Лабораториска задача 3

Правење на видео игра во Godot 4

изработено од Горазд Бишкоски 223145

**Како да играте:**

Ова е 3D игра во која играчот треба да собере 10 парички за да победи. Имате 30 секунди за да ги соберете сите парички. Секогаш кога ќе соберете паричка, таа исчезнува. Ако ги соберете сите 10 парички пред времето да истече, победувате. Ако времето истече пред да ги соберете сите, играта завршува и во горе левиот агол од играта му се прикажува текст на играчот во кој му пишува дека тој ја изгубил играта.

**Контроли:**

* **W, A, S, D** – Движење на играчот
* **Space** – Скок
* **Глушец** – Ротирање на камерата

**Имплементација на кодот:**

**1. Парички (coin.gd)**

* Паричките се базирани на инстанца Area3D, што на кратко значи дека можат да детектира кога играчот влегува во нив. Ова ни овозможува воведување на бројач за колку парички има собрано играчот во текот на играта.
* Кога играчот ќе допре паричка, таа повикува функција во главниот скрипт и се брише (queue\_free()).
* queue\_free е вградена функција која го зема моделот (во овој случај паричката) и ја сместува во редица на чекање за да се „ослободи“ од меморијата, односно се брише од играта. (името queue free означува ослободување од меморијата за играта да не мора да го рендерира моделот повторно).
* функцијата \_process е функција која ја има секој модел во играта и се повикува секој frame од играта. Во оваа функција го „анимираме“ ротирањето на паричката.
* deg\_to\_rad() функцијата ја повикуваме при повик на rorate\_y() поради тоа што играта Godot прима ротации во радијани наспроти степени. Ние испраќаме степени за полесно интерпретирање на кодот.

**-------------------------------------------------------------------------------------------------------------------**

extends Area3D

@onready var main = get\_tree().get\_first\_node\_in\_group("main")

const ROTATION\_SPEED := 2

func \_process(delta: float) -> void:

rotate\_y(deg\_to\_rad(ROTATION\_SPEED))

func \_on\_body\_entered(body: Node3D) -> void:

main.on\_coin\_collected()

queue\_free()

**-------------------------------------------------------------------------------------------------------------------**

**2. Главен скрипт (main.gd)**

* Овaa скриптa ги следи собраните парички.
* Кога играчот собрал 10 парички, повикува функцијата game\_won() фунцкијата од скриптата за лабела која е линкувана и таа ја паузира играта и прикажува порака за успешно завршување на играта.

**3. Тајмер и HUD (LabelScript.gd)**

* Одбројува времето и го прикажува на екранот.
* Кога времето истекува, играта се паузира и се прикажува пораката "Game Over".
* Ако играчот ги собере сите парички, се прикажува пораката "Congrats, You won!" и играта се стопира.

-------------------------------------------------------------------------------------------------------------------

extends Label

@export var countdown\_time : int = 30

var time\_left : int = countdown\_time

var timer : Timer

func \_ready() -> void:

Input.set\_mouse\_mode(Input.MOUSE\_MODE\_CAPTURED)

text = str(time\_left) + "s"

timer = Timer.new()

add\_child(timer)

timer.wait\_time = 1

timer.autostart = true

timer.timeout.connect(\_on\_Timer\_timeout)

timer.start()

func \_on\_Timer\_timeout():

time\_left -= 1

text = str(time\_left) + "s"

if time\_left <= 0:

freeze\_game(false)

func freeze\_game(won: bool):

get\_tree().paused = true

Input.set\_mouse\_mode(Input.MOUSE\_MODE\_VISIBLE)

if won:

text = "Congratulations\nYou won!"

else:

text = "Game Over\nTry again!"

horizontal\_alignment = HORIZONTAL\_ALIGNMENT\_CENTER

vertical\_alignment = VERTICAL\_ALIGNMENT\_CENTER

**-------------------------------------------------------------------------------------------------------------------**

* Овој код го одбројува времето и се грижи за логиката на крајот на играта.
* freeze\_game(won: bool) ја паузира играта и ја менува пораката во зависност од тоа дали играчот победил.

**Цел на играта:**

Собери 10 парички пред времето да истече за да победиш!