Godot - Die Open Source Game Engine

Jean-Frédéric Vogelbacher

Über mich

- Jean-Frédéric Vogelbacher
- 21 Jahre alt
- Informatik-Studium an der Friedrich-Alexander Universität Erlangen-Nürnberg
- Hobby-Spieleentwickler auf der Godot Engine seit 2018
- Projekte:
  - YouTube-Kanal Linux Guides
  - Tux-Tage 2020, 2021
  - Libre TrainSim
  - ...

Inhalt

Was ist eine Game Engine?

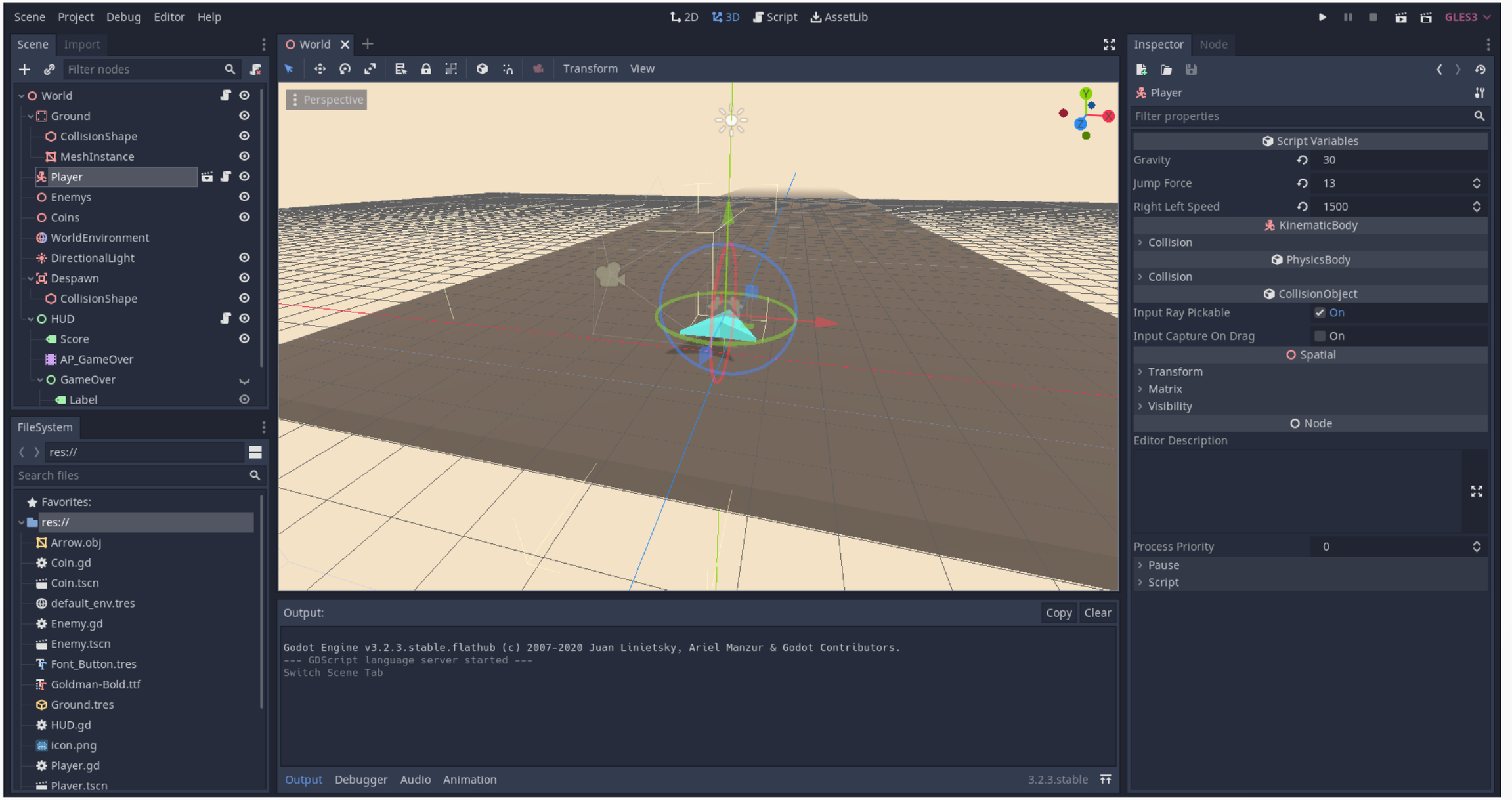
- Zusammenfassung: Framework für Computerspiele, welches es Entwicklern vereinfacht, vor allem Spiele zu entwickeln
  - Meist eigener Editor
  - Bringt viele Vereinfachungen mit (beispielsweise mathematische Bibliotheken…)
  - Sound/Physik/3D/2D-Engine meist direkt implementiert
  - Spiel läuft am Ende „in“ dieser Engine
  - Verschiedenste Programmiersprachen unterstützt
  - I.d.R. einmal Programmieren, für viele Plattformen bauen

