

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria del Cinema
e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale



**Politecnico
di Torino**

Multiplayer game per l'interazione e la formazione su processi industriali

Relatore: Andrea Bottino

Studentessa: Francesca Bellanova

Matricola s301151

INDICE

INDICE DELLE FIGURE.....	iv
INDICE DELLE TABELLE	v
ABSTRACT	1
1. INTRODUZIONE	2
1.1. Obiettivi del progetto	2
1.2. Multiplayer games.....	3
1.2.1. Breve storia dei giochi multiplayer	3
1.3. Serious game.....	4
2. STATO DELL'ARTE	7
2.1. Casi studio	7
2.1.1. Tous Digital - Digital retail training	7
2.1.2. Robot city	8
2.1.3. Muscle car	8
2.1.4. Lean bicycle factory game.....	9
2.2. Tools multiplayer	10
2.2.1. Unity Relay.....	10
2.2.2. Netcode for GameObjects	12
2.2.3. Mirror	12
2.2.4. FishNet	12
2.2.5. Photon Quantum	13
2.2.6. Photon fusion.....	14
3. TECNOLOGIE UTILIZZATE.....	15
3.1. Blender	15
3.2. Unity.....	15
3.2.1. Multiplayer Tools.....	16
3.2.2. Network Manager	16
3.2.4. Network Behaviour.....	17
3.2.5. Server RPC e Client RPC.....	18
3.2.6. Parrel Sync	19
4. GAME DESIGN.....	20
4.1. Sinossi	20
4.1.1. Player	21

4.2.	Meccaniche di gioco.....	22
4.3.	Level design.....	23
4.4.	Interfaccia utente.....	26
5.	SVILUPPO.....	27
5.1.	Menu principale.....	27
5.2.	Lobby.....	28
5.3.	Configurazione livello.....	30
5.4.	Tutorial.....	33
5.5.	Primo livello.....	36
6.5.1.	Player.....	41
6.5.2.	Shader Graph e ParticleSystem.....	42
6.5.3.	Audio.....	44
5.6.	Secondo livello.....	45
6.6.1.	Merge.....	48
5.7.	Fine livello.....	50
6.	RISULTATI.....	52
6.1.	Questionari utilizzati.....	52
6.2.	Test.....	53
6.3.	Usabilità del sistema.....	55
6.4.	Coinvolgimento generale.....	57
6.5.	Apprendimento.....	58
6.6.	Impatto sociale.....	59
7.	CONCLUSIONI.....	62
7.1.	Sviluppi futuri.....	62
	BIBLIOGRAFIA.....	64

INDICE DELLE FIGURE

<i>Figura 1: Aree e utilizzi dell'apprendimento basato sul gioco</i>	5
<i>Figura 2: Schermata di gioco di Tous Digital</i>	8
<i>Figura 3: Schermata di gioco di Lean Bicycle Factory</i>	9
<i>Figura 4: Modello di funzionamento del Relay server</i>	11
<i>Figura 5 - Funzionamento dell'engine Photon Quantum</i>	13
<i>Figura 6: Network Manager nell'Inspector</i>	17
<i>Figura 7: Esempio di chiamata ServerRpc e ClientRpc</i>	19
<i>Figura 8: Tubo finale da produrre</i>	21
<i>Figura 9: Palette per l'ambiente</i>	24
<i>Figura 10: Colori per i giocatori</i>	24
<i>Figura 11: Mappa vista dall'alto</i>	25
<i>Figura 12: Gaming Services utilizzati</i>	28
<i>Figura 13: Scena della Lobby</i>	30
<i>Figura 14: Interfaccia per la configurazione del livello</i>	31
<i>Figura 15: Struttura del player nella scena di configurazione</i>	32
<i>Figura 16: Mappa del tutorial</i>	34
<i>Figura 17: Tutorial - Primo step</i>	35
<i>Figura 18: Tutorial - Secondo step</i>	36
<i>Figura 19: Tutorial - Terzo step</i>	36
<i>Figura 20: Schermata di riepilogo dei controlli</i>	37
<i>Figura 21: GameManager nell'Inspector di Unity</i>	40
<i>Figura 22: Player nell'Inspector</i>	42
<i>Figura 23: Shader graph per le indicazioni di movimento</i>	43
<i>Figura 24: Particle system</i>	44
<i>Figura 25: SuperMarket per le work station</i>	46
<i>Figura 26: Animator per l'apertura della box</i>	47
<i>Figura 27: Guida al merge dei tubi</i>	48
<i>Figura 28: Pipe nell'Inspector</i>	50
<i>Figura 29: Distribuzione anagrafica</i>	54
<i>Figura 30: Frequenza di utilizzo dei videogiochi</i>	54

INDICE DELLE TABELLE

<i>Tabella 1: Confronto tra le soluzioni di networking per Unity.....</i>	<i>14</i>
<i>Tabella 2: Risposte al questionario SUS.....</i>	<i>56</i>
<i>Tabella 3: Media e deviazione standard per il questionario sul coinvolgimento generale.....</i>	<i>57</i>
<i>Tabella 4: Media e deviazione standard per il questionario sull'apprendimento.....</i>	<i>58</i>
<i>Tabella 5: Media e deviazione standard per il questionario sull'impatto sociale</i>	<i>60</i>

ABSTRACT

I serious games si sono affermati come strumenti educativi di grande efficacia in svariati contesti, grazie alla loro capacità di coinvolgimento degli utenti e all'apprendimento che avviene tramite un'esperienza ludica interattiva. Questa tipologia di giochi viene implementata in vari settori per sviluppare competenze come la pianificazione, l'analisi e la risoluzione dei problemi. In particolare, i serious games dedicati alla simulazione di processi sfruttano un ambiente virtuale in cui i giocatori possono acquisire familiarità con dei flussi di lavoro complessi e prendere delle decisioni strategiche, permettendo di valutare gli impatti delle modifiche pianificate senza dover sostenere i costi di una trasformazione reale prima di verificarne gli effetti. L'obiettivo di questo lavoro di tesi è sviluppare un serious game multiplayer volto a facilitare l'apprendimento dei concetti e degli strumenti della Lean Manufacturing all'interno di un ambiente virtuale. Il serious game, sviluppato con Unity 3D, coinvolge dieci utenti, ognuno dei quali ha un ruolo specifico, in una simulazione di una catena di produzione. L'obiettivo principale del gioco è quello di ottimizzare la produzione attraverso l'analisi degli errori commessi durante il gameplay, rendendo l'esperienza di gioco didattica e orientata alla risoluzione di problemi.

CAPITOLO 1

1. INTRODUZIONE

1.1. Obiettivi del progetto

Questa tesi è stata svolta presso l'azienda Synarea Consultant, una realtà che fornisce soluzioni in ambito ICT, con sede a Torino, dove ho avuto l'opportunità di condurre un progetto di ricerca finalizzato all'analisi e allo sviluppo di dinamiche multiplayer impiegando il framework Unity. Il fine principale della mia ricerca è stato esplorare e implementare soluzioni innovative nel contesto delle esperienze di gioco multiplayer.

Seguendo le linee guida dettate dall'azienda, ho usufruito di Unity come strumento principale per lo sviluppo software, affiancato da Blender per la creazione di elementi grafici tridimensionali. Questa scelta degli strumenti si è rivelata cruciale per lo svolgimento del progetto, consentendomi di approfondire le dinamiche di sviluppo e la progettazione grafica nell'ambito di un contesto multiplayer.

Oltre all'analisi delle dinamiche multiplayer e alle implementazioni pratiche con Unity, la tesi si inserisce anche in un contesto più ampio e interdisciplinare, dove l'approccio ai serious games diventa un elemento fondamentale per l'apprendimento efficace, grazie alla loro capacità innata di coinvolgere gli utenti e facilitare il processo di apprendimento attraverso un'esperienza ludica interattiva. Questa particolare categoria di giochi educativi viene attivamente implementata in diversi settori con l'obiettivo di sviluppare competenze chiave, tra cui la pianificazione, l'analisi e la risoluzione dei problemi.

In particolare, i serious games dedicati alla simulazione di processi si collocano in un ambiente virtuale stratificato, offrendo ai giocatori l'opportunità di acquisire familiarità con flussi di lavoro complessi e di prendere decisioni strategiche significative. Tale approccio consente di valutare gli impatti di eventuali modifiche pianificate senza la necessità di sostenere i costi di una trasformazione reale, permettendo così agli utenti di affinare le proprie competenze senza rischi finanziari.

L'obiettivo intrinseco di questa ricerca è di sviluppare un serious game multiplayer focalizzato sull'agevolazione dell'apprendimento dei concetti e degli strumenti associati alla Lean Manufacturing all'interno di un contesto virtuale. Attraverso l'utilizzo di Unity 3D, il serious

game coinvolge attivamente dieci utenti, ognuno dei quali assume un ruolo specifico all'interno di una simulazione accurata di una catena di produzione.

Il nucleo di questo gioco educativo è l'ottimizzazione della produzione, che viene perseguita attraverso l'analisi meticolosa degli errori commessi durante l'esperienza di gioco. Questo approccio didattico e orientato alla risoluzione di problemi rende il serious game un efficace metodo formativo, dove gli utenti non solo acquisiscono una comprensione pratica della Lean Manufacturing, ma affinano anche le proprie capacità decisionali e gestionali in un contesto simulato e sicuro.

1.2. Multiplayer games

I videogiochi multiplayer costituiscono una risorsa importante ed entusiasmante per i videogiocatori, offrendo la possibilità a più utenti di condividere simultaneamente un'esperienza coinvolgente all'interno di uno stesso ambiente virtuale. A seconda della tipologia di gioco, i giocatori possono interagire in modi diversificati, potendo scegliere tra giochi competitivi, dove la sfida reciproca è il motore per raggiungere obiettivi specifici, e giochi cooperativi, dove la collaborazione diventa fondamentale per raggiungere degli obiettivi comuni.

L'accesso ai giochi multiplayer può avvenire attraverso due modalità distinte: localmente, utilizzando la connessione LAN o la suddivisione dello schermo (split screen), richiedendo che i giocatori siano nella stessa stanza o connessi alla stessa rete locale; oppure online, consentendo ai giocatori di partecipare simultaneamente da diverse posizioni attraverso qualsiasi connessione Internet. Questa flessibilità offre agli appassionati di videogiochi la possibilità di godere di esperienze multiplayer in diversi contesti e modalità, contribuendo a plasmare un'ampia varietà di interazioni e sfide nel mondo dei giochi. Infatti, la modalità online ha ampliato significativamente le possibilità di partecipazione, consentendo a giocatori provenienti da diverse posizioni geografiche di condividere un ambiente virtuale e competere in tempo reale. La connessione a Internet è diventata un elemento fondamentale per l'esperienza multiplayer, fornendo una piattaforma che può essere utilizzare per competizioni globali, chat vocali e interazioni sociali.

1.2.1. Breve storia dei giochi multiplayer

Il gioco Tennis for Two, ideato nel 1958 dall'ingegnere statunitense William Higinbotham, emerge come pietra miliare nell'evoluzione dei videogiochi multiplayer. Quest'opera innovativa, sviluppata come dimostrazione, impiegava un oscilloscopio come monitor, permettendo a due giocatori di sfidarsi a tennis controllando le racchette mediante pulsanti e paddle.

Nel 1962, Spacewar! ha lasciato un'impronta significativa come uno dei primi sparattutto per computer. Questo offriva un ambiente di gioco in cui i giocatori potevano competere tra loro in simultaneamente, con regole fisiche che cambiavano in tempo reale. Ciò che lo differenzia dagli altri giochi risiede nel fatto che possedeva una modalità single player, ma possedeva unicamente la modalità multiplayer, essendo anche sprovvisto di intelligenza artificiale.

Nel 1977, Empire ha segnato un altro passo avanti nello sviluppo dei giochi multiplayer. Progettato per il sistema PLATO, permetteva ad un massimo di tre giocatori di connettersi a una rete condivisa e di combattere per la conquista di un mondo virtuale.

Nel 1978, l'avvento dei MUD (Multi-User Dungeon) ha introdotto i giochi di ruolo online testuali, con il MUD1 di Roy Trubshaw e Richard Bartle lanciato su ARPANET, la rete precursora di Internet.

In questo affascinante contesto storico, queste creazioni hanno svolto un ruolo cruciale nello stabilire le fondamenta dei videogiochi multiplayer. Oggi, la storia si è arricchita ulteriormente con l'emergere di giochi multiplayer di grande successo.

Tra questi, i più giocati sono:

- Fortnite, un battle royale veloce e costruttivo lanciato nel 2017, ha attratto milioni di giocatori grazie alla sua combinazione di gioco veloce, elementi costruttivi e stile artistico unico.
- PlayerUnknown's Battlegrounds (PUBG), anch'esso del 2017, ha contribuito a definire il genere battle royale, offrendo una competizione intensa con 100 giocatori su un'isola che si sfidano per essere gli ultimi sopravvissuti.
- League of Legends, un MOBA lanciato nel 2009, che nonostante l'età continua a vantare milioni di giocatori attivi ogni giorno.
- Counter-Strike: Global Offensive (CS:GO), un videogioco competitivo sparattutto in prima persona del 2012, è uno dei titoli più giocati nelle competizioni e online.

1.3. Serious game

I giochi trattati nel paragrafo precedente sono prettamente dei videogiochi sviluppati per intrattenere il giocatore. I serious games sono dei giochi in cui l'obiettivo primario non è intrattenere, bensì ottenere delle competenze ed apprendere. In ogni gioco possiamo riscontrare tre caratteristiche differenti: intrattenimento, simulazione e apprendimento.

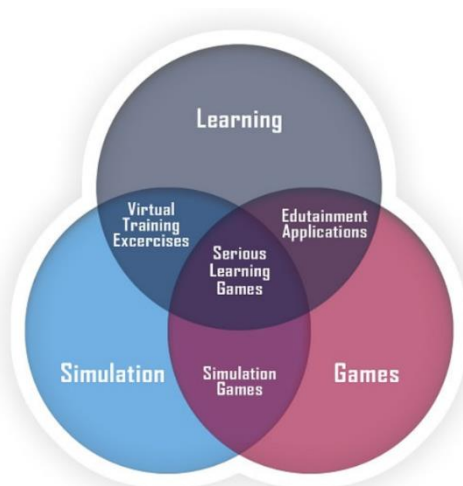


Figura 1: Aree e utilizzi dell'apprendimento basato sul gioco¹.

Queste tre caratteristiche possono sovrapporsi fra di loro, però hanno delle differenze intrinseche nella loro struttura.

I learning games sono dei giochi che hanno come scopo principale l'apprendimento. Un esempio di learning games è Kahoot!, una piattaforma che viene utilizzata principalmente in contesti didattici e che si basa sull'utilizzo di quiz a scelta multipla.

Nella categoria di Games, rientrano tutti i giochi che sono stati progettati principalmente per intrattenere il giocatore.

I giochi di simulazione invece sono progettati per riprodurre un aspetto della realtà. Questi vengono utilizzati per acquisire o allenare varie competenze, insegnando dei comportamenti che sono utili nel contesto delle situazioni simulate. Questi giochi sono utili nel momento in cui offrono l'opportunità di partecipare a delle situazioni impossibili. L'ambiente simulato infatti consente ai partecipanti di sperimentare esperienze che sarebbero troppo costose, rischiose o anche fisicamente impossibili da realizzare. Ad esempio, le simulazioni vengono utilizzate per imparare a pilotare aerei.

Dalle varie intersezioni tra simulation, games e learning troviamo:

- Virtual Training Exercises: questi sono degli esercizi che riguardano delle esercitazioni simulate da un istruttore, utilizzate tipicamente per coinvolgere gli apprendisti in modo divertente o educativo.
- Serious Games: combinano gli elementi ludici, simulativi e di apprendimento con l'obiettivo di raggiungere scopi educativi e formativi. Possono essere utilizzati per

¹ Outlife.in, *Online Game Based Experiential Learning*, <https://www.outlife.in/game-based-experiential-learning.html> (ultimo accesso 27/12/2023)

l'educazione su valori in modalità single player, mentre nella loro versione multiplayer possono essere utilizzati da più giocatori per la costruzione del team, la collaborazione, la leadership e la formazione manageriale. È importante notare che in questa tipologia di giochi si verifica il trasferimento dell'apprendimento dal gioco alla vita reale e viceversa, perché ogni partita viene mediata da riflessioni e feedback con il team stesso.

Molto spesso l'origine dei serious game² viene attribuita alle simulazioni da guerra dell'esercito prussiano nel XVIII secolo, chiamati Kriegsspiel, un gioco di strategia che ricostruiva degli elementi militari realmente accaduti o anche frutto dell'immaginazione. Esistono però anche degli esempi più antichi di serious game, infatti nel 799, Alcuino da York ha compilato dei giochi logico-matematici per istruire gli studenti della Scuola Palatina.

Il termine serious game fu coniato probabilmente da Clark C. Abt nel 1971. nel suo libro intitolato "Serious Games", definisce i giochi educativi, indipendentemente dal supporto che fosse digitale o nel mondo reale, e li identifica come giochi con uno scopo educativo esplicito, non concepiti principalmente per il puro divertimento, sebbene quest'ultimo non sia escluso.

Negli anni 2000, si è verificato un aumento significativo di giochi educativi, specialmente quelli destinati ai bambini. Nel 1999, LeapFrog Enterprises ha lanciato il LeapPad, un ibrido tra un libro interattivo e una cartuccia, consentendo ai bambini di interagire con un libro di carta. Ispirandosi ai successi dei tradizionali sistemi di gioco portatili come il Nintendo Game Boy, nel 2003 hanno introdotto il Leapster, un sistema portatile basato su cartucce con giochi in stile arcade integrati a contenuti educativi.

² Wikipedia, *Serious game*, https://it.wikipedia.org/wiki/Serious_game (ultimo accesso 28/12/2023)

CAPITOLO 2

2. STATO DELL'ARTE

Come visto nel capitolo precedente, i serious game possono essere utilizzati in ambiti molto diversi tra loro. In questo progetto di tesi il serious game sviluppato è stato pensato per essere utilizzato in ambito gestionale, avendo come punto di riferimento la filosofia del Lean Manufacturing.

Quest'ultima è un insieme di metodi che, se adottati, riescono a migliorare le performance aziendali, tramite un'organizzazione efficiente che permette di massimizzare la quantità di valore e minimizzare gli sprechi.

2.1. Casi studio

Prima della realizzazione effettiva del progetto è stata svolta un'analisi dei serious game multiplayer già sviluppati e successivamente dei serious game specifici per la Lean manufacturing, presentati qui in seguito.

2.1.1. Tous Digital - Digital retail training ³

Questo serious game 3D è stato sviluppato per i dipendenti francesi dell'azienda Carrefour ed ha l'obiettivo di migliorare le abilità digitali dei dipendenti, allenandoli all'uso quotidiano delle tecnologie e dei tools per l'e-commerce nel luogo di lavoro, in base al compito che svolgono all'interno della compagnia. Il target scelto per questo serious game viene classificato come eterogeneo, in quanto comprende sia dipendenti che lavorano quotidianamente con il digitale, sia dipendenti che lavorano in luoghi come magazzini e negozi fisici che non utilizzano costantemente le tecnologie digitali. Il gioco si suddivide in tre parti ed i giocatori vengono suddivisi in tre squadre da 10 a 25 persone, le quali competono tra di loro guadagnando dei punteggi in base alla loro performance.

³ Emeraude-escape, *Carrefour – Gamified Digital Retail Training*, <https://emeraude-escape.com/en/case/carrefour-gamified-digital-training/> (ultimo accesso 4/01/2024)



Figura 2: Schermata di gioco di Tous Digital

2.1.2. Robot city ⁴

Questo serious game multiplayer è un gioco collaborativo che ha lo scopo di far imparare linguaggi di programmazione ad un target che comprende studenti delle scuole elementari. È sostanzialmente un gioco di carte che è stato riproposto in versione digitale online, in cui è presente un ambiente 2d. È presente anche una chat per permettere ai giocatori di comunicare tra di loro, rendendo il tutto più interattivo.

2.1.3. Muscle car ⁵

Un esempio di serious game utilizzati per la produzione snella è “Muscle car”, un gioco sviluppato per un evento formativo inerente al progetto ActionPlanT. Questo serious game permette ai giocatori di acquisire, tramite proprie meccaniche di gioco, i concetti della lean manufacturing. Il gioco si basava sull’utilizzo di modelli di auto Lego ed era svolto in 3 turni che andavano a simulare 3 settimane di lavoro. I giocatori venivano suddivisi in due squadre, ed ognuno aveva uno specifico ruolo nel team. Lo scopo primario del gioco è quello di simulare una catena di montaggio di un’automobile. Sostanzialmente i giocatori devono massimizzare il numero di auto assemblate riducendo al minimo i costi associati al magazzino e all’inventario. I giocatori erano in grado di affrontare i vari problemi legati all’industria manifatturiera adottando diverse soluzioni.

