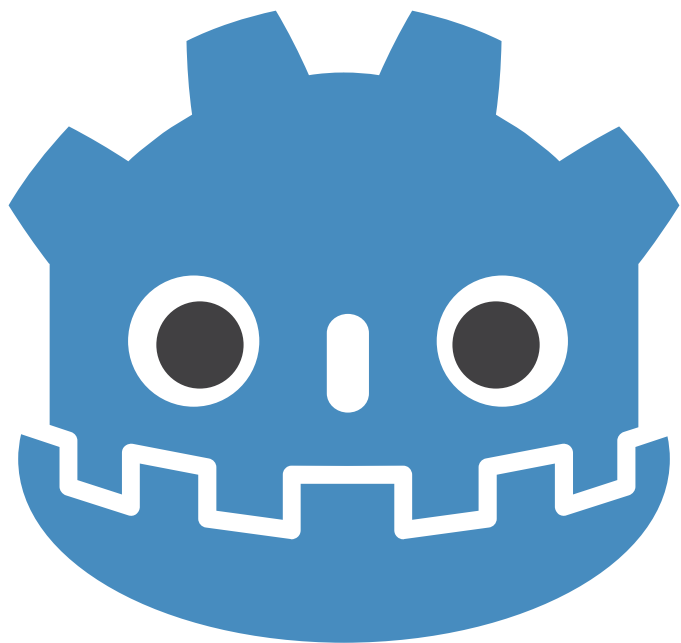


Motor de videojuegos



GODOT

#pyDayTF



Presentación

- Informática ULPGC
- Profesor FP IES Puerto de la Cruz
- Software Libre, GNU/Linux
- No trabajo en la industria de los Videojuegos
- Compartir experiencia

Agenda

- Elegir un motor de videojuegos
- Características de Godot Engine
- GDPython
- Preguntas

Elegir un motor

- Programación de cero => ¡uf!.
- No hay que reinventar la rueda => Game Engine.
- ¿Mis requisitos?
- Software libre, multiplataforma, exportador, 2D/3D, físicas, sonidos/música, input, etc.
- Al final, siempre se necesita programar algo...

Godot Engine

- OKAM libera Godot Engine en 2014.
- Creación de juegos, aplicaciones multimedia e interactivas.
- Objetivo: casero, diversión, profesional.
- Integración de roles en la misma herramienta: diseñador, gráficos, programador.
- Herramientas complementarias: GIT, Inkscape, GIMP, Blender.

Instalación

- URL: godotengine/download
- Descargar ejecutable para nuestro SO.
- Descomprimir y usar.
- Templates: para exportar a otras plataformas.
- Demos y ejemplos.



NEWS

FEATURES

DOWNLOAD

COMMUNITY

SHOWCASE

DEVLOGS

DOWNLOAD

Godot Engine 2.1.2-stable

RELEASED 2017/01/21

 Linux 64-bit | [\[Mirror 2\]](#)

 Linux 32-bit | [\[Mirror 2\]](#)

 Linux Server 64-bit | [\[Mirror 2\]](#)

 OS X (32-bit + 64-bit) | [\[Mirror 2\]](#)

 Windows 64-bit | [\[Mirror 2\]](#)

 Windows 32-bit | [\[Mirror 2\]](#)

 Export Templates | [\[Mirror 2\]](#)



The latest stable release is also available on [Steam](#).



Godot's full source code is available to you under the very permissive [MIT License](#). No royalties are required, you have full control over your game and engine.



Características

- Gráficos 2D/3D
- Input (Teclado, joystick, pantalla táctil)
- Sonido/Música
- Física (Colisiones, formas, fuerzas)
- Efectos (Sistemas de partículas, parallax, etc.)
- Menús/Ventanas
- Acceso a ficheros (JSON)
- Exportación a distintas plataformas