Weitere Game Engines

Inhalt

Features

- Innovatives Design:
  - Einfaches „Node“-System:
    - Alle Spiel-Elemente/Objekte bestehen aus einer speziellen Node-Klasse, welche wieder aus einzelnen Nodes besteht. (Baum-Struktur, „Objekt-Orientierung at its finest“)
    - Sogar die ganze Szene ist am Ende ein Node, welches viele „Kinder-Nodes“ hat.
    - Man kann auch seine eigenen Nodes erstellen, und 1000-fach in Szenen verwenden
  - Leicht zu verstehendes Interface
  - Einfach, bestehenden Editor zu modifizieren, und eigene AddOns zu schreiben

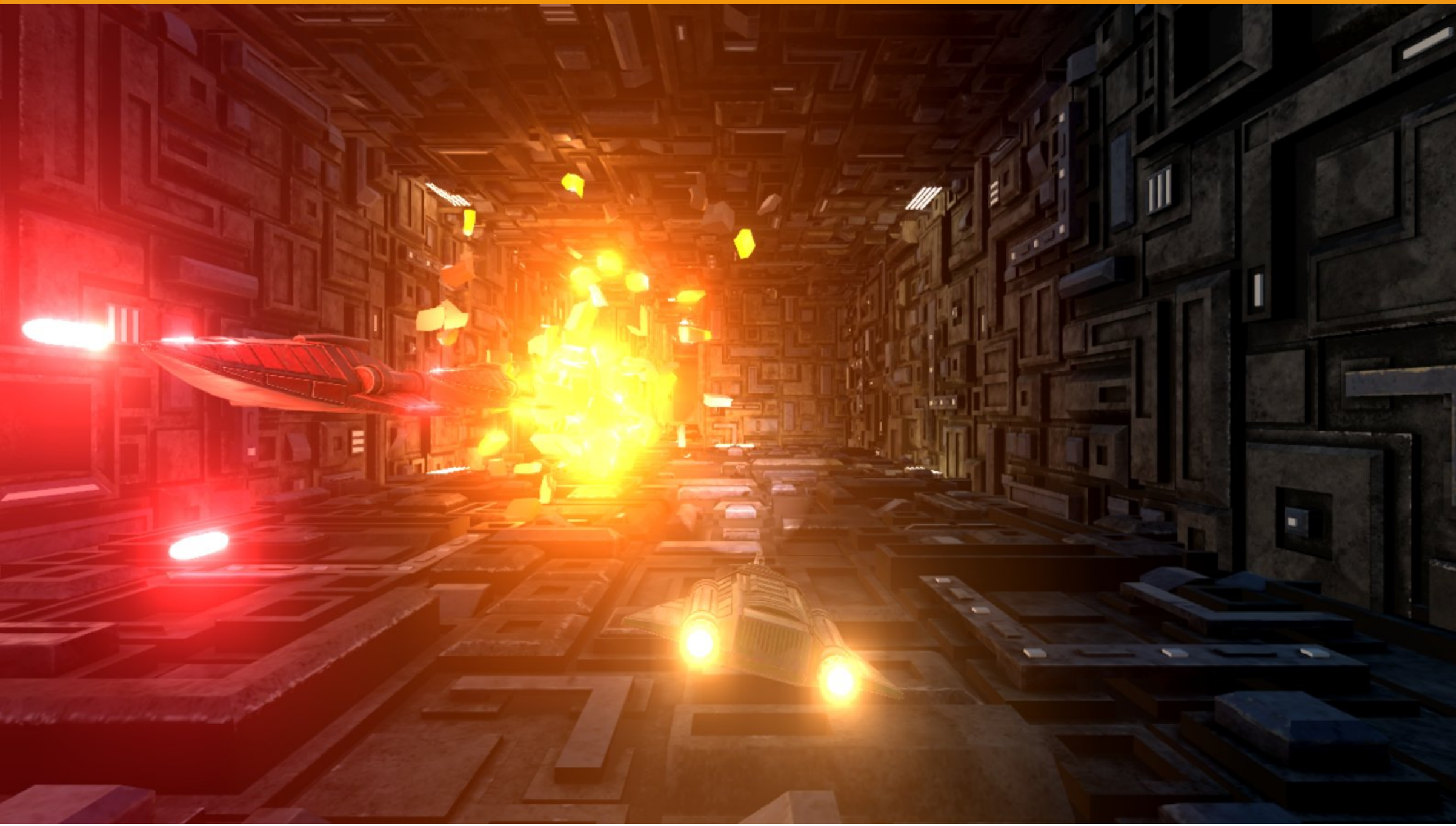

Scene Project Debug Editor Help
2D 3D Script AssetLib
GLES3
Scene Import
Filter nodes
World
Ground
CollisionShape
MeshInstance
Player
Enemys
Coins
WorldEnvironment
DirectionalLight
Despawn
CollisionShape
HUD
Score
AP_GameOver
GameOver
Label
FileSystem
res://
Search files
Favorites:
res://
Arrow.obj
Coin.gd
Coin.tscn
default_env.tres
Enemy.gd
Enemy.tscn
Font_Button.tres
Goldman-Bold.ttf
Ground.tres
HUD.gd
Icon.png
Player.gd
Player.tscn
World
Transform View
Perspective
Output:
Copy Clear
Godot Engine v3.2.3.stable.flathub (c) 2007-2020 Juan Linietsky, Ariel Manzur & Godot Contributors.
--- GDScript language server started ---
Switch Scene Tab
Output Debugger Audio Animation
3.2.3.stable
Inspector Node
Player
Filter properties
Script Variables
Gravity 30
Jump Force 13
Right Left Speed 1500
KinematicBody
Collision
PhysicsBody
Collision
CollisionObject
Input Ray Pickable On
Input Capture On Drag On
Spatial
Transform
Matrix
Visibility
Node
Editor Description
Process Priority 0
Pause
Script

Features

- 2D und/oder 3D:
  - „Eine der besten, wenn nicht die beste 2D-Engine, die es auf dem Markt gibt“
  - Bei 3D gibt es noch etwas Nachholbedarf, was aber mit Godot 4.0 wahrscheinlich geschafft wird.

Features:

- 3D:
  - Physisch basiertes Rendering
  - Komplett implementiertes BSDF-Rendering: Alles, was Materialen brauchen
  - Global Illumination, die auch wahlweise „gebacken“ werden kann
  - Mid- und Post-Processing inklusive HDR, Nebel, Bloom, Reflektionen....
  - Einfach zu nutzende Shader-Sprache auf GLSL basierend
  - obj, fbx, gltf...



