

POLITECNICO DI TORINO

Ingegneria Informatica

Tesi di Laurea Magistrale

**Videogiochi *single-switch* educativi e
inclusivi**



Relatori:

Fulvio Corno

Sebastián Aced López

Luigi De Russis

Candidato:

Nicolò Mandrile

Marzo 2016

Ringraziamenti

Desidero ringraziare il Professore Fulvio Corno, Sebastián Aced López e Luigi De Russis, relatori di questa tesi, per avermi dato l'opportunità di lavorare a questo progetto e per il prezioso aiuto che mi hanno fornito per la sua realizzazione.

Un ringraziamento va anche alla Professoressa Paola Molina e Daniela Bulgarelli del dipartimento di psicologia dell'Università degli studi di Torino, alle logopediste, alle fisioterapiste e alle psicologhe dell'Azienda Sanitaria Locale To1 di Torino per aver fornito utili indicazioni riguardanti alcune caratteristiche e componenti educative dei videogiochi sviluppati. Voglio ringraziarle per la loro partecipazione a questo progetto e per il meraviglioso lavoro che fanno per i bambini disabili.

Ringrazio tutti i miei amici e i miei compagni di corso per aver condiviso con me questi anni di gioie e sacrifici.

Il ringraziamento più importante va alla mia grande famiglia e in particolare ai miei genitori che mi sono sempre stati accanto anche nei momenti di difficoltà, per tutto quello che mi hanno insegnato e perché senza di loro tutto questo non sarebbe stato possibile.

Grazie.

Indice

Ringraziamenti	1
Elenco delle figure	1
1 Introduzione	3
1.1 Panoramica	3
1.2 Organizzazione	4
2 Il gioco nell'infanzia	6
2.1 Cos'è il gioco	6
2.2 L'importanza del gioco nell'infanzia	7
2.2.1 La privazione del gioco	8
2.3 I videogiochi	9
2.3.1 Benefici cognitivi	9
2.3.2 Benefici motivazionali	10
2.3.3 Benefici emozionali	11
2.3.4 Benefici sociali	12
2.4 Il gioco per bambini con disabilità motorie	12
3 Confronto tra i principali meccanismi di accessibilità per persone con disabilità motorie	14
3.1 Tecnologie assistive	14
3.1.1 Defnizioni	14
3.2 Meccanismi a scansione	17
3.2.1 Descrizione	17
3.2.2 Implementazione	19
3.2.3 Limiti	20
3.3 Nomon	21
3.3.1 Visualizzazione	21
3.4 GNomon	23

3.4.1	Differenze con Nomon	24
4	Obiettivi e proposte	27
4.1	Obiettivi	27
4.1.1	La socializzazione	27
4.2	Vincoli	28
4.3	Le proposte	29
4.3.1	<i>Il labirinto dei mostri</i>	29
4.3.2	<i>Unisci i puntini</i>	30
4.3.3	<i>Palloncini</i>	30
4.3.4	La scelta	30
5	<i>Il labirinto dei mostri</i>	32
5.1	Il gioco	32
5.2	Descrizione	32
5.3	Organizzazione	34
5.3.1	Regole del gioco	36
5.4	Movimento	36
5.5	Punteggio	36
5.6	Livelli	39
5.7	Elementi di gioco	40
5.7.1	Stage 1 - Banana	40
5.7.2	Stage 2 - Tapis Roulant	41
5.7.3	Stage 3 - Muro Rosso	41
5.7.4	Stage 4 - Spirale	41
5.7.5	Stage 5 - Muro Verde	42
5.7.6	Stage 6 - Percorso Trappola	42
5.8	Componenti educative	43
5.8.1	Giochi collaborativi	44
6	<i>Palloncini</i>	45
6.1	Il gioco	45
6.2	Descrizione	45
6.3	Organizzazione	47
6.3.1	Modalità di gioco	47
6.3.2	Punteggio	51
6.4	Componenti educative	52

7	Implementazione	54
7.1	Strumenti di sviluppo	54
7.2	Organizzazione	57
7.2.1	Assets	57
7.3	GNomon	61
7.3.1	Modifiche a GNomon	62
7.3.2	Interazione con GNomon	63
7.4	<i>Il labirinto dei mostri</i>	66
7.4.1	Scene	66
7.4.2	Script	68
7.4.3	Personaggi	70
7.5	<i>Palloncini</i>	74
7.5.1	Scene	74
7.5.2	Script	75
7.5.3	Palloncini	76
7.5.4	Modalità di gioco	80
8	Valutazione	82
8.1	Il pulsante utilizzato	82
8.2	Test	83
8.2.1	Partecipanti	83
8.2.2	Metodi e procedure	84
8.3	Risultati	85
8.4	Conclusioni	86
9	Conclusioni	88
9.1	Futuri sviluppi	89
	Bibliografia	90

Elenco delle figure

3.1	Questa immagine mostra il divario che la disabilità crea tra le capacità di una persona e le componenti socio economiche che si cerca di ridurre o eliminare grazie alle tecnologie assistive [24].	16
3.2	Dispositivi <i>single-switch</i>	18
3.3	Esempio di immissione testo con meccanismi a scansione.	18
3.4	Questa immagine raffigura il classico orologio di Nomon, con la lancetta rossa fissa sul mezzogiorno e la lancetta nera che ruota a partire da uno sfasamento determinato da Nomon.	21
3.5	Questa immagine raffigura un orologio diventato giallo dopo un click, indicando che è uno di quelli con la probabilità posteriore maggiore. L'operazione di colorare di giallo gli orologi più probabili permette di visualizzare una progressione verso la selezione finale.	23
3.6	Gioco " <i>One Switch Invaders</i> ".	24
3.7	Questa immagine mostra, a sinistra, l'orologio originale di Nomon e, a destra, quello utilizzato da GNomon con le importanti modifiche consigliate da esperti: linee più spesse, dimensione maggiore, indicatori ai quarti e lancetta nera appuntita.	25
3.8	Meccanismi di selezione alternativi di GNomon.	26
5.1	Schermata iniziale.	33
5.2	Mostri.	33
5.3	Schermata impostazioni.	34
5.4	Scena di selezione stage.	35
5.5	Scena di selezione livelli.	35
5.6	Risultato livello.	37
5.7	Stage completato.	38
5.8	Gioco completato.	38
5.9	Livello 17.	39
5.10	Schermata di pausa.	40
5.11	Banana.	40

5.12	Tapis roulant.	41
5.13	Muro rosso.	41
5.14	Spirale.	42
5.15	Muro verde e pulsante per distruggerlo.	42
5.16	Percorso utilizzabile solamente una volta.	42
6.1	Schermata iniziale del gioco <i>Palloncini</i>	46
6.2	Pagina delle impostazioni del gioco <i>Palloncini</i>	46
6.3	Scena in cui si può scegliere la modalità di gioco.	47
6.4	Modalità “ <i>Colori</i> ”.	48
6.5	Modalità “ <i>Numeri</i> ”.	49
6.6	Modalità “ <i>Sequenza colori</i> ”.	49
6.7	Modalità “ <i>Dimensioni</i> ”.	50
6.8	Modalità “ <i>Parole</i> ”.	51
6.9	Fine livello.	52
7.1	Organizzazione interna di una scena in Unity.	56
7.2	Assets.	57
7.3	Hierarchy.	60
7.4	Camera con due canvas.	61
7.5	Elementi di GNomon.	62
7.6	Container_Clock.	62
7.7	Interazione con GNomon.	64
7.8	Assets.	66
7.9	Organizzazione stage e livelli.	66
7.10	Diagramma delle scene del gioco <i>Il labirinto dei mostri</i>	67
7.11	Script del gioco <i>Il labirinto dei mostri</i>	68
7.12	Componenti dei personaggi.	71
7.13	Scene del gioco <i>Palloncini</i>	74
7.14	Diagramma delle scene del gioco <i>Il labirinto dei mostri</i>	75
7.15	Script del gioco <i>Palloncini</i>	76
7.16	Script del gioco <i>Il labirinto dei mostri</i>	77
8.1	BigRed Switch.	82
8.2	Helpibox 16.	83

Capitolo 1

Introduzione

1.1 Panoramica

Il gioco, nella vita di un bambino, è un momento molto importante per la sua crescita dal punto di vista educativo, cognitivo e relazionale. Esso è una parte fondamentale per lo sviluppo della persona, tramite la quale si sperimenta, si conosce e si prende consapevolezza del mondo circostante e degli altri. Senza il gioco, l'apprendimento, la creatività e le normali funzioni sociali potrebbero non svilupparsi adeguatamente avendo quindi ripercussioni sul futuro del bambino.

Una parte di questa esperienza ludica, al giorno d'oggi, viene anche svolta dai videogiochi. In Italia, infatti 29,3 milioni di persone ne fanno utilizzo [3]. Secondo una ricerca [23] effettuata nei Paesi più sviluppati, tra cui l'Italia, è emerso che il 58% dei bambini di età compresa tra i 2 e i 5 anni sa usare i videogiochi. Negli Stati Uniti d'America, il 91% dei bambini con età compresa tra i 2 e i 17 anni, ne fa utilizzo [16].

La larghissima diffusione dei videogiochi è dovuta alla loro diversità, dinamicità ed alla facilità con cui si possono reperire su vari dispositivi (PC, console, smartphone, tablet, ...).

Nonostante la loro grande importanza nella vita dei bambini, però, non sempre questi aspetti della vita sono accessibili e alla portata di tutti allo stesso modo: le difficoltà che un bambino disabile si trova ad affrontare sono molte e vanno dall'impossibilità di poter comunicare facilmente, di poter giocare ed interagire in modo naturale con gli altri bambini e di acquisire quelle capacità che vengono inconsciamente apprese durante lo svolgimento di un'attività di tipo ludico, come la definizione ed il rispetto delle regole, il rispetto per gli altri, l'autocontrollo, la creatività, la capacità di risolvere problemi e molte altre ancora. Inoltre, al giorno d'oggi, molto spesso anche le tecnologie possono essere un ostacolo per certe persone. Per esempio, alcuni individui con disabilità (in particolare pensiamo a bambini con grave disabilità motoria), non hanno la possibilità di usufruire delle moderne tecnologie a causa di difficoltà incontrate nell'utilizzo dei comuni meccanismi di interazione con dispositivi elettronici (mouse, tastiera, touchscreen, joystick).

Per questo motivo, esistono e sono utilizzate diverse tipologie di tecnologie assistive¹ per consentire a persone con disabilità motorie l'accesso al computer e ad altri dispositivi. Purtroppo, però, per le caratteristiche dinamiche di alcuni videogiochi, molti di questi strumenti, come per esempio i meccanismi a scansione, non sono adeguati per consentire una buona giocabilità causando l'impossibilità di utilizzo da parte di persone disabili.

In questo documento viene descritta l'importanza del gioco e del videogioco durante l'infanzia, con particolare enfasi sugli effetti positivi che comporta, i gravi effetti che avrebbe la sua privazione, le problematiche affrontate dai bambini con disabilità motorie nel giocare normalmente e le possibili soluzioni adottabili. Verranno descritti e discussi i principali meccanismi di accessibilità già esistenti (meccanismi a scansione, Nomon e GNomon) studiandone l'utilizzo in questo particolare tipo di applicazione per questo pubblico di utenti.

Lo scopo di questa tesi è quello di progettare, analizzare e testare dei videogiochi *single-switch* per PC indirizzati a bambini con gravi disabilità motorie. Per *single-switch* si intendono quei giochi accessibili ed interamente giocabili semplicemente facendo pressione su un pulsante in modo da superare l'ostacolo dato dalle disabilità fisiche di questi bambini. Per realizzare questi videogiochi vengono sfruttate le potenzialità del framework GNomon, creato dal gruppo *e-Lite*² del Politecnico di Torino.

1.2 Organizzazione

La tesi è organizzata come segue:

- nel capitolo successivo, il secondo, si riportano alcune ricerche fatte per evidenziare l'importanza del gioco durante l'infanzia per la crescita e l'educazione, il ruolo che hanno o che possono avere i videogiochi nello sviluppo e gli effetti che potrebbe avere la privazione del gioco nei primi anni della vita di un bambino. Inoltre, vengono riportate le difficoltà che possono incontrare nel gioco i bambini con gravi disabilità motorie.
- Nel terzo capitolo vengono descritte alcune tra le principali tecnologie assistive utilizzate per permettere a persone disabili di accedere ed interagire con i computer. Esistono molte tipologie di ausili tecnologici volti a questo scopo ma in questo capitolo sono trattati solamente quelli che richiedono uno sforzo motorio molto contenuto, per andare incontro ad utenti con gravi disabilità motorie. In particolare, vengono descritti i meccanismi a scansione, Nomon e GNomon. Vengono studiati i limiti e la loro applicabilità per la creazione di videogiochi *single-switch*, dinamici e accessibili a bambini con le suddette disabilità.

¹Per tecnologie assistive si intendono tutti gli strumenti e le soluzioni tecniche, hardware e software, che permettono alla persona disabile, superando o riducendo condizioni di svantaggio, di accedere alle informazioni ed ai servizi erogati dai sistemi informatici.

²<http://elite.polito.it>

- Nel quarto capitolo vengono riportati gli obiettivi che hanno guidato la realizzazione dei videogiochi, quali caratteristiche devono avere, quali sono state le principali proposte e quali sono state alcune delle considerazioni effettuate che hanno portato alla scelta finale dei giochi da sviluppare.
- Nel quinto capitolo viene descritto il gioco “*Il labirinto dei mostri*” sviluppato sfruttando i vantaggi della tecnologia GNomon. Vengono descritte le sue componenti educative e sono riportate le sue caratteristiche e potenzialità.
- Nel sesto capitolo viene descritto il gioco “*Palloncini*”. Viene descritta la sua struttura, i suoi aspetti educativi e le abilità dei bambini che si vogliono sviluppare tramite questo videogioco.
- Nel settimo capitolo viene descritta l’implementazione vera e propria dei due giochi descritti nei capitoli precedenti, evidenziando come essi sono strutturati, quali sono le componenti utilizzate, quali scelte sono state fatte e quali librerie sono utilizzate. Vengono inoltre descritti gli strumenti di sviluppo utilizzati.
- Nell’ottavo capitolo vengono riportate le sperimentazioni pratiche del primo gioco, eseguite facendolo provare ad alcuni bambini. Vengono descritte le prove a cui essi hanno partecipato, le tempistiche ed il dispositivo *single-switch* che è stato utilizzato per interagire con il computer. Vengono inoltre riportati e commentati i risultati ottenuti dalle sperimentazioni: gli obiettivi raggiunti, le difficoltà incontrate dai bambini, i punti forti e deboli del gioco, ...
- Nel nono capitolo vengono riportate alcune conclusioni contenenti riflessioni sul lavoro svolto e su possibili sviluppi futuri che si possono realizzare in questo campo.

Capitolo 2

Il gioco nell'infanzia

Questo capitolo affronta una breve riflessione su quanto sia importante l'aspetto ludico ed il divertimento durante l'infanzia, ragionando sulle implicazioni che causerebbe l'assenza del gioco, enfatizzando quali difficoltà può incontrare un bambino con gravi disabilità motorie nel giocare, apprendere, crescere e nel relazionarsi con gli altri e quale ruolo possono avere in questo contesto i videogiochi.

2.1 Cos'è il gioco

Tutti sanno cosa sia il gioco e cosa significhi giocare, ma non è affatto semplice darne una definizione. Esso ha molte caratteristiche e molti aspetti che lo rendono difficile da definire. Per molti, il gioco è il contrario del lavoro, consiste nel dedicarsi ad una attività per puro piacere, per cercare un momento di divertimento e di rilassamento quando non si ha altro da fare. Per gli insegnanti, per esempio, può essere visto come un modo tramite il quale i bambini apprendono. Per altri, invece, il gioco è più importante del lavoro, è quando ci si sente più vivi. Ciò che accomuna tutte queste visioni è che il gioco è un qualcosa che si fa per sé stessi. È la scelta libera di un'attività che si fa per piacere, infatti il gioco può essere piacevole, soddisfacente, divertente, stimolante, . . . Il gioco, più che un comportamento, è un processo flessibile e creativo. Si gioca per esplorare, conoscere, essere stimolati o semplicemente per non annoiarsi [11]. Riassumendo, proviamo a dare una definizione di gioco:

- il gioco è un processo: è il modo in cui si gioca ad essere importante, non a cosa si gioca;
- il gioco è liberamente scelto dal giocatore;
- è personalmente diretto: la modalità di gioco è decisa dal bambino. Per esempio riordinare la camera può trasformarsi in gioco se il bambino lo vede in questo modo;

- è intrinsecamente motivato: l'impulso viene da dentro, il bambino deve trovare l'attività soddisfacente e ricavare piacere da essa.

Il gioco, quindi, non è un comportamento specifico, ma un'attività che si svolge attraverso una vivace attività mentale. Lo psichiatra Stuart Brown scrive che *“giocare è la base dell'arte, dei giochi, dei libri, degli sport, dei film, della moda, del divertimento e quindi, in breve, è la base di quello che viene indicato come civilizzazione.* [15].

2.2 L'importanza del gioco nell'infanzia

Il gioco è essenziale nello sviluppo perché contribuisce al benessere emotivo, fisico, cognitivo e sociale dei bambini [14]. Il teorico del gioco Brian Sutton-Smith sostiene che i bambini, durante il gioco, *“non stanno solamente sviluppando le fondamenta neurologiche che permetteranno loro di risolvere problemi, di comunicare e di creare, ma apprendono mentre giocano. Imparano come relazionarsi con gli altri, come calibrare la loro forza fisica e come pensare in termini astratti. Attraverso il gioco i bambini “imparano ad imparare”. Ciò che si acquisisce attraverso il gioco non sono informazioni specifiche, ma un generale assetto mentale volto alla risoluzione di problemi che includono sia l'astrazione sia la flessibilità, dove i bambini inseriscono pezzi di comportamento per formare nuove soluzioni a problemi che richiedono la ricostruzione di pensieri ed azioni. Un bambino che non viene stimolato o che ha poche opportunità di esplorare i propri dintorni può fallire nel collegare pienamente quelle connessioni nervose necessarie ad un futuro apprendimento.”* [25]

Il gioco permette ai bambini di usare la loro creatività, destrezza e forza cognitiva, fisica ed emotiva. Il gioco è importante per uno sviluppo salutare del cervello. È attraverso il gioco che i bambini iniziano ad interagire con il mondo che li circonda. Esso permette loro di creare ed esplorare un mondo che possono controllare, in cui possono imparare a gestire le proprie paure, sia da soli che in congiunzione con altri bambini o con chi si prende cura di loro [14].

Oltre a questo, molti sostengono che il gioco apporti molti altri aspetti positivi nella vita di un bambino. Essi sviluppano capacità motorie, il ragionamento, il rispetto per le regole e per gli altri. Di seguito sono riportati i principali benefici che il gioco apporta nella vita di un bambino [15]:

- **Benefici emotivi:**

- riduce paura, ansia, stress ed irritabilità;
- crea gioia, intimità, autostima e padronanza in sé stessi;
- aumenta la flessibilità emotiva e la lealtà;
- aumenta la calma, la determinazione, l'adattamento e l'abilità di affrontare la sorpresa ed il cambiamento;
- può curare le sofferenze.