Listado de Proyectos

Templates

Nombre



Editar


Ejecutar


Escanear

Proyecto Nuevo

Importar

Quitar

★  demo-07-nightmare
/home/david/proy/tools/godot-school/demos/07-nightmare

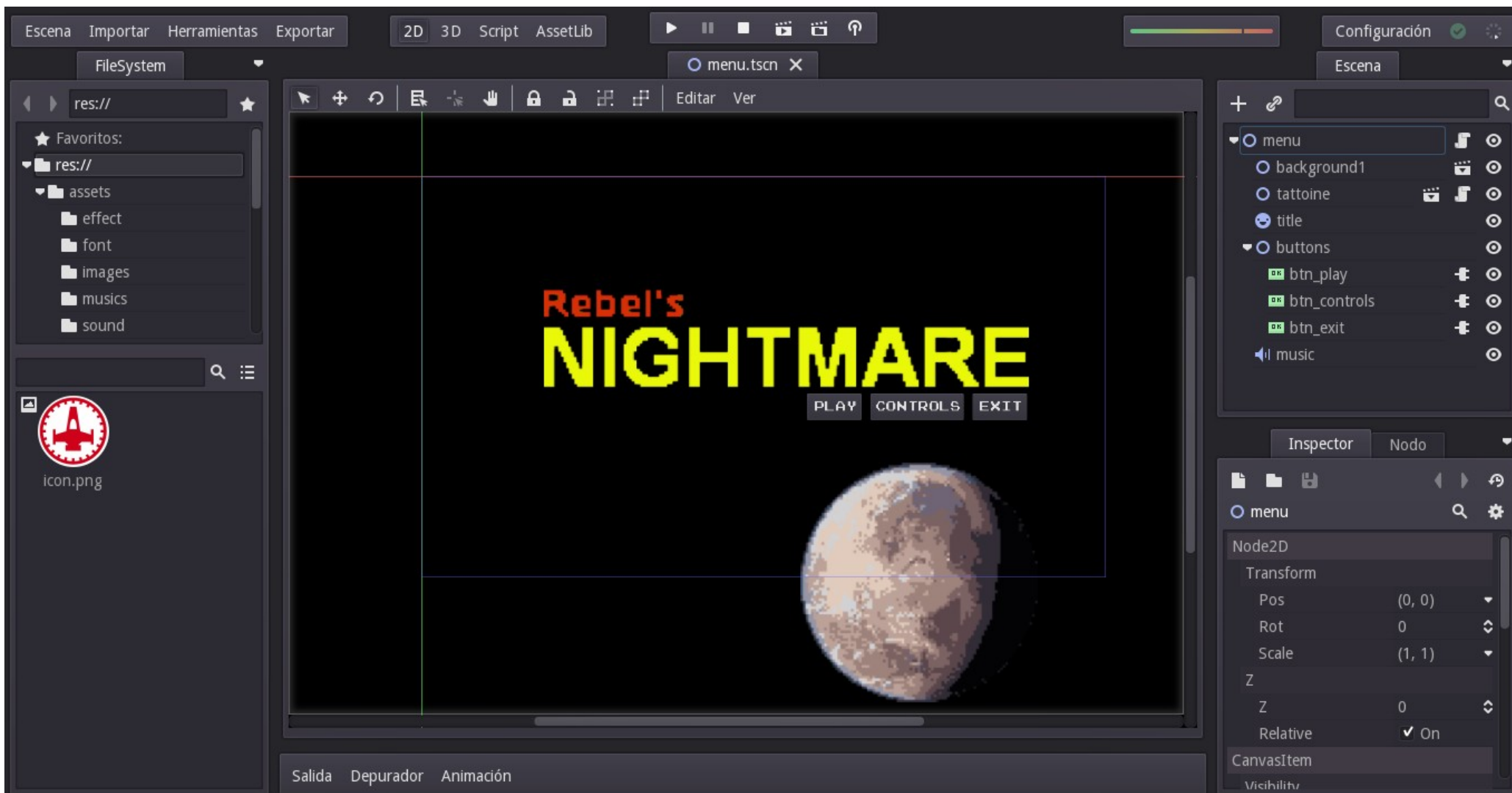
★  demo-08-numfinder
/home/david/proy/tools/godot-school/demos/08-numfinder

★  0002 - Godot Gameboy Shaders
/home/david/Descargas/godotengine/0002 - Godot Gameboy Shaders

