

**UNIVERSITÀ POLITECNICA DELLE MARCHE**  
**FACOLTÀ DI INGEGNERIA**

Dipartimento di Ingegneria dell'Informazione  
Corso di Laurea in Ingegneria Informatica e dell'Automazione

---



**TESI DI LAUREA**

**Integrazione di Realtà Virtuale e Natural Language Understanding  
per la creazione di un Serious Game di tipo Educational**

**Integration of Virtual Reality and Natural Language Understanding  
to create an Educational Serious Game**

Relatore

Prof. Domenico Ursino

Candidato

Vincenzo Guadagnini

---

**ANNO ACCADEMICO 2022-2023**

*È nel momento delle decisioni che si plasma il tuo destino*

Anthony Robbins

## Sommario

La sinergia tra la realtà virtuale (VR) e la comprensione del linguaggio naturale (NLU) all'interno del dominio dei Serious Game si pone l'obiettivo di promuovere un nuovo approccio educativo mediante l'adozione di tecnologie emergenti. Per mezzo di un'indagine dettagliata sulle tendenze correnti e prospettive, il presente studio evidenzia l'importanza cruciale della VR e dell'NLU come veicoli di innovazione pedagogica, sottolineando la loro capacità di modificare in modo radicale i paradigmi tradizionali di apprendimento. Utilizzando un approccio multidisciplinare, la ricerca presenta lo sviluppo di 'Dom the Robot', un Serious Game che sfrutta queste innovazioni per creare un ambiente di apprendimento coinvolgente e interattivo. La tesi dettaglia il processo di progettazione, sviluppo e valutazione del gioco, discutendo le sfide e le strategie adottate per integrare efficacemente le nuove tecnologie. I risultati dipingono un quadro in cui le cooperazioni tra ambienti virtuali e Intelligenza Artificiale tessono un elegante ponte verso nuovi orizzonti educativi, sottolineando il ruolo cruciale della Virtual Reality e del Natural Language Understanding nel plasmare esperienze di apprendimento immersive e significative. È un viaggio che intreccia scienza e pedagogia con l'obiettivo di trasformare l'educazione in un giardino rigoglioso di conoscenza, dove ogni seme di informazione può sbocciare in una comprensione profonda, irrigata dalle innovazioni tecnologiche.

**Keyword:** Serious Game, Virtual Reality (VR), Artificial Intelligence (AI), Natural Language Understanding (NLU), Digital Education, Unity, Hugging Face, Innovation

<b>Introduzione</b>	<b>1</b>
<b>1 Serious Game: Innovazione, Apprendimento e Applicazioni nel Contesto Educativo</b>	
<b>Moderno</b>	<b>4</b>
1.1 Cosa sono i Serious Game? . . . . .	4
1.2 Evoluzione e significato nel contesto moderno . . . . .	5
1.2.1 Dalle origini ad Oggi: La Trasformazione dei Serious Game . . . . .	5
1.2.2 Rilevanza Attuale: I Serious Game nel Mondo Moderno . . . . .	6
1.2.3 Dinamiche di Cambiamento: Innovazioni e Impatti . . . . .	6
1.3 Apprendimento Tramite i Serious Game: Metodologie ed Efficacia . . . . .	7
1.3.1 Principi Pedagogici e Strategie Didattiche nei Serious Game . . . . .	7
1.3.2 Valutazione dell'Efficacia: Risultati e Impatti Educativi . . . . .	8
1.4 Applicazioni Pratiche . . . . .	8
1.4.1 Esempi specifici in diversi settori . . . . .	8
1.4.2 Confronto con strumenti educativi tradizionali . . . . .	10
<b>2 Esplorazione delle Tecnologie Immersive: Realtà Aumentata, Virtuale e Mista</b>	<b>12</b>
2.1 Fondamenti della Realtà Aumentata . . . . .	12
2.1.1 Definizione e Storia della Realtà Aumentata . . . . .	12
2.1.2 Tecnologie e Dispositivi per la Realtà Aumentata . . . . .	14
2.1.3 Applicazioni Pratiche della Realtà Aumentata . . . . .	15
2.2 Realtà Virtuale: Immersione e Interazione . . . . .	16
2.2.1 Concetti Base della Realtà Virtuale . . . . .	16
2.2.2 Hardware e Software per la Realtà Virtuale . . . . .	16
2.2.3 Realtà Virtuale in Educazione e Formazione . . . . .	18
2.3 Realtà Mista: Unire il Fisico e il Digitale . . . . .	18
2.3.1 Introduzione alla Realtà Mista . . . . .	18
2.3.2 Tecnologie chiave della Realtà Mista . . . . .	18

---

2.3.3	Prospettive Future della Realtà Mista . . . . .	19
<b>3</b>	<b>Comprendere il Linguaggio Naturale: Approcci e Tecnologie nell'era dell'IA</b>	<b>20</b>
3.1	Fondamenti del Natural Language Understanding . . . . .	20
3.1.1	Introduzione al Natural Language Understanding . . . . .	20
3.1.2	Metodologie di Analisi nell'NLU . . . . .	21
3.1.3	Applicazioni dell'NLU . . . . .	21
3.2	Il Ruolo delle Reti Neurali e del Deep Learning . . . . .	22
3.2.1	Fondamenti delle Reti Neurali nell'NLU . . . . .	22
3.2.2	Deep Learning e NLU . . . . .	23
3.2.3	Casi Studio e Esempi Pratici . . . . .	24
3.3	Nuove Frontiere e Strumenti nel Mondo dell'NLU . . . . .	25
3.3.1	Panoramica delle Piattaforme di NLU . . . . .	25
3.3.2	Hugging Face: Un Ecosistema Innovativo . . . . .	26
3.3.3	Tendenze Future e Direzioni di Ricerca . . . . .	27
<b>4</b>	<b>Ideazione e progettazione del serious game "Dom the Robot"</b>	<b>28</b>
4.1	Introduzione, Contesto e Pubblico di Destinazione . . . . .	28
4.1.1	Contestualizzazione del Serious Game . . . . .	28
4.1.2	Obiettivi Educativi e di Apprendimento . . . . .	29
4.1.3	Pubblico di Destinazione e Profilo degli Utenti . . . . .	29
4.2	Analisi dei Requisiti . . . . .	29
4.2.1	Requisiti Funzionali . . . . .	29
4.2.2	Requisiti non Funzionali . . . . .	30
4.3	Progettazione del Serious Game . . . . .	31
4.3.1	Struttura del gioco . . . . .	31
4.3.2	Un modello AI che rende Dom uno "Smart Robot" . . . . .	31
<b>5</b>	<b>Realizzazione Tecnica di "Dom the Robot"</b>	<b>33</b>
5.1	Elementi Grafici e Scenografici . . . . .	33
5.1.1	Il Paesaggio di Background . . . . .	33
5.1.2	Game Objects in Scena . . . . .	34
5.1.3	La Creazione Grafica di Dom . . . . .	35
5.2	Dinamiche di Interazione e Integrazione AI . . . . .	36
5.2.1	La Macchina degli Stati di Dom . . . . .	36
5.2.2	Implementazione API di Hugging Face . . . . .	50
5.3	Applicazioni pratiche e guida utente . . . . .	51
<b>6</b>	<b>Analisi e Riflessioni su "Dom the Robot"</b>	<b>57</b>
6.1	SWOT Analysis di "Dom the Robot" . . . . .	57
6.1.1	Forze (Strengths) . . . . .	58
6.1.2	Debolezze (Weaknesses) . . . . .	58

---

6.1.3	Opportunità (Opportunities) . . . . .	59
6.1.4	Minacce (Threats) . . . . .	59
6.2	Lezioni Apprese . . . . .	60
6.2.1	Adattabilità e Flessibilità . . . . .	60
6.2.2	L'importanza del Feedback degli Utenti . . . . .	60
6.2.3	Integrazione delle Tecnologie Emergenti . . . . .	60
	<b>Conclusioni e uno sguardo al futuro</b>	<b>61</b>
	<b>Bibliografia</b>	<b>63</b>
	<b>Ringraziamenti</b>	<b>67</b>

---

## Elenco delle figure

---

1.1	Il modello motivazione/prestazione di J. W. Atkinson . . . . .	7
1.2	Sequenza di screenshot di una simulazione di un intervento su Touch Sourgery	9
1.3	Screenshot del Serious Game Foldit che mostra composizione e caratteristiche di una proteina . . . . .	9
1.4	Un esempio di problema e il suo processo di risoluzione all'interno di Dragon-Box: (a) Moltiplica la DragonBox per 2; (b) Moltiplica tutti gli altri termini su entrambi i lati per 2; (c) Scomponi 4 in $2*2$ ; (d) Combina i 2 nel numeratore e nel denominatore; (e) Sottrai $1/2$ da entrambi i lati; (f) La DragonBox è isolata e l'utente ha ricevuto 3 stelle per aver risolto il problema . . . . .	11
2.1	Esempio di utilizzo dell'app IKEA Place . . . . .	13
2.2	Google Glass . . . . .	15
2.3	Esempio di realtà mista di Vision Pro . . . . .	19
3.1	Illustrazione di un cervello umano e una rete neurale che collaborano iniseme	23
3.2	Illustrazione di un chatbot che tramite l'NLU consiglia strategie operative . .	25
3.3	Illustrazione delle 6 potenze mondiali che competono nella guerra dell'AI . .	26
4.1	Schema di funzionamento del modello all-MiniLM-L6-v2 . . . . .	32
5.1	Paesaggio di sfondo . . . . .	33
5.2	Navigation Mesh della scena di gioco . . . . .	35
5.3	GameObjects in scena . . . . .	35
5.4	Rig umanoide e Dom in posizione di attesa . . . . .	36
5.5	Caso d'uso: salutare . . . . .	52
5.6	Caso d'uso: ballare . . . . .	52
5.7	Caso d'uso: perplessità . . . . .	53
5.8	Caso d'uso: recarsi presso un cubo . . . . .	53
5.9	Caso d'uso: comporre un colore . . . . .	54

---

5.10	Caso d'uso: correggere utente che sbaglia . . . . .	55
5.11	Caso d'uso: complimentarsi con l'utente che esegue la composizione correttamente . . . . .	56
6.1	Matrice SWOT . . . . .	58



5.1	Macchina degli stati . . . . .	36
5.2	Funzione <i>OnOrderGiven</i> . . . . .	37
5.3	Funzione <i>OnTestOrderGiven</i> . . . . .	37
5.4	Funzione <i>Utility</i> . . . . .	38
5.5	Funzione <i>TestUtility</i> . . . . .	39
5.6	Funzione <i>GrabCreate</i> . . . . .	40
5.7	Funzione <i>Drop</i> . . . . .	41
5.8	Funzione <i>DropSecondary</i> . . . . .	41
5.9	Coroutine <i>DestroyClones</i> . . . . .	41
5.10	Coroutine <i>Attend</i> . . . . .	42
5.11	Coroutine <i>ResetScene</i> . . . . .	42
5.12	Stati <i>ComposeCube1-4</i> . . . . .	43
5.13	Stato <i>ComposeCube5</i> . . . . .	44
5.14	Stati <i>Test</i> e <i>Test1-4</i> . . . . .	45
5.15	Stato <i>Test5</i> . . . . .	46
5.16	Stato <i>MoveTo</i> . . . . .	47
5.17	Stato <i>Happy</i> . . . . .	48
5.18	Stato <i>Puzzled</i> . . . . .	48
5.19	Stati <i>Idle</i> e <i>Idle_Moment</i> . . . . .	49
5.20	Stato <i>Reset</i> . . . . .	49
5.21	Variabili della classe <i>RagionamentoDiDom</i> . . . . .	50
5.22	Funzione <i>RankSimilarityScores</i> . . . . .	51

La decisione di approfondire l'integrazione tra realtà virtuale (VR) e comprensione del linguaggio naturale (NLU) nel contesto di un Serious Game emerge da un'analisi accurata delle tendenze attuali e future nel settore dell'educazione e della tecnologia. L'importanza di queste tecnologie per l'innovazione pedagogica è innegabile, considerando il loro potenziale di rivoluzionare il modo in cui l'istruzione viene concepita ed erogata.

La realtà virtuale, con la sua capacità di simulare ambienti complessi e coinvolgenti, offre una nuova dimensione all'apprendimento esperienziale. Gli studenti non sono più limitati dalle pareti della classe; possono esplorare mondi antichi, interagire con fenomeni scientifici o immergersi in scenari linguistici realistici, il tutto in un contesto sicuro e controllato. Questa immersione totale favorisce un apprendimento profondo, stimolando la curiosità e l'engagement, elementi chiave per una formazione efficace. La VR, dunque, si presenta come uno strumento potentissimo per superare i limiti fisici e temporali dell'educazione tradizionale, aprendo nuove strade per la personalizzazione dell'insegnamento e l'accessibilità.

Parallelamente, gli sviluppi nel campo della comprensione del linguaggio naturale stanno trasformando le interazioni tra uomini e macchine. L'NLU permette ai computer di interpretare, comprendere e rispondere al linguaggio umano in modo naturale e intuitivo. Integrare l'NLU in contesti educativi significa poter offrire esperienze di apprendimento personalizzate e interattive, dove il feedback e l'assistenza sono forniti in tempo reale, adattandosi al livello e alle necessità di ogni studente. Questo non solo migliora l'efficacia dell'apprendimento ma rende l'educazione più inclusiva, abbattendo barriere linguistiche e cognitive.

Il connubio tra VR e l'NLU rappresenta, quindi, un campo di studio promettente per l'innovazione educativa. Attraverso l'analisi e lo sviluppo di un Serious Game che utilizza queste tecnologie, la presente ricerca mira a dimostrare come la VR e l'NLU possano non solo migliorare l'esperienza di apprendimento, ma anche contribuire a formare individui più preparati ad affrontare le sfide del futuro.

La scelta di concentrare la tesi su questi temi nasce dall'urgenza di rispondere alle esigenze di un mondo educativo in rapida evoluzione, dove la tecnologia gioca un ruolo sempre più

centrale. L'ambizione è quella di fornire un contributo significativo al campo dell'educazione, proponendo un modello innovativo che integri VR e NLU per creare esperienze di apprendimento immersive, personalizzate e profondamente efficaci. Questo lavoro, pertanto, non solo esplora le frontiere della tecnologia educativa, ma si propone anche come stimolo per ulteriori ricerche e sviluppi in questo ambito, con l'obiettivo ultimo di arricchire il processo educativo e renderlo più aderente alle necessità di studenti e insegnanti nell'era digitale.

Il processo di ricerca e sviluppo intrapreso per la stesura di questa tesi rappresenta un percorso ampio e articolato, che richiede un approccio multidisciplinare per affrontare le sfide poste dall'integrazione tra realtà virtuale (VR) e comprensione del linguaggio naturale (NLU) in ambito educativo. La prima fase vede un'immersione nella letteratura esistente, con l'obiettivo di costruire una solida base teorica che spaziassi dalle fondamenta tecnologiche e scientifiche della VR e dell'NLU, fino alle più recenti innovazioni nel campo dei Serious Game e delle applicazioni educative immersive. Questo studio preliminare permette di identificare le potenzialità e le sfide connesse all'uso di queste tecnologie nell'educazione, nonché di definire gli obiettivi specifici che il progetto "Dom the Robot" si prefigge di raggiungere.

Successivamente, si avvia la fase di progettazione e sviluppo del gioco, che comporta diverse sotto-fasi: dalla concettualizzazione dell'idea, alla definizione della struttura e degli obiettivi educativi, fino alla realizzazione tecnica vera e propria. Questo processo richiede l'integrazione di competenze in vari campi, inclusa la programmazione, il design dell'interfaccia utente, la grafica 3D, e la psicologia educativa. Particolare attenzione è dedicata allo sviluppo di meccaniche di gioco che possono sfruttare al meglio le capacità e potenzialità dell'NLU per creare un ambiente di apprendimento coinvolgente e interattivo. La creazione di scenari virtuali, la modellazione di interazioni naturali con il linguaggio e l'implementazione di algoritmi di Intelligenza Artificiale rappresentano momenti chiave di questo percorso.

L'ultima fase del lavoro include il testing del prodotto con gruppi di utenti reali, al fine di valutare l'efficacia educativa del gioco e raccogliere feedback per ulteriori miglioramenti. L'analisi dei dati raccolti permette di riflettere criticamente sui risultati ottenuti, evidenziando successi e aree di miglioramento.

Il lavoro svolto per la stesura di questa tesi rappresenta un'impresa stimolante, che cerca di spingere i confini dell'innovazione educativa attraverso l'uso di tecnologie avanzate. Si vuole dimostrare come l'integrazione delle nuove tecnologie possa offrire nuove strade per l'apprendimento, creando esperienze educative immersive e personalizzate che rispondano alle esigenze di un pubblico eterogeneo. Attraverso un approccio rigoroso e creativo, questo progetto cerca di contribuire al dibattito sull'evoluzione dei metodi didattici e sull'importanza dell'innovazione tecnologica nel campo dell'educazione.

La presente tesi si articola in sei capitoli, ognuno dei quali dettaglia un aspetto specifico come di seguito specificato:

- Il Capitolo 1 esamina i Serious Game, il loro sviluppo, l'impatto e le applicazioni nel contesto educativo, sottolineando efficacia e metodologie di apprendimento.

- Il Capitolo 2 esamina le tecnologie immersive, come Realtà Aumentata, Virtuale e Mista, le loro definizioni, la loro evoluzione, le applicazioni e l'impatto sull'educazione.
- Il Capitolo 3 discute il Natural Language Understanding, il ruolo delle reti neurali, del Deep Learning e l'innovazione in questo ambito.
- Il Capitolo 4 descrive l'ideazione, i requisiti e la progettazione di "Dom the Robot", un Serious Game sviluppato ex novo.
- Il Capitolo 5 approfondisce la realizzazione del progetto, analizzandone la grafica, l'interattività e soffermandosi sull'integrazione dell'Intelligenza Artificiale nel gioco.
- Infine, il Capitolo 6 analizza il gioco tramite l'analisi SWOT, riflettendo sulle lezioni apprese e sull'integrazione delle tecnologie emergenti.

---

## Serious Game: Innovazione, Apprendimento e Applicazioni nel Contesto Educativo Moderno

---

*Nel panorama dell'educazione e della formazione contemporanea, si assiste a una progressiva e significativa evoluzione delle metodologie didattiche, grazie all'integrazione di tecnologie avanzate e approcci innovativi. Tra questi, i Serious Game emergono come strumenti rivoluzionari, capaci di ridefinire il processo di apprendimento e di offrire nuove prospettive per l'acquisizione di competenze e conoscenze. Questo capitolo intende esplorare in profondità il fenomeno dei Serious Game, analizzando la loro origine, evoluzione, e il ruolo cruciale che rivestono nel contesto educativo moderno. Attraverso un approccio multidisciplinare, che unisce pedagogia, psicologia e tecnologia, si discuterà di come questi giochi digitali si distinguano dai tradizionali videogiochi per finalità e obiettivi, orientandosi non verso il semplice intrattenimento, ma verso la trasmissione di conoscenze e lo sviluppo di competenze specifiche.*

### 1.1 Cosa sono i Serious Game?

I Serious Game rappresentano una categoria innovativa di giochi digitali che si distanziano dai tradizionali videogiochi per scopi e finalità. Mentre i classici videogiochi sono principalmente orientati all'intrattenimento e al divertimento, i Serious Game sono progettati con l'obiettivo specifico di trasmettere conoscenze, competenze o sensibilizzare su determinate tematiche. Questa distinzione è fondamentale per comprendere la natura e l'efficacia dei Serious Game come strumenti educativi e formativi. La definizione di Serious Game è stata oggetto di discussione e analisi nel corso degli anni. Michael Zyda, uno dei pionieri in questo campo, li ha definiti come "giochi che non hanno l'intrattenimento, l'informazione o la persuasione come loro principale intento". Questa definizione sottolinea l'intenzione di andare oltre il semplice divertimento, puntando a obiettivi più significativi, come l'istruzione, la formazione professionale, la salute, la politica pubblica e la sensibilizzazione sociale.

I Serious Game si trovano all'intersezione di tre elementi fondamentali: *simulazione, giochi ed educazione*. Questa intersezione rappresenta un punto di convergenza dove le tecniche

di simulazione si incontrano con le dinamiche dei giochi per creare esperienze educative immersive e coinvolgenti.

Nella *simulazione*, gli utenti sono immersi in ambienti che replicano aspetti del mondo reale o situazioni ipotetiche, consentendo loro di sperimentare e interagire con scenari complessi in un contesto sicuro e controllato. Questo aspetto è cruciale nei Serious Game, poiché permette di ricreare situazioni realistiche in cui gli utenti possono applicare e testare le loro competenze e conoscenze.

I *giochi*, con la loro natura intrinsecamente motivante e coinvolgente, forniscono il contesto ludico che rende l'apprendimento più piacevole e stimolante. L'elemento del gioco, caratterizzato da regole, obiettivi, feedback immediato e un senso di sfida, rende i Serious Game particolarmente efficaci nell'incrementare l'engagement degli utenti. Questo coinvolgimento attivo, unito alla natura interattiva dei giochi, facilita l'apprendimento esperienziale e il consolidamento delle conoscenze.

L'*educazione* è il terzo pilastro fondamentale che completa il quadro dei Serious Game. Al contrario dei giochi pensati puramente per intrattenimento, i Serious Game sono progettati con un intento educativo ben definito. Che si tratti di sviluppare competenze specifiche, di promuovere la consapevolezza su questioni sociali, o di facilitare l'apprendimento in ambiti accademici, ogni aspetto del Serious Game è calibrato per massimizzare l'efficacia educativa.

Spesso non si comprende la differenza tra Gamification e Serious Game; è importante sottolineare che, mentre entrambi utilizzano elementi ludici, si differenziano per obiettivi e applicazioni. La Gamification si riferisce all'applicazione di meccaniche di gioco in contesti non ludici per motivare e coinvolgere gli utenti. Esempi comuni includono l'uso di punti, badge, classifiche e sfide in app educative, programmi di fidelizzazione dei clienti o piattaforme di fitness. La Gamification è, quindi, un approccio più ampio che può essere applicato a una varietà di contesti per aumentare l'engagement e la motivazione. I Serious Game, invece, sono giochi a tutti gli effetti, progettati specificamente per scopi educativi o di formazione. Mentre la gamification introduce elementi di gioco in attività non ludiche, i Serious Game sono intrinsecamente giochi che incorporano obiettivi educativi nel loro design. In altre parole, mentre la gamification "gamifica" un'attività esistente, i Serious Game sono creati da zero con l'intento di insegnare o formare attraverso l'esperienza di gioco.