- **Benefici sociali del gioco:**

- aumenta l'empatia, la compassione e la condivisione;
- crea opinioni;
- crea modelli di relazioni basati sull'inclusione invece che sull'esclusione;
- permette di sviluppare competenze non verbali;
- aumenta l'attenzione e lo spirito di gruppo;

- **Benefici fisici:**

- le emozioni positive aumentano l'efficienza del sistema immunitario, endocrino e cardiovascolare;
- diminuisce lo stress, la fatica e la depressione;
- aumenta la gamma dei movimenti, l'agilità, la coordinazione, l'equilibrio e la flessibilità.

2.2.1 La privazione del gioco

“Un bambino che non viene stimolato nel gioco e che ha poche opportunità di esplorare ciò che lo circonda potrebbe non congiungere pienamente le connessioni neuronali necessarie al successivo apprendimento.” [25]

Poiché il gioco promuove la crescita e lo sviluppo cerebrale, *“i bambini che non hanno abbastanza opportunità per giocare, sperimenteranno un dannoso sviluppo e una scarsa flessibilità del cervello. Queste conclusioni si basano su studi eseguiti su animali, sebbene il lavoro con dei bambini abbia dato risultati simili” [11]*

Quindi i bambini che non giocano o non ne hanno possibilità sono a rischio di sviluppi anormali e comportamenti devianti. Senza il gioco, l'autocontrollo non si sviluppa in maniera adeguata. Come la privazione del sonno, anche quella del gioco produce diverse conseguenze. Senza il gioco il livello ottimale di apprendimento, delle normali funzioni sociali, dell'autocontrollo e di altre funzioni cognitive può non essere raggiunto adeguatamente.

Una pubblicazione di Play Wales¹ riguardo agli studi sul gioco conclude: *“Chiaramente il gioco è estremamente importante per il bambino, in particolar modo dagli 0 ai 7 anni. Non c'è dubbio che i bambini privati del gioco subiscano delle conseguenze fisiche e psicologiche devastanti. I bambini si adattano a molti cambiamenti nel gioco, come l'avvento dei computer giocattolo che ha permesso di scoprire nuove abilità. Comunque la privazione del gioco non riguarda il cambiamento, ma l'assenza degli input sensoriali essenziali per l'essere umano. La privazione cronica del gioco può avere l'effetto graduale di disumanizzazione del bambino, con una conseguente perdita dell'abilità di enfatizzare e provare compassione, o condividere la stessa realtà con altri bambini. L'evidenza suggerisce che i bambini privati del gioco diventano adulti disturbati, aggressivi e violenti.” [15].*

¹Ente di beneficenza indipendente, finanziato dal governo gallese, che sostiene il diritto di giocare dei bambini.
<http://www.playwales.org.uk>

2.3 I videogiochi

In Italia, da un rapporto [3] di AESVI (Associazione Editori Sviluppatori Videogiochi Italiani), nel 2014, il mercato dei videogiochi ammonta ad un totale di 893,3 milioni di euro. Nel 2015 il numero dei giocatori è di circa 29 milioni di persone, contro circa 22 milioni nel 2011.

Secondo uno studio effettuato da *NPD Group*² del 2011, negli Stati Uniti d'America, il 91% (circa 64 milioni) di bambini con età compresa tra i due ed i diciassette anni, fa utilizzo di videogiochi. Un altro studio rappresentativo dei teenager americani, riporta che in particolare il 99% dei maschi ed il 94% delle femmine gioca con dispositivi elettronici [16].

Da questi numeri si capisce quanto sia diffusa, oggi, la cultura del videogioco e quante persone ne facciano utilizzo. Dalla metà degli anni '80 ad oggi, i videogiochi sono diventati una parte integrante nella vita dei bambini e degli adulti, grazie alla diffusione di prodotti sempre più complessi ed intriganti.

Questo cambiamento non ha influenzato soltanto il modo in cui i bambini passano il loro tempo, ma ha implicazioni anche sul modo di pensare e di apprendere. Questo apprendimento è soggetto a particolare interesse date le sue caratteristiche ludiche e il suo alto coinvolgimento che si basa sulla forte attrattiva che i videogiochi hanno su chi li utilizza. I videogiochi infatti coinvolgono bambini ed adolescenti in un processo che può portarli a sviluppare molte abilità, capacità, funzioni mentali e atteggiamenti strategici [5]. Ma quali sono in realtà i benefici che questi giochi possono avere nella vita dei bambini?

La maggior parte delle ricerche da parte di psicologi sugli effetti del gioco si focalizzano principalmente sui suoi aspetti negativi, come il potenziale danno relazionale alla violenza, alla dipendenza ed alla depressione. Considerare però i suoi benefici, è anche molto importante, in parte, perché nell'ultimo decennio la natura dei videogiochi è drasticamente cambiata, diventando complessa, diversa, realistica e sociale [16]. In questa sezione si riassumono i benefici che si possono avere sfruttando i videogiochi, focalizzando l'attenzione su quattro principali categorie: cognitivi, motivazionali, emozionali e sociali.

2.3.1 Benefici cognitivi

Al contrario della convinzione convenzionale che giocare ai videogiochi fosse intellettivamente pigro e sedativo si è scoperto che giocare a questi giochi favorisce lo sviluppo di un ampio dominio di abilità cognitive [16]. È difficile classificare i videogiochi a seconda dei benefici che apportano perché, data la loro complessità e la loro diversità, possono avere aspetti simili o completamente diversi che possono andare a sviluppare delle capacità diverse, anche contemporaneamente.

Data la notevole complessità di alcuni videogiochi, la capacità di gestire simultaneamente più informazioni comporta lo sviluppo di abilità spaziali complesse ed in particolare il potenziamento

²“*National Purchase Diary*”: istituto di ricerche di mercato statunitense.

dell'attenzione visiva. Inoltre viene stimolata l'induzione, perché spesso il giocatore deve scoprire le regole del gioco durante il suo utilizzo. Uno dei principali benefici cognitivi coinvolge la *representazione spaziale*, che costituisce più un insieme di abilità che una singola competenza, come la visualizzazione e la rotazione spaziale mentale, l'abilità a trattare con immagini a due dimensioni in uno spazio bidimensionale o tridimensionale. Un secondo ambito in cui si hanno sviluppi è l'*attenzione selettiva*, cioè la capacità di tenere traccia simultaneamente di molte informazioni differenti.

I videogiochi, per il raggiungimento degli obiettivi che propongono, richiedono delle operazioni cognitive complesse, come per esempio l'identificazione delle caratteristiche di ciò che si deve affrontare, la pianificazione delle proprie strategie e come possono essere applicate, il controllo e la supervisione dell'esecuzione e la visualizzazione dei risultati [5]. Quindi un altro aspetto che viene sviluppato è la propensione al *problem solving*, ovvero la propensione alla risoluzione di problemi. Questo aspetto è centrale a tutti i generi di videogiochi (a partire dai puzzle fino a quelli con contenuti violenti). In particolare, i giochi di strategia apportano evidenti miglioramenti di abilità risolutive e quindi, spesso, anche di migliori voti a scuola.

Infine, i videogiochi (violenti e no) sembrano associati anche ad una crescita della creatività. Secondo alcuni studi si nota che chi gioca ha maggiori capacità creative, ma non è chiaro se il gioco sviluppi la creatività o se alle persone creative piace giocare ai videogiochi [16].

Riassumendo, diversi tipi di videogiochi sviluppano diverse funzioni cognitive, alcune delle quali anche molto importanti e che sembrano avere un'utilità nella quotidianità delle persone. Tuttavia, molte di queste abilità vengono acquisite senza la consapevolezza del giocatore e, se questa rimane totalmente implicita, difficilmente potrà essere trasferita in altri ambiti. Se invece l'individuo diventa consapevole di cosa succede nella sua mente mentre gioca, la capacità sviluppata durante il gioco potrà essere applicata anche in altri contesti. Quindi è importante che il bambino sia aiutato a riflettere sui processi cognitivi attivi durante il gioco, in modo di portare a livello di consapevolezza le abilità acquisite. [5].

Per molti bambini, inoltre, il videogioco è ciò che li introduce nel mondo delle tecnologie informatiche. Per questo motivo essi possono essere visti come un modo per preparare le nuove generazioni ad un futuro in cui le conoscenze informatiche diventeranno sempre più importanti e fondamentali.

2.3.2 Benefici motivazionali

Il successo o il fallimento di un videogioco è strettamente legato alla sua abilità ad impegnare il giocatore nel raggiungimento dello scopo per cui è stato progettato. Il gioco deve proporre obiettivi che siano significativi per il suo utilizzatore, deve fornire incentivi per indurre a perseverare dopo molti fallimenti e premiare il successo, che è una cosa molto meno frequente rispetto ai fallimenti in un gioco stimolante. Decenni di studi psicologici sullo sviluppo e sull'educazione suggeriscono che

gli stili motivazionali caratterizzati dalla persistenza ed uno sforzo continuo sono elementi chiave che contribuiscono al successo ed alla realizzazione di una persona [11].

Secondo alcune ricerche, i bambini sviluppano convinzioni sulla propria intelligenza e le proprie abilità, le quali sottolineano specifici stili motivazionali che influenzano direttamente il loro successo. Per esempio, i bambini che vengono lodati per le loro caratteristiche (es. “*Sei un bambino intelligente*”), sviluppano una percezione dell’intelligenza come se essa sia qualcosa di innato, di fisso e che non può essere migliorato. Al contrario, i bambini che vengono lodati per i loro sforzi (es. “*Bravo, hai lavorato duramente per superare il livello!*”) sviluppano una teoria incrementale dell’intelligenza: essi credono che l’intelligenza sia malleabile, qualcosa che deve essere coltivata tramite sforzi e tempo. Questi ultimi bambini, secondo determinati studi, in circostanze di difficoltà sono più propensi ad essere persistenti, al contrario dei primi che hanno una maggiore propensione ad arrendersi. Quest’attitudine positiva attraverso i fallimenti comporta anche diverse ripercussioni sul successo nella vita reale, come per esempio migliori risultati accademici. I videogiochi quindi possono essere uno strumento molto utile per l’acquisizione di questa percezione incrementale dell’intelligenza perché forniscono feedback immediati e concreti riguardo agli sforzi che il giocatore compie (monete, punti, ricompense, . . .). Questi feedback motivazionali bilanciano i livelli di sfida e frustrazione con una sufficiente esperienza di successo e di realizzazione.

Al contrario di quanto ci si possa aspettare, queste esperienze di fallimenti e sconfitte, non comportano rabbia, frustrazione o tristezza, nonostante queste emozioni siano percepite ad intermittenza dai giocatori, ma eccitazione, interesse e gioia. Di fronte ad una sconfitta, i giocatori sono fortemente motivati a ritornare sul problema e a vincere. È provato che le persone che giocano ai videogiochi hanno un atteggiamento più persistente di fronte ai fallimenti. Questo stile motivazionale può essere generalizzato ed esportato in ambito scolastico o lavorativo, influenzando il successo della persona [16].

2.3.3 Benefici emozionali

Una delle principali ragioni per cui le persone fanno utilizzo di videogiochi è l’influenza che essi hanno sull’umore e sulle emozioni. Per esempio, alcuni studi suggeriscono che alcuni giochi migliorano l’umore, promuovono il rilassamento e riducono l’ansia. Inoltre l’influenza dei videogiochi è collegata a molti risultati positivi nella vita di un adolescente, come l’impegno ed il successo a scuola, una maggiore autostima ed un livello inferiore di ansia. Nonostante i fallimenti durante il gioco possano portare a provare emozioni negative come frustrazione, rabbia, ansia e tristezza, la simulazione del gioco può essere abbastanza reale da far sì che sia importante, per il giocatore, raggiungere gli obiettivi proposti, quindi portandolo ad abituarsi a controllare e manipolare queste emozioni negative per raggiungere il risultato desiderato [13].

Tra le emozioni positive raccontate dai videogiocatori, troviamo spesso la fierezza, ovvero un forte orgoglio che si prova quando si riesce a superare una sfida. Un’altra emozione significativa è il *flow* o flusso, che è un’esperienza descritta dai giocatori, durante la quale essi sono immersi

in un'attività che suscita un elevato senso di controllo mentre evoca contemporaneamente una perdita di coscienza di sé. In psicologia, queste esperienze sono ripetutamente collegate ad una moltitudine di risultati positivi per gli adolescenti, come l'impegno ed il successo in ambito scolastico, una maggiore autostima ed un minore livello di ansia. È dimostrato, non specificatamente all'ambito video ludico, che vivere emozioni positive possa contribuire a costruire relazioni sociali che forniscono il supporto per il perseguimento degli obiettivi e per affrontare i fallimenti. Inoltre le emozioni positive aiutano ad annullare i dannosi e demotivanti risultati dovuti ad emozioni negative. Le emozioni positive sono la base per il benessere, e comunque, se i videogiochi rendono semplicemente le persone più felici, questo è un fondamentale beneficio da considerare [16].

2.3.4 Benefici sociali

Probabilmente, la più grande differenza con i giochi odierni e quelli di dieci o venti anni fa è la loro pervasiva natura sociale. Contrariamente agli stereotipi, il videogiocatore medio non è una persona socialmente isolata. Più del 70% dei giocatori svolge questa attività con un amico, in maniera competitiva o collaborativa. Inoltre i giocatori sembrano acquisire importati abilità prosociali quando utilizzano videogiochi specificatamente sviluppati per ricompensare la cooperazione, il supporto e l'aiuto verso gli altri. Questi giochi mostrano effetti a breve termine sul comportamento di "aiuto" ed effetti longitudinali sono stati trovati nei bambini che fanno maggiore utilizzo di giochi prosociali sono più propensi ad avere atteggiamenti di aiuto verso il prossimo.

Questi effetti non si notano soltanto per questo tipo di giochi. Sorprendentemente, anche i giochi violenti possono avere effetti positivi. Quello che fa la differenza è la misura in cui questi siano competitivi o collaborativi. Per esempio, se un gioco, anche se violento, incoraggia la cooperazione tra i giocatori, si avranno gli stessi effetti positivi descritti precedentemente, in quantità maggiore rispetto a chi non utilizza questi giochi. Inoltre giocare a videogiochi violenti in gruppo, riduce i sentimenti di ostilità rispetto a chi gioca da solo.

Abilità sociali sono anche manifestate in forma di impegno civico, come la capacità di organizzare gruppi e guidare persone in cause sociali. Le persone che giocano a videogiochi con esperienze civiche (come *Guild Worlds* o giochi di ruolo multiplayer online) sono più propensi ad impegnarsi in movimenti civici e sociali nella vita di tutti i giorni (es. fare beneficenza, volontariato, ...) [16].

2.4 Il gioco per bambini con disabilità motorie

Viste tutti le componenti positive che il gioco può apportare nella vita di un bambino ed i gravi problemi che può causarne l'assenza nella loro quotidianità, pare ovvio che questo rischio sia molto più reale per bambini con gravi disabilità motorie data la loro impossibilità di interagire con normalità con giocattoli, con dispositivi elettronici e con gli altri.

Vista l'importanza del gioco, la Convenzione dell'Organizzazione delle Nazioni Unite (ONU) sui diritti delle persone con disabilità, pone una particolare attenzione al diritto al gioco per bambini

con disabilità.

L'articolo 30, "*Partecipazione alla vita culturale e ricreativa, agli svaghi ed allo sport*" stabilisce cinque punti fondamentali, di cui il quinto rappresenta quello di maggior interesse per i temi qui affrontati. Esso dice [22]:

Al fine di consentire alle persone con disabilità di partecipare su base di uguaglianza con gli altri alle attività ricreative, agli svaghi e allo sport, gli Stati Parte adottano misure adeguate a:

- a) incoraggiare e promuovere la partecipazione più estesa possibile delle persone con disabilità alle attività sportive ordinarie a tutti i livelli;
- b) garantire che le persone con disabilità abbiano la possibilità di organizzare, sviluppare e partecipare ad attività sportive e ricreative specifiche per le persone con disabilità e, a tal fine, incoraggiare la messa a disposizione, su base di uguaglianza con gli altri, di adeguati mezzi di istruzione, formazione e risorse;
- c) garantire che le persone con disabilità abbiano accesso a luoghi che ospitano attività sportive, ricreative e turistiche;
- d) garantire che i minori con disabilità possano partecipare, su base di uguaglianza con gli altri minori, alle attività ludiche, ricreative, agli svaghi ed allo sport, incluse le attività previste dal sistema scolastico;
- e) garantire che le persone con disabilità abbiano accesso ai servizi forniti da coloro che sono impegnati nell'organizzazione di attività ricreative, turistiche, di tempo libero e sportive.

Nonostante queste buone intenzioni, ad oggi, giocare non è così semplice per il bambino con disabilità e questo può avere gravi ripercussioni sul suo sviluppo cognitivo, emotivo e sulle sue opportunità di inclusione sociale. Inoltre, spesso, in particolare per bambini con gravi disabilità motorie, anche l'accesso a dispositivi elettronici moderni, come PC, tablet o smartphone può risultare un ostacolo, impedendo loro l'utilizzo di videogiochi che, come descritto in precedenza, possono avere molti riscontri positivi sugli utilizzatori.

Lo scopo di questa tesi è appunto quello di andare a sviluppare dei videogiochi adatti anche a bambini con gravi disabilità motorie, permettendo loro di giocare con gli altri, imparare, socializzare e crescere.

Capitolo 3

Confronto tra i principali meccanismi di accessibilità per persone con disabilità motorie

3.1 Tecnologie assistive

3.1.1 Definizioni

Esistono molte definizioni ufficiali per le tecnologie assistive (o AT - Assistive Technology). Di seguito viene riportata quella della Legge Stanca¹, che definisce:

- “*accessibilità*”: la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari;
- “*tecnologie assistive*”: gli strumenti e le soluzioni tecniche, hardware e software, che permettono alla persona disabile, superando o riducendo le condizioni di svantaggio, di accedere alle informazioni ed ai servizi erogati dai sistemi informatici.

Lo scopo di queste tecnologie è quello di aiutare i disabili mettendoli nelle stesse condizioni di utilizzo del computer o di altri dispositivi rispetto ad un individuo normodotato. Hanno quindi una valenza compensativa. Purtroppo, non tutta l'informazione è fornita in modo accessibile o utilizzabile da tutti allo stesso modo. Molte delle nuove tecnologie potrebbero rappresentare degli ostacoli per persone con disabilità in quanto possono avere difficoltà nell'utilizzarle, subendo le

¹Legge dell'ordinamento italiano del 9 gennaio 2004, n. 4 (G.U. n. 13 del 17 gennaio 2004), recante “Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici”

conseguenze che l'assenza di utilizzo potrebbe comportare. L'accessibilità viene dunque vista come il mezzo tramite il quale queste barriere vengono abbattute.

L'accesso ai dispositivi elettronici, ormai, è di fondamentale importanza in molti aspetti della vita, dalla comunicazione, al divertimento, al lavoro, . . . e ha implicazioni sul successo e sulle opportunità sociali per tutte le persone, disabili e non. Inoltre la tecnologia va sempre più espandendosi prendendo piede in moltissimi aspetti della vita quotidiana, quindi l'utilizzo di tecnologie assistive è essenziale per non escludere persone con disabilità da molti aspetti della vita sociale.

Ma cosa si intende per disabilità? Diamo ora alcune definizioni [24] che saranno utili in questo capitolo e nei successivi.

