

# Esame UE4 CG2

**Per evitare problemi di incompatibilità tra versioni, scaricate e programmate con la versione di Unreal 4.23.1 o con la 4.24.3**

Realizzare i seguenti punti:

**Creazione di un first person project.**

---

**Creazione di due Livelli di gioco:**

- **MainMenuLevel:** un livello vuoto che se avviato mostri solamente la User Interface interagibile (tramite i due pulsanti descritti in seguito) di MainMenu;
  - **GameplayLevel:** il livello contenente l'appartamento e tutte le logiche di Gameplay richieste per l'esame.
- 

**Creazione di un Widget di Gameplay:** Il Widget deve normalmente mostrare in alto a sinistra dello schermo, tramite interfaccia grafica, la vita del MainCharacter (la vita mostrata non deve essere mai inferiore a zero).

**Creazione di un Widget di MainMenu:** Il Widget deve contenere due pulsanti:

- Play Game: se premuto, tramite click sinistro, emette un suono di "click del bottone" e dopo un delay di 1 secondo avvia il livello "GameplayLevel" contenente l'appartamento.
  - Exit Game: se premuto, tramite click sinistro, emette un suono di "click de bottone" e dopo un delay di 1 secondo chiude il gioco.
- 

**Creazione di un Actor "Porta"** contenente la Static Mesh come componente. La porta deve essere in grado di aprirsi se chiusa e viceversa. Quando la porta viene aperta o chiusa deve emettere un Suono. Inserire tali funzionalità all'interno dell'Actor.

**Creazione di un Actor "Porta sbloccabile tramite due chiavi"** contenente la Static Mesh come componente. La porta è normalmente chiusa e si può solamente aprire se il Main Character entra in possesso di due chiavi (e cioè se interagisce con i due Actor Key inseriti in scena).

Quando il MainCharacter interagisce con la Porta, senza aver ottenuto entrambe le chiavi, la porta emette un suono di "porta bloccata" e la porta rimane chiusa.

Inoltre su interfaccia grafica (sfruttate il Widget di Gameplay) deve apparire un messaggio su schermo (un Text) diverso in base al numero di chiavi ottenute dal MainCharacter:

- Se il Character deve ancora prendere (interagire) le due chiavi ed interagisce con la porta (prova ad aprirla), su schermo deve apparire *“La porta è bloccata. Hai bisogno di due chiavi”*.
- Se il Character ha preso una sola delle due chiavi ed interagisce con la porta, su schermo deve apparire *“La porta è bloccata. Hai bisogno di una ulteriore chiave”*.
- Se il Character ha preso entrambe le chiavi in scena ed interagisce con la porta, su schermo deve apparire *“La porta è sbloccata. Puoi proseguire”*. Naturalmente, in questo caso, la porta si aprirà.

Una volta sbloccata la porta dovrà comportarsi come un'altra qualsiasi Porta nel Livello, quindi deve essere in grado di aprirsi se chiusa e viceversa, emettendo un suono di apertura/chiusura.

**Nota bene:** solo questa “porta sbloccabile tramite chiavi” permette di accedere in giardino e quindi di uscire dall'appartamento.

**Creazione di un Actor “Key”** contente la StaticMesh o le Basic Shape (es: Cubi) come componente.

Quando il Character interagisce con la chiave deve:

- essere emesso un suono di “chiave ottenuta”;
- essere mostrato su schermo, tramite User Interface (sfruttate il Widget di Gameplay), un alert di chiave ottenuta (deve essere mostrata su interfaccia l'immagine della chiave tramite un'animazione di fade in/out sulla trasparenza, come fatto a lezione);
- “avvertire” la porta bloccata che la chiave è stata ottenuta;
- “sparire” dalla scena (Il Character la prende, non deve rimanere in scena).

Di questo unico Actor creare due istanze di chiavi in scena (le due chiavi che, solamente se prese entrambe, possono aprire la porta bloccata che permette al Main Character di uscire in giardino).