## 1.2 Evoluzione e significato nel contesto moderno

### 1.2.1 Dalle origini ad Oggi: La Trasformazione dei Serious Game

Nel 799 d.C., abbiamo il primo esempio di gioco a scopo formativo, inventato da Alcuino da York, che creò giochi logico-matematici per la scuola Palatina. Nel suo libro "*Propositiones ad acuendos juvenes*", Alcuino presentava enigmi e indovinelli, per insegnare ai giovani i principi elementari dell'algebra e della geometria.

L'evoluzione dei Serious Game risale agli anni '60 e '70, quando erano principalmente usati in ambito militare e governativo. Inizialmente, si focalizzavano su simulatori di volo e

giochi di guerra per l'addestramento delle forze armate. Con l'avvento di tecnologie avanzate, come i personal computer, i Serious Game hanno iniziato a guadagnare popolarità in altri settori. Negli anni '80 e '90, hanno visto un'espansione nell'educazione e nella formazione aziendale, con giochi progettati per insegnare specifiche competenze o concetti. Nel 21° secolo, l'avanzamento tecnologico ha permesso di sviluppare Serious Game più sofisticati, sfruttando la realtà virtuale, la realtà aumentata e l'Intelligenza Artificiale per creare esperienze educative altamente immersive.

### **1.2.2 Rilevanza Attuale: I Serious Game nel Mondo Moderno**

Oggi i Serious Game sono diventati strumenti importanti in diversi settori. Nel campo dell'educazione, vengono utilizzati per migliorare l'engagement e l'apprendimento degli studenti, offrendo un approccio alternativo alla didattica tradizionale. Nel settore sanitario, i Serious Game aiutano nella formazione di medici e infermieri, oltre a essere usati per terapie riabilitative e promozione di stili di vita sani. Nel settore aziendale, sono impiegati per la formazione dei dipendenti, sviluppando competenze di leadership, gestione del team e risoluzione dei problemi. La loro capacità di simulare scenari complessi e fornire feedback immediati li rende strumenti preziosi per l'apprendimento e la formazione.

### **1.2.3 Dinamiche di Cambiamento: Innovazioni e Impatti**

Le innovazioni tecnologiche, in particolare la realtà virtuale (VR) e l'Intelligenza Artificiale (AI), che analizzeremo nei capitoli successivi, hanno rivoluzionato il campo dei Serious Game, migliorandone notevolmente efficacia e applicabilità. La realtà virtuale, con la sua capacità di creare ambienti completamente immersivi, consente agli utenti di immergersi in scenari simulati che replicano situazioni reali in modo sicuro e controllato. Questo aspetto è particolarmente utile in contesti educativi e formativi dove la pratica in situazioni realistiche è fondamentale. Per esempio, i Serious Game basati su VR vengono utilizzati in ambito medico per simulare interventi chirurgici, permettendo ai professionisti di esercitarsi senza rischi per i pazienti.

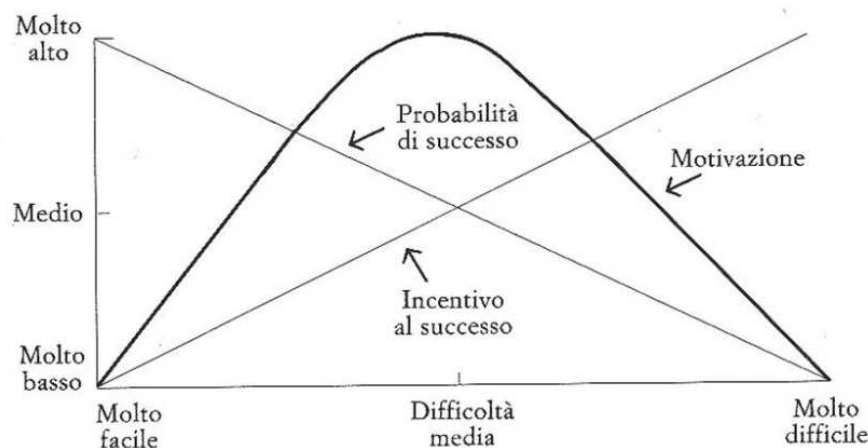
L'Intelligenza Artificiale, d'altra parte, apporta una personalizzazione senza precedenti all'esperienza di apprendimento nei Serious Game. Attraverso l'AI, i giochi possono adattarsi dinamicamente alle esigenze individuali degli utenti, modificando la difficoltà, fornendo suggerimenti personalizzati o adattando i contenuti in base alle prestazioni e al progresso dell'utente. Questo approccio individualizzato non solo rende l'apprendimento più efficace ma aumenta anche il coinvolgimento e la motivazione degli utenti. Inoltre, l'utilizzo dell'AI nei Serious Game permette la raccolta e l'analisi di grandi quantità di dati relativi al comportamento e alle prestazioni degli utenti. Questi dati possono essere utilizzati per valutare l'efficacia dell'apprendimento e per apportare miglioramenti continui ai giochi, rendendoli sempre più efficaci come strumenti educativi. La combinazione di questi avanzamenti tecnologici non solo migliora l'esperienza di apprendimento ma apre anche nuove possibilità per l'utilizzo dei Serious Game in una varietà di contesti educativi e formativi.

## 1.3 Apprendimento Tramite i Serious Game: Metodologie ed Efficacia

### 1.3.1 Principi Pedagogici e Strategie Didattiche nei Serious Game

Anche se non esiste un modello unificato che descriva i Serious Game, è possibile identificare alcuni processi neuro-psicologici chiave che entrano in gioco durante il loro utilizzo:

1. *Sistema Attenitivo*: l'attenzione è una componente cruciale della funzione cognitiva. La sua presenza o assenza può influenzare significativamente l'intero sistema cognitivo. I Serious Game devono, quindi, impegnarsi a mantenere l'interesse dei giocatori, sia attraverso sfide avvincenti sia con una grafica coinvolgente, per garantire un maggior coinvolgimento attentivo.
2. *Modello Motivazione/Prestazione*: questo modello (Figura 1.1), proposto dallo psicologo John William Atkinson, è rappresentato da una curva a U rovesciata e suggerisce che le prestazioni ottimali si ottengono con un livello equilibrato di motivazione. I Serious Game tendono a centrare questa finestra ottimale di motivazione, in quanto le prestazioni degli utenti sono supportate dal software, riducendo la paura di prestazioni scarse e prevenendo l'eccesso di motivazione.



**Figura 1.1:** Il modello motivazione/prestazione di J. W. Atkinson

3. *Circuito del Reward*: fondamentale in qualsiasi gioco, questo circuito serve a stimolare sia il sistema attentivo sia quello motivazionale. Nei Serious Game, il reward funge anche da meccanismo di feedback, essenziale per bilanciare il gioco in modo da evitare effetti negativi, come la dipendenza o la frustrazione.
4. *Sistemi di Apprendimento*: l'obiettivo principale dei Serious Game è quello di stimolare l'apprendimento e migliorare le prestazioni. Un gioco ben progettato permetterà ai giocatori di apprendere dinamiche e strategie man mano che avanzano, migliorando non solo le prestazioni nel gioco ma anche le abilità applicabili nel mondo reale.



### 1.3.2 Valutazione dell'Efficacia: Risultati e Impatti Educativi

Non esiste una teoria unificata che spieghi completamente la pedagogia dei Serious Game, poiché ciascuno di essi presenta dinamiche specifiche. Tuttavia, è noto che i Serious Game derivano i loro modelli funzionali dai principi pedagogici. La mancanza di un modello unificato è parzialmente dovuta alla varietà delle dinamiche di gioco. Nel testo "*Current Practices in Serious Game Research: A Review from a Learning Outcomes Perspective*", gli autori osservano che, esaminando 28 studi sui Serious Game, questi risultano più efficaci di altri metodi di apprendimento solo quando le caratteristiche di gioco sono perfettamente bilanciate. Variabili come la modulazione della difficoltà sono cruciali e possono influenzare significativamente l'efficacia di un Serious Game.

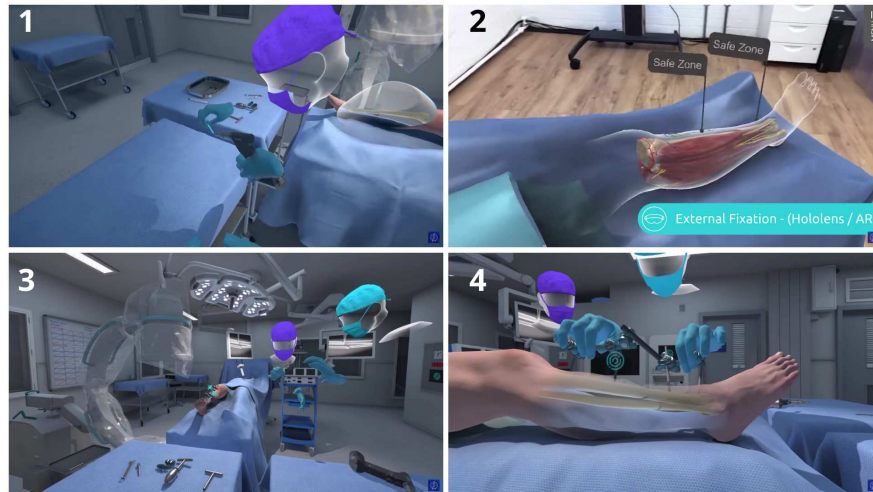
## 1.4 Applicazioni Pratiche

### 1.4.1 Esempi specifici in diversi settori

#### Esempio 1: Touch Surgery - Settore medico

Il "serious game" Touch Surgery rappresenta un'avanguardia nella formazione medico-chirurgica, unendo l'ambito ludico con la serietà dell'apprendimento professionale (Figura 1.2). Questo gioco serio è stato sviluppato per fornire ai chirurghi, sia esperti che in formazione, un'esperienza immersiva e interattiva di simulazione chirurgica. L'obiettivo principale di Touch Surgery è di migliorare le competenze tecniche e decisionali dei professionisti sanitari attraverso una piattaforma digitale accessibile e intuitiva. Touch Surgery utilizza grafica 3D avanzata per simulare un'ampia gamma di procedure chirurgiche. Gli utenti navigano attraverso vari step di un intervento, che comprendono la diagnosi, la pianificazione, l'intervento e la revisione post-operatoria. Ogni fase è progettata per essere il più realistica possibile, offrendo un feedback visivo e tattile che simula l'esperienza reale in sala operatoria. Un aspetto fondamentale di Touch Surgery è il suo sistema di valutazione basato sull'Intelligenza Artificiale. Durante la simulazione, l'IA analizza le azioni degli utenti, fornendo valutazioni immediate e personalizzate sulle performance. Questo feedback è cruciale per l'apprendimento, in quanto permette agli utenti di identificare aree di miglioramento e di monitorare i propri progressi nel tempo. Inoltre, la piattaforma offre un'ampia libreria di casi clinici e procedure, costantemente aggiornata, che consente agli utenti di esplorare e praticare un'ampia varietà di interventi chirurgici.

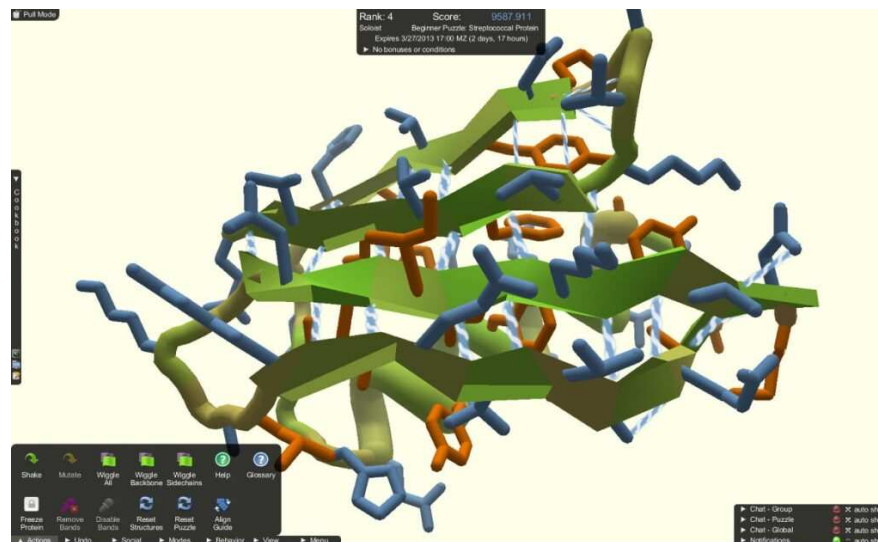
L'integrazione di tecnologie innovative, come la realtà virtuale e l'Intelligenza Artificiale rende Touch Surgery uno strumento estremamente efficace per l'addestramento chirurgico. Questo serious game non solo migliora le competenze tecniche dei chirurghi, ma contribuisce anche alla sicurezza dei pazienti, riducendo il rischio di errori in sala operatoria. La sua capacità di fornire un'esperienza di apprendimento personalizzata e scalabile lo rende una risorsa preziosa per la formazione medica contemporanea.



**Figura 1.2:** Sequenza di screenshot di una simulazione di un intervento su Touch Surgery

### Esempio 2: Foldit - Settore Scientifico

Foldit è un Serious Game che sfida i giocatori a piegare le proteine in configurazioni ottimali, un problema complesso nella biologia molecolare (Figura 1.3). Il gioco si basa sulla collaborazione e sulla competizione, incoraggiando i giocatori a scoprire configurazioni proteiche che potrebbero essere utili nella ricerca scientifica. Foldit è stato particolarmente efficace nell'identificare nuove strutture proteiche, con i giocatori che hanno contribuito a scoperte significative, come nuove configurazioni per le proteine dell'HIV, che hanno eluso i ricercatori per anni.



**Figura 1.3:** Screenshot del Serious Game Foldit che mostra composizione e caratteristiche di una proteina

### 1.4.2 Confronto con strumenti educativi tradizionali

DragonBox è un esempio eccellente di come i Serious Game possano trasformare l'educazione in un'esperienza coinvolgente e interattiva, specialmente in un campo tradizionalmente impegnativo come la matematica. Le dinamiche del gioco sono progettate per introdurre concetti matematici in modo graduale e intuitivo, facilitando l'apprendimento senza che i bambini si rendano conto di affrontare argomenti complessi. Il gioco inizia con livelli che utilizzano simboli e personaggi, invece di numeri e variabili, rendendo l'esperienza visivamente attraente e meno intimidatoria. Questi simboli sono usati per rappresentare concetti matematici di base, come l'uguaglianza e il bilanciamento delle equazioni. Man mano che i giocatori avanzano, il gioco introduce numeri e variabili, sostituendo i simboli iniziali. Questo approccio graduale consente ai bambini di costruire una comprensione solida dei concetti matematici senza sentirsi sopraffatti. Le sfide presentate in DragonBox sono progettate per essere risolte attraverso il ragionamento logico e la sperimentazione. In questo gioco, gli studenti seguono le stesse regole che si applicano nell'algebra reale per manipolare i personaggi digitali e isolare una scatola contenente un drago, analoga alla risoluzione di un'equazione per  $x$  (Figura 1.4). Questo processo di manipolazione e risoluzione degli enigmi rinforza la comprensione dei principi dell'algebra in un contesto pratico e giocoso. DragonBox incoraggia l'apprendimento attraverso il feedback immediato e la gratificazione. Quando un giocatore risolve correttamente un puzzle, riceve un feedback positivo, che non solo rafforza il concetto appreso ma aumenta anche la fiducia e la motivazione. Se un giocatore commette un errore, il gioco fornisce suggerimenti e incoraggiamenti per guidare il bambino verso la soluzione corretta, piuttosto che punirlo.

Nel contesto della pandemia COVID-19 e dell'adozione forzata dell'apprendimento a distanza, lo studio condotto dal Worcester Polytechnic Institute ha esaminato l'efficacia di DragonBox Algebra 12+ nel mitigare la perdita di apprendimento tra gli studenti. Attraverso l'analisi dei risultati di 253 studenti di settima classe che hanno giocato a DragonBox Algebra 12+ per nove sessioni, lo studio ha rivelato che un maggiore coinvolgimento nel gioco è correlato a prestazioni matematiche superiori, soprattutto per gli studenti con conoscenze algebriche pregresse inferiori.



**Figura 1.4:** Un esempio di problema e il suo processo di risoluzione all'interno di DragonBox: (a) Moltiplica la DragonBox per 2; (b) Moltiplica tutti gli altri termini su entrambi i lati per 2; (c) Scomponi 4 in  $2 \cdot 2$ ; (d) Combina i 2 nel numeratore e nel denominatore; (e) Sottrai  $1/2$  da entrambi i lati; (f) La DragonBox è isolata e l'utente ha ricevuto 3 stelle per aver risolto il problema

---

## Esplorazione delle Tecnologie Immersive: Realtà Aumentata, Virtuale e Mista

---

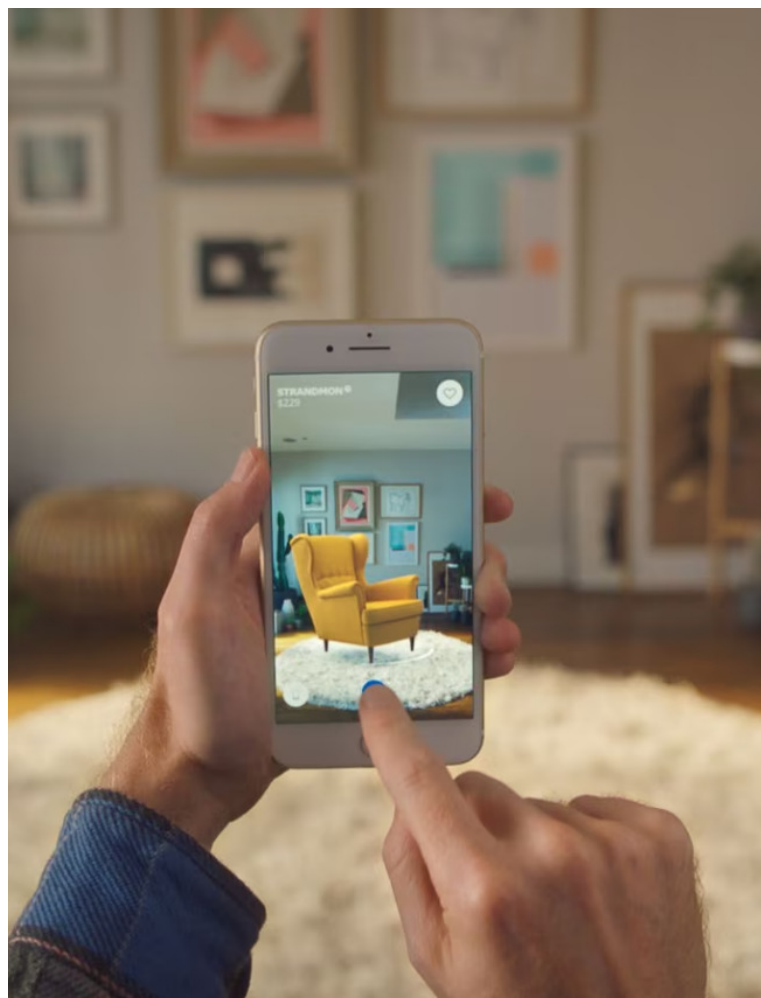
*Questo capitolo offre un'analisi approfondita delle tecnologie all'avanguardia che stanno modellando il nostro modo di interagire con il mondo ovvero Realtà Aumentata, Virtuale e Mista. In particolare, esso esplora l'evoluzione, le applicazioni e l'impatto di queste tecnologie nel mondo reale, analizza dispositivi, software e scenari futuri, evidenziando l'integrazione crescente di elementi digitali nell'ambiente fisico, con implicazioni in vari settori.*

### 2.1 Fondamenti della Realtà Aumentata

#### 2.1.1 Definizione e Storia della Realtà Aumentata

La Realtà Aumentata (AR), nota anche come realtà mediata dall'elaboratore, rappresenta un arricchimento della percezione sensoriale umana tramite informazioni elettronicamente manipolate e convogliate, che altrimenti non sarebbero percepite con i cinque sensi. Gli esempi di AR includono il cruscotto dell'automobile, l'esplorazione della città tramite smartphone e la chirurgia robotica a distanza. La storia della Realtà Aumentata inizia nel 1968 con il lavoro di Ivan Sutherland, che introdusse gli occhiali a realtà aumentata. Negli anni '90, la visione dell'AR come mondo virtuale arricchito prende forma con l'avvento dell'elettronica miniaturizzata, di dispositivi portatili, di Internet e della geolocalizzazione. I primi prodotti d'uso comune emergono alla fine del decennio. In termini di definizione, gli elementi che "aumentano" la realtà possono essere aggiunti attraverso dispositivi mobili, come smartphone, PC con webcam o altri sensori, e dispositivi di visione, ascolto e manipolazione. Queste informazioni aggiuntive possono anche consistere in una diminuzione delle informazioni normalmente percepibili, sempre al fine di presentare una situazione più chiara, utile o divertente. Esistono diversi tipi di realtà aumentata; essi sono:

- *Realtà Aumentata su Dispositivo Mobile*: questa forma di AR utilizza smartphone e tablet per sovrapporre informazioni digitali al mondo reale. Gli utenti possono puntare i loro dispositivi verso oggetti specifici o luoghi per visualizzare informazioni aggiuntive, immagini o animazioni in tempo reale. Un esempio di Realtà Aumentata su Dispositivo Mobile è l'applicazione "IKEA Place" (Figura 2.1). Questa app permette agli utenti di visualizzare come i mobili e gli accessori IKEA apparirebbero nelle loro case prima di acquistarli. Gli utenti possono selezionare prodotti dal catalogo IKEA e, usando la fotocamera del proprio smartphone o tablet, posizionarli virtualmente in qualsiasi parte della loro abitazione. Questo tipo di applicazione consente ai consumatori di vedere in anteprima e sperimentare i prodotti in scala reale e in contesti reali, facilitando la scelta e l'acquisto.



**Figura 2.1:** Esempio di utilizzo dell'app IKEA Place

- *Realtà Aumentata su Computer*: questa tipologia si avvale di un computer con una webcam o altri sensori per rilevare e interpretare l'ambiente circostante. Gli utenti possono interagire con elementi virtuali inseriti nell'ambiente reale tramite schermo. Un esempio può essere un'applicazione che consente di provare virtualmente abiti o

accessori. Questo tipo di AR trova applicazioni in ambito educativo, come in programmi di formazione o simulazioni, offrendo un'esperienza immersiva senza necessità di hardware specifico oltre al computer.