- **Menomazione:** qualsiasi perdita o l'alterazione di una struttura o di una funzione psicologica, fisiologica o anatomica. Queste perdite o modifiche possono essere temporanee o permanenti, e rappresentano la manifestazione di uno stato di malattia che riflette, in linea di principio, un evento fisico.
- **Disabilità:** la restrizione o la mancanza della capacità di compiere un'attività nel quadro e nei limiti considerati normali per gli esseri umani. È caratterizzata da eccessi o carenze nel comportamento o nello svolgimento di un'attività che è comune o normale.
- **Handicap:** è una condizione sociale del danno subito da un dato individuo, risultante da una menomazione o una disabilità che limita o impedisce l'esecuzione di una attività considerata normale per quell'individuo, tenendo conto dell'età, del sesso e dei fattori socio-culturali, . . . Esso rappresenta l'espressione sociale di una infermità o di una incapacità che riflette l'impatto (culturale, sociale, economico e ambientale) da essa causate.

Secondo il modello IFC² *“la disabilità non deve essere vista come l'attributo di un dato individuo, ma come una situazione che potrebbe affliggere qualsiasi persona nel caso in cui esista un divario tra le capacità individuali ed alcuni fattori ambientali, e che questo divario riduca la qualità della vita ed ostacoli il massimo sfruttamento del potenziale dell'individuo nella società”* [10].

Il compito delle AT è proprio quello di ridurre questo divario (o *gap*) come si può osservare nella Figura 3.1.

²“*International classification of Disability, Functioning and Health*”, rilasciato dalla World Health Organization (WHO) nel 2001.

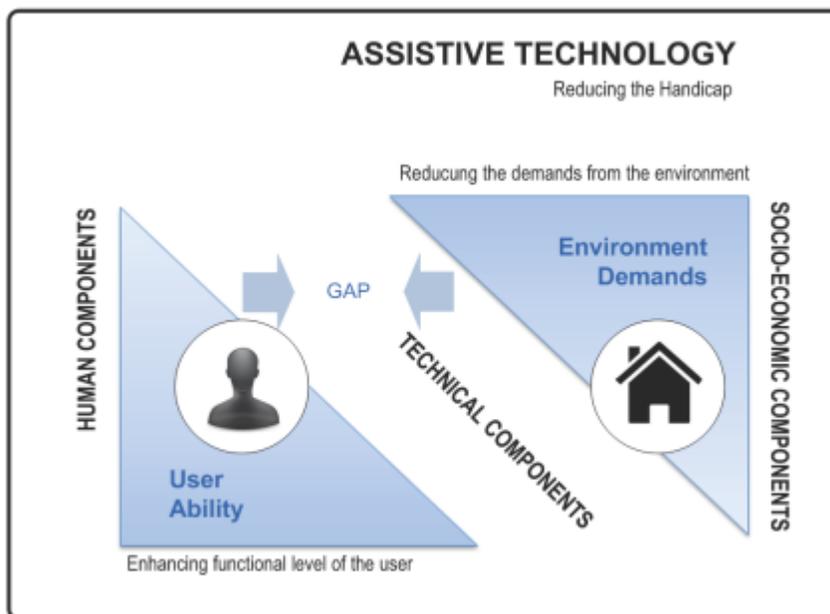


Figura 3.1: Questa immagine mostra il divario che la disabilità crea tra le capacità di una persona e le componenti socio economiche che si cerca di ridurre o eliminare grazie alle tecnologie assistive [24].

Le principali aree in cui queste tecnologie possono migliorare le condizioni dei disabili sono:

- nella prevenzione di malformazioni genetiche;
- nella riabilitazione;
- nel raggiungimento della piena inclusione sociale.

Infatti queste tecnologie possono compensare specifiche disabilità, innate o acquisite, e sono ampiamente utilizzate come uno strumento riabilitativo e di compensazione delle abilità residue [24].

Nel caso di disabilità motorie, dato l'elevato numero di menomazioni e disabilità possibili, è opportuno fornire alcune precisazioni. Non è quasi mai possibile stabilire a priori, conoscendo solamente la diagnosi, quali siano i sistemi più adatti da offrire come ausilio ad una certa persona. Solo raramente, infatti, è possibile stabilire un legame preciso tra *menomazione - disabilità - handicap*. Infatti una stessa menomazione può comportare diverse disabilità a persone diverse. Un'altra cosa da tenere in conto è la frequente presenza di più disabilità sovrapposte. Allo stesso modo non si può associare con precisione una tecnologia assistiva ad una specifica menomazione, in quanto due persone con la stessa menomazione possono aver bisogno di diversi tipi di ausili

tecnologici. Infine, non si può dire se un ausilio sia più o meno valido di un altro: come specificato precedentemente, è necessario valutarne l'utilizzo caso per caso.

Come appena detto, il numero di disabilità motorie è molto elevato. Per i nostri scopi vogliamo considerare soltanto una parte di bambini con disabilità motorie gravi, più precisamente la categoria dei quali non riescono ad utilizzare con precisione, o per nulla, i normali meccanismi di interazione con i PC (come il mouse, tastiera, touchpad o touchscreen) o addirittura neanche alcuni tipi di ausili elettronici (come joystick, trackball o puntatori tramite il movimento del capo), ma che hanno comunque la possibilità di premere, con una qualsiasi parte del corpo (come la mano, il piede, la testa, . . .), un dispositivo *single-switch*. Molto spesso, bambini con questi tipi di disabilità, oltre a problemi nell'apparato motorio, presentano altre patologie che causano una difficoltà a mantenere la concentrazione per un tempo prolungato e, in alcuni, anche problemi cognitivi e di comunicazione più o meno gravi rendendo difficile l'utilizzo di altri meccanismi sofisticati, come il riconoscimento vocale o l'oculometria (o *eye-tracking*). Tutti questi aspetti devono essere tenuti in considerazione durante l'analisi e la scelta della migliore tecnologia assistiva da utilizzare per il nostro specifico scopo.

In questo capitolo sono trattate le principali metodologie *single-switch* che facilitano l'utilizzo di dispositivi elettronici a persone con gravi disabilità motorie. Sono descritti i meccanismi a scansione ed i framework Nomon e GNomon. Non verranno quindi trattati metodi che richiedono uno sforzo muscolare troppo alto per persone disabili, come alcuni meccanismi di selezione diretta³, come joystick, trackball o sistemi di puntamento con il movimento del capo, e nemmeno quelli creati ad hoc per persone con un tipo di disabilità diverso da quello motorio.

In particolare saranno enfatizzati i pregi ed i difetti riguardanti l'applicabilità di queste tecnologie in un contesto video ludico ed il loro utilizzo da parte di un pubblico di bambini con seri problemi legati al movimento.

3.2 Meccanismi a scansione

3.2.1 Descrizione

I meccanismi a scansione sono tecnologie di selezione indiretta, ovvero che non consentono di selezionare direttamente un oggetto sullo schermo. Al contrario dei meccanismi di selezione diretta, l'utente può selezionare un oggetto solo nel momento in cui l'indicatore di scansione (cursore o evidenziatore) del dispositivo elettronico si trova sulla scelta desiderata. [1]

Nel caso di persone con disabilità motorie, questi strumenti sono sfruttati da parte dell'utente tramite l'utilizzo un dispositivo *single-switch* (come quello in Figura 3.2), ovvero un dispositivo dotato di un solo pulsante. Con il termine *single-switch* sarà definito il singolo gesto di input

³Azione che si riferisce all'atto di indicare un elemento sullo schermo, per esempio con un dito o con un dispositivo di puntamento (mouse, trackingball, touchpad, touchscreen, puntatori ad infrarossi, . . .)

ripetuto dall'utente. Definiremo invece la parola *click* come l'operazione tramite la quale viene attivato il dispositivo e quindi inviato l'input al dispositivo elettronico utilizzato.

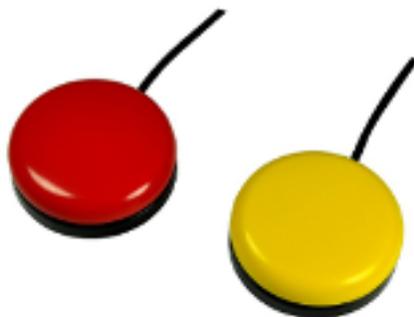


Figura 3.2: Dispositivi *single-switch*.

Nell'immagine sottostante (Figura 3.3), si può vedere un esempio di utilizzo dei meccanismi a scansione applicati ad un software di immissione di testo. Il riquadro con i bordi evidenziati di arancione è quello attualmente selezionabile. Dopo un breve ritardo verrà evidenziata la casella successiva e così via, fino alla pressione del pulsante da parte dell'utente. Sarà selezionata la lettera evidenziata al momento del click, poi il cursore verrà riportato in posizione iniziale e ricomincerà la scansione.

C	A	T				
space	E	A	R	D	U	
T	O	I	L	G	B	
N	S	F	Y	V	J	
H	C	P	K	Q	.	
M	W	X	Z	,	?	

Figura 3.3: Esempio di immissione testo con meccanismi a scansione.

Tramite la scansione, soltanto un oggetto per volta dell'insieme degli elementi selezionabili sarà evidenziato o indicato da un cursore e quindi attivo per essere scelto, solo per un breve periodo di tempo (ritardo di scansione). Se viene premuto il pulsante, verrà selezionato l'elemento attualmente evidenziato.

Questi meccanismi sono sicuramente più lenti rispetto a quelli a selezione diretta e meno intuitivi da utilizzare, ma richiedono un minore sforzo motorio per attivare il dispositivo di selezione.

3.2.2 Implementazione

Schemi di scansione

Esistono diversi schemi tramite cui implementare i meccanismi a scansione. Essi si riferiscono al modo in cui gli oggetti nell'insieme di selezione sono presentati all'utente [19].

I tre schemi principali sono:

- **Scansione circolare:** elementi individuali sono organizzati in cerchio (come i numeri di un orologio) e il cursore di scansione si muove in circolo selezionandone uno alla volta. Nonostante questo tipo di scansione sia impegnativo dal punto di vista visivo, è lo schema più semplice e cognitivamente facile da gestire.
- **Scansione lineare:** gli elementi selezionabili sono organizzati in una griglia. Il cursore di scansione evidenzia un solo elemento di ogni riga per volta. Quando viene evidenziato l'oggetto desiderato, l'utente preme il pulsante per indicare la sua scelta. Quindi basta un click per effettuare la selezione. Nonostante sia cognitivamente più impegnativo rispetto alla scansione circolare, esso è abbastanza lineare e semplice da capire. Questo schema risulta inefficiente se ci sono molti elementi selezionabili. Per esempio, in una griglia contenente 8 elementi per riga, se quello desiderato è nella settima posizione della quarta riga, il cursore di scansione deve scansare 28 elementi indesiderati prima di raggiungere quello che si vuole effettivamente selezionare.
- **Scansione a gruppi di elementi:** gli elementi selezionabili, a differenza della scansione lineare, non sono evidenziati uno per volta ma vengono scansati a gruppi (per esempio per riga o per colonna). L'utente deve prima scegliere il gruppo in cui si trova l'elemento che vuole selezionare e poi, una volta fatto, parte la scansione degli elementi in quel gruppo. Esistono diverse tipologie di scansione a gruppi, ma la più usata è la scansione riga-colonna. Questo metodo è migliore dal punto di vista prestazionale perché consente di selezionare l'elemento voluto in un tempo minore rispetto al metodo lineare. Sono però richiesti due click per effettuare la selezione.

Tecniche di controllo

Esistono diversi metodi per implementare i meccanismi a scansione. Questi definiscono il comportamento da avere tramite il quale l'utente può interagire con lo switch per scegliere l'elemento desiderato. Di seguito la descrizione di alcuni dei principali [4][19]:

- *Scansione regolare o automatica* – Quando l’utente preme il pulsante, il cursore inizia a scansire automaticamente gli elementi. L’utente preme nuovamente per indicare l’oggetto desiderato.
- *Scansione inversa o diretta* – Il cursore avanza solamente quando il pulsante è mantenuto in pressione. Ogni oggetto rimane evidenziato per un breve intervallo di tempo prima di passare al successivo. La selezione avviene quando l’utente rilascia il pulsante.
- *Scansione a step* – L’utente controlla il movimento del cursore di scansione sull’insieme di selezione premendo continuamente il pulsante per farlo avanzare fino all’elemento desiderato. In un contesto *double-switch*, in cui si hanno due pulsanti, per effettuare la selezione si preme sul secondo pulsante. Invece, in ambiente *single-switch* in cui si ha un solo pulsante, l’utente deve smettere di premere quando il cursore si trova sull’elemento desiderato. L’assenza di pressione per una quantità di tempo predeterminata rappresenta il segnale che indica che la scelta è stata effettuata.

Infine, un altro tipo di scansione riportato in letteratura è la scansione adattativa. Questo metodo è stato studiato da Simpson e Koester, secondo i quali il ritardo di scansione potrebbe essere adattato a runtime, basandosi su misurazioni delle prestazioni dell’utente. Dopo alcuni esperimenti, però, emerse che questo non portava alcun vantaggio, né svantaggio, alle prestazioni dei partecipanti [18].

Esistono comunque delle tecniche adattative che, analizzando le statistiche sulle scelte effettuate in precedenza, modificano il layout della griglia mettendo nella prima riga gli elementi più probabili in modo che le selezioni successive siano più rapide. Questo teoricamente potrebbe decisamente migliorare le prestazioni di utilizzo ma causa un aumento di difficoltà all’utente in quanto i simboli non sono sempre nella stessa posizione.

3.2.3 Limiti

Le tecnologie a scansione sono utili per determinate applicazioni ma presentano delle limitazioni se vogliamo utilizzarle nell’ambito dei videogiochi. Infatti sono meccanismi utilizzabili solamente se gli oggetti selezionabili sono disposti in griglie, in modo da poter essere evidenziati sequenzialmente (a gruppi e no). Questo è un grosso impedimento nella creazione di videogame con contenuti dinamici, che cambiano spesso (vengono creati, distrutti, nascosti, ...) e che si muovono liberamente sullo schermo. Quindi il numero di videogiochi implementabili con questo meccanismo è basso, con funzionalità statiche e quindi limitate. Inoltre, questi meccanismi non sono particolarmente efficienti perché si ha un tempo di attesa dovuto alla scansione sequenziale prima di poter arrivare a selezionare l’elemento desiderato. Come già spiegato, nel caso di alcuni bambini, bisogna tenere conto anche di problemi di concentrazione oltre a quelli di movimento, quindi non è una cosa rara che un giocatore perda il momento giusto per selezionare l’elemento dovuto. In questo caso dovrebbe aspettare di nuovo un intero giro per poterlo selezionare.

Per questi motivi, queste tecniche non sono adatte per giochi con molti oggetti selezionabili, dinamici, o che richiedano un ritardo di selezione ridotto. Sono anche meno intuitivi rispetto ai meccanismi di selezione diretta.

3.3 Nomon

Nomon è un framework utilizzabile per selezionare un qualsiasi punto su uno schermo bi-dimensionale. Questo metodo è utile come sostituto per ogni attività *punta e clicca* su di un computer. Le sue applicazioni sono molteplici, è infatti utilizzabile per la navigazione in internet, come interfaccia di scrittura e persino come strumento di disegno.

Nomon è stato presentato in dettaglio da Broderick e MacKey in una pubblicazione [8] del giugno 2009. In questa sezione saranno riportati gli aspetti più significativi ed importanti per comprenderne il funzionamento e per capire come lo si può sfruttare in un contesto video ludico per bambini con gravi disabilità motorie.

Per usufruire di questa tecnologia è richiesto un dispositivo di interazione *single-switch* come quello descritto nella sezione relativa ai meccanismi a scansione e rappresentato in Figura 3.2.

3.3.1 Visualizzazione

Ogni applicazione Nomon si presenta come un insieme di piccoli *orologi*. Questi orologi, o *clock*, sono il meccanismo di selezione e possono essere sovrapposti ad ogni tipo di sfondo. In particolare, ognuno di questi orologi appare a fianco ad ogni opzione selezionabile sullo schermo. Selezionando uno di questi orologi, automaticamente avverrà la selezione dell'elemento ad esso associato.

Su ogni orologio è presente una lancetta nera che ruota in continuazione ad una velocità fissa. Mentre la velocità è la stessa per ogni clock, la posizione angolare di ogni lancetta nera è diversa, in ogni momento, tra tutti gli orologi presenti sullo schermo. Ogni clock ha anche una lancetta rossa fissa sul mezzogiorno.



Figura 3.4: Questa immagine raffigura il classico orologio di Nomon, con la lancetta rossa fissa sul mezzogiorno e la lancetta nera che ruota a partire da uno sfasamento determinato da Nomon.

Per selezionare un'opzione, l'utente deve iniziare trovando l'orologio corrispondente. Successivamente, preme il pulsante più precisamente possibile, quando la lancetta nera si trova allineata al mezzogiorno.

Il numero di click richiesti per selezionare l'oggetto scelto è variabile e dipende sia dalla precisione della pressione dell'utente e sia dal numero di alternative disponibili. Questa flessibilità

consente operazioni efficienti per un'ampia gamma di utenti. L'interfaccia si adatta a utenti molto precisi consentendo loro di premere poche volte per effettuare una selezione. Consente anche ad utenti meno precisi degli extra click in modo che sia difficile che un errore porti ad una decisione finale errata. La probabilità di errori desiderata è uno dei parametri programmabili del software. La selezione può anche dipendere da un insieme di opzioni; infatti, spesso si hanno delle informazioni a priori riguardo a quali sono le scelte più probabili. Per esempio, in un'interfaccia di scrittura, ci si aspetta che la lettera *a* sia più probabile rispetto alla lettera *z*. Questo può migliorare l'efficienza del meccanismo riducendo il tempo di selezione.

A causa della variabilità del numero di click per effettuare una decisione, è importante che l'interfaccia di Nomon dimostri chiaramente quando una selezione è stata effettuata. Questo può essere realizzato in diversi modi, includendo sia notifiche visive che uditive. Per esempio l'opzione scelta può essere evidenziata oppure può emettere un segnale sonoro.

Questi meccanismi di notifica sono utili anche per fornire informazioni riguardanti la progressione verso la scelta finale. Si considera, come vincitore, l'elemento, e quindi equivalentemente del suo orologio, con la probabilità a posteriori⁴ più alta, data la storia dei click effettuati fino a quel momento. In Nomon “vince” l'orologio con la probabilità di essere selezionato più alta, calcolata dal software stesso. In Nomon, il colore di un clock indica la sua probabilità posteriore. In alcuni casi (soprattutto se ci sono molti orologi o se l'utente non è molto preciso) può capitare che, al momento del click, ci siano due o più orologi con la lancetta in movimento vicina a mezzogiorno, quindi che ci siano orologi con probabilità alta ma non abbastanza da essere dichiarati vincitori. In questo caso possono sorgere delle ambiguità, quindi, per gestire questo problema, Nomon prevede ulteriori round, in cui la competizione è soltanto più tra questi due o più orologi.

Più specificamente, gli orologi con una probabilità posteriore superiore ad una certa soglia (quindi più probabili) sono colorati di giallo (Figura 3.4), e quelli con la probabilità posteriore sotto la stessa soglia (quindi meno probabili) sono colorati di bianco. Il colore serve per indicare la progressione verso la selezione. Al procedere dei click, tipicamente, il clock desiderato diventa o rimane giallo, mentre progressivamente tutti gli altri diventano bianchi. Può diventare bianco a causa di selezioni poco precise, quindi segnala una maggiore attenzione per selezionare quell'opzione.

