

Capitolo 1 – introduzione ed obiettivi

L'obiettivo di questo progetto è quello di sfruttare il motore di gioco Unity e le tecnologie per la realtà virtuale per creare un ambiente virtuale di apprendimento che possa funzionare come supporto aggiuntivo durante lo studio di Fisica 1.

Il gioco si pone come obiettivo quello di fornire allo studente un ambiente “ibrido”, in cui le esperienze pratiche tipiche di un classico laboratorio di fisica si fondono al più diffuso metodo di studio fornito dai libri di testo, a cui all'apprendimento dei concetti segue la risoluzione di problemi. Tale mix fornisce allo studente problemi da risolvere effettuando misurazioni su oggetti presenti nell'ambiente ed effettuando calcoli. La differenza tra questo approccio ed una classica esperienza di laboratorio è che, diversamente dalle esperienze puramente pratiche, in questo caso il procedimento da seguire non è noto, e sarà compito dello studente capire quali passi matematici utilizzare per arrivare alla soluzione, come in ogni problema da libro di testo. In più ci sarà anche la difficoltà aggiunta di capire quali misurazioni saranno necessarie per risolvere i problemi forniti.

L'obiettivo di tale approccio è quello di sviluppare le competenze dello studente in modo pratico senza trascurare la parte puramente matematica della disciplina.

In aggiunta al risolvere problemi, lo studente avrà anche la possibilità di sperimentare in modo indipendente utilizzando gli strumenti forniti dal gioco, che si dividono in strumenti di misurazione, e strumenti di creazione. Quest'ultimi hanno lo scopo di permettere allo studente di creare oggetti con cui svolgere esperimenti e misurazioni utilizzando gli stessi strumenti utilizzati per la risoluzione dei problemi.

CAPITOLO 2 – State of the art

CAPITOLO 3 – Tecnologie utilizzate

Unity

Unity è un motore di gioco sviluppato da Unity Technologies, scritto in C++. Esso può essere utilizzato per creare ambienti 3D e 2D, e supporta una grande varietà di piattaforme desktop, console, mobile, e anche dispositivi stand-alone per la realtà virtuale. Uno degli obiettivi di Unity è quello di rendere lo sviluppo di videogiochi più accessibile a sviluppatori indipendenti, e tale accessibilità ha fatto in modo che il motore venisse usato in molti campi al di fuori dell'industria del videogioco.

La sua natura object-oriented rende Unity un motore estremamente intuitivo e facile da apprendere. La struttura base di un progetto in Unity è costituita da “scene”. All'interno di tali scene è possibile posizionare “oggetti”, i quali a loro volta possono contenere oggetti (figli). Ad ogni oggetto possono essere annessi infiniti “componenti”, i quali regolano i vari aspetti del comportamento di tali oggetti all'interno della scena: posizione, rotazione, proprietà fisiche, aspetto, emissione di suoni e particelle, collisioni, animazioni etc...

In più, lo sviluppatore ha la possibilità di creare componenti ad hoc tramite degli script in C#. Quest'ultima funzione di Unity rende il motore di gioco estremamente versatile per qualsiasi tipo di applicazione.

La fisica all'interno di un'applicazione 3D in Unity è gestita dal motore fisico Nvidia PhysX, il quale è in grado di processare il comportamento fisico degli oggetti in tempo reale, tramite accelerazione hardware con la GPU.

HTC Vive

CAPITOLO 4 – Meccaniche del sistema

Idea generale

Il gioco si presenta come un *open world*, ovvero quel genere di videogiochi che permette al giocatore di vagare all'interno di una vasta area senza restrizioni. Il mondo di gioco è popolato da *NPCs* (non-player character) con cui il giocatore potrà parlare. Tali NPC offriranno al giocatore dei problemi da risolvere, a cui il gioco si riferisce con il termine *quest*. Completare con successo una quest premierà il giocatore dandogli accesso a nuove quest o nuovi oggetti, questi ultimi meglio definiti più avanti. Per ottenere una quest da un NPC, il giocatore dovrà parlare con esso, premendo un bottone verde presente sul corpo di ogni NPC.

Profili

Il giocatore avrà bisogno di creare un profilo per poter giocare. All'interno del profilo vengono memorizzati il nome del giocatore, le quest completate, e tutti gli oggetti sbloccati tramite il completamento delle quest. Una volta sbloccato, ogni oggetto è facilmente accessibile in ogni momento tramite un menù di gioco.

Oggetti

La prima categoria di oggetti sbloccabili è quella degli strumenti. Nel gioco sono presenti vari strumenti che il giocatore può tenere in mano utilizzando i controller del visore, e si dividono in due categorie:

- Strumenti di creazione: permettono al giocatore di creare oggetti come cubi, sfere e piattaforme con cui completare alcune quest o sperimentare in modo indipendente.
- Strumenti di misurazione: servono ad ottenere dati numerici con cui completare le quest.

La seconda categoria di oggetti sbloccabili è costituita da frammenti di informazioni leggibili dal menù di gioco, i quali sono l'unica fonte scritta di nozioni all'interno del gioco, e hanno il compito di accompagnare il giocatore lungo il percorso, introducendolo ai concetti e alle formule matematiche necessari a proseguire.

Sulla terza categoria di oggetti si basa il funzionamento della parte puramente matematica del gioco. Essa è costituita dalle datacard. Questi oggetti sono delle cartucce il cui compito è quello di

memorizzare dati. Ne esistono due tipi:

- Datacard valore: contengono al loro interno un singolo valore numerico, accompagnato da un' etichetta, che indica la natura di tale valore, ad esempio: massa, angolo, volume, etc...
- Datacard formula: contengono al loro interno un set di formule matematiche, con cui verranno calcolati nuovi valori numerici a partire da quelli ottenuti tramite le datacard valore. Tale meccanica verrà discussa in seguito.