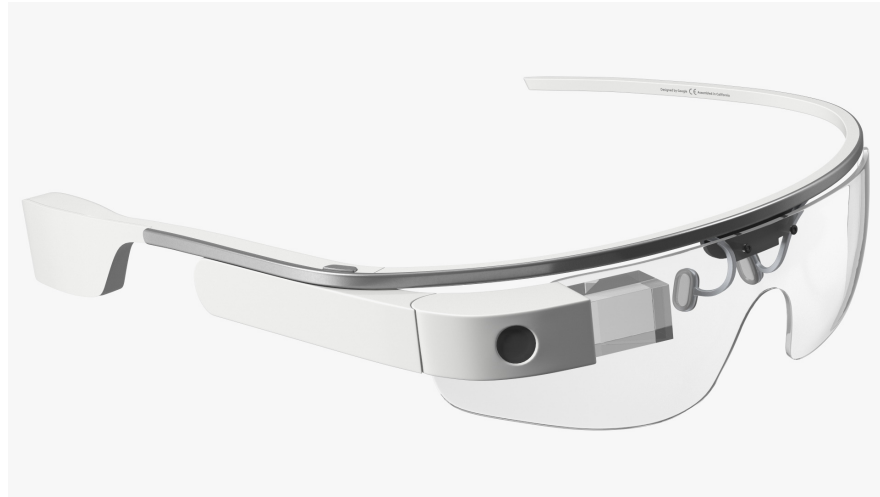
- *Realtà Aumentata Senza Marker*: a differenza delle tipologie precedenti, che spesso richiedono l'uso di codici QR o altri marker per ancorare gli elementi virtuali al mondo reale, la AR senza marker utilizza tecnologie avanzate di riconoscimento dell'ambiente, come il riconoscimento di forme, oggetti o il tracciamento spaziale. Ciò permette agli utenti di interagire con gli elementi aumentati in modo più fluido e naturale. Ad esempio, in una mostra d'arte aumentata, i visitatori potrebbero vedere animazioni o ascoltare storie correlate alle opere d'arte semplicemente avvicinandosi ad esse, senza bisogno di scansionare alcun codice. Questa tecnologia offre un'esperienza più immersiva e intuitiva, ma richiede algoritmi e hardware più sofisticati.

### 2.1.2 Tecnologie e Dispositivi per la Realtà Aumentata

La realtà aumentata è una tecnologia che sovrappone informazioni digitali al mondo fisico, creando un'esperienza immersiva che integra elementi virtuali con l'ambiente circostante. Il concetto di AR risale agli anni '60, con l'invenzione del primo dispositivo di realtà aumentata, chiamato "*Sword of Damocles*", creato da Ivan Sutherland nel 1968. Questo dispositivo, benché primitivo, è stato un pioniere nel campo, fornendo un display grafico che si sovrapponeva alla visione reale. Negli anni '90, la realtà aumentata ha iniziato a guadagnare terreno, con l'emergere di ricerche e sviluppi significativi. Uno dei primi sistemi AR commerciali è stato "KARMA" (*Knowledge-based Augmented Reality for Maintenance Assistance*), sviluppato dall'Università della Georgia, che aiutava gli utenti nella manutenzione di fotocopiatrici utilizzando sovrapposizioni grafiche. Questi primi sistemi, tuttavia, erano spesso ingombranti e limitati nella loro applicabilità. L'arrivo degli smartphone e dei tablet ha aperto nuove frontiere per la realtà aumentata, rendendola più accessibile e diffusa.

Nel 2009, l'applicazione AR "Layar" è stata lanciata su dispositivi mobili, permettendo agli utenti di visualizzare informazioni digitali sovrapposte al mondo reale attraverso la fotocamera del proprio telefono. Questa era l'era in cui la AR diventava un fenomeno di massa, con applicazioni che spaziavano dal gaming alla navigazione, passando per l'istruzione e il marketing. Uno degli sviluppi più significativi negli ultimi anni è stato l'introduzione degli occhiali AR. I Google Glass (Figura 2.2), lanciati nel 2013, sebbene abbiano avuto un'accoglienza mista, hanno segnato un importante passo avanti nella realizzazione di dispositivi AR indossabili e discreti. Successivamente, aziende come Microsoft, con HoloLens, e Magic Leap, con il dispositivo omonimo hanno continuato a spingere i confini della tecnologia AR, offrendo esperienze sempre più immersive e interattive. Oggi, la realtà aumentata è integrata in una vasta gamma di applicazioni, dai giochi ai tool educativi, dalle applicazioni mediche alle soluzioni per il commercio al dettaglio. La tecnologia continua a evolversi, con miglioramenti nella qualità dell'immagine, nell'interazione con l'utente e nella portabilità dei dispositivi. La realtà aumentata sta trasformando il modo in cui interagiamo con il mondo





**Figura 2.2:** Google Glass

che ci circonda, portando un layer digitale alla nostra esperienza quotidiana e promettendo ulteriori innovazioni nei prossimi anni.

### 2.1.3 Applicazioni Pratiche della Realtà Aumentata

L'AR ha trovato applicazioni pratiche in numerosi settori, inclusi l'istruzione, la medicina, il marketing e l'ingegneria. Nell'educazione, ad esempio, l'AR viene utilizzata per creare esperienze di apprendimento immersive e interattive, migliorando la comprensione e l'interesse degli studenti. In campo medico, l'AR aiuta i professionisti nella visualizzazione di dati complessi e nella realizzazione di procedure chirurgiche. Nel marketing, l'AR offre esperienze di acquisto innovative e personalizzate, migliorando l'engagement del cliente. In ingegneria, l'AR supporta la progettazione e la manutenzione, fornendo visualizzazioni dettagliate e istruzioni interattive. Ecco alcuni esempi di applicazioni di realtà aumentata e come funzionano:

- App AR per l'Assistenza all'Acquisto di Alimenti: "Food Store Assistant" è una applicazione AR progettata specificamente per assistere gli anziani durante gli acquisti alimentari. Utilizzando la tecnologia AR, questa app fornisce informazioni dettagliate sugli alimenti per migliorare l'esperienza di acquisto. Sviluppata con Unity 3D, "Food Store Assistant" si collega a un server web per elaborare le richieste ricevute dall'applicazione mobile, rendendo la spesa più accessibile e informativa per gli anziani.
- App AR per la Promozione del Turismo: "Banyumas Explorer" è un'applicazione AR creata per valorizzare il turismo nella Regione di Banyumas. Sfruttando un sistema di tracking basato su marker, l'app consente ai turisti di scoprire attrazioni e punti di interesse, con oggetti virtuali che vengono visualizzati attraverso la fotocamera del dispositivo mobile per arricchire l'esperienza turistica.
- AR per la Manutenzione dell'Equipaggiamento Elettromeccanico: "AR Maintenance Assistant" è un'applicazione AR utilizzata per la manutenzione di equipaggiamenti elettromecca-



nici. L'app fornisce istruzioni dettagliate e supporto durante i processi di manutenzione, migliorando l'efficienza e la sicurezza del lavoro.

Questa soluzione AR aiuta i tecnici a visualizzare informazioni critiche in tempo reale, semplificando compiti complessi. Questi esempi dimostrano la versatilità e l'ampio spettro di applicazioni della tecnologia AR in vari settori, dalla salute all'istruzione, al turismo e alla manutenzione industriale.

## 2.2 Realtà Virtuale: Immersione e Interazione

### 2.2.1 Concetti Base della Realtà Virtuale

La realtà virtuale è un ambiente tridimensionale generato da computer, che crea un mondo immersivo in cui gli utenti possono interagire. Questa tecnologia avanzata utilizza dispositivi come visori VR, guanti sensoriali e sistemi audio, consentendo alle persone di sperimentare e manipolare un universo digitale. Attraverso la realtà virtuale, gli utenti si immergono in scenari simulati che possono replicare situazioni reali o creare esperienze completamente nuove. La realtà virtuale immersiva e quella non immersiva differiscono principalmente per il grado di coinvolgimento e interazione che offrono all'utente.

La *realtà virtuale immersiva* implica un'esperienza totale che isola l'utente dal mondo reale e lo immerge completamente in un ambiente digitale. Un esempio iconico di dispositivo per la realtà virtuale immersiva è l'Oculus Rift, il quale consente agli utenti di sentirsi come se fossero fisicamente presenti all'interno del mondo virtuale, interagendo con esso in modo naturale e intuitivo.

Invece, la *realtà virtuale non immersiva* offre un'esperienza meno coinvolgente, permettendo agli utenti di mantenere una consapevolezza del mondo reale mentre interagiscono con l'ambiente virtuale. Un esempio di dispositivo per la realtà virtuale non immersiva è Google Cardboard. Questo dispositivo è una soluzione semplice ed economica per sperimentare la realtà virtuale. Consiste in un supporto di cartone in cui inserire uno smartphone, che funge da schermo per visualizzare contenuti VR. Sebbene il Google Cardboard offra un'esperienza VR di base, non è paragonabile all'immersione e all'interattività di dispositivi più sofisticati, come l'Oculus Rift.

### 2.2.2 Hardware e Software per la Realtà Virtuale

La VR nasce dalla combinazione di tecnologie hardware e software avanzate. In questa sezione diamo uno sguardo a tali tecnologie:

#### Dispositivi di Input (Hardware)

- *Caschi VR*: sono visori che coprono gli occhi e spesso parte del viso, offrendo un'esperienza visiva immersiva. Hanno display interni e sensori per tracciare il movimento della testa, permettendo agli utenti di guardarsi intorno nell'ambiente virtuale. Oltre ai

display stereoscopici, molti visori VR includono anche audio spaziale per un'esperienza più coinvolgente.

- *Guanti VR*: sono dispositivi indossabili che rilevano i movimenti delle mani e delle dita. Ciò permette un'interazione più naturale e intuitiva con gli oggetti virtuali.
- *Controller VR*: sono dispositivi tenuti in mano, spesso dotati di pulsanti, joystick e sensori di movimento, che permettono di navigare e interagire con l'ambiente VR in modo più tradizionale.

### Sistemi di Tracciamento (Hardware)

- *Room-Scale Tracking*: questo sistema permette agli utenti di muoversi fisicamente in uno spazio definito, con i loro movimenti riflessi nell'ambiente virtuale. Ciò crea un'esperienza più realistica e immersiva.
- *Tracking Basato su Sensori*: utilizza sensori posti nell'area di gioco e sul dispositivo VR per tracciare con precisione i movimenti dell'utente in tempo reale.

### Motori di Gioco (Software)

- *Unity 3D*: è un motore di gioco estremamente versatile e potente, ampiamente riconosciuto e utilizzato sia nell'industria dei videogiochi che in altri settori, come la realtà virtuale e aumentata, l'animazione e l'architettura. La sua interfaccia user-friendly rende Unity accessibile sia ai principianti che ai professionisti, consentendo agli utenti di creare giochi e simulazioni interattive con relativa facilità. Una delle principali caratteristiche di Unity è la sua capacità di esportare progetti su una vasta gamma di piattaforme, tra cui Windows, MacOS, varie console di gioco, dispositivi mobili e web, il che lo rende uno strumento eccezionalmente versatile per gli sviluppatori.

Unity è noto per il suo potente motore grafico, che supporta sia grafica 2D che 3D, e per il suo sistema di scripting flessibile basato su C#, che permette una personalizzazione quasi illimitata e la creazione di gameplay complessi. Inoltre, dispone di un vasto Asset Store, dove gli sviluppatori possono acquistare o vendere asset e strumenti aggiuntivi, facilitando, così, lo sviluppo e riducendo i tempi di produzione. La community di Unity è anche una delle sue grandi forze, con un ampio numero di forum, tutorial e risorse online disponibili, che supportano gli sviluppatori a tutti i livelli di esperienza. L'insieme di questi aspetti positivi hanno indirizzato la nostra scelta all'utilizzo proprio di questo motore di gioco per lo sviluppo del Serious Game "Dom The Robot". Menzione d'onore ad *Unreal Engine* motore di gioco noto per la sua grafica di altissima qualità.

### Strumenti di Sviluppo Software

- *Ambienti di Sviluppo Integrati (IDEs)*: gli sviluppatori utilizzano IDE come Visual Studio per scrivere e testare codice per applicazioni VR.

- *Librerie e API*: sono utilizzate per semplificare lo sviluppo di applicazioni VR, fornendo funzioni precostruite per il tracciamento del movimento, la gestione dell'input e altre funzionalità.

### 2.2.3 Realtà Virtuale in Educazione e Formazione

La realtà virtuale riveste un ruolo sempre più cruciale nell'ambito della formazione ingegneristica, poiché consente agli studenti di applicare in modo concreto le conoscenze teoriche acquisite. Grazie a questa tecnologia, gli aspiranti ingegneri possono immergersi in ambienti simulati che riproducono fedelmente situazioni reali, offrendo un'esperienza di apprendimento estremamente coinvolgente e interattiva. Tale approccio permette di simulare scenari pericolosi o difficili da replicare nella realtà, fornendo agli studenti l'opportunità di affinare le loro abilità e acquisire fiducia prima di entrare nel campo. In sintesi, consentendo di colmare il divario tra la teoria e la pratica, preparando gli studenti in modo più completo ed efficace per le sfide del mondo reale, la realtà virtuale rappresenta una risorsa indispensabile per la formazione ingegneristica.

## 2.3 Realtà Mista: Unire il Fisico e il Digitale

### 2.3.1 Introduzione alla Realtà Mista

La realtà mista è una tecnologia avanzata che unisce elementi della VR e della AR per creare un ambiente interattivo in cui oggetti digitali e informazioni virtuali vengono sovrapposti e integrati nel mondo reale. Questa fusione consente agli utenti di interagire con oggetti tridimensionali e dati digitali mentre sono immersi nel loro ambiente fisico, offrendo esperienze immersive e coinvolgenti. È la più giovane delle tre e quindi anche quella più innovativa.

### 2.3.2 Tecnologie chiave della Realtà Mista

Una delle tecnologie chiave nella realtà mista è la "tracciatura spaziale", che è fondamentale per la sua funzionalità. La tracciatura spaziale si riferisce alla capacità di un sistema di percepire e monitorare con precisione la posizione e l'orientamento di un utente e degli oggetti nel mondo reale. Ciò avviene attraverso l'uso di sensori come telecamere, accelerometri e giroscopi, che rilevano i movimenti e le posizioni degli oggetti nell'ambiente circostante. Questa informazione viene poi utilizzata per sincronizzare in tempo reale gli oggetti virtuali con la vista e la posizione dell'utente, consentendo interazioni fluide e realistiche tra il mondo digitale e quello reale. La tracciatura spaziale è una componente essenziale per creare esperienze coinvolgenti e immersive nella realtà mista ed è supportata da hardware avanzati che includono dispositivi come visori VR, occhiali AR e controller specializzati. Questi dispositivi lavorano in sinergia con i sensori di tracciatura spaziale per offrire agli utenti un'esperienza completa e coinvolgente nella realtà mista.

L'ultimo dispositivo che adotta questa tecnologia rilasciato sul mercato è il Vision Pro di Apple (Figura 2.3), che rappresenta un balzo significativo nel campo della realtà mista, delineando il futuro dell'informatica. Questo dispositivo combina elementi di VR e AR, creando un'esperienza immersiva unica. Tra le sue funzioni principali, il visore offre un'interazione intuitiva attraverso gesti e comandi vocali, permettendo agli utenti di navigare in ambienti virtuali con estrema facilità. Inoltre, grazie a sensori avanzati e a una telecamera ad alta definizione, il dispositivo è in grado di sovrapporre elementi virtuali al mondo reale in modo fluido e realistico, arricchendo l'esperienza dell'utente. Questa tecnologia trova applicazioni in svariati campi, dalla formazione professionale al gaming, fino al design e alla medicina, dimostrando come il visore Apple non sia solo un prodotto di intrattenimento, ma uno strumento versatile per esplorare nuove frontiere dell'informatica e dell'interazione uomo-macchina.



**Figura 2.3:** Esempio di realtà mista di Vision Pro

### 2.3.3 Prospettive Future della Realtà Mista

Le prospettive future della realtà mista sono entusiasmanti e promettono di rivoluzionare numerosi settori. Con l'evoluzione continua delle tecnologie AR e VR, si prevede che la realtà mista diventerà sempre più integrata nella vita quotidiana, offrendo esperienze immersive e interattive in quasi tutti i campi. Nel mondo del lavoro, ad esempio, la realtà mista potrebbe facilitare la collaborazione a distanza, consentendo ai colleghi di interagire in uno spazio virtuale condiviso come se fossero fisicamente presenti nello stesso luogo. Anche l'interazione sociale potrebbe essere arricchita dalla realtà mista, permettendo alle persone di connettersi in modi nuovi e più profondi, superando le barriere fisiche. Le prospettive future della realtà mista indicano un'integrazione sempre più profonda tra il mondo fisico e quello virtuale, aprendo la strada a nuove forme di esperienza, apprendimento e connessione umana.

---

## Comprendere il Linguaggio Naturale: Approcci e Tecnologie nell'era dell'IA

---

*In un mondo dove le parole rappresentano il tessuto dell'esistenza, il Natural Language Understanding emerge come un faro di comprensione, una chiave per decifrare l'intricato linguaggio dell'anima umana. In questa odissea tecnologica, le macchine, un tempo mute e incomprensibili, apprendono a parlare il linguaggio del cuore, esplorando gli abissi dell'espressione umana.*

### 3.1 Fondamenti del Natural Language Understanding

#### 3.1.1 Introduzione al Natural Language Understanding

Il Natural Language Understanding (NLU) è una branca cruciale dell'Intelligenza Artificiale e del trattamento automatico del linguaggio naturale, che si dedica alla capacità dei computer di comprendere e interpretare il linguaggio umano in modo significativo. La sua importanza risiede nella capacità di trasformare il linguaggio naturale, ricco e complesso, in dati strutturati e utilizzabili dalle macchine, permettendo, così, interazioni più intuitive e naturali tra umani e computer. Il campo dell'NLU si è evoluto notevolmente nel corso degli anni, iniziando dalle prime ricerche negli anni '50 e '60, con lo sviluppo di modelli basati su regole, fino all'era moderna del Deep Learning e delle reti neurali, che hanno portato a significativi progressi nella comprensione del linguaggio.

Le componenti chiave dell'NLU includono la capacità di riconoscere l'intento dell'utente, l'estrazione di entità, la comprensione del contesto, e la gestione del linguaggio ambiguo o figurato. Queste abilità permettono alle macchine di processare e interpretare query umane, eseguire comandi e, persino, partecipare a conversazioni complesse. Tuttavia, l'NLU affronta numerose sfide, come la gestione di variazioni linguistiche, dialetti, ironia e ambiguità, che rendono la comprensione del linguaggio un compito particolarmente complesso. Inoltre, il rispetto della privacy e delle questioni etiche relative all'uso dei dati linguistici è fondamentale in questo campo. La continua ricerca e lo sviluppo in questo settore mirano a superare queste

sfide, rendendo l’NLU sempre più sofisticato e capace di comprendere il linguaggio umano in tutte le sue sfumature.

### 3.1.2 Metodologie di Analisi nell’NLU

Le metodologie di analisi nel Natural Language Understanding hanno subito un’evoluzione significativa nel corso degli anni, passando da approcci basati su regole a metodi statistici e, infine, a tecniche basate sull’apprendimento automatico. Inizialmente, l’NLU si affidava a sistemi basati su regole, i quali utilizzavano insiemi di regole grammaticali e linguistiche predefinite per interpretare il testo. Questi sistemi erano efficaci in contesti limitati e ben definiti, ma mancavano di flessibilità e scalabilità, e spesso non riuscivano a gestire le sottili variazioni del linguaggio naturale. Con l’avvento dei metodi statistici, l’NLU ha guadagnato una maggiore capacità di generalizzazione, sfruttando grandi volumi di dati testuali per identificare pattern e correlazioni linguistiche. Questi approcci, basati su modelli probabilistici come i modelli n-gram e le reti Bayesiane, hanno migliorato la capacità dei sistemi di NLU di gestire varietà e ambiguità linguistiche.

L’introduzione dell’apprendimento automatico e, in particolare, delle tecniche di Deep Learning, ha rappresentato un ulteriore passo in avanti, portando a una comprensione del linguaggio ancora più sofisticata e profonda. L’utilizzo di reti neurali, in particolare le architetture come i Long Short-Term Memory (LSTM) e i Transformer, ha permesso la creazione di modelli di NLU che apprendono rappresentazioni linguistiche complesse e astratte dai dati, migliorando notevolmente la precisione e l’efficacia nell’interpretazione del linguaggio naturale. Questi modelli sono in grado di catturare le sfumature semantiche e sintattiche del linguaggio, nonché di contestualizzare le informazioni, rendendo possibile una comprensione molto più ricca e completa del linguaggio umano. L’evoluzione delle metodologie di analisi nell’NLU riflette una transizione da approcci rigidi e limitati a metodi sempre più flessibili, potenti e adattabili, che continuano a spingere i confini della capacità delle macchine di comprendere il linguaggio umano.

### 3.1.3 Applicazioni dell’NLU

Le applicazioni del Natural Language Understanding (NLU) sono vaste e toccano diversi ambiti, mostrando l’impatto significativo di questa tecnologia nella vita quotidiana e nel mondo degli affari. Un’area di grande rilevanza è quella dell’assistenza virtuale e dei chatbot, dove l’NLU gioca un ruolo cruciale nel consentire a questi sistemi di interpretare e rispondere in modo appropriato alle richieste degli utenti. Questi assistenti virtuali sono in grado di comprendere domande complesse, eseguire compiti specifici e offrire assistenza personalizzata, migliorando l’esperienza utente in ambiti come il servizio clienti, le prenotazioni e l’automazione domestica.

Un’altra applicazione significativa dell’NLU è l’analisi del sentiment, che consente alle macchine di identificare e categorizzare le opinioni espresse nel testo, fornendo, così, intuizioni preziose per il monitoraggio dei social media, il feedback dei clienti, e la ricerca di

mercato. Questa capacità di discernere il tono emotivo e le opinioni nei dati testuali aiuta le aziende a comprendere meglio i loro clienti e a reagire in modo più efficace alle loro esigenze e aspettative.

Inoltre, l’NLU facilita l’estrazione di informazioni e il riconoscimento di entità da grandi quantità di testo. Questo include l’identificazione di nomi di persone, organizzazioni, luoghi e altri elementi specifici, rendendo possibile l’organizzazione e l’analisi automatica di grandi volumi di dati non strutturati. Queste capacità sono particolarmente utili per applicazioni come l’intelligence competitiva, la ricerca accademica e l’analisi di dati in settori come finanza, salute e sicurezza. In sintesi, le applicazioni dell’NLU trasformano il modo in cui interagiamo con la tecnologia, offrendo strumenti potenti per l’analisi e l’interpretazione del linguaggio, e aprendo nuove possibilità nell’interazione uomo-macchina e nell’analisi dei dati.