**Creazione di un Actor “Finestra a scorrimento”** contenente la Static Mesh come componente. La finestra deve essere in grado di aprirsi se chiusa e viceversa (è una finestra a scorrimento, modificate la posizione della finestra. Non voglio una rotazione). Quando la finestra viene aperta o chiusa deve emettere un Suono. Inserire tali funzionalità all'interno dell'actor.



**Creazione di un Actor “Interruttore”** contenente la Static Mesh Interruttore come componente che abbia come variabile pubblica un array di Spotlight Actor. Tutte le lampadine inserite nell’array devono accendersi se spente e viceversa. Quando vengono accese/spente l’Actor deve emettere un Suono 3D di lampadina accesa/spenta.

**Creazione di un Actor “Forno da Cucina”** creato tramite Basic Shape di Unreal. Inserire tali Basic Shape come figlie dell’Actor creato, inoltre inserire come figlio dell’Actor un Componente di tipo Particle System (con Template un P\_Fire). Lo sportello del forno deve essere in grado di aprirsi se chiuso e viceversa. Quando il forno viene aperto, avviare il Particle dell’Actor ed un Suono in loop (il rumore di un forno acceso), quando viene chiuso interrompere il Particle ed il Suono. Inserire tali funzionalità all’interno dell’Actor.

**Suggerimento:** Per i Suoni 3D create dei Sound Cue e dei Sound Attenuation Asset. Es. un Sound Cue interruttore (con una logica interna a vostro piacimento) a cui è associato un Sound Attenuation Asset.

---

**Creazione di un Actor “Spline”** contente come Component una Box Collision, la Static Mesh e la Spline. Quando il giocatore entra all’interno della Box Collision, la Static Mesh si dovrà muovere lungo la spline.

Esempio: Un UFO che si muove lungo la Spline, un oggetto in scena che comincia a muoversi.

---

**Creazione di un Actor “Change Camera”** contenente una variabile pubblica Array di Camera Actor ed una variabile pubblica di tipo float di nome “Delay”. Questo Actor prende come input un array di Camera Actor (non attive) presenti nella scena e le attiva una alla volta, dalla prima all’ultima. La camera corrente rimane attiva per il tempo di “delay” assegnato come input. Posizionate le Camera Actor in punti strategici della vostra casa, inquadrare gli scorci della casa su cui volete portare l’attenzione del giocatore.

Al Play del progetto verranno attivate una alla volta le varie camere inserite nell’array ed alla fine verrà riattivata la camera principale del Main Character (la First Person Camera per intenderci), permettendo al giocatore di riprendere a giocare.

---

**Level Blueprint:** All’interno del Level Blueprint, dei due livelli da voi creati, avviate un Sound 2D in loop. Il suono dovrà avviarsi automaticamente con l’avvio del livello. Quindi deve avviarsi un suono di sottofondo per il MainMenuLevel ed un suono di sottofondo per il GameplayLevel (il livello contenente l’appartamento).

Es: scegliete una musica di sottofondo che meglio si addice al Menu ed all’appartamento/villa che avete creato.

**Sistema di interazione del player che sfrutti un LineTraceByChannel** per riconoscere l'Actor con cui interagire e quindi richiamare la specifica funzionalità dell'Actor.

Il raggio deve SEMPRE partire dalla camera e puntare di fronte al giocatore. Mi aspetto che l'interazione con gli Actor Interagibili avvenga solamente quando il linetrace del giocatore abbia colpito (hit) e riconosciuto l'Actor con cui interagire.

L'interazione con L'Actor deve avvenire quando vengono verificate nello stesso momento la seguenti due condizione:

- Linetrace che colpisce l'Actor con cui interagire,
- Giocatore che preme con il click sinistro del mouse (come in un qualsiasi videogame).

Vi consiglio di creare ed utilizzare una Interface contenente una **Function Interact**.

Quando viene cliccato il tasto sinistro del mouse, deve essere lanciato un LineTrace. Quando il LineTrace riconosce un Actor interagibile (es: la porta), tale Actor deve essere interagito e **non deve** essere sparato alcun proiettile.