⁴ Leo J. de Vin, Lasse Jacobsson, JanErik Odhe, *Game-based Lean Production training of university students and industrial employees*, Karlstad University, 651 88 Karlstad, Sweden (2018)

⁵ Messaadia, M., Bufardi, A., Le Duigou, J., Szigeti, H., Eynard, B., Kiritsis, D. (2013). *Applying Serious Games in Lean Manufacturing Training*. In: Emmanouilidis, C., Taisch, M., Kiritsis, D. (eds) *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services. APMS 2012. IFIP Advances in Information and Communication Technology*, vol 397. Springer, Berlin, Heidelberg.

Questo serious game non è un videogame, in quanto è realizzato in versione cartacea ma è comunque un buon esempio in quanto i giocatori hanno valutato positivamente l'esperienza di gioco ed hanno acquisito nuove abilità inerenti alla produzione snella.

2.1.4. Lean bicycle factory game⁶

Questo videogioco, sviluppato da un'azienda svedese, viene ambientato in uno scenario in cui vengono prodotte delle biciclette tramite una serie di processi. Gli operai che si vedono sullo schermo durante la partita si muovono autonomamente, mentre il giocatore segue tutto il percorso dei componenti della bicicletta dallo stampaggio, assemblaggio, verniciatura e controllo qualità. La partita si sviluppa in 5 turni, durante i quali il giocatore o un team di giocatori possono apportare delle modifiche per ottimizzare i processi, avendo precedentemente notato dei movimenti sprecati, degli ordini trascurati o dei magazzini non ottimali. Tra i turni è consentito effettuare solo due modifiche, ognuna delle quali ha un costo. L'obiettivo finale del gioco è quello di raggiungere una disposizione ottimale.

Questo serious game a differenza del precedente è un vero e proprio videogioco isometrico con grafica 2d, nel quale la user interface presenta pochi dettagli come il tempo di gioco, il numero di round e la valuta a disposizione, questo perché la parte di organizzazione avviene offline tra i vari round e non real-time nelle partite.



Figura 3: Schermata di gioco di Lean Bicycle Factory

⁶ leansimulations.org, *LEAN BICYCLE FACTORY GAME*, <https://www.leansimulations.org/2010/07/lean-bicycle-factory-game.html> (ultimo accesso 05/01/2024)

2.1.5. 5S GAME⁷

Questo gioco presenta quattro diversi scenari: un magazzino, un ufficio aziendale, un banco di lavoro condiviso, una scrivania informatica, molto simile al desktop di Windows.

Questo serious game ha come obiettivo quello di migliorare la Lean manufacturing, avendo il focus in particolare sul metodo 5S. Ad ogni giocatore viene presentato un elenco di oggetti disponibili; ognuno deve selezionare quelli strettamente necessari. Ogni oggetto viene contrassegnato con delle etichette in base alla rilevanza e l'urgenza di completamento. Il giocatore deve poi organizzarsi, in modo da porre gli oggetti nei luoghi più opportuni per essere conservati finché non verranno utilizzati. È possibile, inoltre, effettuare delle operazioni di pulizia.

Anche in questo caso il videogame presenta una grafica minimale, che va a simulare la GUI di un computer. Ciononostante, anche in questo caso i giocatori hanno valutato positivamente l'esperienza di gioco.

Da questa analisi ne esce fuori che i serious game trovati hanno una grafica minimale e risulta importante la collaborazione tra i giocatori, che devono avere ruoli diversi affinché la produzione diventi snella. Inoltre, la parte di managing e di gestione delle risorse e suddivisione dei ruoli è necessario che sia offline per permettere di riflettere e comunicare meglio le decisioni con il team. Dopo di ciò è stata svolta una ricerca per trovare i tools e le librerie migliori per implementare un sistema di networking e multiplayer all'interno di Unity. Al momento della creazione del progetto, erano presenti molte soluzioni per lo sviluppo di giochi in multiplayer, tra cui: FishNet, Mirror Networking, Photon Quantum, Photon Fusion, Netcode for GameObjects.

2.2. Tools multiplayer

2.2.1. Unity Relay

Unity fornisce per gli sviluppatori di videogiochi Unity Relay⁸, che permette di avere una connessione P2P (peer-to-peer) tra i giocatori. Questo implica che i giocatori sono connessi tra di loro tramite rete, senza dover utilizzare dei server dedicati; infatti, fornisce connettività attraverso un server Relay universale, che funge da proxy. Viene utilizzato un meccanismo basato su codici di accesso per invitare altri giocatori ad una partita, i quali effettuano l'accesso ad un server Relay,

⁷ Duarte F. Gomes, M. Pereira Lopes, C. Vaz de Carvalho, *Serious Games for Lean Manufacturing: The 5S Game*, IEEE Revista Iberoamericana de Tecnologías del Aprendizaje 8(4):191-196
DOI: 10.1109/RITA.2013.2284955 (2013)

⁸ Docs.unity.com, Unity Relay, <https://docs.unity.com/ugs/manual/relay/manual/introduction> (ultimo accesso 20/12/2023)

comunicando direttamente con esso, utilizzando delle comunicazioni veloci e sicure con protocollo UDP.

Unity Relay si compone di due caratteristiche principali: i Relay Server e i Relay Allocation Service. I server Relay sono indipendenti dal ciclo di vita della sessione di gioco in quanto sono progettati come punti di connessione a lunga durata.

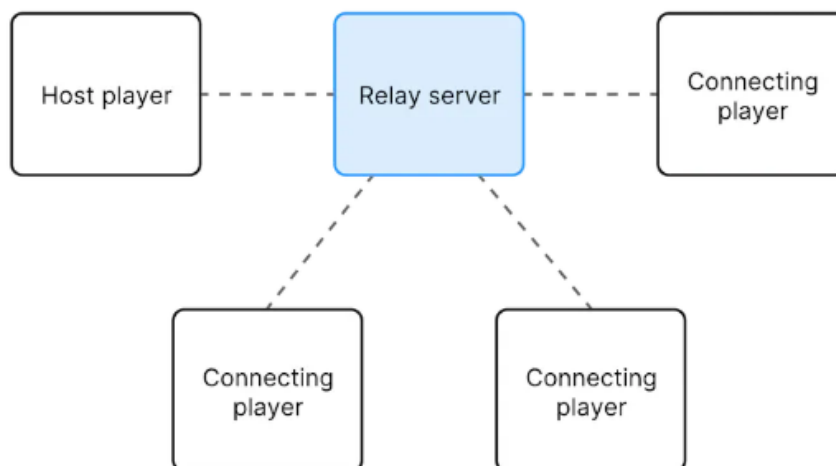


Figura 4: Modello di funzionamento del Relay server⁹

Questi agevolano lo scambio rapido di messaggi tra giocatori attraverso datagrammi a bassa latenza. Questo approccio è ottimale per giochi con una struttura di server in ascolto, dove un giocatore agisce da host e gli altri da client. Il servizio di allocazione relay, invece, si occupa della creazione e partecipazione alle diverse partite di giochi mediante l'utilizzo di appositi codici di accesso.

Relay può essere utilizzato gratuitamente se si rispettano dei limiti di utilizzo di traffico dati mensili, implementa delle connessioni sicure, è scalabile in quanto adatto a un numero moderato di giocatori purché questo sia minore di cento. Inoltre, il servizio utilizza il rate limiting per controllare il traffico di rete, limitando a 60 richieste al minuto, per ogni giocatore autenticato, operazioni fondamentali come "Crea assegnazione", "Crea codice di adesione", "Unisciti a un Relay".

⁹ Docs.unity, *Relay servers*, <https://docs.unity.com/ugs/en-us/manual/relay/manual/relay-servers> (ultimo accesso 5/01/2024)

2.2.2. Netcode for GameObjects

Netcode for GameObjects¹⁰ è un framework high-level di rete, sviluppato direttamente da Unity, che semplifica l'invio dei dati e la sincronizzazione dei GameObjects tra i client. Supporta varie architetture di rete, inclusi modelli peer-to-peer, Client-Server e server dedicati e soprattutto fornisce funzionalità avanzate come la gestione di server dedicati, il matchmaking, le lobby e la chat. La piattaforma, in sintesi, riduce la complessità dello sviluppo dei giochi multiplayer, consentendo agli sviluppatori di focalizzarsi esclusivamente sulla logica del gioco senza doversi preoccupare della gestione della rete. Inoltre, fornisce un'ampia gamma di funzionalità che permettono di creare esperienze multiplayer complete e coinvolgenti per gli utenti.

2.2.3. Mirror

Mirror Networking¹¹ è un framework open-source gratuito che supporta diverse architetture di rete. Questa libreria è usata da una parte molto ampia di sviluppatori, per cui si può trovare una community molto ampia e attiva, che fornisce varie risorse. Mirror offre la possibilità di gestire il contesto, diversificando il codice su client e server, permettendo di personalizzare il comportamento del server. In conclusione, Mirror, contribuisce a ridurre i costi di sviluppo, velocizzando contemporaneamente il processo di realizzazione del gioco. Inoltre, mette a disposizione una vasta gamma di funzionalità che permettono di sviluppare esperienze multiplayer complete e soddisfacenti per gli utenti.

2.2.4. FishNet

Fish-Networking¹², come Mirror, è una libreria open source gratuita. È quindi anch'esso un framework di alto livello che permette di fornire connessioni molto performanti con più giocatori, anche su giochi competitivi, dove la velocità è essenziale. Anche FishNet supporta architetture differenti come P2P e Client-Server e permette di gestire il server in maniera personalizzata. La community di Unity che utilizza questa libreria è attiva e mette a disposizione grandi quantità di risorse. In confronto a Mirror, Fish-Networking è generalmente considerato più performante offrendo una latenza più bassa e una migliore gestione del traffico di rete. Inoltre, Fish-Networking offre un set di funzionalità più completo, che include la gestione automatica di NAT,

¹⁰ Docs-multiplayer.unity3d.com, About Netcode for GameObjects, <https://docs-multiplayer.unity3d.com/netcode/current/about/>, (ultimo accesso 03/01/24)

¹¹ Mirror-networking.gitbook.io, Mirror Networking, <https://mirror-networking.gitbook.io/docs> (ultimo accesso 27/12/23)

¹² Fish-networking.gitbook.io, Fish-Net Introduction, <https://fish-networking.gitbook.io/docs> (ultimo accesso 27/12/23)

il supporto per IPv6 e la compressione dei dati. Al contrario, Mirror Networking presenta un set di funzionalità più basilare, anche se è possibile implementarne di aggiuntive attraverso l'uso di plugin e librerie esterne. Per quanto riguarda la facilità d'uso, Fish-Networking ha un'API più complessa che potrebbe richiedere un periodo di apprendimento più lungo, mentre Mirror Networking vanta un'API più semplice e intuitiva.

2.2.5. Photon Quantum

Photon Quantum¹³, a differenza degli altri elencati precedentemente, è un vero e proprio engine ad alte prestazioni che permette di non utilizzare i Netcode, in quanto tutti gli elementi di gioco sono resi disponibili in rete e sincronizzati automaticamente. Utilizza un approccio di prediction/rollback, ovvero i client effettuano da soli una simulazione del gioco e solo in seguito il server corregge le differenze tra le varie simulazioni che arrivano dai client. Questo motore è costruito su un'architettura ECS (Entity-Component-System) e consente inoltre di effettuare delle verifiche per i fenomeni di cheating, perché le simulazioni vengono analizzate direttamente dal server stesso o effettuando delle verifiche di replay, che possono essere utilizzate in-game o salvate su backend. Photon Quantum è disponibile a pagamento, in base alle prestazioni richiesta e permette inoltre di essere utilizzato con un grande numero di connessioni.

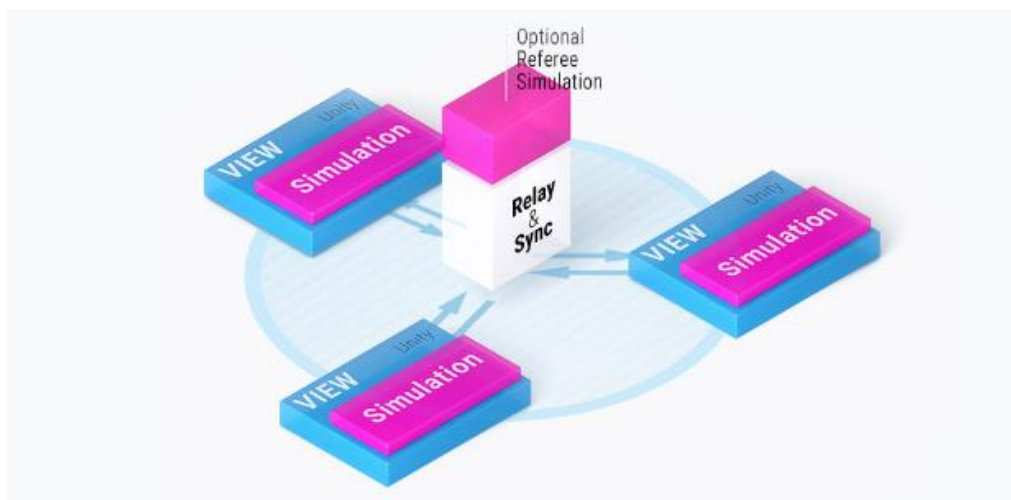


Figura 5 - Funzionamento dell'engine Photon Quantum

¹³ Doc.photoengine.com, Quantum 2 Intro, <https://doc.photonengine.com/quantum/current/quantum-intro> (ultimo accesso 28/12/23)

2.2.6. Photon fusion

Photon fusion¹⁴ è anch'esso un engine sviluppato sempre da Photon, che a differenza di Quantum, utilizza un approccio peer-to-peer. Inoltre, offre un approccio completamente deterministico e non parziale come Quantum. Infatti, mentre il precedente poteva dimostrare dei problemi con la latenza, con Fusion tutti i client hanno lo stesso stato di gioco, perché i client simulano il gioco utilizzando lo stesso input e non è più il server che simula il gioco inviando aggiornamenti al client. In aggiunta Photon fusion risulta essere più performante di quantum, anche se richiede anch'esso un abbonamento o un modello pay-per-use per essere utilizzato.

Di seguito viene riportata una tabella riassuntiva sulle caratteristiche delle soluzioni analizzate.

Soluzione	FishNet	Mirror	Photon Fusion	Photon Quantum	Netcode
Protocollo	RUDP	KCP	TCP	RUDP	UDP
Open source	Si	Si	No	No	Si
Max. giocatori per sessione	200	200	200	500	100
Costo	Gratuito	Gratuito	A pagamento	A pagamento	Gratuito
Prestazioni	Medie	Alte	Elevate	Elevate	Alte
Scalabilità	Buona	Buona	Ottima	Ottima	Ottima
Documentazione	Concisa	Completa	Completa	Completa	Ufficiale

Tabella 1: Confronto tra le soluzioni di networking per Unity

¹⁴ Doc.photonengine.com, Fusion 2 Introduction, <https://doc.photonengine.com/fusion/current/fusion-intro>, (ultimo accesso 27/12/23)

CAPITOLO 3

3. TECNOLOGIE UTILIZZATE

Nel capitolo seguente, verranno visionati dettagliatamente i software impiegati nel corso dello sviluppo del progetto, delineando le loro funzionalità e il ruolo che hanno giocato nella realizzazione del progetto.

3.1. Blender

Blender rappresenta una soluzione avanzata e gratuita per la creazione e la gestione di contenuti 3D. Questo software open source offre un'ampia gamma di strumenti che coprono l'intero processo di produzione, dalla modellazione al rendering, passando per texturing, animazione, simulazione e compositing. Un aspetto distintivo di Blender è la sua flessibilità, che permette agli utenti di estendere le sue funzionalità attraverso script in Python, consentendo la creazione di Add-On personalizzati e la personalizzazione di quelli esistenti sviluppati dalla community.

Nel contesto del nostro progetto, Blender è stato impiegato per la creazione e la modellazione di diversi asset 3D, nonché per la definizione dei loro materiali. In particolare, alcune di queste risorse sono state animate direttamente all'interno del software. Questo processo ha coinvolto la creazione di armature per deformare le mesh degli oggetti e la successiva manipolazione della posizione e dell'orientamento delle ossa attraverso l'uso di keyframes, che permettono di registrare e tracciare variazioni specifiche nel tempo.

3.2. Unity

Unity è uno dei maggiori game engine multiplatforma, utilizzato per la creazione di giochi 2D e 3D. Questo applicativo permette di programmare in C# ed offre la possibilità di sviluppare al suo interno non solo la programmazione logica dell'applicazione, ma anche la modellazione, l'animazione di determinati GameObject.

La versione utilizzata per lo sviluppo del progetto è la 2023.1.0f1 e mette a disposizione dei servizi come il Matchmaking, la Player Identity e la persistenza dei dati, insieme a dei servizi quali hosting, scaling e aggiornamento di server. Tra le soluzioni analizzate nel capitolo precedente per il networking su Unity, quella scelta è stata Unity Netcode, in quanto è la soluzione gratuita integrata che possiede una documentazione ufficiale. Inoltre, il progetto sviluppato non è

pensato per essere utilizzato da più di dieci giocatori; perciò, non necessita di altre soluzioni che supportano attività intensive di networking.

3.2.1. Multiplayer Tools

Il package Multiplayer Tools riveste un ruolo fondamentale nello sviluppo di giochi multiplayer, in quanto fornisce agli sviluppatori una serie di strumenti avanzati per monitorare e ottimizzare le prestazioni del gioco durante la sua esecuzione. Utilizzando la versione 1.1.1 di questo package, è possibile analizzare dettagliatamente l'uso della CPU e il consumo della larghezza di banda, aspetti critici per garantire un'esperienza di gioco fluida e priva di problemi tecnici.

Va sottolineato che, pur essendo un componente estremamente utile, il package Multiplayer Tools è classificato come 'Experimental' da Unity, il che significa che non è completamente supportato. Tra le principali caratteristiche offerte da questo package, troviamo il Network Profiler. Questo strumento consente agli sviluppatori di monitorare in tempo reale i dati trasmessi e ricevuti nel transport layer, fornendo una visione dettagliata delle comunicazioni di rete e facilitando l'identificazione di eventuali problemi o inefficienze. Un altro strumento rilevante è il Runtime Network Stats Monitor (RNSM), che offre una panoramica immediata e dettagliata delle statistiche di rete. Questo include non solo i dati standard provenienti dai pacchetti di rete, ma anche informazioni personalizzate definite dall'utente, permettendo una personalizzazione e un'analisi più approfondita delle prestazioni di rete del gioco. Infine, il Network Simulator è un altro strumento molto utile incluso nel package Multiplayer Tools. Questo permette agli sviluppatori di simulare diverse condizioni di rete, come jitter, perdita e ritardo di pacchetti. Questa funzionalità è particolarmente utile per testare e validare la robustezza del gioco in scenari realistici, aiutando gli sviluppatori a identificare e risolvere potenziali problemi di rete prima del rilascio del prodotto.

3.2.2. Network Manager

Il Network Manager rappresenta un elemento fondamentale per la gestione efficiente della rete all'interno del progetto. Questo componente assume un ruolo centrale nel coordinare le funzionalità di server, client e host, garantendo una sincronizzazione ottimale tra gli utenti e la corretta gestione degli oggetti di rete presenti nelle varie scene del gioco. Una delle caratteristiche distintive del Network Manager è la sua capacità di gestire dinamicamente lo spawn e il despawn dei Network Object, ovvero gli oggetti che necessitano di essere sincronizzati tra i vari client e il server. Questa funzionalità è essenziale per mantenere una coerenza di gioco e assicurare che tutti gli utenti visualizzino gli stessi elementi e le stesse interazioni.

Un aspetto chiave da considerare all'interno del Network Manager è la configurazione del Transport. Questo elemento è responsabile del trasporto dei dati tra i client e il server, e nel contesto del progetto è stato adottato il Networking Transport Layer di Unity, noto come Unity Transport. Questa scelta permette una comunicazione efficiente e sicura tra le diverse componenti di rete, contribuendo a garantire una buona esperienza di gioco per tutti gli utenti.

Inoltre, il Network Manager offre la possibilità di definire un Player prefab, ovvero un modello predefinito del giocatore, che viene automaticamente generato ogni volta che un nuovo client si connette al server. Questa caratteristica semplifica la gestione e la sincronizzazione dei personaggi dei giocatori all'interno del gioco.

Infine, è importante sottolineare l'importanza del componente Network Object e della classe NetworkBehaviour all'interno del Network Manager. Questi elementi sono essenziali per abilitare la sincronizzazione di script e funzionalità specifiche associate ai GameObject del gioco, assicurando che le interazioni e le logiche di gioco siano correttamente replicate e distribuite tra tutti i partecipanti alla sessione di gioco multiplayer.