## 3.2 Il Ruolo delle Reti Neurali e del Deep Learning

### 3.2.1 Fondamenti delle Reti Neurali nell’NLU

Le reti neurali rappresentano una componente fondamentale nel campo del Natural Language Understanding, fornendo un framework potente per l’elaborazione e l’analisi del linguaggio naturale. Possiamo vedere come questi sistemi imitino il funzionamento del cervello umano attraverso strati di neuroni artificiali interconnessi (figura 3.1). Ogni neurone in uno strato riceve input da vari neuroni dello strato precedente, li combina pesandoli con coefficienti (pesi), e applica una funzione di attivazione non lineare per determinare il suo output. Queste funzioni di attivazione, come la sigmoide, la tangente iperbolica o la ReLU (Rectified Linear Unit), sono cruciali per permettere alla rete di modellare relazioni complesse e non lineari. L’apprendimento in una rete neurale avviene attraverso un processo di ottimizzazione, dove i pesi delle connessioni tra i neuroni vengono aggiustati per ridurre l’errore tra l’output predetto e quello desiderato.

Nel contesto dell’NLU, le reti neurali sono utilizzate per comprendere e interpretare il linguaggio naturale in modi che vanno oltre le capacità dei tradizionali metodi basati su regole o dei metodi statistici. Ad esempio, le reti neurali possono essere addestrate per riconoscere l’intento dietro una domanda, per identificare entità e relazioni chiave in un testo, o per generare risposte linguistiche coerenti e pertinenti. Queste reti, specialmente quelle basate su architetture avanzate, come le reti ricorrenti (RNN) e i Transformer, sono in grado di gestire sequenze di dati (come frasi o paragrafi) e catturare le relazioni contestuali e temporali all’interno del testo.

Le applicazioni delle reti neurali nell’NLU sono ampie e includono miglioramenti significativi; la capacità di queste reti di elaborare il linguaggio in modo più naturale e intuitivo ha portato a un’interazione uomo-macchina più fluida e a una migliore comprensione dei dati testuali.



**Figura 3.1:** Illustrazione di un cervello umano e una rete neurale che collaborano in insieme

### 3.2.2 Deep Learning e NLU

Un aspetto fondamentale delle reti neurali è l'apprendimento profondo o "Deep Learning", dove le reti con molti strati (chiamati "deep") sono in grado di estrarre automaticamente caratteristiche e rappresentazioni di alto livello dai dati. Le architetture di Deep Learning più significative in questo ambito includono le Reti Neurali Ricorrenti (RNN), le Long Short-Term Memory (LSTM) e i Transformer, ognuna delle quali ha contribuito in modo unico all'avanzamento dell'NLU.

Le RNN sono particolarmente adatte per lavorare con sequenze di dati, come il testo, poiché sono capaci di mantenere uno stato (o memoria) delle informazioni precedentemente analizzate. Questo le rende ideali per applicazioni come l'analisi del linguaggio, dove il contesto e la sequenzialità delle parole sono fondamentali. Tuttavia, le RNN tradizionali si scontrano con il problema della dipendenza a lungo termine, dove diventa difficile per la rete ricordare informazioni da input lontani nella sequenza.

Le LSTM, una variante avanzata delle RNN, superano questo ostacolo introducendo celle di memoria specializzate e meccanismi di gate. Questi gate controllano il flusso di informazioni all'interno della rete, permettendo alle LSTM di conservare informazioni rilevanti per lunghi periodi e di scartare quelle non necessarie. Questa caratteristica le rende estremamente efficaci nel modellare dipendenze a lungo termine nel linguaggio, essenziali per compiti



come la comprensione del testo e la generazione di risposte coerenti.

L'architettura dei Transformer, d'altra parte, rappresenta un ulteriore salto in avanti. Basata su un meccanismo di attenzione, questa architettura permette al modello di focalizzarsi su parti specifiche dell'input quando produce un output, indipendentemente dalla loro posizione nella sequenza. Ciò consente una comprensione del contesto molto più sofisticata e una maggiore efficienza nel trattamento di sequenze lunghe, rendendo i Transformer particolarmente adatti per applicazioni di traduzione automatica, generazione di testo e analisi semantica.

Il vantaggio principale del Deep Learning nel trattamento del linguaggio naturale risiede nella sua capacità di apprendere automaticamente rappresentazioni complesse e astratte del linguaggio da grandi quantità di dati testuali. Tale approccio riduce la necessità di feature engineering manuale e permette ai modelli di adattarsi a varie sfumature linguistiche, contesti e stili di comunicazione. Inoltre, i modelli basati su Deep Learning continuano a migliorare con l'aumento dei dati disponibili, diventando sempre più accurati e versatili nel tempo. Grazie a queste qualità, il Deep Learning ha trasformato l'NLU, rendendolo uno strumento più potente e flessibile per l'interpretazione e la generazione del linguaggio naturale.

### 3.2.3 Casi Studio e Esempi Pratici

Il successo di ChatGPT (Generative Pretrained Transformer) è dovuto proprio all'architettura Transformer; ma, a livello pratico in cosa questa struttura eccelle rispetto alle altre? Per rispondere a questa domanda consideriamo questo semplice esempio: Nella frase *"Il cugino di mio zio materno che abita dietro la chiesa vicina a casa mia è spagnolo"* per noi è semplice capire che *"è spagnolo"* si riferisce a *"il cugino di mio zio"*, ma non è affatto scontato per una rete neurale che, solitamente, elabora le frasi in maniera sequenziale, parola per parola. I Transformer, invece, grazie al loro meccanismo di attenzione, sono in grado di analizzare l'intera frase contemporaneamente e di stabilire relazioni tra tutte le parti della frase, indipendentemente dalla loro posizione. Ciò porta i Transformer ad eccellere nell'analizzare frasi complesse e nel cogliere relazioni contestuali e riferimenti incrociati all'interno di un testo, superando i limiti delle architetture sequenziali, come le RNN e le LSTM. Uno dei casi d'uso pratici più efficaci del Natural Language Understanding è l'impiego in sistemi di assistenza virtuale personalizzata nel settore delle tecnologie finanziarie, in particolare nell'uso di chatbot per il servizio clienti bancario (Figura 3.2), integrato nel sito web o nell'app di una banca, progettato per assistere i clienti in tempo reale con le loro domande riguardanti il proprio conto bancario. Un cliente potrebbe ad esempio, scrivere: *"Vorrei sapere quanto ho speso in ristoranti questo mese."* Utilizzando l'NLU, il chatbot è in grado di interpretare la richiesta, riconoscendo parole chiave come *"speso"*, *"ristoranti"* e *"questo mese"*. Il sistema, poi, accede alle informazioni del conto del cliente, analizza le transazioni e fornisce una risposta precisa, come *"Hai speso un totale di 150 euro in ristoranti in questo mese."*

In un altro scenario, un cliente potrebbe chiedere: *"Come faccio a trasferire denaro sul mio conto di risparmio?"* Il chatbot, tramite l'NLU, comprende l'intento dell'utente di effettuare



**Figura 3.2:** Illustrazione di un chatbot che tramite l’NLU consiglia strategie operative

un trasferimento di denaro e può guidare il cliente attraverso i passaggi necessari, offrendo un’interfaccia intuitiva e risposte immediate.

Questo caso d’uso mostra come l’NLU possa trasformare l’interazione cliente-banca, rendendo il servizio clienti più efficiente, accessibile e personalizzato. L’NLU permette ai chatbot di fornire risposte rapide e pertinenti, riducendo i tempi di attesa e migliorando l’esperienza complessiva del cliente, il tutto mantenendo un alto grado di precisione e comprensione del linguaggio naturale.

### 3.3 Nuove Frontiere e Strumenti nel Mondo dell’NLU

#### 3.3.1 Panoramica delle Piattaforme di NLU

Nel panorama delle piattaforme di Natural Language Understanding, esistono diversi strumenti e toolkit che si distinguono per le loro capacità, caratteristiche e aree di applicazione. Le aziende che offrono i migliori servizi di questo tipo sono: Google, IBM, Microsoft, Amazon e Hugging Face (figura 3.3).

*Google Cloud Natural Language* si distingue per la sua integrazione profonda con l’ecosistema Google Cloud, offrendo potenti funzionalità di analisi del linguaggio, tra cui l’identificazione di entità, l’analisi del sentimento e l’analisi della sintassi. Questa piattaforma

è particolarmente apprezzata per la sua facilità d’uso e per l’accesso a un vasto set di dati e risorse di Google.

*IBM Watson*, d’altra parte, è rinomato per la sua robusta suite di strumenti di NLU, che comprende la comprensione del linguaggio naturale, la classificazione del testo e la personalità insights. Watson è particolarmente forte nell’apprendimento da dati specifici del settore, permettendo un’analisi del linguaggio altamente personalizzata e adatta a contesti aziendali specifici.

*Microsoft Azure Cognitive Services* offre una gamma di servizi di NLU, inclusi il riconoscimento del parlato, la traduzione e l’analisi del linguaggio. Questa piattaforma si distingue per la sua integrazione con altri servizi Microsoft e per la sua flessibilità, rendendola una scelta popolare per le aziende che già utilizzano altri prodotti Microsoft.

*Amazon Comprehend* è un servizio di NLU completamente gestito che utilizza l’apprendimento automatico per scoprire insight e relazioni nel testo. È noto per la sua facilità di integrazione con altri servizi AWS, la sua scalabilità e l’efficacia nell’analisi di grandi volumi di dati testuali.

*Hugging Face*, infine, si è rapidamente affermato come un leader nell’ambito dei modelli di NLU basati su Transformer, offrendo un’ampia gamma di modelli pre-addestrati e facilmente personalizzabili. La piattaforma è particolarmente apprezzata nella comunità di ricerca e sviluppo per la sua apertura, facilità di utilizzo e le sue performance all’avanguardia.



**Figura 3.3:** Illustrazione delle 6 potenze mondiali che competono nella guerra dell’AI

### 3.3.2 Hugging Face: Un Ecosistema Innovativo

Hugging Face rappresenta un ecosistema innovativo nel panorama dell’Intelligenza Artificiale, specialmente nel campo del trattamento del linguaggio naturale (NLP). Fondata nel 2016, Hugging Face ha iniziato come una startup focalizzata sull’IA per creare un’app

di chatbot divertente. Tuttavia, si è rapidamente evoluta diventando una piattaforma di riferimento per la ricerca e lo sviluppo in NLP. Uno dei principali pilastri di Hugging Face è la sua libreria "Transformers", che fornisce una vasta gamma di modelli pre-addestrati come BERT, GPT-2, GPT-3 e T5. Questi modelli sono diventati standard nell’industria per compiti di NLP come la comprensione del linguaggio, la generazione di testo, e la traduzione automatica, grazie alla loro capacità di gestire efficacemente il linguaggio naturale in modo flessibile e potente.

Oltre a offrire modelli all’avanguardia, Hugging Face si distingue per il suo impegno nella creazione di una comunità collaborativa e open-source. La piattaforma incoraggia i ricercatori e gli sviluppatori a condividere i propri modelli e a collaborare su progetti, accelerando così l’innovazione e la diffusione delle conoscenze nel campo del NLP. Questo approccio ha portato Hugging Face a diventare un punto di riferimento per chiunque lavori nel settore dell’IA, contribuendo significativamente al progresso della ricerca e dello sviluppo in NLP.

I contributi di Hugging Face non si limitano alla fornitura di strumenti tecnologici; l’azienda si impegna anche in iniziative per promuovere l’etica nell’IA, la trasparenza e la responsabilità, affrontando questioni importanti come il bias nei dati e la privacy. In conclusione, Hugging Face non è solo una piattaforma per modelli di NLP avanzati, ma anche un catalizzatore per l’innovazione, la collaborazione e il progresso etico nel campo dell’Intelligenza Artificiale.

### 3.3.3 Tendenze Future e Direzioni di Ricerca

Le tendenze future e le direzioni di ricerca nel campo del Natural Language Understanding (NLU) si prospettano estremamente dinamiche e promettenti, con diverse aree che si prevede saranno al centro dell’innovazione nei prossimi anni. Una tendenza emergente è l’ulteriore sviluppo di modelli di linguaggio basati su Deep Learning, come i Transformer, che continueranno a essere perfezionati per migliorare la comprensione e la generazione del linguaggio naturale. Si prevede che questi modelli diventeranno ancora più potenti e versatili, con un’enfasi crescente sulla personalizzazione e l’adattabilità a specifici contesti e settori.

Un’altra area di ricerca importante è l’incremento dell’interattività e della multimodalità nei sistemi di NLU. Ciò include l’integrazione di input visivi, sonori e testuali per creare esperienze utente più ricche e sistemi più efficaci in contesti complessi. Ad esempio, i sistemi di NLU del futuro potrebbero essere in grado di analizzare contemporaneamente il linguaggio, le espressioni facciali e il tono della voce per una comprensione più profonda delle intenzioni e delle emozioni degli utenti.

Inoltre, si prevede un aumento della ricerca sull’etica, la privacy e il bias nell’NLU. Man mano che i sistemi di NLU diventano più pervasivi è fondamentale affrontare le sfide relative all’utilizzo responsabile dei dati, alla protezione della privacy degli utenti e alla minimizzazione dei pregiudizi nei modelli di linguaggio. Questo aspetto è cruciale per garantire che i progressi nell’NLU siano equi, trasparenti e benefici per tutti.

---

## Ideazione e progettazione del serious game "Dom the Robot"

---

*In questo capitolo esploreremo la genesi di "Dom the Robot", un Serious Game educational sulla teoria dei colori sviluppato interamente da zero. In un'era in cui l'Intelligenza Artificiale sta rimodellando il nostro mondo, "Dom the Robot" vuole dimostrare l'innovativo impiego dell'AI nel settore dei giochi educativi. La sua realizzazione cerca di evidenziare come l'accessibilità a risorse open source possa democratizzare la tecnologia AI, rendendola uno strumento prezioso anche per gli sviluppatori con risorse più limitate.*

### 4.1 Introduzione, Contesto e Pubblico di Destinazione

#### 4.1.1 Contestualizzazione del Serious Game

La realizzazione di "Dom the Robot" si inserisce in un contesto in cui l'Intelligenza Artificiale sta rivoluzionando il panorama tecnologico e, di conseguenza, anche il settore dei Serious Game. Questo progetto mira a dimostrare l'impatto significativamente positivo che l'AI può avere nello sviluppo di giochi con finalità educative e formative. Una delle chiavi di questo impatto è l'accessibilità alle risorse open source, come quelle offerte da Hugging Face, che semplificano notevolmente l'implementazione dell'Intelligenza Artificiale anche per sviluppatori con budget ridotti.

"Dom the Robot" rappresenta un esempio concreto e molto semplice di come un Serious Game possa integrare elementi di AI per arricchire l'esperienza di apprendimento. Il gioco utilizza un modello di Intelligenza Artificiale per gestire interazioni più o meno complesse con l'utente, pur mantenendo un approccio intuitivo e accessibile. Questa semplicità, tuttavia, non sminuisce l'importanza del progetto; al contrario, evidenzia la versatilità e l'efficacia dell'AI nel contesto educativo.

Riflettendo su "Dom the Robot", si può osservare come le piattaforme open source abbiano giocato un ruolo cruciale nell'abbattere le barriere all'adozione dell'Intelligenza Artificiale. Ciò ha aperto nuove opportunità anche per piccoli centri di sviluppo, permettendo ad essi di

integrare tecnologie avanzate nei loro progetti senza l'onere di sviluppare soluzioni complesse da zero, il che li ha portati a realizzare progetti educativi più ambiziosi e strutturati.

### 4.1.2 Obiettivi Educativi e di Apprendimento

Il cuore educativo di "Dom the Robot" risiede nell'insegnamento della teoria dei colori, con particolare attenzione alla comprensione e alla sperimentazione dei colori primari e alla loro interazione per la creazione dei colori secondari. Attraverso un'esperienza di gioco interattiva, il Serious Game si propone di guidare i bambini nella scoperta dei principi fondamentali della miscelazione dei colori. Mentre i giocatori interagiscono con il robot Dom, sono invitati a sperimentare come i colori primari - rosso, blu e giallo - possano essere combinati in modi specifici per produrre una gamma di colori secondari, quali arancione, verde e viola. Questo processo cerca di stimolare la curiosità dei giovani giocatori e di rafforzare la loro comprensione intuitiva sulle basi della teoria del colore. "Dom the Robot" si impegna quindi a fornire un ambiente ludico ma educativo, dove l'apprendimento dei concetti cromatici avviene in modo divertente grazie ai movimenti e alle battute spiritose di Dom, favorendo un approccio dinamico all'educazione artistica.

### 4.1.3 Pubblico di Destinazione e Profilo degli Utenti

"Dom the Robot" è stato concepito specificatamente per bambini in età prescolare, tra i 3 e i 4 anni, un periodo fondamentale per lo sviluppo cognitivo e l'apprendimento di nuovi concetti. Questo gruppo di utenti, che potrebbe non avere ancora familiarità con la teoria dei colori o che necessita di rafforzare tali conoscenze, troverà in "Dom the Robot" un approccio giocoso e interattivo all'apprendimento.

## 4.2 Analisi dei Requisiti

### 4.2.1 Requisiti Funzionali

I requisiti funzionali di 'Dom the Robot' delineano le caratteristiche essenziali che il gioco deve possedere per raggiungere i suoi obiettivi educativi e fornire un'esperienza di apprendimento il più coinvolgente e interattiva possibile. Questi requisiti definiscono le azioni chiave, le interazioni e le funzionalità che dovranno essere eseguite:

- *Interazione Vocale o Testuale*: il robot deve essere in grado di ricevere e interpretare input vocali o testuali dall'utente.
- *Risposta ai Comandi di Saluto*: Dom deve essere programmato per rispondere ai comandi di saluto degli utenti con un'azione appropriata, quale un gesto e una frase di saluto.
- *Esecuzione di Azioni di Ballo*: il robot deve avere la capacità di eseguire movimenti di danza quando richiesto dall'utente e quando avviene la corretta composizione di un colore secondario; ciò renderà l'esperienza più divertente.

- *Dimostrazione della Composizione dei Colori Secondari*: Dom deve essere in grado di mostrare agli utenti come i colori primari possano essere combinati per creare colori secondari.
- *Controllo e Feedback sulla Composizione dei Colori*: il robot deve essere in grado di comprendere quale colore secondario l'utente intende comporre e con quale coppia di colori primari, deve poter eseguire un controllo sulla correttezza della miscelazione effettuata dall'utente, e fornire un feedback appropriato. In caso di errore, deve mostrare la corretta composizione del colore desiderato.
- *Movimento Verso i Cubi dei Colori Primari*: Dom deve essere programmato per spostarsi autonomamente verso i cubi di colore primario indicati dall'utente, facilitando l'esplorazione e l'interazione con i diversi elementi del gioco.
- *Gestione degli Errori*: il gioco deve includere un sistema per gestire gli errori di input in modo delicato e costruttivo.
- *Feedback Visivo e Sonoro Chiari*: ogni azione del robot deve essere accompagnata da feedback visivi e sonori come frasi ed animazioni, per rinforzare il processo di apprendimento e fornire un'esperienza di gioco coinvolgente.

#### 4.2.2 Requisiti non Funzionali

I requisiti non funzionali sono necessari per garantire la qualità, l'efficienza e l'accessibilità del gioco, eccoli di seguito:

- *Usabilità*: il gioco deve essere intuitivo e facile da usare per bambini in età prescolare.
- *Documentazione*: è necessario fornire una documentazione chiara che mostri in maniera intuitiva e semplice come poter giocare.
- *Performance e Tempi di Risposta*: "Dom the Robot" deve avere tempi di risposta rapidi e performance fluide per mantenere l'attenzione e l'interesse dei bambini.
- *Scalabilità*: il gioco deve essere progettato per essere facilmente aggiornabile e ampliabile, consentendo l'aggiunta di nuove funzionalità o contenuti in futuro.
- *Design Estetico e Qualità Grafica*: la qualità visiva e l'attrattiva estetica sono molto importanti per catturare l'attenzione dei bambini, il gioco deve quindi avere un design accattivante.



## 4.3 Progettazione del Serious Game

### 4.3.1 Struttura del gioco

Il gioco "Dom the Robot" è caratterizzato da un layout tridimensionale semplice e intuitivo; il terreno di gioco è uno spazio aperto in cui il robot Dom, protagonista del gioco, si muove agilmente. Disposti a una certa distanza da esso ci sono tre cubi, ognuno rappresentante un colore primario: rosso, giallo e blu. Questi elementi costituiscono i componenti chiave per l'attività principale del gioco, ovvero la miscelazione dei colori.

La dinamica di gioco è diretta e coinvolgente: i giocatori interagiscono con Dom tramite prompt da tastiera, impartendo istruzioni che il robot eseguirà. Su comando, Dom si avvicina ai cubi dei colori primari e li manovra, portandoli davanti alla telecamera. Qui, i cubi entrano in contatto, permettendo ai giocatori di osservare la trasformazione che dà vita ai colori secondari. Questo processo visuale cerca di stimolare la curiosità dei bambini e aiutarli a comprendere concretamente come i colori primari si combinano per formare nuove tonalità.

"Dom the Robot" non presenta livelli progressivi di difficoltà; invece, offre un ambiente di gioco libero e flessibile, dove i bambini possono esplorare e interagire con Dom in maniera illimitata incoraggiando la sperimentazione e la ripetizione, elementi fondamentali per l'apprendimento nella fascia d'età prescolare.

Una sfida stimolante emerge quando il giocatore decide di comporre autonomamente un colore secondario. Questo momento richiede ai bambini di indicare a Dom i giusti cubi dei colori primari da prendere per ottenere la tonalità desiderata mettendo alla prova la comprensione della teoria dei colori.

