گزارش فنی پروژه بازی‌سازی

پیاده‌سازی کلون بازی Man Pepsi

نام دانشجو:
سید عرفان مسعودی

استاد درس:
استاد مینائی

دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
۱۶ دی ۱۴۰۴

۱ مقدمه

در این پروژه، هدف طراحی و توسعه یک بازی در سبک Endless Runner مشابه بازی نوستالژیک Pepsi Man است. اگرچه در صورت تمرین استفاده از موتور Unity پیشنهاد شده بود، اما در این پیاده‌سازی از موتور قدرتمند و متن‌باز Godot Engine 4 و زبان برنامه‌نویسی GDScript استفاده شده است. تمام مکانیک‌های خواسته شده شامل حرکت سه لاینه، تولید مسیر پویا، سیستم امتیازدهی و ذخیره‌سازی رکورد به طور کامل پیاده‌سازی شده‌اند.

۲ توضیحات کلی بازی

بازی Pepsi Man Runner یک بازی سه بعدی از نوع Endless Runner است که با موتور Godot 4 و زبان GDScript پیاده‌سازی شده است. این بازی دارای مکانیک‌های اصلی زیر است:

- حرکت خودکار شخصیت اصلی به سمت جلو
- امکان جابجایی بین سه لاین (چپ، وسط، راست)
- تولید پویای مسیر و موانع
- سیستم امتیازدهی
- سیستم ذخیره‌سازی رکورد
- امکان ریستارت بازی

۳ ساختار بازی

بازی دارای چندین سین (Scene) اصلی است:

۱.۳ Scene Main

این صحنه اصلی بازی است که شامل موارد زیر می‌شود:

- یک نور جهت‌دار (Directional Light)
- محیط جهان (World Environment)
- زمین (Ground) که با یک CSGBox3D ایجاد شده است
- شخصیت بازیکن (Player) که از CharacterBody3D ارث می‌برد
- یک تکه جاده اولیه
- یک نمونه LevelGenerator برای تولید پویای مسیر
- لایه UI شامل نمایش امتیاز، بهترین امتیاز و پنل Game Over

۲.۳ Scene Player

صحنه شخصیت بازیکن که شامل یک CharacterBody3D با مشخصات زیر است:

- یک MeshInstance3D با شکل کپسول
- یک CollisionShape3D با شکل کپسول برای تشخیص تصادف
- یک دوربین (Camera۳D) برای دید بازیکن

۳.۳ Scene RoadTile

یک تکه از جاده که شامل یک BoxMesh با ابعاد ۱۰×۱.۰×۲۰ متر است. این صحنه دارای یک EndConnector در انتهای جاده است که برای تولید پویا مفید است.

۴.۳ Scene Obstacle

صحنه مانع که یک Area3D است و شامل یک BoxMesh و یک CollisionShape3D است. این مانع با ورود شخصیت بازیکن به محدوده خود، بازی را پایان می‌دهد.

۴ اسکریپت‌های اصلی

۱.۴ Player.gd

این اسکریپت کنترل شخصیت بازیکن است و شامل موارد زیر است:

- متغیرهای FORWARD_SPEED (۰.۱۰) برای سرعت حرکت به جلو
- متغیر LANE_DISTANCE (۰.۳) برای فاصله بین لاین‌ها
- متغیر LERP_SPEED (۰.۱۰) برای سرعت جابجایی نرم بین لاین‌ها
- متغیر current_lane که مقدار ۰ برای وسط، -۱ برای چپ و ۱ برای راست است
- تابع physics_process که شامل سه بخش است:
  - مدیریت ورودی برای تغییر لاین (کلیدهای چپ و راست)
  - حرکت خودکار به سمت جلو
  - حرکت نرم به سمت لاین هدف با استفاده از Lerp

۲.۴ Obstacle.gd

این اسکریپت برای تشخیص برخورد بازیکن با مانع استفاده می‌شود:

- زمانی که یک body وارد محدوده می‌شود، چک می‌کند آیا نام آن Player است یا خیر
- اگر بازیکن بود، پیام Game Over را چاپ می‌کند
- تابع ()game_over را در نود اصلی فراخوانی می‌کند

۳.۴ Main.gd

اسکریپت نود اصلی بازی که مسئولیت‌های زیر را دارد:

- نمایش امتیاز و بهترین امتیاز در UI
- مدیریت وضعیت Game Over
- ذخیره و بارگذاری بهترین امتیاز در فایل
- ریستارت بازی

این اسکریپت شامل توابع زیر است:

- update_ui: به‌روزرسانی نمایش امتیاز
- game_over: مدیریت وضعیت باخت
- restart_game: ریستارت بازی
- save_high_score: ذخیره رکورد در فایل
- load_high_score: بارگذاری رکورد از فایل

۴.۴ LevelGenerator.gd

این اسکریپت برای تولید پویای جاده استفاده می‌شود:

- ۱۰ تکه جاده در ابتدا ایجاد می‌شود
- زمانی که بازیکن از یک تکه جاده فراتر می‌رود، آن تکه حذف و یک تکه جدید در انتهای مسیر ایجاد می‌شود
- از یک لیست برای مدیریت تکه‌های جاده فعال استفاده می‌کند

۵.۴ RoadTile.gd

اسکریپت یک تکه جاده که شامل تابع spawn_obstacle است:

- یک عدد تصادفی بین ۰ تا ۲ انتخاب می‌کند
- عدد انتخابی را به ۱-، ۰ یا ۱ تبدیل می‌کند (برای سه لاین)
- یک نمونه از مانع ایجاد می‌کند و آن را در موقعیت تصادفی بر روی جاده قرار می‌دهد

۵ مکانیک‌های بازی

۱.۵ حرکت ۳ لاینی

بازیکن می‌تواند بین سه لاین حرکت کند: چپ X) = ۳-)، وسط X) = ۰) و راست X) = ۳). این کار با فشردن کلیدهای چپ و راست انجام می‌شود. حرکت بین لاین‌ها نرم است و با استفاده از Lerp انجام می‌شود.

۲.۵ تولید پویای مسیر

بازی از یک سیستم تولید پویا برای ایجاد مسیر استفاده می‌کند. هر تکه جاده ۲۰ متر طول دارد و زمانی که بازیکن از یک تکه فراتر می‌رود، آن تکه حذف و یک تکه جدید در انتهای مسیر ایجاد می‌شود.

۳.۵ سیستم امتیازدهی

امتیاز بازیکن در هر فریم افزایش می‌یابد و بهترین امتیاز در فایل user://savegame.cfg ذخیره می‌شود. این فایل در پوشه مخصوص کاربر در سیستم عامل ذخیره می‌شود.

۴.۵ سیستم برخورد

زمانی که شخصیت بازیکن با مانع برخورد می‌کند، تابع game_over() فراخوانی می‌شود که بازی را پایان داده و پنل Game Over را نمایش می‌دهد.

۶ کنترل‌ها

- کلید چپ (Left Arrow): حرکت به لاین چپ
- کلید راست (Right Arrow): حرکت به لاین راست
- دکمه Restart: ریستارت بازی

۷ نتیجه‌گیری

بازی Pepsi Man Runner یک نمونه خوب از یک بازی Endless Runner است که تمام مکانیک‌های اصلی مورد نیاز را پیاده‌سازی کرده است. این پروژه نشان می‌دهد که چگونه می‌توان با استفاده از موتور Godot و زبان GDScript یک بازی جذاب و کاربردی ساخت. معماری کد نیز خوب ساختاریافته است و هر اسکریپت مسئولیت خاصی را بر عهده دارد.

۸ توضیحات گام به گام کدها

۱.۸ Player.gd

```
extends CharacterBody3D

const FORWARD_SPEED = 10.0   \textlr{\# سرعت حرکت رو به جلو}
const LANE_DISTANCE = 3.0    \textlr{\# فاصله بین لاین‌ها}
const LERP_SPEED = 10.0      \textlr{\# سرعت نرم جابجایی بین لاین‌ها}

\textlr{\# 0 = وسط، -1 = چپ، 1 = راست}
var current_lane = 0
var target_x = 0.0

func _ready():
    \textlr{\# تنظیم لاین اولیه}
    target_x = 0.0

func _physics_process(delta):
    \textlr{\# 1. تغییر لاین برای ورودی مدیریت}
    if Input.is_action_just_pressed("ui_left"):
        if current_lane > -1:
            current_lane -= 1

    if Input.is_action_just_pressed("ui_right"):
        if current_lane < 1:
            current_lane += 1

    \textlr{\# محاسبه موقعیت هدف در محور X بر اساس لاین انتخاب شده}
    target_x = current_lane * LANE_DISTANCE

    \textlr{\# 2. حرکت خودکار در جلو به رو (جلو یعنی منفی Z)}
```

```
    velocity.z = -FORWARD_SPEED

    \textlr{\# 3. هدف لایین سمت به نرم حرکت (Interpolation)}
    velocity.x = (target_x - position.x) * LERP_SPEED

    \textlr{\# اعمال حرکت}
    move_and_slide()
```