Salir

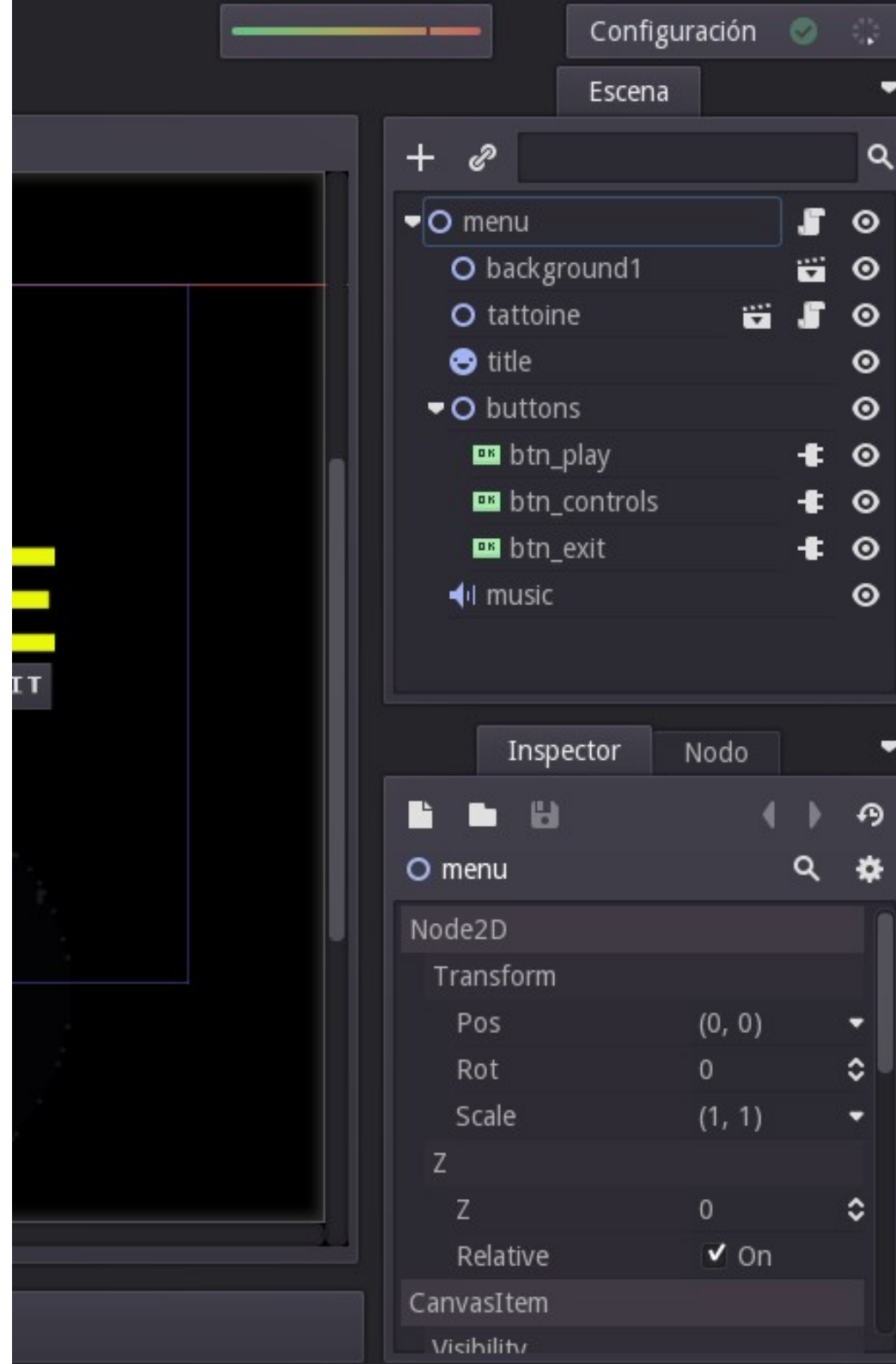
Entorno de trabajo

- Editor 2D, 3D, Scripting
- Ficheros de texto plano (GIT)
- Control de versiones: trabajo individual y en equipo.



Escenas y nodos

- Escena: Divide y vencerás (.tscn).
- Las escenas son árbol de nodos.
- Los nodos agregan propiedades y características a las escenas.
- A las escenas se les puede añadir programación y también otras escenas.



menu.tscn X

Archivo Editar Buscar Depurar Ayuda

Tutoriales Clases Ayuda de Búsqueda

Buscar Clases

Buscar:

sprite

Lista de Clases:

- Object
 - Node
 - CanvasItem
 - Node2D
 - AnimatedSprite
 - Sprite
 - ViewportSprite
 - Spatial
 - VisualInstance
 - GeometryInstance
 - SpriteBase3D
 - AnimatedSprite3D
 - Sprite3D

Cancelar

Abrir

Scripting

- Se requiere un language de scripting.
- Puede invocar todo el catálogo de nodos existente.
- Crear nuevas funcionalidades.
- ¿Por qué GDPython?
- Es como Python... pero con algunas adaptaciones.
- Recolector de basura
- Compilación


```
# a file is a class!

# inheritance
extends BaseClass

# member variables
var a = 5
var s = "Hello"
var arr = [1, 2, 3]
var dict = {"key": "value", 2:3}

# constants
const answer = 42
const thename = "Charly"

# built-in vector types
var v2 = Vector2(1, 2)
var v3 = Vector3(1, 2, 3)
```

```
# function
func some_function(param1, param2):
    var local_var = 5

    if param1 < local_var:
        print(param1)
    elif param2 > 5:
        print(param2)
    else:
        print("fail!")

    for i in range(20):
        print(i)

    while(param2 != 0):
        param2 -= 1

    var local_var2 = param1+3
    return local_var2
```

```
# inner class
class Something:
    var a = 10
```

```
|
```