Per avere più informazioni su Nomon, si può fare riferimento al documento “*Nomon: efficient communication with a single switch.*” di Tamara Broderick [8] in cui sono illustrati i dettagli specifici del funzionamento di questo interessante strumento.

Nomon è un meccanismo molto adatto per la creazione di videogiochi, grazie alla sua elevata flessibilità ed al suo funzionamento come meccanismo di selezione diretta. Grazie alle sue caratteristiche permette ad utenti precisi di arrivare alla selezione con pochissimi click, quindi in

⁴La probabilità a posteriori di un evento aleatorio è la probabilità condizionata che è assegnata dopo che l'informazione rilevante posta in evidenza è tenuta da conto.



Figura 3.5: Questa immagine raffigura un orologio diventato giallo dopo un click, indicando che è uno di quelli con la probabilità posteriore maggiore. L'operazione di colorare di giallo gli orologi più probabili permette di visualizzare una progressione verso la selezione finale.

breve tempo, ed evita che gli errori di utenti meno precisi possano portare alla selezione errata di un'opzione non desiderata.

3.4 GNomon

GNomon è un framework creato dal gruppo *e-Lite*⁵ del Politecnico di Torino. Le informazioni di seguito riportate fanno riferimento al documento *“Playable One-Switch Video Games for Children with Severe Motor Disabilities Based on GNomon”* di Aced López, Corno e De Russis [1].

Lo scopo principale di questo framework è quello di sfruttare le capacità di Nomon adattandolo e configurandolo per l'utilizzo in un contesto videoludico, consentendo la creazione di videogiochi *single-switch*, dinamici e accessibili. Questo meccanismo consente di selezionare uno tra molti possibili elementi presenti sullo schermo senza richiedere che questi elementi siano organizzati in particolari configurazioni sullo schermo o che siano statici.

In particolare GNomon fornisce funzionalità per creare insiemi di oggetti di gioco selezionabili con associato un orologio, come per Nomon, per abilitare meccanismi di gioco dinamico *punta-e-clicca* utilizzando un dispositivo *single-switch* in ingresso. Questi insiemi possono essere ridimensionati in ogni istante aggiungendo o rimuovendo elementi, aspetto necessario per il supporto di azioni proprie di videogiochi dinamici, come la creazione o la distruzione di un elemento o di un personaggio. Inoltre, questo framework può essere utilizzato per creare giochi statici (giocabili con meccanismi a scansione) rendendoli più appetibili e divertenti senza vincoli riguardo alla disposizione degli elementi sullo schermo e con tempi di latenza molto minori rispetto a quelli richiesti dai meccanismi a scansione.

Di seguito, in Figura 3.6, è rappresentata l'istantanea di un videogioco che utilizza GNomon. Il gioco in questione è *“One Switch Invaders”*, sviluppato dal gruppo *e-Lite* del Politecnico di Torino. Lo scopo del gioco è quello di sparare agli alieni prima che raggiungano terra. Si noti che ad ogni alieno è associato un orologio in modo da far sì che essi siano selezionabili indipendentemente dalla loro posizione istantanea e dal loro movimento.

⁵<http://elite.polito.it>



Figura 3.6: Gioco “One Switch Invaders”.

3.4.1 Differenze con Nomon

Per definire le principali linee guida di GNomon, si è lavorato in stretta collaborazione con alcuni logopedisti, fisioterapisti e psicologi di un’Azienda Sanitaria Locale (ASL) di Torino. Essi hanno attivamente consigliato importanti scelte di design e proposto funzionalità nell’interesse dei bambini con disabilità da loro seguiti.

Queste raccomandazioni, di carattere estetico e funzionale, hanno portato GNomon a differire da Nomon sotto alcuni aspetti. Di seguito le principali raccomandazioni:

1. *Fornire indicatori addizionali per facilitare l’interazione.* In particolare, quattro segni circolari sono stati posti nei quarti dell’orologio (ad ore tre, ore sei, ore nove e ad ore dodici) per facilitare l’interazione dei bambini con lunghi periodi di latenza muscolare (tempo trascorso tra il comando di movimento ed il movimento effettivo del muscolo) aiutandoli con indicazioni di quando iniziare a ”prepararsi” alla pressione dello switch.
2. *Fare in modo che l’orologio sia più accattivante.* Gli oggetti del gioco di solito sono molto attrattivi, colorati ed animati, quindi tendono a catturare l’attenzione dei bambini. Inoltre, siccome gli orologi sono soltanto un mezzo per selezionare gli oggetti del gioco, è normale che i primi siano meno attrattivi che i secondi. Comunque, l’apparenza degli orologi è stata ridisegnata per fare in modo che siano il più accattivanti possibile (mantenendo la loro semplicità) per evitare che i bambini più distratti ignorino gli orologi. In particolare, i clock sono stati allargati, i colori resi più brillanti e con maggior contrasto, le linee inspessite e la

lancetta rotante è stata resa appuntita. La Figura 3.7 mostra le differenze tra l'apparenza classica e quella revisionata dell'orologio.

3. *Ridurre la velocità media di rotazione della lancetta in movimento e renderla personalizzabile.* Il periodo di rotazione di questa lancetta va da 1 ai 10 secondi, per consentire a bambini con lunghi periodi di latenza muscolare di divertirsi con giochi basati su GNomon. La velocità di rotazione può facilmente essere modificata dalle persone che si occupano dei bambini.
4. *Fornire feedback sonoro e visivo quando un orologio è selezionato.* Oltre a questo feedback visivo e sonoro ed il comportamento azionato nel gioco, anche l'orologio cambia colore ed un breve segnale sonoro viene riprodotto.
5. *Dinamicità degli orologi.* Gli orologi, in GNomon, possono essere fatti muovere liberamente nell'area di gioco, possono essere rimossi e se ne possono inserire altri a runtime. Questo è un vantaggio fondamentale per la creazione di videogiochi dinamici.

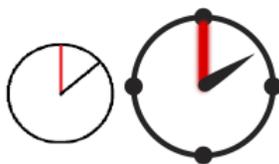


Figura 3.7: Questa immagine mostra, a sinistra, l'orologio originale di Nomon e, a destra, quello utilizzato da GNomon con le importanti modifiche consigliate da esperti: linee più spesse, dimensione maggiore, indicatori ai quarti e lancetta nera appuntita.

Dopo alcune sperimentazioni dei primi giochi implementati con questa tecnologia ci si è resi conto che, dopo un click, gli orologi con probabilità più alta, diventando gialli, risultavano poco diversi da quelli bianchi, e quindi non molto appariscenti, rispetto a quelli bianchi, per l'occhio dei bambini, dato il poco contrasto tra i due diversi colori. Per questo motivo si sono apportate ulteriori modifiche facendo diventare questi orologi di colore rosso anziché giallo.

Sono inoltre state effettuate delle sperimentazioni [2] per studiare l'applicazione di meccanismi di selezione diversi dagli orologi, come si può vedere in Figura 3.8. Questo perché i bambini hanno una certa difficoltà nel seguire la lancetta e a premere il pulsante nel momento in cui essa si trova a mezzogiorno. Questi meccanismi, infatti, sfruttano la capacità dei bambini di riconoscere il concetto di vuoto-pieno e grande-piccolo, rendendo più facile e comprensibile queste metodologie rispetto all'orologio.

Nell'immagine a sinistra si ha un triangolo rettangolo vuoto che viene riempito, da sinistra a destra, colorandosi di blu. Quando è completamente blu, diventa azzurra per un istante e si ha l'equivalente della lancetta in movimento posizionata a mezzogiorno dell'orologio Nomon. Nell'immagine a destra, invece, si ha un cerchio con all'interno una piccola palla di colore blu. Questa



Figura 3.8: Meccanismi di selezione alternativi di GNomon.

palla cresce continuamente fino a raggiungere la dimensione del cerchio che la contiene e diventa azzurra per qualche istante. Questo momento corrisponde esattamente all'istante in cui la lancetta in movimento dell'orologio è posizionata a mezzogiorno.

Anche utilizzando questi meccanismi, come gli orologi, in caso di ambiguità, gli elementi più probabili cambiano il proprio colore diventando rossi.

Da queste sperimentazioni è emerso che la maggior parte dei bambini preferisce la palla, anziché l'orologio o il triangolo, ritenuto il più difficile da utilizzare.

Capitolo 4

Obiettivi e proposte

In questo capitolo sono riportate le considerazioni, gli obiettivi e le proposte che hanno guidato alla decisione di realizzare due giochi dedicati a bambini con disabilità motorie, che sfruttano le potenzialità di GNomon ed utilizzano come input un dispositivo *single-switch*.

4.1 Obiettivi

Il principale obiettivo che ha portato alla realizzazione del primo videogioco è l'ideazione, lo sviluppo e la sperimentazione di un videogioco il cui scopo è quello di permettere a bambini con disabilità motorie gravi di giocare assieme ad altri bambini, disabili e non, con lo scopo di farli socializzare e divertire insieme. Inoltre è stato ideato ed implementato, ma non ancora sottoposto a sperimentazione, un altro gioco *single-player* con diverse componenti educative per aiutare questi bambini a sviluppare determinate capacità.

4.1.1 La socializzazione

Il pericolo che i bambini con disabilità si sentano soli ed esclusi dalle attività svolte dagli altri bambini è molto alto. Infatti, per capire questo problema, viene riportata una ricerca [20] volta a conoscere le paure ed i desideri di 46 bambini tra i 7 ed i 9, tutti con disabilità motorie, maschi e femmine. La scelta di questa età è stata fatta perché il bambino è in una fase di sviluppo in cui la crescita è rapida ed avvengono in lui cambiamenti importanti. A questi bambini vennero fatte due domande: “*Qual è la tua paura più grande?*” e “*Qual è il tuo desiderio più grande?*”. Per i nostri scopi ci concentriamo sulla parte relativa ai desideri. I bambini potevano rispondere a voce o per iscritto, liberamente e senza limiti di tempo. Data la grande varietà di risposte, esse sono state classificate in diverse categorie di desideri:

- *Reale/Essere*: riguardanti la propria identità;
- *Reale/Avere*: riferiti al possesso di oggetti realmente esistenti;

- *Fantasia/Essere*: riguardanti la possibilità di assumere identità fantastiche;
- *Fantasia/Avere*: riferiti al possesso di oggetti non esistenti nella realtà;
- *Desiderio/Sentimento*: personali, riguardanti la sfera dei sentimenti;
- *Desiderio/Sociale*: relativi alla realtà sociale circostante.

La cosa interessante che emerge dalle risposte relative ai desideri è che, al secondo posto, per i bambini disabili emerge la categoria *Desiderio/Sentimento* (23%) che invece raccoglie soltanto il 5% di risposte dei bambini normodotati. Questa differenza, tra disabili e non, cresce leggermente nei bambini di 9 anni. Alcune delle risposte sono state, per esempio, “Vorrei restare con i miei amichetti”, “uscire con gli amici”, “Vorrei mangiare a scuola per stare in compagnia”, . . .

Questo è dovuto al fatto che, molto probabilmente, i bambini disabili hanno un forte bisogno di contatti affettivi con i coetanei, con i genitori o con figure di tipo genitoriale, come insegnanti o ragazzi più grandi. Questo desiderio è importante e deve essere soddisfatto il più possibile perché è proprio grazie al confronto con gli altri che il bambino cresce socialmente, cognitivamente ed affettivamente.

Il primo gioco sviluppato vuole quindi offrire un’opportunità di soddisfare il desiderio di compagnia di questi bambini, facendo in modo che essi abbiano una partecipazione attiva e diretta nell’utilizzo del videogioco e che essi possano giocare pressoché ad armi pari con i loro coetanei normodotati.

4.2 Vincoli

I giochi sviluppati sono nati cercando di rispettare diverse caratteristiche date dagli obiettivi proposti e le difficoltà che possono incontrare i bambini con problemi motori durante lo svolgimento di questa attività:

- il primo gioco deve essere multiplayer ed organizzato in modo per cui almeno un giocatore sia un bambino con disabilità motorie, il quale può interagire sfruttando la tecnologia offerta da GNomon;
- entrambi i giochi devono essere dinamici, in modo da sfruttare appieno le potenzialità di GNomon. Se fossero statici si potrebbero implementare con meccanismi a scansione;
- il primo gioco deve avere un carattere collaborativo, ovvero i due bambini devono giocare insieme, collaborare, per raggiungere uno scopo comune. I giocatori vincono o perdono insieme. Questo perché altrimenti il bambino disabile sarebbe svantaggiato già in partenza rispetto all’altro giocatore;

- entrambi i giochi devono avere delle caratteristiche educative, ovvero devono andare ad aiutare ad accrescere e sviluppare delle capacità dei bambini per stimolarli e farli crescere dal punto di vista cognitivo;
- i giochi devono avere un numero limitato di possibili errori in modo che sbagli dovuti al tempismo ed alla scarsa reattività non rovinino il divertimento;
- il numero di decisioni da prendere non deve essere elevatissimo per non causare troppa confusione sullo schermo rendendo difficile il focalizzarsi su di un orologio per volta;
- i giochi non devono richiedere tempi di reazione brevi perché, a causa della disabilità motoria, sarebbe troppo difficile per i giocatori; inoltre ci va comunque un po' di tempo prima che l'orologio (o il widget) di GNomon arrivi al punto di selezione e quindi il tempismo non è solo dovuto alle tempistiche di reazione degli utenti.

4.3 Le proposte

Di seguito, vengono descritti i principali passi che hanno portato alla realizzazione dei giochi sviluppati in questa tesi. Si è iniziato con un portafoglio di proposte contenente una dozzina di idee. Dopo un primo incontro informale con i relatori di questa tesi si è fatta la scelta di scartare i giochi con una giocabilità scarsa, quelli con componenti principalmente statiche e che quindi non avrebbero sfruttato al meglio le potenzialità di GNomon, quelli con una difficoltà troppo elevata per il dominio di bambini a cui si vogliono dedicare e quelli ritenuti meno divertenti ed educativi. Dopo questa prima scrematura, le principali proposte sono rimaste tre:

4.3.1 *Il labirinto dei mostri*

Gioco multiplayer simile a un labirinto in cui lo scopo di entrambi i giocatori è quello di far raggiungere al proprio personaggio un punto specifico nell'ambiente di gioco, cercando la strada giusta per arrivarci, evitando ostacoli, imprevisti e cercando di raccogliere delle banane (punti extra) sul percorso. Il multiplayer è di carattere collaborativo in quanto i due giocatori devono mettersi d'accordo per arrivare a destinazione senza ostacolare l'altrui movimento e, in alcuni casi, funzionare da appoggio all'altro per superare il livello. Un personaggio, in ogni istante, può muoversi al massimo in quattro direzioni (destra, sinistra, verso l'alto e verso il basso). Ogni direzione che il personaggio che utilizza GNomon può intraprendere in un determinato momento viene indicata con una freccia colorata a cui è assegnato un orologio (o un widget alternativo) GNomon per permetterne la selezione con un dispositivo *single-switch*. L'altro personaggio invece può essere fatto muovere utilizzando le quattro frecce di una normale tastiera per PC. Quando si seleziona il movimento, il personaggio si muove nella direzione scelta finché non trova un ostacolo (o l'altro personaggio) sulla sua strada che ne interrompe la corsa. A quel punto riappaiono le frecce

e si può procedere con la scelta del movimento successivo. I livelli possono crescere di difficoltà cambiando i percorsi e aumentando il numero di elementi e ostacoli presenti nel labirinto.

4.3.2 *Unisci i puntini*

Lo scopo di questo gioco è quello di creare un'immagine unendo dei puntini, presenti sullo schermo, ad ognuno dei quali è associato un orologio (o widget alternativo) GNomon per permetterne la selezione. Quando i puntini sono tutti collegati, spariscono in dissolvenza mostrando l'immagine che vanno a formare. Questo gioco può avere diverse componenti educative, per esempio guidando i bambini ad unire i puntini seguendo una sequenza, che può essere logica, numerica o alfabetica. Il multiplayer può essere implementato in maniera collaborativa, quindi ogni giocatore ha una sequenza separata di puntini che, insieme, formeranno l'immagine finale. Oppure, può essere implementato in maniera competitiva, in cui lo schermo viene diviso in due parti ed ogni giocatore deve costruire la sua immagine unendo i puntini nel minor tempo possibile. Ovviamente vince chi impiega meno tempo. Questo gioco può anche essere sviluppato in modo che sia giocabile anche da un giocatore singolo. Per aiutare i bambini che magari non conoscono ancora l'alfabeto o i numeri, si è pensato di proporre due tipi di suggerimenti durante il gioco. Il primo è quello di evidenziare in qualche modo (facendolo muovere, cambiandone il colore, ...) il prossimo punto da selezionare. Il secondo meccanismo è quello di tracciare una linea tratteggiata che percorre il percorso da seguire, quindi dando già in anticipo ai bambini un'idea di quella che sarà poi l'immagine finale da ricostruire.

4.3.3 *Palloncini*

Gioco il cui scopo è quello di fare scoppiare dei palloncini che fluttuano sullo schermo, ad ognuno dei quali è associato un widget GNomon per permetterne la selezione utilizzando un dispositivo *single-switch*. Questo gioco può essere sia competitivo, quindi facendo superare il livello quando entrambi i bambini riescono a fare esplodere un certo numero di palloncini a testa, e sia collaborativo, facendo vincere il giocatore che riesce a scoppiarne di più in un determinato intervallo di tempo. Questo gioco presenta molteplici potenzialità educative a seconda delle possibili modalità di gioco. Per esempio si possono far fare scoppiare soltanto palloncini di un determinato colore, oppure seguendo una sequenza di colori, di dimensione, di numeri o di lettere associati ad ogni palloncino. Anche questo gioco può essere implementato in modalità single-player.

4.3.4 *La scelta*

Al fine di effettuare la scelta su quale gioco sarebbe stato meglio sviluppare, consigli ed aiuti sono stati richiesti a persone che lavorano quotidianamente a stretto contatto con bambini con disabilità motorie gravi. In particolare sono stati organizzati degli incontri con delle logopediste, di un'Azienda Sanitaria Locale (ASL) di Torino, le quali ci hanno segnalato alcuni punti di forza

e debolezze delle tre alternative proposte, quale potrebbe essere quello più stimolante, divertente ed educativo. Tutte e tre le proposte relative ai giochi sono state commentate positivamente da queste specialiste:

- *Il labirinto dei mostri*: questo gioco è stato apprezzato perché presenta diverse potenzialità di carattere educativo che potrebbero essere utili per i bambini. In particolare questo gioco sviluppa l'abilità di organizzazione spaziale, la capacità di orientarsi, di pianificare un percorso ed una strategia per arrivare a destinazione. Sviluppa la direzionalità¹, abilità visuo spaziale, l'attenzione dei bambini e, infine, insegna loro l'importanza della collaborazione.
- *Unisci i puntini*: la reazione a questo gioco è stata positiva in quanto ci hanno detto che è una buona opportunità per i bambini con disabilità motoria di svolgere questo tipo di attività in quanto non hanno la capacità fisica di poterla svolgere con successo su carta. Hanno proposto anche un suggerimento per una modalità di gioco alternativa, in cui ad ogni puntino sono associate delle lettere, e sono da collegare non in ordine alfabetico ma in ordine logico. Per esempio se il disegno che dovrà uscire è una mela, le lettere sparse per lo schermo saranno M - E - L - A, da unire in sequenza.
- *Palloncini*: questo gioco è stato commentato in maniera positiva per le molteplici potenzialità educative relative alle diverse possibili modalità di gioco. Hanno suggerito di studiare bene la grafica a seconda dell'età dei bambini a cui vogliamo dedicare il gioco. Un consiglio su questo gioco è stato quello di non creare dinamicamente durante il livello un numero di palloncini che salgono dal basso e si muovono lentamente verso l'alto scomparendo dallo schermo quando ne raggiungono il limite superiore, ma di farli rimbalzare continuamente sulle pareti della finestra di gioco finché non vengono fatti esplodere dai bambini. Questo per ridurre la possibilità di errore e per far sì che i bambini siano più tranquilli, consentendo loro di focalizzarsi sui palloncini da scoppiare senza l'ansia che questi spariscono dallo schermo da un momento all'altro.

