

TFG - Videojocs

Miquel Sanuy Trullàs

1. Introducció

Aquest document recull l'origen de diferents elements importants en el disseny del projecte. La idea és veure d'on prové cadascun i definir-ne una mica la seva funció dins del conjunt.

2. Codi

A continuació es llisten diferents documents de codi de C#. Si bé no els menciono tots, sí que mencionaré els que he creat o modificat jo, així com alguns dels que són de tercers però tenen un paper important. El codi està estructurat en diferents arxius repartits en directoris segons la seva funció:

-**AI**: conté el codi que utilitzen els enemics (tant els controladors d'aquests com altres classes de codi amb funcions necessàries).

-Classes BossController, EnemyController, SoldierController, BossMobile, EnemyMobile i SoldierMobile. Defineixen el comportament dels diferents enemics. Totes estan basades en EnemyController i EnemyMobile, que implementava el tutorial, i les altres han sigut modificades per mi. Les diferències principals entre elles es troben en com la IA apunta a l'objectiu i també trobarem diferències en els controladors de les animacions quan estiguin acabades. Les similituds es troben en el pathfinding i l'ús de les armes.

-DetectionModule: implementada en el tutorial, gestiona la detecció d'enemics.

-EnemyManager: modificada per mi per acceptar nous tipus d'enemic, gestiona la llista d'enemics perquè altres classes la puguin utilitzar.

-PatrolPath: modificada per mi per acceptar nous tipus d'enemic, permet crear camins que els enemics poden patricular.

-**Game**: conté diverses classes amb funcions bastant variades.

-Actor: classe creada pel tutorial que permet registrar els actors i la seva afiliació.

-Events: classe del tutorial que crea diversos events i dicta que fa cadascun, permetent a les altres classes invocar-los.

-**Game.Managers**: conté diverses classes creades en el tutorial que permeten gestionar les llistes de diversos aspectes (actors, audio, events i objectius). També conté la classe GameFlowManager que gestiona l'inici i el fi del nivell i quina escena es carrega en acabar.

-**Game.Shared**: conté classes principalment relacionades amb la vida dels actors i els elements que la poden alterar.

-Ammo: classe creada per mi que assigna diferents tipus de munició al jugador (i no a l'arma). Cada tipus el poden utilitzar diferents armes. També conte funcions per afegir o retirar munició.

-Damageable: classe creada en el tutorial que permet que un collider de qualsevol tipus actui com a HitBox, alterant la vida de l'objecte o el seu objecte pare si es que en té.

-DamageArea: classe del tutorial que permet als projectils crear un area a l'impactar. El mal que es fa als elements de l'àrea es pot configurar en funció de la distància del punt d'origen.

-Health: classe modificada per mi que permet registrar la vida d'una entitat i inclou funcions per afegir-ne. Les modificació principal permet que una entitat pugui curar-se en atacar una altra.

-Objective: classe implementada pel tutorial que permet registrar objectius. Es tracta d'una classe genèrica de la que altres més específiques hereden propietats.

-ProjectileBase: classe creada pel tutorial que determina alguns atributs del projectil

-WeaponController: classe modificada per mi que gestiona el comportament de les armes. Això inclou mal, cadència, retrocés, munició i diversos aspectes més. Les modificacions principals són per al sistema de munició.

-**Gameplay**: conté diversos elements i utilitats relacionats amb la jugabilitat.

-Pickup: creada pel tutorial, defineix el comportament d'un pickup genèric (com per exemple si volem que el model roti sobre si mateix), però no té cap efecte en sí.

-HealthPickup: creada pel tutorial, permet al jugador recuperar vida en recollir el pickup. Actualment no s'utilitza, però la menciono perquè és possible que l'utilitzi si és necessari.

-AmmoPickup: creada per mi, permet al jugador recuperar munició en recollir-lo (normalment al derrotar enemics).

-WeaponPickup: creada pel tutorial, permet al jugador recollir una nova arma.

-PlayerCharacterController: creada pel tutorial, gestiona el moviment del jugador.

-ProjectileStandard: creada pel tutorial, determina el comportament dels projectils (velocitat, mal, mida, efectes en impactar, etc.)

-ToggleOnDestroy: classe creada per mi que permet activar/desactivar diversos objectes quan es destrueix un objecte concret. S'utilitza principalment per al progrés dels objectius del nivell.

-ToggleOnReach: classe creada per mi que també activa/desactiva una llista d'objectes, en aquest cas quan el jugador entra en una zona determinada.

-**Gameplay.Managers**:

-PlayerInputHandler: classe del tutorial que gestiona els inputs del jugador. Permet que s'utilitzi tant teclat com mando a l'hora de jugar.

-PlayerWeaponsManager: classe modificada per mi que gestiona les diferents armes del jugador (retrocés, canvi d'arma, recàrrega, apuntar). També permet afegir o eliminar armes

de la col·lecció del jugador.

-Gameplay.Objectives: conté diversos tipus d'objectiu que hereden propietats de la classe pare Objective. Només menciono els que s'utilitzen.

-ObjectiveDestroy: classe creada per mi. L'objectiu es compleix en destruir un objecte determinat.

-UI: conté tots els elements relacionats amb la interfície d'usuari (tant menús com el HUD).

-AmmoCounter: classe modificada per mi que permet mostrar la munició que queda a una arma.

-Compass, CompassElement i CompassMarker: classes del tutorial que permeten gestionar la brúixola i els seus elements.

-FeedbackFlashHUD: classe del tutorial que permet que tota la pantalla s'il·lumini, normalment quan el jugador és atacat.

-FillBarColorChange: classe del tutorial que gestiona les barres, per exemple la de vida.

-FramerateCounter: classe del tutorial que permet veure els FPS en temps real mentre es juga.

-InGameMenuManager: classe del tutorial que gestiona el menú accessible dins del nivell.

-LoadSceneButton: classe del tutorial que permet carregar una nova escena.

-NotificationHUDManager: classe del tutorial que permet mostrar notificacions (per exemple en agafar una nova arma).

-ObjectiveHUDManager: classe del tutorial que permet visualitzar els diferents objectius, així com el seu estat.

-PlayerHealthBar: classe del tutorial que gestiona la barra de vida.

-PlayerHealthPercentage: classe creada per mi que permet veure el percentatge de vida.

-QuitGame: classe que permet tancar l'aplicació.

-ReloadHUD: classe creada per mi que mostra un avís mentre s'està recarregant qualsevol arma.

-StanceHUD: classe del tutorial que mostra l'estat del jugador (ajupit, caminant, corrent).

-WeaponHUDManager: classe modificada per mi que mostra les armes de les que disposa el jugador (índex, munició i icona).

-DoubleSlidingDoorController: classe d'un asset que permet obrir una porta quan s'acosta el jugador.

3. Prefabs

Tots els models, animacions i sons són del tutorial o d'assets de la store (llistats al final del document) i utilitzen el codi mencionat a l'apartat anterior. Així, menciono els que tenen algún interès i que utilitzo.

-Assets del nivell

-Enemic soldat

-Boss

4. Llista d'assets de la Unity Store
