|  |
| --- |
| Software de entretenimiento y Videojuegos Jordán Pascual : pascualjordan@uniovi.es |

|  |  |
| --- | --- |
| 8 | Unreal juego básico - Ampliaciones |

# Ampliaciones

Realiza **4** de las ampliaciones propuestas a continuación. Para la entrega de esta actividad se debe crear un informe explicando que ampliaciones se han decidido hacer y explicando brevemente que se ha realizado en cada una, incluir los fragmentos de código más relevantes.

Con 4 ampliaciones ya se puede obtener la puntuación máxima independientemente de su “Dificultad estimada”.

|  |  |
| --- | --- |
| Ampliación | **Dificultad** |
| 1 Nuevo enemigo animado: Añadir un nuevo enemigo animado (de forma similar al jugador), debe ser “humanoide”.  Utilizar los recursos Toony\_Tiny\_RTS\_Set\_UE5  El enemigo tiene que ser animado en lugar de disparar ataca cuerpo a cuerpo cuando está cerca del Jugador  Debe incluir todas las animaciones. | 6 |
| 2 Enemigo base resiste varios golpes. Hacer que el enemigo base resista dos golpes de espada  Debemos incluir una animación de golpeado en el enemigo. | 6 |
| 3 Jugador que salta. Agregar la capacidad de salto al jugador. | 5 |
| 4 Sistema de vida: Hacer que el jugador pueda ser golpeado 3 veces antes de reiniciar la partida | 7 |
| 5 Power Up de velocidad: Definir un powerup si el jugador lo toca se podrá mover de forma más rápida (modificar desde código las velocidades del componente CharacterMovement) | 6 |
| 6 Bomba: Al jugador la capacidad de lanzar un disparo. El disparo será una bomba que podrá impactar a los enemigos.  No hace falta incluir animación de disparo en el jugador.  Se debe utilizar un efecto, de explosión, se crean con la función Spawn Emitter at location. | 6 |
| 7 Caja de invulnerabilidad. Añadir un nuevo tipo de elemento que al tocarlo nos vuelve invulnerable por 5 segundos.  Para hacer ver al usuario que el jugador es invulnerable debemos acceder a su componente Mesh y cambiar el material de este (por cualquier otro material), una vez la invulnerabilidad finalice el jugador vuelve a su color original (Volver a establecer el material anterior).  Hay dos formas de realizar está ampliación, utilizando Delays o setTimerByFuncionName (permite ejecutar una función cuando pasen N segundos) | 6 |
| 8 Zona que ralentiza los enemigos.  Crear una plataforma de forma etérea (no genera colisión solo overlaps), cuando un enemigo entra en la plataforma ve reducida su velocidad a la mitad (usar el evento BeginOverlap)  Cuando el enemigo sale de la plataforma vuelve a tener su velocidad original (usar el evento EndOverlap) | 6 |
| 9 Sistema de generación de decoración aleatorio  Incluir un sistema que genere de forma aleatoria decoración en el mapa, utilizando diferentes modelos de casa, cada vez que se ejecute el juego se debe generar una decoración diferente | 6 |
| 10 Aldeanos. Incluir aldeanos que se moverán de forma aleatoria por el mapa, si el jugador los golpea pierde  Se deben animar debidamente. | 6 |
| 11 Devolver disparos. Si el jugador golpea con la espada a un disparo de enemigo invierte su dirección, este disparo retornado puede eliminar enemigos. | 7 |