
MONEY

Survival Horror · Godot Engine · GDScript · Blender

Marcos Ruiz Borrega · Yasser Belouafi Bakhat

¿Qué es MONEY?

Videjuego de terror en primera persona hecho con Godot Engine y GDScript.

El jugador vuelve a su antigua casa a recuperar el dinero de su herencia guardado en la caja fuerte.

01

Encuentra el código

Explora las habitaciones buscando el código

02

Abre la caja y recoge el dinero

Baja al sótano, introduce el código

03

Encuentra la llave

Localiza la llave de salida

!

Evita al monstruo

Patrulla y te perseguirá si te detecta



INTERIOR — LUCES APAGADAS, SOLO LINTERNA



EXTERIOR — DE NOCHE

¿Por qué este juego?

INI

Idea original — Roblox Studio

Juego de océano con barcos y jefes. Entorno conocido.
Al mes: obligatorio usar Godot.

02

Segundo intento — Godot + océano

Misma idea trasladada a Godot. Colisiones del agua,
física de navegación y optimización: inviable. Tiempo perdido.

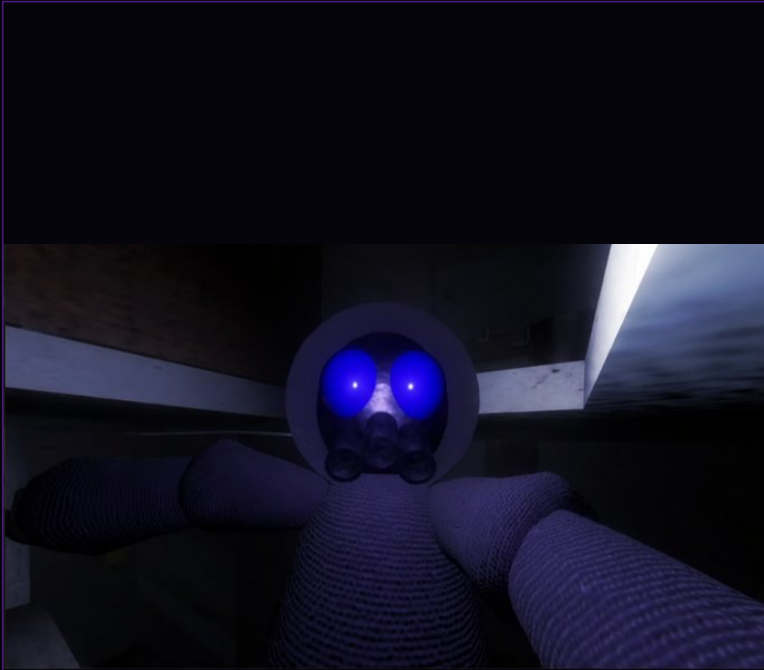
✓

Decisión final — MONEY

Terror en interiores. Un nivel bien construido,
mecánicas claras. Ejecutable en el tiempo disponible.

*En desarrollo, saber cuándo pivotar es tan importante como saber programar.
Reconocer que algo no funciona a tiempo es también una habilidad.*

IA del enemigo — Raycasts



EL MONSTRUO — MODELADO EN BLENDER

5 raycasts detectan al jugador en abanico.

Persecución: máximo 15 segundos.