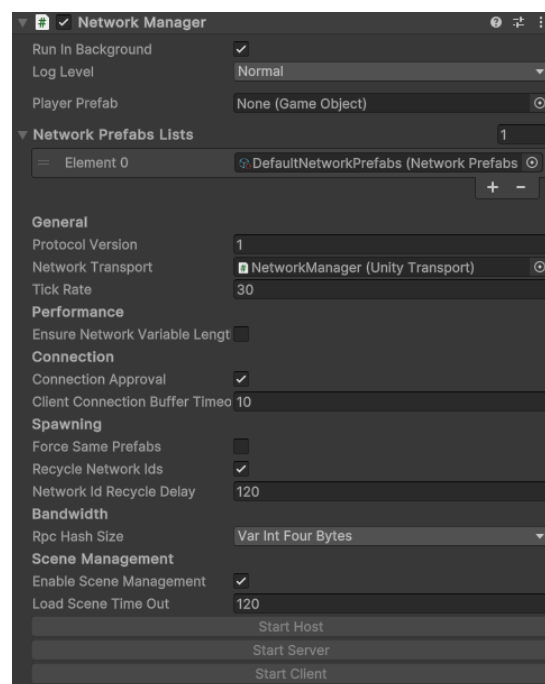


Figura 6: Network Manager nell'Inspector

3.2.4. Network Behaviour

La classe NetworkBehaviour rappresenta un elemento chiave nella struttura di rete del progetto, essendo una classe astratta che consente di gestire le chiamate alle funzioni RPC (Remote Procedure Call) e di mantenere le NetworkVariables. Questo componente abilita la

comunicazione e la sincronizzazione tra i diversi GameObject all'interno dell'ambiente di gioco multiplayer.

Per poter utilizzare le funzionalità RPC su un GameObject, quest'ultimo deve contenere sia un componente NetworkObject che almeno un componente NetworkBehaviour. Questa combinazione permette di stabilire una connessione e di gestire le interazioni tra i diversi elementi del gioco, assicurando che le chiamate alle funzioni RPC siano correttamente indirizzate ed eseguite.

Un aspetto da tenere presente è la relazione tra il componente NetworkBehaviour e il componente NetworkObject. Affinché il componente NetworkBehaviour possa funzionare correttamente, è necessario che il GameObject a cui è associato contenga un componente NetworkObject nello stesso GameObject o in un suo parent. Questa struttura gerarchica consente di mantenere una coerenza e una coesione tra i diversi componenti di rete e di assicurare che le funzionalità di comunicazione siano implementate in modo efficace ed efficiente.

Il metodo OnNetworkSpawn, viene automaticamente invocato su ciascun componente NetworkBehaviour nel momento in cui viene spawnato il corrispondente NetworkObject. Questo metodo rappresenta un punto di inizializzazione fondamentale per il Netcode, consentendo di eseguire le operazioni di setup e di configurazione necessarie per garantire una corretta sincronizzazione e un funzionamento ottimale delle interazioni di rete all'interno del gioco.

3.2.5. Server RPC e Client RPC

Nel contesto dei giochi multiplayer, la comunicazione tra client e server è un aspetto fondamentale per garantire un'esperienza di gioco fluida e coerente. A tale scopo, Unity Netcode introduce due tipologie di procedure chiamate Remote Procedure Calls (RPC): il Server RPC e il Client RPC, ciascuno con caratteristiche e finalità specifiche.

Il Server RPC rappresenta un meccanismo di comunicazione bidirezionale che permette ai client di inviare richieste al server. Una volta ricevuta una richiesta di Server RPC da parte di un client, il server esegue il metodo corrispondente e replica l'azione su tutti gli altri client connessi. Questa modalità è particolarmente adatta per gestire operazioni che richiedono un'autorità centralizzata, come la modifica dello stato globale del gioco o la gestione di eventi critici.

D'altra parte, il Client RPC funge da canale di comunicazione dal server ai client, permettendo di inviare aggiornamenti e sincronizzare informazioni specifiche con tutti i partecipanti alla sessione multiplayer. Questa tipologia di RPC è generalmente utilizzata per trasmettere dati non critici ai client, come la posizione di un personaggio o l'attivazione di effetti visivi, e non richiede una gestione specifica dell'autorità poiché non influisce sullo stato globale del gioco.

Per implementare correttamente un Server RPC o un Client RPC in Unity Netcode, è necessario seguire una convenzione specifica nella definizione dei metodi. Per un Server RPC, il nome del

metodo deve terminare con "ServerRpc" e deve essere preceduto dall'attributo [ServerRpc]. Analogamente, per un Client RPC, il nome del metodo deve concludersi con "ClientRpc" e deve essere preceduto dall'attributo [ClientRpc]. Se si desidera che un client possa invocare un Server RPC, è possibile aggiungere un parametro "RequireOwnership" al metodo e impostarlo su falso. Inoltre, Unity Netcode fornisce strumenti e funzionalità avanzate per gestire l'autorità e garantire che le operazioni RPC vengano eseguite solo dai client autorizzati, contribuendo così a mantenere l'integrità e la coerenza dello stato del gioco durante le sessioni multiplayer.

```
[ServerRpc(RequireOwnership = false)]
2 riferimenti
private void DeliverCorrectRecipeServerRpc()
{
    DeliverCorrectRecipeClientRpc();
}

[ClientRpc]
1 riferimento
private void DeliverCorrectRecipeClientRpc()
{
    pipeCompleted.text = pipeAmount + " / " + CharacterSelectUI.Instance.GetDeliveryValue() / 10;
    pipeAmount++;
    OnPipeBoxed?.Invoke(this, EventArgs.Empty);
}
```

Figura 7: Esempio di chiamata ServerRpc e ClientRpc

3.2.6. Parrel Sync

ParrelSync rappresenta una potente estensione dell'editor di Unity progettata per semplificare e ottimizzare lo sviluppo di giochi multiplayer. Questo strumento consente di testare e sperimentare le funzionalità multiplayer direttamente all'interno dell'editor, eliminando la necessità di compilare e eseguire il progetto ogni volta che si apportano modifiche o aggiornamenti.

L'utilizzo di ParrelSync è intuitivo: è possibile aprire una o più finestre dell'editor che replicano in tempo reale le modifiche apportate al progetto originale. Questo permette di valutare immediatamente l'impatto delle modifiche sul gameplay e di identificare e correggere eventuali problemi o errori con maggiore efficienza e rapidità. La caratteristica principale di ParrelSync è la sua capacità di generare una copia della cartella del progetto, mantenendo un collegamento diretto per le directory degli Asset, i pacchetti e le impostazioni del progetto originale. Questo approccio consente di condividere e sincronizzare le risorse e le configurazioni tra il progetto principale e le sue copie, facilitando la gestione e la manutenzione del codice e delle risorse.

Tuttavia, è importante tenere presente che l'utilizzo di ParrelSync può richiedere una significativa quantità di memoria RAM, poiché lo spazio occupato dal progetto raddoppia per ogni clone creato.

CAPITOLO 4

4. GAME DESIGN

Prima dello sviluppo vero e proprio del progetto è stato necessario analizzare le necessità ed i vincoli posti dalla commissione del serious game. Nel seguente capitolo, quindi, saranno elencati e specificati gli elementi essenziali del serious game, quali: i ruoli dei giocatori, le meccaniche di gioco principali, l'interfaccia utente e il level design.

4.1. Sinossi

Il serious game sviluppato offre un'esperienza educativa e formativa immersiva, mirata a fornire ai giocatori una comprensione pratica delle dinamiche e dei principi fondamentali della Lean Manufacturing. Questo approccio permette ai partecipanti di acquisire competenze essenziali in materia di pianificazione, analisi e collaborazione, attraverso una simulazione realistica e coinvolgente di un ambiente operativo industriale.

Nel contesto del gioco, dieci giocatori assumono ruoli specifici all'interno di una fabbrica di tubi, ciascuno con compiti ben definiti nel processo di produzione. Gli utenti devono collaborare strategicamente tra di loro, ottimizzando le risorse disponibili. In questo modo i giocatori sono stimolati ad utilizzare al meglio le proprie competenze e a coordinarsi efficacemente con gli altri membri del team per raggiungere gli obiettivi comuni.

Il primo livello del gioco introduce i giocatori a una catena di produzione caratterizzata da un flusso lineare, dove ogni pezzo attraversa le diverse stazioni di lavoro in modo sequenziale, minimizzando gli sprechi e massimizzando l'efficienza operativa. Il Product Manager, nel ruolo di supervisore della partita, monitora attentamente i Key Performance Indicators (KPIs) al termine del livello, valutando il numero di pezzi prodotti in tempo e gli errori commessi nelle varie fasi del processo. Questa analisi dettagliata consente al Product Manager di identificare aree di miglioramento e di riorganizzare la partita in vista del secondo livello, introducendo eventuali aggiustamenti e ottimizzazioni per affrontare sfide sempre più complesse e stimolanti.

Nel secondo livello, i giocatori si confrontano con un layout rinnovato della catena di produzione e una nuova distribuzione dei lavoratori. In questa fase avanzata del gioco, vengono introdotti gli strumenti Lean, permettendo ai giocatori di approfondire le proprie conoscenze e di applicare tecniche e metodologie Lean per ottimizzare ulteriormente il processo produttivo. Questa evoluzione del gioco stimola i giocatori a sviluppare strategie innovative, per raggiungere livelli di eccellenza operativa.

In sintesi, il serious game offre un percorso di apprendimento dinamico e interattivo, progettato per sviluppare competenze pratiche e soft skills fondamentali nel campo della Lean Manufacturing, promuovendo la crescita professionale e l'innovazione continua tra i partecipanti.



Figura 8: Tubo finale da produrre

4.1.1. Player

Il serious game prevede la partecipazione di dieci utenti, ciascuno con un ruolo specifico all'interno della catena di produzione. In particolare:

- Il Product Manager è responsabile della gestione dell'assegnazione dei ruoli tra gli utenti, regola la durata delle partite ed emette gli ordini di produzione in base a una pianificazione. Al termine di ciascun livello, il Product Manager si occupa anche dell'analisi dei KPI (Indicatori Chiave di Prestazione) della produzione, esaminando dati relativi al numero di pezzi prodotti in tempo e al numero di errori commessi in ogni stazione di lavoro.
- Il magazziniere che svolge un ruolo fondamentale nell'efficiente funzionamento della catena di produzione. Nel primo livello ha l'incarico di inviare i pezzi necessari a ciascuna stazione. Tuttavia, la sfida aggiuntiva consiste nel poter inviare solo un pezzo alla volta a una stazione specifica, richiedendo il ritiro da parte degli altri giocatori prima di procedere con l'invio successivo. Questa dinamica è progettata per promuovere la collaborazione e l'ottimizzazione della gestione dei materiali. Nel secondo livello, il magazziniere assume una nuova dinamica, avendo la capacità di collocare quanti pezzi desidera all'interno di ogni stazione senza dover attendere il ritiro da parte degli altri giocatori. Questa modifica mira a introdurre una maggiore flessibilità nella gestione dei materiali.

- I lavoratori, che sono assegnati individualmente a diverse workstation, ciascuna dedicata all'unione di due tubi. Questi operatori possono prendere i tubi necessari dal magazzino o dalle altre workstation. Nel primo livello, seguono la catena di produzione lineare, partecipando attivamente all'assemblaggio dei tubi lungo il percorso stabilito. La loro collaborazione è essenziale per garantire un flusso continuo e senza intoppi nella produzione. Nel secondo livello, la dinamica cambia, e i lavoratori utilizzano dei supermarket designati per accumulare i tubi prodotti. Questi supermarket forniscono una capacità di stoccaggio che consente di accumulare più pezzi, ottimizzando il flusso di produzione e riducendo eventuali ritardi.
- Il controllo qualità assume un ruolo chiave nell'assicurare la conformità del tubo finale. Il responsabile del controllo qualità valuta attentamente ciascun tubo prodotto, verificandone la correttezza. Nel caso in cui un tubo non soddisfi gli standard di qualità prestabiliti, il controllo qualità lo scarta. Inoltre, questo utente è anche l'addetto alla consegna, in quanto è incaricato di consegnare il pezzo finale, il tubo completo. La sua responsabilità cruciale è garantire che il pezzo finale raggiunga con successo l'obiettivo prestabilito all'inizio della partita, come definito dal Product Manager.

4.2. Meccaniche di gioco

Il serious game si articola attraverso diverse fasi e meccaniche di gioco che mirano a simulare una catena di produzione reale, promuovendo la collaborazione, la pianificazione e l'ottimizzazione delle risorse tra i giocatori. Di seguito, una descrizione delle principali meccaniche di gioco:

- Inizialmente, prima dell'avvio della partita, il Product Manager è responsabile dell'assegnazione dei ruoli. Ogni giocatore viene assegnato a una specifica posizione, che può variare dal magazziniere, ai lavoratori alle workstation, fino al controllo qualità. Ogni ruolo è dotato di responsabilità uniche e fondamentali all'interno del processo produttivo, contribuendo al funzionamento armonioso dell'intera catena di produzione.
- Successivamente, il Product Manager crea la partita definendo vari parametri, come il numero di pezzi da produrre, la durata del livello e il numero di bot che possono sostituire eventuali assenze. Una volta creato l'ambiente di gioco, gli altri utenti ricevono la chiave d'accesso alla lobby per partecipare.
- Il primo livello si concentra sulla produzione a flusso lineare. Qui, il magazziniere svolge un ruolo centrale, inviando un pezzo alla volta alle stazioni di lavoro. L'aspetto tridimensionale del gioco offre un'esperienza immersiva, permettendo ai giocatori di interagire direttamente con i tubi durante l'assemblaggio.

- Al termine del primo livello, il Product Manager esamina attentamente i KPIs, come il numero di pezzi prodotti in tempo e gli errori rilevati nelle stazioni di lavoro. Questa analisi fornisce informazioni preziose per la riorganizzazione e la pianificazione del secondo livello, dove i giocatori hanno l'opportunità di comunicare e adottare strategie basate sulle pratiche della produzione Lean.
- Nel secondo livello, la dinamica cambia: i giocatori utilizzano "supermarket" designati per accumulare i tubi, introducendo nuove sfide legate alla pianificazione e all'ottimizzazione della produzione. Questa fase richiede una maggiore flessibilità nella gestione dei materiali, sfidando i giocatori a coordinarsi efficacemente per evitare ritardi e interruzioni nella catena produttiva.
- Il reparto di controllo qualità assume un ruolo cruciale nel garantire la conformità dei tubi prodotti. Ogni pezzo viene attentamente esaminato e, se non soddisfa gli standard qualitativi, viene scartato. Inoltre, il controllo qualità è anche responsabile della consegna del tubo completo, assicurandosi che l'obiettivo stabilito dal Product Manager venga raggiunto con successo.

In sintesi, il serious game offre una simulazione dettagliata e coinvolgente di una catena di produzione, mettendo alla prova le capacità organizzative, decisionali e collaborative dei giocatori in un ambiente virtuale.

4.3. Level design

Il design visivo del serious game è stato pensato per garantire una chiarezza e una leggibilità ottimali, sfruttando la versatilità e le performance offerte dallo stile low poly. Questa scelta estetica, caratterizzata da modelli poligonali semplici e stilizzati, non solo conferisce al gioco un aspetto distintivo, ma svolge anche un ruolo fondamentale nell'ottimizzazione delle risorse grafiche. In un contesto multiplayer come quello del serious game, dove la creazione simultanea di numerosi GameObject può mettere a dura prova le capacità di elaborazione, l'approccio low poly si rivela molto utile per mantenere un'esperienza fluida e priva di lag.

Il design low poly, con la sua ridotta complessità grafica, permette di gestire in modo efficiente la manipolazione di un elevato numero di vertici, garantendo prestazioni stabili e reattive anche nelle fasi più intense del gioco. Questa scelta consapevole ha dunque un impatto positivo sull'efficienza complessiva del gioco, contribuendo a una migliore fruibilità e a un'esperienza di gioco più gradevole per tutti i partecipanti.

Prima di procedere con lo sviluppo del design dei livelli, è stata curata la selezione di una palette di colori coerente, pensata per migliorare ulteriormente l'estetica del gioco e facilitare la comprensione delle dinamiche di gioco. In particolare, sono state definite due palette distintive: una destinata alla mappa di gioco e l'altra riservata ai giocatori. Quest'ultima è stata progettata

tenendo conto dei diversi ruoli assegnati ai partecipanti, associando a ciascun utente un colore specifico che ne identifica il ruolo all'interno della catena di produzione. Questa scelta cromatica non solo contribuisce a enfatizzare l'identità visiva dei giocatori, ma agevola anche il riconoscimento rapido dei diversi ruoli, favorendo la collaborazione e la comunicazione tra i membri del team durante la partita.



Figura 9: Palette per l'ambiente

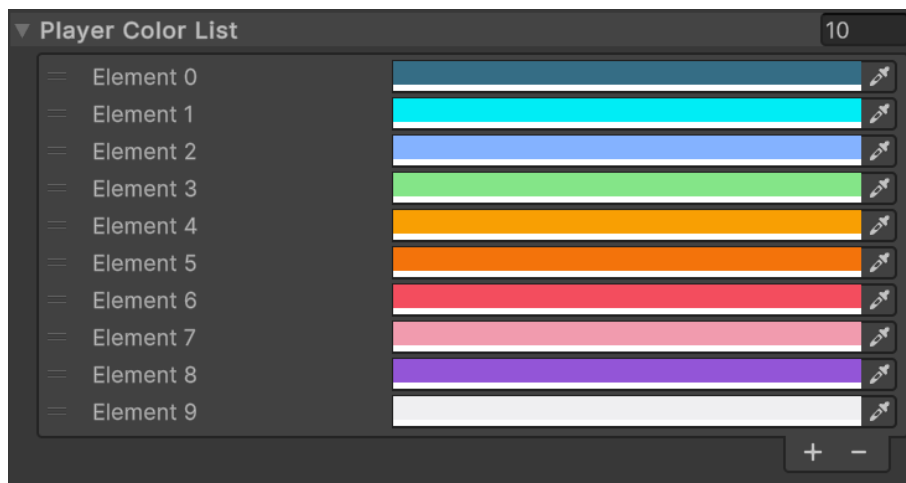


Figura 10: Colori per i giocatori

Ogni colore è stato scelto con attenzione per garantire un riconoscimento visivo immediato e intuitivo degli elementi chiave all'interno del contesto produttivo:

- **Magazzino:** un colore specifico è stato assegnato per distinguere chiaramente gli spazi dedicati allo stoccaggio dei materiali, facilitando la localizzazione e l'identificazione rapida degli oggetti necessari per la produzione.
- **Workstation:** ciascuna postazione è caratterizzata da un colore distinto, facilitando il riconoscimento e la differenziazione tra le diverse stazioni all'interno della catena di produzione.
- **Ambiente circostante:** è stata adottata una palette cromatica neutra e sobria per l'ambiente circostante. Questa scelta contribuisce a creare un'atmosfera coerente e immersiva, senza distogliere l'attenzione dalle dinamiche principali e dalle attività focali del gioco.

L'uso equilibrato dei colori in questi diversi contesti non solo arricchisce l'estetica visiva del gioco, ma funge anche da guida visiva per i giocatori, offrendo indicazioni chiare e intuitive sulla natura e la funzione delle diverse aree di gioco.

Inoltre, la disposizione strategica delle stazioni di lavoro, suddivise da mura ma con spazi aperti che favoriscono la comunicazione, contribuisce ulteriormente a una migliore esperienza di gioco. Ogni stazione è chiaramente identificata da una scritta posizionata sul pavimento, assicurando ai giocatori un facile orientamento e delineando chiaramente il percorso da seguire durante la produzione. Questa progettazione attenta e intuitiva della mappa di gioco agevola la fluidità della catena di produzione, migliorando la coesione di squadra e contribuendo a una esperienza di gioco più fluida e coinvolgente per tutti i partecipanti.

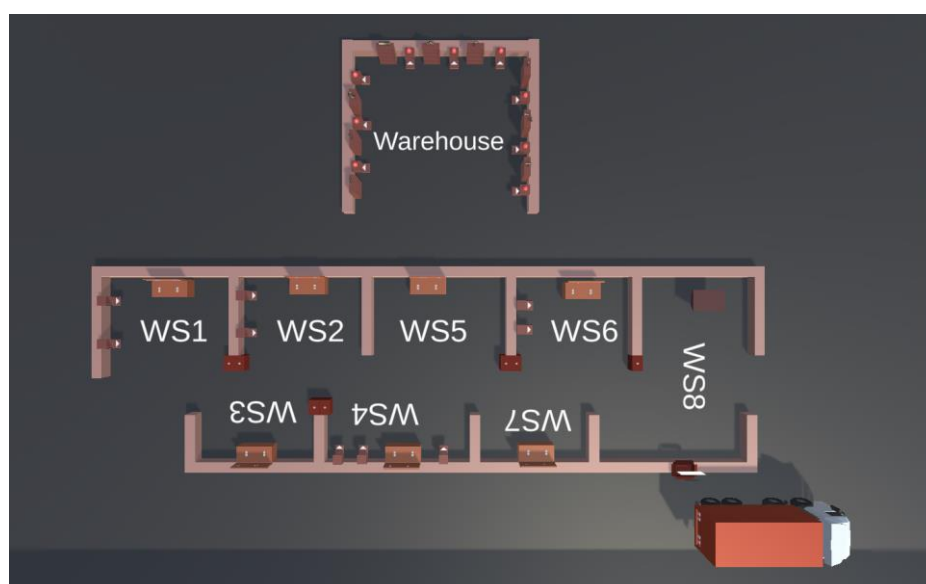


Figura 11: Mappa vista dall'alto

La maggior parte degli asset, compresi i vari elementi ambientali e di produzione, sono stati modellati e progettati utilizzando Blender, un potente strumento di grafica 3D.

