



2021-2-IT02-KA210-SCH-000048086

Erasmus+



PLAY

PROTECT, LEARN, ENGAGE AND ENJOY

Toolkit



A.M.E.F.E

asociación malagueña de
educación y formación europea





MODULO 1

Unità 1 - Panoramica e obiettivi generali del progetto.....	1
Unità 2 - Digital Competence Framework per Educatori (DigCompEdu).....	5
Unità 3 - Gamification.....	9
Unità 4 - La valenza educativa di utilizzare un approccio incentrato sulla tecnologia..	13
Unità 5 - Risultati del progetto.....	16

MODULO 2

Unità 1 - Unity 3D Platform: usi e potenzialità.....	17
Unità 2 - Interfaccia.....	20
Unità 3 - Come creare codici utilizzando il linguaggio di programmazione C#.....	60
Unità 4 - Input, fisica , interfaccia utente, scene e rendering.....	78
Unità 5 - Guida pratica con esempi di codice.....	89



Modulo 1

Unità 1 - Panoramica e obiettivi generali del progetto.

Play - protect, learn, engage and enjoy è un progetto finanziato dal programma Erasmus+ dell'Unione Europea. Erasmus+ è il programma che si occupa di Educazione, Formazione, Gioventù e Sport. Il partenariato è composto da una scuola italiana, una scuola polacca, un'associazione spagnola di educazione e formazione e Paidea, un'azienda che si occupa di formazione nell'ambito edtech. L'idea del progetto è nata dalla volontà di rendere la scuola un ambiente in cui gli studenti possano sperimentare e sviluppare le proprie competenze utilizzando un approccio pratico grazie al supporto della tecnologia. Infatti, secondo le direttive del DigComp, le materie insegnate a scuola dovrebbero essere direttamente collegate al mondo in cui viviamo e dovrebbero migliorare lo sviluppo delle competenze digitali che sono necessarie in una società in costante evoluzione.

Il progetto PLAY ha avuto lo scopo di promuovere le competenze digitali degli studenti tra gli 11 e i 14 anni e aumentare la loro consapevolezza sull'importanza di proteggere il pianeta attraverso il loro diretto coinvolgimento nella realizzazione di un videogioco, sviluppato attraverso la piattaforma Unity 3D. Lo sviluppo del videogioco ha permesso loro di imparare ad utilizzare il linguaggio di programmazione C#. Uno degli obiettivi di sviluppo sostenibile previsti dall'Agenda 2030 sottolinea la necessità di intraprendere azioni urgenti per combattere il cambiamento climatico e il suo impatto. In linea con l'Agenda 2030, PLAY ha utilizzato un approccio ludico per aumentare la consapevolezza degli studenti sull'importanza di modificare il proprio stile di vita e per incoraggiare lo sviluppo di competenze digitali che risultano essere importanti nel mondo del lavoro.

Dopo un periodo di formazione, gli studenti hanno sviluppato un videogioco in 3D dove è possibile interagire con diversi ambienti per comprendere le principali cause dello spreco di risorse e dell'inquinamento, nonché per conoscere alternative sostenibili che contribuiranno a ridurre il nostro impatto sull'ambiente.



Unity è un motore di gioco multiplatforma utilizzato principalmente per sviluppare videogiochi e simulazioni per computer, console, dispositivi mobili e siti web. È possibile programmare in C#, Java, UnityScript o Boo, un linguaggio simile a Python. Nel progetto il linguaggio di programmazione selezionato è stato C# in quanto è molto facile da usare ed adatto agli sviluppatori principianti.

Gli studenti hanno sviluppato il videogioco con la supervisione di programmatori e formatori esperti. In particolare, le attività hanno seguito le seguenti fasi:

- **Formazione e pratica.** Nella prima fase i discenti sono stati formati su come utilizzare Unity e come programmare in C# (insieme agli insegnanti); dopo aver ricevuto un background teorico, si sono esercitati sulla scrittura di codici efficienti.
- **Piano di sviluppo.** Sono state organizzate molte sessioni di brainstorming, dove gli studenti hanno sviluppato le proprie idee e collaborato tra di loro. Dopo di che, hanno svolto ricerche sulle principali cause del cambiamento climatico e sulle soluzioni più sostenibili. Questo ha portato alla selezione di sei ambientazioni, che rappresentano lo sfondo interattivo e i livelli del gioco.
- **Produzione.** La fase in cui il videogioco è stato effettivamente sviluppato.



- Test e validazione. Quando l'output è stato completato, la versione finale è stata testata dagli studenti delle scuole partner che non hanno preso parte al progetto.

Gli insegnanti hanno partecipato attivamente a tutte le fasi della produzione del videogioco, ciò ha permesso loro di apprendere metodi di insegnamento innovativi e anche di sviluppare nuove competenze.

Gli obiettivi principali del progetto possono essere riassunti come segue:

1. promuovere lo sviluppo di competenze di programmazione e digitali in giovani studenti, per renderli competitivi nel mondo del lavoro;
2. aumentare la consapevolezza, tra gli studenti e la comunità in generale, sulle cause del riscaldamento globale e sull'importanza di combattere il cambiamento climatico modificando le nostre abitudini;
3. promuovere lo sviluppo di metodi didattici innovativi e digitali fornendo agli insegnanti le conoscenze e le competenze necessarie per aumentare il livello di motivazione degli studenti e il loro rendimento scolastico.

Attraverso lo sviluppo del videogioco, gli studenti hanno avuto l'opportunità di apprendere praticamente come si programma e hanno acquisito competenze digitali che potranno essere utilizzate nel loro futuro professionale. Infatti, la programmazione ha un ruolo fondamentale nel nostro mondo. Imparare i fondamenti della programmazione può dare agli studenti un vantaggio competitivo nel mondo sempre più digitale. Le competenze nel campo della programmazione sono importanti anche per imparare ad innovare e creare soluzioni eco-friendly per problemi globali. Gli studenti hanno allo stesso tempo ampliato la propria conoscenza sul tema e sono diventati individui responsabili che possono assumere un ruolo attivo nella lotta al cambiamento climatico. Essere consapevoli di queste minacce e aiutare gli altri a comprendere le conseguenze delle nostre azioni ha avuto un impatto importante sulle vite dei partecipanti, rendendoli più responsabili e consapevoli sull'ambiente e il loro ruolo nella società. Le attività realizzate miravano a promuovere l'inclusione e il lavoro di gruppo. Naturalmente alcuni studenti erano



più esperti nel digitale e altri erano maggiormente interessati e sensibili sul tema ambientale. Lavorando insieme hanno avuto la possibilità di imparare gli uni dagli altri.

Nei nuovi programmi di apprendimento è necessario utilizzare metodologie appropriate, come per esempio la *gamification* o *game based learning*, per coinvolgere gli studenti, stimolare la loro creatività, personalizzare i programmi di apprendimento e rendere i contenuti più accattivanti. Ognuno di questi aspetti rappresenta una chiave che gli insegnanti dovrebbero prendere in considerazione per raggiungere gli obiettivi pedagogici e di apprendimento. Ciò richiede che gli insegnanti posseggano competenze digitali: in modo da utilizzare la tecnologia non solo come supporto per le attività, ma anche per modificare i contesti educativi e formativi.



Unità 2 - Digital Competence Framework per Educatori (DigCompEdu)

Il Framework Europeo per le competenze digitali degli educatori (DigCompEdu) è un sistema scientificamente valido che descrive cosa significa per gli educatori essere digitalmente competenti. Esso fornisce un quadro di riferimento generale per supportare lo sviluppo di educatori con specifiche competenze digitali in Europa. DigCompEdu è rivolto agli educatori di tutti i livelli dalla prima infanzia all'istruzione superiore e agli adulti, compresa l'istruzione e la formazione generale e professionale, nell'insegnamento a studenti con bisogni educativi speciali e i contesti di apprendimento non formali.

Le competenze digitali sono necessarie all'apprendimento e al lavoro e ci rendono membri attivi della società.

In relazione all'istruzione, sapere come sviluppare le competenze è importante quanto la comprensione delle competenze stesse.

Cosa sono le competenze digitali?

Le competenze digitali sono tra le 8 competenze chiave indicate dall'Unione Europea e si riferiscono all'uso consapevole e critico dell'intero spettro delle tecnologie che consentono l'acquisizione di informazioni, comunicazione e risoluzione dei problemi in tutti gli aspetti della vita.

Cos'è il quadro delle competenze DigCompEdu?

Il quadro distingue 6 aree di copertura:

1. Impegno professionale - l'uso di tecnologie di comunicazione efficaci, cooperazione e sviluppo della comunicazione.
2. Risorse digitali - utilizzo, creazione, valutazione e condivisione delle risorse.
3. Insegnamento e apprendimento - utilizzo di diversi metodi e strumenti nell'insegnamento e nell'apprendimento.
4. Valutazione - controllo dei vari metodi e strumenti per la valutazione e feedback.



5. Sostenere l'apprendimento - utilizzando le tecnologie di produzione per potenziare e rendere gli ambienti di apprendimento più accessibili, inclusivi e focalizzati sull'apprendimento.
6. Sostenere le competenze degli studenti - sostenere e facilitare lo sviluppo delle loro competenze.



Il quadro europeo delle competenze digitali degli insegnanti è un quadro standardizzato che fornisce un riferimento e una guida per le competenze digitali degli insegnanti. Identifica 6 categorie di attività di apprendimento digitale che sono suddivise in specifiche competenze.

Il quadro intende mostrare come le tecnologie digitali possano essere utilizzate per supportare lo sviluppo professionale nell'istruzione e nella formazione. L'enfasi non è solo sulle competenze tecniche, ma soprattutto sul contesto di insegnamento e apprendimento. Il suo scopo è quello di "fornire un linguaggio e una logica comune come punto di partenza per lo sviluppo, la comparazione e il confronto di diversi strumenti per lo sviluppo delle competenze digitali degli insegnanti a livello nazionale, regionale o locale".



A.M.E.F.E
Asociación Madrileña de
Educación y Formación Europea

Taldea



Erasmus+
Enriching lives, opening minds.



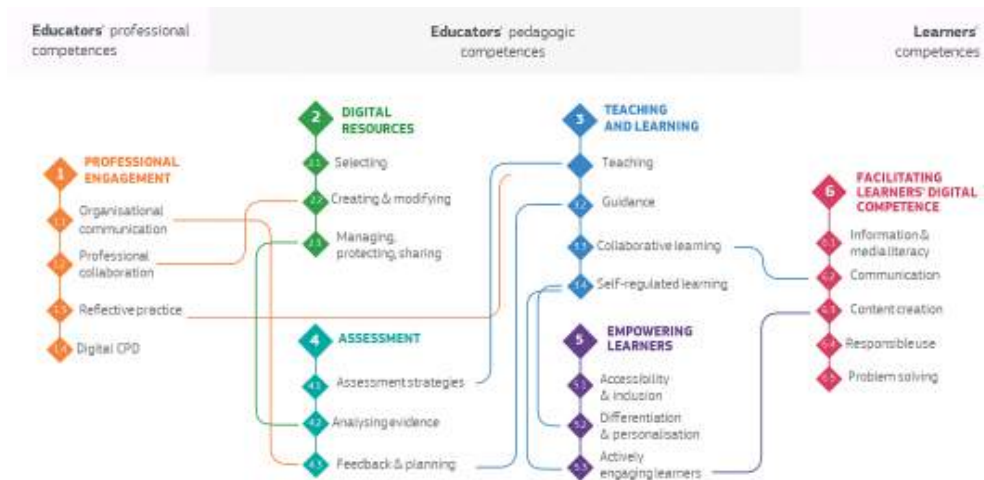
Per meglio comprendere la natura delle competenze digitali, la Commissione Europea ha realizzato il il quadro di riferimento per le competenze digitali dei cittadini (DigComp), suddiviso in 5 aree: Alfabetizzazione su informazioni e dati, comunicazione e collaborazione, creazione di contenuti digitali, sicurezza, risoluzione dei problemi. In totale le aree riguardano 21 competenze.





Il continuo sviluppo delle competenze digitali riguarda sia i bambini che gli insegnanti. Tutti, a prescindere dall'età, acquisendo competenze legate alle nuove tecnologie, trarranno benefici nella società moderna.

È necessario formare i bambini sulla loro responsabilità per l'attuazione e la preparazione alla vita nel mondo moderno, dove i media e le tecnologie si stanno sviluppando rapidamente. È ugualmente necessario ampliare la formazione dei docenti riguardo all'utilizzo delle tecnologie moderne nell'insegnamento in modo da permettere loro di sostenere consapevolmente un processo didattico guidato che incrementi l'efficacia dell'insegnamento e dell'apprendimento.



Per ulteriori approfondimenti, visita questo sito web:
https://joint-research-centre.ec.europa.eu/digcompedu_en



Unità 3 - Gamification

Esistono molte definizioni di gamification / ludicizzazione. Secondo Brian Burke (vicepresidente della ricerca presso la società tecnologica Gartner ed esperto di architettura aziendale) "La gamification è l'uso di meccaniche di gioco e la progettazione dell'esperienza utente per coinvolgere digitalmente e motivare i giocatori a raggiungere i propri obiettivi". L'Oxford Dictionary definisce la gamification come "l'applicazione di elementi tipici del gioco (ad es. punteggio, competizione con altri, regole del gioco) ad altre aree di attività,[...] per aumentare il coinvolgimento". Per il dizionario Cambridge la ludicizzazione è "la pratica di rendere le attività più simili a giochi per renderle più interessanti o divertenti". Secondo Yu-kai Chou (un autore e relatore internazionale sulla gamification e design comportamentale) ci sono 8 unità principali che rendono il gioco coinvolgente per le persone:

1. **Significato epico e call** (nel senso di chiamata a fare) → Fai parte di qualcosa di più grande di te.
2. **Sviluppo e realizzazione** → Sei motivato perché senti che stai migliorando e acquisendo nuove competenze.
3. **Potenziamento della creatività e feedback** → Dare alle persone anche alcuni elementi semplici come i mattoncini da costruzione LEGO® e lasciarli liberi di trovare il proprio modo di combinarli è un processo molto coinvolgente. Come hanno fatto i nostri studenti quando è stato chiesto loro di creare dei "pupazzi" utilizzando materiale plastico e cartaceo di scarto (foto).



A.M.E.F.E
Asociación Madrileña de
Educación y Formación Europea

Taldea



Erasmus+
Enriching lives, opening minds.



4. **Proprietà e possesso** → Quando senti di possedere qualcosa, vuoi migliorarlo, vuoi proteggerlo e vuoi ottenere di più.
5. **Influenza sociale e relazioni** → Siamo influenzati da ciò che fanno le persone intorno a noi.
6. **Scarsità e impazienza** → Si riferisce al fatto che a volte vuoi qualcosa solo perché non puoi averlo.
7. **Imprevedibilità e curiosità** → Se non sai cosa succederà dopo, ci pensi sempre.
8. **Perdita ed evitamento** → Se fai qualcosa, vuoi evitare la perdita, perché nessuno vuole che accada qualcosa di brutto.

Secondo Yu-kai Chou tutto ciò che facciamo è basato su uno o più di questi unità principali. Secondo altri studi i giochi rendono i bambini più intelligenti, perché i videogiochi presentano fondamentalmente un processo continuo di apprendimento per gli utenti. Gli utenti sono in continua evoluzione e vanno avanti. I giochi sono strutturati per produrre una particolare reazione nelle persone. Un gioco rappresenta una sfida e quando lo superi la dopamina viene rilasciata nel tuo cervello, quindi abbiamo un miglioramento dell'apprendimento, il multitasking aumenta le connessioni neurali, si crea un ciclo dopaminergico nel cervello e questo produce un rinforzo intrinseco.



A.M.E.F.E.
Asociación Madrileña de
Educación y Formación Europea

Paidea



Erasmus+
Enriching lives, opening minds.



Dal 2010 la gamification è diventata una metodologia nell'istruzione in tutto il mondo. Studi dimostrano che trasformare gli obiettivi educativi attraverso sfide entusiasmanti, utilizzando il modello del videogioco, ad esempio utilizzando badge di avanzamento o dando la possibilità di vedere i grafici delle prestazioni, soddisfa il bisogno di competenza delle persone in una specifica attività e aumenta il valore percepito del compito. Gamification non significa trasformare la lezione in un gioco, ma introdurre l'elemento del gioco nell'istruzione. I videogiochi possono stimolare nelle persone motivazione, interesse, creatività, senso di appartenenza e felicità. Tutti questi sentimenti sono risorse che possiamo utilizzare nelle nostre attività quotidiane.



A.M.E.F.E.
asociación vallesolista de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.



Quindi, se introduciamo le regole e gli schemi dei videogiochi in tutti i compiti che dobbiamo svolgere nella nostra vita quotidiana, potremmo completarli con più motivazione e soddisfazione, ottenendo risultati migliori.



Unità 4 - La valenza educativa di utilizzare un approccio incentrato sulla tecnologia

L'integrazione di risorse informatiche nell'apprendimento ha numerosi vantaggi: in particolare fornisce modi alternativi di apprendere, permettendo l'acquisizione di competenze cognitive e di conoscenze che sono importanti per l'apprendimento permanente, e promuove la motivazione.

Molti dei posti di lavoro altamente richiesti oggi sono stati creati nell'ultimo decennio, secondo la Società Internazionale per la Tecnologia nell'Educazione (ISTE). Poiché gli sviluppi della tecnologia guidano la globalizzazione e la trasformazione digitale, i docenti dovrebbero aiutare gli studenti ad acquisire le competenze necessarie per eccellere nelle carriere del futuro. La pandemia di Covid-19 ha dimostrato perché l'insegnamento online dovrebbe essere una parte vitale dell'insegnamento e dell'apprendimento. Integrando la tecnologia nei programmi scolastici, e non unicamente come strumento di gestione della crisi, i docenti possono creare delle esperienze educative immersive e coinvolgenti. L'utilizzo di strumenti di apprendimento digitale in classe migliora il coinvolgimento degli studenti, aiuta gli insegnanti a migliorare i propri programmi e facilita l'apprendimento personalizzato. La tecnologia fornisce agli studenti informazioni facilmente accessibili, un apprendimento più veloce e opportunità divertenti per mettere in pratica ciò che hanno appreso. Permette agli studenti di esplorare nuovi argomenti e di approfondire la comprensione di concetti difficili, in particolare relativi alle discipline STEM.



A.M.E.F.E
Asociación Madrileña de
Educación y Formación Europea

Taldea



Erasmus+
Enriching lives, opening minds.



I principali benefici dell'utilizzo della tecnologia nella formazione possono essere così sintetizzati:

- Aumento della collaborazione e della comunicazione: le tecnologie didattiche possono favorire la collaborazione. Non solo gli insegnanti possono interagire con gli studenti durante le lezioni, ma gli studenti possono anche comunicare tra loro. Attraverso attività online e giochi di apprendimento, gli studenti possono lavorare insieme per risolvere i problemi. Nelle attività collaborative, gli studenti possono condividere i propri pensieri e le proprie idee e sostenersi a vicenda.
- Personalizzazione delle opportunità di apprendimento: la tecnologia permette di accedere in qualsiasi momento alle risorse didattiche. Può essere inoltre utilizzata per personalizzare i piani di apprendimento per ogni studente. Gli insegnanti possono creare lezioni in base agli interessi e ai punti di forza degli studenti. Un ulteriore vantaggio è che gli studenti possono imparare al proprio ritmo e rivedere il materiale per ottenere una migliore comprensione dei concetti essenziali.
- Contenuti coinvolgenti per indirizzare la curiosità: attraverso contenuti accattivanti ed educativi, gli insegnanti possono suscitare curiosità negli alunni e aumentarla, che come dimostrato da numerose ricerche influisce sul successo accademico. La creazione di contenuti accattivanti può



comportare l'uso di AR, video, podcast o ovviamente giochi didattici.

L'introduzione di elementi tecnologici e videogiochi nei processi di apprendimento in contesti educativi è sempre più presente nei dibattiti su come migliorare l'esperienza dei giovani in un mondo in cui il digitale viene incorporato in molteplici aspetti della nostra vita. Gli educatori hanno iniziato ad utilizzare i giochi per fini didattici dal 1971, quando tre futuri insegnanti crearono il classico gioco educativo "The Oregon Trail" da utilizzare in un corso di storia degli Stati Uniti.

I primi argomenti a favore di questo approccio erano spesso incentrati sull'idea che "i videogiochi sono divertenti e, come tali, forniscono una prospettiva intrigante per costringere alcuni bambini a imparare" (Silvern 10). Il fatto che i giochi siano efficaci nel motivare i giocatori è rimasto un fondamento logico imprescindibile per l'insegnamento con giochi allineati ai contenuti. Prensky descrive gli studenti che sono diventati maggiorenni nell'era di Internet come "nativi digitali" che naturalmente imparano meglio usando la tecnologia (Prensky, "Listen to the Natives" 9-13).

Gli studi supportano l'idea che i giovani del 21° secolo abbiano una forte affinità per la tecnologia, in particolare per i videogiochi e le reti di comunicazione (Jones et al 6, Lenhart et al 2-3). Altri scrittori attribuiscono il fascino dei giochi al loro coinvolgente mix di fattori tra cui sfida, fantasia e interattività (Malone 162; Owston 978). I videogiochi possono essere uno strumento di apprendimento molto stimolante per i giovani nelle scuole. Oltre ad affrontare determinati contenuti curriculari, possono supportare lo sviluppo di determinate abilità necessarie per promuovere buone abitudini. Proprio questo è l'obiettivo del progetto PLAY: trovare un modo efficace per sensibilizzare i giovani studenti sulla necessità di preservare il nostro pianeta, coinvolgendoli direttamente e facendogli sviluppare competenze tecniche e trasversali.



A.M.E.F.E.
Asociación Madrileña de
Educación y Formación Europea

Taldea



Erasmus+
Enriching lives, opening minds.

Unità 5 - Risultati del progetto

Gli studenti coinvolti nel progetto si sono avvicinati ai linguaggi di programmazione utilizzando Scratch (linguaggio visuale) e poi Unity (linguaggio testuale - C#) al fine di creare il videogioco. Questo li ha aiutati a migliorare le loro competenze digitali e allo stesso tempo ha contribuito ad aumentare il loro interesse e la loro conoscenza relativa al riscaldamento globale, all'inquinamento e all'ecosostenibilità. Gli alunni sono stati stimolati a trovare soluzioni più ecologiche e sostenibili per combattere il riscaldamento globale e l'inquinamento. Creare un videogioco che includesse argomenti di studio, come appunto il riscaldamento globale e l'inquinamento, si inserisce in quella strategia nota come gamification. Non si tratta di una metodologia utilizzata unicamente a scuola, ma anche in contesti lavorativi per stimolare le persone a trovare soluzioni creative per risolvere i problemi e migliorare la motivazione dei lavoratori per il raggiungimento di un obiettivo specifico.

Durante le mobilità gli studenti hanno migliorato la loro conoscenza dell'inglese, sono entrati in contatto con tradizioni e culture di altri paesi aumentando la loro comprensione reciproca, l'empatia e l'accettazione della diversità. Studenti e insegnanti hanno visitato nuovi posti, hanno scambiato conoscenze e competenze e hanno migliorato la loro capacità di lavorare in squadra e di cooperare. Questo progetto ha aiutato studenti e insegnanti a sviluppare una rete di relazioni con istituti educative locali e con scuole di altri paesi che dureranno oltre la fine del progetto.



Modulo 2.

Unità 1 - Unity 3D Platform: usi e potenzialità.

INTRODUZIONE.

L'avanzamento della tecnologia è sorprendente. Chi avrebbe mai immaginato 100 anni fa quello che saremmo stati capaci di fare oggi con il supporto degli strumenti tecnologici.

Uno di questi strumenti è Unity 3D, che ci dà la possibilità di realizzare ambienti del mondo reale. Vorremmo sottolineare che, dopo aver letto questo toolkit, l'utente sarà in grado di costruire il proprio tour virtuale dello scenario di propria scelta, purché dia spazio alla propria immaginazione e dia il proprio tocco artistico al progetto desiderato. Il progetto presentato all'interno del toolkit è stato ideato per persone che non hanno nessuna conoscenze di Unity 3D perciò, se vuoi imparare ex novo sei nel posto giusto, noi inizieremo dalla configurazione base dell'ambiente di sviluppo in Unity 3D, spiegando qual è la funzione di ogni pulsante ed ogni menù del programma, garantendo una conoscenza preliminare di queste applicazioni, prima di iniziare a sviluppare il tour virtuale. Verrà approfondita la gestione delle dimensioni in Unity 3D, perché come indica il suo stesso nome utilizza le 3 dimensioni (x, y, z) e quando non si ha dimestichezza nella gestione del programma questo può risultare noioso, perché è una questione di abitudine imparare ad utilizzare queste dimensioni. Dopo aver approfondito l'uso degli strumenti, inizieremo con la creazione del nostro progetto e se ci fossero dubbi su alcuni pulsanti questi saranno chiariti man mano che procederemo. Quando la creazione del primo spazio o tour virtuale è completata, gli studenti saranno in grado di creare il proprio ambiente. Non si può immaginare quanto sia potente questo strumento e quanto sia facile da usare. Grazie a questo toolkit avrete la possibilità di sviluppare questo tour passo dopo passo in modo da poter raggiungere gli obiettivi desiderati.