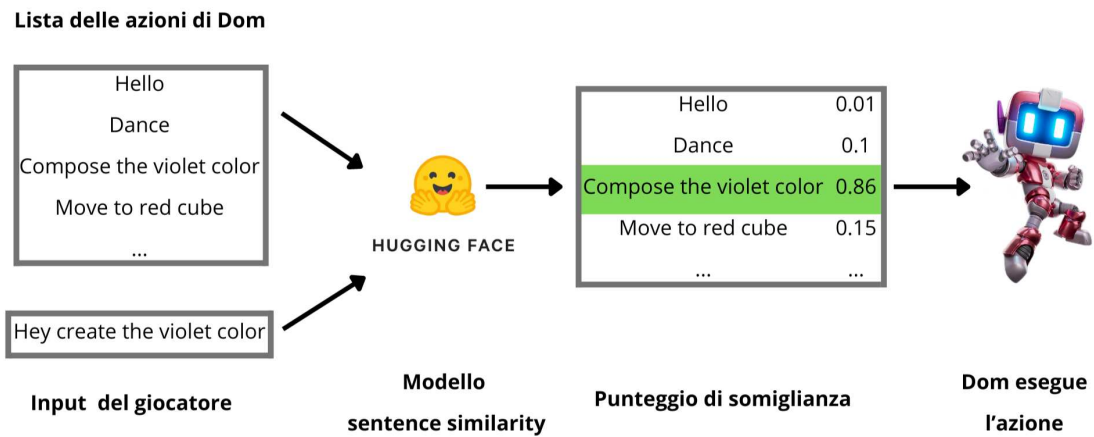
### 4.3.2 Un modello AI che rende Dom uno "Smart Robot"

Il cuore tecnologico che rende "Dom the Robot" uno "Smart Robot" è un modello di Intelligenza Artificiale basato sulla somiglianza delle frasi (sentence similarity). Questo approccio innovativo permette di dare al giocatore un livello di libertà molto elevato nell'interazione con il robot. Invece di limitare la comunicazione a click su pulsanti o a comandi predefiniti, il gioco si apre a un'interazione discorsiva più naturale e dinamica. Il modello agisce come un interprete intelligente tra il giocatore e Dom, analizzando gli input testuali e selezionando l'azione più appropriata tra quelle disponibili nel repertorio del robot (Figura 4.1).

Questo modello di Intelligenza Artificiale è particolarmente efficace nell'elaborare e comprendere variazioni e sfumature del linguaggio umano. Ad esempio, se un giocatore scrive "*ehy, create the green color*", anche se Dom non è stato programmato specificamente per riconoscere questa frase, il modello di sentence similarity riesce a stabilire un collegamento tra la richiesta e l'azione corrispondente di "*compose the green color*". Ciò elimina la necessità di mappare manualmente ogni possibile interazione, rendendo l'interazione con il robot più fluida e meno limitata.

Il modello "*all-MiniLM-L6-v2*" di Hugging Face, utilizzato in "Dom the Robot", è un esempio avanzato di modello di sentence similarity che funziona mappando frasi e paragrafi





**Figura 4.1:** Schema di funzionamento del modello all-MiniLM-L6-v2

in uno spazio vettoriale denso a 384 dimensioni. Questo modello può essere impiegato per varie applicazioni, come il raggruppamento (clustering) o la ricerca semantica. Vediamo più nel dettaglio come funziona:

1. *Trasformazione delle Frasi in Vettori:* il modello converte le frasi in rappresentazioni vettoriali, o embedding, utilizzando una rete neurale basata sull'architettura Transformer. Questi embedding catturano le caratteristiche semantiche e sintattiche delle frasi.
2. *Calcolo della Somiglianza:* una volta ottenuti gli embedding, il modello calcola la somiglianza tra le frasi confrontando i loro vettori. Ciò avviene misurando la distanza tra gli embedding.
3. *Uso del Modello in Dom the Robot:* nel gioco, quando l'utente inserisce un comando, il modello trasforma questo input in un embedding e lo confronta con gli embedding delle possibili azioni di Dom. L'azione con l'embedding più simile all'input dell'utente viene selezionata per l'esecuzione.
4. *Pre-Training e Fine-Tuning:* il modello è inizialmente pre-addestrato su un grande dataset di testo. Durante questo processo, il modello impara a comprendere la lingua, acquisendo una conoscenza generale del linguaggio naturale. Questo pre-addestramento è fondamentale perché fornisce al modello una base solida per elaborare e interpretare le frasi. Dopo il pre-addestramento, il modello è sottoposto a un fine-tuning, un processo in cui viene ulteriormente allenato su un dataset specifico. In questo caso, il modello "all-MiniLM-L6-v2" è stato affinato su un dataset di oltre 1 miliardo di coppie di frasi. Il fine-tuning utilizza un obiettivo di apprendimento contrastivo, il che significa che esso impara a riconoscere quali frasi in un insieme sono correlate tra loro. Ad esempio, data una frase "A", il modello impara a identificare quale altra frase "B" nel dataset è più strettamente correlata a "A".

---

## Realizzazione Tecnica di "Dom the Robot"

---

*Questo capitolo descrive il percorso seguito nella realizzazione tecnica di Dom the Robot, evidenziando gli sforzi e le sfide tecniche affrontate. Attraverso un'esplorazione dettagliata del processo di sviluppo, si mira a fornire una comprensione chiara dei metodi e delle tecnologie impiegate.*

### 5.1 Elementi Grafici e Scenografici

#### 5.1.1 Il Paesaggio di Background

Il paesaggio di sfondo di "Dom the Robot" (Figura 5.1) è stato ispirato dal progetto open source "Making Games with AI" di Thomas Simonini dal quale sono stati importati i prefabbricati dei vari oggetti. Questo ambiente scenico è caratterizzato da una semplice estetica naturalistica; una collinetta ondulata fa da sfondo principale, arricchita da alberi di tre diverse tipologie, che aggiungono varietà. Cespugli sparsi, alcuni massi e tronchi spezzati impreziosiscono ulteriormente la scena, mentre piccoli funghi posti ai piedi dei tronchi introducono un elemento di dettaglio e delicatezza. L'obiettivo di questa ambientazione è creare un'atmosfera tranquilla e immersiva, utile per l'esperienza di gioco educativo e interattivo che "Dom the Robot" vuole offrire ai suoi giovani utenti.



**Figura 5.1:** Paesaggio di sfondo

### 5.1.2 Game Objects in Scena

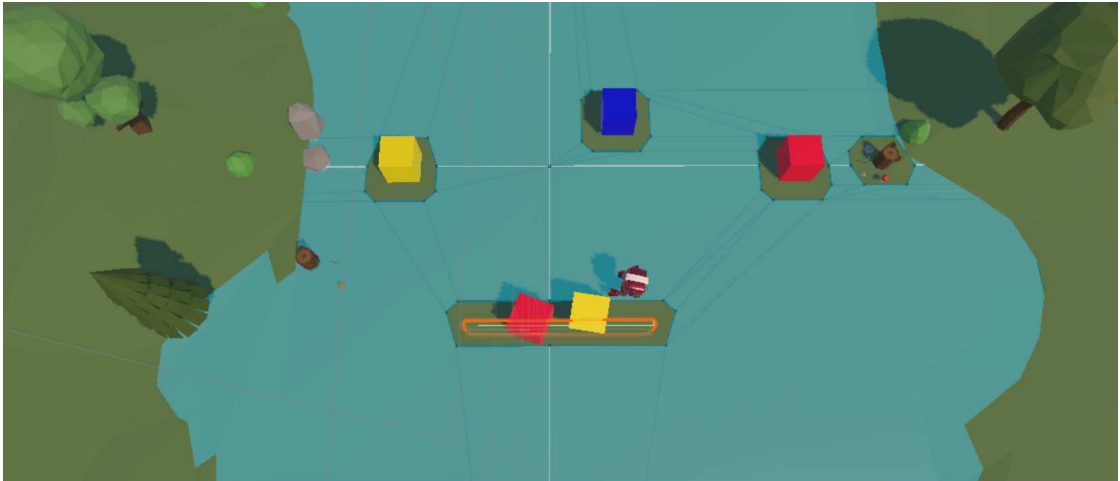
Un `GameObject` in Unity è un oggetto fondamentale nelle scene di gioco, che può rappresentare personaggi, oggetti di scena, scenari, telecamere, posizioni e altro ancora (Figura 5.2). Un `GameObject` di per sé non ha molta funzionalità, ma agisce come un contenitore per i Componenti, che sono parti funzionali di esso. Un `GameObject` può contenere un numero qualsiasi di componenti, che implementano una particolare funzionalità. Ad esempio, quando si applica un componente `Rigidbody` il `GameObject` diventa soggetto alle leggi della fisica del motore di gioco, come gravità e collisioni. Questo componente è essenziale per creare oggetti che interagiscono in modo realistico con l'ambiente di gioco, come cadere, rimbalzare, o essere spinti.

I `GameObject` che all'interno di "Dom The Robot" hanno un componente `rigidbody` sono:

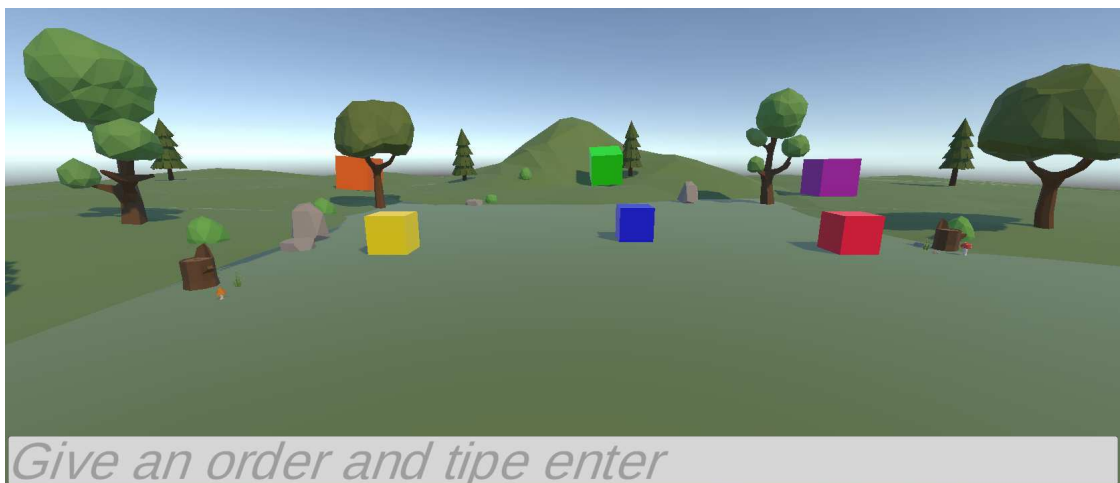
- I cubi dei colori primari, i quali hanno già attivato l'azione `Use Gravity`, responsabile di far agire la forza di gravità sul `GameObject`.
- I cubi dei colori secondari situati nello sfondo, utilizzati per dare contesto all'ambientazione, anch'essi con il componente `rigidbody` e con l'azione `Use Gravity` attiva. La loro presenza non è importante ai fini delle dinamiche di gioco.
- I cubi dei colori secondari che verranno visti cadere dall'alto non appena Dom avrà combinato due colori primari. Questo insieme di `GameObject` è l'unico che non compare in scena quando il gioco parte; inoltre, in essi, la componente `Use Gravity` è inizialmente disattivata in modo tale che possa essere attivata quando Dom compone il colore secondario, facendolo, così, cadere dall'alto, creando un effetto ottico piacevole.

Un altro `GameObject` importante è un rettangolo dello stesso colore e della stessa altezza del terreno in cui si muove Dom, in modo tale da non poter essere notato dal giocatore. La sua posizione coincide con quella in cui vengono rilasciati i cubi dei colori primari ed il suo ruolo è quello di rendere quell'area non calpestabile affinché Dom non ci passi attraverso spingendo via i cubi rilasciati da lui stesso, provocando un effetto spiacevole e disordinato. Ciò è possibile, in quanto l'AI di Unity calcola il terreno calpestabile (`Navigation Mesh`) in base alla disposizione degli oggetti sulla scena (Figura 5.3).

L'ultimo `GameObject` presente in scena è di tipo UI (`User Interface`) ed è il campo in cui il giocatore fornisce il prompt a Dom.



**Figura 5.2:** Navigation Mesh della scena di gioco



**Figura 5.3:** GameObjects in scena

### 5.1.3 La Creazione Grafica di Dom

La creazione grafica di Dom in "Dom the Robot" è il risultato di una collaborazione tra Mix And Jam, un canale YouTube specializzato nello sviluppo di giochi sperimentali, e Omni e Andrew del canale Kinematic. Il personaggio è stato creato con tecniche speciali per modellare e scolpire il suo design utilizzando un software chiamato Blender. Dopo aver creato il modello, gli autori hanno utilizzato un altro programma, Substance Painter, per aggiungere dettagli come colori e texture. Infine, hanno preparato Dom per muoversi realisticamente nei giochi aggiungendo una struttura interna, nota come rig umanoide, che permette al personaggio di avere movimenti simili a quelli umani (Figura 5.4). Dom è stato progettato per essere facilmente integrabile in prototipi di gioco, offrendo un design accattivante e versatile per diverse applicazioni di sviluppo (Figura 5.4).

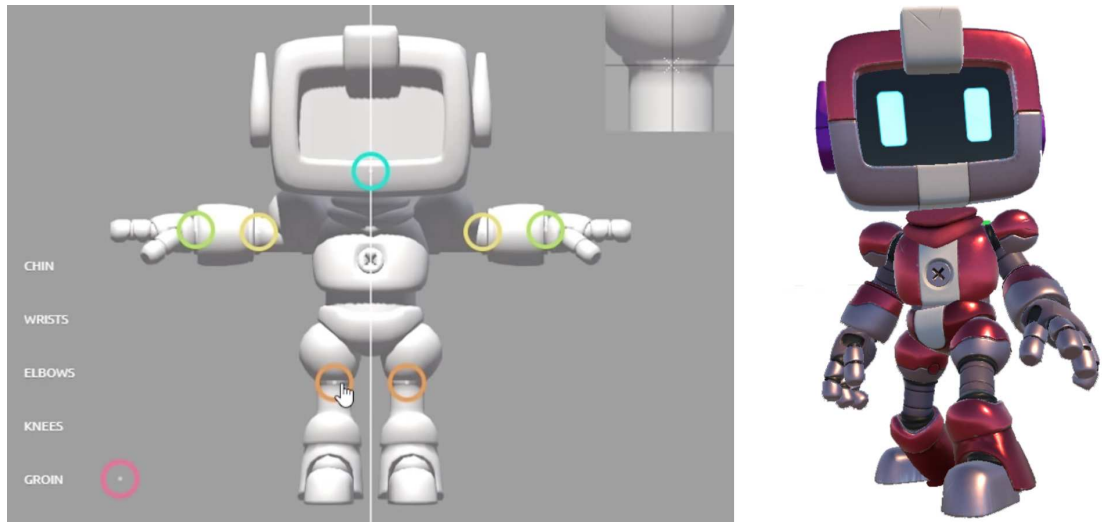


Figura 5.4: Rig umanoide e Dom in posizione di attesa

## 5.2 Dinamiche di Interazione e Integrazione AI

### 5.2.1 La Macchina degli Stati di Dom

Dom può eseguire sette azioni principali: creare un colore secondario, correggere il giocatore che tenta di creare un colore secondario a sua scelta, recarsi vicino a un cubo di un colore primario, esprimere perplessità, ballare, rimanere in attesa di comandi e tornare nella posizione iniziale alla fine di una interazione. Queste azioni sono rese possibili attraverso una macchina degli stati (Listato 5.1).

```

1 private enum State
2 {
3     Idle,
4     Idle_Moment,
5     Hello,
6     Happy,
7     Puzzled,
8     MoveTo,
9     ComposeCube1,
10    ComposeCube2,
11    ComposeCube3,
12    ComposeCube4,
13    ComposeCube5,
14    Reset,
15    Test,
16    Test1,
17    Test2,
18    Test3,
19    Test4,
20    Test5    }

```

Listing 5.1: Macchina degli stati

In particolare:

1. La creazione di un colore secondario è gestita dagli stati *ComposeCube1-5*
2. La correzione del giocatore avviene attraverso gli stati *Test1-5*, con ulteriori azioni in *ComposeCube1-5* in caso di errore in fase di composizione da parte del giocatore.
3. Per spostarsi verso un cubo di colore primario si utilizza lo stato *MoveTo*.
4. Per ballare è responsabile lo stato *Happy*.
5. Quando Dom è fermo o in attesa di un comando si utilizzano *Idle* e *Idle\_Moment*.
6. Quando riceve un comando che non rientra nel suo repertorio di azioni mostra perplessità attraverso lo stato *Puzzled*.
7. Infine quando completa un'azione mediante lo stato *Reset* torna nella posizione iniziale e attende un successivo comando.

Prima di entrare nel dettaglio dell'implementazione dei vari stati è utile fare una panoramica su tutte le funzioni utilizzate all'interno del progetto affinché si possa avere una maggiore comprensione.

### Funzioni e Coroutine utilizzate

*OnOrderGiven*: Questa funzione viene attivata quando l'utente ha finito di digitare il comando (prompt) nella casella di testo; tale cambiamento viene inviato ad una funzione (che vedremo nel Paragrafo 5.2.2) che calcola il coefficiente di similarità e determina l'azione più simile nella *actionList* di Dom (Listato 5.2).

```

1 public void OnOrderGiven (string prompt)
2 {
3     if(prompt != "")
4         ragionamentoDiDom.RankSimilarityScores(prompt, sentencesArray);
5 }

```

**Listing 5.2:** Funzione *OnOrderGiven*

*OnTestOrderGiven*: Simile alla precedente, ma per il prompt inserito nella modalità di test, ovvero i colori dei due cubi che Dom dovrà prendere (Listato 5.3).

```

1 public void OnTestOrderGiven (string prompt)
2 {
3     if (prompt != "")
4         ragionamentoNelTestDiDom.RankSimilarityScores(prompt, sentencesTestArray)
5     ;
6 }

```

**Listing 5.3:** Funzione *OnTestOrderGiven*

*Utility*: analizza i risultati prodotti dalla funzione invocata da *OnOrderGiven*, controlla il coefficiente di similarità e assegna lo stato estrapolato dal modello a Dom. Se il coefficiente è inferiore a 0.6, Dom entra in uno stato di confusione (*Puzzled*); altrimenti, assegna alla variabile *goalObject* il "noun" del prompt, il quale può essere o il colore primario verso cui Dom si deve dirigere oppure il colore secondario che si vuole comporre. Successivamente, attraverso una serie di "if", assegna alle variabili *color1* e *color2* i cubi dei colori primari necessari alla composizione di quel colore secondario(Listato 5.4).

```
1 public void Utility(float maxScore, int maxScoreIndex)
2 {
3     if (maxScore < 0.6f)
4     {
5         state = State.Puzzled;
6     }
7     else
8     {
9         // Ricava verb e noun (se presente) dal prompt e li assegna
10        // rispettivamente a state e goalObject
11        goalObject = GameObject.Find(actionsList[maxScoreIndex].noun);
12
13        string verb = actionsList[maxScoreIndex].verb;
14
15        state = (State)System.Enum.Parse(typeof(State), verb, true);
16
17        // identifica colori che compongono il colore secondario
18        if (goalObject == GameObject.Find("OrangeCube"))
19        {
20            color1 = GameObject.Find("RedCube");
21            color2 = GameObject.Find("YellowCube");
22        }
23        else if (goalObject == GameObject.Find("PurpleCube"))
24        {
25            color1 = GameObject.Find("BlueCube");
26            color2 = GameObject.Find("RedCube");
27        }
28        else if (goalObject == GameObject.Find("GreenCube"))
29        {
30            color1 = GameObject.Find("YellowCube");
31            color2 = GameObject.Find("BlueCube");
32        }
33    }
```

**Listing 5.4:** Funzione *Utility*

*TestUtility*: analizza i risultati prodotti dalla funzione invocata da *OnTestOrderGiven*, relativi ai due colori primari scelti dal giocatore che dovranno essere presi da Dom per la composizione;

questa volta per non entrare in stato di perplessità il coefficiente dovrà essere maggiore di 0.3. Oltre a decretare il valore delle variabili *color1* e *color2* nella stessa maniera di *Utility*, verranno assegnate le variabili *goalTestObject1* e *goalTestObject2*, le quali corrispondono ai cubi inseriti dall'utente, e *testCube* che rappresenta il cubo composto da essi (Listato 5.5).

```
1 public void TestUtility(float maxTestScore, int maxTestScoreIndex)
2 {
3     if (maxTestScore < 0.30f)
4     {
5         state = State.Puzzled;
6     }
7     else
8     {
9         goalTestObject1 = GameObject.Find(actionsTestList[maxTestScoreIndex].
cube1);
10        goalTestObject2 = GameObject.Find(actionsTestList[maxTestScoreIndex].
cube2);
11
12        if (goalObject == GameObject.Find("OrangeCube"))
13        {
14            color1 = GameObject.Find("RedCube");
15            color2 = GameObject.Find("YellowCube");
16        }
17        else if (goalObject == GameObject.Find("PurpleCube"))
18        {
19            color1 = GameObject.Find("BlueCube");
20            color2 = GameObject.Find("RedCube");
21        }
22        else if (goalObject == GameObject.Find("GreenCube"))
23        {
24            color1 = GameObject.Find("YellowCube");
25            color2 = GameObject.Find("BlueCube");
26        }
27        // Assegnazione di goalTestObject1, goalTestObject2 e testCube
28        if ((goalTestObject1 == GameObject.Find("RedCube") && goalTestObject2 ==
GameObject.Find("YellowCube")) || (goalTestObject2 == GameObject.Find("
RedCube") && goalTestObject1 == GameObject.Find("YellowCube")))
29        {
30            testCube = GameObject.Find("OrangeCube");
31        }
32
33        else if ((goalTestObject1 == GameObject.Find("RedCube") &&
goalTestObject2 == GameObject.Find("BlueCube")) || (goalTestObject2 ==
GameObject.Find("RedCube") && goalTestObject1 == GameObject.Find("BlueCube"))
)
34        {
35            testCube = GameObject.Find("PurpleCube");
36        }
37    }
```



```

38     else if ((goalTestObject1 == GameObject.Find("YellowCube") &&
goalTestObject2 == GameObject.Find("BlueCube")) || (goalTestObject2 ==
GameObject.Find("YellowCube") && goalTestObject1 == GameObject.Find("BlueCube
")))
39     {
40         testCube = GameObject.Find("GreenCube");
41     }
42     state = State.Test1;
43
44 }
45 }

```

Listing 5.5: Funzione *TestUtility*

*GrabCreate*: questa funzione ha il compito di creare un clone di un cubo di colore primario e permettere a Dom di prenderlo. Il clone viene generato nella stessa posizione dell'originale. Questa operazione permette di mantenere il cubo originario invariato nella scena, modificandone soltanto l'altezza di 10 unità in modo tale che possa ricadere nella sua posizione generando un effetto "rimpiazzo". Il clone viene impostato come figlio della posizione di presa (*grabPosition*) per facilitarne il trasporto (Listato 5.6).

```

1 void GrabCreate(GameObject gameObject)
2 {
3     Vector3 position = gameObject.transform.position;
4     Quaternion rotation = gameObject.transform.rotation;
5     clone = Instantiate(gameObject, position, rotation);
6     Vector3 fallPosition = gameObject.transform.position;
7     fallPosition.y += 10.0f;
8     gameObject.transform.position = fallPosition;
9
10    // Imposta il gameObject come figlio di grabPosition
11    clone.transform.parent = grabPosition.transform;
12
13    // Per evitare bug, imposta la velocita del clone e la velocita angolare a 0
14    Rigidbody cloneRigidbody = clone.GetComponent<Rigidbody>();
15
16    // Per evitare bug, imposta la velocita dell'oggetto e la velocita angolare a
17    0
18    cloneRigidbody.isKinematic = true;
19    cloneRigidbody.velocity = Vector3.zero;
20    cloneRigidbody.angularVelocity = Vector3.zero;
21
22    // Imposta la posizione del clone del gameObject a grabPosition
23    clone.transform.position = grabPosition.transform.position;
24 }