Quando invece viene premuto il tasto sinistro del mouse ed il LineTrace non riconosce un Actor interagibile, il proiettile **può** essere sparato.

**Creazione di un Character con associato un Animation Blueprint:** Il Character deve essere un "mostro", scegliete voi quale (es: lo zombie fatto a lezione).

Tale Character deve avere 10 punti vita (hp), la vita decrementa di uno ogni volta che viene colpito da un proiettile del giocatore (il FirstPersonProjectile).

Inoltre, il "mostro" deve essere in grado di inseguire il giocatore tramite NavMesh. Creare un BlendSpace di Locomotion e gestire nella State Machine le animazioni di hit (quando il mostro viene colpito dal proiettile) e di morte (quando il mostro raggiunge vita uguale a zero).

**Costruite la logica che permette all'Enemy di attaccare il giocatore.** Assegnate quindi al giocatore 100 punti vita, tale vita viene decrementata di cinque ogni volta che viene colpito dal Mostro nemico. Il Mostro attacca il protagonista solamente quando si trova vicino ad esso ed è in grado di effettuare un attacco in un intervallo che varia dai due ai cinque secondi.

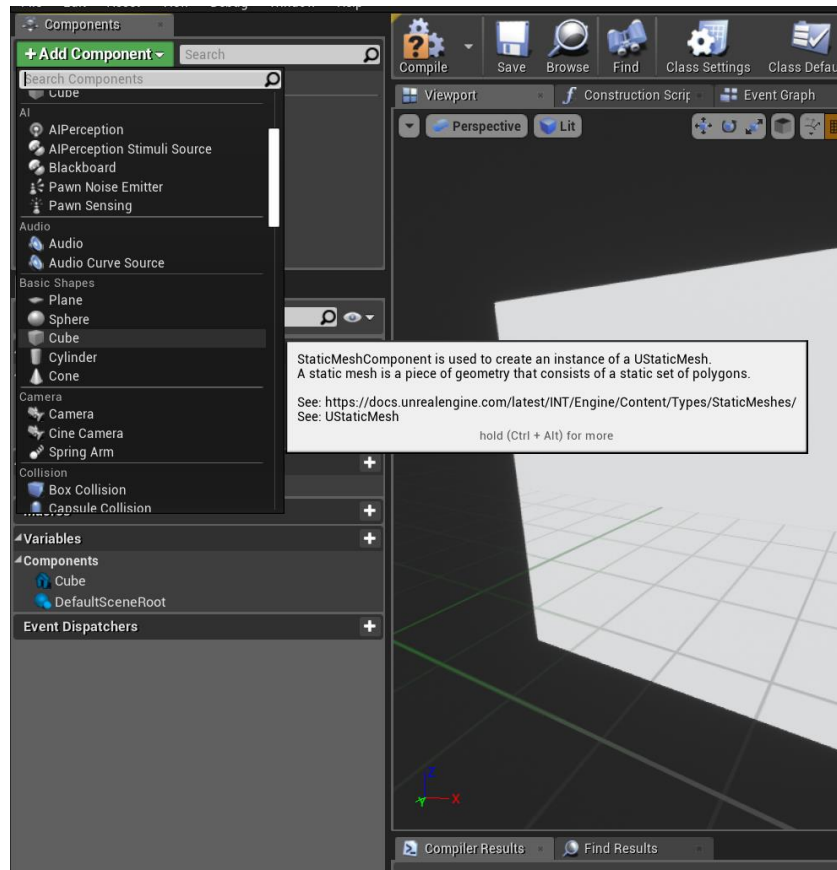
Quando la vita del protagonista scende a zero, viene riavviato il livello di MainMenu.

**Asset Animation:** Inserite almeno uno SkeletalMesh Actor con una Animazione (Asset Animation). Scaricate l'animazione e la Skeletal Mesh da Mixamo.