UNITY 3D:

Unity 3D è uno strumento per sviluppare principalmente videogiochi, pochi sanno come utilizzarlo e non c'è bisogno di essere un esperto in programmazione per



riuscire ad utilizzarlo, perchè con poche nozioni del linguaggio richiesto, è possibile strutturare qualsiasi gioco interattivo. “Unity 3D, un motore grafico 3D per PC e Mac, è utilizzato per sviluppare giochi, applicazioni interattive, visualizzazioni e animazioni 3D. Unity supporta piattaforme come PC, Mac, Nintendo, Wii, Iphone, Android e il web utilizzando il plugin “Unity web player”. Unity è un’applicazione realizzata da Unity Technologies che è stata fondata nel 2004 da David Helgason (CEO), Nicholas Francis (CCO), e Joachim Ante (CTO) a Copenhagen, in Danimarca dopo che il loro primo gioco, GooBall, non ebbe successo. I tre riconobbero il valore del motore e degli strumenti di sviluppo e stabilirono di realizzare un motore che potesse essere utilizzato da tutti ad un prezzo ragionevole.

Unity Technologies ha ricevuto finanziamenti da Sequoia Capital, Capital WestSummit e iGlobe Partners. Il successo di unity è dovuto in parte dal focus sui bisogni degli sviluppatori indipendenti che non possono né realizzare il proprio motore di sviluppo né avere gli strumenti necessari o le licenze per utilizzare pienamente le opzioni che sono disponibili sul mercato. Il focus dell’azienda è “rendere democratico lo sviluppo dei giochi”, e rendere lo sviluppo di contenuti interattivi in 2D e 3D il più possibile accessibile al maggior numero di persone nel mondo. Con l’aumento degli iPhone nel 2008, Unity è stato uno dei primi motori a iniziare a supportare questa piattaforma e oggi è utilizzato dal 53.1% degli sviluppatori secondo un’intervista condotta da Game Developer mobile and social technology, per creare centinaia di giochi per dispositivi Android e iOs. Unity ha un editor visuale per riuscire a creare giochi, perché tutto il contenuto del gioco è costruito da questo editor e il mondo in cui gli oggetti si comportano è programmato utilizzando un linguaggio script (JavaScript); quanto detto ci dà la possibilità di capire che non bisogna essere esperti in linguaggi come C++ per essere capaci di sviluppare un gioco o un’animazione con Unity 3D. Unity è strutturato gestendo e creando scene per lo sviluppo dell’applicazione desiderata, una scena può essere una qualsiasi parte di un gioco o di un’animazione: sia un livello del gioco sia un certo ambiente.

Si inizia con un spazio vuoto in cui è possibile modellare qualsiasi cosa si voglia creare utilizzando gli strumenti di unity. Questo motore di unity contiene anche un



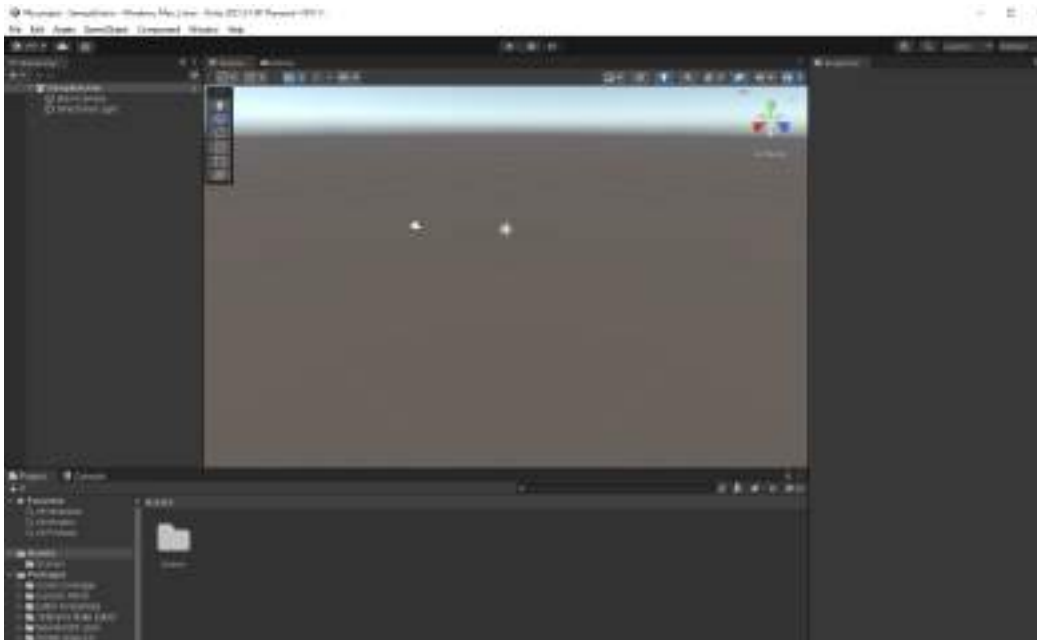
editor del terreno, dove è possibile modellare la forma del terreno utilizzando gli strumenti visuali offerti da unity, è possibile disegnare, aggiungere texture, aggiungere erba, posizionare alberi o oggetti simili, o anche importare altri materiali da altri motori di sviluppo. Unity è utilizzabile da ogni tipo di pubblico, poichè è sviluppato in due versioni: gratis e professionali. Entrambe sono vantaggiose per lo sviluppo, tuttavia, la versione più completa è quella professionale, ma ha un costo elevato e per un utente alle prime armi la versione gratuita è più che sufficiente per il momento. Oltre a Unity 3D, ci sono altri strumenti sul mercato, che potrebbero anche essere più famosi, come UDK, Epic Games o CryEngine. Tuttavia, Unity 3D ha un grande vantaggio e cioè che non siamo obbligati a sviluppare su sistemi operativi Windows, poiché Unity 3D ha anche una versione per sistemi operativi Mac. Unity fornisce anche la possibilità non solo di importare terreni, ma anche modelli 3D, texture, suoni, con pochi passaggi che possono essere effettuati in qualsiasi momento durante lo sviluppo.



Unità 2 - Interfaccia

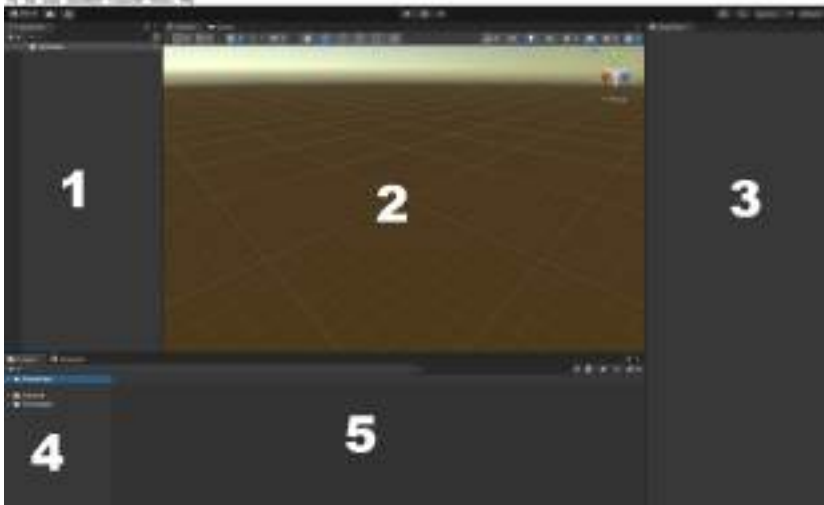
1. PRIMO ACCESSO

Ci sono due versioni di unity, una gratuita e una professionale. Questo manuale si baserà sulle funzionalità accessibili ai principianti nella versione gratuita. La versione gratuita può essere trovata sul sito web di Unity. Una volta che l'applicazione è stata installata, è possibile accedere ad una *demo* dove vedere le caratteristiche di Unity e dove nuovi utenti possono scoprire le funzionalità che offre studiando le creazioni dei suoi sviluppatori.



2. INTERFACCIA UTENTE

In questa parte del manuale vedremo da cosa è composta l'interfaccia utente di Unity. Ci sono principalmente 5 aree dell'interfaccia, elencate nell'immagine sottostante:



- Visualizzazione gerarchica (*Hierarchical view*). Dove ci sono tutti gli oggetti della scena corrente.
- Visualizzazione scene (*Scene view*). Questa è l'area dove noi costruiamo visivamente ogni scena. Il modo più semplice di utilizzare è quella di trascinare gli oggetti *project view* alla *scene view* che metterà gli oggetti nella scena; è quindi possibile posizionarli, scolarli e ruotarli senza lasciare la *scene view*. La *scene view* è anche il luogo in cui si modificano i terreni (scolpendoli, dipingendo la texture e posizionando elementi), posizionando luci e telecamere e altri oggetti.
- Menù impostazioni (*Inspector view*). Questo menù ha molte funzioni in base a che gli utenti selezionano, se vogliono visualizzare le caratteristiche di un oggetto o un terreno. Se si seleziona "telecamere", verranno mostrate le proprietà in cui è possibile personalizzare funzioni come rotazione, posizione o scala.
- Visualizzazione del progetto o gestione degli oggetti (*Project view*). Questa è la libreria per il progetto, simile ad una raccolta di strumenti. È possibile importare oggetti 3D da diverse applicazioni nella raccolta, importare texture e creare altri oggetti che è possibile memorizzare per l'uso futuro nel progetto. Un progetto normale conterrà diverse ambientazioni e una grande quantità di risorse, quindi è necessario strutturare la raccolta in diverse cartelle che renderanno le risorse organizzate e più facili da usare.



- Barra di ricerca (*Search view*). È usata per localizzare gli oggetti che dobbiamo utilizzare nella realizzazione del progetto.

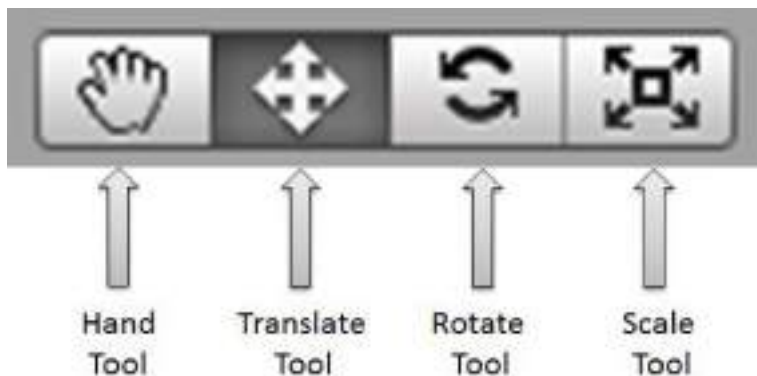
3. MENU DELLE APPLICAZIONI

Il menu delle applicazioni che si trova in alto a sinistra sullo schermo è mostrato di seguito. All'interno del manuale, sarà mostrata la funzione di ognuna delle sezioni.



3.1 CONTROLLO DEI PULSANTI DI CONTROLLO

Di seguito le opzioni di visualizzazione, vedrai una riga con 4 pulsanti di controllo che puoi usare Q, W, E, R per alternare tra ciascuno dei controllo che sono spiegati di seguito:



- **Hand Tool (Q)**: Questo controllo ci permette di muovere inella *scene view*.

ALT ci permette di ruotare.

CTRL ci permette di aumentare e ridurre *zoom*.

SHIFT aumenta la velocità di movimento durante l'utilizzo di uno strumento.

- **Translate Tool (W)**: ci permette di muovere qualsiasi oggetto selezionato nella scena degli assi X,Y e Z.

- **Rotate Tool (E)**: ci permette di ruotare qualsiasi oggetto nella scena.

- **Scale Tool (R)**: ci permette di scalare qualsiasi oggetto selezionato nella scena.

Quando si utilizzano gli strumenti di conversione, rotazione o scalabilità, si vedrà un'icona intorno all'oggetto selezionato con delle linee per ciascuno degli assi che



A.M.E.F.E.
asociación española de
educación y formación europea

Taldea



Erasmus+
Enriching lives, opening minds.

chiameremo gizmo Possiamo usare questo gizmo per eseguire operazioni di conversione, rotazione e scalabilità.

3.2 PULSANTI DI RIPRODUZIONE

In Unity è possibile eseguire il gioco senza lasciare l'editor, ciò è un vantaggio per i designer durante la costruzione dei livelli e sviluppatori che aggiungono nuovi meccanismi di gioco.



L'immagine mostra i controlli di riproduzione, che sono posizionati in alto nell'editor. È possibile riprodurre il gioco in qualsiasi momento premendo il pulsante play per vedere lo status del progetto (il primo a sinistra), mettere in pausa utilizzando il bottone pausa (quello centrale) o andare avanti con il pulsante destro. Puoi giocare dal menù del gioco o estendere al massimo la schermata.

3.3 SCRIPT

Gli script consentono la definizione della logica di gioco. Nel toolkit vedrai diversi script semplici come definire un movimento, accendere o spegnere le luci. L'operazione è molto semplice, prima si definisce il comportamento desiderato in uno script e questo viene aggiunto come componente all'oggetto al quale si desidera assegnare tale comportamento, è anche possibile aggiungere uno script a un oggetto vuoto. Ci sono 3 funzioni di base per gli script:

- **Funzione Update():** funzione che definisce il ciclo principale del gioco in modo che venga eseguito una volta per ogni fotogramma reso.
- **Funzione Awake():** le variabili definite all'interno di questa funzione devono essere inizializzate poco prima che inizi il gioco. Questa funzione deve essere eseguita una sola volta per tutta la durata *script instance*.
- **Funzione Start():** Questa funzione è molto simile a "Awake()" ma ha una differenza vitale; verrà eseguita solo se *script instance* è abilitata. Questo ci permetterà di ritardare l'inizializzazione di alcune variabili di gioco. Dobbiamo ricordare che la funzione "Start()" sarà sempre eseguita dopo che la funzione "Awake()" sarà terminata.

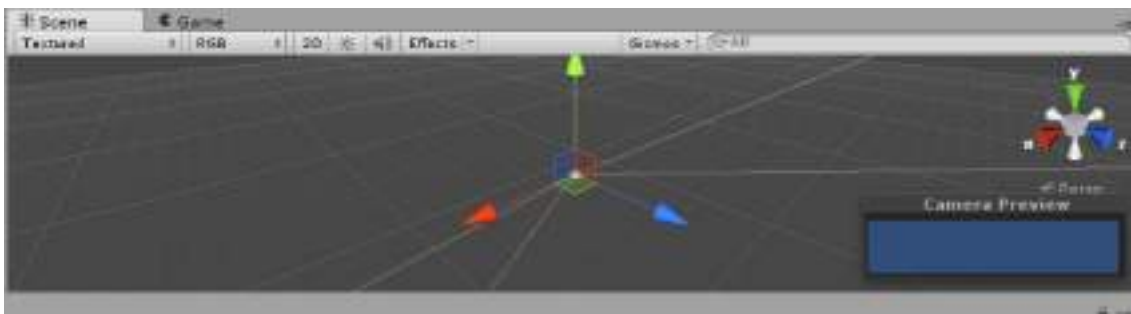


Unity 3D ha anche un editor di script predefinito chiamato uniscite.

```
rotation.js - UnisCITE
File Edit Search View Options Language Buffer Help
rotation.js
var rotationSpeed: float = 2.0;
function Update () {
  transform.Rotate(Vector3(0,rotationSpeed,0));
}
```

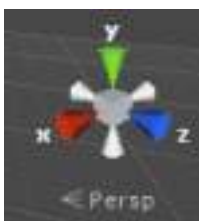
4. AVVIARE UNITY 3D

La prima cosa da fare è avviare Unity 3D. A questo punto si aprirà un progetto vuoto. Se si vuole inserire un'altra scena basta nel menu: "File -> Apri scena" e guardiamo nella directory del progetto ("Assets/scenes"). Per impostazione predefinita, la visualizzazione della scena è in prospettiva 3D.



Prospettiva

È possibile modificare la *scene view* in vari punti vista : dall'alto verso il basso, laterali e anteriori. Sul lato destro della *scene view* vedrai un "Gizmo" che sembra una scatola con i coni che ne escono.



- Con questo Gizmo possiamo cambiare la prospettiva della visualizzazione.
- Cliccando sulla scatola si passerà in modalità prospettiva.



- Cliccando sul cono verde (y) si passerà alla modalità Top-Down (dall'alto verso il basso).
- Cliccando sul cono rosso (x) si passerà alla modalità laterale (destra).
- Cliccando sull'icona blu (z) si passerà alla modalità frontale
- Abbiamo anche 3 coni grigi che ci porteranno alle seguenti modalità: "Indietro, a sinistra e in basso".

4.1 HIERARCHY VIEW

La *hierarchy view* contiene tutti gli elementi che compongono la scena di gioco. Puoi manipolare tutti gli oggetti nella scena da questo menù, ogni volta che inserisci un elemento nella scena, una voce per questo elemento viene aggiunta nel menù. Selezionando un oggetto in *hierarchy view* lo si seleziona anche nella scena e nell'inspector. Ciò consente e facilita lo spostamento, la scalabilità, la rotazione e l'eliminazione dell'oggetto o la modifica dei suoi parametri. Tutto questo senza toccare una sola riga di codice.

Per inserire un nuovo oggetto nella scena è possibile fare clic su "CREA" e verranno visualizzati tutti gli oggetti che si desidera aggiungere.



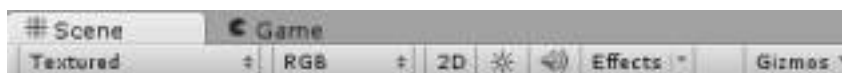


4.2 SCENE VIEW

La *scene view* è la parte dell'ambiente di grafica 3D per creare ogni scena. Per lavorare nella *scene view* il modo più semplice è trascinare un oggetto dalla *project view* alla *scene view*, ciò collocherà l'oggetto nella scena, e quindi possibile posizionarlo dove si preferisce.

La *scene view* è anche il luogo in cui si modificano i terreni (scolpendoli, dipingendo texture e posizionando elementi), posizionando luci, telecamere e altri oggetti. Di seguito verranno spiegate le opzioni di visualizzazione della scena:

Sul lato sinistro della *scene view* troverete una serie di pulsanti per modificare le impostazioni generali di visualizzazione. Diamo un'occhiata a ciascuno di essi, da sinistra a destra:



Modalità di rendering

Il valore predefinito sarà "Textured". Se cliccato, apparirà un elenco a discesa con un numero di diverse opzioni di rendering:

- Textured: Le texture sono renderizzate nel menù.
- Wireframe: Le superfici non sono renderizzate, vediamo solo la mesh.
- Textured wire: Le texture sono renderizzate, ma vediamo anche la mesh.
- Render Paths: modifica le rappresentazioni e le caratteristiche degli oggetti in base ai progetti creati.
- lightmap Resolution: per dare una risoluzione più ottimale agli oggetti che compongono la scena.





Modalità colore, visualizzata come "RGB".

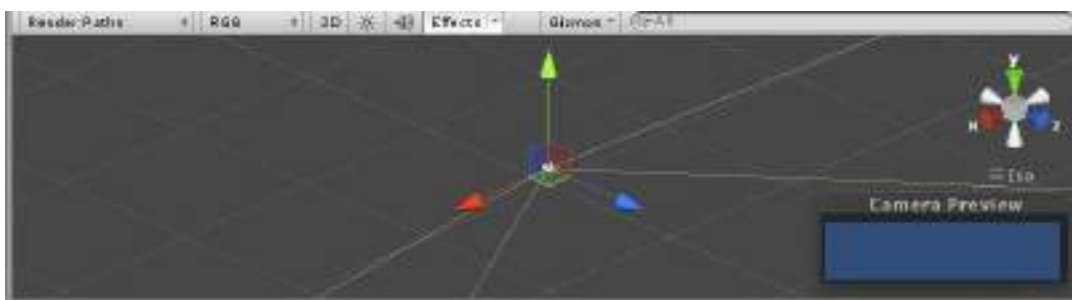
Se clicchi su questo pulsante, apparirà un elenco a discesa che mostra le modalità di colore disponibili e sarai in grado di vedere i diversi modi di visualizzazione della scena:

- RGB: sono renderizzati tutti i colori.
- Alpha: la modalità viene cambiata in "Alpha".
- Overdraw: la modalità viene cambiata in "Overdraw".
- Mipmaps: la modalità viene cambiata in "Mipmaps".
-



3D-2D option

Dove cambiamo il display della scena da un piano X, Y, Z (tridimensionale) a un piano di osservazione X, Y (bidimensionale).





A.M.E.F.E.
Asociación Madrileña de
Educación y Formación Europea

Taldea



Erasmus+
Enriching lives, opening minds.



Interruttore della luce

Il pulsante successivo attiva o disattiva l'illuminazione della scena.



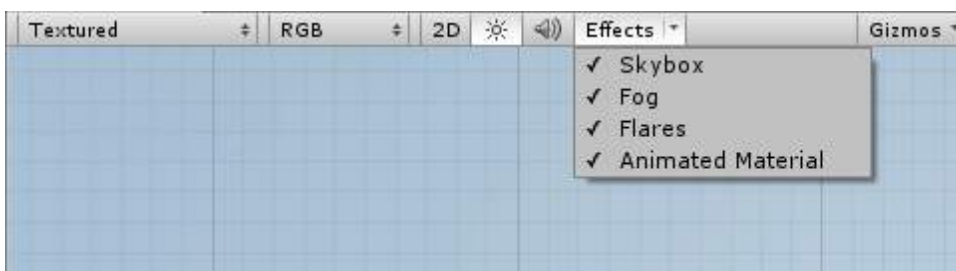
Audio

Il pulsante successivo attiva o disattiva l'audio della scena.



Effetti

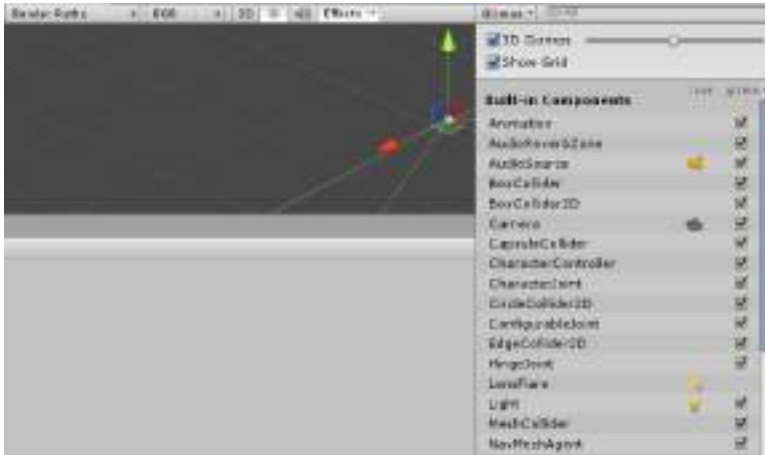
Il pulsante effetti viene utilizzato per disabilitare gli effetti video per migliorare le prestazioni durante la fase di progettazione.





Gizmos

Aiutano a perfezionare la grafica della scena anche quando si utilizzano linguaggi di programmazione.



4.3 GAME VIEW

La game view ti permette di vedere in anteprima il gioco in una finestra ridotta o massimizzandola a schermo intero. In questo menù è possibile vedere il gioco in movimento per verificare che il gioco funzioni correttamente.

La Game view è molto utile ma ha un inconveniente, quando si prova il progetto sul computer, si muove con i tasti o il mouse che non aiuta molto se il progetto è in fase di sviluppo per un'altra piattaforma, ad esempio un dispositivo mobile. In questo caso i test devono essere eseguiti utilizzando un emulatore o compilando una versione mobile.



- Il pulsante "Maximize on Play" : consente di massimizzare la



visualizzazione a schermo intero del gioco.

- Il pulsante "Stats":
- Il pulsante "Gizmos":
- Free Aspect" a discesa:

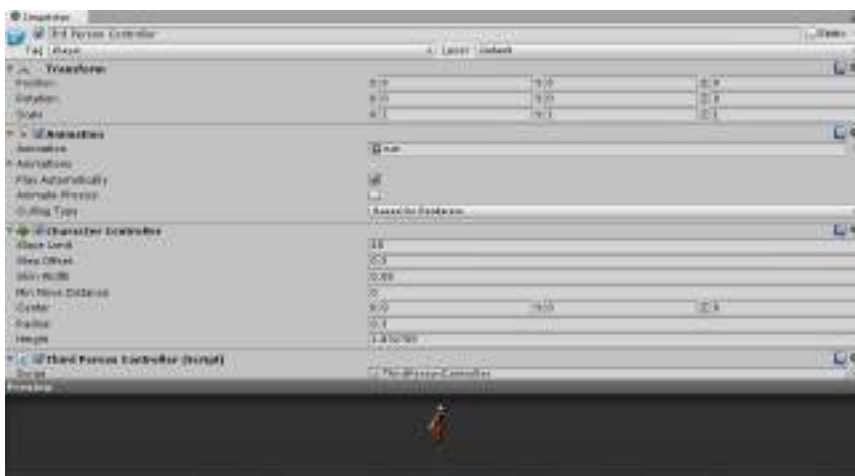
4.3 INSPECTOR'S VIEW

Nella inspector view, sono mostrati i parametri degli oggetti selezionati nella scene view o nella hierarchy view. In questo modo quando si seleziona un oggetto nella scena, si possono modificare i parametri nella inspector (posizione, rotazione, conversione, etc).

L'inspector è utilizzato anche come pannello di strumenti per alcuni elementi come i terreni poichè di scolpirli, aggiungere texture e altro ancora.

Nella inspector view si possono vedere gli elementi che definiscono il comportamento degli oggetti, questi elementi sono chiamati componenti. Inoltre, gli oggetti e i loro componenti associati possono essere attivati o disattivati qui. I componenti possono essere script, animazioni, collisori...ecc.

Infine, l'inspector cambia a seconda dell'elemento selezionato.