```

Listing 5.6: Funzione *GrabCreate*

*Drop*: consente a Dom di rilasciare un oggetto che sta trasportando, in particolare il clone di un cubo di colore primario. La funzione disabilita la modalità *isKinematic*, la quale era stata abilitata durante il trasporto, permettendo all'oggetto di interagire con l'ambiente fisico (Listato 5.7).

```
1 void Drop(GameObject gameObject)
2 {
3     gameObject.GetComponent<Rigidbody>().isKinematic = false;
4     gameObject.transform.SetParent(null);
5 }
```

**Listing 5.7:** Funzione *Drop*

*DropSecondary*: fa apparire in scena il colore secondario appena composto; viene utilizzata sia per il colore secondario composto da Dom sia per quello composto dal giocatore, facendolo cadere dall'alto. Questo crea un effetto visivo che mette in risalto la composizione del nuovo colore (Listato 5.8).

```
1 void DropSecondary(GameObject gameObject)
2 {
3     // Setta la posizione in cui il clone del cubo deve cadere
4     Vector3 fallPositionCompose0 = new Vector3(0.0f, 14.0f, 5.5f);
5     Quaternion rotation = gameObject.transform.rotation;
6     cloneCompose = Instantiate(gameObject, fallPositionCompose0, rotation);
7     Rigidbody cloneComposeRigidbody = cloneCompose.GetComponent<Rigidbody>();
8     // Attiva gravita per farlo cadere
9     cloneComposeRigidbody.useGravity = true;
10 }
```

**Listing 5.8:** Funzione *DropSecondary*

*Coroutine DestroyClones*: programma la distruzione dei cloni dei cubi primari dopo un certo lasso di tempo, pulendo la zona della scena in cui cadrà il colore secondario appena composto (Listato 5.9).

```
1 IEnumerator DestroyClones(GameObject gameObjectToDestroy, float delay)
2 {
3     yield return new WaitForSeconds(delay);
4     Destroy(GameObject.Find("YellowCube(Clone)"));
5     Destroy(GameObject.Find("RedCube(Clone)"));
6     Destroy(GameObject.Find("BlueCube(Clone)"));
7
8 }
```

**Listing 5.9:** Coroutine *DestroyClones*

*Coroutine Attend*: dopo un periodo di tempo, elimina il clone del cubo secondario composto erroneamente dall'utente e mostra la composizione corretta del colore secondario, impostando lo stato di Dom a *ComposeCube1* (Listato 5.10).

```
1 IEnumerator Attend(float delay)
2 {
3     yield return new WaitForSeconds(delay);
4     state = State.ComposeCube1;
5     Destroy(cloneCompose);
6 }
```

**Listing 5.10:** Coroutine *Attend*

*Coroutine ResetScene*: resetta tutte le variabili dopo che un'azione è stata completata e imposta lo stato di Dom a *Reset* (Listato 5.11).

```
1 IEnumerator ResetScene(float delay)
2 {
3     yield return new WaitForSeconds(delay);
4     Destroy(cloneCompose);
5
6     goalObject = null;
7     color1 = null;
8     color2 = null;
9     goalTestObject1 = null;
10    goalTestObject2 = null;
11    testCube = null;
12
13    anim.SetBool("backflip", true);
14    state = State.Reset;
15
16 }
```

**Listing 5.11:** Coroutine *ResetScene*

### Macchina degli stati

Dopo aver delineato le funzioni utilizzate nei vari stati, siamo ora pronti a procedere con l'esame dettagliato di come sono state implementate le sette azioni che Dom è in grado di eseguire.

1. *Creazione di un colore secondario*: nello stato *ComposeCube1* Dom si dirige verso la posizione di *color1* ovvero il primo colore primario utile per la composizione del colore secondario richiesto, quando la sua distanza dal cubo è inferiore a *reachedObjectPositionDistance* (1,5 unità) viene afferrato il cubo mediante la funzione *GrabbCreate* e viene impostato lo stato *ComposeCube2*. In questo stato Dom si dirige verso la *playerPosition*, ovvero la posizione dell'utente (di fronte alla camera che riprende la scena del gioco),

una volta che la sua distanza è minore di *reachedObjectPositionDistance* dalla *playerPosition*, il cubo viene rilasciato mediante la funzione *Drop* e viene impostato lo stato *ComposeCube3*; qui e in *ComposeCube4* si ripeterà lo stesso procedimento per il secondo cubo di colore primario, ovvero *color2* (Listato 5.12).

```
1 case State.ComposeCube1:
2     agent.SetDestination(color1.transform.position);
3
4     if (Vector3.Distance(transform.position, color1.transform.position) <
5         reachedObjectPositionDistance)
6     {
7         GrabCreate(color1);
8         state = State.ComposeCube2;
9     }
10    break;
11
12 case State.ComposeCube2:
13     agent.SetDestination(playerPosition.transform.position);
14     if (Vector3.Distance(transform.position, playerPosition.transform.
15         position) < reachedObjectPositionDistance)
16     {
17         Drop(clone);
18         state = State.ComposeCube3;
19     }
20    break;
21
22 case State.ComposeCube3:
23     agent.SetDestination(color2.transform.position);
24
25     if (Vector3.Distance(transform.position, color2.transform.position) <
26         reachedObjectPositionDistance)
27     {
28         GrabCreate(color2);
29         state = State.ComposeCube4;
30     }
31    break;
32
33 case State.ComposeCube4:
34     agent.SetDestination(playerPosition.transform.position);
35     if (Vector3.Distance(transform.position, playerPosition.transform.
36         position) < reachedObjectPositionDistance)
37     {
38         Drop(clone);
39         state = State.ComposeCube5;
40     }
41    break;
```

Listing 5.12: Stati *ComposeCube1-4*

Nello stato *ComposeCube5*, Dom procede verso la *dancePosition* per eseguire un balletto. Parallelamente, la funzione *DropSecondary* fa apparire il cubo di colore secondario, facendolo cadere dall'alto nella posizione in cui erano posizionati i cubi primari che, a loro volta, vengono rimossi dalla scena tramite la coroutine *DestroyClones*. Al termine di questa sequenza, viene eseguita la coroutine *ResetScene* per riportare Dom alla sua posizione iniziale e prepararlo per ricevere nuovi comandi (Listato 5.13).

```
1 case State.ComposeCube5:
2     agent.SetDestination(dancePosition.transform.position);
3
4     if (Vector3.Distance(transform.position, dancePosition.transform.
5 position) < reachedPositionDistance)
6     {
7         anim.SetBool("happy", true);
8         state = State.Idle\_Moment;
9
10        DropSecondary(goalObject);
11
12        StartCoroutine(DestroyClones(GameObject.Find("YellowCube(Clone)"), 1.5f)
13 );
14        StartCoroutine(DestroyClones(GameObject.Find("RedCube(Clone)"), 1.5f));
15        StartCoroutine(DestroyClones(GameObject.Find("BlueCube(Clone)"), 1.5f));
16
17        StartCoroutine(ResetScene(5.0f));
18
19        // Se testCube non e' nullo vuol dire che Dom sta componendo il colore
20        // per mostrare la soluzione corretta dopo che l'utente ha sbagliato
21        // Questa condizione permette di controllare cosa deve dire Dom se:
22        // "Eccolo! Questa la corretta composizione per il colore da te
23        // scelto!" come nel caso dell'azione (2) oppure
24        // "Amo comporre i colori" come nel caso dell'azione (1) dove gli si
25        // chiede solamente di comporre un colore.
26        if (testCube != null)
27        {
28            AudioSourceEccolo = GameObject.Find("Eccolo").GetComponent<
29 AudioSource>();
30            AudioSourceEccolo.Play();
31        }
32        else
33        {
34            AudioSourceAmoComporre = GameObject.Find("AmoComporre").GetComponent<
35 AudioSource>();
36            AudioSourceAmoComporre.Play();
37        }
38    }
39    break;
```

Listing 5.13: Stato *ComposeCube5*

2. *Correzione del giocatore*: in *Test* viene mostrata nello schermo la casella di testo che permette di specificare all'utente i due colori primari che dovrà prendere Dom; una volta premuto invio, verrà impostato lo stato *Test1*. Da questo stato fino a *Test4* il comportamento di Dom è pressochè identico alla sequenza di stati *ComposeCube1-4* con la differenza che le variabili dei cubi primari sono *goalTestObject1* e *goalTestObject2* (Listato 5.14).

```
1 case State.Test:
2     testInputField.gameObject.SetActive(true);
3     state = State.Test1;
4     break;
5
6 case State.Test1:
7     if (goalTestObject1 != null)
8     {
9         agent.SetDestination(goalTestObject1.transform.position);
10
11         if (Vector3.Distance(transform.position, goalTestObject1.transform.
12             position) < reachedObjectPositionDistance)
13         {
14             GrabCreate(goalTestObject1);
15             state = State.Test2;
16         }
17     }
18     break;
19
20 case State.Test2:
21     agent.SetDestination(playerPosition.transform.position);
22     if (Vector3.Distance(transform.position, playerPosition.transform.
23         position) < reachedObjectPositionDistance)
24     {
25         Drop(clone);
26         state = State.Test3;
27     }
28     break;
29
30 case State.Test3:
31     agent.SetDestination(goalTestObject2.transform.position);
32
33     if (Vector3.Distance(transform.position, goalTestObject2.transform.
34         position) < reachedObjectPositionDistance)
35     {
36         GrabCreate(goalTestObject2);
37         state = State.Test4;
38     }
39     break;
40
41 case State.Test4:
42     agent.SetDestination(playerPosition.transform.position);
```

```

40     if (Vector3.Distance(transform.position, playerPosition.transform.
position) < reachedObjectPositionDistance)
41     {
42         Drop(clone);
43         state = State.Test5;
44     }
45     }
46     break;

```

Listing 5.14: Stati *Test* e *Test1-4*

Nello stato *Test5*, Dom si dirige verso la *dancePosition*, dove possono verificarsi due scenari a seconda delle azioni del giocatore. Se il giocatore ha selezionato correttamente i cubi che compongono il colore secondario precedentemente scelto, Dom esprimerà apprezzamento attraverso un balletto e una frase di elogio, per poi ritornare alla sua posizione iniziale tramite la funzione *ResetScene*, prontamente disponibile per ricevere ulteriori comandi. In caso contrario, se la scelta del giocatore non è corretta, Dom, in maniera pacata, evidenzierà l'errore con una specifica animazione e un commento. Viene, poi, impostato lo stato su *ComposeCube1* per mostrare la corretta procedura di composizione del colore desiderato. Al completamento di questa dimostrazione, nel momento in cui lo stato raggiunge *ComposeCube5* (e *testCube* risulta diverso da *null* siccome la variabile assume valore appena si entra in stato di *Test*), Dom non pronuncerà la solita frase "Amo comporre i colori!", ma opererà per "Eccolo! Questa è la corretta composizione per il colore da te scelto!", per guidare l'utente verso la corretta comprensione della teoria dei colori (Listato 5.15).

```

1 case State.Test5:
2     agent.SetDestination(dancePosition.transform.position);
3
4     if (Vector3.Distance(transform.position, dancePosition.transform.
position) < reachedPositionDistance)
5     {
6         if(testCube == goalObject)
7         {
8             anim.SetBool("happy", true);
9             audioSourceBenFatto = GameObject.Find("BenFatto").GetComponent<
AudioSource>();
10            audioSourceBenFatto.Play();
11            state = State.Idle\Moment;
12
13            DropSecondary(goalObject);
14
15            StartCoroutine(DestroyClones(GameObject.Find("YellowCube(Clone) "
), 1.5f));
16            StartCoroutine(DestroyClones(GameObject.Find("RedCube(Clone) "
), 1.5f));

```

```

17         StartCoroutine(DestroyClones(GameObject.Find("BlueCube(Clone)"),
18             1.5f));
19
20         StartCoroutine(ResetScene(5.0f));
21     }
22     else
23     {
24         anim.SetBool("no", true);
25         audioSourceNo = GameObject.Find("No").GetComponent<AudioSource
26 >();
27         audioSourceNo.Play();
28         state = State.Idle\_Moment;
29
30         DropSecondary(testCube);
31
32         StartCoroutine(DestroyClones(GameObject.Find("YellowCube(Clone) "
33             ), 1.5f));
34         StartCoroutine(DestroyClones(GameObject.Find("RedCube(Clone) "
35             ), 1.5f));
36         StartCoroutine(DestroyClones(GameObject.Find("BlueCube(Clone) "
37             ), 1.5f));
38
39         StartCoroutine(Attend(4.0f)); //show the right way to compose
40         the secondary color and destroy the clone of the wrong cube
41     }
42 }
43 break;

```

Listing 5.15: Stato *Test5*

3. *Spostarsi verso un cubo di colore primario*: in questo semplice stato viene impostata come destinazione di Dom la variabile *goalObject* che, ovviamente, in questo caso corrisponde al colore primario, inserito nel prompt, a cui l'utente vuole che Dom si avvicini. Una volta giunto in prossimità si imposta lo stato *Idle\_Moment* e si attiva la coroutine *ResetScene* (Listato 5.16).

```

1 case State.MoveTo:
2     agent.SetDestination(goalObject.transform.position);
3     if (Vector3.Distance(transform.position, goalObject.transform.position)
4 < reachedPositionDistance)
5     {
6         state = State.Idle\_Moment;
7         StartCoroutine(ResetScene(5.0f));
8     }
9     break;

```

Listing 5.16: Stato *MoveTo*



4. *Ballare*: Dom balla in più occasioni all'interno del gioco. Sia quando gli è richiesto da prompt sia quando compone un colore. Lo stato *Happy* si occupa di far ballare Dom quando è l'utente a volerlo; viene impostata la posizione *playerPosition* e successivamente viene lanciata l'animazione (Listato 5.17).

```

1 case State.Happy:
2     agent.SetDestination(playerPosition.position);
3     if (Vector3.Distance(transform.position, playerPosition.position) <
4         reachedPositionDistance)
5     {
6         RotateTo();
7         anim.SetBool("happy", true);
8         state = State.Idle;
9     }
10    break;

```

**Listing 5.17:** Stato *Happy*

5. *Dom è perplesso*: quando la funzione *Utility* riceve il coefficiente di similarità della frase inserita dall'utente controlla che esso sia inferiore di 0.6. Se la condizione è rispettata allora viene immediatamente impostato lo stato *Puzzled* (Perplesso) dove, analogamente a ciò che avviene per lo stato *Happy*, Dom raggiunge la *playerPosition* e comunica di non poter eseguire il comando, sia verbalmente sia mediante un'animazione di incertezza (Listato 5.18).

```

1 case State.Puzzled:
2     agent.SetDestination(playerPosition.position);
3     if (Vector3.Distance(transform.position, playerPosition.position) <
4         reachedPositionDistance)
5     {
6         RotateTo();
7         anim.SetBool("puzzled", true);
8         AudioSourceNonPossoFarlo = GameObject.Find("NonPossoFarlo").
9         GetComponent<AudioSource>();
10        AudioSourceNonPossoFarlo.Play();
11        state = State.Idle;
12    }
13    break;

```

**Listing 5.18:** Stato *Puzzled*

6. *Nullafacente*: nel comportamento programmato di Dom, l'utilizzo di due distinti stati, *Idle* e *Idle\_Moment*, risponde alla necessità di distinguere momenti differenti di inattività. Lo stato *Idle* viene attivato quando Dom ha completato un'azione e si trova in attesa di nuovi comandi da parte dell'utente; in questo scenario, nello schermo viene visualizzata una casella di testo per l'inserimento del prompt. Tuttavia, vi sono circostanze in cui

Dom necessita di rimanere fermo senza che ciò implichi la fine di un'azione o l'attesa di nuove istruzioni. Un esempio chiaro si verifica nella transizione tra *ComposeCube4* e *ComposeCube5*, durante la quale Dom attende la caduta del cubo che rappresenta il colore secondario appena composto. In questi frangenti, il robot si trova nello stato *Idle\_Moment*, che, a differenza dello stato *Idle*, non prevede la comparsa della casella di testo per il prompt, poiché non si trova in attesa di ulteriori comandi. Questa distinzione consente di gestire con maggiore precisione il flusso di gioco, assicurando che la casella di testo per l'inserimento dei comandi appaia soltanto quando Dom è effettivamente disponibile a ricevere nuove azioni da eseguire, come ad esempio dopo la conclusione di una sequenza di azioni, segnalata dallo stato *Idle* impostato dalla coroutine *ResetScene* (Listato 5.19).

```
1 if (state == State.Idle)
2 {
3     inputField.gameObject.SetActive(true);
4 }
5 else
6 {
7     inputField.gameObject.SetActive(false);
8 }
9 case State.Idle:
10     RotateTo(); // Ruota Dom verso la telecamera
11     break;
12
13 case State.Idle\_Moment:
14     RotateTo(); // Ruota Dom verso la telecamera
15     break;
```

**Listing 5.19:** Stati *Idle* e *Idle\_Moment*

7. *Tornare in posizione*: la coroutine *ResetScene* setta lo stato di Dom *Reset* in modo tale che ritorni nella posizione centrale, *playerposition*, eseguendo un salto all'indietro e successivamente assegnando a Dom lo stato *Idle* rendendolo pronto a nuove interazioni (Listato 5.20).

```
1 case State.Reset:
2     agent.SetDestination(playerPosition.transform.position);
3     if (Vector3.Distance(transform.position, playerPosition.transform.
4         position) < reachedPositionDistance)
5     {
6         anim.SetBool("backflip", true);
7         state = State.Idle;
8     }
9     break;
```

**Listing 5.20:** Stato *Reset*

### 5.2.2 Implementazione API di Hugging Face

Per integrare le API di Hugging Face (HF) in Unity, abbiamo sviluppato la classe *RagionamentoDiDom*, che funge da ponte tra l'input dell'utente e il modello di sentence similarity di HF. Questa classe gestisce la chiamata all'API, elaborando la frase inserita dall'utente e confrontandola con un set di frasi candidate, al fine di determinare l'azione più adeguata da intraprendere.

Il processo di integrazione si avvale del package API Wizard, che semplifica la creazione e la gestione delle chiamate API all'interno dell'ambiente di sviluppo Unity. Grazie a questo strumento, siamo stati in grado di concentrarci sulla logica applicativa, delegando al package la gestione delle specifiche tecniche legate alla comunicazione con le API di Hugging Face.

La classe *RagionamentoDiDom* possiede tre variabili nascoste ([HideInInspector]):

- *source\_sentence*: rappresenta il testo inserito dall'utente;
- *maxScore*: conserva il valore di similarità più alto tra la frase dell'utente e le frasi candidate;
- *maxScoreIndex*: indica l'indice della frase candidata che ha ottenuto il punteggio più alto.

Inoltre, è presente un riferimento alla classe *ComportamentoDiDom*, che consente di comunicare il risultato della valutazione al sistema di controllo del comportamento di Dom (Listato 5.21).

```
1 public class RagionamentoDiDom: MonoBehaviour
2 {
3     [HideInInspector]
4     public string source_sentence;
5
6     [HideInInspector]
7     public float maxScore;
8
9     [HideInInspector]
10    public int maxScoreIndex;
11
12    public ComportamentoDiDom comportamentoDiDom;
13
14    private SentenceSimilarity sentenceSimilarity = new SentenceSimilarity();
```

**Listing 5.21:** Variabili della classe *RagionamentoDiDom*

La funzione principale, *RankSimilarityScores*, accetta come parametri la frase inserita dall'utente (*source\_sentence*) e un array di frasi candidate (*sentences*). Questa funzione si occupa di invocare il metodo *SentenceSimilarity* della classe *HuggingFaceAPI*, passando la frase dell'utente e le frasi candidate. Il risultato viene poi elaborato per identificare il massimo valore

di similarità (*maxScore*) e l'indice corrispondente (*maxScoreIndex*), che vengono utilizzati per determinare l'azione più pertinente da eseguire (Listato 5.22).

```
1 public (float maxScore, int maxScoreIndex) RankSimilarityScores(string
   source_sentence, string[] sentences)
2 {
3     HuggingFaceAPI.SentenceSimilarity(source_sentence, results =>
4     {
5         maxScore = sentenceSimilarity.FindBestSimilarityScoreValue(results);
6         maxScoreIndex = sentenceSimilarity.FindBestSimilarityScoreIndex(results);
7         comportamentoDiDom.Utility(maxScore, maxScoreIndex);
8     },
9     error =>
10    {
11        Debug.LogError(error);
12    },
13    sentences);
14    return (maxScore, maxScoreIndex);
15 }
```

**Listing 5.22:** Funzione RankSimilarityScores

La chiamata all'API di Hugging Face avviene attraverso una richiesta asincrona, gestendo sia il caso di successo, che aggiorna lo stato e le azioni di Dom attraverso la funzione *Utility*, sia il caso di errore, che viene segnalato tramite il log degli errori di Unity.

### 5.3 Applicazioni pratiche e guida utente

Per interagire con Dom nel gioco, l'utente deve semplicemente inserire i comandi nella casella di testo dedicata. Dom è programmato per comprendere e reagire a una varietà di frasi che indicano ad esso di eseguire specifiche azioni. I prompt che permettono a Dom di avere una risposta congrua con l'input sono tutte le frasi *simili* a quelle del seguente elenco:

- "*Compose the green cube*"
- "*I try to create the orange cube correct me*"
- "*Move to blue cube*"
- "*Dance*"
- "*Hello*"

Nella seguente sezione, presenteremo vari casi d'uso del gioco attraverso sequenze di frame catturati durante il gameplay. Questo approccio mira a fornire una comprensione visiva e dettagliata di tutte le interazioni possibili tra l'utente e il gioco, evidenziando come Dom, il robot protagonista, risponde ai diversi comandi e situazioni, dimostrando così l'efficacia e la versatilità del sistema di controllo e della macchina a stati di Dom. Partiremo dalle dinamiche più semplici per arrivare alle più complesse.