Si no atrapa → vuelve a patrulla aleatoria.

```
// enemy.gd - detección con 5 raycasts

# _physics_process: comprueba los 5 raycasts
func _physics_process(_delta):
    chase_player($chasecast)
    chase_player($chasecast2)
    chase_player($chasecast3)
    chase_player($chasecast4)
    chase_player($chasecast5)

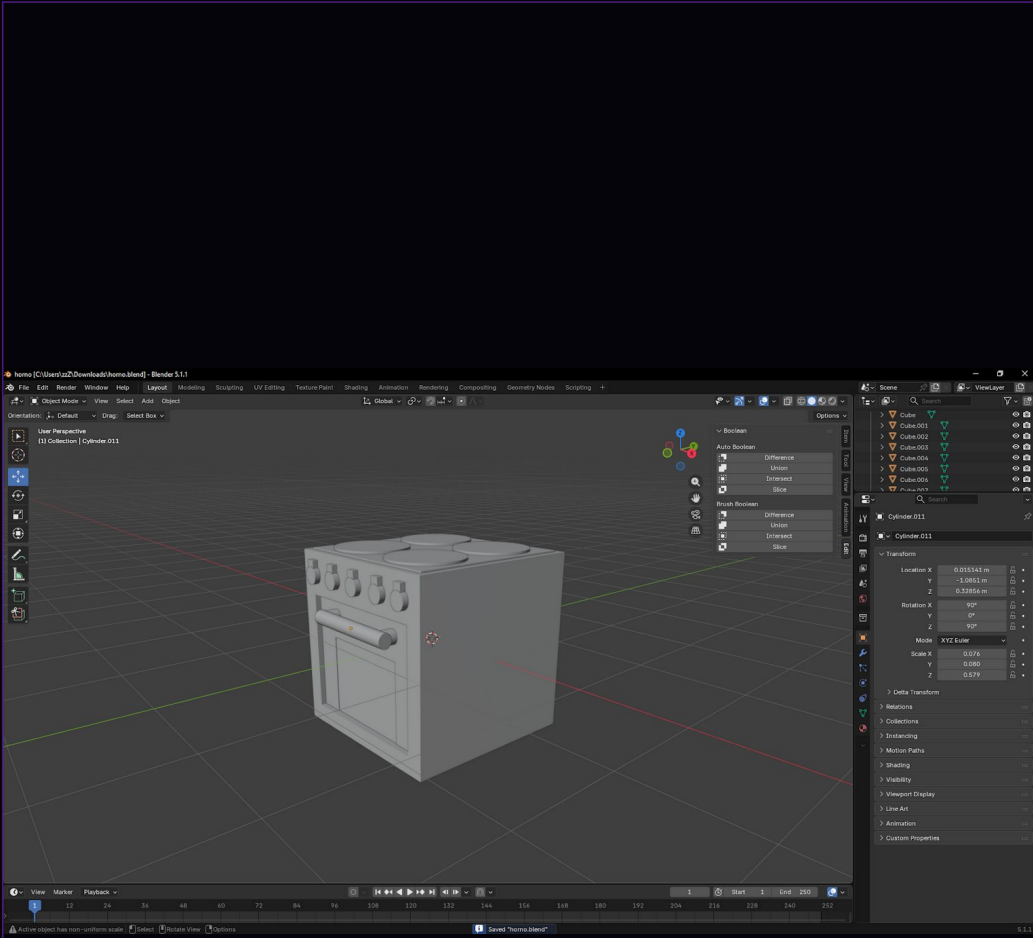
# Si el raycast toca al jugador → persecución
func chase_player(cast: RayCast3D):
    if cast.is_colliding():
        var hit = cast.get_collider()
        if hit.name == "player":
            chasing = true
            destination = player

