

# The Deadly Paradise

Shaghayegh Mobasher

Parisa Alaie



# خلاصه ی داستان بازی

- ▶ **Ace** جوان ترین و قوی ترین محافظ جزیره با گروهی از دشمنان تاریکی مبارزه می کند.
- ▶ **Ace** برای نجات جزیره به جمع آوری تکه های سنگ روی می آورد. او با گرفتن هر تکه از سنگ می تواند تخته جلوی قلمرو بعدی را شکسته و به آن وارد شود تا با شکست دادن **boss** آن قلمرو تکه دیگری از سنگ را به دست آورد.
- ▶ اما آیا او قادر است که همه ی تکه ها سنگ را بدست آورد؟ در این ماجراجویی **ace** را همراهی کنید تا از پایان ها ی متفاوت لذت ببرید.

# دیالوگ

- ▶ در این قسمت برای فهمیدن و یادگیری مراحل نیاز به دیالوگ هایی بود که کاراکترهای NPC از آن بریا ارتباط با player استفاده می کنند.
- ▶ نحوه ی ساختن دیالوگ ها : برای پیاده سازی این بخش نیاز به سه پیاده سازی متفاوت داشتیم.
- ▶ Dialogue: ابتدا یک اسکریپت برای دیالوگ ها داشتیم که اسم و جمله ها و اکشن هایی که بعد و قبل از دیالوگ انجام میشد را ذخیره کردیم (در هنگام دیالوگ بازی کننده نباید بتواند حرکت کند).
- ▶ Dialoguemanager: در این بخش متدهایی برای شروع دیالوگ، نحوه ی نشان دادن آن و چگونگی نشان دادن جمله ها و جلو زدن دیالوگ ها و چگونه به پایان رسیدن دیالوگ پیاده سازی شده است.
- ▶ بخش آخر مربوط به دیالوگ هایی است که هر فرد میگوید. آن ها را در ارایه میگذاریم و در بعد از آن ها استفاده می کنیم.

# Save and Load System

- ▶ در 4 بخش این عملیات انجام شده است. به دلیل اینکه یونیتی خود سیستمی برای save ندارد باید دستی همه چیزهایی که میخواستیم در یک save انجام شود را زدیم.
- ▶ saveData: در اینجا هر یک از مواردی که میخواهیم در یک save و load نیاز است را ذخیره کنیم مثل scene ای که در آن قرار داریم و مقدار HP ای که داریم و تعداد شمشیرهایی که باز کرده ایم و position ای که Ace داشته است و Camera position
- ▶ SaveSystem: در اینجا اطلاعات لازم را به اطلاعات باینری تبدیل میکنیم و در مسیری که بازی نصب شده است در فایلی با فرمتی که مشخص کرده ایم ذخیره میکنیم. در اینجا برای لود اطلاعات از فایل ذخیره شده اطلاعات باینری را تبدیل به اطلاعات اولیه میکنیم و در saveData ذخیره میکنیم تا استفاده کنیم. در اینجا ریست و خروج از بازی هم زده شده است که برپا ریست کردن باید اطلاعات و پوزیشن هایی که در اولین scene داشته ایم را دوبار نشان دهیم.
- ▶ gameStatus: در اینجا برای نشان دادن scene ای که در قبل بوده ایم برای تعویض به scene بعدی همراه با اطلاعاتی که داشته ایم استفاده کردیم
- ▶ SaveStatue: در هر مرحله از مجسمه ای خاص برای ذخیره کردن اطلاعات استفاده میشود که بعد از انجام عملیات save به ما دیالوگی نشان میدهد که نشان میدهد که اطلاعات با موفقیت ذخیره شده است و hp را به ماکس میرساند.
- ▶ scene\_transmissin\_controler: در این کد یک سری شروط ها برای اجازه به ادامه ی مسیر از scene ای به scene دیگر و نمایش scene بعدی پیاده سازی شده است

# Menu

- ▶ Main Menu: در اینجا فرد میتواند که انتخاب کند که به ادامه ی بازی برود یا بازی را ریست کند یا به قسمت option برود یا از بازی خارج شود.
- ▶ Options: در اینجا صدای بازی را میتوان کنترل کرد و میتوان سباز تصویر بازی را به fullScreen تبدیل کرد و به main menu در نهایت بازگشت.
- ▶ Resume Menu: در اینجا با زدن esc بازی باز میشود و سه گزینه برای فرد می آورد اینکه از بازی خارج شود یا به options برود یا به ادامه ی بازی پردازد
- ▶ Died Menu: هنگامی که جان ace به 0 می رسد باید بازی وارد منو died شود که این شانس را به فرد می دهد که دوباره امتحان کند یا به main menu باز گردد
- ▶ Good Ending: اگر بعد از بار اول که بازی به پایان رسیده است و ما از بای بدون save کردن خارج شویم و دوباره مرحله ی آخر را برویم با یک سری مرحله های مخفی مواجه میشویم که بعد از آن به این پایان که زمان صرف شده و دکمه ای برا ی بازگشت به main menu است را نشان می دهد
- ▶ Bad Ending: در اولین باری که بازی به پایان می رسد ما به این اندینگ میرسیم که با میتوانیم زمانی که برای بازی صرف کرده ایم را ببینیم و به main menu باز گردیم