Tuttavia, per alcuni elementi specifici, è stata fatta una scelta di integrazione con risorse esterne per risparmiare tempo e sfruttare risorse di alta qualità già disponibili sul mercato. Ad esempio, i tubi¹⁵ utilizzati nel gioco sono stati selezionati e scaricati dall'Asset Store di Unity, forniti dall'azienda Synarea. Questa scelta ha consentito di integrare rapidamente elementi dettagliati e ben progettati nel gioco, risparmiando tempo nel processo di sviluppo.

Un altro elemento non modellato personalmente è il furgoncino¹⁶, anch'esso acquisito dall'Asset Store per assicurare una consistenza stilistica con gli altri elementi del gioco.

¹⁵ Assetstore.unity.com, *Pipe-constructor*, <https://assetstore.unity.com/packages/3d/environments/industrial/pipe-constructor-186974>, (ultimo accesso 25/01/24)

¹⁶ Assetstore.unity.com, *Low Poly Cars*, <https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-cars-101798>, (ultimo accesso 25/01/24)

Tra gli asset modellati, il supermarket e il trash bin si distinguono per l'aggiunta di animazioni interattive. Queste animazioni, realizzate direttamente in Blender, simulano l'apertura e la chiusura di questi oggetti, arricchendo l'esperienza di gioco con dettagli dinamici e interattivi. Dopo la creazione, queste animazioni sono state esportate e integrate in Unity, dove sono state gestite e sincronizzate con le interazioni del giocatore per garantire una rappresentazione fluida e coinvolgente nel contesto del gioco.

In conclusione, l'approccio combinato di utilizzo di asset personalizzati e di risorse preesistenti disponibili sull'Asset Store ha permesso di realizzare un'esperienza di gioco coerente, dettagliata e visivamente accattivante, ottimizzando allo stesso tempo l'efficienza e la produttività durante il processo di sviluppo del gioco.

4.4. Interfaccia utente

L'attenzione alla coerenza stilistica e all'usabilità ha guidato la progettazione dell'interfaccia utente (UI) del gioco, risultando in un design pulito e intuitivo che si sposa perfettamente con l'estetica low poly dell'ambiente di gioco. La scelta di adottare un design essenziale e privo di dettagli superflui per gli elementi grafici della UI è stata fondamentale per assicurare una chiarezza visiva e una facilità di navigazione ottimali, soprattutto nei menu e nelle schermate di configurazione.

Una caratteristica distintiva dell'approccio alla UI del gioco è stata l'integrazione di alcuni elementi grafici direttamente nell'ambiente tridimensionale del gioco durante il gameplay dei livelli. Questo approccio permette ai giocatori di accedere alle informazioni utili in modo intuitivo e diretto, senza dover interrompere l'esperienza di gioco per consultare un'interfaccia separata.

Ad esempio, un timer visivamente integrato con l'ambiente tridimensionale del gioco fornisce ai giocatori un riscontro immediato sul tempo rimanente per completare il livello, mantenendo l'attenzione focalizzata sull'azione senza distrarre con elementi UI invasivi.

In modo simile, un indicatore visuale che mostra il numero di tubi già consegnati e quelli ancora da consegnare è stato integrato nell'ambiente di gioco. Questo elemento permette ai giocatori di monitorare il proprio progresso e di pianificare le proprie mosse strategiche in tempo reale, mantenendo al contempo la coerenza stilistica e visiva con il resto del gioco low poly.

In conclusione, l'approccio olistico e ben ponderato alla progettazione dell'interfaccia utente ha permesso di creare un'esperienza di gioco fluida, intuitiva e immersiva, dove la UI si fonde armoniosamente con l'ambiente di gioco, garantendo al contempo una chiarezza informativa e una facilità di navigazione ottimali per i giocatori.

CAPITOLO 5

5. SVILUPPO

In questo capitolo, verranno analizzate le funzionalità del serious game sviluppate su Unity. La struttura del capitolo è suddivisa in base alla sequenza delle scene realizzate; in particolare, si esamineranno la scena del Menu principale, della Lobby, della Configurazione del livello e scelta del ruolo, del tutorial e dei due livelli. Si inizierà esplorando la scena del Menu principale, dove i giocatori hanno accesso alle opzioni di avvio del gioco, alle impostazioni e alle informazioni generali sul serious game. Successivamente, si analizzerà la scena della Lobby, dove i giocatori possono connettersi, interagire e prepararsi per l'inizio della partita, scegliendo il ruolo da ricoprire e stabilendo la configurazione del livello di gioco. Si proseguirà con l'esame della scena dedicata alla Configurazione del livello e alla scelta del ruolo, dove i giocatori hanno la possibilità di personalizzare le impostazioni del livello e di selezionare il ruolo che desiderano assumere durante la partita. Questa sezione del gioco è fondamentale per garantire un'esperienza di gioco equilibrata e coinvolgente, permettendo ai giocatori di sperimentare diverse dinamiche di gioco e strategie. Successivamente, si analizzerà la scena del tutorial, dove i nuovi giocatori hanno l'opportunità di familiarizzare con le regole, le meccaniche di gioco e le interazioni di base, preparandosi al meglio per affrontare i livelli successivi con sicurezza e competenza. Infine, verranno esplorate le due scene dei livelli di gioco, dove vengono presentate e messe in pratica tutte le funzionalità e le dinamiche di gioco precedentemente analizzate. Si esaminerà la struttura, le meccaniche e le sfide proposte in ciascun livello, focalizzandosi sull'esperienza di gioco, sull'interazione tra i giocatori e sull'efficacia delle strategie di gioco implementate.

Attraverso questa strutturata analisi delle diverse scene e funzionalità del serious game, si avrà l'opportunità di comprendere in profondità il processo di sviluppo, le scelte progettuali e le soluzioni implementate per garantire un'esperienza di gioco stimolante, educativa e coinvolgente per tutti i partecipanti.

5.1. Menu principale

Dal punto di vista grafico, la schermata principale dell'applicazione presenta quattro bottoni all'apertura: un pulsante per ciascun livello e due bottoni con icone che rappresentano l'uscita dal gioco e l'apertura del menu delle impostazioni. In questa schermata, due script giocano un ruolo fondamentale nel funzionamento del gioco. Il primo script gestisce il reset dei dati relativi ai giocatori e alle workstation ad ogni nuovo avvio della partita. Questa funzionalità assicura che ogni sessione di gioco inizi con un ambiente pulito, eliminando eventuali dati residui o anomalie dalle partite precedenti.

Il secondo script si occupa della distruzione delle istanze precedentemente create legate alla rete. Questo processo è cruciale per garantire un'esperienza fluida e priva di errori agli utenti che hanno creato o aderito a una partita. Eliminando le istanze precedenti, il sistema consente agli utenti di accedere nuovamente alla stessa partita o di crearne una nuova utilizzando lo stesso indirizzo IP, senza conflitti o complicazioni legate alle sessioni di gioco precedenti. Entrambi gli script contribuiscono a ottimizzare e garantire l'affidabilità del sistema, assicurando che i giocatori possano godere di un'esperienza di gioco stabile, coerente e priva di interruzioni indesiderate.

5.2. Lobby

Prima di immergersi nel vero e proprio gameplay, gli utenti vengono accolti da una schermata in cui possono creare o unirsi a una lobby e scegliere un nome utente. Per sfruttare al meglio le funzionalità delle lobby fornite da Unity, è indispensabile installare il pacchetto Lobby disponibile nel registro di Unity. Una volta completata l'installazione del pacchetto, è stato cruciale associare il nostro progetto a un Project ID. Questo passaggio è fondamentale per poter usufruire pienamente dei servizi cloud e delle funzionalità messe a disposizione da Unity Gaming Services. All'interno del progetto che abbiamo sviluppato, abbiamo fatto uso specifico dei seguenti servizi forniti da Unity: Lobby, Player Authentication e Relay. Il servizio di Player Authentication riveste un'importanza particolare. Infatti, molte applicazioni necessitano dell'identificazione dell'utente per assicurare la sicurezza e la coerenza dei dati elaborati.

Quanto al servizio di Relay, come abbiamo evidenziato nel capitolo dedicato allo stato dell'arte, esso permette di stabilire una connessione tra i giocatori. Questo garantisce un'esperienza di gioco multiplayer fluida e coinvolgente, senza l'esigenza di disporre di un server dedicato costoso. Relay facilita le comunicazioni peer-to-peer e supporta listen-server UDP, assicurando un collegamento efficiente, semplice e sicuro tra i partecipanti.

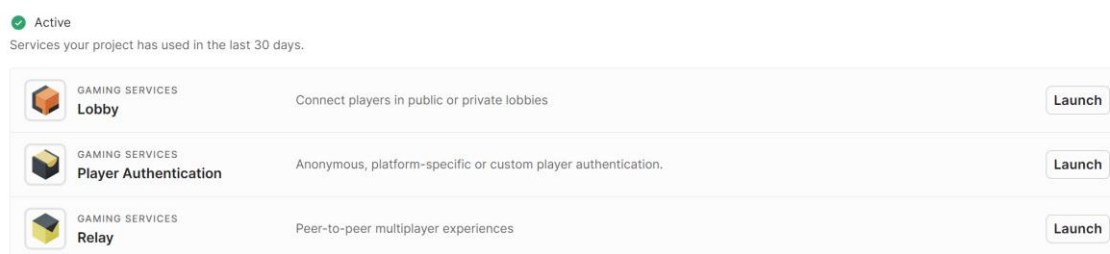


Figura 12: Gaming Services utilizzati

All'interno di questa schermata iniziale del gioco, troviamo una serie di script fondamentali che gestiscono diverse funzionalità cruciali. Tra questi script, spiccano il NetworkManager, il LobbyManager e un terzo script chiamato GameMultiplayer, dedicato alla gestione dell'aspetto

multiplayer del gioco. Ognuno di questi script include una funzione Awake che viene chiamata durante il caricamento dell'istanza dello script. Utilizzando la funzione DontDestroyOnLoad(), viene garantito che questi script siano persistenti e possano essere trasferiti nelle scene successive senza essere eliminati o inizializzati nuovamente. Questo è particolarmente utile per mantenere lo stato e le configurazioni del gioco tra diverse fasi o livelli.

Gli UnityServices vengono inizializzati all'interno di questi script, se non sono già stati avviati in precedenza. Questo approccio consente di eseguire contemporaneamente più build del gioco sulla stessa macchina, facilitando il processo di test e debug dell'applicazione.

Un aspetto chiave gestito da questi script è la possibilità per i giocatori di unirsi a una lobby esistente. Sono anche gestite le disconnessioni dei giocatori, che possono essere causate sia da un'uscita volontaria dalla lobby che da problemi di connessione. Inoltre, l'host della partita ha il potere di espellere un giocatore dalla lobby nel caso in cui si sia unito accidentalmente o ci siano altre ragioni valide.

Nell'interfaccia utente della lobby, viene visualizzato il nome del giocatore. Questo nome viene aggiornato dinamicamente all'interno dello script di gestione del multiplayer quando ci si sposta nella scena di gioco principale. Inoltre, attraverso questo script, viene definito anche il numero massimo di giocatori che possono partecipare alla lobby, assicurando una gestione ottimale delle sessioni multiplayer.

All'interno dello script del LobbyManager, la prima fase riguarda l'inizializzazione del servizio di autenticazione di Unity, che è essenziale per l'identificazione dell'utente. Inoltre, vengono eseguiti controlli periodici per gli aggiornamenti, tra cui l'elenco delle lobby disponibili. Se l'utente non è già in una lobby, viene fornita una procedura per unirsi a una lobby esistente. Nel caso in cui l'utente funga da host della lobby, viene inviato periodicamente un ping di heartbeat al server, un piccolo pacchetto di dati inviato regolarmente da un dispositivo o software ad un altro per verificare che sia ancora attivo e funzionante, per mantenere attiva la connessione.

Ecco una descrizione dettagliata delle funzioni presenti nello script del LobbyManager:

- **CreateLobby:** Questa funzione è responsabile della creazione di una nuova lobby. Inizia generando una lobby con un nome specifico e allocando un server Relay dedicato per la comunicazione di rete. Successivamente, ottiene e memorizza il codice di join associato a questo server nella lobby. Infine, configura il Network Manager per utilizzare il server Relay e avvia il gioco come host, caricando la scena corrispondente. Questo processo assicura una transizione fluida dalla creazione della lobby al gioco attivo, garantendo una corretta sincronizzazione delle comunicazioni tra i giocatori.
- **JoinWithCode:** Questa funzione consente di unirsi a una lobby specifica utilizzando un codice personalizzato fornito dall'host. Dopo aver ricevuto il codice, si connette al server Relay associato alla lobby utilizzando quel codice. Una volta stabilita la connessione,

configura il Network Manager per utilizzare il server Relay e avvia il gioco come client, preparandosi a partecipare alla lobby e caricando la scena di gioco appropriata.

- DeleteLobby: Questa funzione elimina la lobby a cui l'utente è attualmente unito. Verifica se l'utente è l'host della lobby prima di procedere all'eliminazione, garantendo che solo l'host possa cancellare la lobby.
- LeaveLobby': Questa funzione consente all'utente di abbandonare la lobby a cui è attualmente unito, offrendo una maggiore flessibilità nella gestione della partecipazione alle sessioni multiplayer.
- KickPlayer: Questa funzione è progettata per essere utilizzata esclusivamente dall'host della lobby e consente di rimuovere un giocatore specifico dalla lobby in base al suo ID. Questo permette all'host di mantenere il controllo sulla composizione della lobby, rimuovendo giocatori indesiderati o gestendo situazioni particolari durante la preparazione del gioco.

In sintesi, lo script del LobbyManager gestisce in modo completo e dettagliato tutte le operazioni legate alla creazione, gestione e disconnessione delle lobby nel contesto del tuo gioco multiplayer, assicurando una esperienza di gioco fluida e ben coordinata per tutti i partecipanti.

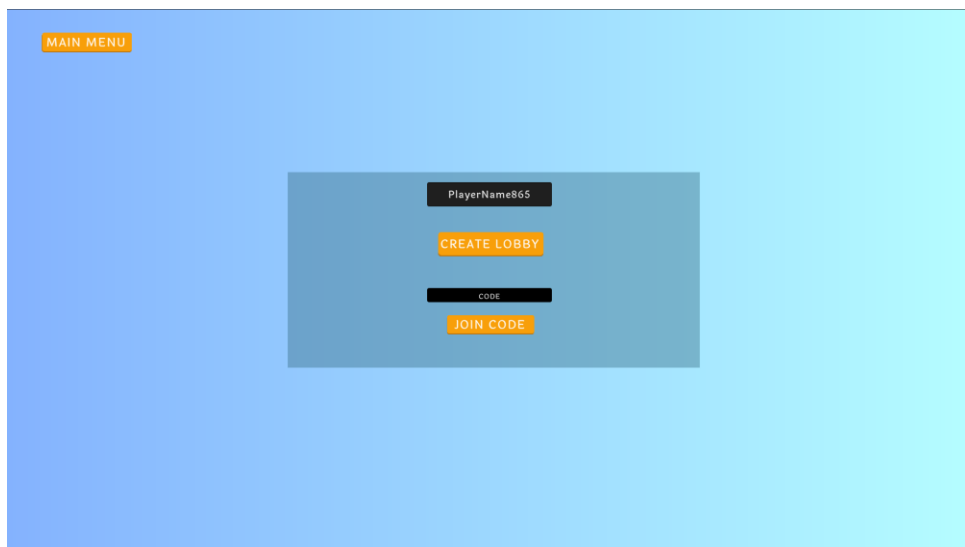


Figura 13: Scena della Lobby

5.3. Configurazione livello

La configurazione del livello presenta variazioni significative a seconda che l'utente che ne sta visualizzando la scena sia l'host o uno dei giocatori che si sono uniti alla lobby. Nel caso dell'host, la sua interfaccia utente sarà più complessa, poiché avrà la responsabilità di impostare varie

opzioni in qualità di project manager della partita. Questo include la possibilità di personalizzare e definire diverse impostazioni cruciali per lo svolgimento del gioco.

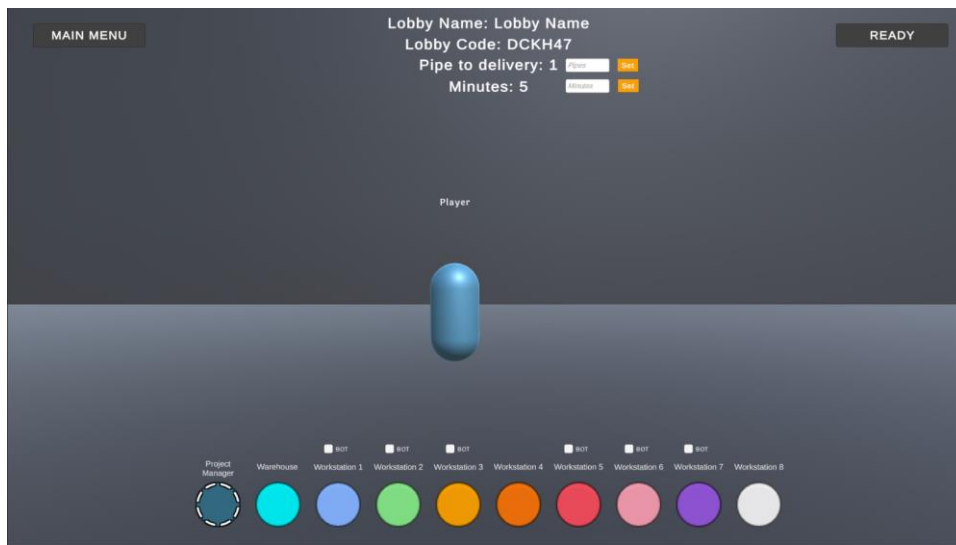


Figura 14: Interfaccia per la configurazione del livello

Durante la fase di impostazione della partita, la schermata dell'host presenta il proprio avatar e il nome associato. Nella parte superiore della schermata, sono visualizzati il nome della lobby, il codice di invito per gli altri giocatori, il numero di oggetti da raccogliere e la durata stimata della partita in minuti. Ogni elemento dispone di un campo di input numerico per facilitare la personalizzazione delle impostazioni, accettando solo valori interi. Per facilitare la selezione dei ruoli, sono disponibili pulsanti circolari nella parte inferiore dello schermo, con ciascun colore associato a un ruolo specifico definito dal product manager prima dell'inizio della partita.

È anche possibile aggiungere bot controllati dall'intelligenza artificiale alle workstation, nel caso in cui il numero di giocatori sia inferiore ai dieci previsti. Una volta condiviso il codice della lobby, i giocatori iniziano a comparire sulla schermata, ciascuno con il proprio nome di gioco. Accanto a ogni giocatore, è presente un'icona 'X' che consente all'host di espellere eventuali giocatori indesiderati. Se un giocatore desidera uscire, può farlo tornando al menu principale attraverso un pulsante situato in alto a sinistra. Al contrario, se è pronto per iniziare, può selezionare il tasto 'Ready'. Una volta che tutti i giocatori hanno confermato di essere pronti ad iniziare, la scena cambia per avviare il livello di gioco selezionato.

Nel caso di disconnessione dell'host, agli altri giocatori viene mostrata una schermata di avviso che comunica l'evento. Un messaggio informa gli utenti della disconnessione dell'host e li invita a tornare al menu principale. Questa notifica consente agli altri giocatori di gestire la situazione, decidendo se aspettare il ritorno dell'host o procedere con altre opzioni disponibili nel menu principale del gioco.

