

## 1. ساختار کلی پروژه پایانی یونیتی

پروژه با استفاده از معماری Component-Based در موتور Unity پیاده‌سازی شده است. هر رفتار اصلی بازی در یک اسکریپت مجزا تعریف شده و اشیای بازی با ترکیب این اسکریپت‌ها عملکرد نهایی خود را به‌دست می‌آورند.

اسکریپت‌های اصلی پروژه شامل موارد زیر است:

- EnemyScript
- BossScript
- Enemy\_Maker
- SwipeAndTapManager
- game\_maneger
- SmartBomb
- menu\_Controller

---

## 2. تعریف نوع دشمن‌ها

برای دسته‌بندی دشمن‌ها از Enum زیر استفاده شده است:

- Enemy\_type
- Reproducible
  - firend
  - enemy
  - boos

این Enum در اسکریپت‌های EnemyScript ، BossScript و SmartBomb برای تصمیم‌گیری منطقی استفاده می‌شود.

---

### 3. اسکریپت EnemyScript

این اسکریپت رفتار دشمن‌های غیر باس را کنترل می‌کند.

#### متغیرهای مهم:

- type نوع دشمن
- Desrection\_main مداقل زاویه مجاز برش
- Desction\_max حداکثر زاویه مجاز برش
- MoveSpeed سرعت حرکت دشمن
- target هدف دشمن
- game\_Maneger مرجع مدیریت بازی

#### تشفیص برش صمیع:

public bool GetSlice(float angle)

این متد بررسی می‌کند که زاویه اسلش کاربر در بازه مجاز دشمن قرار دارد یا خیر.

#### حرکت دشمن:

در متد Update ، دشمن به سمت هدف حرکت می‌کند:

- جهت حرکت با نرمال‌سازی بردار موقعیت هدف مناسب می‌شود
- حرکت با توجه به Time.deltaTime انجام می‌شود

#### برفورد با دروازه:

در متد: OnCollisionEnter2D

- اگر دشمن از نوع firend نباشد، جان بازیکن کم می‌شود
  - دشمن از صحنه حذف می‌شود
- 

#### 4. اسکریپت BossScript

این اسکریپت رفتار باس بازی را مدیریت می‌کند.

کلاس angel\_Holder

برای تعریف مراحل مختلف باس از یک کلاس Serializable استفاده شده است که شامل:

- min\_angel
- max\_angel
- arrowObject

هر نمونه نشان‌دهنده یک مرحله آسیب‌پذیری باس است.

متغیرهای اصلی:

- angel\_Holder لیست مراحل باس
- index مرحله فعلی باس
- target هدف باس
- MoveSpeed سرعت حرکت باس
- game\_Manejer مرجع مدیریت بازی

منطق باس:

- در Start تمام فلش‌ها غیرفعال می‌شوند

- فلش مربوط به مرحله فعلی فعال می‌شود
- در صورت برش صحیح، اندیس مرحله افزایش می‌یابد
- پس از اتمام مراحل، باس از بین می‌رود

### تشخیص برش باس:

public bool IsInSlice(float angle)

---

### 5. اسکریپت Enemy\_Maker

این اسکریپت وظیفه تولید دشمن‌ها را بر عهده دارد.

### قابلیت‌ها:

- انتخاب تصادفی Prefab دشمن
- انتخاب تصادفی نقطه اسپاون
- امکان محدود یا نامحدود بودن تعداد دشمن‌ها
- استفاده از Coroutine برای کنترل زمان اسپاون

### منطق اسپاون:

در: Coroutine SpawnEnemies

- دشمن ساخته می‌شود
  - اگر EnemyScript داشته باشد، هدف و GameManager به آن اختصاص داده می‌شود
  - در غیر این صورت BossScript بررسی می‌شود
- 

### 6. اسکریپت game\_manager

این اسکریپت مدیریت کلی بازی را انجام می‌دهد.

### متغیرهای اصلی:

- Score امتیاز بازیکن
- life تعداد جان
- Score\_Text متن نمایش امتیاز
- Win\_Panel و Lost\_Panel پنل‌های پایان بازی

### مدیریت جان:

```
public void lostOneLife()
```

با هر فضا:

- جان کم می‌شود
- آیکون AI جان غیرفعال می‌شود
- در صورت صفر شدن جان، بازی متوقف می‌شود

### مدیریت امتیاز:

```
public void AddScore(int Ammont)
```

در صورت رسیدن امتیاز به 150، بازی به حالت برد می‌رود.

### تکثیر دشمن:

```
public void Add_Gost_Kids(...)
```

دو دشمن جدید در موقعیت‌های مختلف ساخته می‌شود.

---

## 7. اسکرپت SwipeAndTapManager

این اسکرپت ورودی لمسی بازیکن را مدیریت می‌کند.

### تشفیص لمس:

از `Input.touchCount` و `TouchPhase` برای مدیریت مراحل لمس استفاده شده است.

### تشفیص Tap و Swipe:

- فاصله کم Tap →
- فاصله زیاد Swipe →

### محاسبه زاویه اسلش:

`Vector2.SignedAngle(Vector2.right, delta)`

### برفورد اسلش:

- با `Physics2D.Linecast` بررسی می‌شود
- در صورت برفورد با `EnemyScript` یا `BossScript`، منطق مربوطه اجرا می‌شود

### واکنش‌ها:

- برش صمیم → حذف دشمن و افزایش امتیاز
- برش غلط → افزایش سرعت دشمن یا نمایش افکت

## 8. سیستم فنجر

برای نمایش بصری اسلش از Prefab فنجر استفاده شده است.

## ویژگی‌ها:

- ایجاد در نقطه شروع لمس
  - حرکت همزمان با انگشت
  - فعال و غیرفعال شدن فودکار
  - حذف در هنگام نابودی اسکریپت
- 

## 9. اسکریپت SmartBomb

این اسکریپت بمب هوشمند را پیاده‌سازی می‌کند.

## ویژگی‌ها:

- محدودیت تعداد استفاده
- تشخیص دشمن‌ها در محدوده دید دوربین
- حذف فقط دشمن‌های Reproducible
- حذف دکمه پس از رسیدن به حد مجاز

تشخیص دشمن‌ها با `Physics2D.OverlapAreaAll` انجام می‌شود.

---

## 10. اسکریپت menu\_Controller

این اسکریپت کنترل منوی اصلی بازی را انجام می‌دهد.

## قابلیت‌ها:

- شروع بازی با بارگذاری Scene بعدی
- خروج از برنامه با `Application.Quit`

---

## 11. جمع‌بندی نهایی

این پروژه شامل پیاده‌سازی کامل:

- سیستم ورودی لمسی (Tap) و (Swipe)
- تشخیص زاویه برش با منطق برداری
- دشمن‌های چندرفتاره
- باس چندمرحله‌ای
- مدیریت امتیاز، جان و پایان بازی

می‌باشد و تمامی منطق‌ها مستقیماً بر اساس کدهای پیاده‌سازی شده عمل می‌کنند.

توسعه دهندگان: زهرا تقوی و سارا تقوی