Quelle: https://godotengine.org

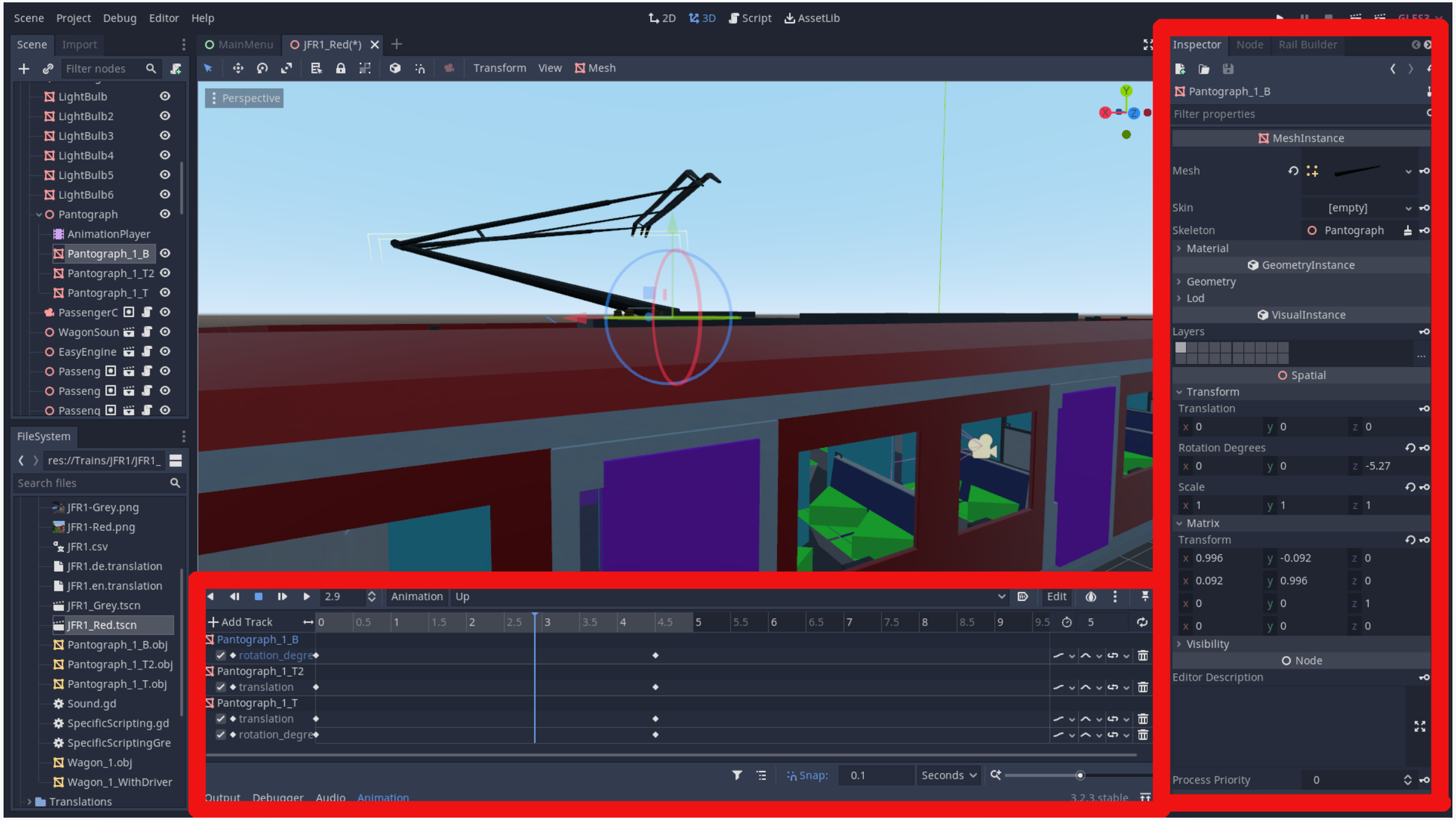
Features

- 2D:
  - Tile-Map Editor mit autotiling, rotation, mehrere Schichten, ...
  - 2D Lichter und normal Maps
  - 2D Animator