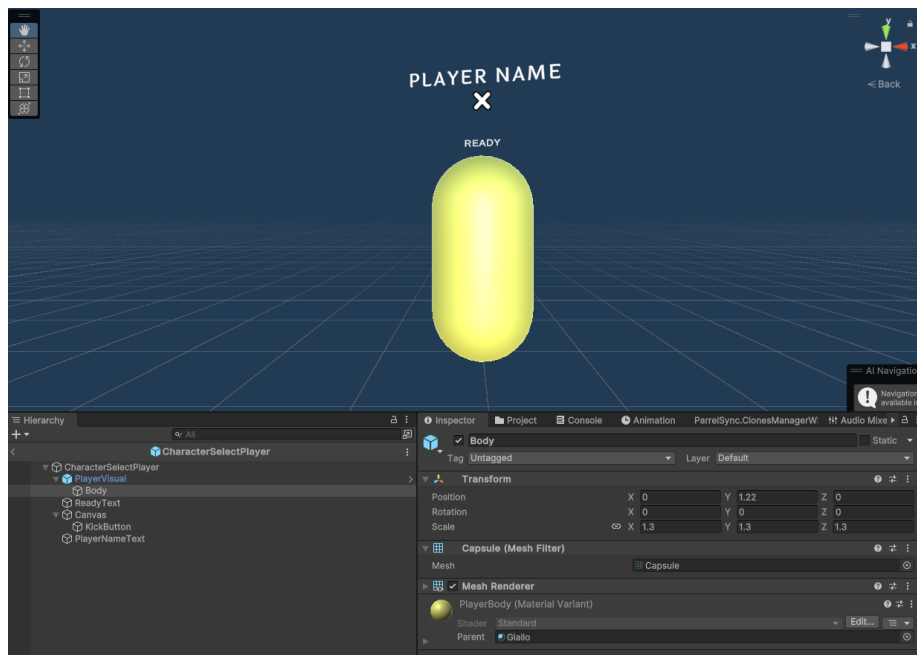


Figura 15: Struttura del player nella scena di configurazione

La scena è strutturata con diverse Canvas che gestiscono le interfacce per i client e l'host, oltre ai prefabs dei giocatori che vengono generati dinamicamente al momento della connessione di un utente. Nonostante la sua apparente semplicità visiva, questa configurazione richiede una gestione avanzata del networking. È fondamentale garantire una sincronizzazione tempestiva dei dati tra tutti gli utenti per assicurare che le azioni e le scelte effettuate siano percepite in modo coerente e immediato da tutti i partecipanti. Questo introduce una notevole complessità nella gestione delle connessioni e delle comunicazioni in tempo reale tra i diversi nodi della rete.

Lo script 'GameMultiplayer' svolge un ruolo cruciale nella gestione delle dinamiche di networking. Presente non solo in questa scena ma anche nelle successive, lo script non viene distrutto durante i cambiamenti di scena. È in questo script che vengono definiti il numero massimo di giocatori consentiti e viene aggiornato il nome dei giocatori in base alle informazioni fornite da ciascun utente. Questo permette di mantenere una coerenza tra le diverse sessioni di gioco e di gestire in modo efficace le interazioni tra i partecipanti all'interno dell'ambiente multiplayer.

Lo script 'PlayerData' è progettato per memorizzare le informazioni associate a ciascun giocatore ed è strutturato come una classe che integra due interfacce principali. La prima interfaccia permette di confrontare due istanze di 'PlayerData' per verificarne l'uguaglianza, mentre la seconda interfaccia, chiamata 'INetworkSerializable', gestisce la serializzazione e deserializzazione dei dati per facilitare la comunicazione di rete. Dato che le strutture sono tipi di valore, vengono memorizzate direttamente in memoria, a differenza delle classi che sono tipi di riferimento.

All'interno della struttura 'PlayerData', sono definite diverse variabili essenziali, tra cui un ID univoco per identificare il client associato al giocatore, un ID per il colore scelto dall'utente e il nome del giocatore stesso. Questa classe gioca un ruolo cruciale nel gestire e comunicare le informazioni specifiche di ciascun giocatore all'interno dell'ambiente di gioco multiplayer.

Nello script di inizializzazione del giocatore, nel metodo Start, sono implementate delle logiche che permettono solo al server di avere accesso e controllo sui bottoni e sugli InputField utilizzati per configurare i dati del giocatore. Successivamente, queste informazioni vengono trasferite al componente 'GameMultiplayer' attraverso funzioni specifiche. L'obiettivo di questa progettazione è garantire che il server abbia l'autorità esclusiva sulla gestione delle configurazioni dei giocatori e che i dati risultanti siano distribuiti in modo uniforme e sincronizzato tra tutti i client.

Questo approccio architetturale è tipico nelle applicazioni multiplayer, dove il server agisce come punto focale per coordinare e sincronizzare le operazioni tra i vari client. Utilizzando funzioni dedicate per il trasferimento dei dati, si assicura una trasmissione affidabile e una condivisione uniforme delle informazioni tra tutti gli utenti connessi, garantendo così un'esperienza di gioco coesa e sincronizzata per l'intera durata della partita.

5.4. Tutorial

Prima di iniziare a giocare effettivamente, gli utenti hanno la possibilità di completare un tutorial progettato per offrire una comprensione dettagliata delle meccaniche di gioco. Questo tutorial è strutturato per guidare gli utenti attraverso le diverse fasi di gioco e per aiutarli a familiarizzare con le funzionalità e gli elementi chiave del gioco prima di iniziare a giocare contro altri utenti.

Il tutorial inizia con una chiara e dettagliata spiegazione su come creare una lobby e sull'utilizzo degli elementi disponibili nell'interfaccia per la creazione dei livelli. Una volta che gli utenti hanno acquisito una comprensione di base di queste funzionalità, vengono introdotti nella mappa del tutorial, che presenta gli oggetti e gli elementi essenziali del gioco.

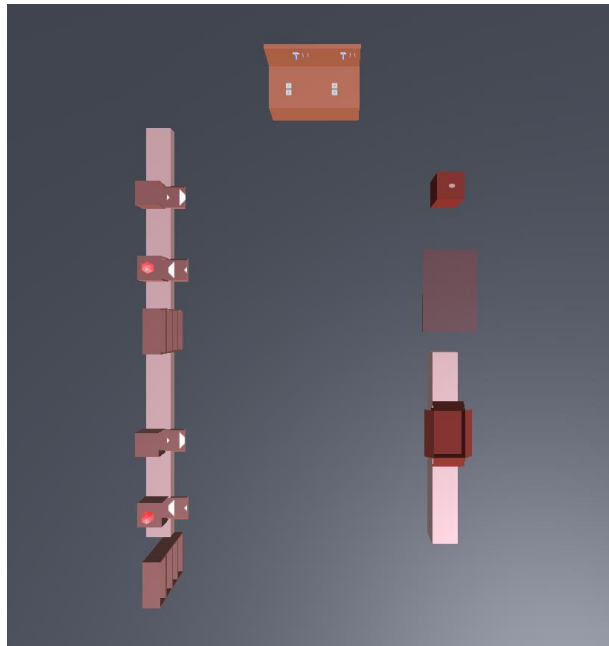


Figura 16: Mappa del tutorial

Il tutorial è diviso in tre fasi distinte, ognuna delle quali introduce e spiega specifici comandi e meccaniche di gioco. Nella prima fase del tutorial, agli utenti viene assegnato il compito di prelevare un tubo da uno scaffale e di posizionarlo all'interno di una scatola situata nelle vicinanze dello scaffale. Questo esercizio serve a familiarizzare gli utenti con il ruolo del magazziniere nel gioco e a illustrare come, una volta che il tubo viene collocato nella scatola designata, una lampadina si spegne e il tubo viene automaticamente trasferito in un'altra scatola.

Lo scaffale in questa fase del tutorial è gestito da uno script chiamato 'Warehouse'. Questo script è una classe pubblica che eredita le funzionalità da 'WorkStation'. La classe 'WorkStation', a sua volta, eredita da 'NetworkBehaviour' e implementa l'interfaccia 'IPipeParent', che si occupa di gestire i genitori dei tubi nel gioco. All'interno dello script 'WorkStation', è presente un evento statico che viene attivato ogni volta che un oggetto viene posizionato sulla workstation, una variabile per memorizzare un riferimento a un componente di tipo 'Light', se presente, e un booleano che tiene traccia dello stato della luce nella workstation. Questa struttura complessa e ben organizzata consente di gestire in modo efficace e preciso le interazioni degli utenti con gli oggetti e gli elementi di gioco all'interno della mappa del tutorial. L'ulteriore strutturazione dello script 'WorkStation' include una variabile di tipo Transform denominata 'workStationTopPoint', che specifica il punto più alto della stazione di lavoro. Questo punto è essenziale per determinare la posizione corretta in cui posizionare i tubi durante le interazioni di gioco. Il metodo 'Start', che viene invocato una sola volta all'istanziamento del GameObject associato a questo script, è responsabile di verificare la presenza di un componente 'Light'. Se presente, il riferimento a questo componente viene memorizzato nella variabile 'light'.

Inoltre, lo script contiene un metodo virtuale chiamato 'Interact', che accetta un argomento di tipo 'Player'. Questo metodo virtuale consente alle sottoclassi di sovrascriverlo e implementare logiche di interazione specifiche per i diversi tipi di giocatori. Oltre a questi metodi, sono presenti ulteriori funzioni che gestiscono vari aspetti del gioco, in particolare riguardanti la gestione degli oggetti 'Pipe' all'interno del gioco.

Lo script associato alla scatola accanto allo scaffale ha lo scopo di gestire il flusso dei tubi verso le diverse stazioni di lavoro nel gioco. Contiene campi per memorizzare un oggetto, un materiale e un riferimento al Renderer. Durante l'inizio del gioco, lo script verifica se sono presenti le condizioni necessarie per inizializzare questi campi e, se le condizioni sono soddisfatte, disabilita l'illuminazione. Durante il gioco, lo script controlla le condizioni per aggiornare le impostazioni della stazione di lavoro, come la presenza di un oggetto 'Pipe'. Quando un giocatore interagisce con la stazione di lavoro, ha la possibilità di collegare o scollegare gli oggetti 'Pipe', influenzando l'illuminazione della lampada situata al di sopra della stazione di lavoro.

Questo script è utilizzato nella stazione di lavoro che appare nella seconda parte del tutorial, dove agli utenti viene chiesto di unire due tubi. Nella terza parte del tutorial, vengono introdotti altri tre elementi che caratterizzano le stazioni di gioco: una stazione per posizionare e rendere disponibili i tubi per altre stazioni, una stazione per gettare via i tubi e una terza stazione per consegnare i tubi. Questi elementi aggiungono ulteriori livelli di complessità e sfida al tutorial, permettendo agli utenti di familiarizzare con le diverse meccaniche e interazioni presenti nel gioco.

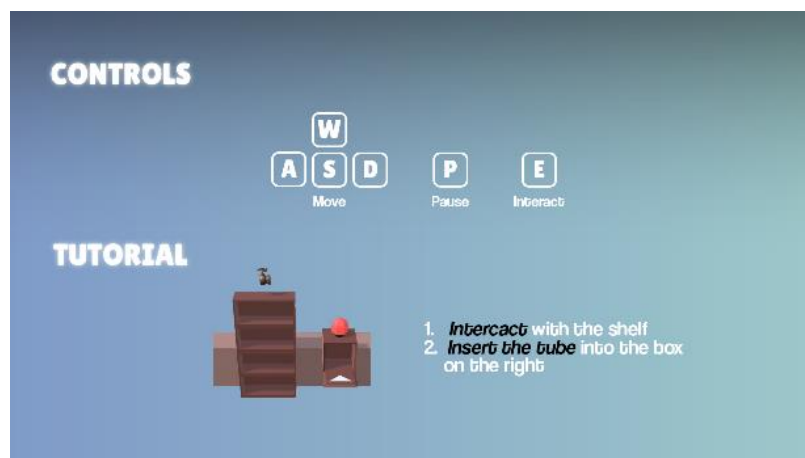


Figura 17: Tutorial - Primo step

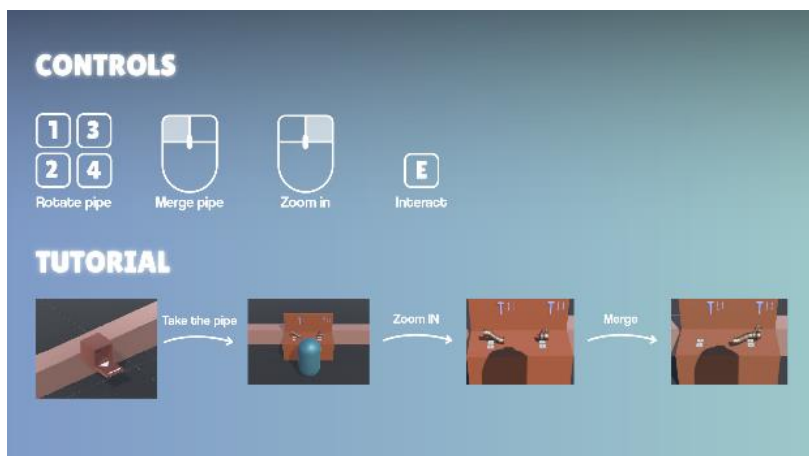


Figura 18: Tutorial - Secondo step

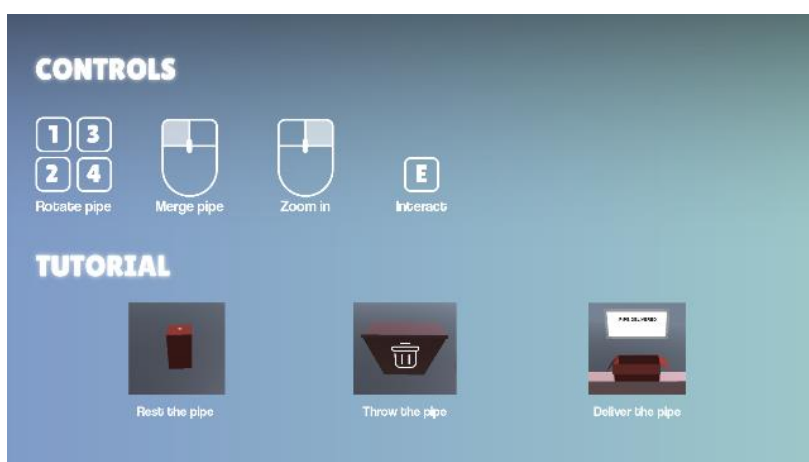


Figura 19: Tutorial - Terzo step

5.5. Primo livello

Una volta completato il tutorial, gli utenti hanno la libertà di tornare al menu principale e iniziare il primo livello effettivo del gioco. Nella schermata di configurazione del livello, i giocatori hanno la possibilità di impostare la durata in minuti e il numero di tubi da consegnare. Dopo aver fatto le loro scelte, possono selezionare il pulsante "Ready" per confermare le loro impostazioni e avanzare alla scena del primo livello.

All'avvio del primo livello, i giocatori vengono accolti da una schermata iniziale che fornisce un riepilogo dei controlli di gioco. Accanto ai controlli, compare un messaggio che incoraggia i giocatori a raggiungere la loro postazione di lavoro prima che il timer raggiunga lo zero. Questo aspetto del gioco enfatizza l'importanza della preparazione e della pianificazione strategica per affrontare con successo le sfide che si presenteranno durante il livello.

Una volta che i giocatori si sentono pronti per iniziare, hanno la possibilità di procedere. Tuttavia, è importante notare che verranno trattenuti in una schermata di attesa fino a quando tutti gli altri

giocatori non saranno anche loro pronti a iniziare il livello. Questa caratteristica assicura che tutti i partecipanti siano sulla stessa pagina e pronti a collaborare o competere in modo sincronizzato, creando un'esperienza di gioco coesa e ben coordinata per tutti i giocatori coinvolti.



Figura 20: Schermata di riepilogo dei controlli

Una volta che tutti i giocatori hanno selezionato il pulsante "Ready", viene avviato un conto alla rovescia di dieci secondi prima dell'inizio effettivo del livello. Durante questo breve periodo di tempo, i giocatori devono rapidamente individuare e posizionarsi alla propria postazione di lavoro, facilmente identificabile da una scritta sul pavimento che la contrassegna. Questa fase di preparazione sottolinea l'importanza della tempestività e dell'organizzazione per affrontare con successo le sfide del livello.

Una volta che ogni giocatore ha trovato e raggiunto la propria postazione di lavoro, sono pronti per iniziare il gioco. In alto a destra dello schermo, è visibile un timer dinamico che mostra il tempo rimanente per completare il livello. Subito sotto il timer, vengono visualizzati due numeri chiave: il primo indica la quantità di tubi già consegnati con successo, mentre il secondo rappresenta la quantità totale di tubi che ancora devono essere consegnati. Queste informazioni sono fondamentali per tenere traccia del progresso e per pianificare le mosse successive durante il livello.

Il GameObject centrale che gestisce lo stato complessivo del gioco e coordina gli aspetti multiplayer è il GameManager. Lo script con lo stesso nome agisce come un hub centrale, responsabile della gestione dello stato di gioco, della sincronizzazione degli elementi multiplayer e dell'interazione con i giocatori durante una partita. Il GameManager assicura che tutte le componenti del gioco funzionino in modo armonico e coeso, garantendo un'esperienza di gioco fluida, coinvolgente e ben coordinata per tutti i partecipanti. Lo script del GameManager è progettato per gestire e coordinare vari aspetti dinamici del gioco, e per fare ciò utilizza eventi definiti tramite EventHandler. Questi eventi permettono di notificare ad altri script all'interno del gioco ogni volta che si verifica un cambiamento di stato specifico, garantendo una comunicazione

efficace e una sincronizzazione tra le diverse componenti del gioco. In particolare, i cinque eventi principali definiti nello script del GameManager sono:

- **OnStateChanged:** Questo evento viene attivato ogni volta che lo stato generale del gioco cambia. Ciò può includere transizioni come l'ingresso in modalità pausa, l'attesa dell'inizio della partita, la vittoria o la sconfitta. Gli altri script possono sottoscrivere a questo evento per eseguire determinate azioni o aggiornamenti quando lo stato del gioco cambia.
- **OnLocalPlayerReady:** Questo evento viene attivato quando lo stato del giocatore locale cambia. Ad esempio, può essere utilizzato per notificare agli altri giocatori o agli script correlati che il giocatore locale è ora pronto a partecipare alla partita o che ha completato una determinata azione.
- **OnLocalGamePaused:** Questo evento viene attivato quando il giocatore locale entra nel menu di pausa o riprende il gioco dal menu di pausa. Questo permette di gestire correttamente le interruzioni e le riprese di gioco, garantendo che tutti i giocatori siano sincronizzati sullo stato corrente del gioco.
- **OnMultiplayerGamePaused:** Questo evento viene attivato quando il gioco viene messo in pausa o viene ripreso dal lato del server per tutti i giocatori. Questo è particolarmente utile per mantenere una coerenza di gioco tra tutti i partecipanti, assicurando che tutti vedano e sperimentino le stesse condizioni di gioco.
- **OnBotSpawn:** Questo evento viene attivato quando, dal lato del server, viene spawnato un bot nel gioco. Questo può essere utilizzato per gestire dinamicamente la presenza e il comportamento dei bot nel gioco, garantendo una partita equilibrata e coinvolgente per tutti i giocatori.

In sintesi, l'uso di eventi tramite EventHandler nello script del GameManager contribuisce a una gestione efficace e dinamica degli stati di gioco, migliorando l'esperienza complessiva del giocatore e assicurando una sincronizzazione ottimale tra le diverse componenti del gioco multiplayer. L'implementazione di un enum per gestire le diverse fasi di gioco fornisce una struttura chiara e organizzata per definire e monitorare lo stato corrente del gioco. Questo enum, chiamato "State", include una serie di stati chiave che rappresentano le diverse fasi del gioco, come l'attesa dell'inizio, il conteggio alla partenza, lo svolgimento del gioco, la conclusione e la visualizzazione della classifica finale.

La variabile di rete di tipo "State" consente di mantenere una sincronizzazione tra il server e i client riguardo allo stato attuale del gioco. Le transizioni tra gli stati avvengono in risposta alle azioni dei giocatori e al progresso del timer, garantendo una progressione fluida e coesa del gioco.

Dal lato del client, lo script `GameInput` si occupa di gestire gli input dei giocatori, permettendo loro di interagire con il gioco in modo intuitivo. Questo script gestisce anche lo stato di preparazione del giocatore, invocando metodi lato server per azioni come confermare la propria prontezza o mettere in pausa il gioco.

Dal lato server, vengono gestite le dinamiche multiplayer del gioco. Le variabili di rete vengono utilizzate per sincronizzare lo stato di gioco e lo stato di preparazione tra il giocatore locale e gli altri client. Per gestire l'interazione dei giocatori e aggiornare lo stato del gioco sul server, notificando tutti i client di eventuali cambiamenti, vengono utilizzati metodi di tipo `"ServerRpc"`. Questi metodi consentono al server di comunicare direttamente con i client e di garantire che tutti i giocatori ricevano le informazioni aggiornate in tempo reale. Inoltre, il server si occupa di gestire il timer del countdown prima dell'inizio del gioco e tiene traccia del tempo trascorso durante la partita. Vengono forniti anche metodi per verificare se la partita è terminata e se tutti i tubi sono stati consegnati. Questo permette di concludere il livello anticipatamente se necessario, assicurando una conclusione adeguata e una transizione fluida alle fasi successive del gioco.