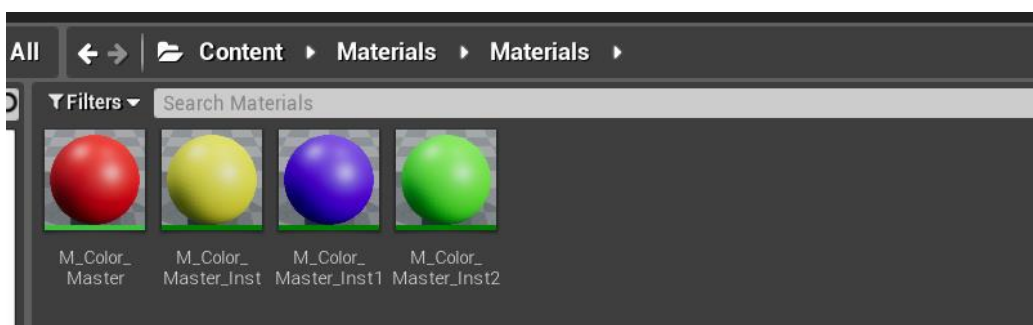
Esempio: l'animazione di un Character spaventato dagli zombie

**Creazione di un appartamento/villa con giardino:** la casa dovrà essere credibile, arredata con mobili e percorribile interamente tramite il First Person Character. Per creare la struttura della casa utilizzate le geometry base di Unreal (box, sphere...). Per i mobili potete utilizzare quelli contenuti nei plug-in scaricati durante la durata del corso, oppure crearne di vostri tramite le Basic Shape di Unreal (Sphere, Cube, Cylinder...). Per la creazione di mobili tramite Basic Shape vi consiglio di creare degli Actor contenenti dei Componenti di tipo Static Mesh BasicShape.

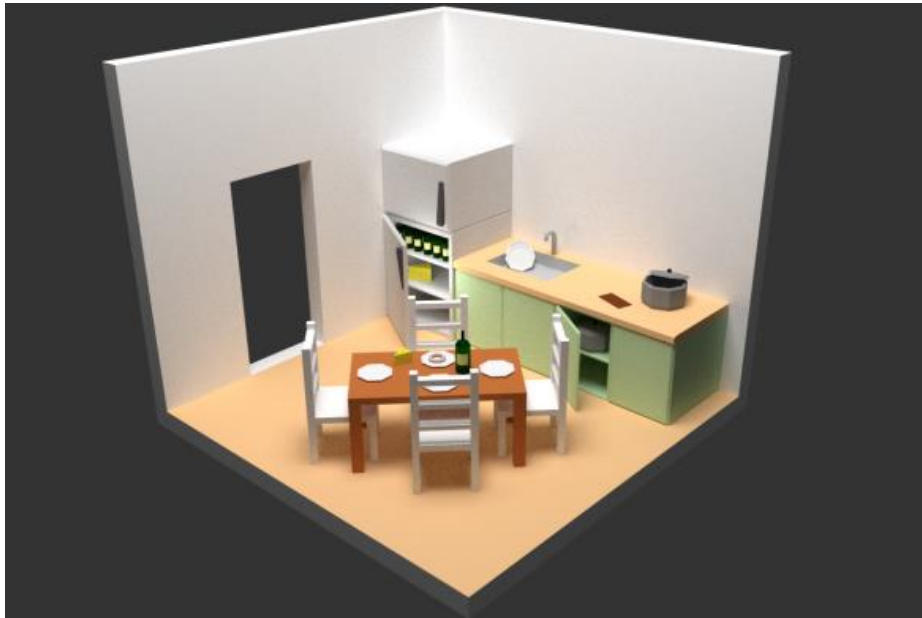
**ESEMPIO:** Per la creazione di una sedia, create un Actor “Sedia” contenente sei Componenti di tipo Cube: quattro cubi per i piedi della sedia, uno per la seduta ed uno per lo schienale.



Vi suggerisco di assegnare a tutte le mesh presenti in scena dei Materiali (delle Material Instance che differiscono per Base Color o per Texture).