Per questi motivi, seguendo i consigli avuti, il primo gioco che è stato sviluppato è: *Il labirinto dei mostri*¹. Come secondo gioco, riadattato in modalità *single-player* è stato sviluppato *Palloncini*. Entrambi i giochi saranno descritti nel dettaglio nei capitoli successivi. Il gioco *Unisci i puntini* per il momento non è ancora stato sviluppato e potrà essere uno dei futuri lavori legati a questa tesi.

¹Ovvero la capacità di identificare la parte destra e sinistra nello spazio e nella relazione fra gli oggetti.

Capitolo 5

Il labirinto dei mostri

5.1 Il gioco

Il labirinto dei mostri è un gioco multiplayer, che consente a due giocatori, di cui uno disabile motorio e l'altro normodotato, di giocare insieme. Il giocatore con disabilità sfrutta, per interagire con il gioco, le funzionalità offerte da GNomon utilizzando un dispositivo *single-switch*. L'altro giocatore invece utilizza il mouse e la tastiera.

Questo gioco è simile a un labirinto all'interno del quale ci sono due personaggi diversi (mostri colorati), due pulsanti dello stesso colore dei personaggi ed altri elementi che saranno descritti in seguito. I due giocatori, ognuno dei quali utilizza un personaggio, hanno lo scopo di fare muovere il proprio mostro in modo da portarlo sopra al pulsante avente lo stesso colore del personaggio utilizzato.

La caratteristica principale di questo videogioco è la componente collaborativa. In questo gioco non si ha un singolo vincitore, ma i due giocatori devono mettersi d'accordo, organizzarsi e coordinarsi in modo da completare insieme i labirinti. Molti livelli, infatti, non possono essere superati senza una collaborazione tra i due giocatori.

5.2 Descrizione

La scena iniziale del gioco si presenta come in Figura 5.1. Come si nota in questa immagine, tutti i pulsanti di gioco selezionabili hanno associato un widget GNomon per consentire al giocatore disabile di poter gestire quasi completamente il gioco, di muoversi tra gli stage, selezionare i livelli, attivare o disattivare la musica di sottofondo e giocare. In particolare, in questa scena si ha la possibilità di uscire dal gioco (pulsante “*Esci*” in alto a sinistra), di attivare o disattivare la musica principale (pulsante “*Audio*” in alto a destra) e di iniziare il gioco (pulsante “*Inizia*” in basso). In alto al centro si trovano altri due pulsanti. Il pulsante “*Crediti*” serve per visualizzare la scena in cui sono riportate le attribuzioni per gli elementi del gioco che sono protetti da copyright e



Figura 5.1: Schermata iniziale.

per i quali l'utilizzo è consentito a patto di riportare il collegamento al sito web da cui sono stati scaricati. L'altro pulsante, "Reset", è utile perché quando si esce dal gioco viene automaticamente salvata la situazione attuale rappresentante la progressione del gioco, ovvero quali livelli sono stati superati, con che punteggio e varie impostazioni. Con questo pulsante si possono cancellare queste informazioni e ricominciare il gioco da capo. A questo pulsante non è stato associato un widget di GNomon per evitare, dato che l'azione descritta non è reversibile, che venga premuto per errore cancellando informazioni utili.

Premendo sul pulsante "Inizia", si accede alla scena delle impostazioni (Figura 5.3). In questa scena, i giocatori possono inserire il proprio nome, selezionare il mostro da comandare e si può selezionare, per il giocatore che utilizza GNomon, il widget e la velocità di rotazione (per l'orologio) o di riempimento (per la palla).

I mostri selezionabili sono sei, tutti di colori diversi e si possono vedere in Figura 5.2. Questo permette ai bambini di scegliere il proprio personaggio con cui giocare.



Figura 5.2: Mostri.



Figura 5.3: Schermata impostazioni.

5.3 Organizzazione

Il gioco è composto da 60 livelli, organizzati in 6 stage diversi, da 10 livelli ognuno.

Ogni stage serve per introdurre un nuovo elemento di gioco, rendendo quindi i livelli man mano più complessi ed articolati. All’inizio del gioco tutti gli stage sono bloccati, ad eccezione del primo, come si può osservare in Figura 5.4. Si noti che soltanto agli stage sbloccati viene assegnato il widget di GNomon, consentendone così la selezione tramite il pulsante *single-switch*. I pulsanti relativi agli stage bloccati vengono disabilitati in modo che, anche se selezionati con il mouse, non rispondano all’input. Per accedere ad uno stage bisogna aver completato quello precedente. Solo a quel punto lo stage successivo verrà sbloccato, il suo pulsante verrà abilitato e gli verrà associato un widget GNomon.

Lo stesso discorso vale per i livelli. All’inizio del gioco e quando viene sbloccato un nuovo stage, soltanto il primo livello è accessibile. Per sbloccare un livello bisogna, come per gli stage, completare quello precedente. Inoltre, come si vede in Figura 5.5, ad ogni livello sbloccato sono associate 3 stelle che servono per rappresentare il punteggio ottenuto.

Se il livello non è stato superato o è stato superato col minimo punteggio possibile, le tre stelle saranno bianche. Man mano che il punteggio cresce si coloreranno progressivamente di giallo. Nella figura in esame, per esempio, i livelli 1 e 2 sono stati superati col punteggio massimo, il livello 8 e

il 10 con un terzo del punteggio massimo e tutti gli altri con il minimo.



Figura 5.4: Scena di selezione stage.



Figura 5.5: Scena di selezione livelli.

5.3.1 Regole del gioco

In questo gioco non ci sono particolari regole. Entrambi i personaggi si possono muovere liberamente all'interno del labirinto, possono scontrarsi l'uno contro l'altro, utilizzare o evitare gli elementi aggiuntivi di gioco, raccogliere oggetti e cercare di ottenere il maggior punteggio possibile.

5.4 Movimento

In questa sezione viene descritto come i personaggi si muovono all'interno della sezione di gioco.

- Il personaggio del giocatore normodotato si muove semplicemente utilizzando le frecce della tastiera.
- Il personaggio utilizzato dal bambino disabile sfrutta la tecnologia GNomon. Esso viene circondato, nelle possibili direzioni in cui si può muovere, da delle frecce a cui è associato un orologio (o widget alternativo) per consentirne la selezione. Una volta selezionata la direzione desiderata, le frecce ed i widget scompaiono durante il movimento del personaggio e ricompaiono quando esso si ferma.

Una volta selezionata una freccia, sia con GNomon che con la tastiera, il personaggio si muove nella direzione scelta finché non incontra un ostacolo (il bordo del labirinto, l'altro personaggio o alcuni elementi del gioco) che ne causa l'arresto.

5.5 Punteggio

Lo scopo del gioco è quello di portare i due mostri sopra ai pulsanti aventi il loro stesso colore. Il raggiungimento di questo obiettivo comporta 100 punti e rappresenta l'unica condizione necessaria per superare il livello. In ogni livello, ad eccezione del primo (che serve principalmente ad introdurre il gioco), sono presenti tre banane. Questi elementi possono essere raccolti da entrambi i personaggi quando, muovendosi, ci passano sopra. Ogni banana consente di ottenere 300 punti aggiuntivi. In conclusione, il punteggio minimo per superare il livello è 100 punti e il massimo, avendo 3 banane, è 1000. Se il livello viene superato con solo 100 punti non si ottiene nessuna stella, se superato con 400 punti se ne ottiene una sola, con 700 punti due stelle e con 1000, punteggio massimo, tre. Alla fine di ogni livello viene riprodotta una clip audio trionfale e viene mostrato, come in Figura 5.6, il numero di stelle ottenute. In questo caso è stato completato il livello 18, avendo ottenuto due stelle su tre.



Figura 5.6: Risultato livello.

In questa scena ci sono quattro pulsanti:

- in alto a sinistra il pulsante “*Indietro*” che permette di tornare alla pagina di selezione dei livelli;
- in alto a destra il pulsante “*Audio*” che permette attivare o disattivare la musica principale del gioco;
- in basso a sinistra il pulsante “*Rigioca*” che permette di ricominciare il livello appena terminato;
- in basso a destra il pulsante “*Livello Seguente*” che permette procedere con il livello successivo;

Inoltre, quando si completa uno stage, per segnalare ai giocatori che si è raggiunto un traguardo, viene mostrata la scena rappresentata in Figura 5.7. Compare la scritta “*Stage finito*” e le informazioni relative all’ultimo livello dello stage completato, il tutto ravvivato da un’animazione rappresentante una pioggia di stelle.



Figura 5.7: Stage completato.

In conclusione, al completamento del gioco, si mostra la scena rappresentata in Figura 5.8, con la pioggia di stelle ed un effetto sonoro trionfale. In questa scena compare anche il pulsante “Esci”, per chiudere l’applicazione.



Figura 5.8: Gioco completato.

5.6 Livelli

La sottostante Figura 5.9 riporta una scena rappresentante un livello di gioco. La zona in cui i personaggi si possono muovere è quella pavimentata dalle caselle grigie e delimitata dai bordi blu con i tratteggi gialli. Oltre ai personaggi sono presenti i due pulsanti colorati sui quali i giocatori devono portare i mostri ed altri elementi di gioco (banane, tapis roulant) che verranno descritti nella sezione seguente.



Figura 5.9: Livello 17.

Si noti che vengono mostrate soltanto le frecce rappresentanti le possibili direzioni in cui il personaggio che utilizza GNomon può intraprendere in quel momento. Inoltre, in alto a sinistra, è presente il tasto “Pausa” che ferma il gioco e mostra la scena raffigurata in Figura 5.10. In questa schermata sono presenti quattro pulsanti che permettono di tornare indietro alla selezione del livello, di attivare o disattivare la musica del gioco, ricominciare il livello dall’inizio o di continuare da dove si era interrotto.



Figura 5.10: Schermata di pausa.

5.7 Elementi di gioco

Come già scritto in precedenza, il gioco è suddiviso in 6 stage, ognuno dei quali introduce un nuovo elemento, apportando difficoltà e varietà al gioco. Nelle successive sottosezioni vengono descritti questi elementi e il loro funzionamento.

5.7.1 Stage 1 - Banana

In ogni livello, ad eccezione del primo, sono presenti tre banane (Figura 5.11). Come già spiegato nella sezione relativa al punteggio, ogni banana, se raccolta, aumenta il punteggio di 300 punti.



Figura 5.11: Banana.

5.7.2 Stage 2 - Tapis Roulant

Questo elemento consiste in una casella con due frecce che lampeggiano di giallo, indicanti una direzione (Figura 5.12). Queste infatti possono essere orientate verso l'alto, verso il basso, verso destra o verso sinistra. Passando sopra a questo elemento, il personaggio viene spinto nella direzione indicata dalle frecce. In alcuni livelli, in cui è presente più di un tapis roulant, è possibile che questi elementi creino un *loop* nel quale il personaggio resti intrappolato. In questi casi può uscirne solo con l'aiuto dell'altro personaggio, oppure ricominciando il livello dall'inizio.

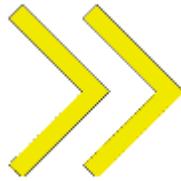


Figura 5.12: Tapis roulant.

5.7.3 Stage 3 - Muro Rosso

Questo elemento consiste in un muro di mattoni rossi (Figura 5.13). Se il personaggio va a collidere contro questo elemento, si ferma, ma il muro viene distrutto. È quindi un appoggio utile ma che può essere utilizzato soltanto una volta. È importante, nel gioco, fare attenzione a non distruggere questi elementi prima del dovuto o a non distruggerli quando potrebbero essere utili all'altro personaggio.

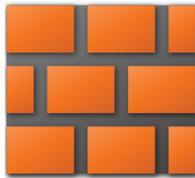


Figura 5.13: Muro rosso.

5.7.4 Stage 4 - Spirale

Questo elemento consiste in una spirale blu che ruota in continuazione (Figura 5.14). Se il personaggio, muovendosi, passa sopra a questo elemento, viene teletrasportato in una diversa zona del labirinto, in cui è presente una seconda spirale. La spirale può essere utile per raggiungere zone del labirinto normalmente non accessibili e per portarsi in posizioni migliori per raggiungere la posizione finale.



Figura 5.14: Spirale.

5.7.5 Stage 5 - Muro Verde

Questo elemento consiste in un muro di mattoni verdi (Figura 5.15 a sinistra). Se il personaggio va a collidere contro questo elemento, si ferma. In questo caso, però, il muro non viene distrutto. Questo muro può essere rimosso solamente quando un giocatore passa sul pulsante circolare verde (Figura 5.15 a destra). Viene spesso utilizzato come ostacolo per un giocatore che può però essere rimosso dall'altro. A volte invece viene utilizzato come appoggio, quindi bisogna fare attenzione ed evitare di distruggerlo.

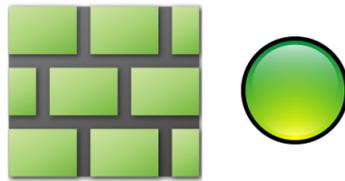


Figura 5.15: Muro verde e pulsante per distruggerlo.

5.7.6 Stage 6 - Percorso Trappola

Questo elemento consiste in un percorso caratterizzato dall'immagine nella figura sottostante (Figura 5.16). Questo elemento rappresenta un passaggio che può essere utilizzato solamente una volta. Quando un personaggio ci passa sopra, infatti, questo elemento si trasforma in un muro di mattoni verdi che non può essere distrutto impendendo al giocatore di tornare indietro per la stessa strada.



Figura 5.16: Percorso utilizzabile solamente una volta.

5.8 Componenti educative

Questo gioco, nel rispetto degli obiettivi proposti, presenta diverse caratteristiche educative volte a sviluppare alcune abilità dei bambini che ne usufruiscono. Tra queste troviamo:

- **Flessibilità cognitiva:** capacità di adattarsi a nuove situazioni di apprendimento e di muoversi abilmente tra vari stati mentali [21]. In questo gioco infatti i livelli sono sempre diversi e, man mano, vengono introdotti nuovi elementi che obbligano il giocatore ad adattarsi a questi cambiamenti.
- **Memoria di lavoro:** questo termine fa riferimento al sistema cerebrale che fornisce il salvataggio temporaneo e la manipolazione delle informazioni necessaria per lo svolgimento di compiti cognitivamente complessi, come per esempio il linguaggio, la comprensione, l'apprendimento ed il ragionamento [6].
- **Pensiero strategico:** consiste nella capacità di identificare ed analizzare problemi, di creare e monitorare dei piani per la loro risoluzione [17]. Il pensiero strategico è composto da tre parti fondamentali:
 - formulazione del problema da risolvere;
 - selezione di appropriate strategie per risolverlo;
 - monitoraggio dei progressi verso la risoluzione.
- **Capacità di gestire più informazioni contemporaneamente:** in questo gioco, infatti, bisogna capire qual è la strada migliore, come interagire con l'altro personaggio, con gli elementi di gioco al fine di superare il livello e come ottenere il maggior numero possibile di punti.
- **Pianificazione:** riguarda quel processo cognitivo che permette di eseguire comportamenti finalizzati ad un obiettivo seguendo tappe ben ordinate e precise. Questa funzione è una componente di base del nostro vivere quotidiano, viene impiegata in azioni relativamente semplici fino ad altre molto più complesse [21].
- **Collaborazione:** questa è probabilmente la componente educativa più importante di questo gioco. L'aspetto collaborativo è molto importante per i bambini perché non è una cosa semplice da imparare. Viene riportata una descrizione più dettagliata di questo aspetto nella sezione seguente.

5.8.1 Giochi collaborativi

Uno degli elementi più importanti che differenziano i giochi collaborativi da quelli competitivi e le gare è il piacere di giocare. Nei giochi cooperativi tutti si possono divertire e tutti possono partecipare perché in questo modo, cioè in uno stile di gioco basato sull'accettazione e non sull'esclusione e il giudizio, ciascuno può trovare un migliore equilibrio con il gruppo e il gruppo può trovare sempre nuovi obiettivi comuni da raggiungere. Nei giochi cooperativi nessuno vince, nessuno perde e nessuno viene escluso. I giocatori non si sfidano a vicenda, ma sfidano loro stessi, i limiti della loro creatività e fantasia per raggiungere uno scopo comune. Gli obiettivi che un gruppo riesce a raggiungere velocemente tramite il gioco cooperativo sono:

- creare di un senso comunitario nel gruppo, basato sull'accettazione di ognuno;
- favorire la conoscenza reciproca e l'affiatamento;
- creare un clima di fiducia e rispetto reciproco nel quale può crescere l'autostima di ognuno. Questo aspetto è molto importante. Capita spesso di vedere bambini particolarmente timidi, soprattutto con persone che non conoscono, che nel gioco cooperativo iniziano a superare e dimenticare, almeno nel momento del coinvolgimento nel gioco, la loro timidezza.

La cooperazione favorisce che il gruppo sviluppi al massimo le sue potenzialità grazie al contributo di ciascuno. Ciò è dovuto al fatto che non viene perduto il contributo, anche il più piccolo, dei "perdenti" (che spesso sono sempre gli stessi, finché perdono anche la voglia di partecipare al gioco). Il gioco cooperativo, inoltre, può rendere l'individuo più flessibile e anche più sereno nel rapporto con gli altri. Quando manca lo stress del doversi misurare, combattere e sottoporre ai giudizi degli altri, si è più rilassati e sereni.

La cooperazione favorisce la creatività del gruppo soprattutto grazie al clima di accettazione che permette appunto lo sviluppo dell'autostima. Spesso infatti la creatività viene uccisa sul nascere dalla paura del giudizio degli altri o dalla abitudine al giudizio su di sé. La creatività viene favorita dalla cooperazione anche perché permette l'incontro casuale di idee diverse [9].

Capitolo 6

Palloncini

6.1 Il gioco

Palloncini è un gioco single-player per bambini con disabilità motoria, i quali possono interagire con il PC utilizzando un dispositivo *single-switch* attraverso la tecnologia GNomon. È un multigioco educativo il cui scopo è quello di fare scoppiare dei palloncini presenti sullo schermo, secondo alcuni criteri derivanti da diverse modalità di gioco, cercando di ottenere il più alto punteggio possibile.

6.2 Descrizione

La scena iniziale del gioco si presenta come in Figura 6.1.