Sostanzialmente l'architettura implementata offre una solida base per un'esperienza di gioco multiplayer ben strutturata, coesa e sincronizzata, garantendo un'interazione fluida tra i giocatori e un progresso armonioso attraverso le diverse fasi del gioco.

All'interno del `GameObject GameManager` è incluso lo script `DeliveryManager`, che agisce come hub centrale per la gestione del processo di consegna, il monitoraggio del numero di consegne e la notifica ad altre parti del gioco riguardo alle consegne. Inoltre, si occupa della visualizzazione dell'avanzamento delle consegne e della segnalazione della fine del livello dal lato del server. Nello script, è definito un evento statico che segnala agli altri quando viene consegnato un tubo nella workstation 8. Sul lato server, viene mantenuto un conteggio del numero totale di tubi consegnati, e questi valori vengono confrontati con quelli da consegnare impostati dal project manager all'inizio della partita per verificare il completamento del livello. Se tutti i tubi sono stati consegnati, viene segnalata la fine del livello.

All'interno dello script chiamato `ProductManagement`, vengono effettuate le operazioni necessarie per consentire al product manager di analizzare i dati e le prestazioni ottenute durante la partita. Questo script scrive i dati su un file di testo e in contemporanea li legge e li analizza per contare gli errori per ciascuna stazione di lavoro, generando un report CSV che riassume i conteggi degli errori. L'utilizzo di un file CSV può rendere il report più intuitivo e veloce da leggere rispetto al file di testo.

L'ultimo script presente all'interno del `GameManager` è il `BotManager`. Questo script si occupa della gestione dello spawn dei bot sul lato server, collaborando con lo script `GameMultiplayer` per la gestione dei dettagli della creazione dei bot. In quest'ultimo infatti è presente una funzione chiamata `SpawnBot` che si occupa di creare e posizionare i bot nel gioco, ed è attiva solo sul server. In particolare, esegue la chiamata al metodo `SetNumberOfBot` dell'oggetto

CharacterSelectUI al fine di determinare quanti bot devono essere creati, in base alle azioni effettuate nell'interfaccia di selezione dei personaggi. Successivamente, assegna il risultato di questa operazione ad una variabile che salva gli indici dei bot e verifica se ciascuna flag bot è attiva nell'oggetto CharacterSelectUI. Se lo è, esegue le seguenti azioni: recupera il prefab del bot corrispondente dall'indice dell'array listaBot, crea un'istanza del prefab del bot utilizzando Instantiate, chiama il metodo Spawn sul componente NetworkObject del bot appena creato per registrarlo sul network.

Questi script lavorano in sinergia per garantire un'esperienza di gioco fluida, dinamica e ben strutturata, gestendo in modo efficace le diverse componenti e dinamiche del gioco, dallo spawn dei bot alla gestione delle consegne, dall'analisi delle prestazioni alla generazione di report dettagliati. L'architettura modulare e organizzata contribuisce a una migliore manutenibilità, scalabilità e estensibilità del gioco, facilitando l'implementazione di nuove funzionalità e miglioramenti futuri.

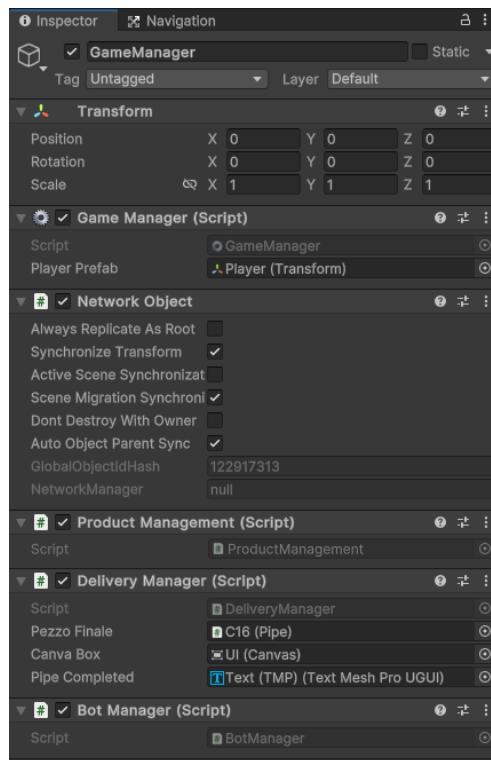


Figura 21: GameManager nell'Inspector di Unity

Lo script BotController regola il comportamento dei bot all'interno dell'ambiente di gioco, gestendo i loro movimenti e le azioni specifiche in base alle condizioni di gioco e alle interazioni con le workstation assegnate.

Il comportamento di ciascun bot è determinato dal livello in cui si trova e dalle interazioni con le workstation disponibili. Se il bot si trova al livello 1, inizia raccogliendo un tubo dalla prima workstation, poi un altro dalla seconda. Successivamente, consegna questi tubi a una terza

stazione di lavoro. Una volta completato questo ciclo, il bot ripete la sequenza in modo continuativo. Altrimenti, se il bot non è al livello 1, raccoglie un tubo da un supermarket e lo consegna a una work station.

Il comportamento del bot è influenzato da diversi fattori, tra cui:

- La presenza di un tubo in possesso del bot.
- La necessità di visitare workstation specifiche, determinate dal livello del bot.
- La disponibilità di tubi da raccogliere nelle workstation.

Per quanto riguarda l'implementazione, il codice del bot utilizza un componente NavMeshAgent per il movimento all'interno dell'ambiente di gioco. Inoltre, per supportare il sistema di navigazione, il pavimento di gioco è dotato di un componente di tipo NavMeshSurface.

6.5.1. Player

Il GameObject di tipo Player rappresenta l'entità giocatore all'interno del gioco e svolge un ruolo cruciale nella gestione delle interazioni e della sincronizzazione tra client nel contesto multiplayer. All'interno di questo GameObject, il componente Client Network Transform assume un ruolo fondamentale, essendo responsabile della sincronizzazione della posizione, rotazione e scala del GameObject tra i vari client connessi alla partita.

Tuttavia, lo script Player è il fulcro delle funzionalità legate al giocatore. Esso si occupa di:

- Gestire l'inizializzazione corretta del giocatore su server e client attraverso il metodo OnNetworkSpawm, posizionando il giocatore in base ai punti di spawn definiti da GameMultiplayer.
- Attivare l'evento OnAnyPlayerSpawned, permettendo agli altri giocatori di reagire alla comparsa del nuovo giocatore nella partita.
- Gestire le disconnessioni sul server e interagire con le varie work station durante la partita.
- Aggiornare la rappresentazione visiva del giocatore in base al colore selezionato nella fase di configurazione precedente al livello.
- Gestire i dati del giocatore in base alle informazioni ottenute dallo script GameMultiplayer.

Dal punto di vista del controllo del movimento e dell'input, lo script Player utilizza un CharacterController per gestire il movimento all'interno del mondo di gioco. Riceve input tramite il sistema PlayerInputActions e gestisce la rotazione della telecamera in risposta all'input del mouse. Inoltre, può alternare tra una modalità di movimento libero e una modalità che visualizza

il cursore nella scena per facilitare l'interazione con gli oggetti, come il merge dei tubi. Lo script tiene anche traccia della direzione del movimento del giocatore per le interazioni.

Per quanto riguarda le interazioni, lo script Player utilizza raycast per individuare le workstation e i tubi di fronte al giocatore, permettendo di rilevare e interagire con gli oggetti nel mondo di gioco. Inoltre, tiene traccia dello stato del tubo posseduto dal giocatore, fornendo metodi per manipolarlo quando necessario durante la partita.

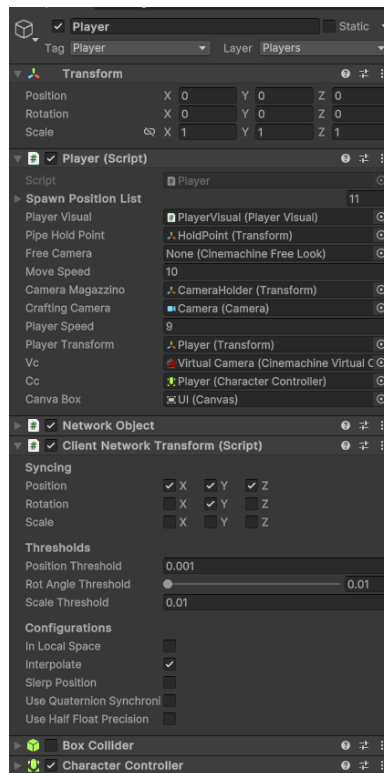


Figura 22: Player nell'Inspector

6.5.2. Shader Graph e ParticleSystem

Per ottimizzare l'esperienza utente all'interno del gioco e guidare efficacemente il magazziniere nella consegna dei tubi, sono stati utilizzati degli shader personalizzati utilizzando le funzionalità di Moving Visual di Unity. Questi shader non solo simulano il movimento dei tubi all'interno dell'ambiente di gioco, ma creano anche un effetto di dissolvenza che fa gradualmente scomparire un oggetto, fornendo indicazioni visive chiare e intuitive.

Gli shader integrano diversi nodi chiave per gestire sia il movimento che l'aspetto visivo delle indicazioni. Tra i nodi utilizzati troviamo il Vertex Position e UV Coordinates per la gestione delle coordinate e della posizione, il Time per monitorare il tempo trascorso dall'inizio del gioco, e Lerp per combinare valori in base a un fattore di interpolazione controllato dal tempo. Questo approccio dinamico non solo crea un effetto visivo in cui le frecce sembrano muoversi, ma permette anche

di gestire l'effetto di dissolvenza attraverso nodi come Multiply e Fragment Color, che moltiplicano il valore di dissolvenza con il colore della texture e determinano il colore finale del frammento.

Ulteriori nodi, come Dissolve Amount, Dissolve Curve e Dissolve Mask, sono stati integrati per controllare la velocità, la curva e l'area di dissolvenza dell'oggetto, permettendo di creare effetti visivi complessi e dettagliati. Questo approccio combinato offre un'esperienza visiva coinvolgente e facilita la comprensione delle direzioni da parte del magazziniere e delle stazioni di lavoro, migliorando complessivamente l'esperienza di gioco.

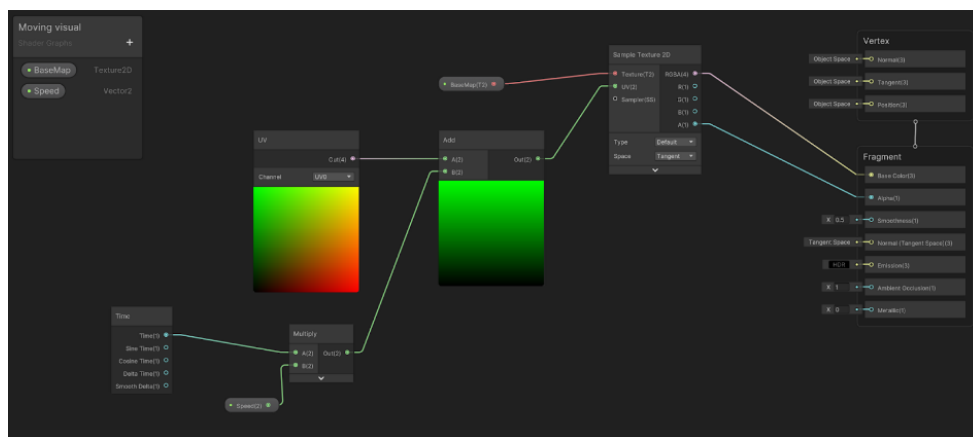


Figura 23: Shader graph per le indicazioni di movimento

Nelle Middle Work Station della scena di gioco è presente un altro effetto visivo. Queste stazioni, poste a metà percorso, permettono ai giocatori di posare i tubi. Grazie all'uso dei particle system di Unity, è stato creato un effetto visivo ondulato. Questo effetto ha lo scopo di invitare i giocatori a posare i tubi in queste stazioni, rendendo l'interazione con esse meno monotona e più intuitiva. Il sistema di particelle è progettato per emettere particelle con una forma circolare, generando 100 particelle al secondo per una durata complessiva di 2 secondi. Le particelle iniziano il loro percorso con una velocità di 1 unità al secondo e una direzione e rotazione iniziale casuale. La loro vita utile è stabilita a due secondi con una curva di vita lineare.

Dal punto di vista dimensionale, le particelle partono con una dimensione iniziale di 1 unità, che si riduce gradualmente fino a raggiungere 0 unità nel corso del loro ciclo di vita. In parallelo, il loro colore inizia come bianco e si trasforma in trasparente grazie a una specifica curva di animazione nel tempo.

In termini di forza e fisica, le particelle iniziano il loro movimento senza alcuna forza iniziale e senza forza radiale. L'attenuazione è stabilita a 1 unità. Riguardo alla simulazione, le particelle operano in uno spazio di simulazione locale senza l'influenza di gravità o velocità del vento.

Dal punto di vista dell'interazione e dell'analisi, il sistema di particelle non emette eventi specifici né è soggetto a collisioni. In sintesi, il sistema emette particelle circolari con una velocità iniziale

variabile, la cui dimensione e colore si modificano nel tempo, creando un effetto visivo dinamico e fluido.

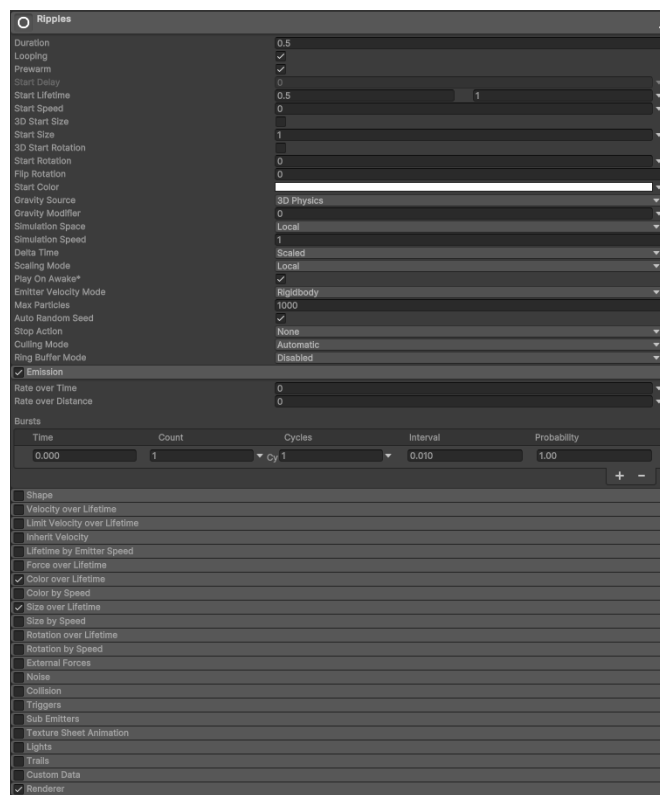


Figura 24: Particle system

6.5.3. Audio

All'interno della scena è presente un GameObject chiamato SoundManager. Questo contiene uno script che si occupa della gestione degli effetti sonori all'interno del serious game. La sua funzione è quindi quella di riprodurre effetti sonori in base agli eventi, modificando il volume dell'audio in base a come l'utente lo imposta nel menù delle impostazioni. All'interno dello script viene memorizzato un riferimento all'oggetto AudioManager, utilizzato per controllare il volume principale. Il volume iniziale viene caricato dalle preferenze del giocatore utilizzando una chiave denominata "SoundEffectsVolume". Per gli effetti sonori, si fa riferimento a uno scriptable object chiamato AudioClipRefsSO, che contiene riferimenti ai vari effetti sonori utilizzati nel gioco. Vengono definiti diversi metodi per riprodurre gli effetti sonori, tra cui PlaySound per riprodurre un effetto sonoro casuale da un array di clip in una posizione specifica con un volume specificato, o per riprodurre un effetto sonoro specifico in una posizione specifica con un volume specifico, PlayCountdownSound per riprodurre l'effetto sonoro del conto alla rovescia, e PlayWarningSound per riprodurre l'effetto sonoro di avviso in una posizione specifica. Per quanto riguarda gli

ascoltatori di eventi, l'oggetto si registra per ascoltare gli eventi di altri script all'avvio. Nei metodi di ascolto eventi, vengono recuperati gli effetti sonori appropriati in base all'evento e riprodotti nella posizione pertinente. Infine, per il controllo del volume, viene definito un metodo `GetVolume` che recupera il volume corrente degli effetti sonori. Questo metodo chiama un altro script, chiamato `VolumeControl` e responsabile della regolazione del volume, per ottenere il volume attuale. Il volume recuperato viene memorizzato in una variabile e viene definito un metodo `ChangeVolume`. I suoni utilizzati all'interno del gioco sono stati presi da una piattaforma online collaborativa chiamata `Freesound`¹⁷.

5.6. Secondo livello

Il passaggio al secondo livello del gioco introduce nuove dinamiche e strumenti che arricchiscono l'esperienza di gioco e la complessità delle sfide da affrontare. Di seguito ci sono le novità introdotte rispetto al livello 1:

- **Supermarket:** Una delle principali innovazioni del secondo livello è l'introduzione del supermarket. Questo nuovo elemento consente al magazziniere di gestire e consegnare più tubi contemporaneamente, migliorando l'efficienza delle operazioni. Ogni stazione di lavoro dispone di uno scaffale contenente una box simile alla precedente, ma con una luce rossa sostituita da un numero. Questo numero indica la quantità di tubi disponibili nel supermarket per quella specifica stazione, permettendo al personale di monitorare costantemente lo stato delle scorte e intervenire prontamente in caso di necessità.
- **Nuova Tipologia di Box nelle Work Station:** Le nuove work station presentano una box con un `Canvas` sul lato frontale che mostra il numero di tubi contenuti al suo interno. Questa nuova tipologia di box è gestita da uno script dedicato che controlla l'interazione con la box, monitorando la quantità di tubi disponibili e coordinando le dinamiche di interazione con i giocatori.
- **Sincronizzazione tra Script delle Work Station e SuperMarket:** Lo script associato alle box nelle work station è progettato per sincronizzarsi con lo script `SuperMarket` del magazziniere. Questa sincronizzazione garantisce che le informazioni sull'inventario dei tubi siano sempre aggiornate e consistenti tra tutte le stazioni di lavoro. In questo modo, i giocatori possono interagire con la box nella loro stazione, visualizzare il numero di tubi disponibili e agire di conseguenza, sapendo che le informazioni sono sempre aggiornate e condivise con il sistema centrale di gestione dell'inventario.

¹⁷ *Freesound*, <https://freesound.org/>, ultimo accesso (20/02/2024)

In sintesi, il secondo livello del gioco introduce nuove sfide e opportunità, migliorando l'efficienza delle operazioni attraverso l'introduzione del supermarket e garantendo una gestione dell'inventario più accurata e dinamica grazie alla nuova tipologia di box e alla sincronizzazione tra gli script delle work station e il SuperMarket del magazziniere. Queste novità arricchiscono l'esperienza di gioco, richiedendo ai giocatori di adattare e ottimizzare le proprie strategie per affrontare con successo le sfide del secondo livello.



Figura 25: SuperMarket per le work station

L'introduzione delle nuove box nelle workstation introduce un elemento dinamico e visivamente coinvolgente nel gioco. Queste box, similmente a quanto avviene per il trash bin nella stazione otto, sono animate per riflettere le interazioni dei giocatori, grazie a un lavoro di animazione realizzato con Blender. La gestione dell'animazione avviene attraverso l'utilizzo dell'Animator all'interno dello script del gioco, che modifica una variabile booleana in risposta alle azioni dei giocatori, attivando o disattivando l'animazione di conseguenza.

Per assicurare che l'esperienza di gioco sia omogenea e sincronizzata tra tutti i partecipanti in una sessione multiplayer, si fa ricorso a un NetworkAnimator. Questo strumento è essenziale per garantire che le animazioni vengano visualizzate correttamente e in modo coordinato su tutti i client, mantenendo così l'integrità dell'esperienza di gioco condivisa.

In questo modo, l'animazione delle box non solo migliora l'aspetto visivo del gioco ma aggiunge anche un livello di interazione e di reattività all'ambiente di gioco, rendendo le azioni dei giocatori più tangibili e gratificanti. La sincronizzazione attraverso il NetworkAnimator, inoltre, assicura che tutti i giocatori possano vedere gli stessi eventi nello stesso momento, fondamentale in un contesto multiplayer per mantenere l'equità e l'immersività del gioco.