Listing 1: بازیکن شخصیت اسکریپت

توضیحات:

۱. ابتدا کلاس CharacterBody3D اکستند می‌شود که یک نود فیزیکی است که می‌تواند حرکت کند و با محیط تعامل داشته باشد.

۲. سه ثابت تعریف می‌شوند: FORWARD_SPEED برای سرعت حرکت به جلو، LANE_DISTANCE برای فاصله بین لاین‌ها و LERP_SPEED برای سرعت جابجایی نرم بین لاین‌ها.

۳. متغیر current_lane نشان‌دهنده لاین فعلی بازیکن است (۰ برای وسط، -۱ برای چپ، ۱ برای راست).

۴. در تابع _ready، موقعیت اولیه هدف روی ۰ تنظیم می‌شود.

۵. در تابع _physics_process، ابتدا ورودی‌های کلیدهای چپ و راست بررسی می‌شوند.

۶. اگر کلید چپ فشرده شود و بازیکن در لاین انتهایی چپ نباشد، یک لاین به سمت چپ حرکت می‌کند.

۷. اگر کلید راست فشرده شود و بازیکن در لاین انتهایی راست نباشد، یک لاین به سمت راست حرکت می‌کند.

۸. موقعیت هدف در محور X بر اساس لاین فعلی محاسبه می‌شود.

۹. سرعت در جهت Z (به جلو) تنظیم می‌شود تا شخصیت به طور خودکار حرکت کند.

۱۰. سرعت در جهت X برای حرکت نرم به سمت لاین هدف محاسبه می‌شود.

۱۱. در نهایت تابع move_and_slide فراخوانی می‌شود تا حرکت اعمال شود.

۲.۸ Obstacle.gd

```
extends Area3D

func _on_body_entered(body):
    \textlr{\# نه یا است بازیکن شده وارد که چیزی می‌کنیم بررسی}
    if body.name == "Player":
        print("Game Over!")

        \textlr{\# باخت تابع زدن صدا و Main نود کردن پیدا}
        \textlr{\# چون Obstacle فرزند RoadTile و RoadTile فرزند
    LevelGenerator ،است}
        \textlr{\# می‌کنیم استفاده زیر روش از .است سخت میتوقیم دسترسی}
        var main = get_tree().current_scene
        if main.has_method("game_over"):
            main.game_over()
```

Listing 2: مانع اسکریپت

توضیحات:

۱. این اسکریپت از کلاس Area3D اکستند می‌شود که برای تشخیص تصادف بدون فیزیک استفاده می‌شود.

۲. تابع _on_body_entered زمانی فراخوانی می‌شود که یک نود با کلاینت فیزیکی وارد محدوده مانع شود.

۳. ابتدا بررسی می‌شود که آیا نام نود وارد شده Player است یا خیر.

۴. اگر بازیکن بود، پیام Game Over چاپ می‌شود.

۵. سپس نود اصلی صحنه با استفاده از get_tree().current_scene پیدا می‌شود.

۶. اگر نود اصلی تابع game_over را داشت، آن تابع فراخوانی می‌شود.

Main.gd ۳.۸

```
extends Node3D

\textlr{\# نوده‌ای به دسترسی UI}
@onready var score_label = $UI/ScoreLabel
@onready var high_score_label = $UI/HighScoreLabel
@onready var game_over_panel = $UI/GameOverPanel
@onready var player = $Player

var score = 0
var high_score = 0
var is_game_over = false

\textlr{\# ری‌سم فایل در ذخیره‌سازی مخصوص پوشه کاربر در متن‌سی عامل ذخیره می‌شود)}
const SAVE_PATH = "user://savegame.cfg"

func _ready():
    \textlr{\# ری‌استارت دکمه کردن وصل}
    $UI/GameOverPanel/RestartButton.pressed.connect(restart_game)

    \textlr{\# فایل از رکورد بهترین کردن لود}
    load_high_score()

    \textlr{\# اولیه نمایش}
    update_ui()

func _process(delta):
    if is_game_over:
        return

    \textlr{\# امتیاز افزایش}
    score += 1

    \textlr{\# شود آپدیت همزمان های‌اسکور شکست‌تم، را رکورد اگر}
    if score > high_score:
        high_score = score

    \textlr{\# متن‌ها به‌روزرسانی}
    update_ui()

func update_ui():
    score_label.text = "Score: " + str(score)
    high_score_label.text = "Best: " + str(high_score)