Questa scena, come molte altre del gioco, riprende il layout e le caratteristiche principali di quelle del gioco *Il labirinto dei mostri* descritto nel capitolo precedente. L'unica differenza in questa scena sta nel fatto che è stato omesso il pulsante “*Reset*”. Questo perché alla chiusura del gioco vengono soltanto salvate le impostazioni relative al giocatore e nulla riguardante i livelli e lo stato del gioco, inutile quindi un pulsante per cancellare le informazioni salvate. Premendo sul pulsante “*Inizia*”, si accede alla scena delle impostazioni (Figura 6.2). In questa scena, il giocatore può inserire il proprio nome, selezionare il widget GNomon, la velocità di rotazione (per l'orologio) o di riempimento (per la palla) e la velocità con cui far fluttuare i palloncini durante il gioco. A velocità zero i palloncini sono in una posizione fissa sullo schermo e non si muovono. Questa opzione è stata data per rendere accessibile il gioco anche a bambini con maggiori difficoltà a livello di concentrazione. Aumentando la velocità, invece, i palloncini si muovono più velocemente rendendo il gioco un po' più accattivante e meno banale per gli altri bambini. In questo modo si riesce a raggiungere un insieme più ampio di possibili giocatori.



Figura 6.1: Schermata iniziale del gioco *Palloncini*.



Figura 6.2: Pagina delle impostazioni del gioco *Palloncini*.

6.3 Organizzazione

6.3.1 Modalità di gioco

Questo gioco è organizzato come un multi-gioco. Ovvero, come si vede in Figura 6.3, si dà al giocatore la possibilità di selezionare una tra cinque diverse modalità. Per ogni modalità sono stati sviluppati cinque livelli. Progredendo nel gioco cresce il numero di palloncini presenti, il che rappresenta un limite al numero possibile dei livelli in quanto si è voluto evitare che troppi palloncini causassero troppa confusione facendo perdere giocabilità. La scelta delle diverse modalità di gioco è stata fatta perché, oltre al fatto che ogni modalità va a sviluppare una diversa capacità del bambino, in questo modo il gioco può essere utilizzato da bambini che hanno un livello cognitivo diverso. Infatti, se un bambino ha qualche deficit che gli impedisce di giocare ad una modalità, può comunque utilizzare le altre. Per esempio, un bambino può non conoscere i numeri o le lettere ma distinguere bene i colori, o viceversa.



Figura 6.3: Scena in cui si può scegliere la modalità di gioco.

Il gioco vero e proprio si svolge in una scena il cui background si presenta diviso in due sezioni, come si può vedere in Figura 6.4. La sezione in basso, azzurra, è lo spazio in cui si muovono i palloncini. Essi rimbalzano l'uno contro l'altro, contro i lati della schermata e contro la linea di separazione con la sezione superiore. La sezione in alto invece contiene il pulsante "Pausa" e una sezione con un quadrato ed un rettangolo. Nel quadrato viene mostrato un consiglio che indica qual è l'attuale modalità di gioco. Un po' più a destra, nel rettangolo, si trovano diverse sagome di palloncini. Queste rappresentano il numero dei palloncini che devono essere scoppiati. Ogni volta che un palloncino viene selezionato inizia a muoversi verso la prima sagoma libera. Se il palloncino

scelto è quello esatto allora esso va a sovrapporsi alla sagoma e ne prende il posto riproducendo un suono che emula l'esplosione del palloncino. Invece, se il palloncino selezionato non è quello corretto, esso si muove verso la sagoma ma, una volta arrivato, viene riprodotto un segnale acustico di errore e il palloncino viene fatto rimbalzare fino alla sua posizione iniziale. Nelle sezioni successive vengono descritte le cinque modalità di gioco.

Colori

In questa modalità il giocatore deve fare scoppiare soltanto i palloncini di un determinato colore. Questo colore viene scelto in maniera casuale all'inizio di ogni livello ed è indicato, con un apposito palloncino posto in alto, all'interno del quadrato, come si può vedere in Figura 6.4. Nel primo livello, il più semplice, sono presenti solamente palloncini del colore indicato, nei successivi invece ci sono anche palloncini di altri colori che non devono essere scoppiati. Nel caso quelli errati vengano selezionati, si ha un errore.

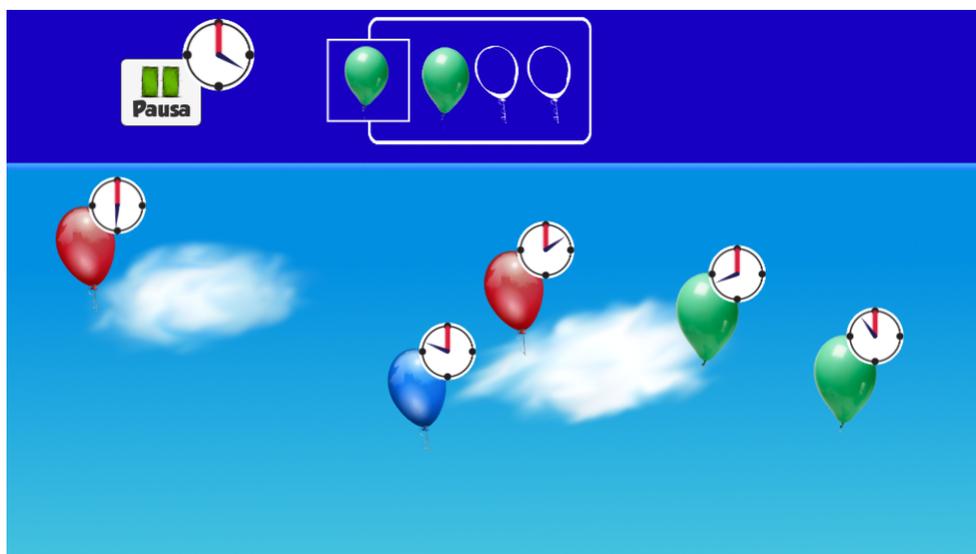


Figura 6.4: Modalità “Colori”.

Numeri

In questa modalità, i palloncini sono creati dello stesso colore perché la differenza di colore non crei ulteriori difficoltà nel gioco e all'interno di ognuno viene scritto un numero. Il compito del bambino è quello di farli scoppiare in ordine numerico crescente. Nel riquadro in alto sono rappresentati dei numeri indicanti questa modalità di gioco (Figura 6.5).



Figura 6.5: Modalità “Numeri”.

Sequenza colori

In questa modalità di gioco, i palloncini sono creati con colori casuali e il compito del bambino è quello di farli scoppiare secondo un determinato ordine. In alto, nel quadrato, viene mostrato qual è il colore del prossimo palloncino che deve essere fatto esplodere (Figura 6.6). Una volta fatto esplodere il palloncino, l'indicazione cambia indicando il colore del prossimo da selezionare.

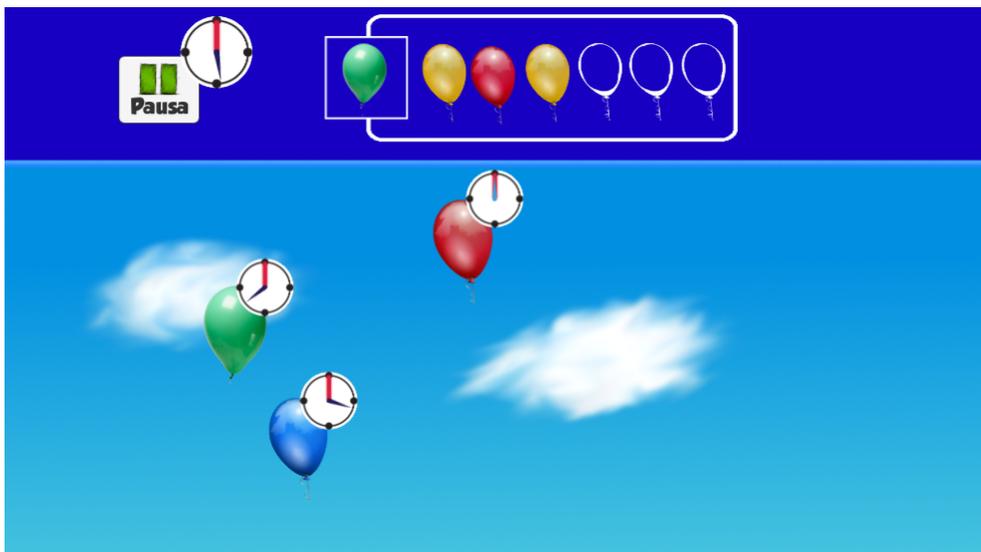


Figura 6.6: Modalità “Sequenza colori”.

Dimensioni

In questa modalità di gioco, i palloncini sono tutti dello stesso colore ma sono tutti di dimensioni diverse. Lo scopo del gioco è quello di farli esplodere in ordine di grandezza crescente o decrescente, come indicato in alto al centro nella schermata di gioco (Figura 6.7). Vengono infatti mostrate le sagome dei palloncini ridimensionate ed ordinate dalla più piccola alla più grande se l'ordine da seguire è crescente o viceversa se l'ordine è decrescente.

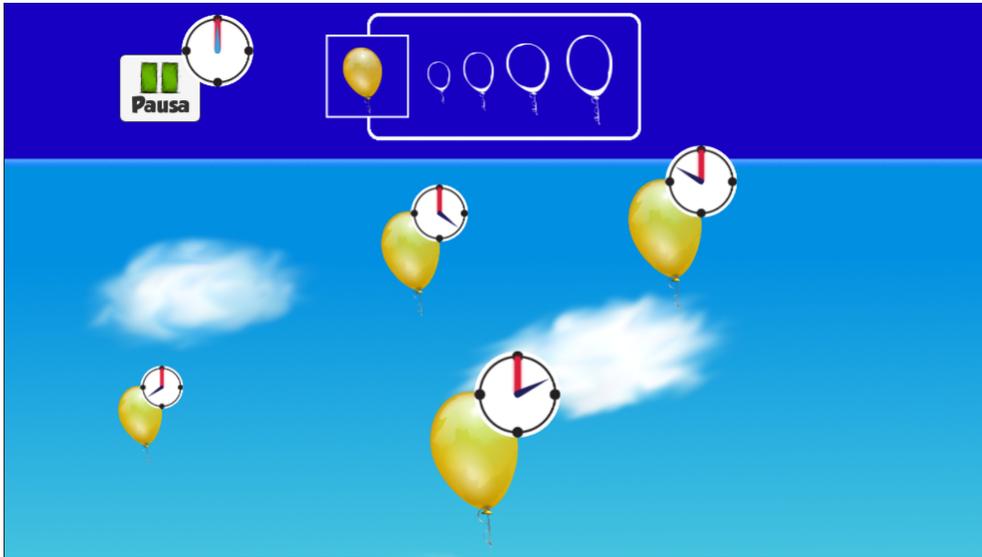


Figura 6.7: Modalità “Dimensioni”.

Parole

In questa modalità di gioco, i palloncini hanno tutti lo stesso colore e all'interno di ognuno viene mostrata una lettera. In alto nel quadrato viene mostrata un'immagine con a fianco tante sagome quante sono le lettere che compongono la parola indicata dall'immagine. Il compito del bambino è quello di fare scoppiare in ordine le lettere in modo da formare la parola raffigurata.

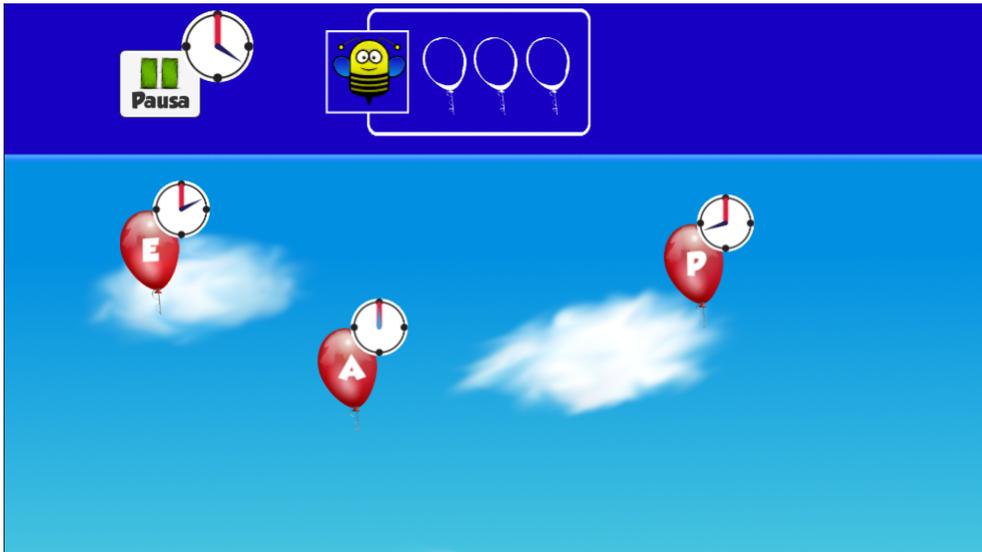


Figura 6.8: Modalità “Parole”.

6.3.2 Punteggio

Alla fine di ogni livello vengono mostrate tre stelle che indicano il punteggio ottenuto (Figura 6.9), similmente a come descritto per il gioco precedente. Il punteggio massimo, tre stelle gialle, si ottiene quando non si fa nessun errore. Al primo errore viene sottratta la terza stella (e si mostra una stella bianca al posto), a tre la seconda e, oltre ai cinque errori tutte le stelle non sono colorate. Per errore si intende ogni volta in cui viene selezionato un palloncino che non rispetta la sequenza richiesta. Non viene considerato come errore un *click* poco preciso che causa il secondo round di selezione di GNomon, facendo diventare rossi gli orologi.



Figura 6.9: Fine livello.

6.4 Componenti educative

Questo gioco presenta diverse componenti educative che si ipotizzano utili per bambini con età mentale che va da 3 ai 6 anni circa. Per capire alcune di queste caratteristiche è necessario definire alcuni diversi tipi di intelligenza:

- *intelligenza fluida*: corrisponde ai processi che risentono di influenze biologiche e permettono all'individuo di elaborare informazioni, solitamente di natura non verbale, senza l'utilizzo di contenuti influenzati dall'esperienza e dalla cultura [7];
- *intelligenza cristallizzata*: dipende in larga misura dalle abitudini acquisite ed è perciò soggetta all'influenza della scuola e delle altre condizioni culturali [12]. Quindi questo tipo di intelligenza dipende dalla vastità delle conoscenze che una persona possiede, dal suo vocabolario alla capacità di ragionare usando parole e numeri. È sostanzialmente il risultato di esperienze educative e culturali.

Questa distinzione tra diversi tipi di intelligenza è molto importante in psicologia soprattutto per fornire una misura non verbale dell'intelligenza, indipendente dal livello culturale. In particolare, tutte le modalità di gioco hanno componenti legati maggiormente ad aspetti logici e cognitivi eccetto la modalità *Parole* che va a sviluppare capacità verbali del bambino.

- *Colori e Sequenze colori*: queste due modalità vanno a migliorare la capacità di associazione logica tra oggetti. In entrambi i casi viene infatti data l'indicazione di quale deve essere il prossimo oggetto da selezionare e il bambino deve avere la capacità di individuare e scegliere quello corretto all'interno dell'insieme dei palloncini presenti.

- *Numeri*: in questo tipo di attività, contrariamente a come si può pensare, il bambino non va a sviluppare una capacità matematica (componente dell'intelligenza cristallina) ma una componente fluida che è la capacità di capire e rispettare un ordine sequenziale.
- *Dimensioni*: questa modalità va a sviluppare una capacità fluida: la capacità di seguire una progressione logica degli oggetti mostrati. La capacità di distinguere la differenza di dimensione tra diversi oggetti è infatti una capacità logica di base.
- *Parole*: questa modalità è l'unica che va a sviluppare una componente dell'intelligenza cristallina. In particolare va a sviluppare le capacità verbali del bambino. Quest'ultima modalità differisce dalle altre dal punto di vista educativo perché è l'unica che richiede un certo livello di scolarizzazione. Il bambino deve già avere una conoscenza di base per poter giocare con successo a questa variante del gioco. È possibile che per questo tipo di modalità l'età dei bambini che giocano debba essere più alta rispetto a quella richiesta dalle altre.

Capitolo 7

Implementazione

7.1 Strumenti di sviluppo

Per lo sviluppo di questi due videogiochi sono stati utilizzati i seguenti strumenti:

- **Unity 5 Personal Edition** è una piattaforma potente e flessibile per lo sviluppo di videogiochi ed esperienze interattive multiplatforma 3D e 2D. Questo strumento è stato utilizzato per creare e gestire la parte grafica del videogioco sviluppato. La scelta è stata dettata dal fatto che Unity è molto diffuso nell'ambito dello sviluppo di videogiochi e dal fatto che può esportare il lavoro su diverse piattaforme. Di seguito sono descritti brevemente alcuni elementi di Unity, per facilitare la comprensione di questo capitolo in cui verranno descritti i giochi sviluppati:
 - **GameObject**: è l'elemento fondamentale di Unity. Esso rappresenta un oggetto di gioco che può essere integrato con numerosi componenti che servono per implementare le effettive funzionalità e caratterizzazioni. Può essere usato per rappresentare personaggi, proprietà e scenari. Un *GameObject* può contenere altri *GameObject* instaurando una relazione padre-figlio. Ognuno di questi oggetti può contenere facoltativamente uno o più di questi componenti:
 - * **Collider**: componente il cui scopo è quello di definire la forma dell'oggetto con scopi relativi alle collisioni fisiche dell'elemento. Un *collider*, che è invisibile, non deve necessariamente seguire perfettamente la forma dell'oggetto in cui è inserito, ma un'approssimazione è spesso sufficiente. I più semplici sono tipi primitivi: *Box Collider*, *Sphere Collider*, *Capsule Collider* per il 3D e *Box Collider 2D*, *Circle Collider 2D* per il 2D. Questi elementi si possono utilizzare anche contemporaneamente per creare forme complesse. Nei giochi sviluppati sono utilizzati solo componenti 2D.