Computer

Il giocatore è seguito in ogni momento dal suo assistente. L' assistente è un NPC dalle sembianze di un robot volante dotato di un computer. Tale computer è l'elemento principale della parte matematica del gioco. Esso appare come un uno schermo affiancato da diversi slot in cui andranno inserite le datacard. Una volta inserita una datacard formula nell'apposito slot, sullo schermo del computer apparirà una serie di tasti che accompagnerà il giocatore nella scelta della formula da utilizzare per il calcolo. Tale formula dovrà essere scelta basandosi sulla natura del problema che si ha di fronte, sfruttando le nozioni apprese dai frammenti di informazioni sbloccati. Una volta scelta la formula da utilizzare, il computer richiederà di inserire delle datacard valore all'interno di appositi slot, su cui sarà specificato il tipo di dato richiesto. Inserendo più di una datacard valore all'interno dello stesso slot si otterrà la media aritmetica fra tutti i valori inseriti, la media immagazzinata in ogni slot può essere azzerata tramite un apposito tasto.

CAPITOLO 5 – Implementazione

Profili

I profili dei giocatori e i relativi dati vengono salvati all'interno di un file JSON. Ogni oggetto contiene al suo interno i seguenti campi:

- Nome del giocatore.
- Quest completate.
- Frammenti di informazioni sbloccati.
- Strumenti sbloccati.
- Datacard sbloccate.

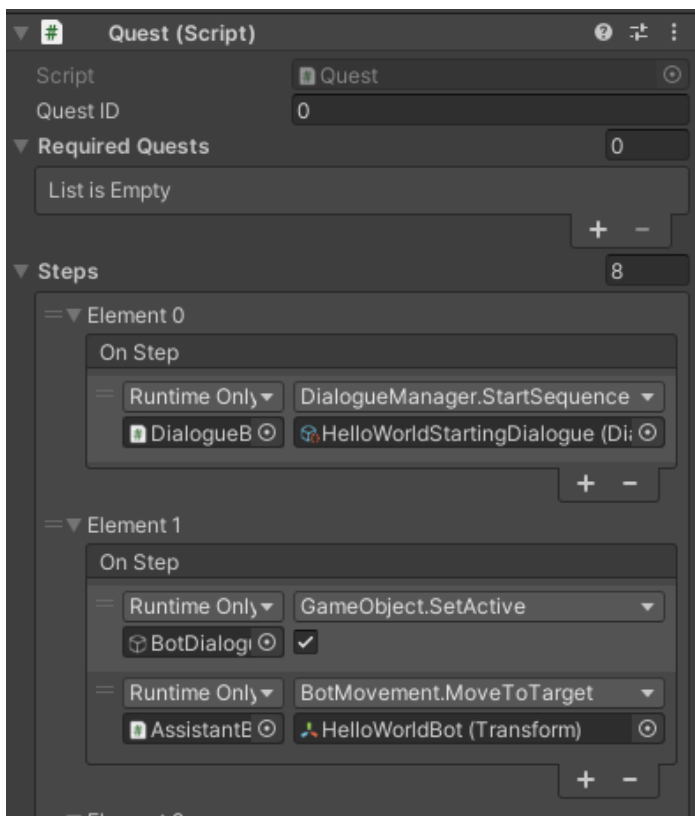
Eccetto per il nome del giocatore, ogni altro campo è rappresentato come un array di numeri interi. Questo perché all'interno del gioco, ogni oggetto sbloccabile e ogni quest sono rappresentati da un ID unico. In tal modo, ad un nuovo caricamento del profilo, il gioco non proporrà quest già completate al giocatore, e nel menù di gioco saranno disponibili tutti gli oggetti sbloccati.

Il profilo da cui vengono caricati i progressi è accessibile in ogni momento per mezzo di uno script "ProfileManager", contenuto all'interno di un oggetto "GameManager". Tale script si occupa anche di salvare i progressi qualora il giocatore dovesse sbloccare un nuovo oggetto o completare una quest. Le funzioni dello script ProfileManager vengono utilizzate all'interno di altri script, primo tra tutti, lo script Quest.

Quest

Le quest sono state implementate nel gioco come oggetti dotati di uno script "Quest". All'interno di tale script si trovano 3 elementi principali:

- ID unico della quest.
- Quest richieste: è un array contenente gli ID delle quest che devono essere completate prima di poter iniziare la quest rappresentata da quest'oggetto.
- Passi: è un array contenente structs chiamate QuestStep. Ognuno di questi QuestStep contiene a sua volta un elemento di tipo UnityEvent. Tale struct serve a definire una sequenza di funzioni che devono essere chiamate durante il proseguimento della quest. Ogni volta che al componente Quest viene ordinato di andare al passo successivo, l'evento contenente le corrispondenti funzioni viene invocato. Ciò serve a stabilire una sequenza cronologica di azioni che gli oggetti sulla scena devono compiere, come ad esempio far partire un dialogo con un NPC.



In più, ogni Quest possiede un campo OnInsufficientRequirements di tipo UnityEvent. Tale evento verrà invocato nel caso in cui il giocatore non ha completato le quest richieste per affrontare la suddetta.

NPC

I personaggi non giocanti all'interno della mappa hanno il compito di offrire nuove quest al giocatore. Essi sono equipaggiati con lo script "QuestGiver", il quale contiene al suo interno un array di riferimenti a tutti le quest che l'NPC è in grado di offrire al giocatore, oltre ad un campo OnAllQuestsCompleted di tipo UnityEvent, che verrà invocato una volta che le tutte quest offerte dall'NPC verranno completate e il giocatore proverà a chiederne un'altra.