```

```gdscript
func game_over():
    if is_game_over: return
    is_game_over = true

    \textlr{\# ذخیره رکورد جدید اگر( تغییرات کرده باشد)}
    save_high_score()

    game_over_panel.visible = true
    player.set_physics_process(false)

func restart_game():
    get_tree().reload_current_scene()

\textlr{\# --- توابع ذخیره و لود تنظیمات( PlayerPrefs) ---}

func save_high_score():
    var config = ConfigFile.new()
    \textlr{\# مقدار high_score را با کلید "best_score" در بخش "game" ذخیره
    می‌کنیم}
    config.set_value("game", "best_score", high_score)
    config.save(SAVE_PATH)

func load_high_score():
    var config = ConfigFile.new()
    \textlr{\# تلاش برای باز کردن فایل}
    var err = config.load(SAVE_PATH)

    \textlr{\# اگر فایل با موفقیت باز شد}
    if err == OK:
        \textlr{\# اگر( بود، پیش‌فرض 0 برگرداند) مقدار را می‌خواند}
        high_score = config.get_value("game", "best_score", 0)
    else:
        \textlr{\# اگر( بازی اجرای اول بار نبود) فایلی
        high_score = 0
```

Listing 3: اسکریپت نود اصلی بازی

توضیحات:

۱. این اسکریپت از کلاس Node3D اکستند می‌شود و نود اصلی بازی است.

۲. چهار متغیر @onready تعریف شده‌اند که بعد از لود صحنه مقداردهی می‌شوند: score_label، high_score_label، player و game_over_panel.

۳. متغیرهای score، high_score و is_game_over برای مدیریت وضعیت بازی استفاده می‌شوند.

۴. ثابت SAVE_PATH مسیر فایل ذخیره‌سازی را تعریف می‌کند.

۵. در تابع _ready، دکمه ریستارت به تابع restart_game وصل می‌شود، بهترین امتیاز لود می‌شود و رابط کاربری به‌روزرسانی می‌شود.

۶. تابع _process در هر فریم اجرا می‌شود و اگر بازی تمام نشده باشد، امتیاز را افزایش می‌دهد.

۷. اگر امتیاز فعلی بیشتر از بهترین امتیاز باشد، بهترین امتیاز آپدیت می‌شود.

۸. تابع update_ui برچسب‌های امتیاز و بهترین امتیاز را به‌روز می‌کند.

۹. تابع game_over وضعیت باخت را فعال می‌کند، امتیاز را ذخیره می‌کند و پنل Game Over را نمایش می‌دهد.

۱۰. تابع restart_game صحنه فعلی را دوباره لود می‌کند.

۱۱. توابع save_high_score و load_high_score برای ذخیره و بارگذاری بهترین امتیاز در فایل استفاده می‌شوند.

