1 class_name Game extends Node
2
3
4 @onready var _pause_menu := $InterfaceLayer/PauseMenu as PauseMenu
5
6
7 func _unhandled_input(event: InputEvent) -> void:
8     if event.is_action_pressed("toggle_fullscreen"):
9         var mode := DisplayServer.window_get_mode()
10        if mode == DISPLAY_MODE_WINDOWED:
11            mode = DISPLAY_MODE_FULLSCREEN
12        else:
13            mode = DISPLAY_MODE_WINDOWED
14        DisplayServer.window_set_mode(mode)
```

The bottom half is a debugger interface with tabs for 'Stack Trace', 'Errors', 'Profiler', 'Visual Profiler', 'Monitors', 'Video RAM', 'Misc', and 'Network Profiler'. The 'Stack Trace' tab is selected, showing the stack frames:

- 0 - res://player/player.gd:79 - at function: try_jump
- 1 - res://player/player.gd:31 - at function: _physics_process

The 'Breakpoints' tab on the right lists member variables with their object IDs:

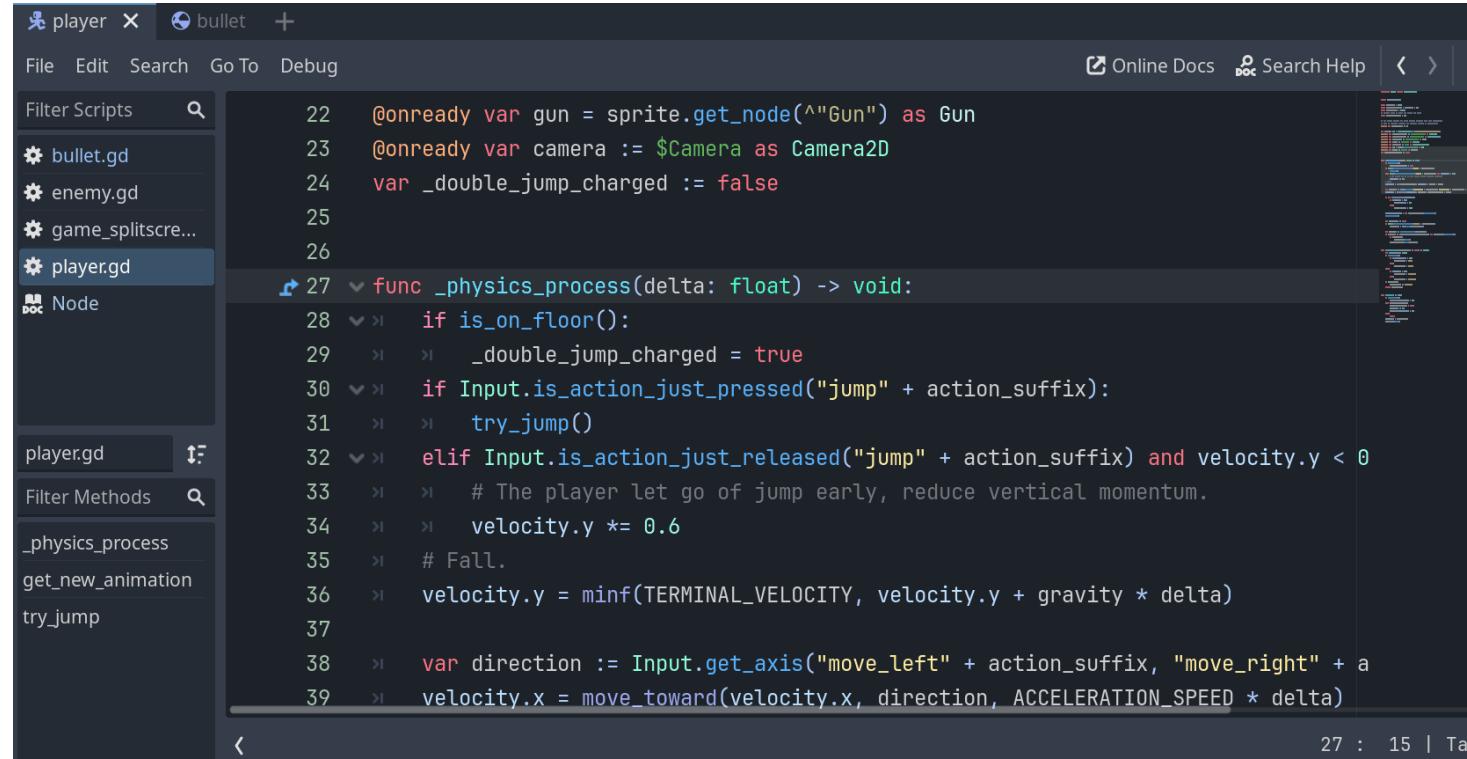
Members	Object ID
self	Object ID: 47898953851
action_suffix	
gravity	2100
platform_detector	Object ID: 47932508285
animation_player	Object ID: 48016394370
shoot_timer	Object ID: 47915731068
sprite	Object ID: 47949285502
jump_sound	Object ID: 48066726021

简单可用的debugger

1. 支持常见的debug操作
2. 支持同时连接到四个running instance
3. 和引擎的集成好

脚本

整合性好



The screenshot shows the GDScript editor interface. The title bar says "player" and "bullet". The menu bar includes File, Edit, Search, Go To, Debug, Online Docs, Search Help, and navigation arrows. On the left, there's a sidebar with "Filter Scripts" and a search icon, followed by a list of files: bullet.gd, enemy.gd, game_splitscre..., and player.gd (which is selected). Below that is a "Node" section. The main area shows the "player.gd" script with the following code:

```
22 @onready var gun = sprite.get_node(^"Gun") as Gun
23 @onready var camera := $Camera as Camera2D
24 var _double_jump_charged := false
25
26
27 func _physics_process(delta: float) -> void:
28     if is_on_floor():
29         _double_jump_charged = true
30     if Input.is_action_just_pressed("jump" + action_suffix):
31         try_jump()
32     elif Input.is_action_just_released("jump" + action_suffix) and velocity.y < 0
33         # The player let go of jump early, reduce vertical momentum.
34         velocity.y *= 0.6
35     # Fall.
36     velocity.y = minf(TERMINAL_VELOCITY, velocity.y + gravity * delta)
37
38     var direction := Input.get_axis("move_left" + action_suffix, "move_right" + a
39     velocity.x = move_toward(velocity.x, direction, ACCELERATION_SPEED * delta)
```

The status bar at the bottom right shows "27 : 15 | Tabs".

其他优势

1. Signal链接情况的可视化
2. 内置离线文档
3. 和引擎走一样的RefCount，没有(双)GC问题

脚本

贴合游戏需求

```
# No leading slash means it is relative to the current node.  
"A" # Immediate child A  
"A/B" # A's child B  
".." # The parent node.  
^"..." # The grandparent node.  
# A leading slash means it is absolute from the sceneTree.  
"/root" # Equivalent to get_tree().get_root().  
"/root/Main" # If your main scene's root node were named "Main".  
  
# Points to the Sprite2D node.  
"Path2D/PathFollow2D/Sprite2D"  
# Points to the Sprite2D node and its "texture" resource.  
"Path2D/PathFollow2D/Sprite2D:texture"  
# Points to the Sprite2D node and its "position" property.  
"Path2D/PathFollow2D/Sprite2D:position"  
# Points to the Sprite2D node and the "x" component of its "position" property.  
"Path2D/PathFollow2D/Sprite2D:position:x"
```

NodePath

```
# my_buildings.gd  
extends Node  
  
# This value is a constant, so it spawns when the script object loads.  
# The script is preloading the value. the editor can  
# offer autocompletion since it must be a static path.  
const Buildingscn = preload("res://building.tscn")  
  
# The script preloads the value, so it will load as a dependency  
# of the 'my_buildings.gd' script file.  
@export var a_building : PackedScene = preload("office.tscn")
```

preload

脚本

贴合游戏需求

```
# character.gd                                signal
extends Node

# A signal named health_depleted.
signal health_depleted

# another.gd
func _ready():
    var character_node = get_node('character')
    character_node.health_depleted.connect(_on_signal)

func _on_signal():
    get_tree().reload_current_scene()
```

```
func wait_confirmation():                  await
    print("Prompting user")
    await $Button.button_up # Waits for the button_up
                           # signal from Button node.
    print("User confirmed")
    return true

func request_confirmation():
    print("will ask the user")
    var confirmed = await wait_confirmation()
    if confirmed:
        print("User confirmed")
    else:
        print("User cancelled")
```

脚本

Modern C#