Quelle: https://godotengine.org

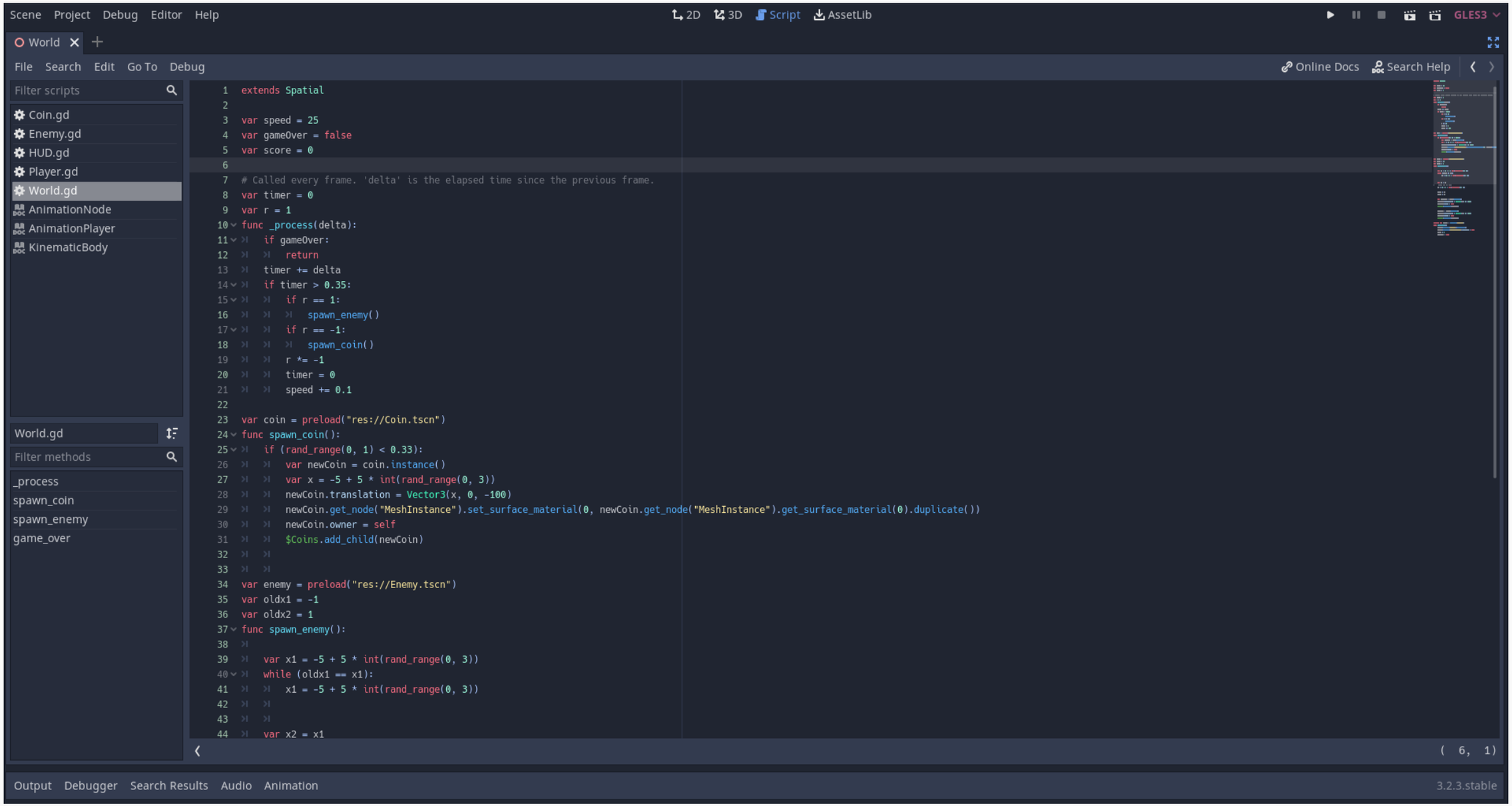
Features

- Animationen:
  - Man kann einfach fast alles animieren (2D & 3D)
  - Helfer, um 2D-Skelette zu animieren
  - Custom Transition Curves


Scene Project Debug Editor Help
2D 3D Script AssetLib
Scene Import
Filter nodes
LightBulb
LightBulb2
LightBulb3
LightBulb4
LightBulb5
LightBulb6
Pantograph
AnimationPlayer
Pantograph_1_B
Pantograph_1_T2
Pantograph_1_T
PassengerC
WagonSoun
EasyEngine
Passeng
Passeng
Passeng
FileSystem
res://Trains/JFR1/JFR1_
Search files
JFR1-Grey.png
JFR1-Red.png
JFR1.csv
JFR1.de.translation
JFR1.en.translation
JFR1_Grey.tscn
JFR1_Red.tscn
Pantograph_1_B.obj
Pantograph_1_T2.obj
Pantograph_1_T.obj
Sound.gd
SpecificScripting.gd
SpecificScriptingGre
Wagon_1.obj
Wagon_1_WithDriver
Translations
MainMenu
JFR1_Red(*)
Transform View Mesh
Perspective
2.9
Animation
Up
Edit
Add Track
Pantograph_1_B
rotation_degre
Pantograph_1_T2
translation
Pantograph_1_T
translation
rotation_degre
Snap:
0.1
Seconds
Output Debugger Audio Animation
3.2.3.stable
Inspector
Node
Rail Builder
Pantograph_1_B
Filter properties
MeshInstance
Mesh
Skin
[empty]
Skeleton
Pantograph
Material
GeometryInstance
Geometry
Lod
VisualInstance
Layers
Spatial
Transform
Translation
Rotation Degrees
Scale
Matrix
Transform
Visibility
Node
Editor Description
Process Priority

Features

- Skripting:
  - GDScript: Python ähnliche Sprache, sehr einfach zu erlernen
  - C#, welches Mono benutzt
  - Visual Scripting