### Salutare

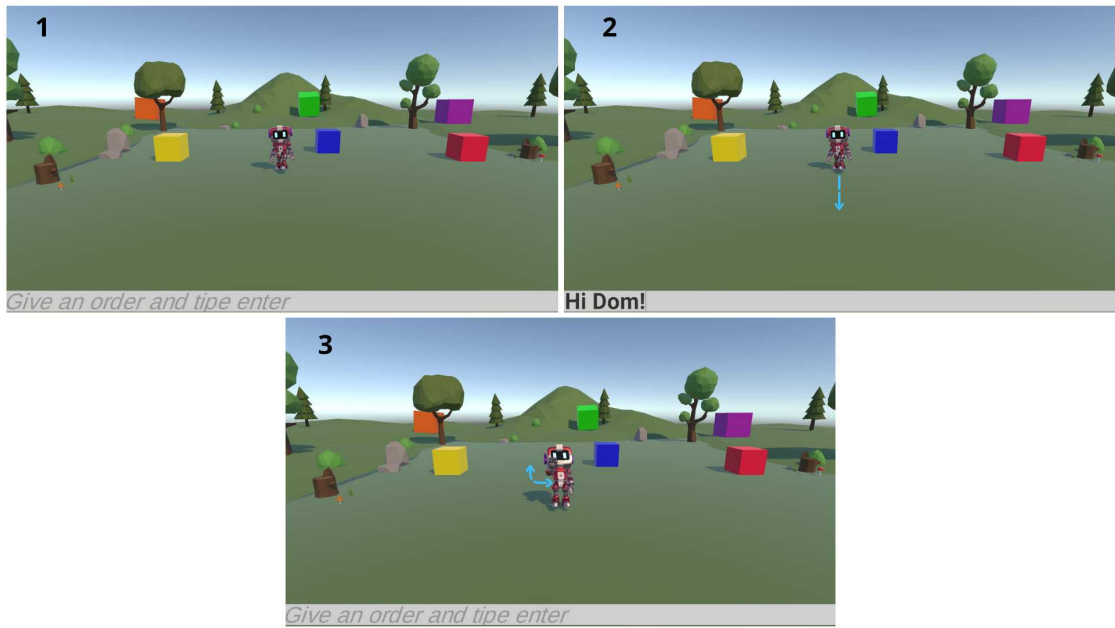


Figura 5.5: Caso d'uso: salutare

### Ballare

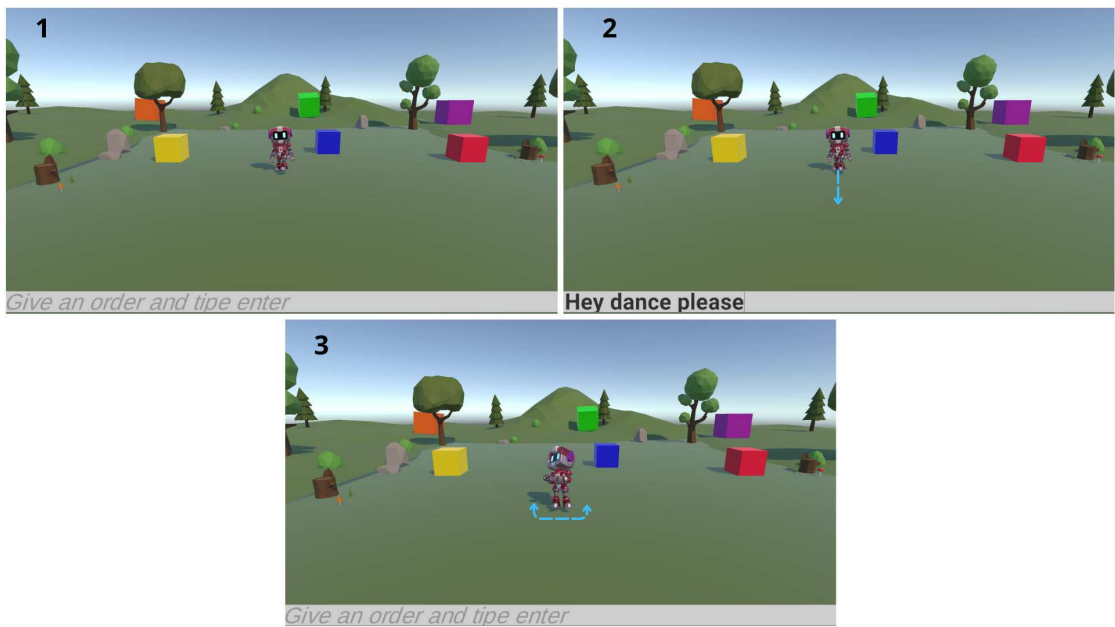


Figura 5.6: Caso d'uso: ballare

### Perplexità

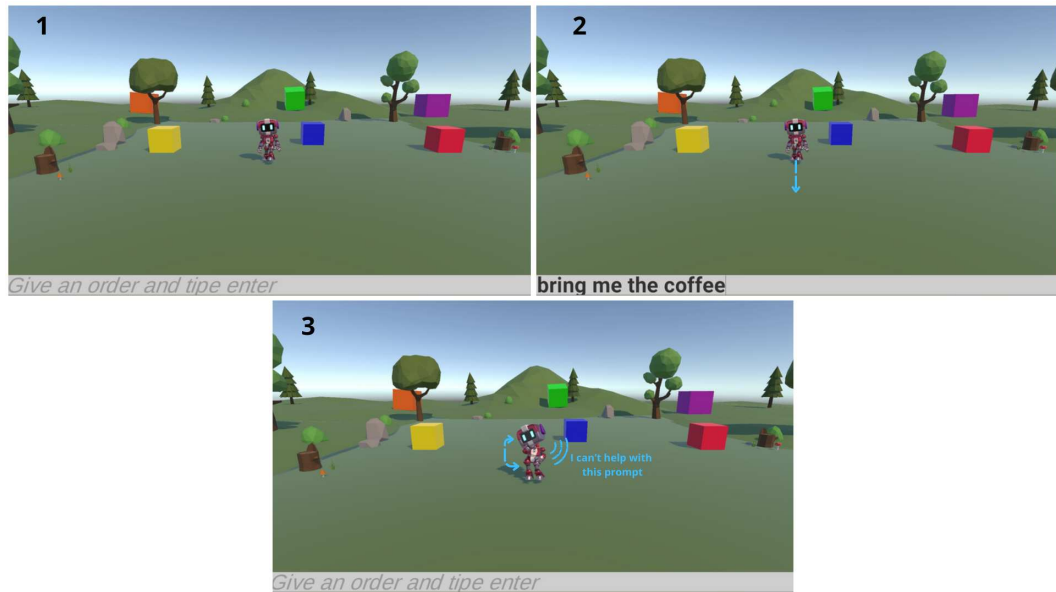


Figura 5.7: Caso d'uso: perplessità

### Recarsi presso un cubo

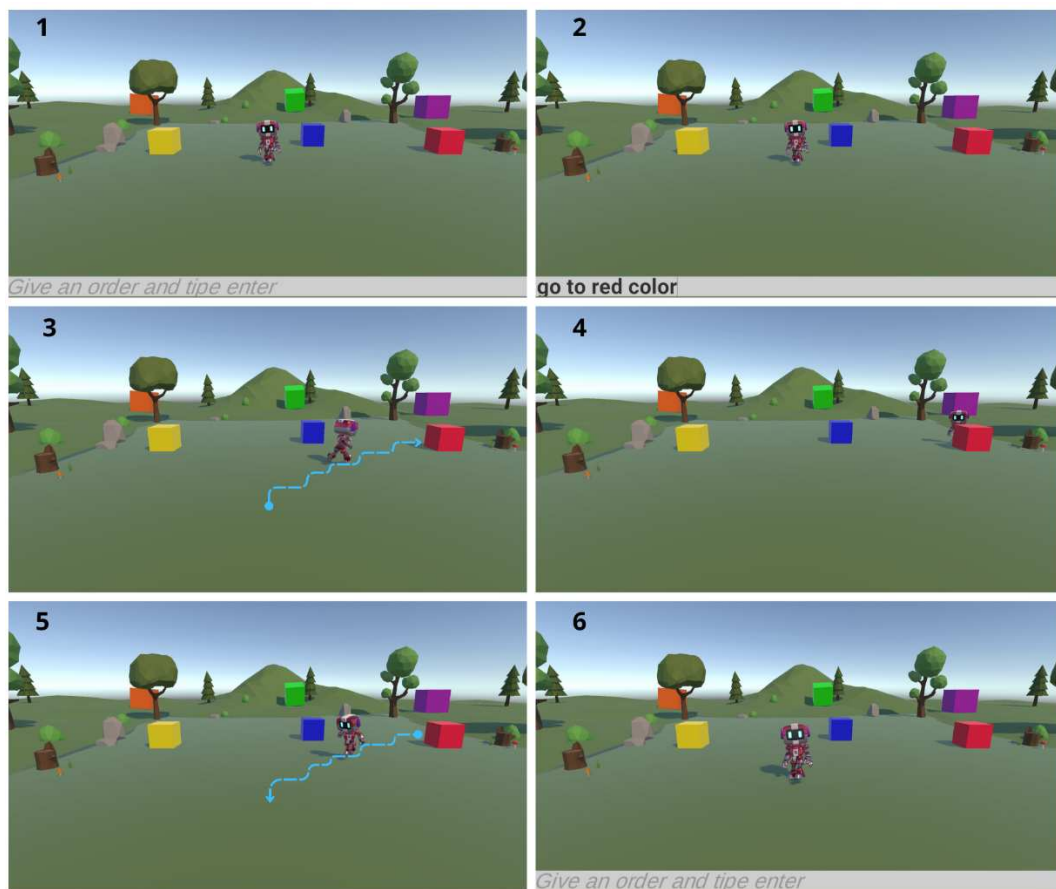


Figura 5.8: Caso d'uso: recarsi presso un cubo

### Comporre un colore

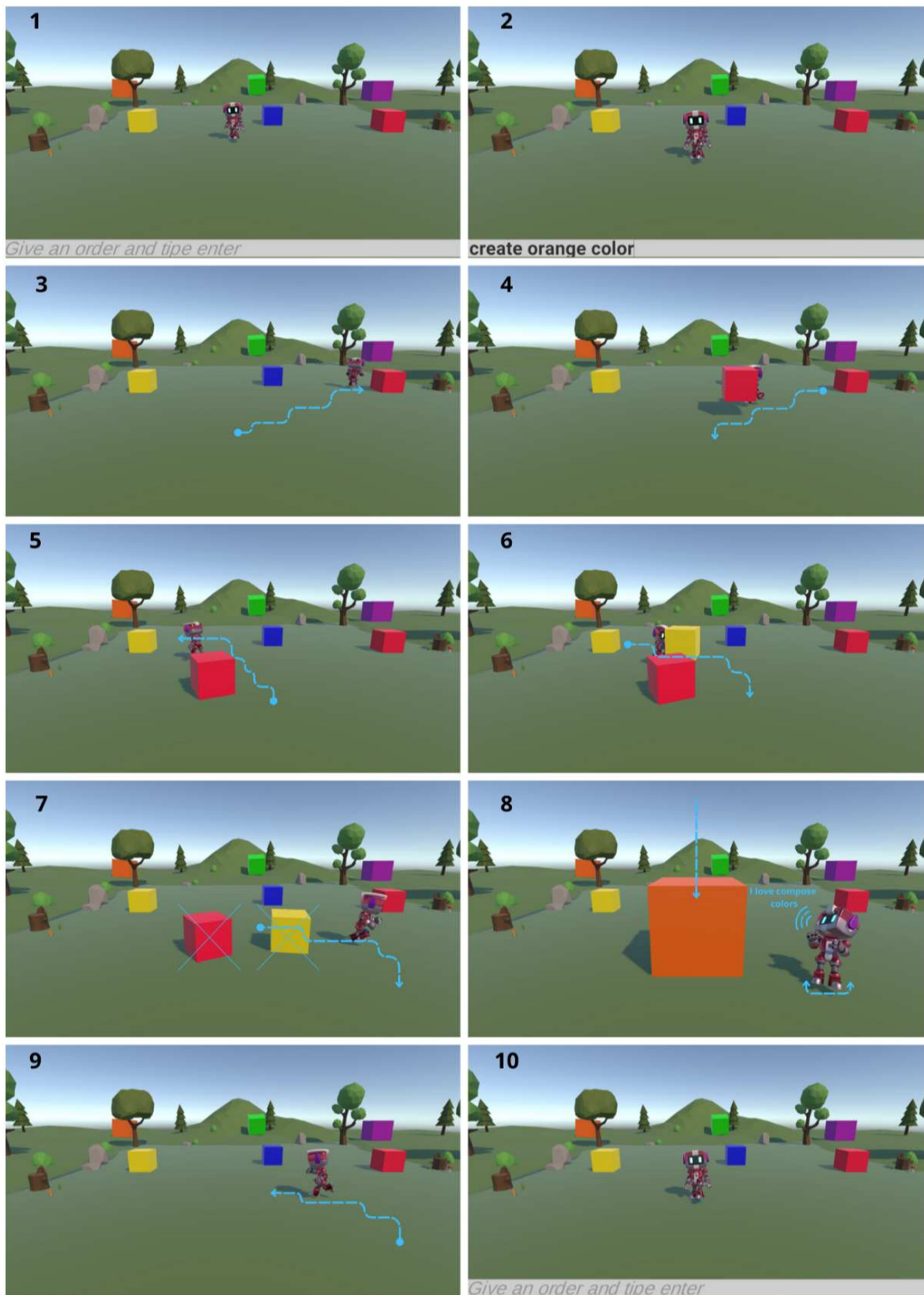
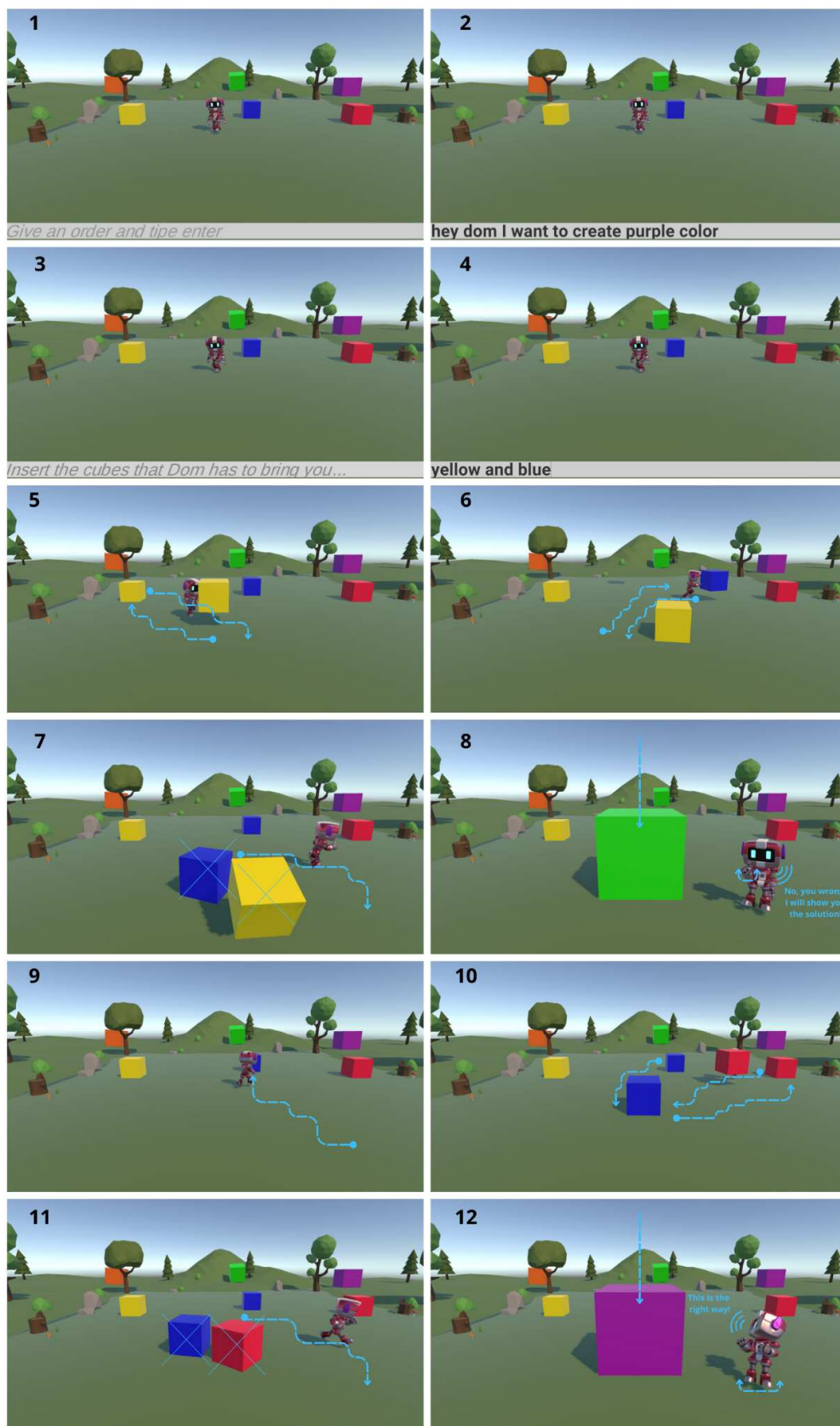


Figura 5.9: Caso d'uso: comporre un colore



### Correggere utente che sbaglia



**Figura 5.10:** Caso d'uso: correggere utente che sbaglia



Complimentarsi con l'utente che esegue la composizione correttamente

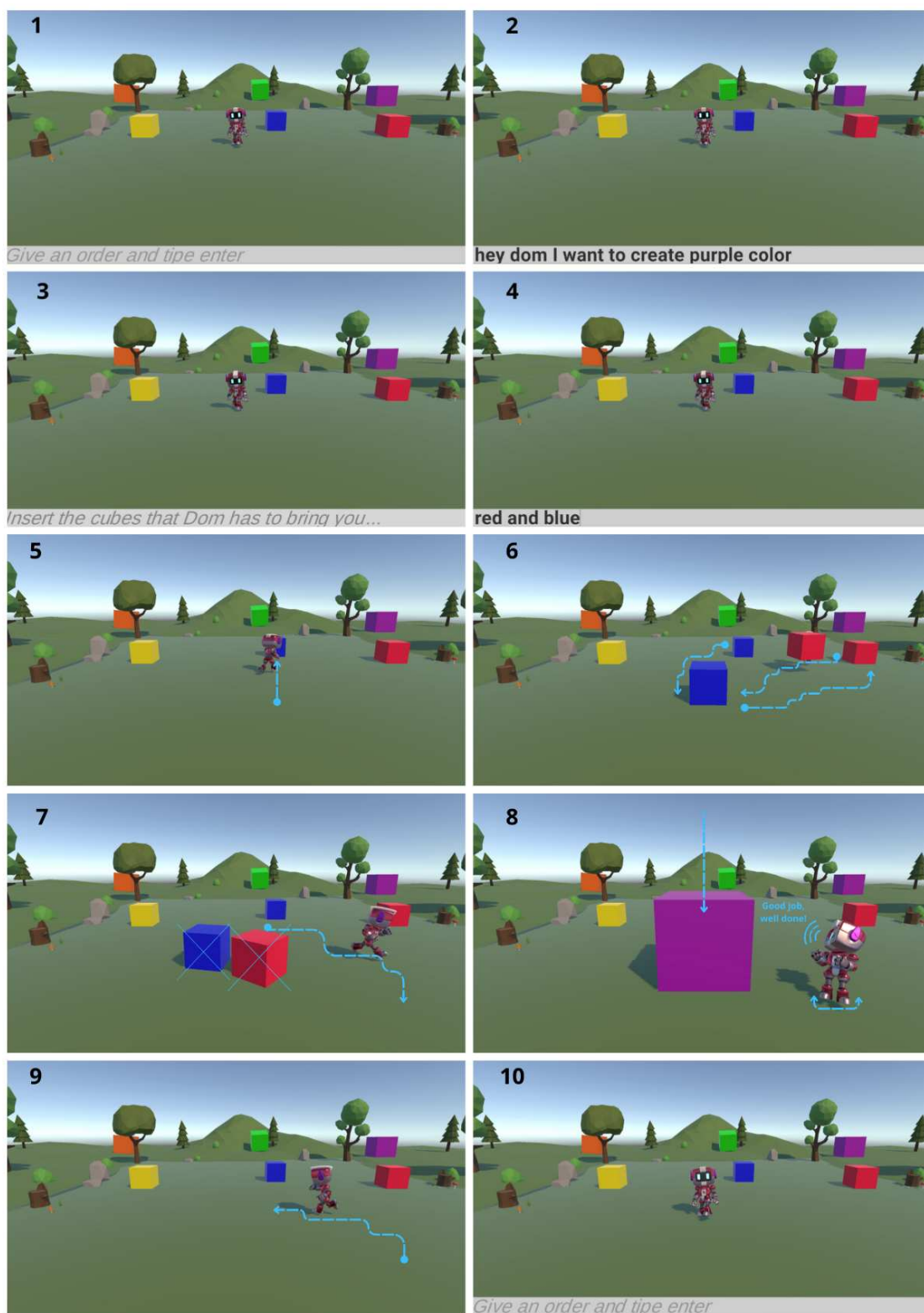


Figura 5.11: Caso d'uso: complimentarsi con l'utente che esegue la composizione correttamente

---

## Analisi e Riflessioni su "Dom the Robot"

---

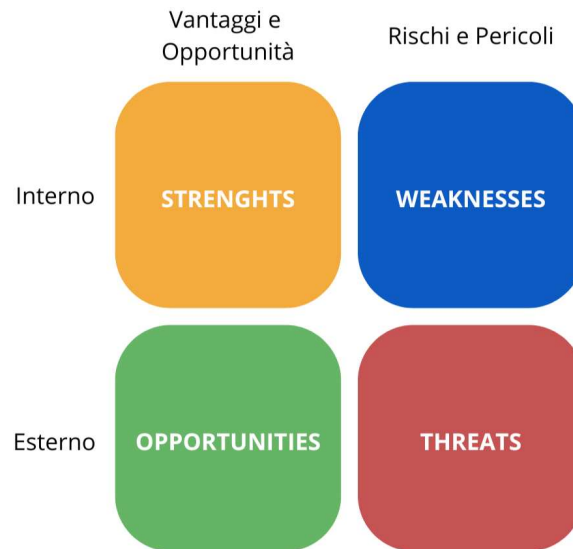
*In questo capitolo conclusivo, ci immergiamo nell'analisi SWOT di "Dom the Robot", il culmine di uno splendido viaggio che ha intrecciato passione, innovazione e apprendimento. Riflettendo sul percorso intrapreso, emergono i contorni di un'avventura che ha superato le aspettative, arricchita da sfide, scoperte e momenti di spensierata creazione. L'emozione di aver portato a termine questo progetto non è solo un segno di realizzazione personale ma un invito a guardare avanti, ispirati dalle lezioni apprese e dalla promessa di future esplorazioni. Questo capitolo non solo chiude una fase significativa ma apre le porte a infinite possibilità, celebrando il potere dell'innovazione e l'inesauribile curiosità che ci guida.*

### 6.1 SWOT Analysis di "Dom the Robot"

L'analisi SWOT è un metodo strutturato per valutare le caratteristiche interne ed esterne che influenzano un'entità. Originariamente concepita per l'analisi dei business plan aziendali, la sua versatilità ha permesso un'applicazione più ampia. Utilizzata tipicamente prima di decisioni critiche, facilita la scelta strategica evidenziando gli aspetti cruciali da considerare. Questo approccio categorizza i fattori in:

- "S" - *Strengths*, punti di forza;
- "W" - *Weaknesses*, punti di debolezza;
- "O" - *Opportunities*, opportunità;
- "T" - *Threats*, minacce.