```
using Godot;
public partial class MyNode : Node
{
    // Member variables here, example:
    private int Count;

    public override void _Ready()
    {
        // called every time the node is added to the scene.
        Count = 0;
    }

    public override void _Process(double delta)
    {
        // called every frame. Delta is time since the last frame.
        Count = Count + 1;
        GD.Print(Count);
    }
}
```

```
class_name my_node
extends Node

var count : int

func _ready():
    count = 0

func _process(delta):
    count = count + 1
    print(count)
```

脚本

Current state of C# platform support in Godot 4.2

Platform	Runtimes supported	Minimum required .NET version
Windows	CoreCLR, Mono, NativeAOT	6.0 (CoreCLR, Mono), 7.0 (NativeAOT)
macOS	CoreCLR, Mono, NativeAOT	6.0 (CoreCLR, Mono), 7.0 (NativeAOT)
Linux	CoreCLR, Mono, NativeAOT	6.0 (CoreCLR, Mono), 7.0 (NativeAOT)
Android	Mono	7.0
iOS	NativeAOT	8.0
Web	-	-

Godot3.x时代主要使用Mono，4.0开始支持.NET Core，现在有了NativeAOT就可以支持导出至IOS

<https://godotengine.org/article/platform-state-in-csharp-for-godot-4-2/>

脚本

Why and why not C#

生态

Full C# 10 support

NuGet packages can be installed and used with Godot, as with any C# project.

性能

The performance of C# in Godot is roughly ~4× that of GDScript in some naive cases.

每次微软性能优化，godot也能被优化(跟车的好处)

GC

Godot整体目前不是特别在意C#的GC问题

<https://github.com/godotengine/godot-proposals/issues/7842>

支持

GDScript一等公民，C#只能算1.5等

<https://sampruden.github.io/posts/godot-is-not-the-new-unity/>

脚本

GDExtension



Godot Jolt is a native extension for the [Godot game engine](#) that allows you to use the [Jolt physics engine](#) to power Godot's 3D physics.