  - Community-Erweiterungen für Rust, Nim, D, und andere Sprachen
  - Eingebauter Script-Editor mit Syntax Highlighting und Autocompletion mit RealTime Parser
  - (C++ mit GDNative, nicht für‘s alltägliche Skripten gedacht)


Scene Project Debug Editor Help
2D 3D Script AssetLib
GLES3
World
File Search Edit Go To Debug
Online Docs Search Help
Filter scripts
Coin.gd
Enemy.gd
HUD.gd
Player.gd
World.gd
AnimationNode
AnimationPlayer
KinematicBody
World.gd
Filter methods
_process
spawn_coin
spawn_enemy
game_over
extends Spatial
var speed = 25
var gameOver = false
var score = 0
Called every frame. 'delta' is the elapsed time since the previous frame.
var timer = 0
var r = 1
func _process(delta):
if gameOver:
return
timer += delta
if timer > 0.35:
if r == 1:
spawn_enemy()
if r == -1:
spawn_coin()
r *= -1
timer = 0
speed += 0.1
var coin = preload("res://Coin.tscn")
func spawn_coin():
if (rand_range(0, 1) < 0.33):
var newCoin = coin.instance()
var x = -5 + 5 * int(rand_range(0, 3))
newCoin.translation = Vector3(x, 0, -100)
newCoin.get_node("MeshInstance").set_surface_material(0, newCoin.get_node("MeshInstance").get_surface_material(0).duplicate())
newCoin.owner = self
$Coins.add_child(newCoin)
var enemy = preload("res://Enemy.tscn")
var oldx1 = -1
var oldx2 = 1
func spawn_enemy():
var x1 = -5 + 5 * int(rand_range(0, 3))
while (oldx1 == x1):
x1 = -5 + 5 * int(rand_range(0, 3))
var x2 = x1
(6, 1)
Output Debugger Search Results Audio Animation
3.2.3.stable