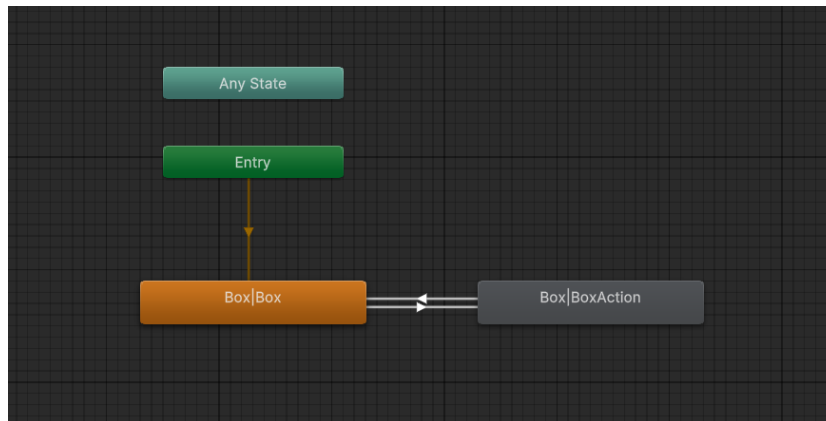


Figura 26: Animator per l'apertura della box

Il cestino della spazzatura nella stazione otto segue un modello di sviluppo simile a quello delle box di interazione precedentemente descritte. Utilizza lo stesso Animator della box per garantire una coerenza visiva e di esperienza utente all'interno del gioco. Il suo script è focalizzato principalmente sull'animazione di apertura e chiusura del cestino e sulla notifica di tubi distrutti. Questa notifica è essenziale per informare tutti i client presenti nel gioco che un tubo è stato eliminato.

Un altro elemento importante, non precedentemente menzionato nel contesto del primo livello, è la box relativa alla consegna del tubo finale. Questa box è dotata di uno script specifico che gestisce l'interazione e la sincronizzazione della consegna del tubo finale tra tutti i client. Quando un giocatore completa con successo la consegna del tubo finale, lo script associato alla box attiva una notifica per informare tutti i partecipanti che il tubo finale è stato consegnato con successo. Per migliorare l'esperienza del giocatore e garantire un feedback visivo immediato, al momento della consegna del tubo finale viene istanziato un canvas. Questo canvas è controllato da uno script che assicura che il testo di conferma sia sempre orientato correttamente in base alla posizione e alla rotazione della telecamera del giocatore. Questa funzionalità garantisce che il messaggio di conferma della consegna sia sempre visibile e leggibile, indipendentemente dalla posizione o dalla direzione in cui il giocatore si trova.

Infine, anche se nel paragrafo precedente non abbiamo approfondito l'argomento, è fondamentale sottolineare che la funzionalità di unione dei tubi rimane costante e uniforme sia nel primo che nel secondo livello del gioco. Nel prossimo paragrafo, esamineremo più dettagliatamente il processo di merge dei tubi, esplorando le caratteristiche uniche che questi componenti presentano nel contesto del gioco.

6.6.1. Merge

All'interno delle impostazioni e nei menu sia principale che di pausa, è stata introdotta una guida specifica suddivisa per stazioni. Questa guida è progettata per fornire un supporto prezioso agli utenti, specialmente a coloro che si avvicinano per la prima volta al serious game. L'obiettivo principale di questa funzionalità è quello di guidare gli utenti attraverso il complesso processo di merge dei tubi, offrendo istruzioni dettagliate e chiare per ogni stazione di lavoro presente nel gioco. Questa guida interattiva permette agli utenti di comprendere meglio le dinamiche e le meccaniche del gioco, assicurando che possano eseguire il merge dei tubi in modo corretto e efficiente. Grazie a questa funzionalità, gli utenti possono acquisire familiarità con le diverse stazioni di lavoro, imparando le procedure corrette e acquisendo le competenze necessarie per progredire con successo nei livelli del gioco. L'inclusione di una guida dettagliata e suddivisa per stazioni rappresenta un importante strumento didattico all'interno del serious game. Essa contribuisce a ridurre la curva di apprendimento, facilitando l'ingresso nel gioco anche a utenti meno esperti, e garantendo un'esperienza di gioco più fluida e gratificante per tutti gli utenti.

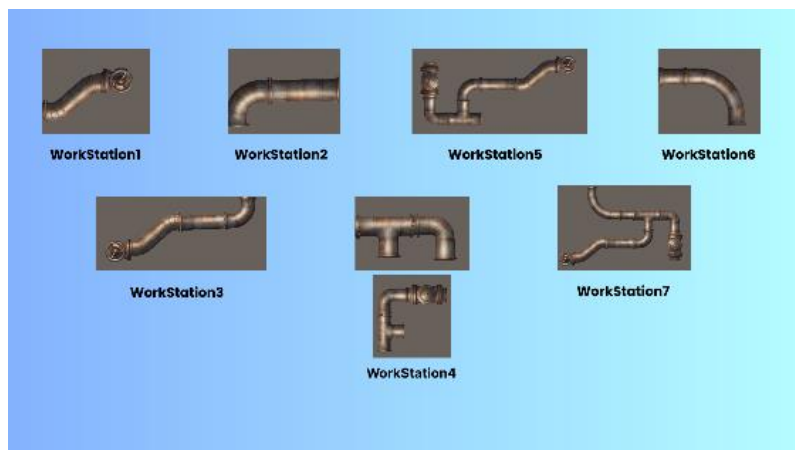


Figura 27: Guida al merge dei tubi

La classe 'ScriptableObject' viene utilizzata per definire e gestire le informazioni dei tubi nel gioco. Questa classe memorizza dati essenziali relativi ai tubi, inclusi il prefabbricato del tubo stesso, il suo nome e uno stato di errore che indica la correttezza dei merge precedenti effettuati per ottenere quel particolare tubo. Per ogni tubo, vengono memorizzati i seguenti attributi:

- Prefab del tubo: Questo attributo rappresenta il prefab del tubo, ovvero il modello tridimensionale che viene istanziato nel gioco. In sintesi, contiene le informazioni visive del tubo.

- Nome del tubo: Questo attributo rappresenta il nome del tubo. È utile per identificare il tubo all'interno del gioco e nei menu di gestione.
- Stato di errore: Questo attributo booleano indica lo stato di errore del tubo. Viene utilizzato per determinare se ci sono errori relativi al merge dei tubi precedenti che hanno portato alla creazione di questo tubo specifico.

Nel gioco, la manipolazione e l'unione dei tubi rappresentano un elemento fondamentale per il gameplay. La dinamica all'interno del gioco funziona nei seguenti modi:

- Rotazione dei tubi: Durante la partita, i giocatori hanno la possibilità di ruotare i tubi di 90° lungo gli assi X e Y prima di procedere con l'unione dei tubi. Questa funzionalità non è gestita direttamente dai tubi stessi ma è controllata tramite uno script associato a un GameObject di tipo empty, che è collegato alle crafting table. Questo approccio permette una maggiore flessibilità e controllo sulla rotazione dei tubi e consente di integrare ulteriori meccanismi di gioco.
- Unione dei tubi: Per effettuare l'unione dei tubi, è presente un GameObject specifico nella scena con uno script denominato "Draggable". Questo script consente di trascinare gli oggetti tubo attraverso interazioni mouse. Quando un giocatore fa clic su un tubo, lo script identifica il tubo attraverso un raggio lanciato dalla posizione della telecamera e gestisce il trascinamento del tubo. Durante questo processo, il corpo rigido del tubo diventa non cinematico e viene avviata una coroutine per gestire il trascinamento del tubo.
- Prefab dei tubi: I prefab dei tubi includono diverse funzionalità e componenti per gestire vari aspetti dei tubi nel gioco. Ad esempio, il componente FollowTransform consente di impostare la posizione del tubo in base al giocatore che lo possiede. Lo script Pipe gestisce le proprietà del tubo e offre metodi per restituire l'oggetto tubo, distruggere o spawnare dinamicamente il tubo. In generale, questo script si occupa di gestire l'intero ciclo di vita dei tubi all'interno del livello.
- Elemento di casualità: Dopo l'unione dei due tubi, la rotazione del GameObject tubo viene variata casualmente, aggiungendo un elemento di casualità e complessità al gameplay. Questo contribuisce a rendere l'esperienza di gioco più dinamica e stimolante per gli utenti, poiché ogni combinazione di tubi può produrre risultati diversi.

Questo sistema offre ai giocatori una sfida stimolante e una varietà di opzioni e strategie per completare con successo i livelli del gioco.

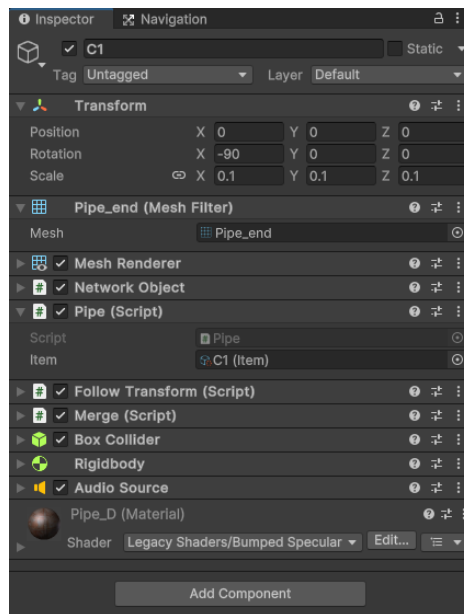


Figura 28: Pipe nell'Inspector

5.7. Fine livello

La gestione della conclusione di una partita è fondamentale per garantire una buona esperienza di gioco e fornire feedback chiari ai giocatori sul loro rendimento. I diversi scenari di conclusione della partita nel gioco vengono gestiti nel seguente modo:

- Nel caso di disconnessione dell'host, che assume il ruolo di product manager, la partita viene immediatamente interrotta. Questa situazione viene comunicata agli altri giocatori attraverso un messaggio o una notifica, informandoli dell'interruzione della partita. A questo punto, i giocatori hanno l'opzione di tornare al menu principale del gioco per avviare una nuova partita o scegliere altre opzioni disponibili.
- Se i giocatori non riescono a consegnare la quantità di tubi richiesta entro il tempo limite stabilito per la partita, la partita termina automaticamente. Una schermata di sconfitta viene visualizzata sullo schermo, che mostra il numero di stelle ottenute dai giocatori in base alla loro performance durante la partita. Questo feedback aiuta i giocatori a comprendere le aree in cui possono migliorare e fornisce un incentivo per riprovare e completare con successo i livelli successivi.
- Nel caso in cui i giocatori riescono a consegnare tutti i tubi richiesti entro il tempo limite, la partita si conclude con una vittoria. Una schermata di successo viene visualizzata, comunicando ai giocatori il loro successo e congratulandosi per il completamento del livello. Inoltre, i giocatori vengono premiati con il massimo numero di stelle possibile, che riconoscono la loro abilità e efficienza nel gestire le sfide del gioco.

Questi tre scenari delineano le varie possibilità di conclusione di una partita nel gioco, garantendo che i giocatori siano informati sui risultati e sulle eventuali conseguenze delle loro azioni durante il gameplay. Il sistema di assegnazione delle stelline viene gestito tramite uno script denominato `LeaderboardManager`, il quale è assegnato a un empty `GameObject` all'interno della scena del gioco. Questo script ha principalmente il compito di animare le stelle in base al punteggio ottenuto dai giocatori. Questo viene gestito attraverso i metodi come `ShowStars`, `ShowStarsRoutine`, `EnlargeAndShrinkStar` e `ChangeStarScale`, che permettono di far apparire e animare un numero specificato di stelle in modo fluido e accattivante. In sintesi, lo script `LeaderboardManager` si occupa esclusivamente dell'animazione delle stelle per i punteggi dei giocatori, rendendo visivamente piacevole e dinamica l'esperienza di gioco senza coinvolgere la gestione della classifica dei giocatori. Il Canva relativo alla schermata di vittoria è gestito dallo script `WinUI`, che si occupa dell'interfaccia utente complessiva. Questo script rende disponibile il bottone che consente ai giocatori di tornare al menu principale e imposta il numero massimo di stelle da mostrare sulla base delle prestazioni dei giocatori. In modo simile, lo script `GameOverUI` gestisce l'interfaccia utente relativa alla schermata di game over. Tuttavia, per l'assegnazione delle stelle, viene utilizzato un approccio basato su un rapporto tra i tubi consegnati e la quantità totale di tubi da consegnare, come impostato dal product manager. Nel dettaglio, il rapporto tra i tubi consegnati e la quantità totale di tubi da consegnare viene calcolato e valutato per determinare il numero di stelle da assegnare ai giocatori. Ad esempio:

- Se la proporzione è inferiore o uguale a 0.2, viene mostrata una stella.
- Se è inferiore o uguale a 0.4, ne vengono mostrate due.
- Se è inferiore o uguale a 0.6, ne vengono mostrate tre.
- Se è inferiore o uguale a 0.9, ne vengono mostrate quattro.

Questo sistema di assegnazione delle stelle tiene conto delle prestazioni dei giocatori rispetto agli obiettivi stabiliti dal product manager, offrendo una valutazione visiva delle loro prestazioni alla fine del gioco.

CAPITOLO 6

6. RISULTATI

In questo capitolo, verranno esaminati i dati ottenuti dai test effettuati sugli utenti del serious game. Questa analisi ci consente di valutare l'efficacia del gioco e di comprendere come sia stato percepito e accolto dagli utenti.

Un aspetto importante considerato nell'analisi è l'interazione degli utenti con il gioco. Questo include come gli utenti hanno navigato nell'interfaccia, come hanno interagito con gli elementi di gioco e come hanno risposto alle dinamiche proposte.

Oltre all'interazione, abbiamo esaminato le reazioni emotive degli utenti durante il gioco. Questo ci aiuta a comprendere se il gioco ha suscitato emozioni positive, negative o neutre e a identificare eventuali momenti di frustrazione o di soddisfazione.

Le prestazioni degli utenti sono state anche un fattore chiave nell'analisi. Abbiamo valutato la loro abilità nel completare le sfide proposte, la velocità con cui hanno raggiunto gli obiettivi e la precisione nel compiere determinate azioni.

Infine, sono stati raccolti le opinioni generali degli utenti sull'esperienza di gioco. Queste informazioni forniscono un feedback diretto sulle percezioni degli utenti riguardo al gioco, evidenziando i punti di forza e le possibili aree di miglioramento.

Tutti questi dati e informazioni saranno fondamentali per identificare i punti di forza del gioco e le aree che necessitano di ulteriori miglioramenti. Questi risultati saranno dettagliatamente discussi nel capitolo successivo.

6.1. Questionari utilizzati

Durante l'esperimento, sono stati utilizzati due questionari per valutare l'esperienza degli utenti: il System Usability Scale (SUS) e il Motivation and Engagement Evaluation Guide for Application (MEEGA+)¹⁸. Tuttavia, nel caso del MEEGA+, è importante sottolineare che non sono state utilizzate tutte le domande presenti nel questionario, ma solo una selezione mirata. Questo approccio è stato adottato per concentrare l'analisi sui fattori di motivazione e engagement ritenuti più significativi per l'obiettivo specifico del nostro studio. La scelta delle domande da includere è stata basata su una valutazione preliminare delle caratteristiche e degli obiettivi del serious game, al fine di ottenere informazioni pertinenti e utili per l'analisi dei risultati.

In sintesi, il questionario si suddivide in sei sezioni. Nella prima sezione, si chiedono agli utenti dati personali quali età e ruolo durante il gameplay. La seconda sezione analizza l'usabilità del

¹⁸ Petri, Giani & Gresse von Wangenheim, Christiane & Borgatto, Adriano, *MEEGA+: A Method for the Evaluation of Educational Games for Computing Education*, (2018).

gioco, valutando la facilità con cui gli utenti possono interagire con l'interfaccia e navigare nel sistema. Nella terza sezione, si esamina il coinvolgimento generale, valutando l'emozionalità e l'interesse suscitati durante l'esperienza di gioco. La quarta sezione si concentra sull'apprendimento, esaminando in che misura il gioco favorisce l'acquisizione di nuove conoscenze e competenze. La quinta sezione valuta l'impatto sociale del gioco, considerando gli elementi di cooperazione, competizione o interazione sociale presenti nell'esperienza di gioco. Infine, nella sesta sezione vengono raccolte informazioni sulle conoscenze degli utenti riguardanti i serious game e i videogiochi in generale, al fine di comprendere il loro livello di familiarità con il medium videoludico.

6.2. Test

I test sono stati condotti durante la settimana compresa tra l'11 e il 17 marzo, coinvolgendo complessivamente venti partecipanti divisi in tre gruppi distinti. Di conseguenza, sono state organizzate tre sessioni di gioco complete. Ciascuna sessione di gioco ha avuto una durata approssimativa compresa tra i 30 e i 45 minuti, a cui si aggiungevano ulteriori 5 minuti dedicati alla compilazione del questionario post-gioco. Complessivamente il questionario è composto da un totale di trentotto domande. La suddivisione dei test ha consentito di ottenere una copertura significativa dei dati e di garantire un'ampia varietà di risposte ed esperienze da parte dei partecipanti.

Il range d'età dei partecipanti al test è abbastanza vario, con una forte concentrazione sull'età 24 anni. La diversificata fascia d'età dei partecipanti nello studio relativo al serious game multiplayer rappresenta un aspetto cruciale nell'analisi delle dinamiche interattive e dell'esperienza ludica. Con un campione di 20 partecipanti che spaziano dall'età giovanile fino alla maturità, si ha la possibilità di esplorare in profondità come diverse coorti rispondano e si impegnino nel contesto del gioco. Questa varietà di età offre un'ampia prospettiva sull'efficacia del gioco nel coinvolgere e intrattenere un pubblico diversificato, consentendo così di formulare raccomandazioni più mirate per il miglioramento del design e dell'esperienza utente.

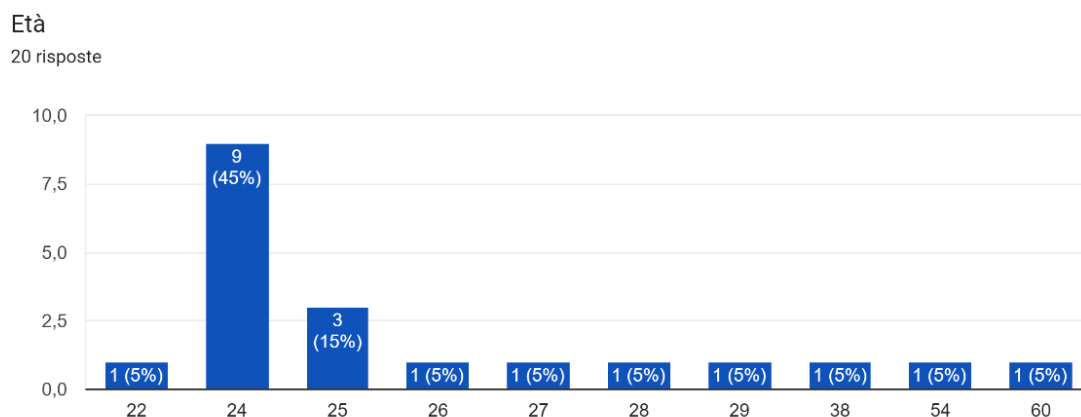


Figura 29: Distribuzione anagrafica

Le informazioni aggiuntive sui partecipanti e il loro rapporto con i videogiochi, in particolare con i serious game, sono state ottenute attraverso tre domande specifiche. Queste domande miravano a raccogliere dati sulle abitudini di gioco dei partecipanti, sulla loro familiarità con i serious game e sulla loro esperienza diretta con tali giochi. È interessante notare che la metà dei partecipanti sapeva cosa fosse un serious game e ci aveva già giocato, nonostante il 35% della popolazione presa in considerazione abbia dichiarato di non giocare mai ai videogame. Questo suggerisce una certa consapevolezza e coinvolgimento con il genere dei serious game anche da parte di coloro che non sono giocatori abituali di videogiochi.

Quante volte giochi ai videogame?