It functions as a drop-in replacement for Godot Physics, by implementing the same nodes that you would use normally, like `RigidBody3D` or `CharacterBody3D`.

<https://github.com/godot-jolt/godot-jolt>

GDExtension is a Godot-specific technology that lets the engine interact with native shared libraries at run-time.

You can use it to run native code without compiling it with the engine.

其他

请参看Godot文档

动画

Everything within an Object is subject to Godot animation system, so the AnimationPlayer node is far more powerful than it first appears. It can control property changes, method calls, signal emissions, signal responses, etc.

网络

Use the low-level networking API (TCP, UDP, HTTP) for maximum control and implement everything on top of bare network protocols or use the high-level API based on SceneTree (RPC, Channel) that does most of the heavy lifting behind the scenes in a generally optimized way.

物理

！请用Godot Jolt！

At Godot 4.0 alpha, the person who was responsible to develop and maintain Godot Physics left Godot to do commercial work while Godot Physics was nowhere near being ready for a stable release.

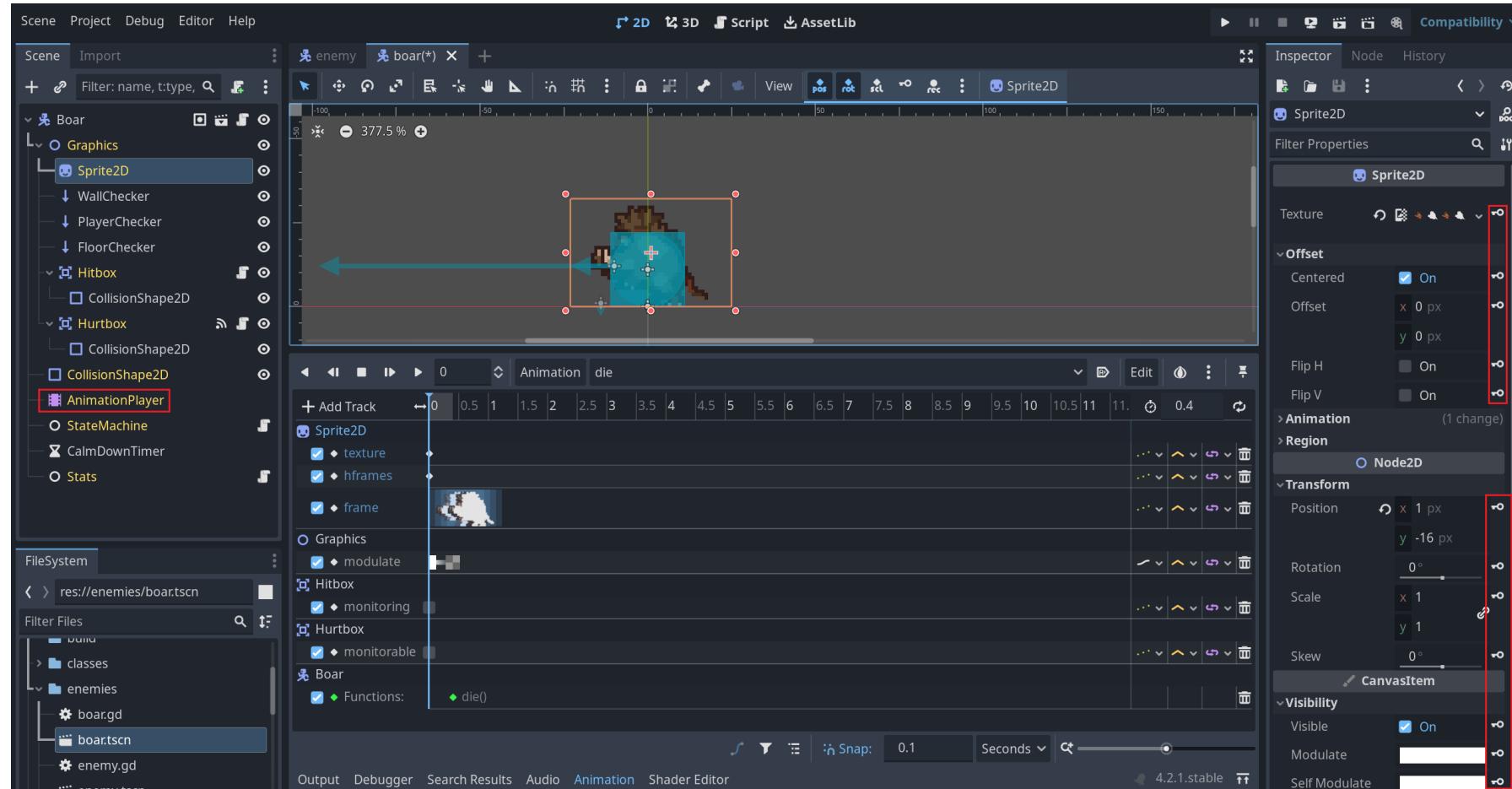
音频

Godot audio processing code has been written with games in mind, with the aim of achieving an optimal balance between performance and sound quality (Juan本身是音频编程高手)

<https://github.com/godotengine/godot/pull/64488>

其他

Extra: 动画



Animation Support

The ability to animate almost any property in any node, as well as having dedicated transform, bezier, function calling, audio and sub-animation tracks, is pretty much unique.