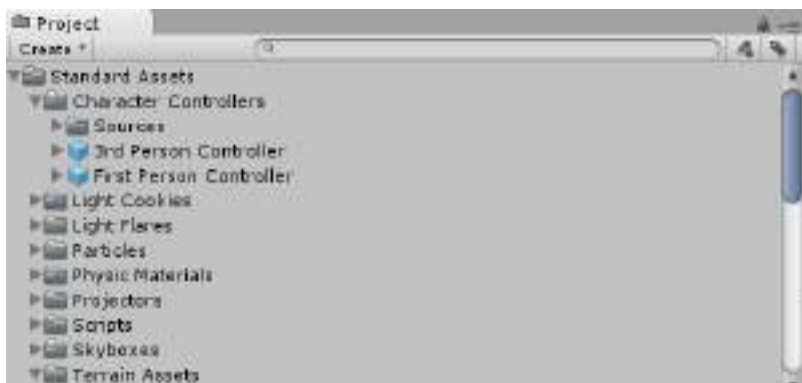


4.3 PROJECT VIEW



La project view è una libreria di risorse(asset) per la creazione del gioco. In questa view sono memorizzati tutti gli elementi che sono creati e importati per essere utilizzati nel gioco. Questi elementi possono essere oggetti 3D, texture, suoni e script.

In questa finestra è possibile avere una moltitudine di risorse ma questo non implica l'utilizzo di tutte risorse per la creazione del gioco.



Poiché questo menù contiene sia gli elementi utilizzati nella scena che quelli non utilizzati nella scena, potrebbe contenere una grande quantità di dati. Ciò implica che per facilitare il lavoro dell'utente deve essere mantenuta una buona struttura. L'organizzazione di questa vista può essere fatta creando una gerarchia di cartelle. Il menù a tendina "create" contiene le opzioni che consentono una buona organizzazione della scena.

5. CREARE UN NUOVO PROGETTO.



Per creare un nuovo progetto bisogna passare alla barra superiore di Unity e cliccare su File -> Nuovo progetto e aprire la finestra di dialogo "Create new project".

Per prima cosa occorre selezionare la cartella in cui si vuole salvare il progetto cliccando su "Browse...". Per creare un nuovo progetto, si devono importare i pacchetti necessari.

I pacchetti sono insiemi di risorse e possono essere importati durante la creazione del progetto o in qualsiasi momento successivo.

I pacchetti importati possono essere quelli predefiniti in Unity, altri creati dall'utente o presi dal Web, alcuni di essi sono scaricabili gratuitamente e altri a pagamento.

Dalla lista che appare sul sito web, selezionare le risorse e fare clic su "download".

Per gli asset che si trovano all'interno del programma, basta selezionare quelli che vogliamo e clicchiamo su "Crea". Il pacchetto base da importare è il set standard di asset.

Quando il nuovo progetto viene creato, Unity 3D viene riavviato creando la struttura del progetto nella cartella specificata. Si apre la schermata iniziale di Unity 3D che mostra le risorse importate nella project view e una telecamera nella scene e hierarchy view.

Alcuni dei pacchetti più importanti sono:

- Character Controller : consente di creare script per spostare il personaggio.
- Particles: consente la creazione di particles come fuoco, fumo, cascate ed esplosioni.
- Skyboxes: consente la creazione di cieli.
- Terrain: consente di creare e scolpire il terreno.
- Water: per creare acqua (mare, fiumi, ecc).
- Tree: per creare alberi.



5.1 CREARE UNA NUOVA SCENA

Unity 3D funziona per scene e quando creiamo un nuovo progetto crea automaticamente una nuova scena, se vogliamo creare un'altra scena, andiamo su File _ New Scene e la salviamo automaticamente in File _ Save scene. La scena viene salvata nella cartella degli asset del progetto e apparirà in project view (le cartelle possono essere trovate in Documents/New Unity Project o con il nome della scena).

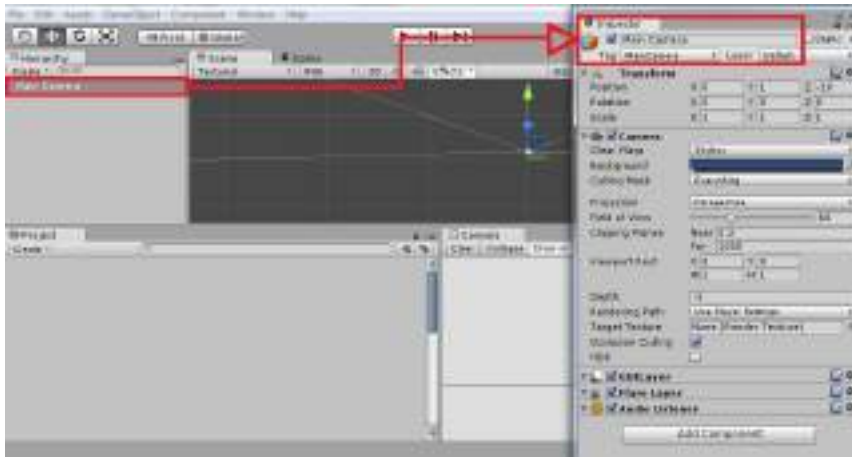
Una volta salvata la scena possiamo creare tutti gli elementi che la compongono, e abbiamo diversi modi per gestirla.

Alcuni trucchi per la gestione della scena sono:

- Se si vuole vedere la scena da diverse angolazioni possiamo ruotare usando ALT e muovendo il mouse.
- Se si vuole aumentare o diminuire lo zoom si può utilizzare la rotellina del mouse.
- Le frecce sulla tastiera consentono di spostare la scena a sinistra, destra, su e giù. Lo stesso può essere fatto tenendo premuto il mouse e spostandolo in una direzione o nell'altra.
- Per aumentare la velocità di movimento della scena premere SHIFT.

Quando si crea una scena, appare di default una telecamera nel nostro mondo. Se clicchiamo su "Main Camera", le caratteristiche della telecamera possono essere visualizzate nell'inspector e possono essere modificate (posizioni, rotazione, scala).

La telecamera definisce il punto di vista del giocatore e diverse telecamere possono essere combinate nella stessa scena.

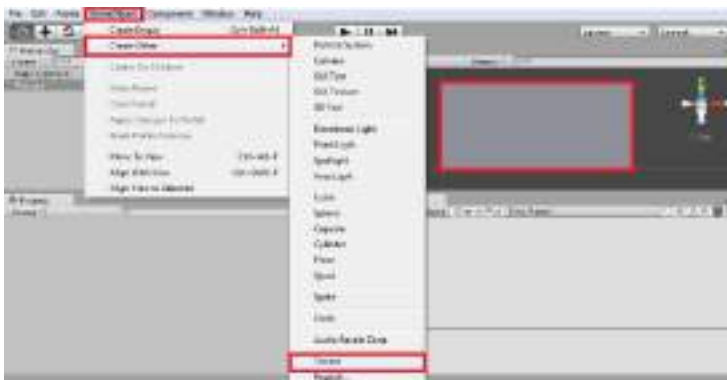


5.2 CREARE IL SUOLO

Unity 3D ha uno strumento con cui è possibile definire qualsiasi terreno, normalmente gestisce terreni come una mesh piatta, ma è possibile modellare la texture e scolpire senza lasciare l'editor. Dal menu è possibile creare un terreno cliccando su:

- "Terrain -> Create Terrain".

Si ottiene un terreno pianeggiante, se il terreno non è visibile è necessario disattivare le luci nella scene view.



Una volta creato il terreno, si ottiene il seguente risultato:

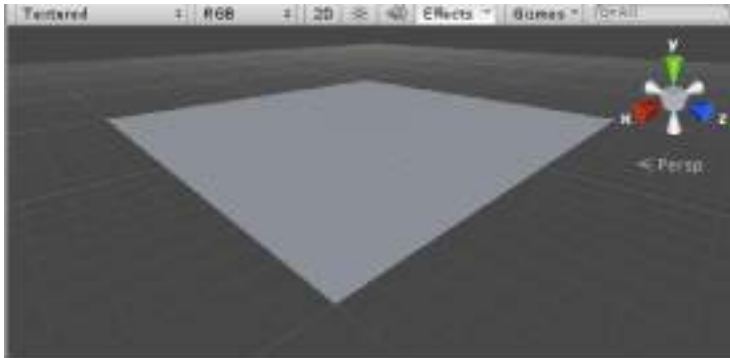


A.M.E.F.E.
asociación española de
educación y formación europea

Taldea



Erasmus+
Enriching lives, opening minds.



Se vogliamo personalizzare il terreno e renderlo più realistico, dobbiamo utilizzare lo strumento *terrain*. Il terreno viene selezionato nella scene view o nella hierarchy view in modo che lo strumento di modifica del terreno appaia nella inspector view.

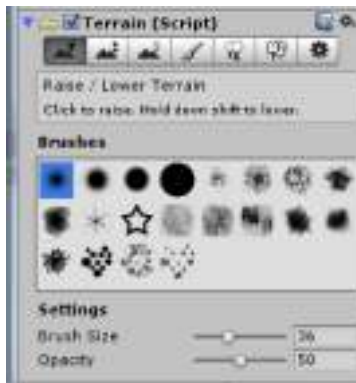


Questo strumento è diviso in 3 parti:

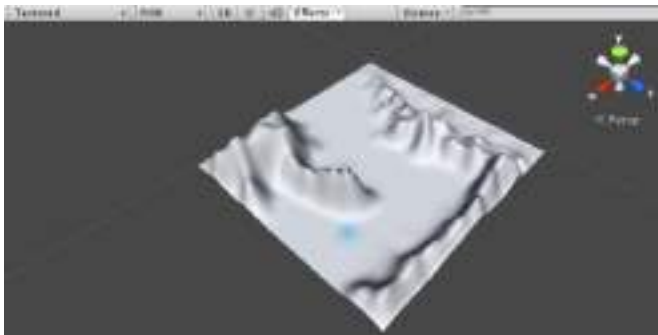
- Transform: consente di spostare, ruotare e scalare il terreno sugli assi x, y, z.
- Terrain Script: contiene diversi strumenti e proprietà per il terreno che verranno spiegate in seguito.
- Terrain Collider: contiene le proprietà di collisione del terreno.
- Possiamo procedere a dettagliare del pannello "Terrain Script" e vedere una fila di pulsanti. Questi pulsanti dell'editor del terreno permettono di eseguire diverse attività. La descrizione di ciò che i pulsanti permettono di fare da sinistra a destra è:
 - Raise / Lower: consente di alzare e abbassare la geometria del terreno utilizzando un pennello.
 - Set Height: dipingiamo il terreno con un limite di altezza.



- Smooth: consente di levigare un terreno per eliminare gli angoli.
- Paint Texture: permette di dipingere texture sulla superficie del terreno.
- Place Trees: permette di posizionare gli alberi.
- Paint Detail: consente di disegnare dettagli del terreno come l'erba.
- Terrain Settings: Accesso alle proprietà del terreno dove possiamo cambiare diverse proprietà.



Per prima cosa occorre definire le altezze del terreno facendo clic sui luoghi in cui si desidera posizionare una montagna, più a lungo si preme il mouse più grande diventerà la montagna. Il terreno può essere trasformato nella seguente immagine:



La seconda opzione "Set Height" ti permette di impostare un'altezza massima nel campo "Altezza".



A.M.E.F.E.
Asociación Madrileña de
Educación y Formación Europea

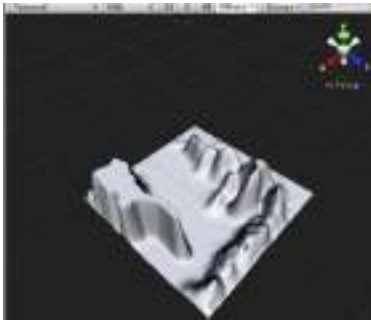
Taldea



Erasmus+
Enriching lives, opening minds.



Impostando un'altezza massima, il terreno può essere ulteriormente modellato per dargli la giusta geometria. Una volta raggiunta l'altezza massima, il terreno viene appiattito. La terza opzione "Smooth" consente di levigare le cime del terreno. Si consiglia di scolpire per gradi, prima le parti grandi e poi affinare i dettagli più piccoli. Alla fine si ottiene un terreno come mostrato nella foto:

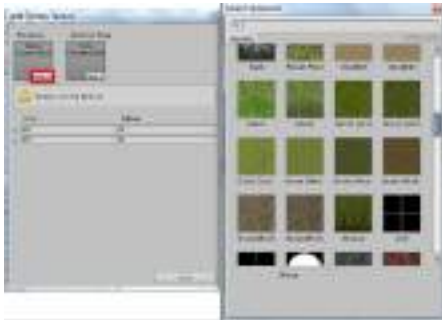


Una volta ottenuto il terreno con l'aspetto desiderato, è possibile applicare diverse texture. La prima texture è considerata quella di base e viene applicata all'intero terreno. Le altre texture verranno applicate negli strati più alti.

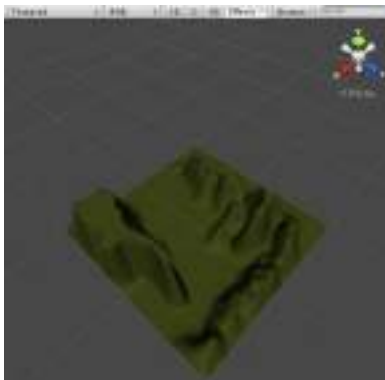
La quarta opzione "texture: paint the terrain texture" permette di aggiungere la texture al terreno. Per prima occorre scegliere il tipo di pennello da utilizzare nella sezione "Pennelli", poi si deve aggiungere la texture selezionata in "Texture", cliccando su "Edit texture> Add Texture".



Si aprirà una finestra con “Add terrain texture”, clicca sul cerchio evidenziato in rosso per aprire la finestra “Select Texture 2D”, si potrà scegliere una texture che apparirà nel campo “Splat” e nel campo “Texture” nella inspector view. Si può anche scegliere la profondità e la lunghezza della texture.



Quando le texture sono state applicate, compariranno sul terreno, che apparirà come mostrato nell’immagine in basso:



Unity 3D offre un supporto speciale per l’aggiunta di alberi. Si possono



aggiungere quanti alberi si vogliono utilizzando la quinta opzione "Places tree". Prima occorre utilizzare l'opzione "Edit trees" per scegliere l'albero che si vuole aggiungere al proprio mondo, poi è possibile personalizzarlo definendo vari parametri, la dimensione del pennello con cui posizionare l'albero, la densità degli alberi (determina il numero di alberi da posizionare), la variazione di colore (differenza di colore tra un albero e l'altro), l'altezza e la larghezza dell'albero.

Per finire, occorre cliccare su un punto qualsiasi del terreno per aggiungere l'albero nella posizione selezionata. Per eliminare gli alberi, bisogna tenere premuto il tasto Shift e cliccare sul terreno.



Puoi anche aggiungere diversi alberi in blocco al mondo realizzato per creare una foresta in una volta sola. Clicca su "Terrain->MassPlace Trees" e apparirà una finestra di dialogo in cui si potrà determinare il numero di alberi che si vuole introdurre nel gioco.

Un albero può essere selezionato andando nella cartella "Standard assets > Tree Creator > Textures", occorre cliccare e trascinare sul terreno.





5.3 LUCE

Se si vuole illuminare la scena con la stessa intensità basta aggiungere punto luce e posizionare le luci desiderate sulla scena: create>point light



Se si desidera un'illuminazione di alta qualità, Unity integra la tecnologia Beast, che è un software Autodesk che genera mappature dell'illuminazione di alta qualità.

"Create Lightmap" viene utilizzato per diverse creazioni di luce che consentono un'illuminazione varia per diverse texture del terreno.

Accedendo a Windows>Lightmapping apparirà la seguente finestra:





A.M.E.F.E.
Asociación Española de
Educación y Formación Europea

Taldea



Erasmus+
Enriching lives, opening minds.



Per controllare la risoluzione delle mappe di luce, il valore di risoluzione viene impostato nel pannello Bake e applicato automaticamente.



Il lightmapping migliora la definizione sul terreno come un'area di alberi e le loro ombre.

5.4 AMBIENTE DEL TERRENO

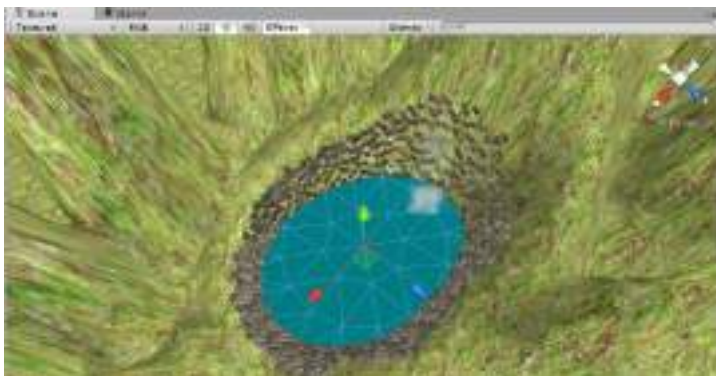
L'ambiente del terreno aiuterà l'utente a personalizzare l'ambiente in cui si trova la scena che sta costruendo, ad esempio aggiungendo il cielo o l'acqua.

5.4.1 AGGIUNGERE ACQUA ALLA SCENA

Unity include diversi modi per aggiungere acqua alla scena, l'utente può farlo ad esempio in maniera diretta. Due modi diretti sono il "Bake de lightmapping" e il "Nighttime Simple Water".



Come esempio aggiungeremo "Daylight Simple Water" all'interno del terreno. Occorre andare su "Asset standard> Water", selezionare "Daylight Simple Water" e trascinarlo nella scena.



Le caratteristiche dell'acqua possono essere viste nell'inspector view:



5.4.2 AGGIUNGERE IL CIELO

È possibile aggiungere il cielo in due modi,collegando uno skybox alla telecamera o direttamente alla scena. Bisogna andare su "Edit > Render Settings" per attivare i campi nell'inspector view come segue:



Questo menu consente di configurare il cielo della scena, gli effetti di nebbia e alcuni effetti di illuminazione globale. Puoi importare le tue texture o utilizzare quelle predefinite dagli asset di Unity. Ci sono 2 opzioni per aggiungere il cielo alla scena:

Opzione 1: andare nel menu Render Settings, cliccare sull'icona del piccolo disco a destra del campo Skybox Material, si aprirà una nuova finestra con tutte le risorse materiali caricate nel progetto.



Opzione 2: nella project view in "Standard Asset > Skybox", selezionare il cielo appropriato per la propria scena e trascinarlo su "Skybox Material". Per questo risultato è stato scelto il cielo "Sunny 1 Skybox".



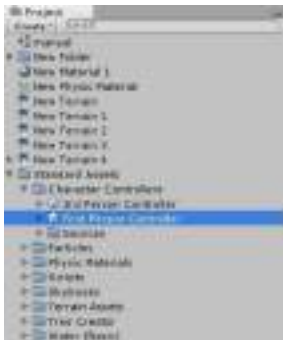
Unity 3D offre diverse opzioni per personalizzare l'ambiente della scena, tra cui:

- Fog: consente di aggiungere nebbia alla scena, la scena diventa annebbiata.
- Fog Color: il colore della nebbia, il colore predefinito è il grigio.
- Fog mode: lineare, esponenziale o esponenziale al quadrato. Controlla come svanisce la nebbia.
- Fog Density: densità della nebbia, usata solo per esponenziale ed esponenziale al quadrato. Una densità maggiore ridurrà la visibilità e una densità inferiore aumenterà la visibilità.
- Linear fog start/end: inizio e fine delle distanze di nebbia, utilizzato solo se l'opzione nebbia lineare è attivata.
- Ambient light: l'impostazione predefinita è un grigio. Se vogliamo dare alla scena uno schema di illuminazione diverso, cambiare la luce ambientale è un buon inizio.
- Skybox Material : la trama del cielo della scena.
- Halo Strength: dimensione di tutti gli aloni luminosi in relazione alla loro portata.
- Flare Strength: Intensità di tutti i bagliori nella scena.
- Halo Texture: riferimento a una texture che apparirà come il bagliore di tutti gli aloni nelle luci.
- Spot Cookie: il riferimento a una Texture2D che apparirà come maschera dei cookie per tutti i faretti.



5.4.3 MOVIMIENTO

Unity include un controller in prima persona che semplifica la creazione di giochi con una visuale in prima persona. Accedendo a "Standard Assets > Character controller", occorre selezionare il "First person controller" e trascinarlo nella posizione in cui si desidera inserirlo nella scena.



Nelle versioni precedenti di Unity era possibile programmare il movimento solo con il seguente script:

```
var velocidad=8;
public var posAllida:GameObject;
function update () {
    var direccion=transform.TransformDirection( Vector3( 0, 0, 0.1 ));
    if( Input.GetKey("up")){
        //transform.Translate( 0 , 0 , 0.1 );
        rigidbody.AddForce( direccion*velocidad );
    }
    if( Input.GetKey("left"))
        transform.Rotate( 0 , -1 , 0 );
    if( Input.GetKey("down")){
        //transform.Translate( 0 , 0 , -0.1 );
        rigidbody.AddForce( direccion*velocidad*-1 );
    }
    if( Input.GetKey("right"))
        transform.Rotate( 0 , 1 , 0 );
}
```

È possibile vedere la telecamera che è stata aggiunta alla propria scena, dai quattro punti della telecamera è possibile aumentare o diminuire la messa a fuoco.

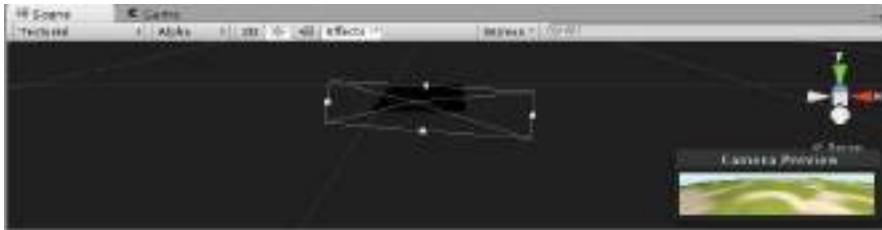


A.M.E.F.E.
Asociación Madrileña de
Educación y Formación Europea

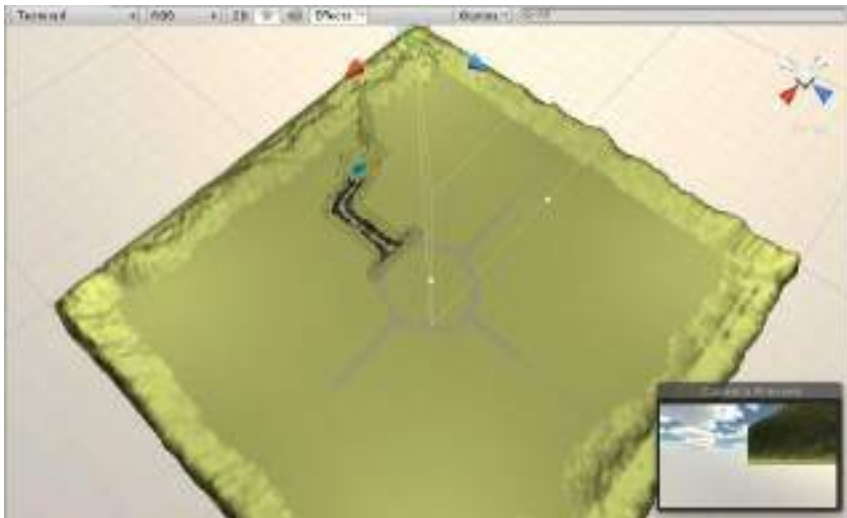
Taldea



Erasmus+
Enriching lives, opening minds.

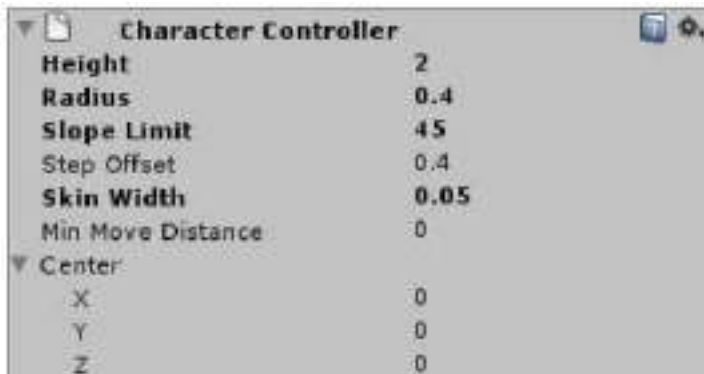


Una volta che la telecamera è pronta, il movimento può essere fatto attraverso diverse opzioni: Movimento: con frecce o tasti (W: avanti, A: sinistra, D: destra e S: indietro). La telecamera si controlla con il mouse.



Il salto con la barra spaziatrice.

Il "First Person Controller" può essere personalizzato come gli altri elementi nell'inspector view. Per fare ciò, viene selezionato nella hierarchy view e le sue caratteristiche appariranno nell'inspector.



Spiegazione dei comandi:

- Height: altezza del controller
- Radius: radio controller
- Slope limit: valore massimo della pendenza che può essere superato dal controller
- Step offset: per un essere umano di 2 metri questo valore è tra 0.1 e 0.4
- Skin width: consente di controllare i collisori del personaggio in modo da far sì che non sia bloccato
- Min move distance: il personaggio resta fermo se cerca di muoversi sotto il valore indicato. Può essere utilizzato per ridurre le vibrazioni. Nella maggior parte delle situazioni questo valore deve essere 0.
- Center: indica la posizione del controller e le sue coordinate x, y, z.

5.4.4 AUDIO NELLA SCENA

In Unity è possibile inserire suoni nel gioco. Ci sono due tipi differenti di audio: 3D o 2D.

