DOCUMENTACIÓN

PROYECTO GODOT

(Your First 3D Game)

Nombre: Diego Flores

Asignatura: Desarrollo de Videojuegos

Fecha: 05-11-2024

1. Creación de escena “Player”

Player.tscn:

En esta escena agregamos los siguientes nodos y los configuramos



player.gd (Script):

extends CharacterBody3D

signal hit

@export var speed = 14

@export var jump\_impulse = 20

@export var bounce\_impulse = 16

@export var fall\_acceleration = 75

func \_physics\_process(delta):

var direction = Vector3.ZERO

if Input.is\_action\_pressed("move\_right"):

direction.x += 1

if Input.is\_action\_pressed("move\_left"):

direction.x -= 1

if Input.is\_action\_pressed("move\_back"):

direction.z += 1

if Input.is\_action\_pressed("move\_forward"):

direction.z -= 1

if direction != Vector3.ZERO:

direction = direction.normalized()

basis = Basis.looking\_at(direction)

$AnimationPlayer.speed\_scale = 4

else:

$AnimationPlayer.speed\_scale = 1

velocity.x = direction.x \* speed

velocity.z = direction.z \* speed

if is\_on\_floor() and Input.is\_action\_just\_pressed("jump"):

velocity.y += jump\_impulse

velocity.y -= fall\_acceleration \* delta

move\_and\_slide()

for index in range(get\_slide\_collision\_count()):

var collision = get\_slide\_collision(index)

if collision.get\_collider().is\_in\_group("mob"):

var mob = collision.get\_collider()

if Vector3.UP.dot(collision.get\_normal()) > 0.1:

mob.squash()

velocity.y = bounce\_impulse

break

rotation.x = PI / 6 \* velocity.y / jump\_impulse

func die():

hit.emit()

queue\_free()

func \_on\_MobDetector\_body\_entered(\_body):

die()Creación de escena “Mob”

Mob.tscn:

En esta escena agregamos los siguientes nodos (La configuración de nodos es similar a la de player) y los configuramos.

Texto

Descripción generada automáticamente con confianza baja

Mob.gd (Script):

extends CharacterBody3D

signal squashed

@export var min\_speed = 10

@export var max\_speed = 18

func \_physics\_process(\_delta):

move\_and\_slide()

func initialize(start\_position, player\_position):

look\_at\_from\_position(start\_position, player\_position, Vector3.UP)

rotate\_y(randf\_range(-PI / 4, PI / 4))

var random\_speed = randf\_range(min\_speed, max\_speed)

velocity = Vector3.FORWARD \* random\_speed

velocity = velocity.rotated(Vector3.UP, rotation.y)

$AnimationPlayer.speed\_scale = random\_speed / min\_speed

func squash():

squashed.emit()

queue\_free()

func \_on\_visible\_on\_screen\_notifier\_screen\_exited():

queue\_free()

1. Creación de escena “Main”

main.tscn:

En esta escena agregamos los siguientes nodos y los configuramos. Hay que instanciar la escena de Player.tscn

Texto

Descripción generada automáticamente con confianza media

Main.gd (Script):

extends Node

@export var mob\_scene: PackedScene

func \_ready():

$UserInterface/Retry.hide()

func \_unhandled\_input(event):

if event.is\_action\_pressed("ui\_accept") and $UserInterface/Retry.visible:

get\_tree().reload\_current\_scene()

func \_on\_mob\_timer\_timeout():

var mob = mob\_scene.instantiate()

var mob\_spawn\_location = get\_node("SpawnPath/SpawnLocation")

mob\_spawn\_location.progress\_ratio = randf()

var player\_position = $Player.position

mob.initialize(mob\_spawn\_location.position, player\_position)

add\_child(mob)

mob.squashed.connect($UserInterface/ScoreLabel.\_on\_Mob\_squashed)

func \_on\_player\_hit():

$MobTimer.stop()

$UserInterface/Retry.show()

Por ultimo hay que definir el sript de score

ScoreLabel.gd

extends Label

var score = 0

func \_on\_Mob\_squashed():

score += 1

text = "Score: %s" % score