介绍与展示

Scene Node System

Editor

渲染

脚本

Final Thoughts

Caveats

Third party support

Will [insert closed SDK such as FMOD, GameWorks, etc.] be supported in Godot? [1](#)

The aim of Godot is to create a free and open source MIT-licensed engine that is modular and extendable. There are no plans for the core engine development community to support any third-party, closed-source/proprietary SDKs, as integrating with these would go against Godot's [ethos](#)

That said, because Godot is open source and modular, nothing prevents you or anyone else interested in adding those libraries as a module and shipping your game with them, as either open- or closed-source.

To see how support for your SDK of choice could still be provided, look at the Plugins question below.

If you know of a third-party SDK that is not supported by Godot but that offers free and open source integration, consider starting the integration work yourself. Godot is not owned by one person; it belongs to the community, and it grows along with ambitious community contributors like you.

自由的灵魂受到了协议的限制

Godot does not include an FBX importer in its editor code, as doing so would require linking against a proprietary SDK.

“It seems the biggest blockers to Godot adoption are no longer big technical issues, it is mostly ecosystem stuff (SDKs, file formats, asset store) and workflow papercuts (i.e. profiling tools, advanced settings etc.)”

— Clay john

Caveats

积压了大量的陈年pr和issue

The screenshot shows a list of open pull requests and issues from the Godot GitHub repository. At the top, it displays the total count of open issues (2,357) and closed issues (38,215). Below this, five specific items are listed:

- Don't store TileMapLayer data if empty (enhancement, topic:2d): Opened 54 minutes ago by KoBeWi, review required, priority 4.3.
- Fix error when saving nested non-persistent resource (bug, topic:core): Opened 2 hours ago by timothyqiu, review required, priority 4.3.
- Strip ANSI escape codes from file logging (bug, topic:core): Opened 2 hours ago by Calinou, review required, priority 4.3.
- Set animation step from importers. Increase default step from 10 to 30FPS (topic:core): Opened 7 hours ago by lyuma, approved, priority 4.3.

The screenshot shows a list of open pull requests and issues from the Godot GitHub repository. At the top, it displays the total count of open issues (9,937) and closed issues (39,329). Below this, four specific items are listed:

- Godot segfault at exit with threads (bug, discussion, enhancement): Opened 37 minutes ago by novalis.
- MultiplayerSynchronizer visibility results in client peer (topic:core): Opened 38 minutes ago by graydoubt.
- Web audio is distorted when using headphones (bug): Opened 54 minutes ago by kus04e4ek.

Maintainer的精力是有限的，(尤其是绝大部分也不拿工资)，而大部分Godot使用者并不具备引擎开发能力。

Feature谁来写，Bug谁来修 (QA、Soundness的问题)，维护者跑路怎么办？(例如Godot Physics)

Caveats

性能

Filters ▾ is:open is:issue label:performance

Clear current search query, filters, and sorts

(●) 278 Open ✓ 274 Closed

(●) File Dialog is extremely slow inside the editor needs testing performance
#90871 opened yesterday by aliasbody

Filters ▾ is:open is:issue label:usability