Features

- Debugging & Optimierungen:
  - Live-Debugging: Debuggen, während das Spiel läuft (aus eigener Erfahrung sehr sehr hilfreich)
  - Eigener Profiler für Speicherauslastung, FPS, Netzwerkauslastung, ...
  - Netzwerk-Profiler


Scene Project Debug Editor Help
2D 3D Script AssetLib
GLES3
Debugger Errors (74) Profiler Network Profiler Monitors Video RAM Misc
Monitor Value
Static 82.87 MiB
Dynamic 36.09 MiB
Static Max 114.8 MiB
Dynamic Max 68.09 MiB
Msg Buf Max 2.53 KiB
Object
Objects 2383
Resources 221
Nodes 1016
Orphan Nodes 4
Raster
Objects Drawn 38
Vertices Drawn 7821354
Mat Changes 65
Shader Changes 51
Surface Changes 140
Draw Calls 186
2d
Items 52
Draw Calls 241
Video
Video Mem 76.41 MiB
Texture Mem 73.56 MiB
Vertex Mem 2.84 MiB
Video Mem Max 0 B
Physics 2d
Active Objects 0
Collision Pairs 0
Islands 0
Physics 3d
Active Objects 0
Collision Pairs 0
Islands 0
Audio
Output Latency 54.32 ms
Fps
30
24
18
12
6
0
Process
0.29 ms
76.23 ms
57.17 ms
38.11 ms
19.05 ms
0.00 ms
Static
82.87 MiB
70.92 MiB
53.19 MiB
35.46 MiB
17.73 MiB
0 B
Dynamic
36.09 MiB
28.87 MiB
21.65 MiB
14.43 MiB
7.21 MiB
0 B
Objects
2383
2160
1620
1080
540
0
Objects Drawn
38
31
23
16
8
0
Vertices Drawn
7821354
6466260
4849695
3233130
1616565
0
Video Mem
76.41 MiB
61.13 MiB
45.84 MiB
30.56 MiB
15.28 MiB
0 B
Output Debugger (74) Audio Animation
3.2.3.stable

Features

- Cross-Plattform Editor mit allen Features auf:
  - Windows
  - MacOS
  - Linux und *BSD
- Kleiner Download: (30 MB !)
- Einfaches Kompilieren

  (optional, aus den Quellen)

Features

- Cross-Plattform Support für Exports:
  - Windows
  - macOS
  - Linux, UWP, *BSD
  - iOS
  - Android
  - HTML 5, Webassembly
  - Konsolen: (Drittanbieter Software benötigt)
    - Switch
    - PS4
    - Xbox One
    - (die neue Konsolen-Generation kommt bestimmt auch)

Features

- XR-Support
  - Augmented und Virtual Reality
  - OpenVR, OpenXR, Oculus SKDs, AR Kit, WebXR
  - Geräte:
    - HTC Vive
    - Valve Index
    - Oculus Rift
    - Oculus Go
    - Oculus Quest
    - Microsoft MR headsets
    - ...

Features

- Einfaches Teamwork:
  - Problemlose Versionsverwaltung
    - Git, Subversion, Mercurial, …. laufen gut damit
  - Szenen Instanziierung
    - Das einfache „Node“-System macht es möglich
  - GitHub unterstützt auch GDScript

Features

Sonstige:

- Audio Engine (Bus-System, 2D/3D Sound, Effekte, Equalizer, Filter...)
- Physik Engine (optional auch mit Bullet)
- Input Unterstützung (Maus, Multitouch, Controller, etc...)
- Starker GUI-Editor
- Networking (ENET, Websocket, WebRTC)
- Multi-Threading (Threads, Mutex, Semaphores, ...)