- * **RigidBody**: componente che abilita il *gameObject* a comportarsi secondo le leggi della fisica. Un *rigidBody* può ricevere una forza e fare muovere l'oggetto che lo contiene in modo realistico. Ogni oggetto deve contenere un *rigidBody* per essere influenzato dalla gravità e muoversi sotto effetto di forze imposte via script o per interagire con altri oggetti attraverso il motore fisico NVIDIA PhysX.
 - * **Animator Controller**: strumento che permette di arrangiare ed organizzare un insieme di animazioni relative ad un personaggio o un oggetto. In molte situazioni, è normale avere più animazioni e passare tra esse quando accade un certo evento nel gioco. È rappresentato come una macchina a stati che indica quale animazione è attualmente in fase di riproduzione, quali sono gli altri stati possibili e quando si transita da uno stato ad un altro.
 - * **Scripts**: componente che può essere associato ad ogni oggetto del gioco. Unity consente di realizzare script in C# o in Javascript. È un ingrediente essenziale in ogni gioco. Serve per rispondere a determinati input da parte dell'utente, reagire a determinati eventi di gioco, per creare effetti grafici, controllare il comportamento fisico degli elementi e per implementare una sorta di intelligenza artificiale dei personaggi del gioco. Una cosa molto utile in Unity è il fatto che si possono dichiarare delle variabili pubbliche le quali possono essere viste direttamente all'interno dell'editor di gioco. In questo modo si possono facilmente assegnare *GameObject* a variabili degli script. In generale il funzionamento di Unity è legato a tre funzioni:
 - **void Start()**: funzione eseguita una sola volta, all'inizializzazione dello script.
 - **void Awake()**: ha lo stesso scopo di *Start()*, ma viene eseguita prima di ogni funzione *Start()*, utile per dare un ordine all'inizializzazione degli script.
 - **void Update()**: funzione che viene eseguita ad ogni frame, ovvero ad ogni istante. Utile per controllare gli elementi di gioco e per ricevere input dagli utenti.
 - * **SpriteRenderer**: componente che permette di visualizzare immagini come *Sprite* sia per uso 2D che 3D. Gli *Sprite* sono oggetti grafici 2D usati per rappresentare i personaggi ed altri elementi del gioco. Questi elementi sono ottenuti tramite immagini bitmap.
 - * **Transform**: in Unity, è la componente che determina la posizione, la rotazione e la scala di un *GameObject*. Ogni *GameObject* ha un componente di questo tipo.
 - * **RectTransform**: mentre la *Transform* rappresenta un singolo punto, la *RectTransform* rappresenta un rettangolo in cui può essere contenuto l'elemento di gioco.
- **Scene**: schermate in cui viene suddiviso il gioco. Una scena contiene tutti i *GameObject* che devono essere utilizzati in un particolare istante del gioco. Durante l'esecuzione questi oggetti possono essere rimossi, creati e modificati dinamicamente. Le scene possono essere utilizzate per creare menu, singoli livelli, o qualunque altra cosa. Ogni scena

può essere riempita con elementi di gioco, personaggi, scenari, pulsanti, testi, Una scena comprende un insieme di *GameObject* e ognuno di questi è formato da diversi componenti o altri *GameObject*, come mostrato in Figura 7.1.

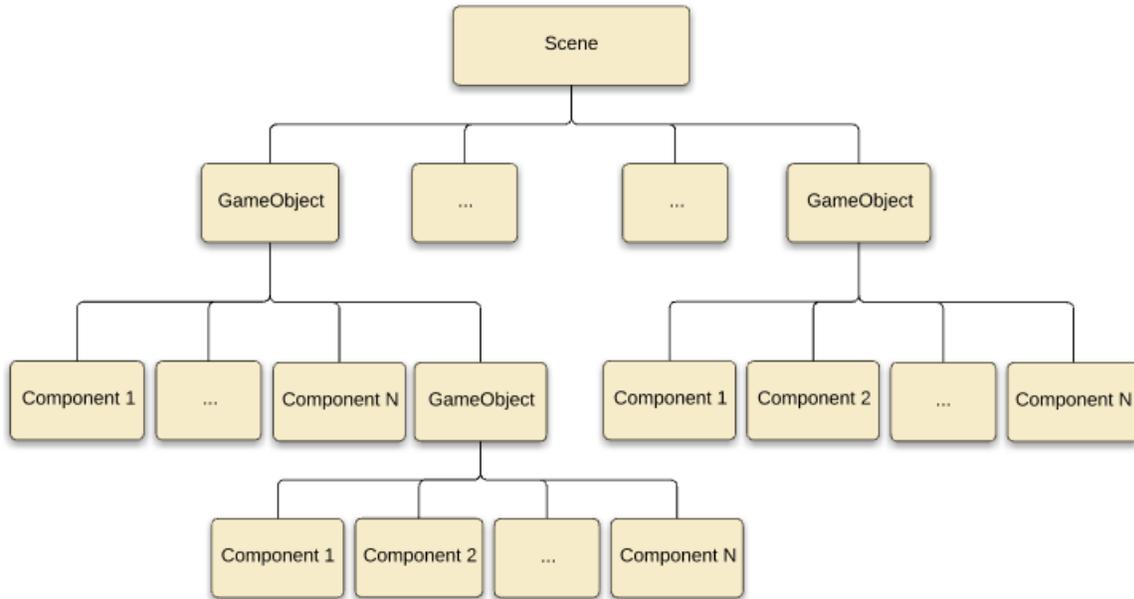


Figura 7.1: Organizzazione interna di una scena in Unity.

- **Tag:** marcatore che può essere utilizzato per identificare oggetti all'interno del progetto.
 - **Layer:** utilizzati per creare gruppi di oggetti che condividono delle caratteristiche particolari. I primi otto layer sono utilizzati da Unity di default e non sono modificabili, mentre dall'ottavo al trentunesimo possono essere personalizzabili.
 - **SortingLayer:** utilizzati all'interno di componenti di tipo *SpriteRenderer*. Servono per definire un ordine di sovrapposizione delle diverse sprite del gioco.
- **Visual Studio 2012 Professional** è un ambiente di sviluppo integrato (*Integrated development environment* o IDE) sviluppato da Microsoft. Attualmente supporta numerosi linguaggi di programmazione ed è multipiattaforma, ovvero consente la realizzazione di programmi per server, workstation, PC, smartphone e per i browser. In questo lavoro è stato sfruttato per la creazione e la modifica di script *C#* necessari per gestire il gioco creato con Unity.
 - **Gimp 2.8 - GNU Image Manipulation Program** è uno strumento di editing di immagini multi-piattaforma, disponibile per sistemi operativi GNU/Linux, OS X, Windows ed altri ancora. È un software libero il cui codice sorgente può essere liberamente modificato e ridistribuito. GIMP fornisce numerosi strumenti per la manipolazione di immagini

ad alta qualità e consente di aumentare la produttività consentendo molte personalizzazioni riguardanti le opzioni e l'utilizzo plug-in di terze parti.

7.2 Organizzazione

In questa sezione vengono descritte alcune caratteristiche implementative comuni ad entrambi i giochi sviluppati. Nelle sezioni successive, invece, quelle specifiche a ciascun gioco.

7.2.1 Assets

In Unity, i file utilizzati sono organizzati nella sezione *Project* presente nell'editor. Questa sezione contiene gli *Assets*, ovvero tutti i componenti utilizzati all'interno del gioco. L'organizzazione di questi componenti è molto simile per entrambi i giochi sviluppati e si può osservare in Figura 7.2.

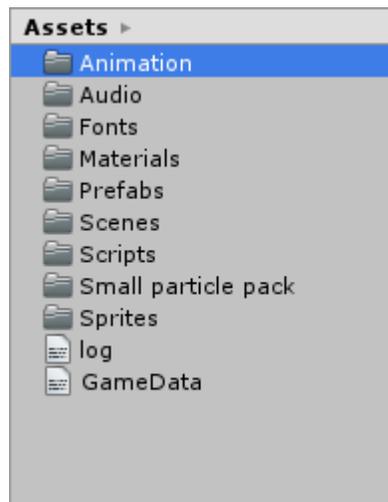


Figura 7.2: Assets.

La cartella degli assets contiene file e cartelle:

- *Animation*: cartella in cui sono contenute le animazioni dei vari elementi di gioco.
- *Audio*: cartella in cui sono contenuti tutti i file audio che vengono riprodotti durante l'esecuzione del gioco.
- *Font*: cartella in cui sono contenuti i tipi di carattere di testo utilizzati.
- *Materials*: cartella in cui sono contenuti i materiali. Questi elementi sono utilizzati nei Collider per definire il comportamento che deve avere l'oggetto quando viene rilevata una collisione. In questa cartella è contenuto un tipo di *Material*:

- *BouncyMaterial*: un oggetto con collider di questo materiale rimbalza contro l'oggetto con in quale entra in collisione. Questo è il materiale associato ai collider dei palloncini nel gioco *Palloncini*

Questa cartella non è presente nel gioco *Il labirinto dei mostri* perché vengono utilizzati materiali di default di Unity.

- *Prefabs*: o prefabbricati. Sono dei GameObject che vengono salvati nel progetto (e non solo nella scena) per poi essere riutilizzati in diverse situazioni. Di fatto i prefab sono utilizzati ogni volta in cui un oggetto deve comparire più di una volta con caratteristiche simili o uguali. Ogni volta che un prefab viene inserito in una scena si dice che quella è un'istanza del prefab stesso. L'utilità di questi elementi è data dal fatto che, modificando il prefab, si possono apportare le modifiche direttamente a tutte le istanze, senza doverle modificare una per una.
- *Scenes*: cartella contenente le scene del gioco.
- *Scripts*: questa cartella contiene tutti gli script utilizzati durante il gioco. All'interno si trova anche la cartella *Gnomon* in cui sono salvati gli script che definiscono il comportamento del framework che fornisce i meccanismi di accesso per persone con disabilità motorie.
- *Small particle pack*: questa cartella è stata importata per creare l'animazione rappresentante la pioggia di stelle. Questa animazione è stata scaricata dall'Asset Store di Unity.
- *Sprites*: cartella contenente gli sprite utilizzati.
- *log*: il file di log è un file in cui viene registrata ogni azione compiuta durante il gioco e l'esatto istante in cui essa è stata intrapresa.
- *GameData*: file in cui vengono salvate le impostazioni e i dati di gioco alla chiusura dell'applicazione e che verranno reimpostati durante i successivi avvii. Questo file viene eliminato quando nel gioco *Il labirinto dei mostri* viene premuto il pulsante "Reset". In entrambi i giochi viene scritto quando si preme il pulsante "Esci" o si preme il pulsante "ESC" sulla tastiera.

Scripts

In entrambi i giochi vengono riutilizzati gli stessi script (oltre a quelli di GNomon), alcuni leggermente modificati e adattati al singolo gioco ma fondamentalmente con lo stesso scopo e le stesse caratteristiche. Questi script sono:

- *AspectUtility.cs*: script che permette di impostare un rapporto tra le dimensioni relative all'aspetto del gioco. In entrambi i giochi il rapporto è di 16:9. Per questo motivo, all'avvio del gioco, tra le impostazioni relative alla scelta della risoluzione è possibile soltanto selezionare risoluzioni che rispettino questo rapporto.

- **AudioOnOff.cs**: script che gestisce la musica principale. Questo script è associato al pulsante che consente di attivare o disattivare la musica principale del gioco.
- **ButtonManager.cs**: script che gestisce il comportamento dei principali pulsanti utilizzati in entrambi i giochi sviluppati. Per ognuno di essi è presente un metodo che viene eseguito ogni volta in cui vengono premuti. Per quelli specifici dei giochi sono utilizzati altri script che saranno descritti in seguito.
- **GameState.cs**: questo script contiene una classe singleton e che quindi viene istanziata una volta soltanto durante il gioco. Questa contiene le variabili principali dei giochi ed è usata per condividere variabili tra oggetti presenti in scene diverse.
- **GameParams.cs**: script che serve a raccogliere tutti i dati (nome giocatore, personaggio, velocità di rotazione, ...) inseriti dall'utente, a processarli e a salvarli.
- **GnomonController.cs**: script che ha il compito di controllare la creazione, la gestione, il comportamento e la distruzione dei widget GNomon presenti nelle scene.

Scene

In entrambi i giochi sono utilizzate alcune scene che, seppur con qualche diversità a livello di contenuti, sono fondamentalmente le stesse dal punto di vista funzionale. Le relazioni tra le scene saranno mostrate nelle sezioni relative ai singoli giochi. Esse sono:

- **Welcome**: scena iniziale del gioco. Contiene il titolo del gioco, il pulsante per iniziare a giocare, per attivare o disattivare l'audio, per uscire dal gioco, per accedere alla scena *Credits* e, nel gioco *Il labirinto dei mostri*, per resettare il gioco.
- **Settings**: scena che permette al giocatore di personalizzare alcune impostazioni di gioco come il nome dei personaggi, il widget GNomon e la sua velocità di rotazione o di riempimento e nel gioco *Palloncini* la velocità di movimento dei palloncini.
- **Results**: scena mostrata alla fine di ogni livello nella quale vengono mostrate le stelle ottenute durante lo svolgimento del livello completato.
- **Credits**: scena in cui sono riportate le attribuzioni degli elementi utilizzati protetti da copyright.
- **EndGame**: scena mostrata al completamento del gioco o, nel caso del gioco *Palloncini*, al completamento di una modalità di gioco. Riproduce un'animazione raffigurante una pioggia di stelle (animazione fornita dallo **SmallParticlePack**) e una clip audio trionfale.

Nell'editor di Unity è presente la sezione *Hierarchy* in cui sono presenti tutti gli oggetti e le istanze dei prefab che formano l'attuale scena di gioco, come si può vedere in Figura 7.3. In questa immagine è rappresentata la gerarchia della scena *Welcome* del gioco *Il labirinto dei mostri*

(Figura 5.1) ma la struttura è simile per tutte le scene.

Gli elementi scritti in blu sono istanze di prefab in quanto sono utilizzati più volte all'interno del

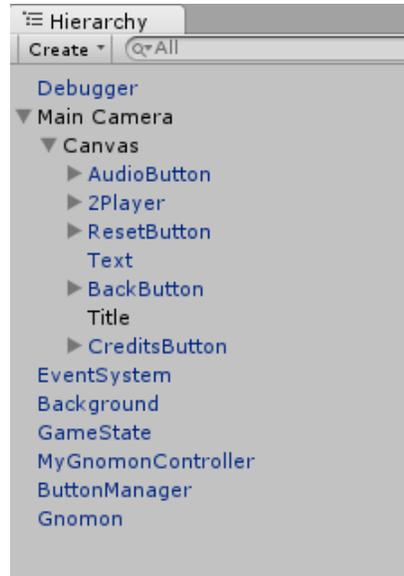


Figura 7.3: Hierarchy.

gioco o in entrambi i giochi. Quelli scritti in nero sono semplici `GameObject`. Una scena è così strutturata:

- *Debugger*: oggetto presente soltanto nella scena iniziale di entrambi i giochi. Contiene lo script `Dbg.cs` che inizializza e consente l'utilizzo del logger. Questo script crea il file di log nel percorso specificato e fornisce funzioni per scrivere su questo file.
- *Main Camera*: la camera è l'elemento che cattura e mostra al giocatore lo spazio di gioco. Essendo entrambi i giochi in due dimensioni viene utilizzata soltanto una camera che resta fissa in ogni scena. All'interno di questo elemento è necessario utilizzare un canvas per poter visualizzare gli elementi con i quali è possibili interagire con il giocatore (buttons, inputFields, ...). In questo esempio troviamo alcuni pulsanti, il testo e l'immagine contenente il titolo del gioco. Tra i componenti di questa camera è contenuto lo script `AspectUtility.cs`. Nelle scene in cui si svolge il gioco vero e proprio, all'interno della camera sono presenti due canvas. Il canvas che contiene la scena di gioco e un altro, inizialmente disabilitato, che contiene gli oggetti che vengono mostrati quando viene premuto il pulsante "Pausa" (Figura 7.4).

Alla pressione di questo pulsante, il primo canvas viene disabilitato e viene abilitato il secondo, così da modificare quello che viene mostrato all'utente. Se il giocatore decide di continuare



Figura 7.4: Camera con due canvas.

il livello, viene disabilitato il secondo canvas e riabilitato il primo, che riporta lo stesso stato del gioco dell'istante in cui si era interrotto.

- *EventSystem*: componente che ha il compito di inviare gli input dei giocatori (mouse, tastiera, ...) agli oggetti di gioco (pulsanti, aree di testo, ...).
- *Background*: oggetto che rappresenta lo sfondo del gioco.
- *GameState*: oggetto contenuto soltanto nelle scene iniziali di entrambi i giochi. Esso contiene lo script `GameState.cs`.
- *GnomonController*: oggetto che contiene l'omonimo script.
- *ButtonManager*: oggetto che contiene lo script `ButtonManager.cs`.
- *Gnomon*: oggetto che contiene l'omonimo script. Necessario in ogni scena per utilizzare GNomon.

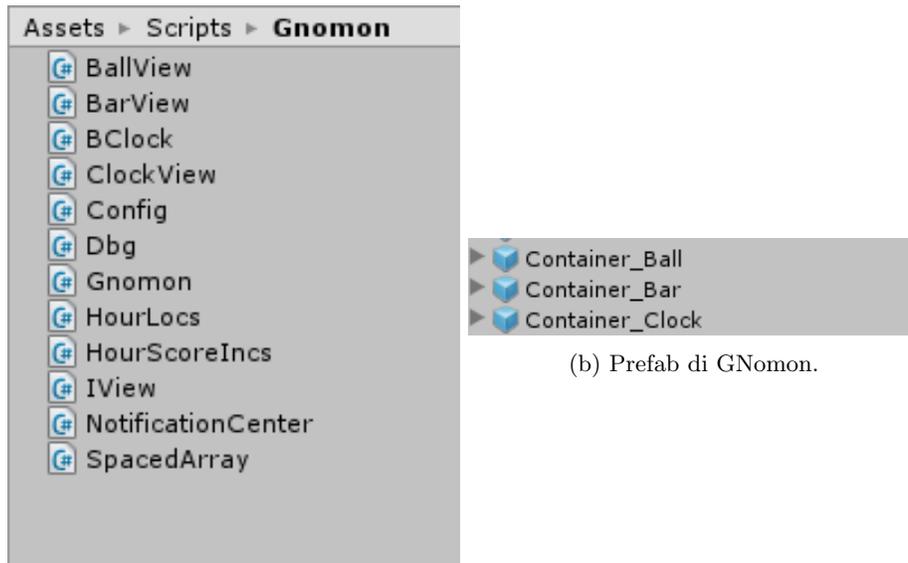
In entrambi i giochi, nella scena *Settings* è presente l'oggetto *GameParams* contenente lo script per gestire le informazioni ricevute dall'utente.

7.3 GNomon

GNomon si presenta come un insieme di script, di prefab e di sprite che si possono vedere in (Figura 7.5).

Tra gli script troviamo `Gnomon.cs` che è lo script principale, quello con cui bisogna interfacciarsi per utilizzare questo framework. Gli altri script sono utilizzati per calcolare le probabilità, la posizione delle lancette e per gestire le diverse viste (palla: `BallView.cs`, triangolo: `BarView.cs` e orologio: `ClockView.cs`). I tre prefab rappresentano i widget GNomon, costruiti con le sprite che si possono vedere nella figura precedente.

La struttura interna dei prefab invece la si può vedere in Figura 7.6. Questa figura rappresenta la struttura del prefab dell'orologio GNomon, ma quella della palla e il triangolo è la stessa, cambiano solamente le sprite utilizzate. L'oggetto principale è `Container_Clock` che raggruppa i vari elementi del widget. All'interno di questo contenitore si ha un `Associated_Object` che è l'oggetto a cui il widget viene associato e il `Selector`. Questo oggetto è lo sfondo dell'orologio, ovvero l'immagine circolare su cui gira la lancetta che è l'`Indicator`.



(a) Script di GNomon.



(c) Sprite di GNomon

Figura 7.5: Elementi di GNomon.

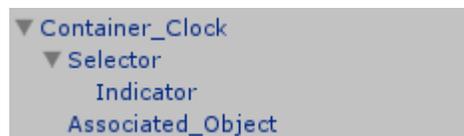


Figura 7.6: Container_Clock.

7.3.1 Modifiche a GNomon

Per adattare e far funzionare al meglio GNomon con i giochi da me sviluppati sono state apportate alcune piccole modifiche allo script `Gnomon.cs`, nel metodo `Awake()`.