۴.۸ LevelGenerator.gd

```
extends Node

\textlr{\# ی‌ای که باید در اینسپکتور مقداردهی شود}
@export var road_tile_scene: PackedScene
@export var player: Node3D

var spawn_z = 0.0       \textlr{\# موقعیت شروع ساخت جاده}
var tile_length = 20.0  \textlr{\# طول هر قطعه جاده}
var active_tiles = []   \textlr{\# لیستی برای نگه داشتن تمام قطعه‌های جاده فعال}

func _ready():
    \textlr{\# در شروع بازی، ۱۰ تکه جاده می‌سازیم تا جلو خالی نباشد}
    for i in range(10):
        spawn_tile()

func _process(delta):
    \textlr{\# اگر بازیکن وجود نداشت (مثلاً هنوز ساخته نشده) کاری نکن}
    if not player:
        return

    \textlr{\# بررسی ساده: اگر بازیکن از اولین جاده لیست عبور کرد}
    \textlr{\# (چون حرکت به سمت منفی Z است، یعنی مقدار کم‌تر می‌شود)}
    if active_tiles.size() > 0:
        var first_tile = active_tiles[0]
        if player.position.z < first_tile.position.z - tile_length:
            \textlr{\# جاده قدیمی را حذف کن}
            active_tiles.pop_front()  \textlr{\# حذف از لیست}
            first_tile.queue_free()   \textlr{\# حذف از بازی}

            \textlr{\# یک جاده جدید در انتها بساز}
            spawn_tile()

func spawn_tile():
    var tile = road_tile_scene.instantiate()
    add_child(tile)

    \textlr{\# موقعیت تنظیم}
    tile.position.z = spawn_z

    \textlr{\# موقعیت بعدی را ۲۰ متر جلوتر به (منفی) می‌بریم}
    spawn_z -= tile_length

    \textlr{\# به لیست اضافه می‌کنیم}
    active_tiles.append(tile)
```

Listing 4: اسکریپت تولیدکننده سطح

توضیحات:

۱. این اسکریپت از کلاس Node اکستند می‌شود و مسئول تولید پویای جاده است.

۲. دو متغیر @export تعریف شده‌اند که در اینسپکتور قابل تنظیم هستند: road_tile_scene و player.

۳. متغیرهای spawn_z، tile_length و active_tiles برای مدیریت موقعیت و لیست تکه‌های جاده استفاده می‌شوند.

۴. در تابع _ready، ۱۰ تکه جاده اولیه ایجاد می‌شوند.

۵. تابع _process در هر فریم چک می‌کند که آیا بازیکن از اولین تکه جاده عبور کرده است یا خیر.

۶. اگر عبور کرده بود، اولین تکه جاده از لیست حذف و از بازی آزاد می‌شود.

۷. سپس یک تکه جاده جدید در انتهای مسیر ایجاد می‌شود.

۸. تابع spawn_tile یک نمونه از صحنه جاده ایجاد می‌کند، موقعیت آن را تنظیم می‌کند و به لیست تکه‌های فعال اضافه می‌کند.

RoadTile.gd ۵.۸

```
extends Node3D

\textlr{\# بارگذاری صحنه مانع}
var obstacle_scene = preload("res://Scenes/Obstacle.tscn")

func _ready():
    spawn_obstacle()

func spawn_obstacle():
    \textlr{\# یک عدد تصادفی بین 0 تا 2 بخوان که نشان می‌دهد برای( 3 کدام)}
    \textlr{\# هاینال: -1 چپ()، 0 وسط()، 1 راست()}
    var random_index = randi() % 3   \textlr{\# انتخاب شهی 0 یا 1 یا 2}
    var lane = random_index - 1      \textlr{\# تبدیل شهی به -1 یا 0 یا 1}

    \textlr{\# ساخت نمونه از مانع}
    var obstacle = obstacle_scene.instantiate()
    add_child(obstacle)

    \textlr{\# تنظیم موقعیت مانع}
    \textlr{\# X: بر اساس کدام لاین قرار( 3 گیرد)}
    \textlr{\# Y: کمی از بالای زمین الش( 0.5)}
    \textlr{\# Z: در یک جای تصادفی روی طول جاده الش( بین -5 تا -15)}
    var random_z = randf_range(-5.0, -15.0)
    obstacle.position = Vector3(lane * 3.0, 0.5, random_z)
```

Listing 5: اسکریپت تکه جاده

توضیحات:

۱. این اسکریپت از کلاس Node3D اکستند می‌شود و یک تکه از جاده را مدیریت می‌کند.

۲. صحنه مانع با استفاده از preload بارگذاری می‌شود.

۳. در تابع _ready، یک مانع تصادفی ایجاد می‌شود.

۴. تابع spawn_obstacle یک عدد تصادفی بین ۰ تا ۲ تولید می‌کند.

۵. این عدد به ۱-، ۰ یا ۱ تبدیل می‌شود که نشان‌دهنده لاین چپ، وسط یا راست است.

۶. یک نمونه از مانع ایجاد می‌شود و به تکه جاده اضافه می‌شود.

۷. موقعیت مانع بر اساس لاین ،(X) ارتفاع کمی بالاتر از زمین (Y) و یک موقعیت تصادفی روی محور Z تنظیم می‌شود.