Clear current search query, filters, and sorts

(●) 636 Open ✓ 2,910 Closed

(●) TreeItem checkbox can be toggled when clicking to the left of
#91685 opened 8 hours ago by KoBeWi

性能很重要

1. 用户很少主动要求/抱怨，尤其是独立游戏开发者
2. 好的性能很难达到（源码里到处是链表，品味问题）
3. Juan本人重视 usability > performance

Per Melin @pmelin · Mar 18 ...
In a game engine, everything must have decent performance from the start or it is practically useless. That is not true for most other software. Especially not for the web. There you often start with something that is 5% as complicated and fast as an optimized solution.

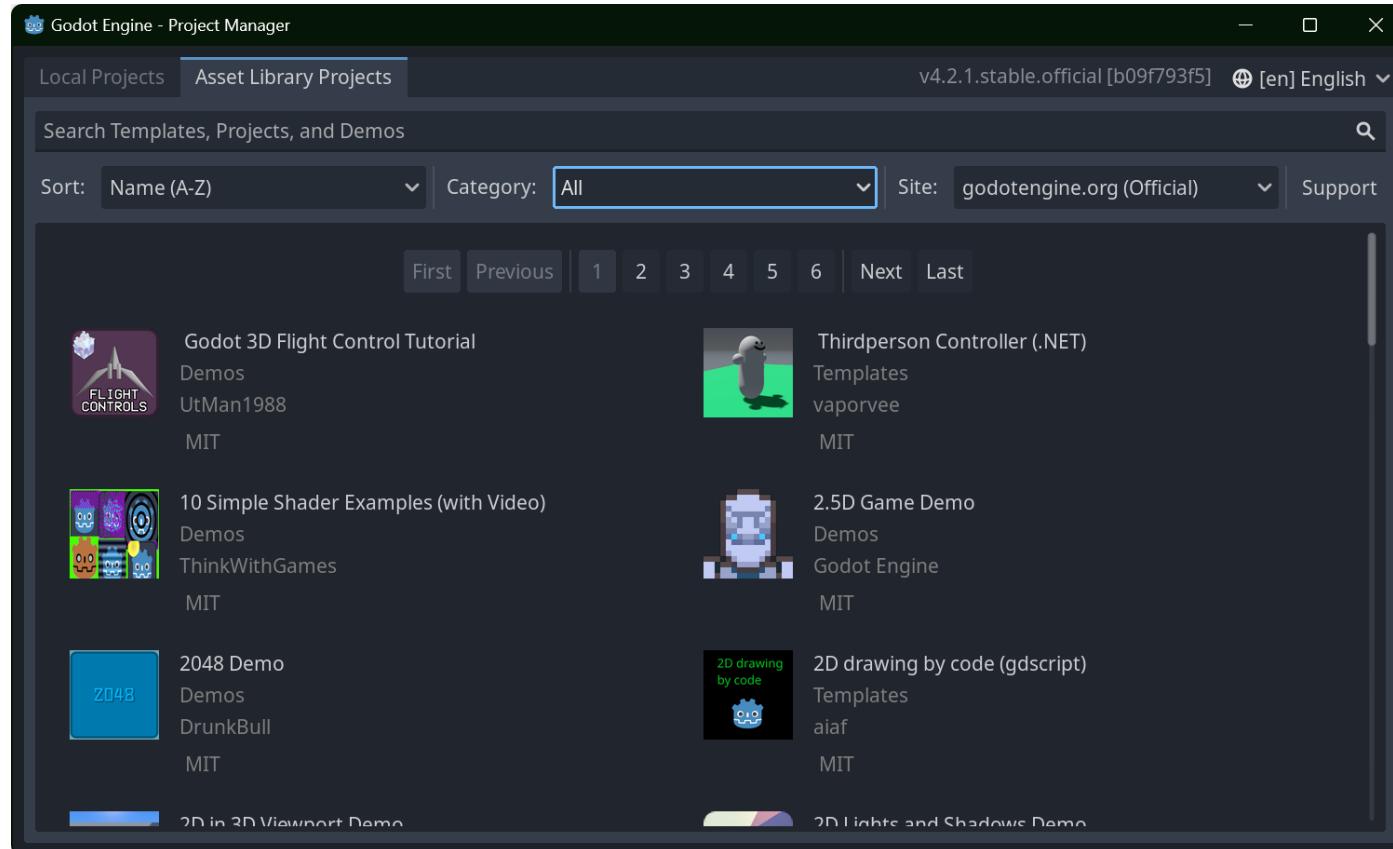
1 1.1K 5 ↗ ↘

Juan Linietsky @reduzio · Mar 18 ...
Yes but..
You can compromise on performance up to some point, initially.
You can absolutely not compromise on usability.
Further optimization can happen later after usability is ironed out.

4 998 18 ↗ ↘

Caveats

Asset store生态



内容很少

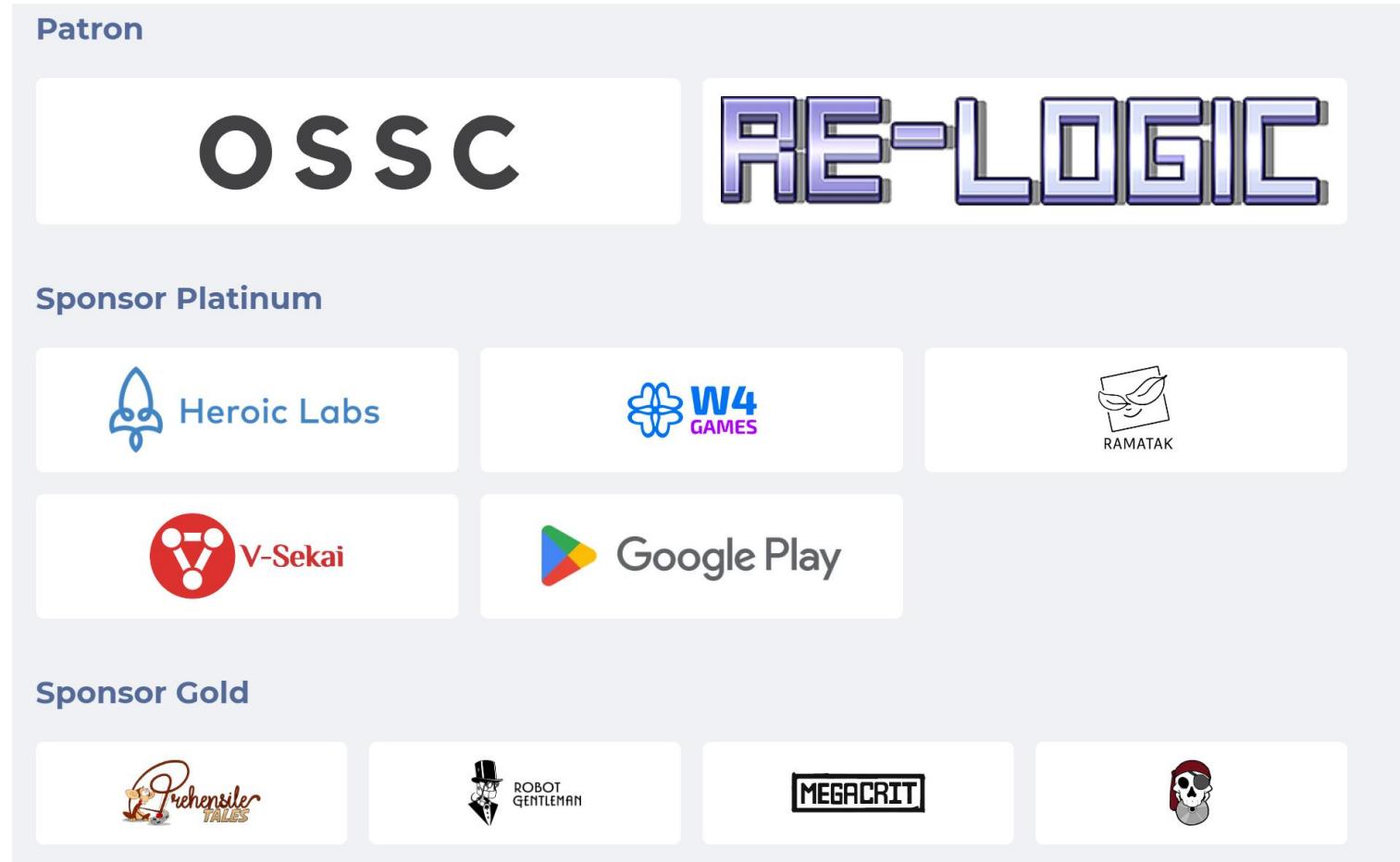
As part as SFC, Godot had no legal means to do an asset store, but the Godot Foundation now can.

目前里面都是免费的插件等

<https://godotengine.org/asset-library/asset>

Caveats

Solution



<https://fund.godotengine.org/>

Godot Foundation

- 利用Godot Foundation的资金雇佣人写代码，但对它雇佣的Maintainer也没有特别的要求，一般都是「在某一领域内，找你自己感兴趣的东西去实现/修复」
- As part as SFC, Godot had no legal means to do an asset store, but the Foundation now can.



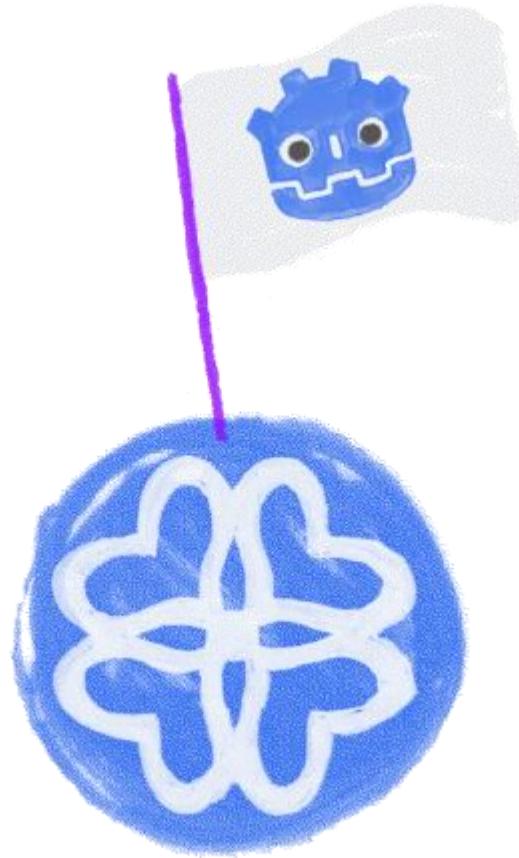
<https://godotengine.org/article/godots-graduation-godot-moves-to-a-new-foundation/>

Caveats

Solution



<https://www.w4games.com/>



偏B端

1. 拉投资成立W4 games: Godot作为开源项目，无法与微软索尼任天堂签开发协议，也无法将主机支持相关的代码开源，但w4可以去做主机相关的支持。