L'audio 3D si sente in una posizione e si attenua man mano che ci si allontana da quella posizione. L'audio 2D si sente in modo uniforme in tutta la scena.

Per prima cosa bisogna creare una cartella chiamata "sound" in cui salvare un



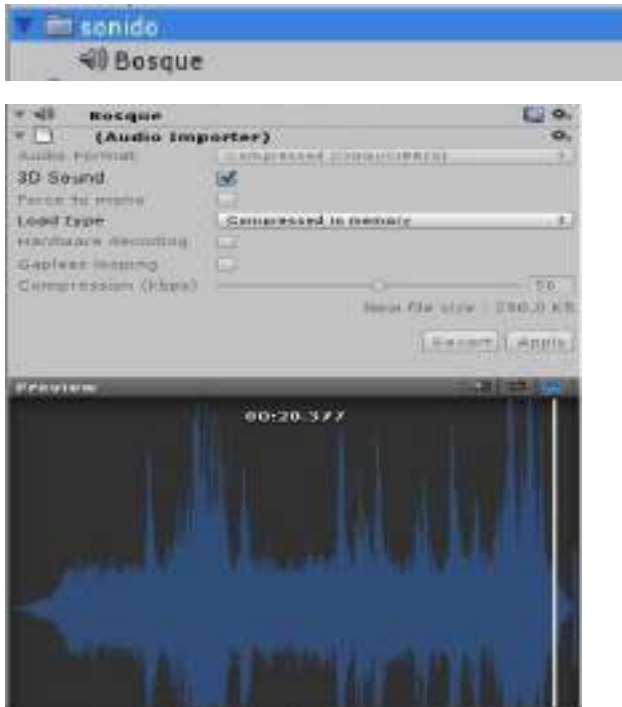
A.M.E.F.E.
asociación española de
educación y formación europea

Taldea

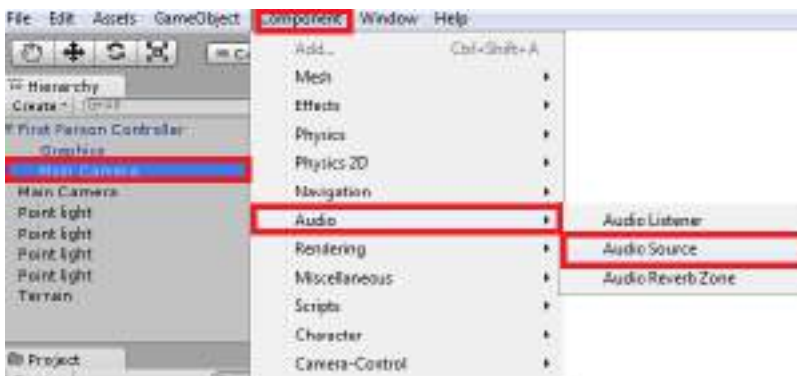


Erasmus+
Enriching lives, opening minds.

suono importato, ad esempio uno che rappresenta i suoni di una giungla o di una foresta.



Poi bisogna selezionare la voce "Main Camera" e cliccare su Component > Audio > Audio Source.



Il suono importato viene quindi trascinato nella casella "Clip audio" nell'inspector



view e l'audio viene configurato:

La casella di controllo "Play On Awake" è attivata in modo che il suono parta all'inizio del progetto.

La casella di controllo "Loop" è attivata in modo che la musica venga riprodotta in loop e ripetuta tutte le volte necessarie.

Sul grafico "Listener" nel componente, bisogna selezionare il punto verde nel punto più alto e trascinarlo al centro delle coordinate (0,0)



Una minima distanza di 0.1 può essere impostata inserendo il valore in "Min Distance".



5.4.5 ACCENDERE E SPEGNERE LE LUCI NELLA SCENA

In questa versione di Unity, è possibile accendere e spegnere la luce della scena facendo clic sul pulsante della luce, che si trova nelle opzioni di stage view, se invece si vuole spegnere la luce in uno specifico momento, si può scrivere un semplice script come mostrato di seguito:

```

var myLight : Light;
var myLight1 : Light;

-function Start(){
    myLight = Light.FindObjectOfType(typeof(Light));
}

-function Update () {
    if(Input.GetButtonDown("Fire1")){
        myLight.enabled = !myLight.enabled;
        myLight1.enabled = !myLight1.enabled;
    }
}

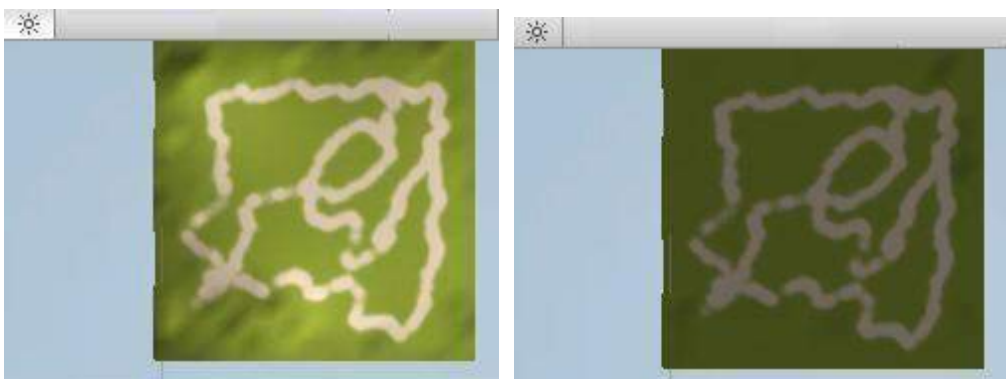
```

Spiegazione degli script:

- myLight = Light.FindObjectOfType(typeof(Light)); _ cerca nella scena l'oggetto di tipo "luce" (la luce nella scena).
- if(Input.GetButtonDown("Fire1")) quando si clicca il tasto sinistro del mouse.

- `myLight.enabled = !myLight.enabled;` `_` cambia lo stato della luce, se è accesa viene spenta e viceversa.
- `myLight1.enabled = !myLight.enabled` `_` questa riga è duplicata perchè nella nostra scena ci sono due luci.

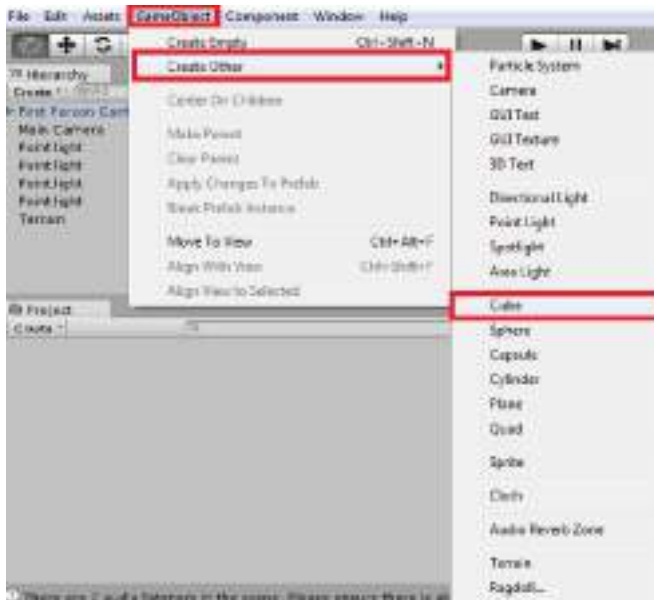
Una volta creato lo script, viene aggiunto all'oggetto. Nei parametri che appaiono nell'inspector view, trascina ciascuna delle 2 luci in modo da ottenere questo effetto:



5.4.6 CREAZIONE DI UN OGGETTO

In questa sezione ti spiegheremo come creare un oggetto, in questo caso un cubo.

Per creare oggetti predefiniti in Unity 3D, clicca su `GameObject > Create Other`. Una lista di oggetti verrà mostrata.





A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

Alcune componenti importanti dei GameObject sono:

- Camera: consente di inserire una telecamera di default nella scena, anche se durante la creazione di un progetto, Unity posiziona una telecamera per impostazione predefinita.
- Directional light: inserisce una luce direzionale uniforme che permette di vedere la scena.
- Cube: inserisce un cubo nella scena.
- Sphere: inserisce una sfera
- Capsule: inserisce una capsula
- Cylinder: inserisce un cilindro.
- Plane: inserisce un piano nella scena. Quando uno dei "GameObjects" è selezionato, appare nella scene view, nella game view e nella hierarchy view. Inoltre, nelle impostazioni è possibile vedere gli attributi dell'oggetto per la sua parametrizzazione.



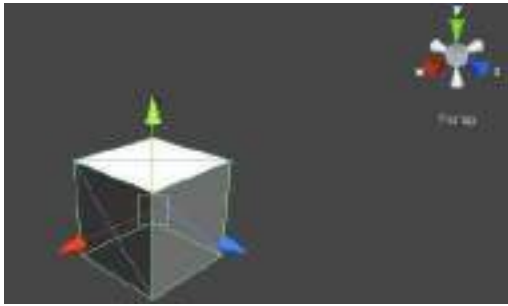
5.4.7 MANIPOLAZIONE DELL'OGGETTO

Quando viene creato un oggetto, in questo caso un cubo, è possibile modificare le dimensioni, la posizione e la forma per ottenere le caratteristiche desiderate.

La posizione del cubo può essere cambiata utilizzando il secondo pulsante di controllo:



Come possiamo vedere nell'immagine seguente, è possibile spostare il cubo sugli assi X (rosso), Y (verde) e Z (blu). Per spostare il cubo è sufficiente fare clic sull'asse in cui si desidera posizionarlo e trascinarlo nella nuova posizione.



Il terzo pulsante di controllo permette di ruotare il cubo.



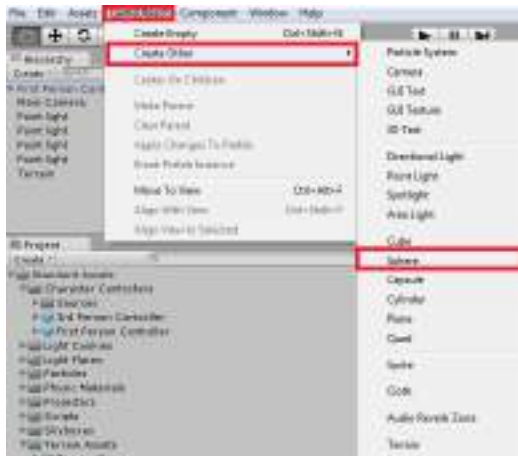
Confrontando l'immagine precedente, in questa vediamo come ruota il cubo.

I diversi cerchi che appaiono attorno al cubo quando viene selezionata l'opzione di rotazione consentono all'utente di scegliere l'asse su cui desidera eseguire questa operazione. Per ruotare un elemento basta cliccare su quello identificato dai cerchi colorati e muovere il mouse nella direzione selezionata.



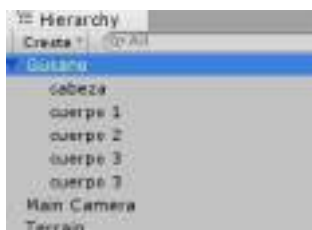
5.4.7 CREAZIONE DI UN BRUCO

Visto che sappiamo come creare un oggetto, creeremo un semplice personaggio, un bruco con delle sfere. La prima sfera viene creata in "GameObject > Create Other > Sphere":



Occorre posizionare la sfera sul terreno dove desideri. È possibile creare altre sfere identiche, occorre premere "CTRL + D", questa nuova sfera apparirà sopra quella vecchia e potrà essere spostata. Ripetere la stessa azione fino ad avere 4 sfere. Si può vedere che le sfere appaiono nella hierarchy view dove è possibile rinominare la prima sfera in "testa" e le altre in "corpo". Per rinominare un oggetto di gioco, occorre selezionarlo nella hierarchy view, fai clic con il tasto destro e "Rinomina". Modificare le sfere che rappresentano il corpo rendendole più piccole usando l'inspector per una maggiore precisione.

Inoltre, per ottimizzare l'organizzazione del nostro progetto, viene creato un oggetto vuoto in "GameObject > Crea empty", rinominato come bruco e la testa e il corpo vengono trascinati sul bruco.



A questo punto viene importata una texture che viene applicata al personaggio, che determina il seguente risultato:



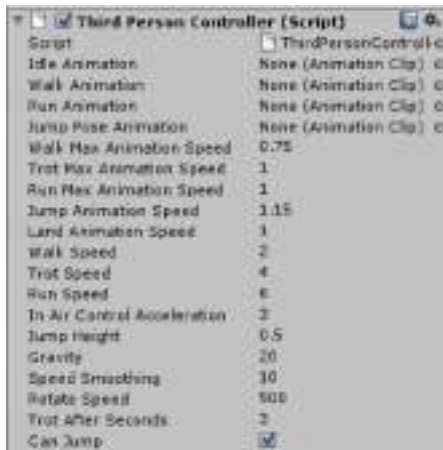
Infine, per fare in modo che tutti gli elementi che compongono il corpo del bruco seguano la testa, verrà utilizzato il componente "Smooth follow", che verrà spiegato nella prossima sezione.

5.4.8 MOBILITÀ DELL'OGGETTO

Selezionare la testa nella hierarchy view, andare su Component> Script> Third Person Controller. In questo modo è possibile dare mobilità alla testa e controllarla in terza persona. Se si vuole spostare l'intero bruco si deve applicare il "Third Person Controller" all'intero elemento.

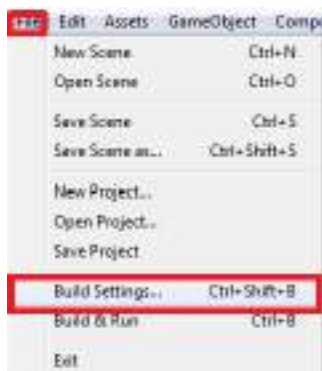


Nell'inspector view è possibile vedere come i nuovi componenti appaiono con tutti i parametri che possono essere configurati.



5.5 ESPORTARE IL PROGETTO

Per finalizzare il progetto possiamo esportarlo:



Bisogna selezionare la piattaforma e la struttura su cui si vuole esportare il progetto, si può esportare su piattaforme WINDOWS, MAC, LINUX, ANDROID, IOS con licenza gratuita, per esportare su console di gioco come XBOX, WII e PLAYSTATION è necessario pagare la licenza.



Infine, cliccando su Build and Run per esportare il progetto, occorre dargli un nome e salvarlo nella cartella che si preferisce, terminando con il file della piattaforma che si sceglie, tenendo conto delle caratteristiche del computer su cui il progetto viene importato.



Alla fine del progetto si può concludere che i motori grafici per i videogiochi sono strumenti molto utili che facilitano il lavoro dei creatori permettendo loro di concentrarsi sugli aspetti creativi e non tanto su quelli strutturali, al fine di soddisfare le richieste dei consumatori.

Unity 3D è uno dei motori più attuali per la progettazione di videogiochi, è molto completo e potente. Inoltre, ogni versione è stata adattata alle nuove tendenze come il design multiplatforma. Questa caratteristica conferisce a Unity 3D un vantaggio rispetto ad altri motori di videogiochi.

Per realizzare questo progetto abbiamo utilizzato la versione gratuita di Unity 3D, ma il fatto che sia gratuita non la rende meno potente. Unity mette a disposizione degli utenti un ambiente semplice che permette la creazione di un progetto in maniera molto veloce e, non essendo necessario avere conoscenze approfondite di programmazione, chiunque può utilizzarlo con poche nozioni di base.

Dobbiamo tenere presente che Unity non solo permette di creare videogiochi o tour virtuali, ma è anche uno strumento molto apprezzato da architetti e designer, poiché consente la creazione di ambienti 3D in modo semplice e veloce.



A.M.E.F.E.
asociación italiana de
educación y formación europea



 **Erasmus+**
Enriching lives, opening minds.

Unità 3 - Come creare codici utilizzando il linguaggio di programmazione C#

INTRODUZIONE

Nel 2002, Microsoft ha annunciato il lancio del linguaggio C#, simile a C++ e Java, ma migliorato (nella musica il simbolo # sta per "un semitono più alto"). Questo sviluppo è stato una parte importante dell'iniziativa "dot net" di Microsoft, che descriveremo in seguito. A causa della similitudine di C# con i suoi predecessori, apprendere questo linguaggio renderà più facile la comprensione di C, C++ e Java, nel caso si avesse necessità di utilizzarle.

- C# è un linguaggio object-oriented (orientato agli oggetti), derivato da C++ e Java.
- Il framework .NET di Microsoft è un prodotto importante, che include i linguaggi C#, C++ e Visual Basic.
- I programmi sono elenchi di istruzioni a cui i computer obbediscono automaticamente
- Al giorno d'oggi, la tendenza principale nella pratica della programmazione è la metodologia orientata agli oggetti (OOP) e C# la supporta pienamente.

1. CREARE UN PROGETTO CON C#

In questo manuale utilizzeremo l'acronimo "IDE" (Integrated Development Environment) per riferirci all'ambiente di sviluppo integrato di C#. Come i word processor, programmi che facilitano la creazione di documenti, l'IDE dispone di strumenti per creare (sviluppare) programmi. Tutti gli strumenti di cui abbiamo bisogno a tale scopo sono integrati, quindi possiamo svolgere tutto il nostro lavoro all'interno dell'IDE invece di dover utilizzare altri programmi. In altre parole, l'IDE ci fornisce un ambiente di programmazione completo.

1.2 INSTALLAZIONE E CONFIGURAZIONE

Per seguire gli esercizi è necessario scaricare Microsoft Visual C# 2008 Express Edition, seguire le istruzioni di installazione e registrare il prodotto se richiesto.



Spiegheremo ora come raggruppare i programmi creati con C# in una cartella specifica.

Quando crei un programma, C# genera diversi file e li inserisce in una nuova cartella. Il risultato dell'insieme dei file è chiamato "progetto". È consigliabile creare una cartella di primo livello per archiviare tutti i tuoi progetti. La cartella predefinita utilizzata da C# è denominata "Progetti" e si trova all'interno della cartella "Documenti" in Visual Studio 2008 nell'unità C. Quasi tutti gli utenti lo trovano appropriato, ma se per qualche motivo è necessario modificare la posizione predefinita, attenersi alla seguente procedura:

1. Fare clic sul menu Strumenti nella parte superiore dello schermo. Selezionare Opzioni... Assicurarsi che la casella Mostra tutte le impostazioni sia spuntata.
2. Fare clic sull'opzione Progetti e soluzioni e quindi sul pulsante ... a destra della sezione Ubicazione del progetto.
3. Si aprirà quindi la finestra Posizione progetto, in cui è possibile scegliere una cartella esistente o crearne una nuova. Fare clic sul pulsante OK una volta terminato il procedimento.

D'ora in poi C# salverà tutto il lavoro nella cartella del progetto di default o in quella che è stata appositamente selezionata. In ogni caso, la prima volta che un progetto viene salvato, comparirà una finestra di dialogo con il pulsante Sfoglia, nel caso in cui si avesse bisogno di utilizzare una cartella diversa da quella selezionata.

1.3 CREARE IL PRIMO PROGRAMMA

Il programma che creeremo successivamente mostrerà il messaggio Hello World sullo schermo del tuo computer. Indipendentemente da quanto sopra, bisognerà seguire questi passaggi generali per creare qualsiasi programma.



Figura 2.1 Home page



Figura 2.2 Finestra nuovo progetto

-
- Aprire l'IDE. Apparirà la Home Page, come mostrato nella Figura 2.1.
- **Fare clic sul collegamento Crea: Progetto...** La Figura 2.2 mostra la finestra **Nuovo progetto** che apparirà in risposta.
- Assicurarsi che il modello **Applicazione Windows Form** sia selezionato. Scegliere un nome per il progetto, che diventerà anche un ID cartella. Si consiglia di utilizzare solo lettere, cifre e spazi. Nel nostro caso abbiamo usato il nome **First Hello**. Fare clic su **OK**. Apparirà un'area di layout simile (ma non necessariamente identica) a quella mostrata nella Figura 2.4.
- Per facilitare il tuo utilizzo nel linguaggio C#, la casella degli strumenti dovrebbe essere sempre visibile. Fare clic sul menu **Visualizza** e selezionare **Toolbox (Pannello degli strumenti)**. Ora fare clic sul simbolo "puntina da disegno" nel toolbox, come mostrato in Figura 2.3; il pannello sarà permanentemente fissato e aperto. Fare clic su **Tutti i Windows Form** per visualizzare l'elenco degli strumenti disponibili. Lo schermo apparirà ora quasi identico a quello della Figura 2.4.

Come accennato in precedenza, i passaggi elencati sono comuni nella realizzazione di qualsiasi progetto. Eseguiamo ora un compito specifico per questo particolare progetto.

Notare il form e il toolbox mostrati nella Figura 2.4. Ora selezioneremo un controllo dalla casella degli strumenti e lo posizioneremo nel modulo. Ecco i passi da seguire:

- Individuare la casella degli strumenti e fare clic sul controllo **Etichetta**.
- Sposta il puntatore del mouse sul modulo. Fare clic e, senza rilasciare il pulsante, trascinare per creare un'etichetta come nella Figura 2.5.
- Ora imposteremo alcune proprietà dell'etichetta: fare clic con il tasto destro sull'etichetta e selezionare **Proprietà**. Spostare il mouse sull'elenco Proprietà in basso a destra dello schermo, come mostrato nella Figura 2.6

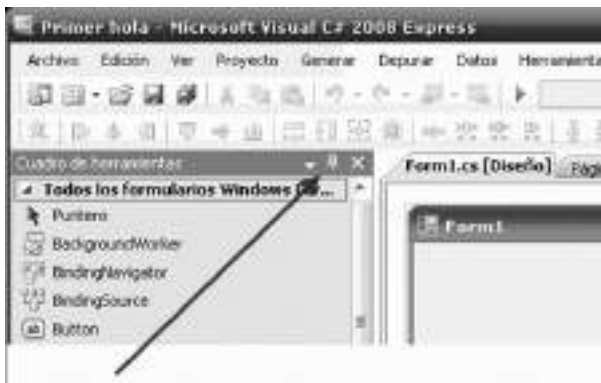


Figura 2.3 Il toolbox è fissato e aperto.

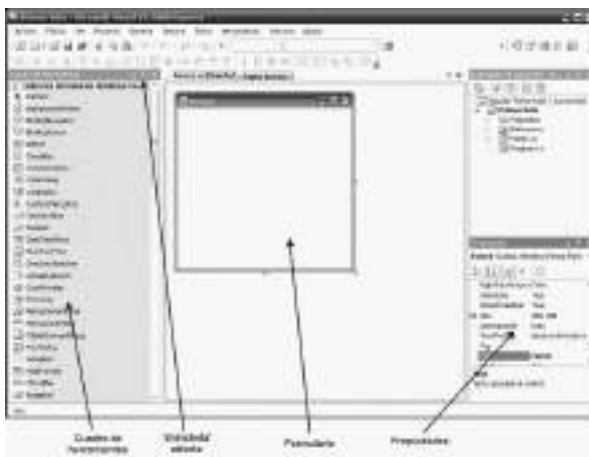


Figura 2.4 L'IDE in fase di progettazione.



Figura 2.5 Un'etichetta è stata aggiunta al modulo.

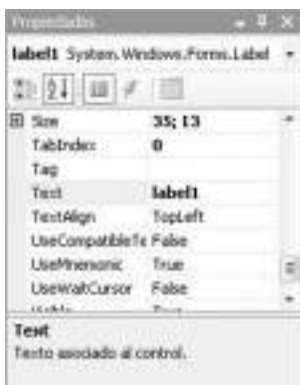


Figura 2.6 Proprietà dell'etichetta.



Figura 2.7 Fare clic sulla freccia per eseguire il programma.

- Scorrere verso il basso fino alla proprietà **Text** e sostituire il testo **label1** con **Hello World**.
- Ora eseguiamo il programma cliccando sulla freccia in alto nell'IDE (Figura 2.7); questo è il pulsante Avvia debug.

Apparirà una nuova finestra, come mostrato nella Figura 2.8. Questo è il programma creato. Mostra solo testo, ma è una vera finestra in quanto puoi spostarla, ridimensionarla e chiuderla facendo clic sull'icona **X** nell'angolo in alto a destra. Sperimenta con questa finestra e poi chiudila.

Per salvare il programma per un uso successivo:

- Andare nel menu **File** e selezionare l'opzione **Salva tutto** (di seguito, per indicare questo tipo di azioni si utilizzerà una forma abbreviata come questa: **File | Salva tutto**).



- Apparirà la finestra **Save Project (salva progetto)**, come mostrato nella Figura 2.9. Assicurati che l'opzione **Create directory for solution** sia deselezionata. Lasciare invariate le altre opzioni e fare clic su Salva. Quando si salva nuovamente il progetto, verranno utilizzate automaticamente le stesse opzioni e la finestra Salva progetto non verrà più visualizzata.

È ora possibile utilizzare il **Comando File | Exit** per chiudere l'IDE.

La prossima volta che utilizzerai C#, il nome del tuo progetto apparirà nella pagina iniziale, così potrai aprirlo con un solo clic, senza dover ripetere il lavoro che abbiamo fatto per impostare il progetto.



Figura 2.8 Il **First Hello** programma in esecuzione.