La matrice SWOT è una matrice 2x2 che ha come colonne "Vantaggi e Opportunità" e "Rischi e Pericoli" le quali vanno ad incrociare le righe "Interno" ed "Esterno" (Figura 6.1). Forze e debolezze sono attributi interni, legati direttamente all'entità in esame, mentre opportunità e minacce derivano da elementi esterni, al di fuori del nostro controllo diretto. I punti di forza e le opportunità costituiscono gli aspetti positivi, indicati come "Vantaggi e



**Figura 6.1:** Matrice SWOT

Opportunità", al contrario, debolezze e minacce segnalano gli elementi negativi, categorizzati sotto "Rischi e Pericoli".

Il processo dell'analisi SWOT comporta il riempimento di ciascun campo della matrice con gli elementi corrispondenti a forze, debolezze, opportunità e minacce. Completata la matrice, questa analisi preliminare guida la presa di decisioni strategiche significative, aiutando a delineare il percorso futuro.

### 6.1.1 Forze (Strengths)

Il cavallo di battaglia di "Dom the Robot" è sicuramente la modalità d'impiego dell'AI. La capacità di integrare tecnologie AI, accessibili grazie a risorse open source come quelle di Hugging Face, consente a "Dom the Robot" di offrire un'esperienza educativa personalizzata. Questo serious game sfrutta un modello di AI basato sulla somiglianza delle frasi per facilitare un'interazione dinamica e naturale tra il giocatore e il robot, superando le limitazioni dei tradizionali sistemi basati su comandi preimpostati.

L'approccio intuitivo promosso da "Dom the Robot" migliora l'engagement e l'efficacia dell'apprendimento, cercando di essere un esempio di integrazione dell'AI nel settore educativo.

### 6.1.2 Debolezze (Weaknesses)

Una delle principali criticità associate a "Dom the Robot" riguarda la modalità di interazione prevista per il suo pubblico di riferimento, bambini di età compresa tra i 3 e i 5 anni. L'uso di una tastiera per inserire prompt di testo può rappresentare un ostacolo significativo, considerando le limitate competenze di scrittura dei bambini in questa fascia di età. Tale barriera alla partecipazione potrebbe compromettere l'engagement e la fruibilità del serious game, limitando di fatto il raggiungimento degli obiettivi educativi prefissati.

L'introduzione di una tecnologia di trascrizione vocale avanzata, come il modello *whisper-large-v3* di *OpenAI*, si profila come una soluzione strategica per superare questa sfida, promuovendo un'interazione più intuitiva e inclusiva. Implementando un sistema di riconoscimento vocale, "Dom the Robot" potrebbe offrire un'esperienza di apprendimento più accessibile e coinvolgente, facilitando l'esperienza utente.

### 6.1.3 Opportunità (Opportunities)

L'integrazione dell'Intelligenza Artificiale in "Dom the Robot", e più in generale nei serious game, rappresenta un'evoluzione significativa nel panorama educativo, offrendo molteplici opportunità. Questa tecnologia permette una personalizzazione dell'esperienza di apprendimento, adattando i contenuti alle esigenze e al ritmo di ogni individuo, e fornisce valutazioni e feedback immediati, incrementando l'efficacia dell'insegnamento. L'AI facilita lo sviluppo di un ambiente educativo più interattivo e coinvolgente, promuovendo l'apprendimento attraverso il gioco e la sperimentazione diretta. Inoltre, essa apre la strada a nuove metodologie didattiche, capaci di stimolare il pensiero critico, la creatività e l'autonomia degli utenti.

Queste innovazioni non solo arricchiscono l'esperienza educativa ma aprono anche nuovi orizzonti per la ricerca e lo sviluppo nel campo dell'*istruzione digitale*, delineando future traiettorie per l'integrazione della tecnologia nell'apprendimento.

### 6.1.4 Minacce (Threats)

L'integrazione di un modello di Intelligenza Artificiale basato sulla somiglianza delle frasi in "Dom the Robot" rappresenta una sfida significativa, evidenziando la dipendenza del gioco da questa specifica tecnologia AI. L'accesso al modello tramite API comporta rischi come interruzioni del servizio, ritardi nella comunicazione dati o problemi di sicurezza legati ai server, i quali possono influenzare negativamente l'esperienza dell'utente, limitando l'accessibilità e l'affidabilità del gioco. Altro aspetto da considerare è l'obsolescenza tecnologica e la necessità di aggiornamenti continui per mantenere l'efficacia didattica e l'engagement dei giocatori.

In aggiunta, la generalizzazione dell'uso dell'Intelligenza Artificiale nei serious game solleva preoccupazioni etiche e di privacy, in quanto la gestione dei dati degli utenti richiede standard di sicurezza elevati. La sfida maggiore risiede nel garantire che queste tecnologie non solo siano accessibili e inclusive ma anche che non amplifichino le disparità esistenti nel campo dell'educazione. La necessità di bilanciare innovazione e responsabilità etica diventa cruciale per assicurare che l'evoluzione dei serious game contribuisca positivamente al panorama educativo senza introdurre nuove barriere o rischi per i suoi utenti.

## 6.2 Lezioni Apprese

### 6.2.1 Adattabilità e Flessibilità

Nell'odierno panorama tecnologico in rapida evoluzione, l'adattabilità e la flessibilità emergono come qualità imprescindibili, soprattutto nel campo dell'apprendimento e dello sviluppo di nuove competenze. La mia esperienza personale, iniziando a utilizzare Unity senza precedenti conoscenze, solo un mese prima di redigere questa tesi, sottolinea l'importanza di essere dinamici e pronti ad apprendere rapidamente. La dedizione e la passione, qualità distintive dei giovani, mi hanno permesso di sviluppare un serious game integrando un modello di AI in meno di tre settimane. Questo processo ha evidenziato non solo l'importanza dell'elasticità intellettuale e della proattività ma anche il valore del coraggio nell'affrontare sfide sconosciute, dimostrando che, con l'impegno, è possibile superare rapidamente le barriere dell'ignoto e realizzare progetti innovativi.

### 6.2.2 L'importanza del Feedback degli Utenti

La fase di sviluppo di "Dom the Robot" ha messo in luce l'importanza cruciale del feedback degli utenti. Far testare il gioco ad amici e familiari, in particolare ai miei giovani nipoti, e ascoltare con attenzione le loro impressioni, mi ha permesso di individuare e migliorare aspetti del gameplay che inizialmente non avevo considerato. Questa esperienza ha sottolineato come l'apertura all'opinione altrui, accolta con umiltà e senza pregiudizi, rappresenti una risorsa straordinariamente potente, spesso sottovalutata, nel processo creativo e di sviluppo. La capacità di ascolto trasforma il feedback in un mezzo per affinare e perfezionare il prodotto finale, evidenziando l'importanza di una comunicazione bidirezionale tra sviluppatori e utenti.

### 6.2.3 Integrazione delle Tecnologie Emergenti

L'esperienza nello sviluppo di "Dom the Robot" ha rivelato l'importanza fondamentale dell'integrazione delle tecnologie emergenti, come l'Intelligenza Artificiale, nel panorama attuale. Comprendere il funzionamento dei modelli AI e la loro applicazione pratica si è dimostrato non solo utile ma essenziale per innovare e rimanere competitivi. Questa competenza si rivelerà preziosa nel futuro, aprendo nuove possibilità di impiego delle tecnologie avanzate in svariati contesti, potenziando l'efficacia e l'efficienza delle soluzioni proposte.

---

## Conclusioni e uno sguardo al futuro

---

Il lavoro svolto in questa tesi ha rappresentato un viaggio esplorativo nell'innovativo campo dell'integrazione tra realtà virtuale (VR) e comprensione del linguaggio naturale (NLU) applicate ai Serious Game per finalità educative. Attraverso un'approfondita analisi teorica e la realizzazione pratica del Serious Game "Dom the Robot", questo studio ha dimostrato come l'uso sinergico di queste tecnologie emergenti possa trasformare radicalmente l'approccio all'apprendimento, rendendolo più immersivo, personalizzato ed efficace.

Il progetto ha evidenziato l'importanza della VR per creare ambienti di apprendimento coinvolgenti e dell'NLU per facilitare interazioni naturali e intuitive, evidenziando come queste tecnologie, unite, possano migliorare significativamente l'esperienza educativa. L'analisi dettagliata del processo di ideazione, progettazione, sviluppo tecnico e valutazione di "Dom the Robot" ha sottolineato le sfide incontrate e le strategie adottate per superarle, dimostrando la fattibilità e l'efficacia di tali approcci innovativi nell'educazione. Cruciale è stato il ruolo delle piattaforme open source, come Hugging Face, che hanno facilitato l'integrazione delle funzionalità di NLU nel Serious Game, permettendo una maggiore accessibilità, democratizzazione e personalizzazione dell'apprendimento.

Il lavoro ha cercato di contribuire al dibattito sull'innovazione pedagogica, fornendo prove concrete del potenziale dei Serious Game supportati da VR e NLU nel ridefinire i metodi di insegnamento e apprendimento. Questo lavoro non solo vuole aprire nuove prospettive per la ricerca futura ma sottolinea anche l'urgenza di integrare tecnologie avanzate nel panorama educativo per preparare gli studenti alle sfide del futuro.

Proiettandosi avanti, l'espansione del progetto "Dom the Robot" potrebbe concentrarsi sull'ampliamento delle capacità di interazione e personalizzazione, introducendo la possibilità per gli utenti di esplorare e comporre tutte le nuance di colori. Questo aggiornamento potrebbe arricchire l'esperienza educativa, stimolando la creatività e l'apprendimento dei concetti di base della teoria dei colori. Inoltre, l'integrazione di nuovi livelli e rompicapi potrebbe aumentare la complessità e la sfida educativa del gioco, incoraggiando il problem-solving e il pensiero critico. Questi sviluppi rappresentano passi avanti naturali nel rendere

"Dom the Robot" un'esperienza ancora più ricca e coinvolgente, mantenendo, al contempo, il focus sull'apprendimento interattivo e personalizzato.

Il futuro dell'industria dell'educazione digitale sembra essere molto promettente e ricco di potenziali evoluzioni. L'integrazione di tecnologie avanzate come l'Intelligenza Artificiale, la realtà virtuale e aumentata, con le piattaforme di apprendimento personalizzato continuerà a trasformare il modo in cui studenti e insegnanti interagiscono con il materiale didattico. Si prevede un aumento dell'accessibilità e della personalizzazione dell'apprendimento, con esperienze educative sempre più immersive e interattive. Questo non solo migliorerà l'efficacia dell'insegnamento ma aprirà anche nuove strade per l'apprendimento continuo e l'auto-miglioramento, rendendo l'educazione più inclusiva e adattabile alle esigenze individuali.

---

## Bibliografia

---

- AZUMA, R. T. (1997), *A Survey of Augmented Reality*, vol. 6, MIT Press.
- BISHOP, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer.
- BOWMAN, D. A., KRUIJFF, E., LAVIOLA JR, J. J. e POUPYREV, I. (2007), «3D user interfaces: theory and practice», *Course Notes*.
- CARMIGNIANI, J. e FURHT, B. (2011), *Augmented Reality: An Overview*, Springer.
- CONNOLLY, T. M., BOYLE, E. A., MACARTHUR, E., HAINEY, T. e BOYLE, J. M. (2012), «A systematic literature review of empirical evidence on computer games and serious games», *Computers & Education*, vol. 59 (2), p. 661–686.
- COOPER, S., KHATIB, F., TREUILLE, A., BARBERO, J., LEE, J., BEENEN, M. e OTHERS (2010), «Predicting protein structures with a multiplayer online game», *Nature*, vol. 466 (7307), p. 756–760.
- DJAOUTI, D., ALVAREZ, J., JESSEL, J.-P. e RAMPNOUX, O. (2011), «Origins of serious games», in «Serious games and edutainment applications», p. 25–43, Springer.
- DOMINGOS, P. (2015), *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, Basic Books.
- GOODFELLOW, I., BENGIO, Y. e COURVILLE, A. (2016), *Deep Learning*, MIT Press.
- GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. e BENGIO, Y. (2014), «Generative adversarial nets», p. 2672–2680.
- GRAAFLAND, M., SCHRAAGEN, J. M. e SCHIJVEN, M. P. (2014), «Systematic review of serious games for medical education and surgical skills training», *British Journal of Surgery*, vol. 99 (10), p. 1322–1330.



- JORDAN, M. I. e MITCHELL, T. M. (2015), «Machine Learning: Trends, Perspectives, and Prospects», *Science*.
- KAEHLING, L. P., LITTMAN, M. L. e MOORE, A. W. (1996), «Reinforcement Learning: A Survey», *Journal of Artificial Intelligence Research*.
- KAPP, K. M. (2012), *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*.
- KATO, P. M., COLE, S. W., BRADLYN, A. S. e POLLOCK, B. H. (2008), «A video game improves behavioral outcomes in adolescents and young adults with cancer: A randomized trial», *Pediatrics*, vol. 122 (2), p. e305–e317.
- KICKMEIER-RUST, M. D. e ALBERT, D. (2010), «Micro-adaptivity: Protecting immersion in didactically adaptive digital educational games», *Journal of Computer Assisted Learning*, vol. 26 (2), p. 95–105.
- LANIER, J. (2017), *Dawn of the New Everything: Encounters with Reality and Virtual Reality*, Henry Holt and Co.
- LECUN, Y., BENGIO, Y. e HINTON, G. (2015), «Deep learning», *Nature*, vol. 521 (7553), p. 436–444.
- LOH, C.-S., SHENG, Y. e IFENTHALER, D. (2015), «Serious games analytics: Theoretical framework», in «Serious Games Analytics», p. 3–29, Springer.
- LUGER, G. F. (2008), *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Pearson, 6 ed.
- MAERTENS, H., MADANI, A., LANDRY, T., VERMASSEN, F., VAN HERZEELE, I. e AGGARWAL, R. (2016), «Systematic review of e-learning for surgical training», *British Journal of Surgery*, vol. 103 (11), p. 1428–1437.
- MAIMONE, A. e FUCHS, H. (2014), *Holographic Near-eye Displays for Virtual and Augmented Reality*, vol. 36, ACM.
- MARCZEWSKI, A. (2014), *What is the difference between a serious game and gamification?*
- MICHAEL, D. R. e CHEN, S. (2006), *Serious Games: Games That Educate, Train, and Inform*, Muska & Lipman/Premier-Trade.
- MILGRAM, P. e KISHINO, F. (1994), *A Taxonomy of Mixed Reality Visual Displays*, vol. E77-D, Institute of Electronics, Information and Communication Engineers.
- MITCHELL, T. M. (1997), *Machine Learning*, McGraw-Hill.
- MURPHY, K. P. (2012), *Machine Learning: A Probabilistic Perspective*, MIT Press.

- NILSSON, N. J. (1998), *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann.
- NORVIG, P. (2014), *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*, Morgan Kaufmann.
- POOLE, D. e MACKWORTH, A. (2017), *Artificial Intelligence: Foundations of Computational Agents*, Cambridge University Press, 2 ed.
- RICONSCENTE, M. M. (2013), «Results from a controlled study of the iPad fractions game motion math», *Games and Culture*, vol. 8 (4), p. 186–214.
- RITTERFELD, U., CODY, M. J. e VORDERER, P. (2009), «Serious games: Mechanisms and effects», *Routledge*.
- RUPP, M. A., KOZACHUK, J., MICHAELIS, J. R., ODETTE, K. L., SMITHER, J. A. e MCCONNELL, D. S. (2016), «The effects of immersiveness and future VR expectations on subjective experiences during an educational 360° video», *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 60 (1), p. 2108–2112.
- RUSSELL, S. J. e NORVIG, P. (2020), *Artificial Intelligence: A Modern Approach*, Pearson, 4 ed.
- SCHMALSTIEG, D. e HOLLERER, T. (2016), *Augmented Reality: Principles and Practice*, Addison-Wesley Professional.
- SICART, M. (2019), *Play Matters*, Playful Thinking, MIT Press.
- SILVER, D., SCHRITTWIESER, J., SIMONYAN, K., ANTONOGLOU, I., HUANG, A., GUEZ, A., HUBERT, T., BAKER, L., LAI, M., BOLTON, A. e OTHERS (2017), «Mastering the game of Go without human knowledge», *Nature*, vol. 550 (7676), p. 354–359.
- SUSI, T., JOHANNESSON, M. e BACKLUND, P. (2007), «Serious games: An overview», .
- SUTHERLAND, I. E. (1968), *A Head-mounted Three Dimensional Display*.
- SUTTON, R. S. e BARTO, A. G. (2018), *Reinforcement Learning: An Introduction*, MIT Press, 2 ed.
- TEGMARK, M. (2017), *Life 3.0: Being Human in the Age of Artificial Intelligence*, Knopf.
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L. e POLOSUKHIN, I. (2017), «Attention is all you need», p. 5998–6008.
- WOOLDRIDGE, M. (2020), *The Road to Conscious Machines: The Story of AI*, Pelican.
- WOUTERS, P., VAN NIMWEGEN, C., VAN OOSTENDORP, H. e VAN DER SPEK, E. D. (2013), «A meta-analysis of the cognitive and motivational effects of serious games», *Journal of Educational Psychology*, vol. 105 (2), p. 249–265.
- ZYDA, M. (2005), *From visual simulation to virtual reality to games*, vol. 38.

## Siti web consultati

- Autodesk, – [www.autodesk.com/solutions/virtual-reality](http://www.autodesk.com/solutions/virtual-reality)
- Jasoren, – [jasoren.com/how-to-develop-serious-ar-and-vr-games/](http://jasoren.com/how-to-develop-serious-ar-and-vr-games/)
- ARVRtech, – [arvrtech.eu/serious-game-with-ar-fi-content/](http://arvrtech.eu/serious-game-with-ar-fi-content/)
- GameDev Academy, – [gamedevacademy.org/best-xr-tutorials/](http://gamedevacademy.org/best-xr-tutorials/)
- ProjectFun, – [projectfun.com/serious-game-guida/](http://projectfun.com/serious-game-guida/)
- Serious Games Society, – [www.seriousgamesociety.org](http://www.seriousgamesociety.org)
- VRFocus, – [www.vrfocus.com](http://www.vrfocus.com)
- AI and Games, – [www.aiandgames.com](http://www.aiandgames.com)
- The VR Scout, – [www.vrscout.com](http://www.vrscout.com)
- ARPost, – [www.arpost.co](http://www.arpost.co)
- Thomas Simonini web site, – <https://thomassimonini.substack.com/p/building-a-smart-robot-ai-using-hugging>
- Mixed Reality Developer, – [developer.microsoft.com/en-us/mixed-reality](http://developer.microsoft.com/en-us/mixed-reality)
- Gamasutra, – [www.gamasutra.com](http://www.gamasutra.com)
- Road to VR, – [www.roadtovr.com](http://www.roadtovr.com)
- Unity Learn, – [learn.unity.com](http://learn.unity.com)
- MIT Game Lab, – [gamelab.mit.edu](http://gamelab.mit.edu)
- Wikipedia, – [www.wikipedia.org](http://www.wikipedia.org)
- HuggingFace, – [www.huggingface.co](http://www.huggingface.co)
- OpenAI, – [www.openai.com](http://www.openai.com)
- StudioSamo, – [www.studiosamo.it](http://www.studiosamo.it)
- Forum Unity, – [forum.unity.com](http://forum.unity.com)

---

## Ringraziamenti

---

Ringrazio, innanzitutto, il professor Domenico Ursino, il quale mi ha permesso di svolgere questo lavoro estremamente stimolante, riguardante argomenti assolutamente attuali; grazie per essere stato sempre disponibile, per avermi seguito con costanza, pazienza e tempestività, nonostante i molteplici impegni.

Ringrazio Andrea, per tutte le possibilità che mi sta permettendo di cogliere. Ti stimo e ti ammiro sotto qualsiasi aspetto di vita: sportivo, imprenditoriale, umano e familiare.

Ringrazio mamma e papà, per i loro sacrifici volti ad un mio futuro prosperoso, per non aver mai dubitato di me, per avermi fatto capire quanto coraggio e tenacia servano nella vita per raggiungere i propri obiettivi.

Ringrazio mia sorella, fonte smisurata di consigli utilissimi nei periodi più complicati.

Ringrazio zio Matteo e zia Simona, che mi hanno sempre tenuto a cuore, supportandomi e rendendosi sempre disponibili in qualsiasi occasione.

Ringrazio Sara, che mi ha sopportato e supportato facendolo tuttora; non deve essere per niente facile avere a che fare con me. Mi sei sempre stata vicina. Ormai sei parte integrante della mia vita, del mio cuore. Sei quasi più felice di me per questo traguardo, e ciò la dice lunga sul tipo di persona che ho al mio fianco.

Ringrazio, ovviamente, tutti i ragazzi con cui ho condiviso questo splendido percorso. Senza di loro questi anni sarebbero stati molto diversi. Grazie, quindi, a Lu, Cima, Ale, Sandro, Brilian, Serena, Mattia, Den, Annarita, Greta e Gaia.

Ringrazio Matte, che ho conosciuto verso la fine del secondo anno, ma che ormai sembra una vita fa; il Capodanno a Barcellona non si potrà mai dimenticare.

Infine, c'è la persona più importante di tutti. Adriano. Abbiamo convissuto per più di tre anni. Grazie a te ho imparato la disciplina, che applichi meticolosamente in qualsiasi ambito, dall'allenamento, passando per lo studio, fino all'alimentazione. Con il tempo ho appreso così tanto da te; spero di averti lasciato qualcosa, o mi sentirei in debito. Mio complice a 360 gradi. Basta uno sguardo e ci capiamo. Siamo stati fortunati. Grazie di tutto.