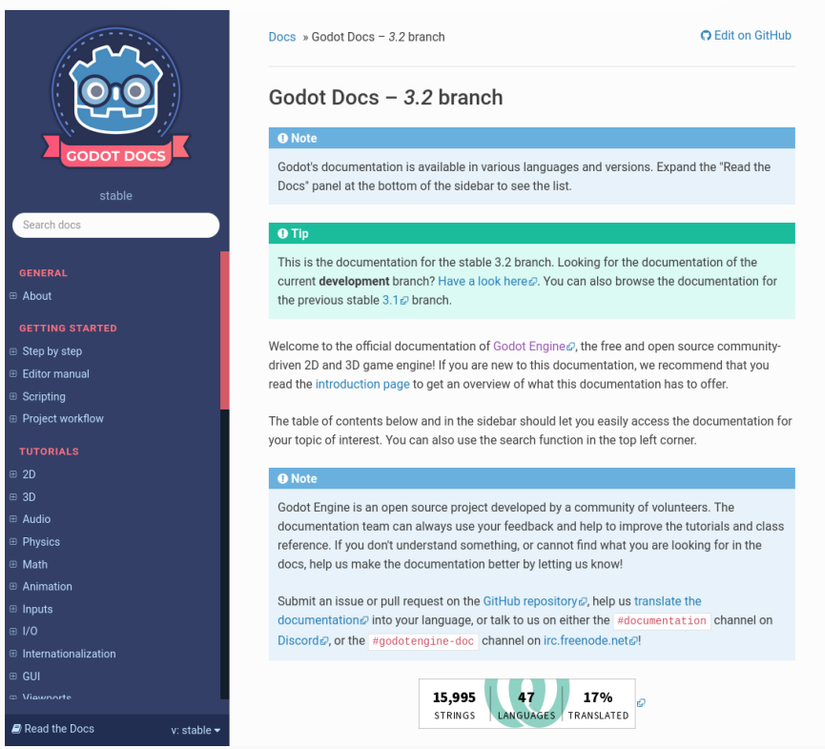
Features

- Hilfreiche & nette Community
  - StackOverflow-ähnliche Seite
  - Discord
  - IRC
  - Matrix
  - Reddit
  - Forum
  - Lokale Gemeinschaften auf der gesamten Welt
  - Asset Libary mit 100en Demos, Templates, Plugins

Quelle: https://github.com/Zylann/godot_heightmap_plugin

Features

- Gute, mehrsprachige Dokumentation, die auch teilweise im Editor aufgerufen werden kann
  - Tutorials zu verschiedensten Themen



Inhalt

1) Was ist eine Game Engine?
2) Godot vorgestellt
3) **Was macht Godot als Game Engine besonders?**
4) Nachteile
5) Spiele entwickelt mit Godot
6) Wie selber anfangen?

Was macht die Godot Engine besonders?

- Sie ist komplett frei zu nutzen, und wird nie etwas kosten
- Sehr einfach zu erlernen und zu nutzen
- Viele Features, kann (fast) mit großen kommerziellen Engines mithalten
- Editor funktioniert ohne Einschränkungen auch auf Linux
- Durch „Node“-System sehr intuitiv und effizient

Inhalt

1) Was ist eine Game Engine?
2) Godot vorgestellt
3) Was macht Godot als Game Engine besonders?
4) **Nachteile**
5) Spiele entwickelt mit Godot
6) Wie selber anfangen?

Nachteile

- 3D Engine ist noch etwas performance hungrig, besonders mit Schatten (wird hoffentlich in Godot 4.0 mit Vulkan behoben)
- Noch vergleichsweise geringer Marktanteil
  - → Dieser wächst aber stetig
- Community noch kleiner als im Vergleich zu UE und Unity.
  - Es gibt einfach noch nicht so viele Bücher, Ressourcen, Kurse, etc.
- Noch viel einfacher, einen Job mit UE oder Unity Erfahrung in dem Bereich zu bekommen

Inhalt

1) Was ist eine Game Engine?
2) Godot vorgestellt
3) Was macht Godot als Game Engine besonders?
4) Nachteile
5) Spiele entwickelt mit Godot
6) Wie selber anfangen?