Un esempio di risultato:

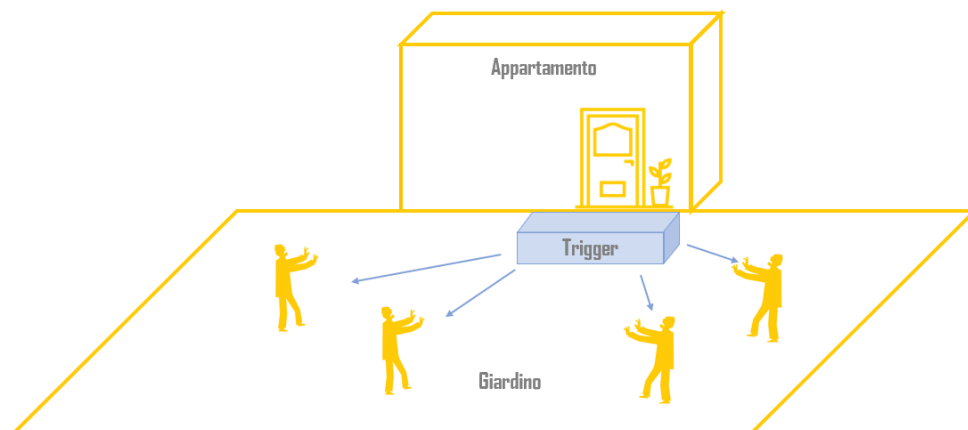


Il giocatore inizierà il gioco all'interno della villa e potrà uscire solamente dopo aver sbloccato la porta (sbloccabile tramite le due chiavi) per arrivare in giardino.

In giardino lo attenderanno diversi mostri (scegliete voi quanti). Tutti i mostri saranno normalmente fermi in Idle (mi aspetto che durante l'esplorazione della villa, quando il protagonista non è ancora uscito nel giardino, i mostri stiano fermi) e si attiveranno solamente quando il giocatore arriverà in giardino. **Solo quando attivi** i mostri seguiranno il Character.

**Prevedere la creazione di un Trigger da inserire all'uscita dalla villa** che attivi i mostri in scena. Quando il giocatore attraversa tale Trigger, il Trigger "sveglierà" i Character "mostri" in scena che, da quel momento in poi, cominceranno ad inseguire il protagonista del gioco.

Vi consiglio di utilizzare nel Trigger una **variabile array pubblica** dello stesso tipo del Monster che avete creato.



**Verrà valutato come bonus sul punteggio finale dell'esame** ogni vostra singola aggiunta spontanea non richiesta nell'esame. Ad esempio la creazione di altri tipi di Actor con cui interagire, l'aggiunta di particle all'interno della scena, ecc...

## **Organizzazione del Progetto (anche l'organizzazione sarà valutata).**

Nella root del ContentBrowser create una cartella Levels contenente i due Livelli di MainMenuLevel e GameplayLevel (contenente la villa con giardino).

Nella root del ContentBrowser create una cartella Blueprints, contenente tutti gli Actor da voi creati.

Nella root del ContentBrowser create una cartella Sounds, contenente tutti i suoni da voi utilizzati nel progetto.

Nella root del ContentBrowser create una cartella Animations, contenente tutti le Animazioni da voi importate da Mixamo.

## **CONSEGNA**

**Create una cartella con il vostro nome e cognome, inserite dentro di essa la cartella contenente il progetto d'esame e copiatela all'interno del Drive.**

Se scaricate plugin, eliminate gli asset inutilizzati all'interno del progetto. Preferibilmente non superate i 5GB di progetto.

L'esame dovrà essere consegnato **entro e non oltre Venerdì 26/06/2020 ore 20.00**.

La data dell'esame invece è il 1 Luglio ore 18.30, per confermare o alzare il voto tramite domande orali.

Rimango a disposizione per dubbi e/o chiarimenti alla mail: [mic.dantimi@gmail.com](mailto:mic.dantimi@gmail.com)

Grazie e buon lavoro ragazzi.

Michele D'Antimi