# Música + velocidad reactivos al estado
if chasing:
    chase_music.play()
    ambient_music.stop()
    speed = 2.5
    chase_timer += delta
    if chase_timer >= 15.0:
        chasing = false
        pick_destination()

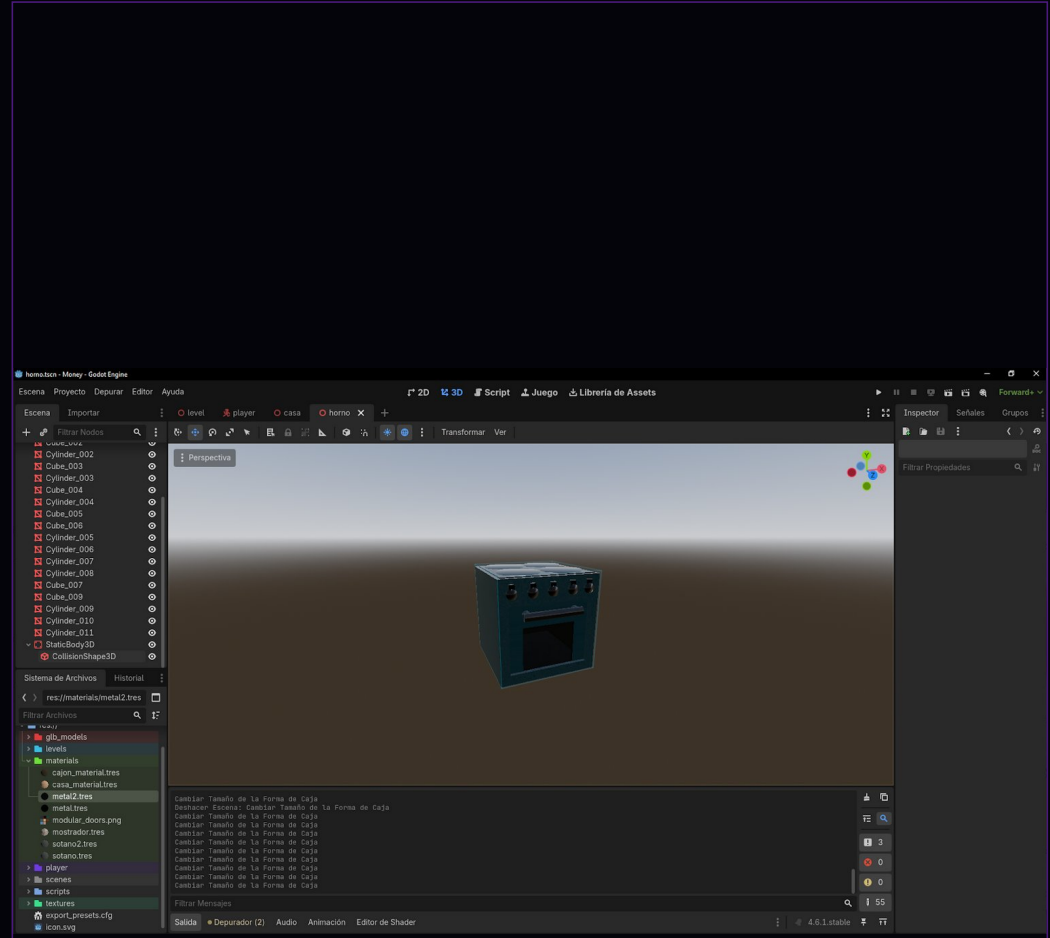
# Killcast: si toca → jumpscare + muerte
func kill_player():
    $killcast.look_at(player.global_transform.origin)
    if $killcast/killcast.is_colliding():
        if hit.name == "player":
            $jumpscare_cam.current = true
            await get_tree().create_timer(2.0).timeout
            get_tree().change_scene_to_file("muerte.tscn")
```