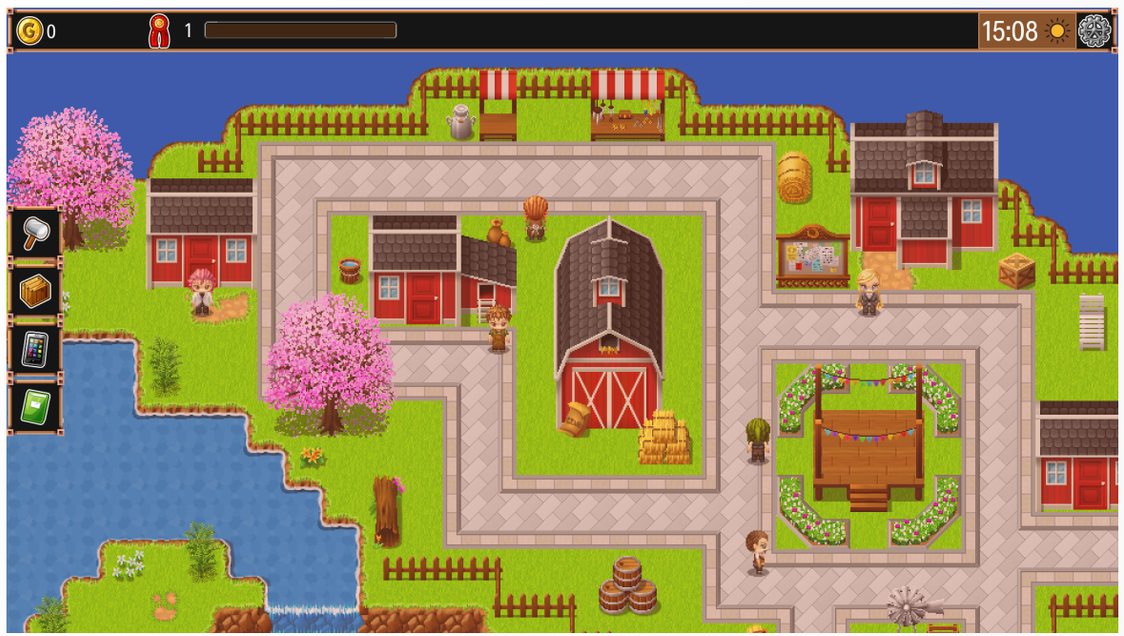
SuperTuxParty (Alpha)

- Mario Party Klon mit einigen Minispielen
- Frei erhältlich über Flathub

Quelle: http://supertux.party/

Go Farm Yourself

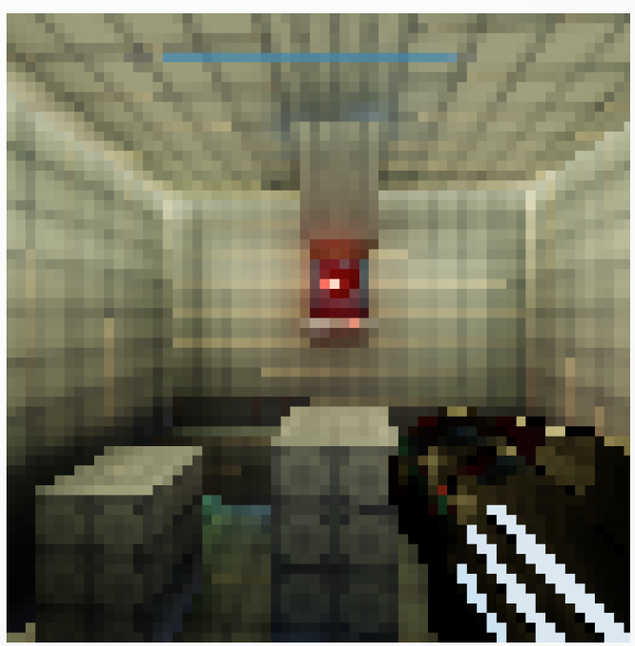
- 2D Farming Spiel
- Auch auf dem Handy spielbar
- Frei erhältlich über itch



Quelle: https://codegoons.itch.io/go-farm-yourself

Meteorite

- An Metroid Prime angelehnter Shooter
- 3D Pixel Grafik
- Frei erhältlich über itch



Quelle: https://bauxite.itch.io/meteorite

Locked UP

- Stimmungsvolles „Adventure-Game“
- Frei spielbar über itch.io

Quelle: https://tntc-lab.itch.io/locked-up

Libre TrainSim

- Freier 3D Zug-Simulator
- Frei erhältlich über Flathub

Inhalt

1) Was ist eine Game Engine?
2) Godot vorgestellt
3) Was macht Godot als Game Engine besonders?
4) Nachteile
5) Spiele entwickelt mit Godot
6) **Wie selber anfangen?**

Wie selber anfangen?

- Herunterladen von Godot und anschauen verschiedener Demos, die sehr einfach in Godot selbst heruntergeladen werden können, und ein bisschen mit den Demos versuchen zu „spielen“
- Programmieren eines ersten Spiels:
  - https://docs.godotengine.org/en/stable/getting_started/step_by_step/your_first_game.html
- Programmieren eines kleinen eigenen 2D Spiels (innerhalb einer Woche fertig sein)
- An Game Jams teilnehmen!
- Erst nach und nach mit dem eigenen „Spiele Traum“ anfangen. Wichtig: Du bist kein Entwicklerstudio mit 100en Mitarbeitern!

Wie selber anfangen?

- Godot-Ressourcen:
  - https://docs.godotengine.org/
  - YouTube: GDQuest
  - YouTube: Games from Scratch
  - Demos
  - Discord-Gruppe
  - Linux Guides Playlist
- Allgemeine Ressourcen:
  - YouTube: Jonas Tyroller

Vielen Dank für‘s Zuhören!

Noch Fragen?