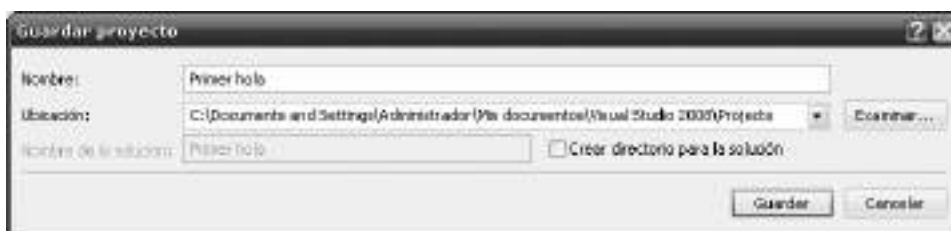


Figura 2.9 La finestra **Salva progetto**, con l'opzione **Create directory** disattivata.

1.4 CONTROLLI DESIGN-TIME

Nel nostro programma First Hello abbiamo inserito un'etichetta su un modulo e modificato il testo da visualizzare. Lo scopo principale dell'esercizio era quello di rivedere le fasi coinvolte nella creazione di un progetto. Successivamente, impareremo alcune delle basi dei controlli e delle proprietà.

Cos'è un controllo? È un "dispositivo" che appare sullo schermo, sia per visualizzare informazioni, sia per consentire all'utente di interagire con



l'applicazione, o entrambi. Lo stesso IDE C# usa molti controlli. Ad esempio, nell'esercizio precedente abbiamo utilizzato i menu a discesa per salvare i progetti e il pulsante OK per confermare le azioni. Per le applicazioni Windows, il pannello degli strumenti contiene circa 70 controlli; parte dell'attività di programmazione comporta la selezione dei controlli appropriati, il loro posizionamento su un modulo e l'impostazione delle relative proprietà. Contrariamente all' "esecuzione", questa fase è legata all'utilizzo dei controlli ed è chiamata "tempo di progettazione". Quando il programma viene eseguito, l'utente interagisce con i controlli. Il compito del programmatore è creare un'interfaccia utente grafica (GUI) per facilitare questa interazione. Vediamo ora come i controlli possono essere manipolati in fase di progettazione.

- È possibile selezionare un controllo dalla casella degli strumenti e posizionarlo su un modulo. La posizione iniziale assegnatagli non è importante, in quanto possiamo cambiarla facilmente.
- È anche possibile spostare il controllo. Se si posiziona il puntatore del mouse su un controllo, accanto ad esso apparirà una freccia a quattro punte, come mostrato nella Figura 2.10. È quindi possibile trascinare il controllo con il mouse.
- È possibile modificare le dimensioni del controllo. Quando si fa clic su un controllo, ai suoi bordi vengono visualizzate diverse piccole caselle (punti di manipolazione). Posizionare il mouse su una di queste piccole caselle per far apparire una freccia a due punte, come mostrato nella Figura 2.11. Trascinare il bordo o l'angolo per ridimensionare il rettangolo.

Infatti, il metodo di ridimensionamento dipende dal controllo stesso. Ad esempio, il controllo label ha una proprietà chiamata AutoSize (ridimensionamento automatico, in base alle dimensioni del testo inserito) che impostiamo a

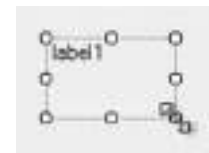
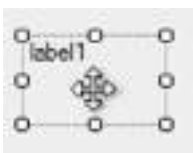


Figura 2.10 I controlli possono essere mossi... **Figura 2.11** ... e ridimensionati secondo necessità



A.M.E.F.E
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

Se lasciamo la proprietà `AutoSize` impostata su `True`, la larghezza dell'etichetta sarà determinata in base alla larghezza del testo da visualizzare e l'altezza sarà determinata in base alla dimensione del carattere del testo. Altri controlli permettono di variare la larghezza trascinando i punti con il mouse, ma non l'altezza (che viene impostata in base ai font). Alcuni controlli, come i pulsanti, consentono di modificarne le dimensioni in qualsiasi direzione. In generale, la dimensione dei controlli dipende dalla loro destinazione d'uso, quindi consigliamo di sperimentarli. Tuttavia, poiché le etichette sono così comuni, non dobbiamo dimenticare la loro proprietà `AutoSize`.

Successivamente, esamineremo le proprietà. Ecco un'analogia: i televisori hanno proprietà come il colore del case, la dimensione dello schermo, il canale che sta trasmettendo in un dato momento, il suo prezzo e la sua marca.

Allo stesso modo, ogni controllo ha un insieme di proprietà che possono essere regolate in fase di progettazione per soddisfare le nostre esigenze. Vedremo in seguito come modificare una proprietà in fase di esecuzione (runtime).

Dopo aver posizionato un controllo sulla scheda, per visualizzarne le proprietà, fare clic con il tasto destro su di esso e selezionare `Proprietà`; questo mostrerà la finestra delle proprietà del controllo scelto. La colonna di sinistra contiene i nomi delle proprietà e la colonna di destra contiene il loro valore corrente. Per cambiare una proprietà dobbiamo modificare il valore nella colonna di destra. Per alcune proprietà potrebbe essere necessario selezionare diverse opzioni aggiuntive, ad esempio quando si modificano le impostazioni del colore e i caratteri del testo. A volte, quando il range di valori tra cui scegliere è molto ampio, è necessario aprire una finestra aggiuntiva.

Un altro aspetto fondamentale di un controllo è il suo nome. Questa non è di per sé una proprietà, ma per comodità viene mostrata nell'elenco delle proprietà come `(Nome)`. Le parentesi indicano che in realtà non è una proprietà.

Quando posizioniamo più etichette su un modulo, l'IDE seleziona i loro nomi come segue:

`label1label2label3 ...`



A.M.E.F.E.
Asociación Madrileña de
Educación y Formación Europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

Questi nomi sono accettabili per ora, ma nei capitoli seguenti vedremo che è meglio cambiare i nomi predefiniti di alcuni controlli con nomi più rappresentativi. Per modificare il nome di un controllo è necessario modificare il testo a destra di (Nome) nell'elenco delle proprietà.

1.5 **EVENTI E PULSANTE CONTROLLO**

Il programma che abbiamo creato nell'esercizio precedente non è molto rappresentativo, poiché mostra sempre le stesse parole e l'utente non può interagire con esso. Successivamente arricchiremo questo programma in modo che appaia un messaggio di testo quando l'utente fa clic su un pulsante. Questo è un esempio di come utilizzare un evento.

Quasi tutti gli eventi vengono attivati dall'utente e vengono generati quando l'utente manipola un controllo in qualche modo in fase di esecuzione. Ogni controllo ha diversi eventi a cui può rispondere, come il clic di un pulsante del mouse, un doppio clic o il posizionamento del puntatore del mouse sul controllo. Altri tipi di eventi non provengono dall'utente; ad esempio, la notifica che una pagina Web ha terminato il download.

Nel programma che creeremo successivamente, rileveremo un evento (il clic di un pulsante); quindi faremo visualizzare un messaggio di testo su un'etichetta. Ecco come creeremo l'interfaccia utente:

- Creare un nuovo progetto chiamato Hello button.
- Posizionare un'etichetta e un pulsante sul modulo. La posizione non è importante.
- Digitare Fare clic qui nella proprietà Text del pulsante.
- Modificare la proprietà Text dell'etichetta, in modo che appaia senza contenuto.

Eeguire il programma, anche se non è ancora completo. Si noti che è possibile fare clic sul pulsante e che cambia leggermente il suo aspetto per confermare che viene premuto; non succede nient'altro. Chiudere il modulo.

Ora vediamo come rilevare l'evento clic. Fare doppio clic sul pulsante all'interno del modulo di layout. Si aprirà un nuovo pannello informativo, come mostrato nella Figura 2.12. Notare le schede in alto:



Form1. csHome page

Form1.cs [Design] [Design]

È possibile fare clic su queste schede per cambiare pannello. Il riquadro Form1.cs visualizza un programma C#. Questo è noto come "testo del programma" o "codice" C#. Modifichiamo questo codice usando l'editor IDE.

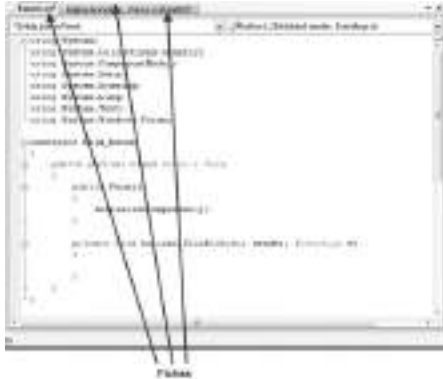


Figura 2.12 Codice C# nel riquadro di modifica.

Sfoggia il codice fin quando trovi la prossima sezione del codice:

private void button1_Click(object sender, EventArgs e)

```
{  
  
}
```

Questa sezione del codice è conosciuta come metodo (*method*). Il nome della sezione metodo è **button1_Click**. Quando l'utente clicca sul pulsante **button1** in fase di esecuzione, effettuerà o "eseguire" o qualunque istruzione abbiamo posto tra le due righe sopra.

In questo specifico programma useremo un'istruzione per inserire il messaggio **Hello World** nel testo dell'etichetta **label1**. L'istruzione per eseguire questa azione è:

label1.Text = "Hello world";

Scrivi l'istruzione esattamente come mostrato qui, tra le righe { and }. Nei prossimi capitoli spiegheremo l'esatto significato delle righe come quella sopra (che implicano un'istruzione attribuzione)

Il passo successivo è quello di eseguire il programma. Ci sono due possibilità per questo:

- Se il programma viene eseguito correttamente, quando si clicca sul pulsante apparirà il messaggio **Hello World** sull'etichetta.



Figure 2.13 Azione sbagliata. Selezionare la casella di controllo e fare clic su **No**.

- L'altra possibilità è che il programma non venga eseguito a causa di un errore. In questo caso C# mostrerà un messaggio come quello nella Figura 2.13. Selezionare la casella di controllo per l'opzione **Do not show this dialog box again**, e cliccare **No** per continuare. Il messaggio non apparirà più; d'ora in poi l'unica prova dell'errore sarà una sottolineatura nel testo del codice.

In ogni caso, l'errore deve essere corretto: rivedere il codice, correggere eventuali errori di battitura ed eseguire nuovamente il programma. Discuteremo gli errori più in dettaglio in seguito.

Queste sono le nuove caratteristiche del programma:

- Può rispondere al clic di un pulsante. Il processo di immissione del codice in modo che l'azione corrispondente venga eseguita quando si verifica un evento è noto come "gestione" dell'evento.
- Modifica del valore della proprietà di un controllo in fase di esecuzione. Precedentemente lo abbiamo fatto solo in fase di progettazione. Questa è una parte molto importante della programmazione, poiché spesso dobbiamo mostrare i risultati ponendoli in una certa proprietà di controllo.

1.6 APRIRE UN PROGETTO ESISTENTE

Per riaprire un progetto creato in precedenza, salvare e chiudere il progetto in corso e andare alla pagina iniziale. Questo mostra i progetti su cui hai lavorato più di recente; per aprire un progetto, fai semplicemente clic sul suo nome. Se il progetto che stai cercando non è nell'elenco, fai clic sul link Open: Project...link e naviga nella cartella appropriata. All'interno della cartella c'è un file di tipo .sln



(solution). Seleziona questo file e apri il progetto. Per visualizzare il modulo vai al Solution Explorer, situato in alto a destra della finestra, e fai doppio clic sul nome del modulo.

1.7 DOCUMENTAZIONE DEI VALORI DELLA PROPRIETÀ

In questo toolkit dobbiamo fornire i valori delle proprietà che useremo nei controlli. Quando il numero di proprietà è piccolo possiamo specificarlo in poche istruzioni, ma se il numero è più grande useremo una tabella. In generale abbiamo diversi controlli; ogni controllo ha diverse proprietà e ogni proprietà ha un valore. Si noti che ci sono un gran numero di proprietà per ogni controllo, ma elencheremo solo quelle che devono essere modificate. Gli altri manterranno il loro valore predefinito. Ecco un esempio:

Control	Property	Value
label1	Text	Hello world
label1	BackColor	Web
button1	Text	Click here

I valori elencati corrispondono ai seguenti controlli:

- label1 il cui testo è Hello world, e il cui colore di sfondo è rosso;
- button1 il cui testo è Clic here.

Fai attenzione quando scrivi i nomi delle proprietà. Ne parleremo più dettagliatamente nei capitoli successivi; per ora, basti dire che dovresti farlo con una lettera maiuscola iniziale e senza spazi.

1.8 ERRORI NEI PROGRAMMI.

È comune che si verifichino bug nei programmi; fortunatamente, l'IDE può rilevarne alcuni per noi. Ecco un esempio di errore:

label1.Text = "Hello world".

Se digiti questa riga ed esegui il programma, l'IDE la sottolineerà. Posizionando il puntatore del mouse sulla parte sottolineata verrà visualizzata una spiegazione dell'errore. Si noti che la spiegazione può essere difficile da capire (anche per un



esperto), o che l'errore può trovarsi altrove nel programma. Tuttavia, il primo passo dovrebbe sempre essere quello di ispezionare l'area sottolineata per errori di battitura. Man mano che impari di più sul linguaggio C# migliorerai la tua capacità di rilevare gli errori.

Una volta che gli errori evidenziati sono stati corretti, il programma deve essere eseguito. Infatti, prima che questo accada, viene attivato un altro programma chiamato compilatore, che controlla il codice per più errori. Solo fino a quando tutti gli errori di compilazione sono stati corretti il programma può essere eseguito.

Gli errori di compilazione vengono visualizzati nella finestra Error list, che si apre sotto il codice. Facendo doppio clic sull'errore nella finestra dei risultati, l'IDE ti porterà al punto del codice in cui l'errore deve essere corretto.

Chiunque inizi a scrivere programmi dovrà affrontare molti errori di compilazione. Non scoraggiarti; fa parte del processo di apprendimento.

Dopo aver corretto gli errori di compilazione, il programma sarà in grado di eseguire. A questo punto si può notare che lo fa in modo errato, il che significherebbe che c'è un "errore di esecuzione", o bug. Tali errori sono più difficili da correggere, e il debug è quindi necessario.

1.9 **LE FUNZIONI DELL'EDITOR**

L'editor non si limita a mostrare ciò che il programmatore scrive; di seguito alcuni dei suoi vantaggi aggiuntivi:

- Offre operazioni standard di taglio, copia e incolla negli appunti, consentendo di copiare il codice da altri programmi di Windows.
- Per scrivere programmi più grandi, sono necessari controlli ed eventi aggiuntivi. Gli elenchi a discesa nella parte superiore dell'editor consentono di selezionare i controlli, oltre a un evento specifico per tale controllo. L'editor viene automaticamente inserito nel metodo appropriato.
- L'editor visualizza il codice C# mentre viene digitato in base a determinate regole. Per esempio, gli spazi verranno inseriti automaticamente attorno al simbolo =.
- Alcune linee avranno bisogno di un rientro - spostando il testo a destra - per le quali è necessario inserire spazi bianchi aggiuntivi. Ciò è di solito fatto in intervalli di quattro spazi. Come vedremo nei capitoli successivi, il codice



A.M.E.F.E.
asociación italiana de
educación y formación europea



Erasmus+
Enriching lives, opening minds.

C# è composto da sezioni all'interno di sezioni, e i rientri aiutano a indicare dove ogni sezione inizia e finisce.

Il processo di rientro di un programma è noto anche come formattazione. L'IDE può farlo automaticamente tramite il menu:

Editing | Advanced | Formatting the document It is a good idea to format the code frequently.

- Ogni tipo di controllo ha un particolare insieme di proprietà. Se si digita il nome di un controllo seguito da un punto, come mostrato di seguito: ***label1.***

e si aspetta un po', l'editor fornirà una lista di proprietà dell'etichetta label. Se questo strumento non è configurato sul sistema, è possibile attivarlo selezionando l'opzione:

Tools | Options | Text Editor | C#

e poi spuntando la casella di controllo Automatic membership list.

- Sezioni di codice su cui non stiamo lavorando possono essere raggruppate in una singola riga di codice; per fare ciò fare clic sul piccolo simbolo "-" a sinistra dell'editor. Per espandere una sezione, fare clic sul simbolo "+". Troverete questi simboli accanto alle linee come descritto di seguito:

private void button1_Click (object sender, EventArgs e)

Non è necessario ricordare tutti questi strumenti, in quanto sono sempre lì a portata di mano per aiutarvi.

Uno dei compiti relativi al layout del codice che non viene eseguito automaticamente è la divisione di lunghe righe . Per garantire che un'intera riga sia visibile sullo schermo, possiamo scegliere un posto adatto per dividerla e premere il tasto Invio. L'IDE rientrerà nella seconda parte della riga di quattro spazi. Ecco un esempio:

private void button1_Click (object sender, EventArgs e)

Se si preme il tasto Invio dopo la virgola, il codice apparirà in questo modo:

private void button1_Click (object sender, EventArgs e)

Spesso l'IDE crea righe molto lunghe che, naturalmente, non dovrebbero contenere errori, quindi è meglio non dividerle. Tuttavia, man mano che diventi più

esperto con C#, alcune di queste lunghe righe saranno di tua creazione, e sarà molto conveniente dividerle in modo da poter vedere tutto il testo contemporaneamente. Questo migliora anche la leggibilità del codice stampato (noto come elenco).

1.10 LA CASELLA DEL MESSAGGIO

In precedenza abbiamo utilizzato il controllo dell'etichetta per visualizzare il testo sullo schermo, ma possiamo anche utilizzare una finestra di messaggio per farlo. Questo controllo non appare nella casella degli strumenti, perché non occupa spazio permanente su un modulo; appare solo quando è richiesto. Di seguito è riportato un frammento di codice che visualizza il messaggio Hello World all'interno di una finestra di messaggio quando si fa clic su un pulsante:

```
private void button1_Click(object sender, EventArgs e)  
{  
    MessageBox.Show("Hello world");  
}
```

Figura 2.14 mostra cosa succede quando si esegue il programma e si fa clic sul pulsante: viene visualizzata la finestra di messaggio, e si deve fare clic sul pulsante OK per farlo sparire. Questa funzione implica che le caselle di messaggio vengano utilizzate per visualizzare messaggi importanti che l'utente non può ignorare.

Per utilizzare una finestra di messaggio, digitare una riga come quella sopra, ma utilizzare il proprio messaggio all'interno delle virgolette doppie. A questo punto non spiegheremo lo scopo di Show statement o perché sono richieste le parentesi. Parleremo di tutto questo quando studieremo i metodi in modo più dettagliato



Figure 2.14 Una casella di messaggio

1.11 AIUTO



Il sistema di aiuto funziona sulla base di due possibili fonti: una locale, situata sul proprio computer, e l'altra che utilizza Internet. La sorgente Internet viene aggiornata frequentemente; tuttavia, la versione locale è sufficiente quando si inizia a lavorare con C#. Nel caso in cui non si dispone di una connessione Internet, il programma utilizzerà automaticamente la guida locale.

Se si desidera specificare la provenienza dell'aiuto, è possibile riconfigurare il sistema come segue:

1. Nella finestra principale di C#, scegliere **Help | Search....**
2. Nella nuova finestra di aiuto, selezionare **Help | Help on Help**. Ti verranno presentate diverse opzioni su quale fonte di aiuto avrà la massima priorità.

Il sistema di aiuto di C# è vasto, ma se siete nuovi alla programmazione potete trovare le informazioni che contiene complicate. È più utile quando si ha una certa esperienza con C# e si richiedono dettagli tecnici più precisi.

Le opzioni più utili nel menu Help (aiuto) sono Index e Search, in quanto consentono di digitare del testo per il sistema per individuare le pagine che potrebbero essere utili. La differenza è che Index guarda solo i titoli delle pagine, mentre Search guarda anche il contenuto della pagina.

1.11.1 **Basi della programmazione**

- I controlli possono essere posizionati su un modulo al momento della progettazione.
- È possibile impostare le proprietà dei controlli al momento della progettazione.
- I programmi sono in grado di modificare le proprietà in fase di esecuzione.
- Quando si verifica un evento (ad esempio facendo clic su un pulsante), il sistema C# utilizza il metodo appropriato, ma spetta al programmatore inserire il codice all'interno del metodo per gestire l'evento.

1.11.2 **Errori di programmazione comuni**

- Dimenticare di terminare l'esecuzione del programma prima di provare a modificare il modulo o il codice.



A.M.E.F.E.
asociación italiana de
educación y formación europea



 **Erasmus+**
Enriching lives, opening minds.

- Confondere il modulo di progettazione con il modulo di esecuzione.

1.11.3 Segreti del coding

- Nel codice C#, per fare riferimento alla proprietà di un controllo usiamo il suo nome seguito da un punto e il nome della proprietà, come in:

label1.Text

- In una sezione di codice tra:

```
private void botonNumero_Click(Object sender, EventArgs e)
```

```
{  
}
```

è conosciuto come metodo (method).

- Le caselle dei messaggi non sono poste direttamente sui moduli. Per farli apparire sullo schermo dobbiamo usare le seguenti istruzioni:

```
MessageBox.Show ("Here is the text you want");
```

1.11.4 Nuovi elementi linguistici

Introduzione a proprietà, metodi ed eventi.

1.11.5 Nuove caratteristiche IDE.

- I programmi sono contenuti nei progetti.
- L'IDE crea una cartella contenente i file necessari per un progetto.
- È possibile posizionare e spostare i controlli sui moduli.
- La casella degli strumenti contiene una vasta gamma di controlli.
- Facendo clic destro su un controllo ci permette di selezionare le sue proprietà.
- Fare doppio clic su un pulsante al momento della progettazione crea metodi per gestire gli eventi.

1.11.6 Sommario

Parte dei compiti della programmazione consiste nel posizionare i controlli sui moduli e impostarne le proprietà iniziali. L'IDE C# esegue



A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

questo compito direttamente, ma è necessario fare pratica e leggere l'IDE.



Unità 4 - Input, fisica , interfaccia utente, scene e rendering

Interazione con l'utente

Input

Per permettere all'utente di interagire con le nostre applicazioni, Unity mette automaticamente a disposizione la classe "Input". Per approfondimenti vedere:

<https://docs.unity3d.com/Manual/class-InputManager.html>

<https://docs.unity3d.com/ScriptReference/Input.html>

Input fornisce la possibilità di verificare se e quando un tasto fisico (joypad, mouse tastiera) viene premuto, di ottenere la posizione di una o più assi fisiche (es. analog joystick, spostamento del mouse) o di creare assi virtuali partendo da due pulsanti digitali (es. tasto accelera-tasto rallenta).

Unity è in grado di ricevere input dagli schermi touch e da diversi sensori (es. giroscopio, camera, accelerometro...) presenti sugli smartphone.

```
void Update(){
    if(Input.GetKeyUp(KeyCode.Space)){ //Input from keyboard or joypad
        Debug.Log("Space has been released");
    }
}
```

```
void Update(){
    float myVerticalValue = Input.GetAxis("Vertical"); //input from analog
    joystick or virtual axes
    if(myVerticalValue != 0)
    {
        //eg move the player
    }
}
```

```
void Update(){
    if(Input.GetMouseButton(0)){ //for mouse and touchscreen input
        Debug.Log("Mouse 0 button has been clicked");
        Ray myRay =
        Camera.main.ScreenPointToRay(Input.mousePosition);
    }
}
```



```
if(Physics.Raycast(myRay)){
    //I clicked on something because myRay hit something
}
}
}

void Update(){
    Vector3 myDeviceAcceleration = Input.acceleration; //Input from
accelerometer
    if(myDeviceAcceleration.magnitude > 10)
    {
        Debug.Log("Il device è stato scosso forte");
    }
}
```

È bene controllare e gestire l'input nella funzione Update (perchè la funzione è richiamata ogni volta che viene creato un frame) in modo da avere una rapidità di reazione proporzionale al Framerate dell'applicazione.

Interazione con l'utente

Output

In caso di output testuale su console è possibile utilizzare la funzione:

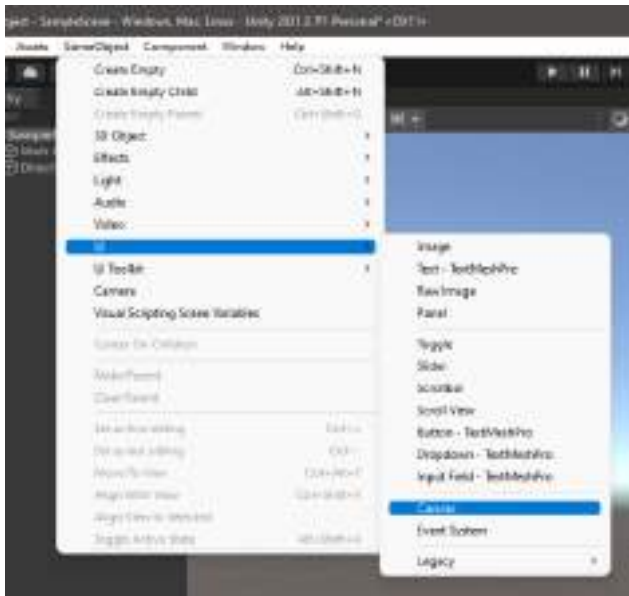
```
Debug.Log("customText");
```

La maggior parte dell'interazione tra utente e applicazione avviene attraverso l'interfaccia utente o UI.

Unity permette facilmente di creare un interfaccia utente utilizzando il GameObject Canvas. Per approfondimenti vedere:

<https://docs.unity3d.com/2021.3/Documentation/Manual/UIColorCanvas.html>

<https://docs.unity3d.com/ScriptReference/Canvas.html>



Canvas permette di creare un'interfaccia utente programmabile e personalizzabile utilizzando diversi elementi.



È possibile aggiungere testi e pulsanti.