# Ace (player)

- ▶ شخصیت اصلی داستان قابلیت راه رفتن در 4 جهت بالا و پایین و چپ و راست را دارد و در مرحله ی خاصی میتواند دو مدل پریدن داشته باشد
- ▶ این شخصیت 4 شمشیر مختلف دارد که هرکدام برای کشتن نوعی خاصی دشمن استفاده میشوند و نحوه ی عوض کردن شمشیر و نشان دادن شمشیر در این قسمت برای `playerController` نوشته شده است و همچنین در هنگام تلپورت جایی که باید در صحنه ی جدید شروع کند را نیز در این قسمت قرار داده ایم و عوض شدن سرعت شخصیت و تغییراتی `hp` و نحوه ی محاسبه ها و مقدار هایی که عوض میشود نیز در اینجا نوشته شده است.
- ▶ `Prefab` از آن درست شده است که در `scene` ها مختلف به راحتی از آن استفاده شود.

# Health Bar

- ▶ از کد نوشته شده برای این قسمت هم برای player و هم برای boss fight ها استفاده میشود.
- ▶ در این کد سه اتفاق رخ میدهید :
  - ▶ تعیین hp به مقدار ماکس
  - ▶ عوض شدن hp و ست کردن مقدار جدید
  - ▶ گرفتن مقداری که جان عوض میشود
- که با slider ای که برای این بخش تعریف شده است مقدار در الی عوض می شود.

# Swords

- ▶ برای انکه ace بتواند شمشیر بزند نیاز داشتیم که یک ابجکت خالی درست کنیم و روی دست ace بگذاریم که هنگام استفاده از کد نوشته شده برای animatoin که با کمک این کد میتوان شیء را در هر جهتی که خواستیم جابه جا کنیم یا بچرخانیم، شمشیر به درستی از بالا به پایین حرکت کند و تا مقدار ای که تعریف کرده ایم بیشتر حرکت نکند.
- ▶ برای این قسمت باید همه ی حالت هایی که برای چندبار زدن و عوض شدن یهویی در بعد حمله و نحوه ی گرفتن شمشیر برای دفاع و پریدن یکدفعه ای از حمله به دفاع را باید درست میکردیم
- ▶ شمشیر ها به صورت ارایه از prefab به ace داده شده اند که با مشخص کردن تعداد unlock میتوان از آن ها استفاده کرد.



# Enemies

## :Regular Enemies

- ▶ **plant**: دارای انیمیشن برای حمله است و گلوله ای از انرژی به طرف ace پرتاب میکند و مقدار hp ای برای خود دارد.
- ▶ **Bull**: دارای انیمیشن برای شروع حرکت خود دارد و برای مردن هم انیمیشن خاص دارد. این دشمن دارای هوش مصنوعی ایت که با کمک A\* مسیری از جایی که قرار دارد تا ace را پیدا میکند و موانع و مسیر های قابل رفت و آمد را در نظر میگیرد.
- ▶ **HellOfMirrors**: دارای انیمیشن برای شروع حرکت خود دارد و برای حمله کردن به پلیر هم انیمیشن خاص دارد. این دشمن دارای هوش مصنوعی ایت که با کمک آن مسیری از جایی که قرار دارد تا ace را پیدا میکند. (بسیار ترسناک)
- ▶ **Andromalius**: درار انیشن برای حالت ایستاده است و از ریشه ی درختان برای حمله به ace کمک میگیرد. حمله او position ای که ace در آن قرار دارد را میگیرد و از زیر زمین به ace حمله میکند.
- ▶ **Skeleton**: دارای انیمیشن برای شروع حرکت خود دارد و برای مردن و حمله با شمشیر هم انیمیشن خاص دارد. این دشمن دارای هوش مصنوعی ایت که با کمک A\* مسیری از جایی که قرار دارد تا ace را پیدا میکند و موانع و مسیر های قابل رفت و آمد را در نظر میگیرد.

## Boss Fight

- ▶ **Oak**: این باس فایت دارای 3 حمله ی مختلف است. اول از طرف صفحه به سمت دیگر حرکت میکند و بعد ریشه هایی از زیر زمین در میاورد تا به ace حمله کند و حمله ی دیگر این است که ریشه ها با سرعت و تعداد زیاد با الگوریتم خاص از دو طرف صفحه شروع به حرکت می کنند و تنها راه نجات از آن ها پریدن از آن ها است و یادگیری الگوریتم آن.
- ▶ **Jinx**: این دشمن هم توانایی تلپورت شدن دارد و هم گلوله ای از انرژی به سمت ace پرتاب میکند و با زدن رعد برق به صورت رندوم و بعد از آن احضار دشمنانی که تا به حال با آن ها جنگیدیم به ما حمله میکند و با سوراخ کردن زمین به صورت رندم باعث مردن ace مس شود. دارای انیمیشن برای حمله و ایستادن است.
- ▶ **Cyrus**: دارای هوش مصنوعی است که این شخصیت با شمردن تا 3 شماره به دنبال ace می افتد (با کمک A\*) و هیچ کدام از حملات و دفاع روس آن تاثیر ندارد.