Blender → Godot

Modelado Blender → Export .GLB → Import Godot → Colisiones → Materiales → Escena lista



HORNO MODELADO EN BLENDER — OPERACIONES BOOLEANAS



MISMO MODELO IMPORTADO EN GODOT CON MATERIALES

Cajones: modelo + animación + script

```
// cajon.tscn - animación "open" en AnimationPlayer

# Dos keyframes de posición: cerrado → abierto
[sub_resource type="Animation"]
resource_name = "open"
length = 0.5
tracks/0/path = NodePath(".:position")
tracks/0/interp = 2 # interpolación cúbica (suave)
keys = {
    times: [0, 0.467],
    values: [
        Vector3(0, 0, 0), # cerrado
        Vector3(1.45, 0, 0) # abierto (1.45u en X)
    ]
}
```

```
-----
// door.gd - lógica de interacción
```

```
var open = false

func interact():
    open = !open
    if open:
        $AnimationPlayer.play("open")
        $open.play() # AudioStreamPlayer3D
    else:
        $AnimationPlayer.play_backwards("open")
        $close.play() # AudioStreamPlayer3D

# AudioStreamPlayer3D: radio 30u · bus SFX
```

BLD

Modelado Blender

Cuerpo + cajones + tiradores cilíndricos

GLB

Export → Import

Exportado .glb, importado en Godot con colisiones

ANI

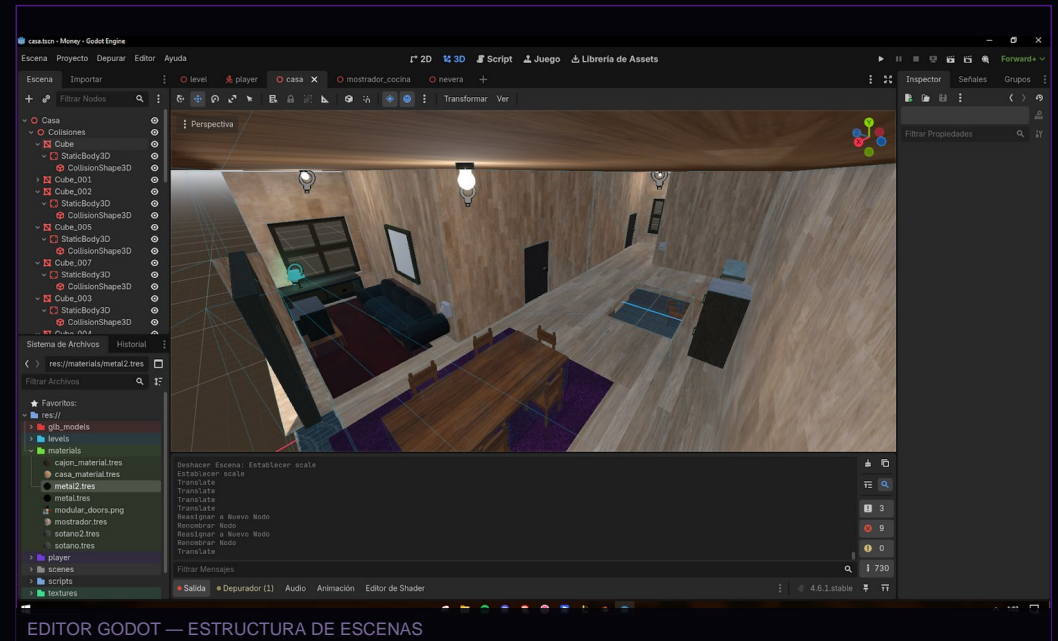
AnimationPlayer

Keyframes de posición con interpolación cúbica

GD

Script + Audio 3D

door.gd gestiona estado + AudioStreamPlayer3D



Movimiento y sigilo

```
// player.gd - crouch con lerp + pasos reactivos

# Agacharse: lerp suave en la cápsula de colisión
if crouching and shape.height > 0.25:
    shape.height = lerp(shape.height, 0.25, 0.2)
if !crouching and shape.height < 2.0:
    shape.height = lerp(shape.height, 2.0, 0.2)

# Velocidad según estado
if crouching:    SPEED = 1.25 # modo sigilo
if !crouching:  SPEED = 3.5  # velocidad normal

# Pasos reactivos al material del suelo
func footsteps(sounds, min_r, max_r):
    if !$feet.playing:
        $feet.pitch_scale =
            rng.randf_range(min_r, max_r)
        $feet.stream = sounds[
            rng.randi_range(0, sounds.size()-1)
        ]
        $feet.play()

if floor_type == 0: # madera → pitch 0.8-1.1
    footsteps(footstep_sounds, 0.8, 1.1)
elif floor_type == 1: # moqueta → pitch 1.3-1.6
    footsteps(grass_sounds, 1.3, 1.6)
```

■ Crouch dinámico con lerp

La cápsula de colisión se reduce suavemente. Necesario para entrar al sótano.

■ Pasos por material

Sonidos distintos madera/moqueta. Pitch aleatorio cada paso.

■ Headbob animado

AnimationPlayer en cámara. Acelera al moverse, se detiene al parar.

■ Linterna + interruptores

Linterna siempre activa. Interruptores encienden luces del nivel.

■ Sistema de muerte

Al morir: desactiva jugador, espera 2s, cambia a escena de muerte.

Sistemas del juego



Audio reactivo

3 buses: ambientación, persecución y SFX.
La música cambia con el estado del enemigo.
Efectos posicionales en 3D.

AudioStreamPlayer3D



Caja fuerte

UI con teclado numérico para introducir
el código encontrado explorando.
Animación y sonido al abrirse.

caja_fuerte_ui.tscn



Sistema de tareas

HUD que muestra el objetivo activo.
Se actualiza automáticamente al completar
cada fase del juego.

task_trigger.gd



Menú de pausa

Accesible en cualquier momento.
Ajustes de audio, vídeo y controles
implementados con settings.gd.

pause_menu.tscn



Intro + créditos

Secuencia de introducción animada.
Pantalla de muerte y créditos
al completar el juego.

game_intro + credits



Pathfinding

NavigationAgent3D gestiona movimiento
del enemigo. Patrulla aleatoria entre
nodos exportados.

NavigationAgent3D

Conclusión

◆ Lo que hemos aprendido

Godot Engine y GDScript desde cero. Pipeline Blender → Godot.
Colisiones, iluminación, NavigationAgent3D, AnimationPlayer,
audio reactivo. Gestionar un proyecto con cambios de dirección.

◆ De lo que estamos satisfechos

El juego es jugable y completo. Modelos 3D originales.
Sistema de sonido da atmósfera real.
IA del enemigo con raycasts genera tensión auténtica.

◆ Lo que mejoraríamos

Código más limpio y organizado.
Resolución de bugs menores pendientes.
Más niveles. Modo multijugador tipo speedrun.

◆ Aprendizaje clave

El desarrollo no es lineal. Adaptarse,
decidir bajo presión y sacar algo complejo desde cero
es la habilidad más valiosa que nos llevamos.

GRACIAS

Pasamos a la demo del juego

github.com/awpzz/money

Marcos Ruiz Borrega · Yasser Belouafi Bakhat