2. Maintainer自己开咨询公司，提供2B的商业支持。例如，
<https://twitter.com/vertexludi>

总结

提出正确的问题

~~Godot为什么这也没有那也没有~~

总结

提出正确的问题

Godot为什么能这么好

开源精神

程序员的浪漫

人人为我， 我为人人

开源精神

游戏开发的未来

Godot as a Part of the Foundation for Future Game Development

Perhaps a part of this future involves software that is open source, democratically owned, and community-funded like Godot

<https://www.youtube.com/watch?v=EYt6uDr-PHQ>

结语

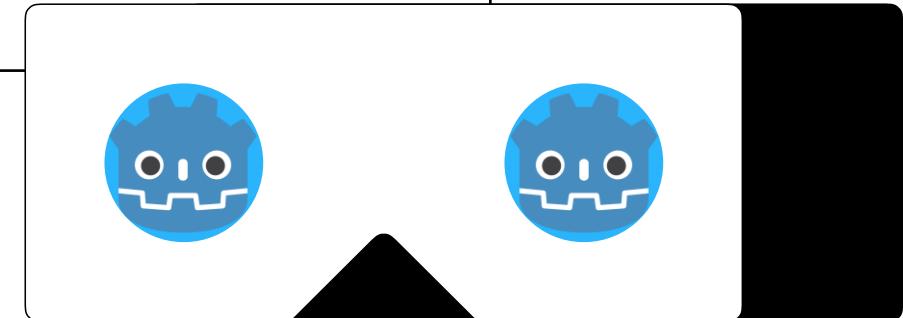
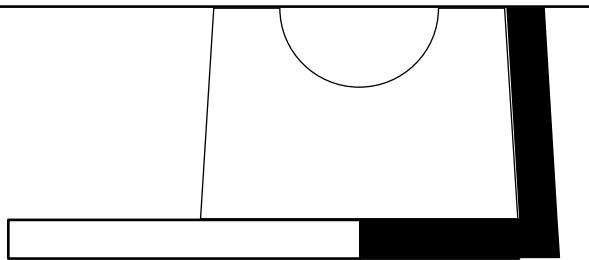
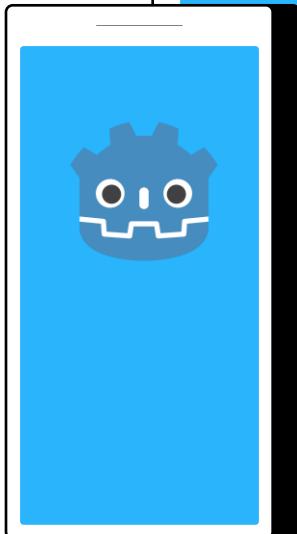
Join us

Godot belongs to all of us

Make game development open for everyone and help devs to own the technology they use.

<https://gdcvault.com/play/1027937/Open-Source-Game-Development-Summit>

谢谢



More

一些资料

术

1. 官方文档: <https://docs.godotengine.org/en/stable/index.html>
2. Github仓库地址: <https://github.com/godotengine/godot>
3. timothyqiu大佬的Godot教程: <https://space.bilibili.com/7092/channel/collectiondetail?sid=1304862>
4. Brackeys的教程: <https://www.youtube.com/watch?v=LOhfqjmasi0>
5. Juan给Unity迁移到Godot的建议: <https://twitter.com/reduzio/status/1703328476424548756>
6. Godot源码分析: <https://www.youtube.com/watch?v=vTIEM2tJavo>
7. Ways to contribute: https://docs.godotengine.org/en/stable/contributing/ways_to_contribute.html

道

1. The Future of Game Development: <https://www.youtube.com/watch?v=EYt6uDr-PHQ>
2. Godot as an Open Ecosystem: <https://www.youtube.com/watch?v=XduuHNOGGqI>
3. Understanding the Godot Engine Architecture and Development Process:
<https://gdcvault.com/play/1027937/Open-Source-Game-Development-Summit>

Bonus

基于Rust的游戏引擎