20 risposte

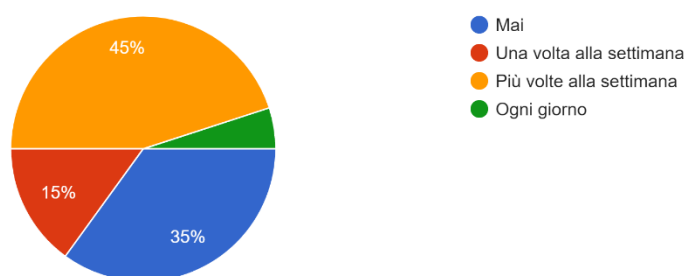


Figura 30: Frequenza di utilizzo dei videogiochi

6.3. Usabilità del sistema

Come detto precedentemente per valutare l'usabilità del sistema è stato utilizzato il SUS¹⁹. Il System Usability Scale (SUS) rappresenta uno strumento affidabile e immediato per valutare l'usabilità di prodotti e servizi. È un questionario composto da 10 domande ed offre cinque opzioni di risposta, da "Fortemente d'accordo" a "Fortemente in disaccordo". I vantaggi nell'utilizzo del SUS sono molteplici. Prima di tutto, è estremamente facile da somministrare ai partecipanti, richiedendo poco tempo per essere completato. Inoltre, è in grado di fornire risultati affidabili anche con campioni di dimensioni ridotte, rendendolo particolarmente adatto per ricerche con limitate risorse di partecipazione. Infine, il SUS è scientificamente valido, consentendo di distinguere efficacemente tra sistemi che sono considerati utilizzabili e quelli che non lo sono. Grazie alla sua diffusa adozione e alla sua comprovata validità, il SUS è diventato uno standard nel settore. Per calcolare i punteggi del questionario si suddividono le domande in pari e dispari. Per le domande dispari si sottrae uno da ciascun punteggio dato, mentre per le domande pari si sottrae il punteggio dato da cinque. Successivamente si sommano i valori ottenuti e si moltiplica il totale per 2.5, convertendo così il punteggio in una scala compresa tra zero e cento. Un punteggio SUS inferiore a 68 indica un livello di usabilità molto basso, sottolineando la necessità di interventi significativi per migliorare l'esperienza utente. Gli utenti potrebbero incontrare numerosi problemi e difficoltà nell'utilizzo del prodotto o servizio. Se il punteggio si situa tra 68 e 78, l'usabilità è considerata sufficiente ma con ampi margini di miglioramento. Nel caso di un punteggio compreso tra 78 e 88, l'usabilità viene valutata come buona. Infine, un punteggio superiore a 88 indica un livello di usabilità eccellente, ciò vuol dire che gli utenti percepiscono il prodotto o servizio come estremamente intuitivo, efficace e piacevole da utilizzare, con pochi o nessun problema evidente.

Nel caso specifico è stato ottenuto un punteggio complessivo di 75.4, evidenziando un livello di usabilità del sistema positivo, superando ampiamente il punteggio di riferimento di 68, indicando che l'usabilità del serious game è buona.

Dall'analisi delle risposte al questionario, emergono i seguenti punti di forza e di debolezza del serious game:

Punti di forza

- La maggior parte degli utenti ha rilevato una facilità d'uso e d'apprendimento del sistema.
- Il serious game è stato valutato come utile e informativo.

¹⁹ Usability.gov, System Usability Scale (SUS), <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>, (ultimo accesso 18/03/24)

- Gli utenti hanno espresso un apprezzamento generale per il contenuto del serious game.

Punti di debolezza

- Alcune persone hanno riscontrato una certa complessità nel sistema.
- È stato evidenziato un bisogno di miglioramento in termini di fluidità e semplicità d'uso del sistema.

	Non d'accordo	Poco d'accordo	Abbastanza d'accordo	D'accordo	Molto d'accordo
1. Penso che mi piacerebbe utilizzare questo sistema frequentemente	0%	5%	25%	45%	25%
2. Ho trovato il sistema complesso senza che ce ne fosse bisogno	45%	35%	5%	15%	0%
3. Ho trovato il sistema molto semplice da usare	0%	0%	30%	60%	10%
4. Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sistema	20%	30%	30%	5%	15%
5. Ho trovato le varie funzionalità del sistema bene integrate	0%	0%	20%	45%	35%
6. Ho trovato incoerenze tra le varie funzionalità del sistema	50%	35%	5%	0%	0%
7. Penso che la maggior parte delle persone potrebbero imparare ad utilizzare il sistema facilmente	0%	0%	25%	40%	35%
8. Ho trovato il sistema molto macchinoso da utilizzare	45%	30%	10%	15%	0%
9. Ho avuto molta confidenza con il sistema durante l'uso	0%	10%	20%	50%	20%
10. Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio il sistema	50%	30%	15%	5%	0%

Tabella 2: Risposte al questionario SUS

6.4. Coinvolgimento generale

Grazie al questionario MEEGA+ è stato possibile fare un calcolo del coinvolgimento generale dell'utente all'interno del serious game. In particolare, sono state fornite delle domande che misurano aspetti specifici dell'esperienza di gioco, ovvero:

- Immersione: “Quanto ti sei sentito immerso nel gioco?”
- Flow: “Quanto ti sei sentito coinvolto e concentrato nel gioco?”
- Sfida: “Quanto hai trovato il gioco impegnativo?”
- Soddisfazione: “Quanto sei soddisfatto del gioco?”

Per ognuna di queste domande sono state fornite delle risposte su una scala da 1 a 5, dove 1 indica la risposta “per niente” e 5 “molto”.

Domanda	Media	Deviazione standard
Quanto ti sei sentito immerso nel gioco?	4	0,8
Quanto ti sei sentito coinvolto e concentrato nel gioco?	3,8	0,9
Quanto hai trovato il gioco impegnativo?	3,5	0,7
Quanto sei soddisfatto del gioco?	4	0,7

Tabella 3: Media e deviazione standard per il questionario sul coinvolgimento generale

Nell'analisi dei dati del questionario MEEGA+, è emerso che la media per l'immersione è di 4,0 con una deviazione standard di 0,8, indicando che la maggioranza degli utenti si è sentita immersa nel gioco. Tuttavia, vi è una certa variabilità nelle risposte, suggerendo che l'esperienza di immersione può variare tra gli individui. Per quanto riguarda il flow, la media è di 3,8 con una deviazione standard di 0,9, evidenziando che il gioco è stato percepito come coinvolgente e capace di favorire una sensazione di concentrazione. Anche in questo caso, si nota una certa variabilità nelle risposte degli utenti. Riguardo alla sfida, la media è di 3,5 con una deviazione standard di 0,7. Questo indica che la maggior parte degli utenti ha ritenuto il gioco abbastanza impegnativo, ma non eccessivamente frustrante, riflettendo un equilibrio tra la sfida offerta dal

gioco e il livello di frustrazione percepito. Infine, per la soddisfazione, la media è di 4,0 con una deviazione standard di 0,7, suggerendo che la maggior parte degli utenti è soddisfatta dell'esperienza di gioco. Complessivamente, i risultati del questionario mostrano un apprezzamento del serious game, con immersione e soddisfazione come gli aspetti meglio valutati e la sfida come l'aspetto leggermente meno apprezzato, ma comunque al di sopra della sufficienza.

6.5. Apprendimento

Per quanto riguarda la sezione inerente all'apprendimento del questionario MEEGA+, sono state fornite agli utenti quattro domande:

- Quanto hai imparato di nuovo giocando a questo gioco?
- Quanto ritieni che le informazioni apprese nel gioco siano utili nella vita reale?
- Quanto hai migliorato le tue abilità apprese giocando a questo gioco?
- Quanto ritieni che le abilità apprese nel gioco siano utili nella vita reale?

Domanda	Media	Deviazione standard
Quanto hai imparato di nuovo giocando a questo gioco?	4,2	0,7
Quanto ritieni che le informazioni apprese nel gioco siano utili nella vita reale?	3,8	0,9
Quanto hai migliorato le tue abilità apprese giocando a questo gioco?	3,7	0,8
Quanto ritieni che le abilità apprese nel gioco siano utili nella vita reale?	3,9	0,8

Tabella 4: Media e deviazione standard per il questionario sull'apprendimento

Gli utenti hanno riportato di aver imparato nuove cose con una media di 4,2 e una deviazione standard di 0,7, suggerendo che il gioco è stato efficace nel fornire nuove conoscenze. Inoltre,

con una media di 3,8 e una deviazione standard di 0,9, la maggior parte degli utenti ha considerato le informazioni apprese nel gioco utili per la vita reale. Riguardo all'abilità acquisita, la media è stata di 3,7 con una deviazione standard di 0,8, indicando che gli utenti ritengono di aver effettivamente migliorato le proprie capacità. Infine, con una media di 3,9 e una deviazione standard di 0,8, gli utenti hanno anche valutato positivamente l'utilità delle abilità apprese nel gioco per la vita reale. I dati del questionario evidenziano che il gioco è stato generalmente considerato un'esperienza di apprendimento benefica. Gli utenti hanno non solo ampliato le loro conoscenze e affinato le loro abilità ma hanno anche percepito un valore pratico nelle informazioni e nelle competenze acquisite, considerandole utili nel contesto della vita reale.

6.6. Impatto sociale

L'ultima sezione relativa al MEEGA+ comprende l'analisi dell'impatto sociale percepito dagli utenti durante l'utilizzo del serious game. Anche in questo caso sono state selezionate quattro domande:

- Quanto ti sei sentito immerso nel gioco?
- Quanto ti sei sentito coinvolto e concentrato nel gioco?
- Quanto hai trovato il gioco impegnativo?
- Quanto sei soddisfatto del gioco?

Domanda	Media	Deviazione standard
Quanto ti sei sentito immerso nel gioco?	4	0,8
Quanto ti sei sentito coinvolto e concentrato nel gioco?	3,8	0,9
Quanto hai trovato il gioco impegnativo?	3,5	0,7
Quanto sei soddisfatto del gioco?	4	0,7

Tabella 5: Media e deviazione standard per il questionario sull'impatto sociale

Gli utenti hanno riportato un elevato grado di immersione con una media di 4,0 e una deviazione standard di 0,8, suggerendo che la maggior parte si è sentita completamente coinvolta nel gioco. Per quanto riguarda il flusso di gioco, la media è stata di 3,8 con una deviazione standard di 0,9, indicando che il gioco è stato percepito come coinvolgente e ha mantenuto l'attenzione degli utenti, anche se con una certa variabilità nelle risposte. Per quanto riguarda la sfida, la media è stata di 3,5 con una deviazione standard di 0,7, suggerendo che il gioco ha offerto un equilibrio tra sfida e divertimento senza risultare troppo frustrante. Infine, la soddisfazione generale è stata elevata con una media di 4,0 e una deviazione standard di 0,7. I risultati del questionario MEEGA+ evidenziano un apprezzamento generale del gioco da parte degli utenti. L'immersione e la soddisfazione sono stati gli aspetti valutati più positivamente, mentre la sfida ha ricevuto un punteggio leggermente inferiore, sebbene rimanga al di sopra della sufficienza. In sintesi, il gioco è stato ben accolto, con gli utenti che hanno trovato un buon equilibrio tra coinvolgimento, sfida e soddisfazione.

In generale i test sugli utenti hanno prodotto risultati positivi, indicando un apprezzamento complessivo del progetto. Tuttavia, l'analisi dettagliata dei dati ha rivelato alcune aree in cui potrebbero essere apportate miglioramenti. Da un punto di vista dell'usabilità, ci sono aspetti che

potrebbero essere ottimizzati per garantire un'esperienza utente ancora più fluida e intuitiva. Inoltre, i risultati del questionario MEEGA+ hanno evidenziato alcuni punti in cui il progetto potrebbe essere arricchito o affinato per meglio soddisfare le esigenze e le aspettative degli utenti. Al termine del questionario, è stata inclusa una domanda aperta, non obbligatoria, per consentire agli utenti di condividere liberamente i loro pensieri e feedback personali riguardo all'esperienza di gioco. Questo tipo di domanda permette agli utenti di esprimere in modo spontaneo le loro impressioni, fornendo spesso informazioni preziose che potrebbero non emergere da domande più strutturate.

Tra i commenti ricevuti, sono stati evidenziati diversi aspetti del serious game, tra cui alcune opinioni riguardanti il tutorial. È emerso che alcuni utenti avrebbero preferito un tutorial più dettagliato e strutturato. In particolare, alcuni suggeriscono l'introduzione di una sezione video iniziale, in cui vengono illustrate le dinamiche di gioco e mostrate le azioni da compiere durante la partita. Questa modalità di apprendimento visivo potrebbe aiutare gli utenti a comprendere più facilmente le meccaniche di gioco e a sentirsi più preparati prima di immergersi nella parte pratica del gioco. Integrare un tutorial video potrebbe quindi rappresentare un valore aggiunto, rendendo l'esperienza di apprendimento più intuitiva e gratificante per gli utenti, specialmente per coloro che sono alle prime armi con il serious game.

CAPITOLO 7

7. CONCLUSIONI

L'uso dei serious game sta guadagnando sempre più terreno sia nel contesto accademico che in quello aziendale, emergendo come una soluzione efficace per facilitare il processo di apprendimento senza esaurire risorse. L'accessibilità delle applicazioni sviluppate per desktop o browser consente a un vasto pubblico di usufruire di queste risorse educative in modo semplice e intuitivo. I risultati dei test condotti hanno ulteriormente rafforzato l'efficacia di tali strumenti, evidenziando che l'introduzione del multiplayer ha arricchito l'esperienza di gioco, conferendo maggiore dinamicità e divertimento, senza compromettere l'obiettivo educativo sottostante. Questo dimostra che è possibile coniugare intrattenimento e apprendimento in modo equilibrato, offrendo agli utenti un'esperienza coinvolgente e formativa al tempo stesso. In un mondo sempre più orientato alla digitalizzazione e all'apprendimento online, i serious game si presentano quindi come una soluzione promettente per l'educazione del futuro.

Per una valutazione ancora più completa e dettagliata del progetto, sarebbe auspicabile coinvolgere un numero maggiore di tester, specialmente individui con una formazione o esperienza nell'ambito della lean manufacturing. Questo permetterebbe di ottenere feedback da parte di persone che già possiedono una conoscenza dei concetti base della produzione, consentendo così di valutare più accuratamente l'efficacia del serious game nel contesto specifico. Inoltre, l'interazione tra tester che già si conoscono potrebbe favorire una comunicazione più fluida ed efficace durante il gioco, contribuendo a creare un ambiente collaborativo e stimolante.

7.1. Sviluppi futuri

Per arricchire l'esperienza educativa offerta dal serious game e renderla più inclusiva per tutti gli utenti, sarebbe opportuno ampliare la gamma di indicatori chiave di performance (KPI) monitorati dal product manager. Attualmente, l'attenzione è principalmente focalizzata sulla correttezza del lavoro svolto nelle singole stazioni di lavoro. Tuttavia, per una valutazione più completa e dettagliata, potrebbe essere utile introdurre ulteriori controlli. Ad esempio, si potrebbero implementare KPI specifici per monitorare i pezzi gestiti dal magazziniere e il tempo impiegato da ciascun utente per completare le attività nelle diverse stazioni. Questi aggiornamenti consentirebbero di fornire agli utenti una visione più completa del processo produttivo, offrendo al contempo un feedback più dettagliato e informativo sulle prestazioni individuali e collettive nel contesto del serious game. Per migliorare l'efficacia dell'analisi condotta dal product manager, sarebbe consigliabile sostituire i tradizionali file di testo con grafici interattivi. Questa soluzione visuale permetterebbe una rappresentazione più immediata e comprensibile dei dati,

semplificando la valutazione delle prestazioni degli utenti nel contesto del serious game. L'utilizzo di grafici renderebbe il processo di analisi più intuitivo e accessibile, in questo modo, il product manager potrebbe prendere decisioni informate in modo più efficiente, ottimizzando l'esperienza complessiva del gioco e contribuendo a una valutazione più accurata e dettagliata delle performance degli utenti.

Sarebbe inoltre vantaggioso ampliare la varietà dei pezzi forniti, concentrandosi in particolare sui tubi. Questa diversificazione non solo renderebbe le attività del magazziniere più variegata e stimolante, riducendo la monotonia associata a compiti ripetitivi, ma aggiungerebbe anche un elemento di sfida e coinvolgimento per gli utenti responsabili dell'assemblaggio dei tubi nelle diverse workstation. Introducendo una maggiore varietà di componenti, si potrebbe infatti incrementare il livello di interesse e divertimento del gioco, rendendo l'esperienza complessiva più avvincente e coinvolgente per tutti i giocatori.

Come sviluppo futuro, sarebbe opportuno considerare l'implementazione di una funzione di chat vocale o testuale direttamente all'interno del serious game. Questa caratteristica potrebbe facilitare e migliorare la comunicazione tra i giocatori durante la partita, senza la necessità di utilizzare strumenti esterni aggiuntivi. L'integrazione di un sistema di chat interno, infatti, permetterebbe una interazione più fluida e immediata tra gli utenti, favorendo la collaborazione e il coordinamento delle attività all'interno del gioco. Questo aggiornamento potrebbe contribuire a rendere l'esperienza di gioco più coesa e interattiva, potenziando il coinvolgimento e la partecipazione degli utenti e arricchendo ulteriormente l'aspetto sociale e collaborativo del serious game. Questo è possibile implementarlo direttamente da Unity grazie ai servizi offerti per il multiplayer, in particolare con Vivox²⁰. Utilizzando queste risorse integrate, si potrebbe facilitare l'implementazione di un sistema di comunicazione interno al serious game senza la necessità di ricorrere a soluzioni esterne o a strumenti aggiuntivi.

Infine, come emerso dai feedback dei tester, vi è un consenso unanime riguardo alla necessità di implementare un tutorial più approfondito in future versioni del serious game. Questa raccomandazione rispecchia la volontà degli utenti di avere a disposizione risorse didattiche più dettagliate e strutturate, che possano guidarli in modo esaustivo attraverso le dinamiche di gioco e le meccaniche di base. Integrare un tutorial più dettagliato potrebbe non solo migliorare l'usabilità del gioco, ma anche arricchire l'esperienza di apprendimento degli utenti, permettendo loro di acquisire competenze in modo più efficace e gratificante. Pertanto, l'implementazione di un tutorial più esaustivo rappresenta una delle principali aree di miglioramento su cui concentrarsi per ottimizzare ulteriormente il serious game e soddisfare le esigenze di una gamma più ampia di giocatori.

²⁰ Docs.unity.com, Unity Documentation, <https://docs.unity.com/vivox/Article.html>, (ultimo accesso 16/03/24)

BIBLIOGRAFIA

1. Assetstore.unity.com, Low Poly Cars, <https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-cars-101798>, (ultimo accesso 25/01/24)
2. Assetstore.unity.com , *Pipe-constructor*, <https://assetstore.unity.com/packages/3d/environments/industrial/pipe-constructor-186974>, (ultimo accesso 25/01/24)
3. Docs-multiplayer.unity3d.com, About Netcode for GameObjects, <https://docs-multiplayer.unity3d.com/netcode/current/about/> , (ultimo accesso 03/01/24)
4. Doc.photonengine.com, Fusion 2 Introduction, <https://doc.photonengine.com/fusion/current/fusion-intro>, (ultimo accesso 27/12/23)
5. Doc.photoengine.com, Quantum 2 Intro, <https://doc.photonengine.com/quantum/current/quantum-intro> (ultimo accesso 28/12/23)
6. Docs.unity.com, Unity Documentation, <https://docs.unity.com/vivox/Article.html>, (ultimo accesso 16/03/24)
7. Docs.unity.com, Unity Relay, <https://docs.unity.com/ugs/manual/relay/manual/introduction> (ultimo accesso 20/12/2023)
8. Duarte F. Gomes, M. Pereira Lopes, C. Vaz de Carvalho, *Serious Games for Lean Manufacturing: The 5S Game*, IEEE Revista Iberoamericana de Tecnologias del Aprendizaje 8(4):191-196 DOI: 10.1109/RITA.2013.2284955 (2013)
9. Emeraude-escape, *Carrefour – Gamified Digital Retail Training*, <https://emeraude-escape.com/en/case/carrefour-gamified-digital-training/> (ultimo accesso 4/01/2024)
10. Fish-networking.gitbook.io, Fish-Net Introduction, <https://fish-networking.gitbook.io/docs> (ultimo accesso 27/12/23)
11. *Freesound*, <https://freesound.org/> , ultimo accesso (20/02/2024)
12. leansimulations.org, *LEAN BICYCLE FACTORY GAME*, <https://www.leansimulations.org/2010/07/lean-bicycle-factory-game.html> (ultimo accesso 05/01/2024)
13. Leo J. de Vin, Lasse Jacobsson, JanErik Odhe, *Game-based Lean Production training of university students and industrial employees*, Karlstad University, 651 88 Karlstad, Sweden (2018)
14. Messaadia, M., Bufardi, A., Le Duigou, J., Szigeti, H., Eynard, B., Kiritsis, D. (2013). *Applying Serious Games in Lean Manufacturing Training*. In: Emmanouilidis, C.,

- Taisch, M., Kiritsis, D. (eds) *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services. APMS 2012. IFIP Advances in Information and Communication Technology*, vol 397. Springer, Berlin, Heidelberg.
15. Mirror-networking.gitbook.io, Mirror Networking, <https://mirror-networking.gitbook.io/docs> (ultimo accesso 27/12/23)
 16. Outlife.in, *Online Game Based Experiential Learning*, <https://www.outlife.in/game-based-experiential-learning.html> (ultimo accesso 27/12/2023)
 17. Petri, Giani & Gresse von Wangenheim, Christiane & Borgatto, Adriano, *MEEGA+: A Method for the Evaluation of Educational Games for Computing Education*, (2018).
 18. Usability.gov, System Usability Scale (SUS), <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>, (ultimo accesso 18/03/24)
 19. Wikipedia, *Serious game*, https://it.wikipedia.org/wiki/Serious_game (ultimo accesso 28/12/2023)