Particularidades

- Acceso a los nodos de la escena.
- Acceso a métodos/atributos de los nodos y escenas.
- Bucle del juego.

```
extends KinematicBody2D
```

```
var speed = 200
```

```
var state = "playing"
```

```
func _ready():
```

```
    add_to_group("player")
```

```
    set_fixed_process(true)
```

```
func _fixed_process(delta):
```

```
    var vel = Vector2(0,0)
```

```
    if state=="playing":
```

```
        var pos = get_pos()
```

```
        var screen = get_viewport_rect().size
```

```
        var margin = 30
```

```
        if Input.is_action_pressed("player_left") and pos.x>margin:
```

```
            vel += Vector2(-speed*delta,0)
```

```
        if Input.is_action_pressed("player_right") and pos.x<(screen.x-margin):
```

```
            vel += Vector2(speed*delta,0)
```

```
        if Input.is_action_pressed("player_up") and pos.y>margin:
```

```
            vel += Vector2(0,-speed*delta)
```

```
        if Input.is_action_pressed("player_down") and pos.y<(screen.y-margin):
```

```
            vel += Vector2(0,speed*delta)
```

```
        if Input.is_action_pressed("player_fire"):
```

```
            shot()
```

```
        move(vel)
```

```
        get_node("engine").update_engine(vel,delta)
```

```
        laser_interval += delta
```

```
    if state=="exploding":
```

```
        close_interval -= delta
```

```
        if close_interval <=0:
```

```
            get_tree().change_scene("res://src/mainscreens/menu.tscn")
```

Exportar

2D

3D

Script

AssetLib



menu.tscn X

Archivo Buscar Depurar Ayuda



Tutoriales



Clases



Ayuda de Búsqueda

- controls.gd
- enemy.gd
- engine.gd
- explosion.gd
- global.gd
- laser.gd
- level1.gd
- menu.gd
- player.gd
- shot.gd
- star.gd
- tattoine.gd
- SamplePlayer
- Sprite

Clase: Sprite

Hereda: [Node2D](#) , [CanvasItem](#) , [Node](#) , [Object](#)

Descripción Breve:

General purpose Sprite node.

Métodos Públicos:

```
int    get_frame() const
int    get_hframes() const
Color  get_modulate() const
Vector2 get_offset() const
Rect2  get_region_rect() const
Texture get_texture() const
int    get_vframes() const
bool   is_centered() const
bool   is_flipped_h() const
bool   is_flipped_v() const
bool   is_region() const
void   set_centered( bool centered )
void   set_flip_h( bool flip_h )
void   set_flip_v( bool flip_v )
void   set_frame( int frame )
```



stable

Search docs

TUTORIALS

- ⊕ Learning step by step
- ⊕ Engine
- ⊕ 2D tutorials
- ⊕ 3D tutorials
- ⊕ Networking
- ⊕ Editor plugins
- ⊕ Miscellaneous
- ⊕ Asset pipeline

[Docs](#) » Documentation for Godot Engine 2.1.x (stable)

Documentation for Godot Engine 2.1.x (stable)

Looking for the documentation of the current **development** branch?

Note: Godot Engine is an open source project developed by a community. If you find a mistake in the documentation, the documentation team can always use your feedback and help to improve it. If you have a reference. So if you don't manage to understand something, or can't find what you need in the docs, help us make the documentation better and let us know! Submit a [pull request](#) to the [repository](#), or annoy us at the #godotengine-devel IRC channel!

The main documentation for the site is organized into a few sections:

- [Tutorials](#)
- [Reference](#)
- [Community](#)

Tutorials

- [Learning step by step](#)
- [Engine](#)
- [2D tutorials](#)

Para terminar...

- Fichero ejecutable para nuestro SO.
- Demos y ejemplos para seguir aprendiendo.
- Templates de exportación para varios sistemas.

Ajustes de Exportación del Proyecto

Objetivo

Recursos

Imágenes

Grupos de Imágenes

Exportar a Plataforma



Android



BlackBerry 10



HTML5



Linux X11



Mac OSX



Windows Desktop

Opciones

Debug

Debugging Enabled

Custom Binary

Debug

Release

Resources

Pack Mode

Bundle Dependencies

Binary

64 Bits

¡Muchas gracias!

- Twitter: @dvarrui
- Email: dvarrui@gmail.com
- ¿Preguntas?