**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**

Fundamentos de programación

Actividad Asíncrona #17

Alumno: Camacho Garduño Miguel Angel

Grupo: 3

Fecha: 27/11/2020

Actividad Asíncrona 17

Estructuras de repetición

Para mi proyecto que es realizar un videojuego de disparos en primera persona, decidí usar una plantilla de Unity llamada [FPS Microgame](https://learn.unity.com/project/fps-template) y en ella la estoy modificando para agregarle más objetivos y enemigos

# AudioManager.cs

Lo utilizo para que dentro de for, se evalúe la duración de un audio x

# ChargedProjectileEffectsHandler.cs

Lo utilizo para que, si se lanza un proyectil, se altere los renders de los objetos dañaos

# DamageArea.cs

Lo utilizo para crear una colección de componentes de salud únicos que se dañarían en el área de efecto (para evitar dañar una misma entidad varias veces). Y luego uso otro para aplicar daños con caída de distancia

# DetectionModule.cs

Lo utilizo para que dentro de “foreach” se evalúe si un actor (cualquier personaje x) puede ver a otro actor (cualquier personaje y) y si hay obstáculos como paredes o no.

# EnemyController.cs

Lo utilizo para se generen los renders de partes del cuerpo de los enemigos, y para que oriente su arma al jugador

# EnemyTurret.cs

Lo utilizo para que la torreta genere sonido al ver al jugador, detiene su sonido al no ver al jugador

# MeshCombiner.cs

Lo ocupo para que se pueden ser tangibles las nuevas texturas al compilar el juego y que también los enemigos tengas direcciones en los nuevos cuartos compilados

# MeshCombinerUtility.cs

Lo utilizo para crear lotes de renderizado para todas las combinaciones de materiales únicos + submeshIndex. También uso un for para combinar cada lote de renderizado único

# OverheatBehavior.cs

Lo utilizo para que se genere el render de sobrecalentamiento de arma cada vez que el arma se sobrecalienta

# PlayerWeaponsManager.cs

Lo utilizo para que cada vez que se recoja un arma nueva del suelo, permita la habilidad de cambiar al arma deseada

# PrefabReplacerOnInstance

Lo ocupo para que se evalúe si se ha insertado un prefab o textura prehecha, se pueda hacerle modificaciones de posición x-y-z, o sin se destruye el prefab si el usuario lo elimina del mapa en el editor

# WeaponPickup.cs

Lo ocupo para establecer todas las capas secundarias por defecto (para evitar ver armas a través de mallas)