# شمشیر های موثر روی هر کی از دشمنان

## Regular Enemies ▶

- ▶ plant: دو تا شمشیر اول (شمشیر عادی و شمشیر آتشی)
- ▶ Bull: سه تا شمشیر اول (شمشیر عادی و شمشیر آتشی و شمشیر الکتریکی)
- ▶ HellOfMirrors: فقط شمشیر اینه ای
- ▶ Andromalius: فقط شمشیر آتشی
- ▶ Skeleton: فقط شمشیر عادی

## Boss Fight ▶

- ▶ Oak: فقط شمشیر آتشی
- ▶ Jinx: فقط شمشیر الکتریکی
- ▶ Cyrus: هیچ یک از شمشیرها

# نحوه ی ساخت Regular Enemy

- ▶ RegularEnemyConfig: هر دشمن دارای یک config برای اسم و قدرت حمله و hp ای که دارد و زمان لازم بین حمله ها و اسم آن دشمن است.
- ▶ RegularEnemyController: هر دشمن یک controller برای بیس دارد که در آن مقدار HP و شمشیرهای افکتیو و مقداری که با ضربه ی ace از او کم میشود و یک بول که برای موقع حمله است دارد. در اینجا بیس کدی که برای همه enemy ها اجرا میشود مثل getHurt,onTriggerEnter,Hit شده شده است و در صورت لازم برای کنترلر هایی که از آن ارث بری میکنند override میشوند.
- ▶ EnemySubWeaponConfig: هر دشمن میتواند یک سلاح برای حمله داشته باشد. در اینجا مقداری که صدمه میزند و اسم آن سلاح را میگیریم
- ▶ EnemySubWeaponController: در اینجا بیس چیزی که لازم بود برای سلاح ها زده شده است مثلا Destroy و OnTriggerEnter

# نحوه ی ساختن Boss Fight

- ▶ `BossEnemyController`: این کلاس از `regularEnemyController` ارث بری میکند و `start` و `getHurt` در آن `override` شده است.
- ▶ بقیه ی قسمت ها مثل `regular enemy` است با این تفاوت که حملات `boss` ها خاص هستند و برای هر کدام از آبجکت هایی که ظاهر میشود باید کد خاص نوشته میشد و آن را درست میکرد به صورت جداگانه.

# Power Ups

- ▶ Speed Power Up: این prefab برای افزایش سرعت گذاشته شده است و کد آن به جز قسمتی که کال میکند در playerController نوشته شده است (کد آن برای زمان onTriggerEnter و Destroy است). و سرعت به تدریج افزایش میابد وقتی این المنت گرفته شود برای مدت 5 ثانیه
- ▶ Health Power Up: این prefab نیز دارای کد خود است برای زمان onTriggerEnter و Destroy و برای تغییر Hp متدی که در playerController پیاده سازی شده است را صدا میزند.

# Scenes

Village ►

First scene ►

Second Scene ►

Third Scene ►

Portal Secen ►

Endless Forest ►

Forest ►

Oak ►

The Cursed Circus ►

Jinx ►

HellOfMirrors ►

HellOfMirrors ►

MirrorsRoom ►

# Tile maps

► برای ساختن scene های استفاده شده در صفحه های قبل عکس های لازم را در نت برای محیط پیدا کرده و تنظیم کردیم و با ساختن TileMap از آن ها مراحل را خودمان ساخته ایم. (هر مرحله دارای tilemap برای خودش است)

# prototype

- ▶ در این قسمت regular enemy و کدهای مربوط به ان پیاده سازی شد و کدهای مربوط به player و swords و نحوه های حرکت sword و عوض کردن ان و پریدن ace
- ▶ Boss fight Oak در این مرحله به صورت کامل پیاده سازی شد.
- ▶ دشمن HellOfMirrors نیز به صورت کامل در اینجا پیاده سازی شد.
- ▶ HP برای ace و boss fight ها نیز طراحی شد.



# Main game

- ▶ همه ی طراحی های صحنه و تغییرات صحنه و همه چیز هایی که نشان داده شده است و در prototype ذکر نشده است در این دوره پیاده سازی شده است.
- ▶ (تقریبا 80 درصد بازی)

# وظایف افراد

- ▶ مراحل 1 و 4 به صورت همکاری از طریق اسکایپ زده شده اند
- ▶ مرحله 2 توسط خانوم مبشر پیاده سازی شد
- ▶ مرحله 3 توسط خانم پریسا علایی پیاده سازی شده است.