Per creare un codice che controlli il testo utilizzando il pulsante, aggiungere una componente script a un Empty GameObject.

```
using UnityEngine.UI;
public class HealthDisplay : MonoBehaviour {
    public Text myHealthText;
    int myHealth = 10;
    public void myButtonFunction() { //function to connect
to the button
        myHealth--; //lower the heat
    }
}
```

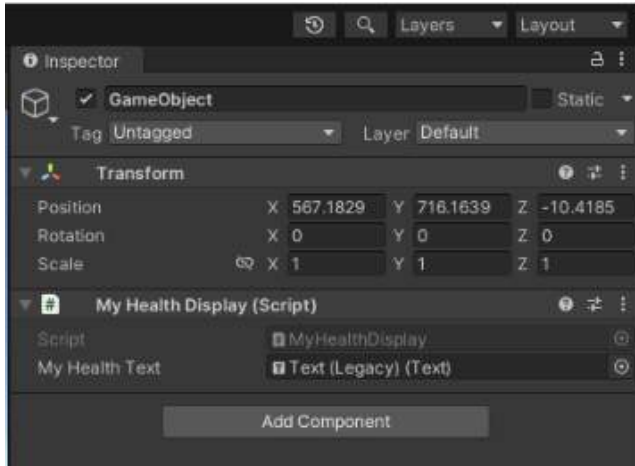



```

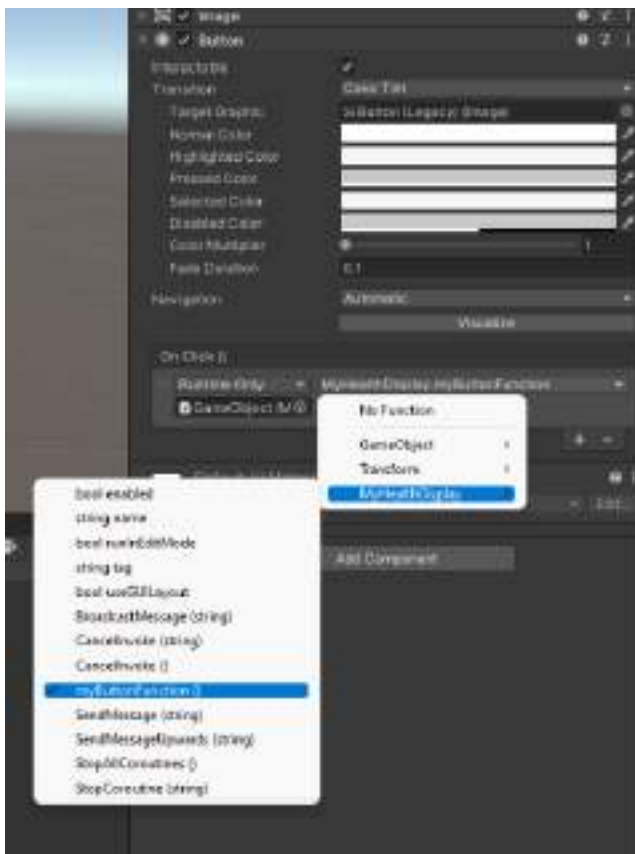
myHealthText.text = "HEALT : " + myHealth;           //updates the
label
}
}

```

Collegare il testo allo script utilizzando l'attributo Public Text (My Health Text).



Collegare il pulsante allo script dell'Empty GameObject attraverso la funzionalità `OnClick()` del pulsante utilizzando la funzione `public (void myButtonFunction())`.





Fisica

Collisioni

Le collisioni in Unity vengono elaborate grazie alle componenti Collider (BoxCollider, SphereCollider, MeshCollider) e gestite ad eventi (OnCollisionEnter, OnCollisionExit, OnCollisionStay).

<https://docs.unity3d.com/Manual/collision-section.html>

<https://docs.unity3d.com/ScriptReference/Collision.html>

```
void OnCollisionEnter(Collision acollision){
    if(acollision.relativeVelocity.magnitude >= 10){ //if a ball collides with
gameObject
        Destroy(gameObject); //delete gameObject
        Destroy(acollision.gameObject); //Destroy the ball
    }
}
```

In Edit > Project Settings > Tags and Layers è possibile aggiungere nuovi Layers

In Edit > Project Settings > Physics > Collision Matrix è possibile definire tra quali layers devono verificarsi le collisioni.

La modifica della componente transform oppure l'utilizzo delle funzionalità transform.Translate o Transform.Rotate non generano collisioni.

Per evitare computazioni inutili inserire la componente collision solo quando necessario.

RigidBody

Gli aspetti principali della dinamica vengono simulati attraverso la componente Rigidbody.

<https://docs.unity3d.com/Manual/rigidbody-physics-section.html>

<https://docs.unity3d.com/ScriptReference/Rigidbody.html>



Attraverso l'utilizzo della componente rigidbody è possibile applicare forze e momenti, impostare velocità, impostare forze di attrito e gravità e creare movimenti che generano collisioni.

```
private void Update(){
    int jumpForce = 100;
    int speed = 10;
    if(Input.GetKeyDown(KeyCode.Space))
        gameObject.rigidbody.AddForce(Vector3.up * jumpForce,
ForceMode.Impulse);
    if(Input.GetKeyDown(KeyCode.R))
        gameObject.rigidbody.AddTorque(transform.right * speed,
ForceMode.Force);
}
```

È preferibile slegare l'utilizzo delle simulazioni fisiche dal frame rate utilizzando la funzione

FixedUpdate piuttosto che Update.

```
private bool goingForward;
private void Update(){
    goingForward = Input.GetKey(KeyCode.UpArrow);
}
private void FixedUpdate(){
    if(goingForward){
        gameObject.rigidbody.AddForce(Vector3.Forward * speed,
ForceMode.Force);
    }
}
```

Velocità di movimento molto veloci potrebbero non venir gestite in tempo.

Physic Material

I Physic Material, se utilizzati su un GameObject con una collider, permettono di definirne alcune proprietà fisiche come la creazione di attrito statico, dinamico e l'elasticità dell'oggetto.



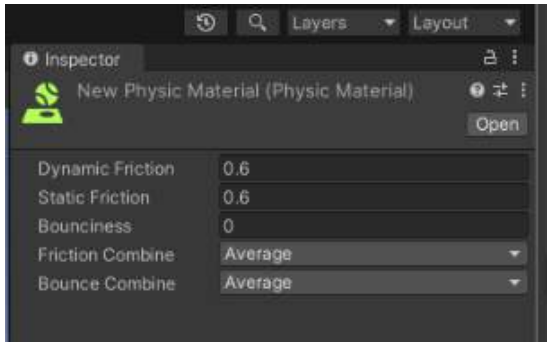
A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

<https://docs.unity3d.com/Manual/class-PhysicMaterial.html>



Joints

È possibile collegare due GameObject (con rigidbody) insieme attraverso la componente joint.

<https://docs.unity3d.com/Manual/joints-section.html>

FixedJoint crea una giunzione fissa tra l'oggetto su cui è presente e un altro oggetto. Il risultato è una giunzione fissa in cui uno degli oggetti segue fisicamente i movimenti dell'altro (simile a come un gameObject figlio segue un GameObject genitore).

Hingejoint crea una giunzione cardine tra l'oggetto su cui è presente e un altro oggetto. Il risultato è un vincolo fisso tra i due oggetti i quali hanno solo la possibilità di ruotare sull'asse del cardine.

SpringJoint crea una giunzione molleggiata tra l'oggetto su cui è presente e un altro oggetto.

Il risultato è un vincolo tra i due oggetti che segue la fisica di una molla.

Ray (ostacoli e proiettili)

La classe Ray permette di creare raggi per sondare la presenza di oggetti partendo da un punto e una direzione.

La classe RaycastHit viene utilizzata per memorizzare le collisioni con i raggi.

La funzione Physics.Raycast(.....) viene utilizzata per verificare la presenza di collisioni lungo il raggio.



<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>

```
Ray myRay = new Ray(myRayStartPoint,myRayDirection);
//Ray(transform.position, transform.forward) is a ray that starts from the object and
is directed forward
RaycastHit myHit;
float rayDistance = 100.0;
if (Physics.Raycast(myRay, out myHit, rayDistance )){
    Debug.Log(myHit.collider.name); //we can get the collider of the hit object and
its name or other components
}
```

<https://docs.unity3d.com/Manual/CameraRays.html>

La classe Physics fornisce molte funzionalità utili per la fisica.

<https://docs.unity3d.com/ScriptReference/Physics.html>

<https://docs.unity3d.com/Manual/PhysicsSection.html>

Rendering basics

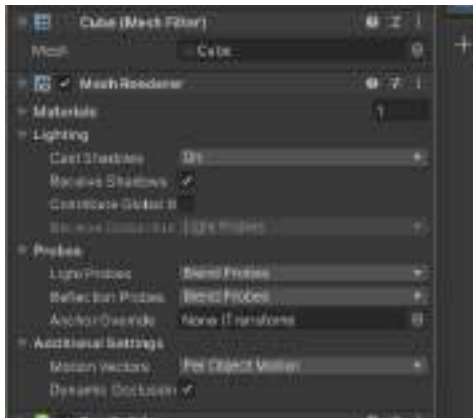
Ci sono diverse componenti che caratterizzano la grafica con cui gli oggetti vengono visualizzati.

La componente mesh filter definisce la forma del nostro GameObject.

<https://docs.unity3d.com/Manual/mesh.html>

Con la componente Mesh Render è possibile definire se le mesh del nostro GameObject devono proiettare e ricevere ombre e come devono comportarsi quando riflettono la luce.

<https://docs.unity3d.com/Manual/class-MeshRenderer.html>



Shaders

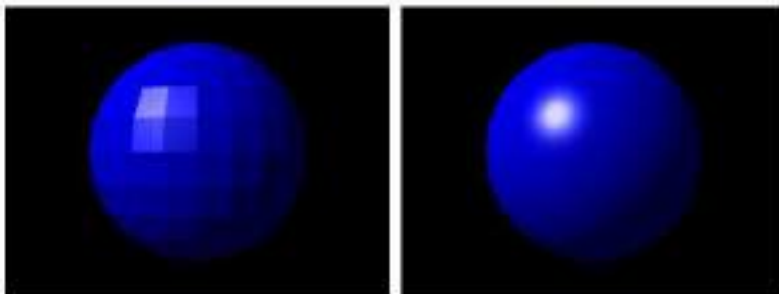
Gli shaders sono dei modelli matematici utilizzati da Unity per indicare alla scheda grafica come rendere a schermo i nostri oggetti tenendo conto di luci, ombre, effetti...

Ogni shader esegue calcoli differenti e ottiene effetti visivi differenti (es ombre realistiche, effetto cartoon, effetto pixel...)

<https://docs.unity3d.com/Manual/Shaders.html>

I vertex shaders calcolano riflessi e ombre su una faccia della mesh calcolando la media di ciò che accade sui vertici della faccia (più veloce ma meno qualità. Riflessi e ombre seguono la forma della mesh).

I pixel shaders calcolano luci e ombre tenendo conto di tutto ciò che interessa un oggetto presente in un singolo pixel dello schermo (più lento ma con qualità maggiore, le ombre e i riflessi sono indipendenti dalla forma della mesh su cui sono posizionate).



Materiali

I materiali definiscono i valori che gli shaders devono utilizzare durante i calcoli grafici.

Utilizzando lo stesso shader è possibile definire materiali con colori, trasparenze, lucentezza, opacità differenti.

Non tutti gli shader riescono a gestire tutte le proprietà (es trasparenza).



A.M.E.F.E.
asociación italiana de
educación y formación europea



 **Erasmus+**
Enriching lives, opening minds.



<https://docs.unity3d.com/Manual/Materials.html>

Texture

Le texture possono definire il colore e altre proprietà dei materiali pixel per pixel.

Importare una texture in Unity è tanto semplice quanto trascinarla nei propri Assets. Per utilizzarla in un materiale basta trascinarla nella proprietà del materiale per cui vogliamo utilizzarla. (es albedo).

Texture ad alta risoluzione complicano e allungano i calcoli grafici in maniera esponenziale.

<https://docs.unity3d.com/Manual/Textures.html>

Fonti di Luce

In Unity esistono differenti fonti luminose.

Point light è una luce che è posizionata in un punto della scena ed emette raggi di luce uguali in tutte le direzioni.

Spot Light è una luce che è posizionata in un punto della scena ed emette luce in una forma di un cono.



A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

Directional Light è una luce che è posizionata infinitamente lontana ed emette luce in una sola direzione.

Area light è una luce che è definita da un poligono nella scena ed emette luce in maniera uniforme, in tutte le direzioni attraverso la sua area di superficie ma solo da un lato.

Per ogni fonte di luce è possibile definire la sua intensità, il suo colore, in alcuni casi la direzione, la distanza di effetto e se questa produce o meno delle ombre.

<https://docs.unity3d.com/Manual/Lighting.html>

<https://docs.unity3d.com/Manual/LightingOverview.html>



Su dispositivi mobile è sconsigliato l'utilizzo delle ombre



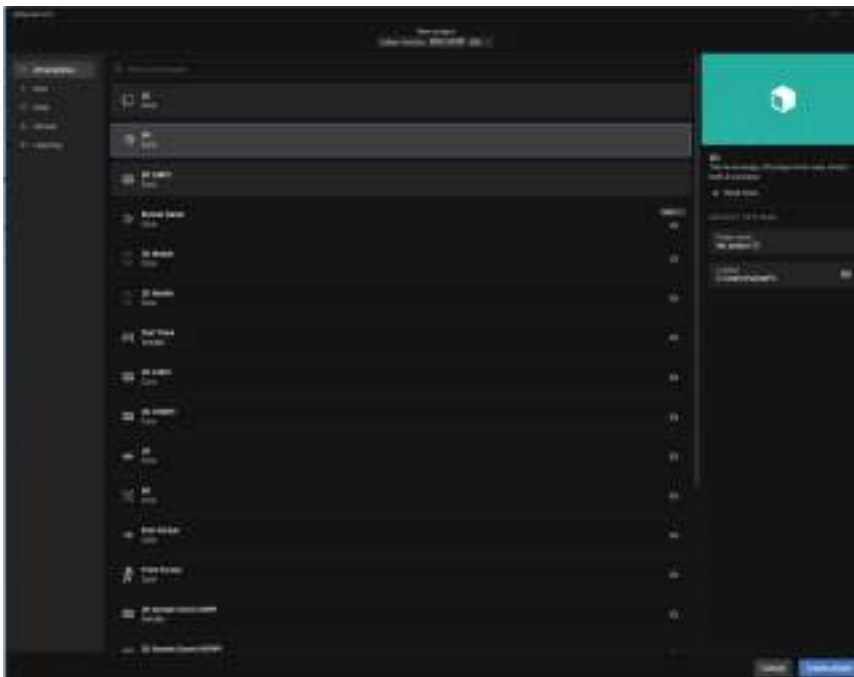
Unità 5 - Guida pratica con i codici del videogioco PLAY.

In questo capitolo vedremo come realizzare alcune delle funzionalità presenti nel videogioco.

Creiamo un nuovo progetto premendo il pulsante New project



Selezioniamo “3D-core” , impostiamo il nome del progetto e selezioniamo “Create Project”



Una volta creato comparirà l'interfaccia utente Unity.

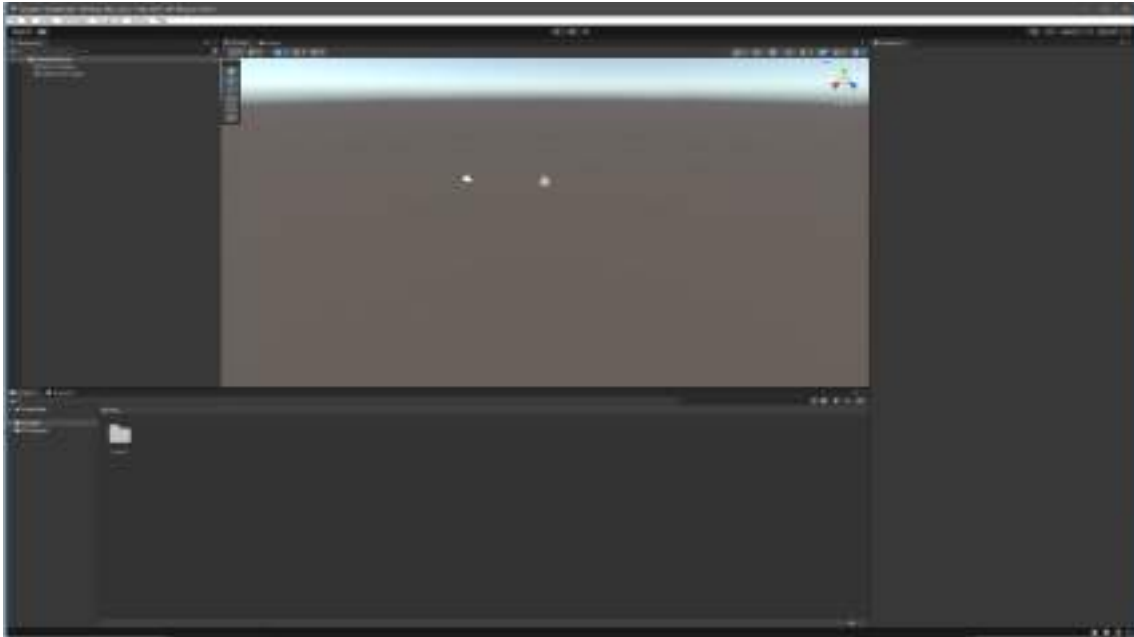


A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea

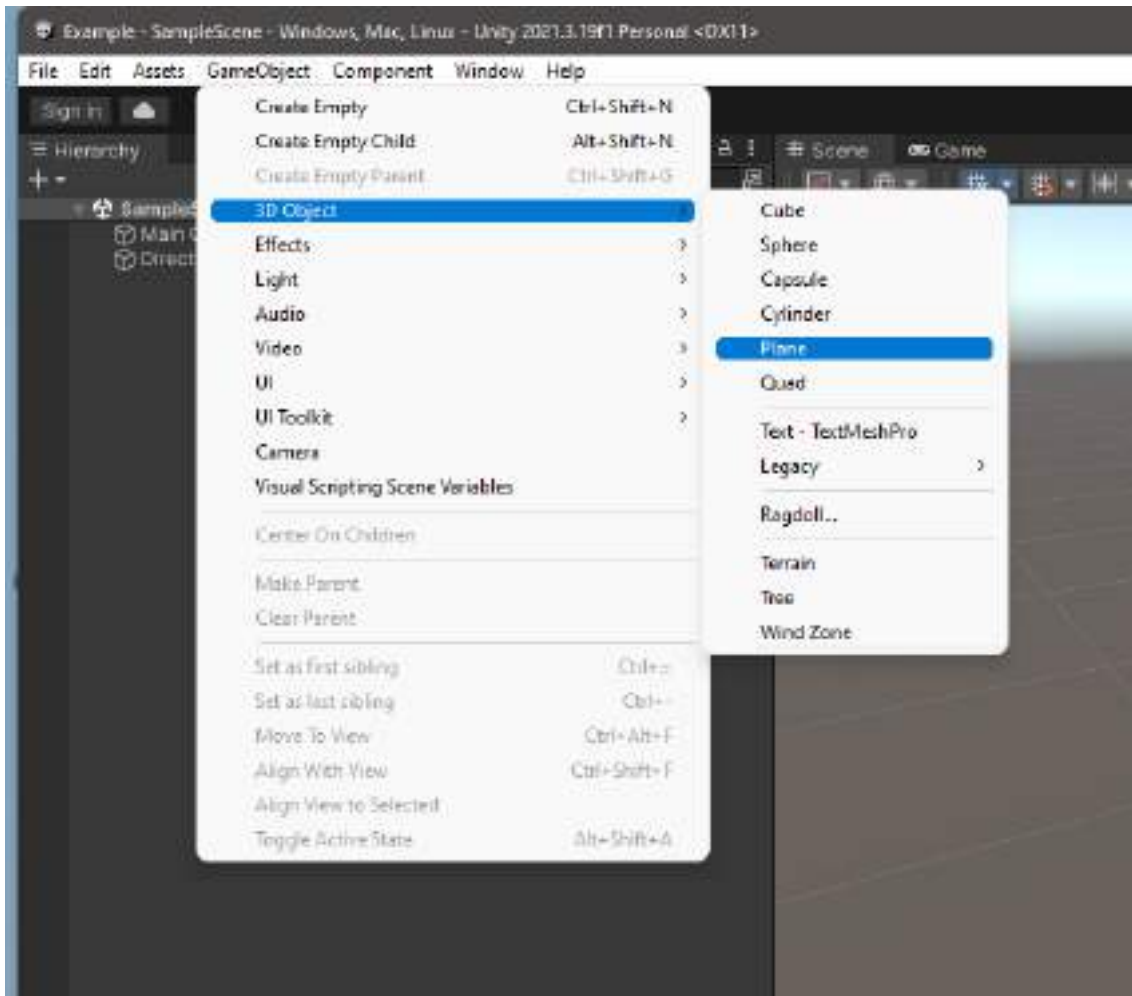


 **Erasmus+**
Enriching lives, opening minds.

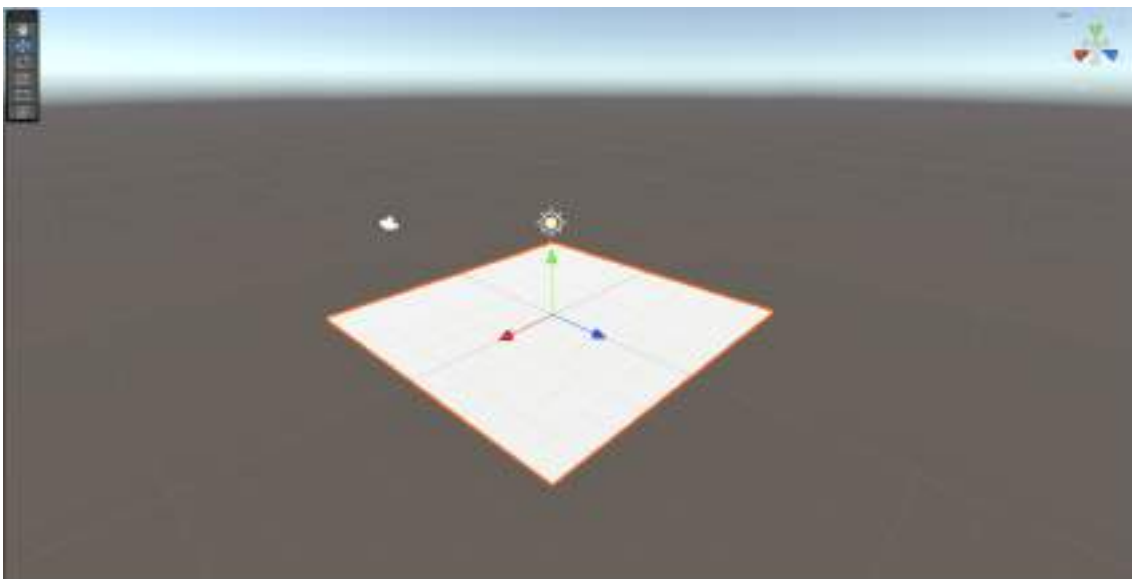




Aggiungiamo un piano dal menu GameObject -> 3D Object -> Plane



Il piano apparirà così.





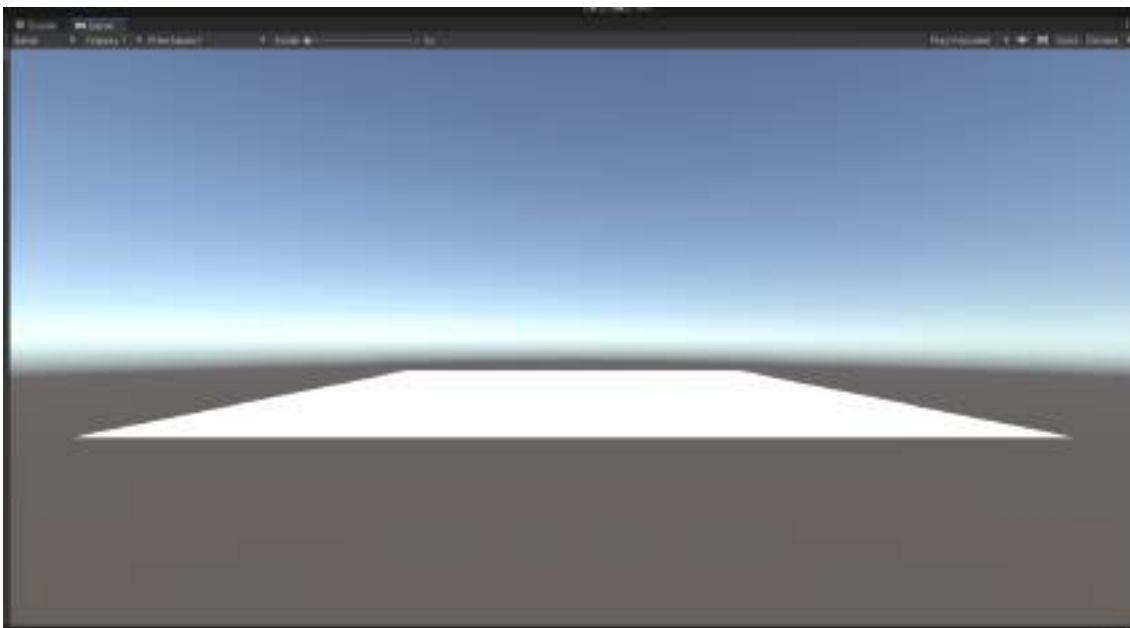
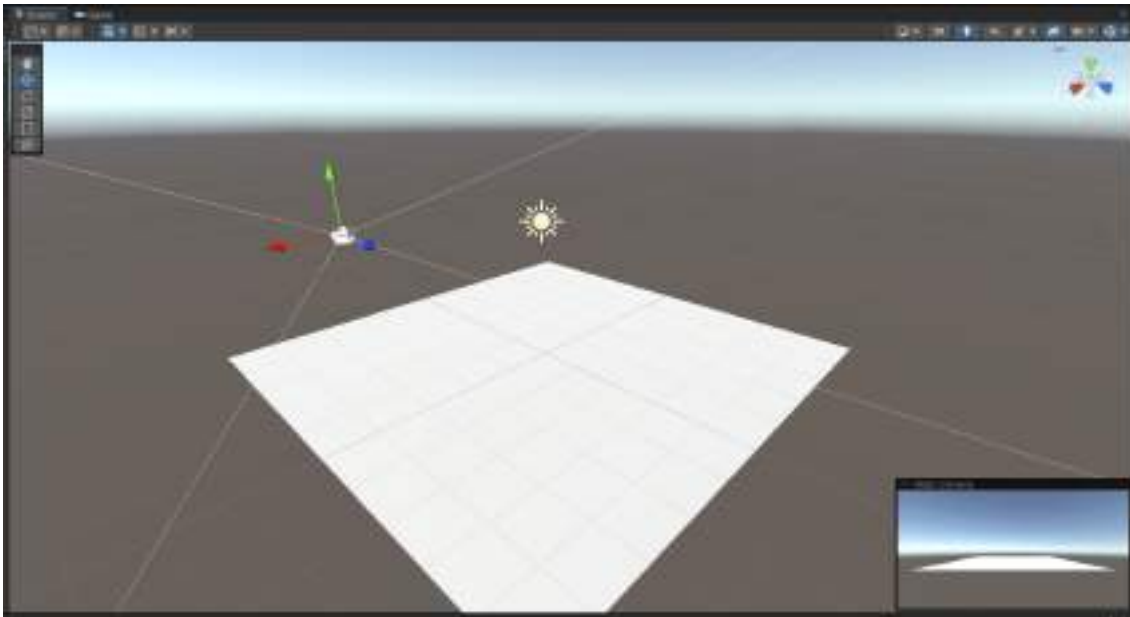
A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

Selezioniamo la videocamera e assicuriamoci che il piano sia inquadrato correttamente





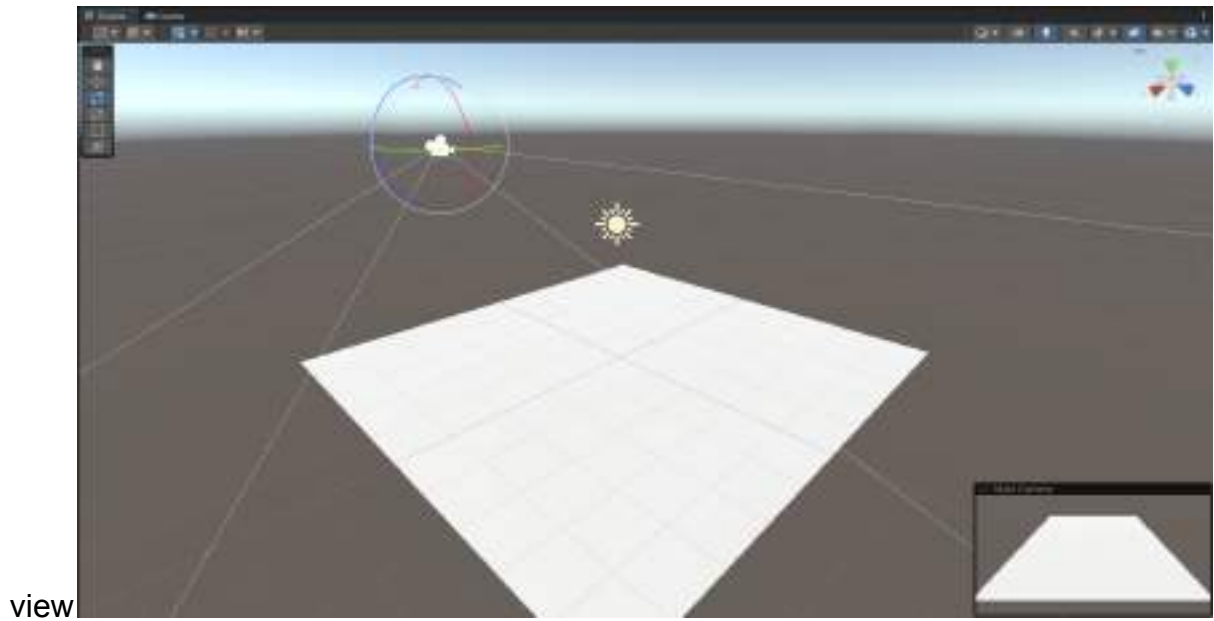
A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

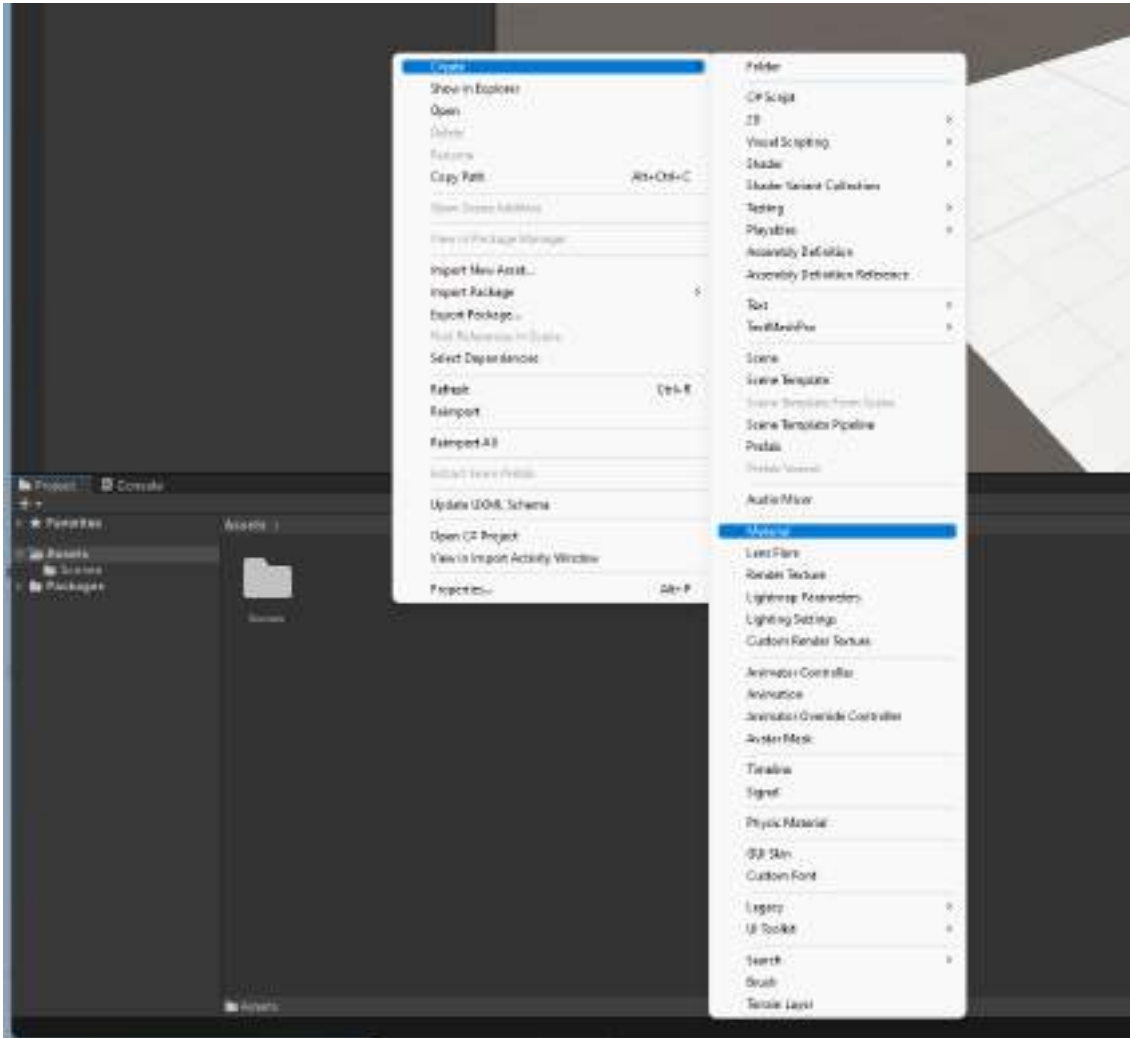
Muoviamo e ruotiamo la videocamera con gli appositi tool sul lato sinistro della scene





Creiamo un nuovo materiale per colorare il piano di verde.

Facciamo click con il tasto sinistro del mouse sul pannello Assets, Create -> Material

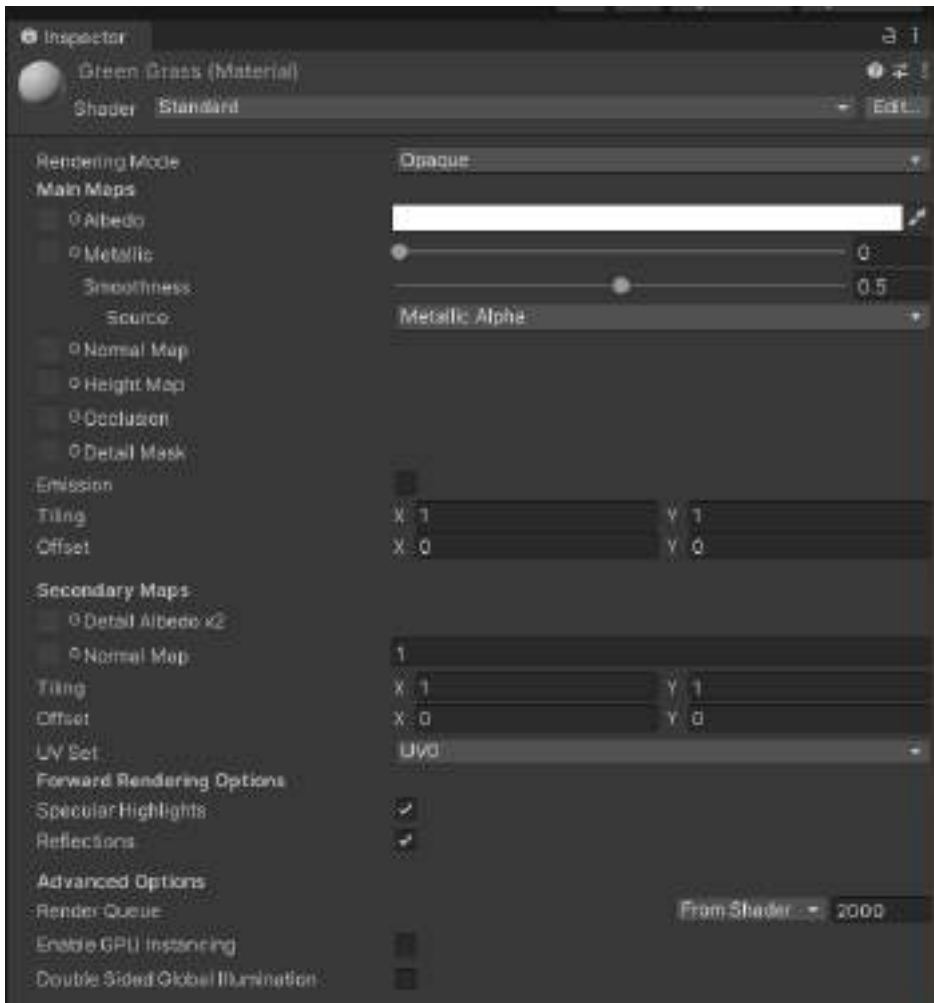


Soprannominiamo il materiale





Impostiamo il colore nel pannello “inspector” premendo sul rettangolo colorato in bianco.





Scegliamo il colore





A.M.E.F.E.
Asociación Española de
edición y formación europea

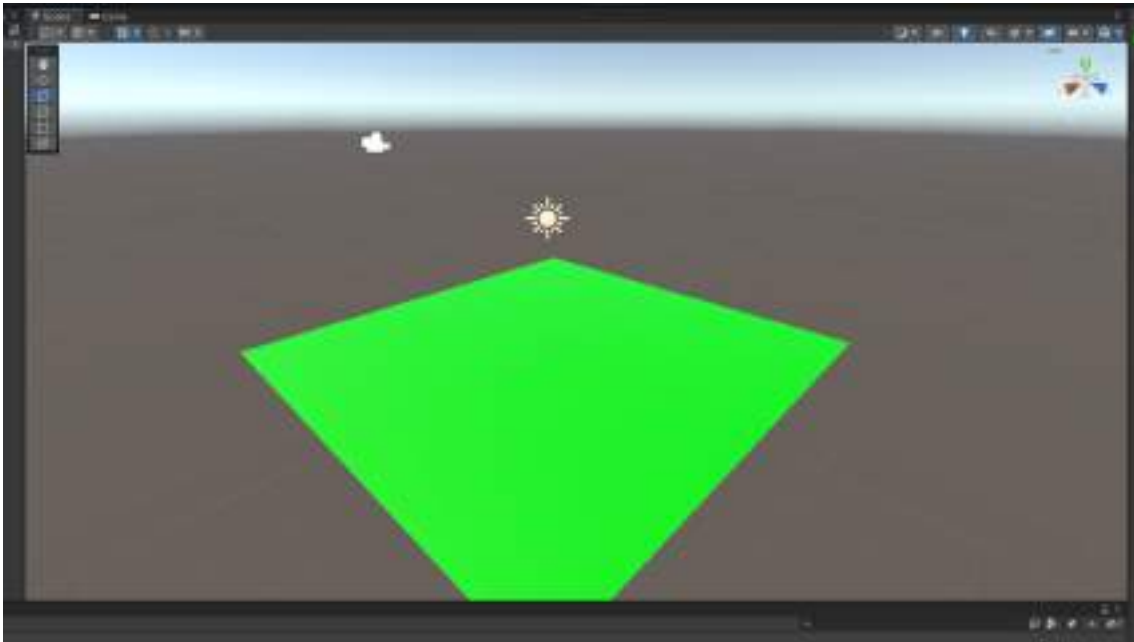
Paidea



 **Erasmus+**
Enriching lives, opening minds.



Trasciniamo il materiale dal Assets e portiamo sul piano nella scene view per cambiare il materiale del piano





A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

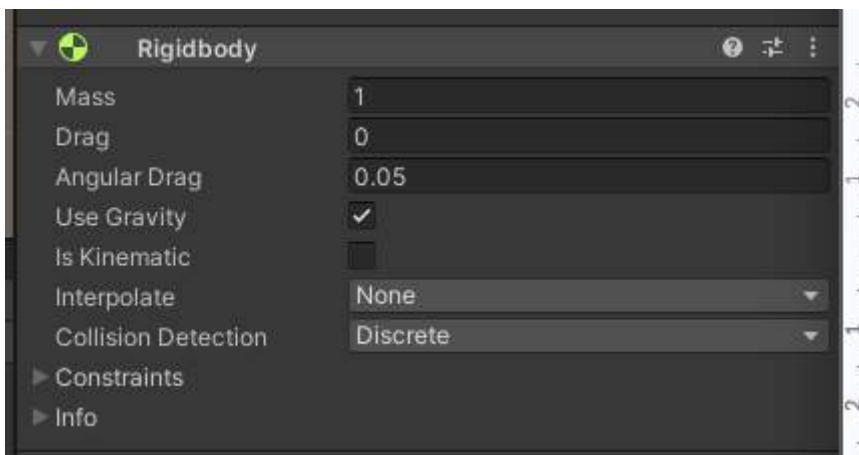
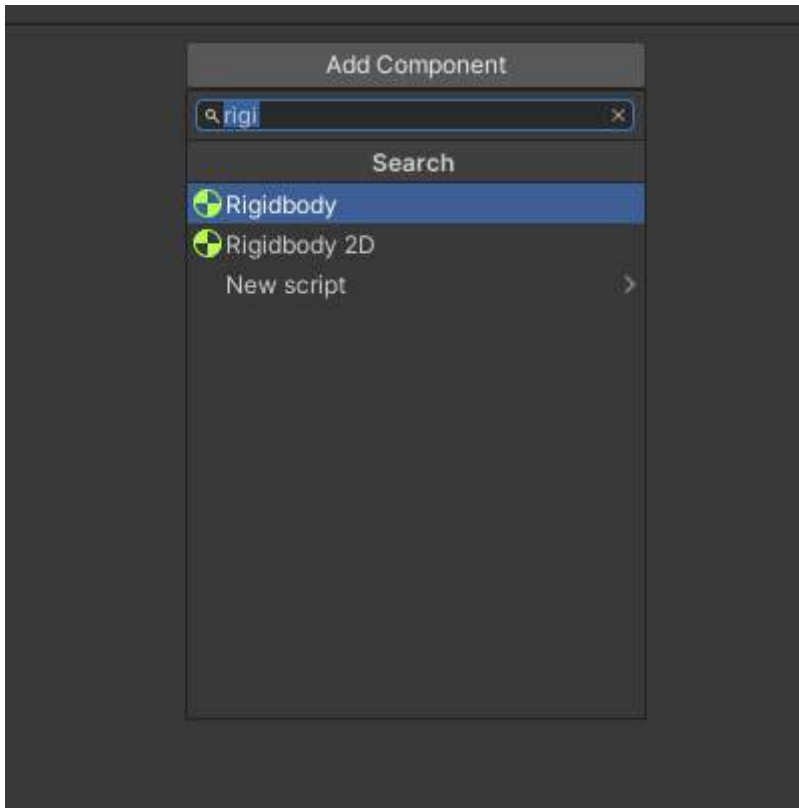
Con lo stesso metodo creiamo un cubo e coloriamolo di azzurro



Aggiungiamo la componente Rigidbody:

Selezioniamo il cubo, premiamo Add Component sul pannello Inspector, cerchiamo e selezioniamo "Rigidbody".

Questo permetterà al cubo di essere controllato utilizzando la fisica e di essere soggetto alla gravità.

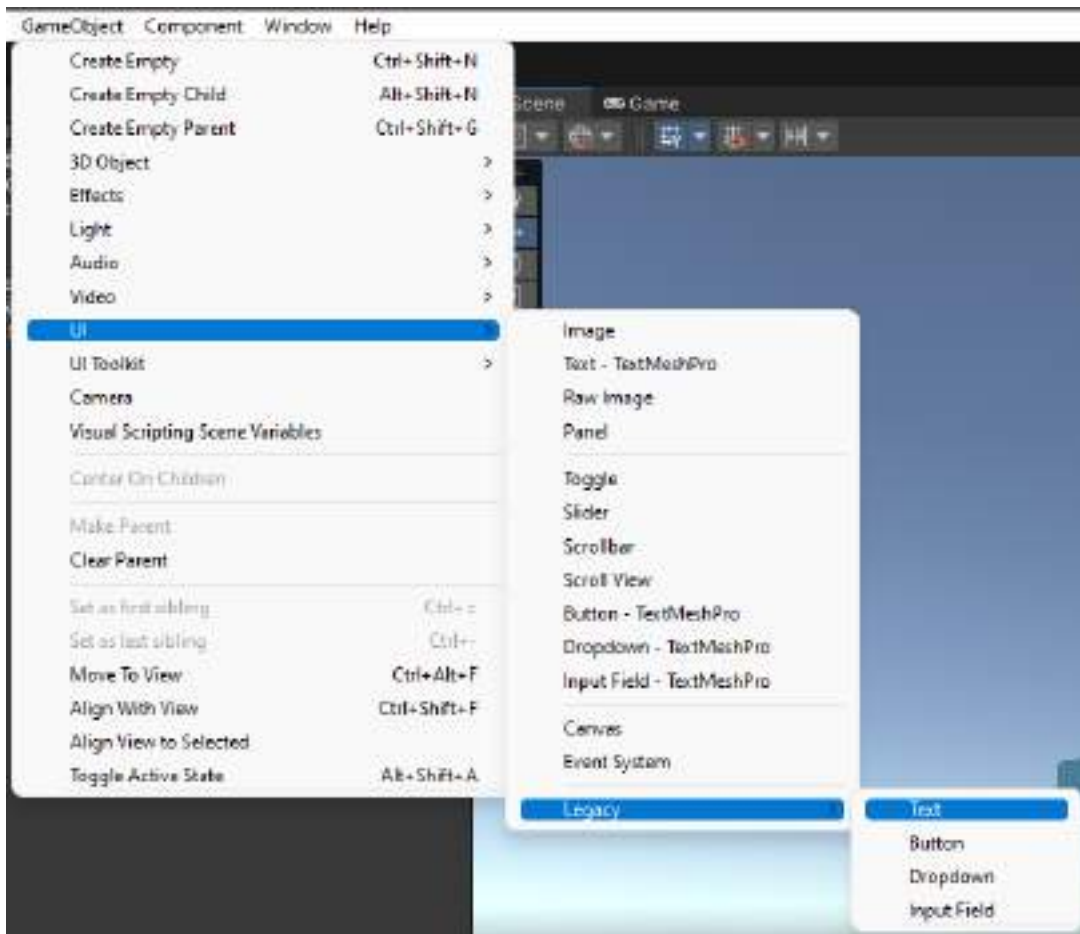


Seleziona Freeze Rotation (X, Y, Z) se vuoi impedire al cubo di ruotare durante gli spostamenti

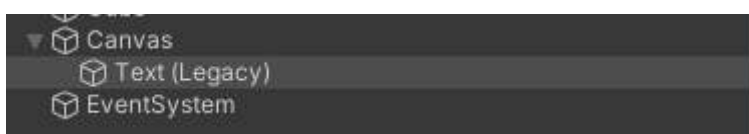


(suggerimento: invece di tener premuto il tasto sinistro del mouse e muoverti usando WASD puoi selezionare un oggetto dal pannello “Hierarchy” e premere SHIFT+F per focalizzare la visuale direttamente su quell’oggetto)

Aggiungiamo del testo in sovrapposizione per comunicare con l’utente:
GameObject -> UI -> Legacy -> Text



Comparirà un “Canvas” (Tela) molto grande dove potremo aggiungere testi e altri elementi per l’interfaccia grafica utente



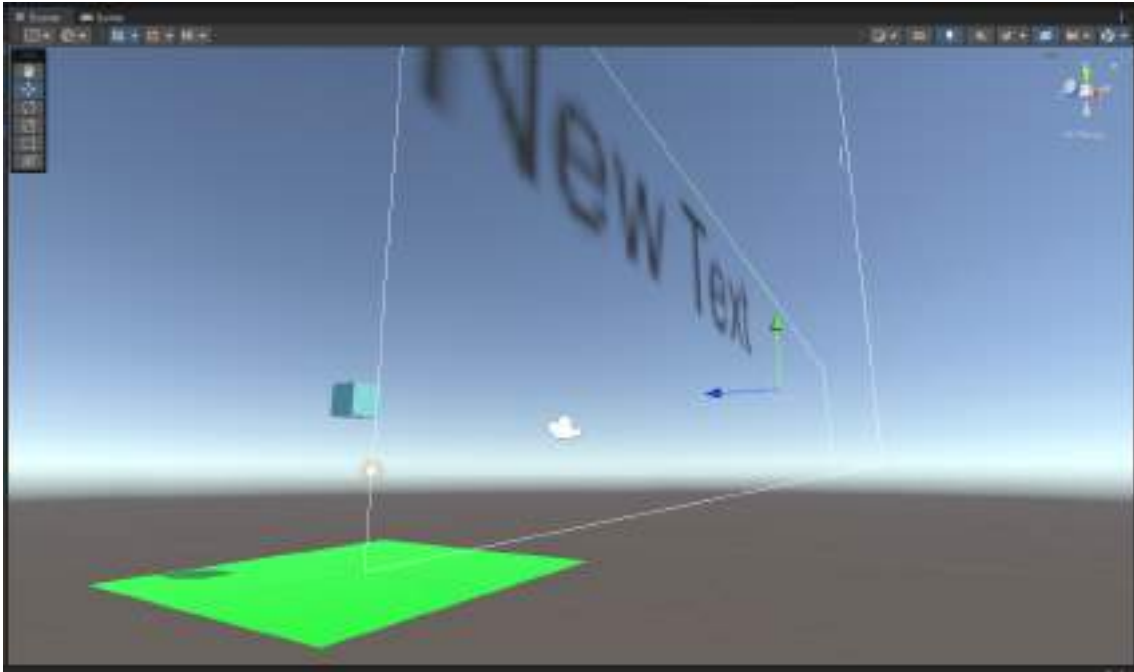


A.M.E.F.E.
Asociación Italiana de
Educación y Formación Europea

Paidea



Erasmus+
Enriching lives, opening minds.



Seleziona Text (Legacy)

Cambia width, height, text e font size dal pannello Inspector;



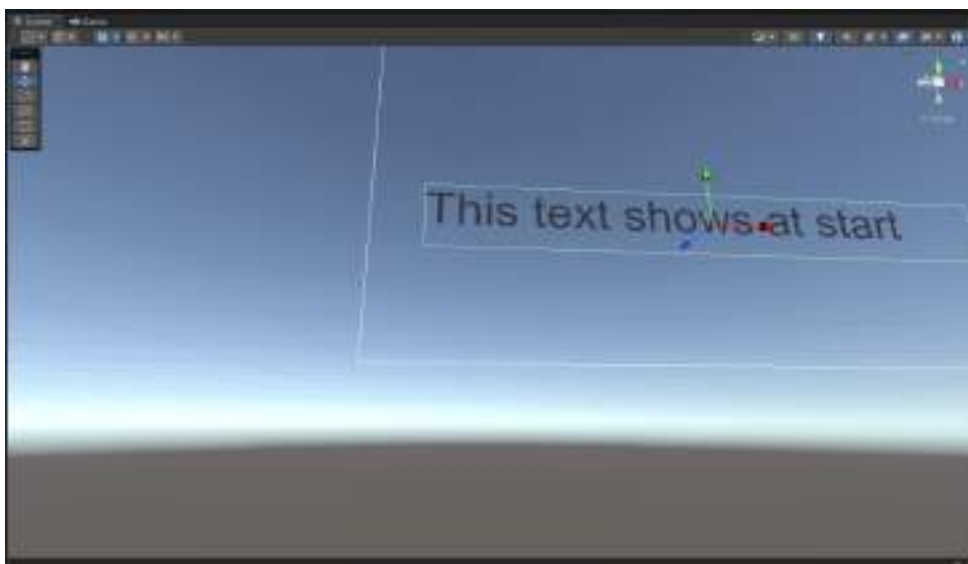
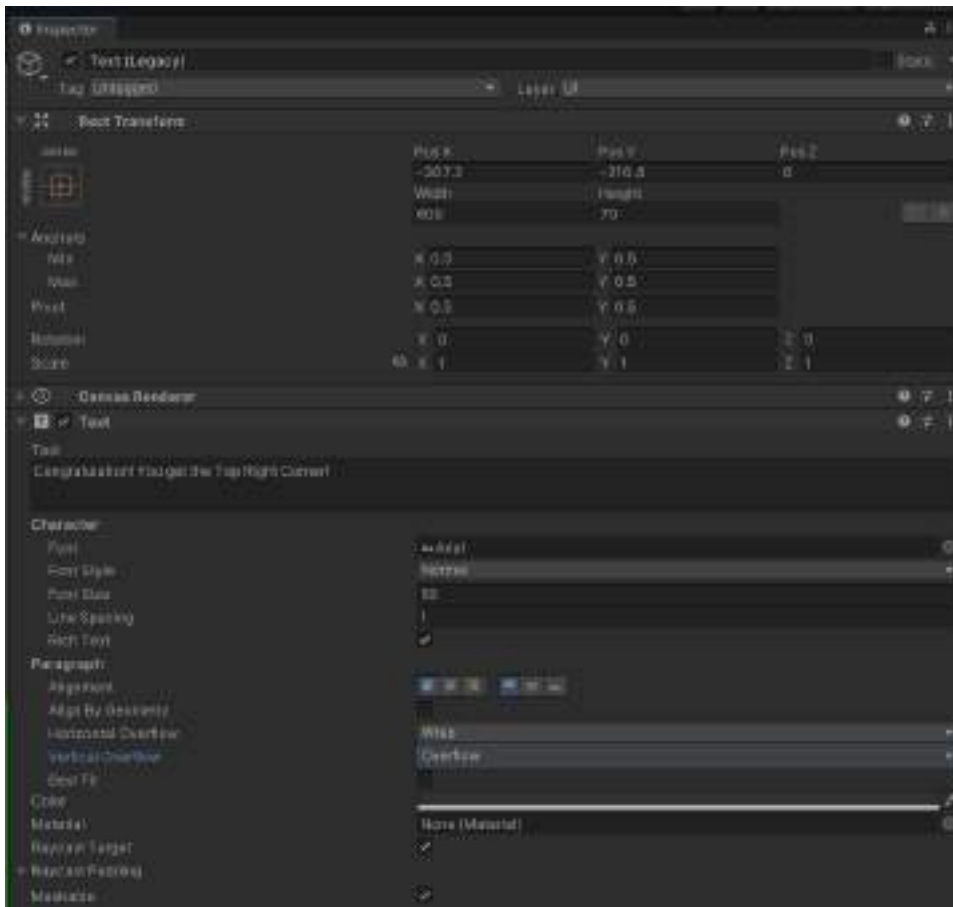
A.M.E.F.E.
Asociación Española de
Educación y Formación Europeas

Traidea



 **Erasmus+**
Enriching lives, opening minds.

Sposta il testo nella posizione desiderata.





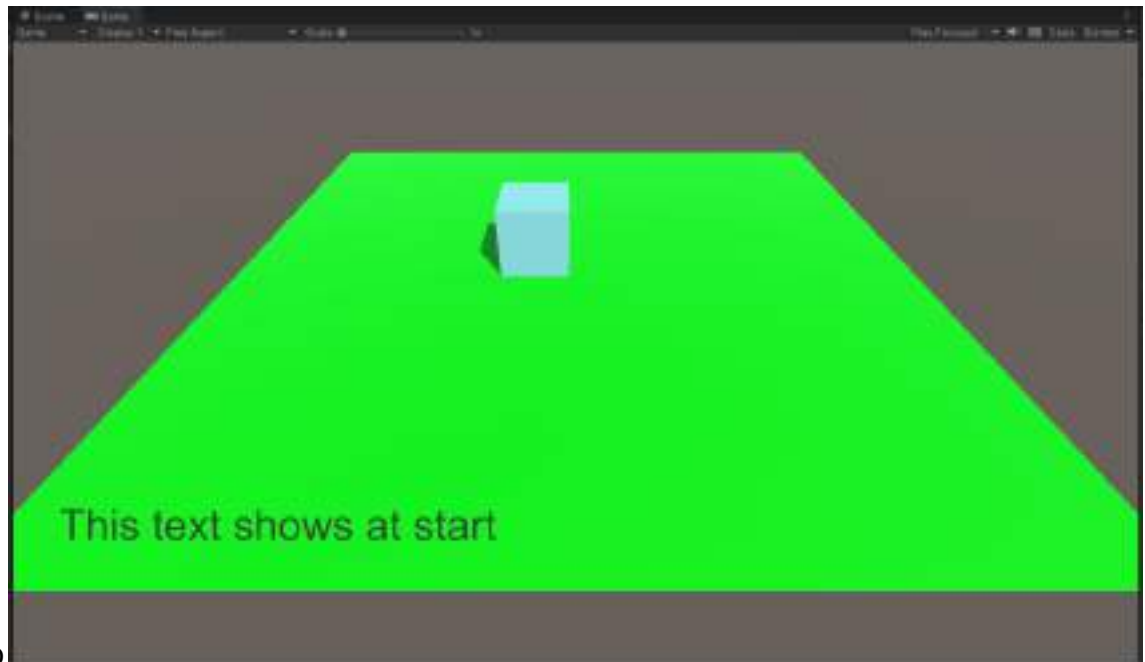
A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

La posizione del testo nei canvas corrisponde alla posizione sullo schermo durante il



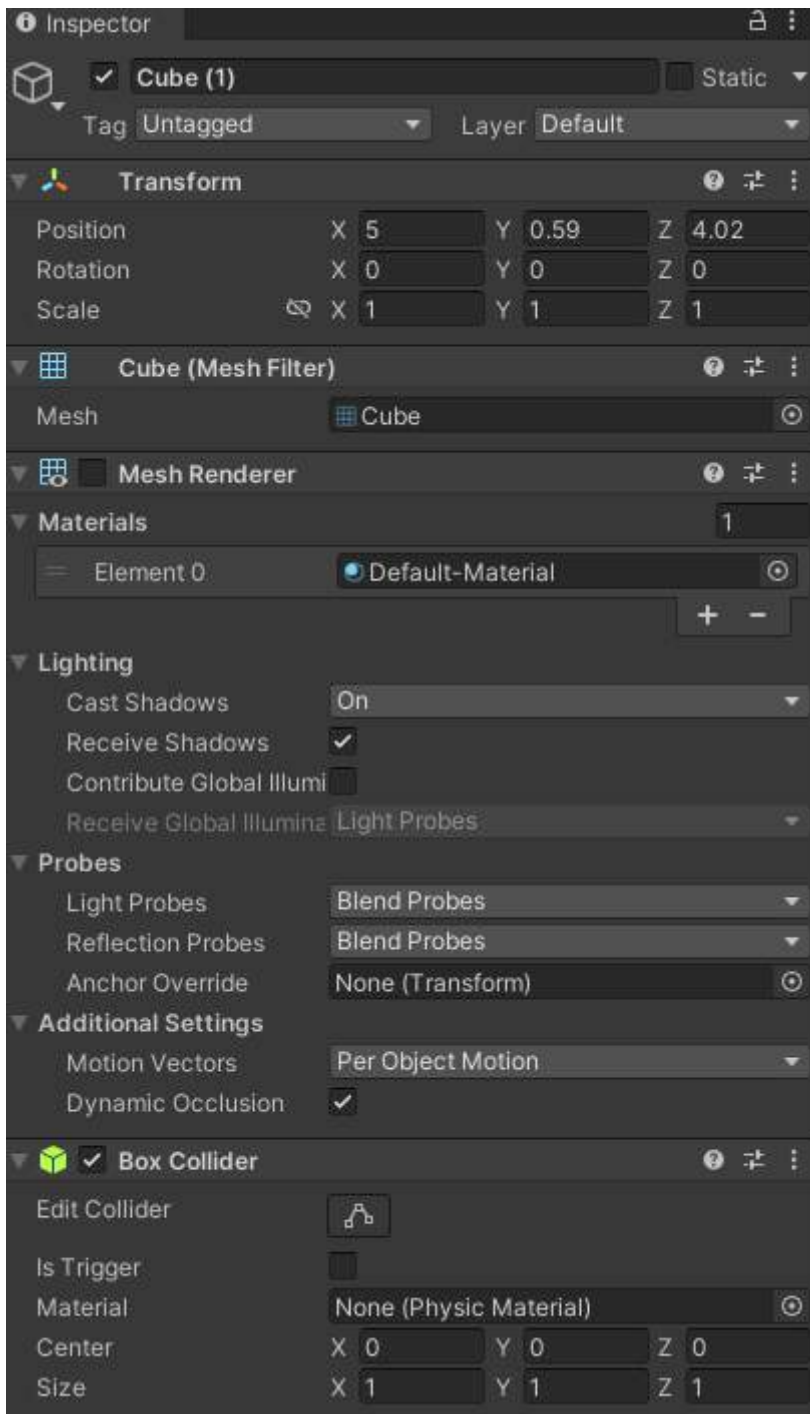
gioco

Creiamo un altro cubo:

In Inspector -> Box Collider spuntiamo la spunta is Trigger per trasformare il nostro cubo in un'area che dà il via a un evento.



Togliamo la spunta da Mesh Render per rendere invisibile l'area appena creata.





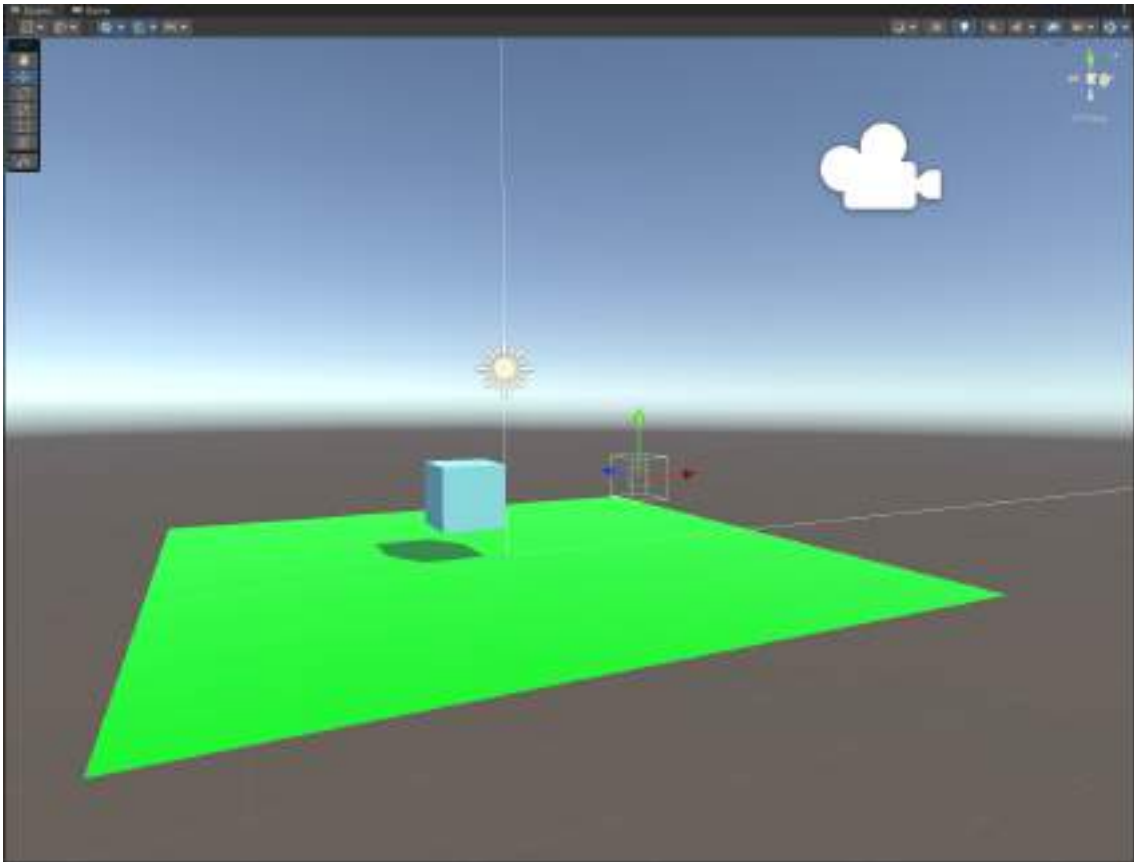
A.M.E.F.E.
Asociación Española de
Educación y Formación Europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

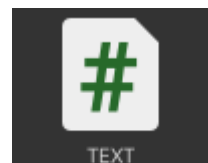
Posizioniamo il nuovo cubo nell'angolo in alto a destra del nostro piano



Aggiungiamo una componente script con nome TEXT



Facciamo doppio click sul file apparso in assets
aggiungiamo questa istruzione di codice a riga 8



e

```
Public string textToShow;
```



```
TEXTes - X
Miscellaneous Files
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5
6 public class TEXT : MonoBehaviour
7 {
8     public string textToShow; //public variables can be used from other scripts
9
10    // Start is called before the first frame update
11    void Start()
12    {
13
14    }
15
16    // Update is called once per frame
17    void Update()
18    {
19
20    }
21 }
22
```

SALVIAMO e chiudiamo l'editor di codice.

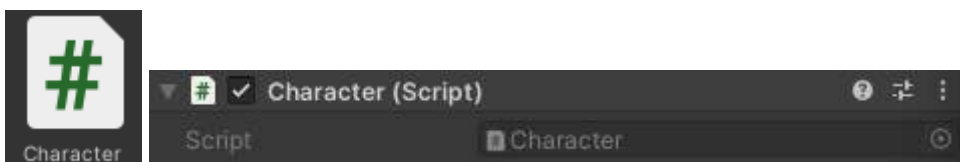
L'istruzione inserita ci permette di utilizzare e impostare la stringa textToShow dal pannello inspector.

Cambiamo il valore di Text To Show in *“Congratulazioni! Hai raggiunto l'angolo in alto a destra!”*



Selezioniamo il cubo azzurro e aggiungiamo una componente script anche al cubo azzurro.

Soprannominiamo la script “Personaggio”



Apriamo lo script aggiungiamo questa riga di codice dopo la prima parentesi graffa aperta prima di void Start()



```
public float forceAmount = 0.8f; //questa riga imposta il valore della forza
```

Inseriamo le seguenti righe di codice tra le due parentesi graffe della funzione Update

```
Vector3 direction = new Vector3(Input.GetAxis("Horizontal"), 0,
Input.GetAxis("Vertical")); //ottiene la direzione utilizzando gli assi "Horizontal" e "Vertical"
mappati sui tasti WASD o sulle levette analogiche
direction=direction*forceAmount; //moltiplica la direzione per la forza
rigidbody.AddForce(direction); //applica la forza utilizzando il vettore direzione
```

Queste istruzioni ci permetteranno di muovere il nostro cubo azzurro utilizzando i tasti WASD della tastiera.

Il codice ottenuto sarà simile a questo:

```
using UnityEngine;
using System.Collections.Generic;
using UnityEngine.Events;

public class Movimento : MonoBehaviour
{
    public float forceAmount = 0.8f; //questo valore imposta la forza
    //questo valore imposta la forza

    void Update()
    {
        //ottiene la direzione utilizzando gli assi "Horizontal" e "Vertical"
        Vector3 direction = new Vector3(Input.GetAxis("Horizontal"), 0,
        Input.GetAxis("Vertical")); //ottiene la direzione utilizzando gli assi "Horizontal" e "Vertical"
        direction=direction*forceAmount; //moltiplica la direzione per la forza
        rigidbody.AddForce(direction); //applica la forza utilizzando il vettore direzione
    }
}
```

SALVIAMO

Abilitiamo la possibilità di utilizzare le istruzioni relativi all'UI aggiungendo quanto segue in riga 4

```
using UnityEngine.UI;
```



A.M.E.F.E.
asociación española de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

Inseriamo la stringa seguente successivamente alla dichiarazione di “forceAmount”

```
public Text textToChange;
```

La riga di codice precedente ci permetterà di collegare textToChange al testo presente in Canvas utilizzando il pannello Inspector.

Aggiungiamo questo codice prima della chiusura dell’ultima parentesi graffa

```
void OnTriggerEnter(Collider Other)  
{  
    textToChange.GetComponent<Text>().text =  
    Other.gameObject.GetComponent<TEXT>().textToShow;  
}
```

La funzione precedente cambierà il testo in sovraimpressione con quello presente nel codice del cubo invisibile impostato come area di trigger.



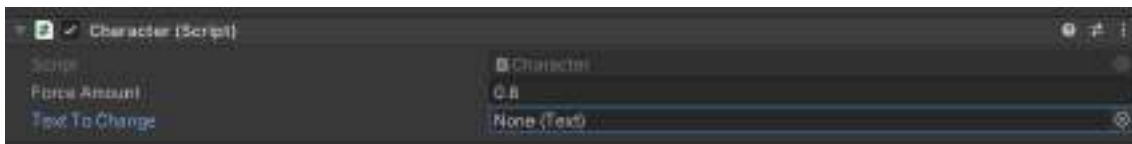
Il codice ottenuto sarà simile a questo:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

SALVIAMO e chiudiamo l'editor di codice.

Per poter cambiare il testo dell'UI bisogna collegarlo al cubo azzurro utilizzando il pannello Inspector del cubo azzurro.

Impostiamo Character(script) -> Text To Change.



Trasciniamo Text (Legacy) dal pannello Hierarchy su Character(script) -> Text To Change nel pannello inspector del cubo azzurro.



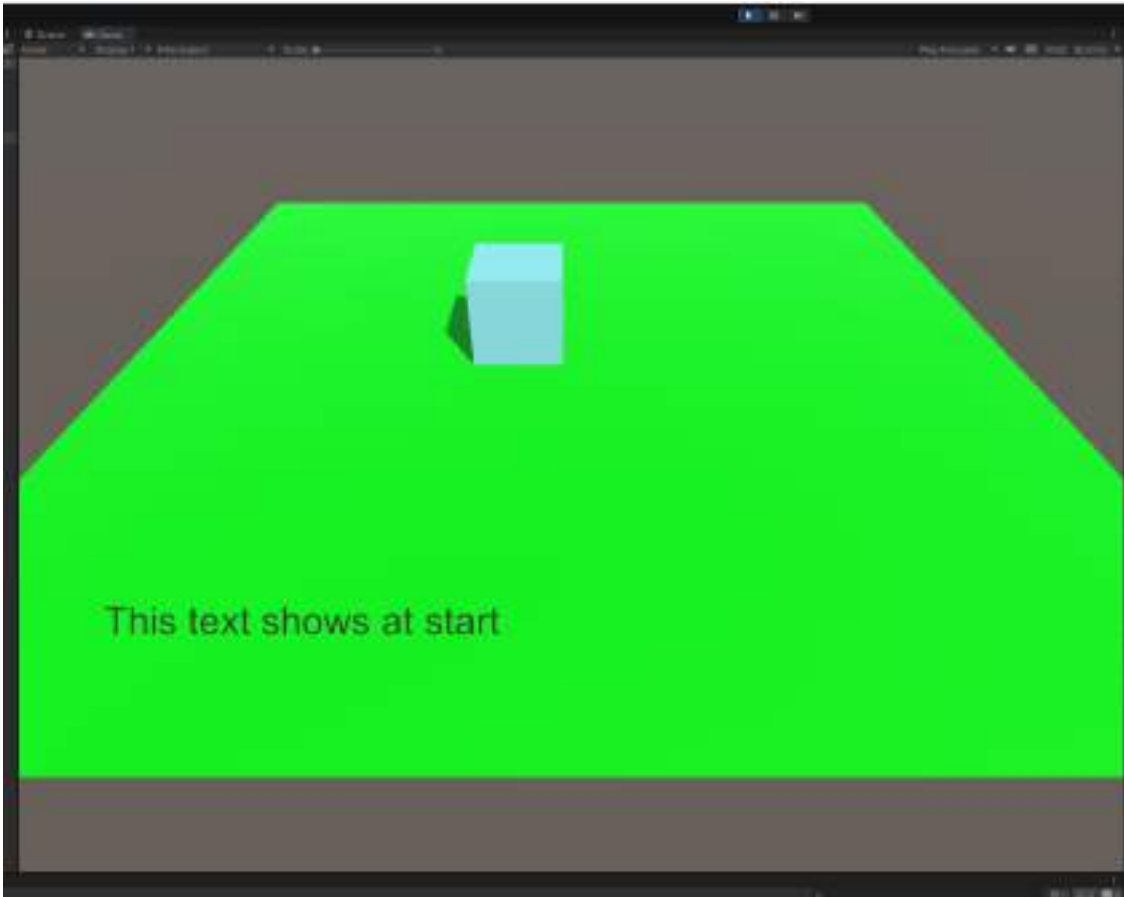
A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

SALVIAMO il progetto e premiamo play per testare il nostro gioco.



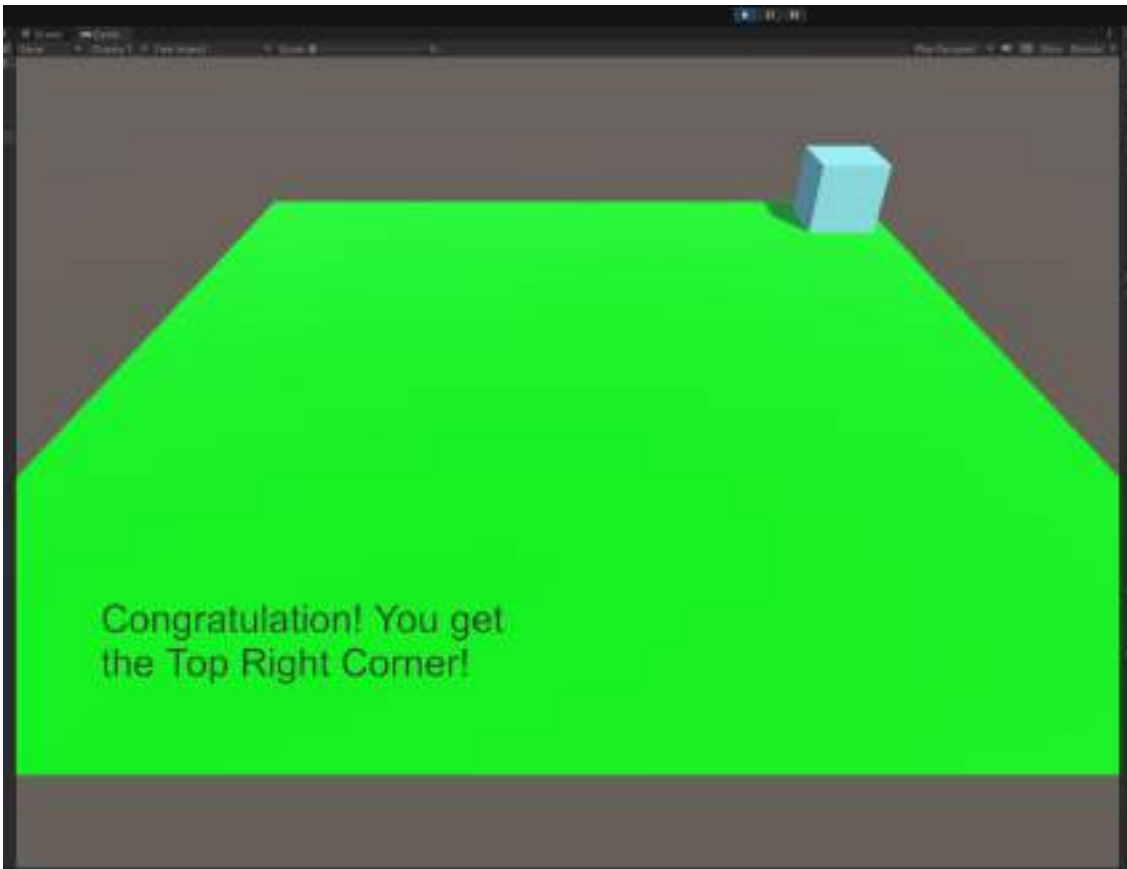


A.M.E.F.E.
asociación italiana de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.



Possiamo copiare e incollare più volte il cubo invisibile per creare aree differenti dove compariranno testi differenti.



A.M.E.F.E
Asociación Madrileña de
Educación y Formación Europeas

Paidea



 **Erasmus+**
Enriching lives, opening minds.

-  Cube (1)
-  Cube (2)
-  Cube (3)
-  Cube (4)
-  Cube (5)