Una prima modifica è stata fatta per prendere, all'inizio del gioco, la velocità impostata inizialmente nello script `GameState.cs`, come si può osservare nel listato 7.5.

```
if (GameState.gameState != null)
{
```

```
rotation_speed = GameState.rotationSpeed;  
}
```

Listing 7.1: Modifica codice `Gnomon.cs`

Questa modifica è utile in quanto all'inizio del gioco vengono caricati i dati salvati durante l'ultima chiusura. Tramite queste righe di codice è possibile impostare in GNomon la stessa velocità di rotazione o di riempimento che era stata selezionata dall'utente durante il suo ultimo accesso. Quindi, se esiste l'istanza dell'oggetto *GameState* allora viene recuperato il valore della rotazione. In caso contrario si utilizza quello di default di GNomon.

In entrambi i giochi non viene fornita l'opportunità di utilizzare come widget di GNomon la barra triangolare perché risulta essere quella più difficile da utilizzare per i bambini [2].

7.3.2 Interazione con GNomon

In ogni scena, per poter interagire con GNomon è necessaria avere diversi *GameObject*. Un *GameObject* chiamato GNomon che contiene come componente lo script `Gnomon.cs` e un altro che contiene lo script `GnomonController.cs`. All'interno di questo script è presente una variabile pubblica `Gnomon` di tipo `Gnomon`. Come spiegato nella descrizione degli script di Unity, alle variabili pubbliche si può associare un valore o l'istanza di un *GameObject* direttamente all'interno dell'editor grafico. A questa variabile quindi viene passata l'istanza dell'oggetto GNomon. Grazie a questa struttura, all'interno dello script `GnomonController.cs` si può interagire direttamente con GNomon utilizzando le funzioni da esso fornite. Allo stesso modo, al `GameController` vengono passate altre istanze di *GameObject*, come per esempio quella del `ButtonManager` (Figura 7.7).

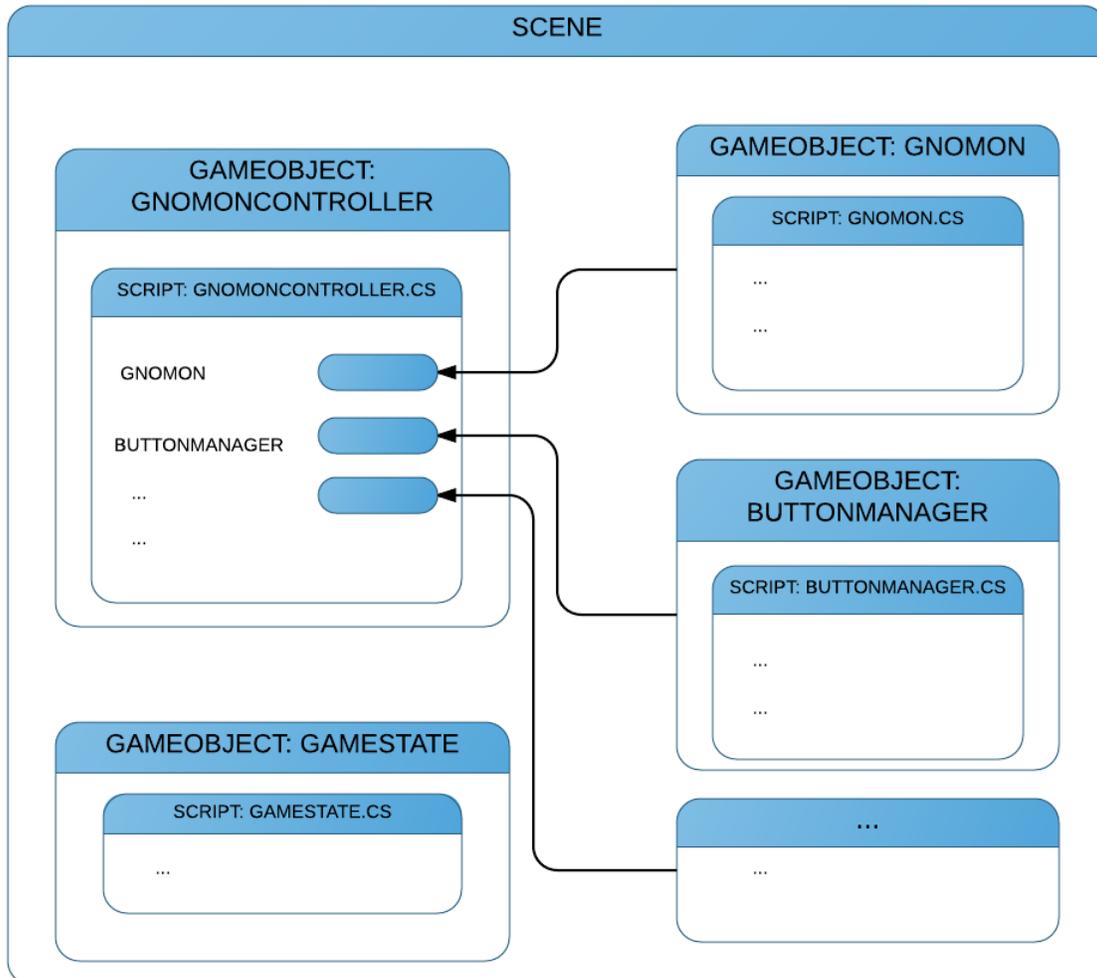


Figura 7.7: Interazione con GNomon.

Le funzioni utilizzate per interagire con GNomon sono le seguenti:

- `public GameObject add_bclock(GameObject instance)`: questa funzione permette di creare un `BClock`, oggetto di GNomon, a partire dall'istanza passata come parametro (per esempio il `container_clock` debitamente inizializzato).
- `public int select(float time_in)`: funzione a cui bisogna passare come parametro il tempo attuale (`Time.time`) e che ritorna un intero che rappresenta l'indice del widget vincitore del round GNomon.

- `public void remove_clock(int clock_index)`: funzione che rimuove da GNomon l'istanza del widget identificato dall'indice passato come parametro.
- `public void pause()`: funzione che mette GNomon in pausa, fermando la rotazione o il riempimento dei widget.
- `public void end_pause()`: funzione fa uscire GNomon dalla pausa e riattiva i widget.

7.4 *Il labirinto dei mostri*

7.4.1 Scene

Nella Figura 7.8 sono mostrate le scene utilizzate in questo gioco.

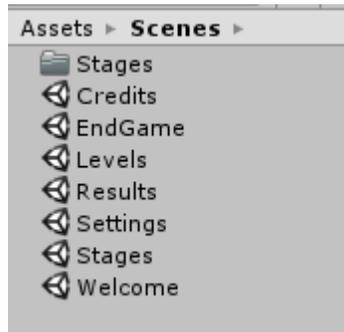


Figura 7.8: Assets.

Nella Figura 7.9 invece viene mostrato l'albero delle cartelle in cui sono contenuti gli stage e i livelli.

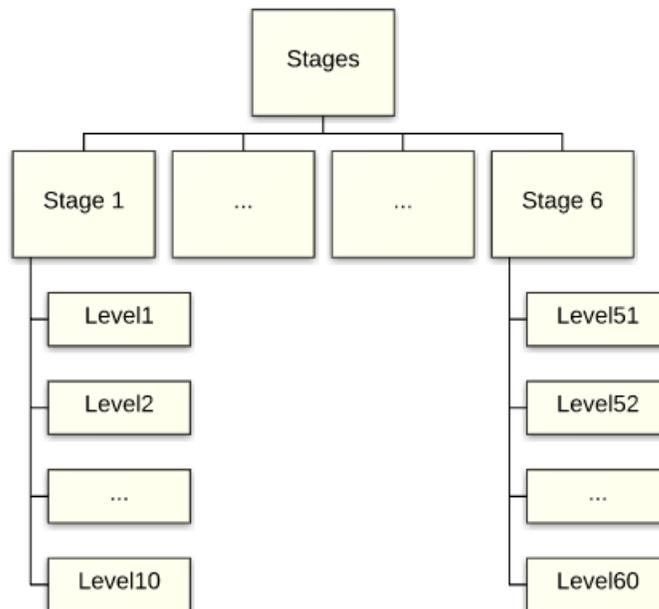


Figura 7.9: Organizzazione stage e livelli.

Oltre alle scene comuni ad entrambi i giochi abbiamo:

- **Cartella Stages:** questa cartella contiene sei cartelle rappresentanti i diversi stage del gioco. Ognuna di queste sottocartelle contiene le scene di 10 livelli, per un totale di 60 livelli, come raffigurato nella figura precedente.
- **Stages:** scena nella quale bisogna selezionare lo stage desiderato.
- **Levels:** scena che contiene i dieci pulsanti con i quali è possibile selezionare il livello desiderato.

Nella Figura 7.10 vengono mostrate le scene (nei riquadri) con le azioni (ovvero i pulsanti premuti, rappresentati dalle frecce) che causano il passaggio da una scena all'altra.

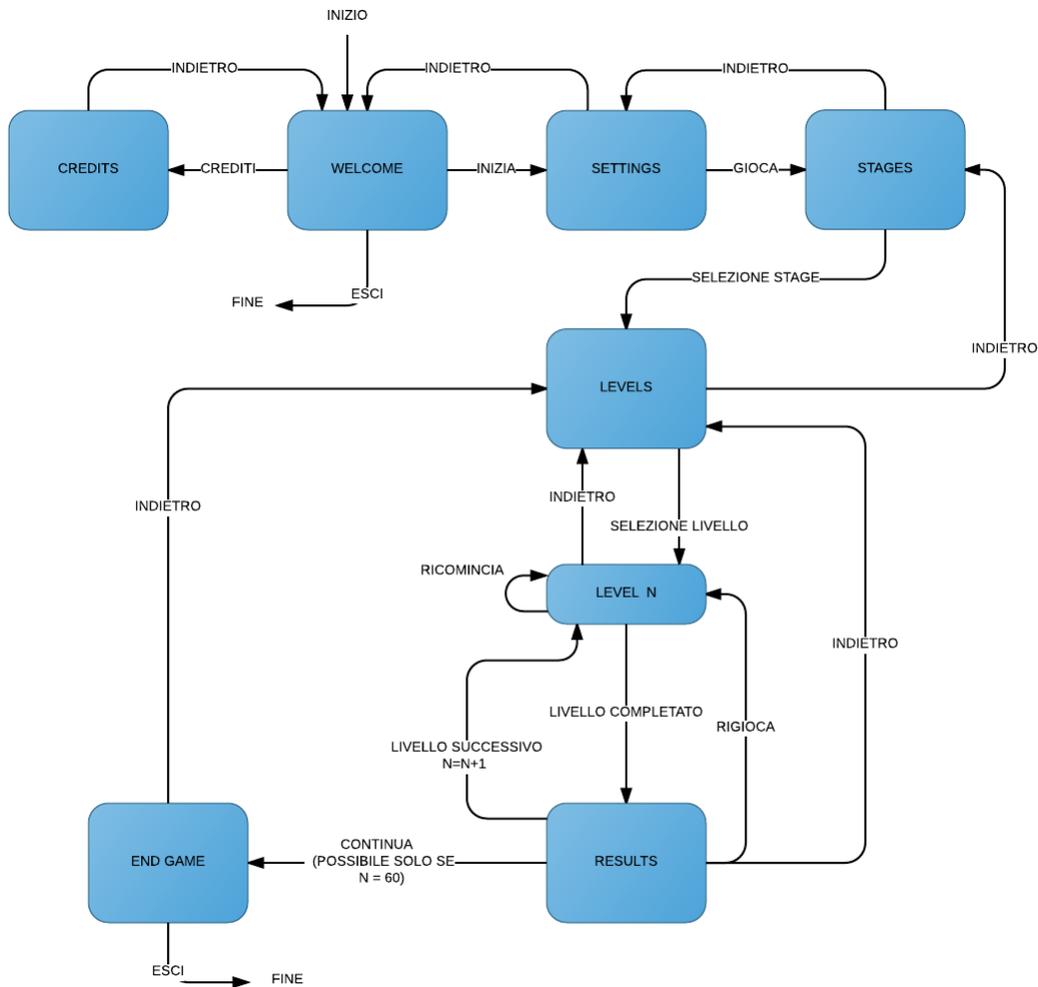
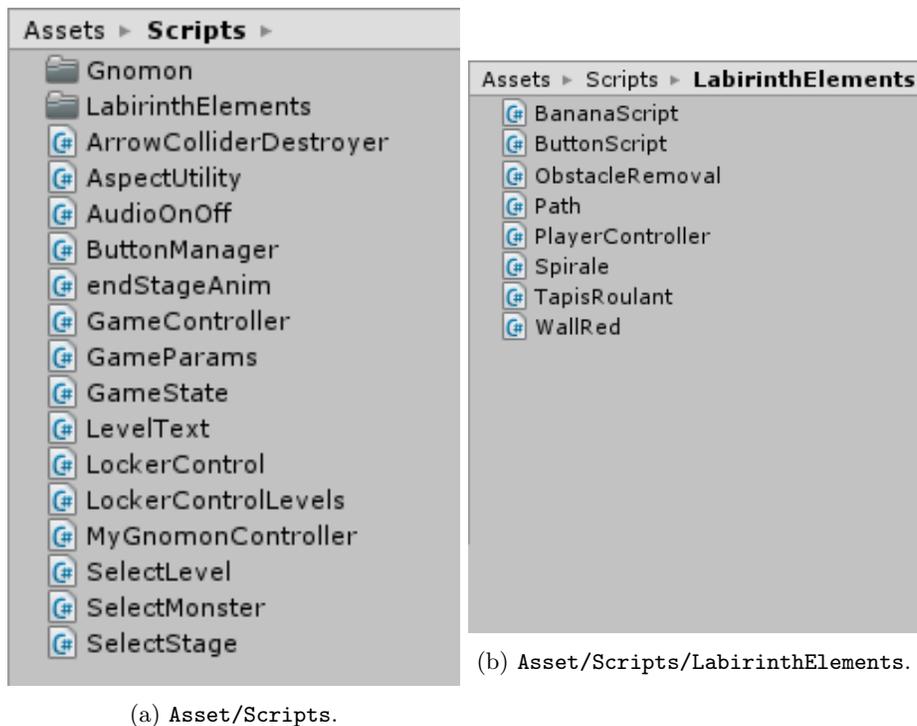


Figura 7.10: Diagramma delle scene del gioco *Il labirinto dei mostri*.

Si inizia dalla scena *Welcome*. A seconda dello stage selezionato si possono selezionare 10 diversi livelli. Ogni livello è implementato in una scena a parte. Alla scena *EndGame* si può accedere solamente quando viene completato l'ultimo livello.

7.4.2 Script

Nella Figura 7.11 sono mostrate gli script utilizzate in questo gioco.



(a) Asset/Scripts.

(b) Asset/Scripts/LabirinthElements.

Figura 7.11: Script del gioco *Il labirinto dei mostri*.

Oltre agli script comuni ad entrambi i giochi abbiamo:

- **Cartella LabirinthElements:** questa cartella contiene gli script che servono per gestire gli elementi di gioco che si trovano all'interno del labirinto, come i personaggi, i pulsanti e gli elementi aggiuntivi di gioco.
 - **BananaScript.cs:** questo script, associato ad ogni banana, ha il compito di rilevare l'avvenuta collisione tra il personaggio e la banana stessa. A collisione avvenuta riproduce un feedback acustico, aumenta i punti di gioco, scrive una riga sul file di log e fa partire l'animazione collegata a questo tipo di elemento.
 - **ButtonScript.cs:** script associato ai pulsanti sui quali si devono posizionare i personaggi per completare il livello. Tramite il rilevamento di collisioni questo script riesce

a capire se il personaggio corretto si trova in posizione finale e, nel caso lo sia, imposta il valore della variabile booleana `Player1OnPosition` o `Player2OnPosition` a `true`. Quando entrambe queste variabili sono vere si fa partire un timer. Se i personaggi rimangono entrambi sul proprio pulsante fino allo scadere del timer il livello viene superato. Il timer serve per evitare che il livello venga superato quando un personaggio passa sopra al proprio pulsante senza effettivamente fermarsi sopra.

- `ObstacleRemoval.cs`: script associato al pulsante su cui i giocatori devono posizionarsi per poter distruggere l'elemento di gioco raffigurato dal muro con i mattoni verdi. Questo script riceve in input il `GameObject` del muro e quando rileva la collisione tra il pulsante e un personaggio lo distrugge facendo partire un'animazione ed un feedback acustico.
 - `Path.cs`: script per gestire il comportamento delle caselle sulle quali il passaggio è consentito solamente una volta. Rileva la collisione col personaggio, riproduce un feedback acustico e fa partire un'animazione che sostituisce la casella con un muro.
 - `PlayerController.cs`: script utilizzato per gestire i personaggi. Rileva la pressione delle frecce della tastiera per muovere il personaggio utilizzato dal bambino non disabile. Per il movimento dell'altro personaggio la gestione di `GNomon` è fatta nello script `GnomonController.cs` ma il movimento viene effettuato chiamando il metodo `movePlayer(...)` di questo script.
 - `Spriale.cs`: rileva la collisione col personaggio, ne cambia la posizione e riproduce un feedback acustico.
 - `TapisRoulant.cs`: rileva la collisione, riproduce un segnale audio e cambia la direzione del movimento del personaggio in base all'orientamento di questo elemento.
 - `WallRed.cs`: rileva la collisione del giocatore contro il muro con i mattoni rossi che viene distrutto riproducendo un'animazione e un feedback acustico.
- `ArrowColliderDestroyer.cs`: script che serve a gestire le frecce utilizzate dal giocatore che utilizza `GNomon` per muoversi. Il codice di questo file ha il compito di visualizzare le frecce e i widget relativi alle direzioni che il personaggio può intraprendere in un determinato istante e di nascondere le altre. Questa gestione viene effettuata utilizzando dei `CircleCollider2D` all'interno delle frecce. Quando viene rilevata una collisione o quando il personaggio è in movimento la freccia viene disabilitata, quindi resa invisibile. Quando il personaggio si ferma, se la freccia non è in collisione, ritorna ad essere visibile.

La gestione di `GNomon` invece è leggermente più complessa. Uno dei vantaggi di `GNomon` è quello di dare la possibilità di poter aggiungere o rimuovere i widget durante l'esecuzione del gioco. L'idea è quella per cui ogni volta che si rimuove una freccia si deve rimuovere anche il relativo widget `GNomon` (con la funzione `remove_clock(...)` di fornita dallo script `Gnomon.cs`). Seguendo questo procedimento però nasce un problema che riduce la giocabilità

facendo aumentare la possibilità di errori. Il problema in questione è causato dal fatto che GNomon ricalcola la posizione delle lancette degli orologi ogni volta in cui uno di essi viene rimosso o aggiunto. Quindi, siccome anche quando si ha una nuova collisione o l'uscita da una collisione con l'altro personaggio si ha un nuovo calcolo delle fasi di GNomon, è possibile che queste vengano modificate nell'istante prima del quale il giocatore disabile sta per premere il pulsante per far muovere il suo personaggio. Se vengono cambiate e il pulsante premuto immediatamente dopo, la lancetta non si trova più al punto previsto dal giocatore ma può avere un'angolazione diversa portando a movimenti errati. Per risolvere questo problema, questo script rileva quando la collisione di una freccia avviene con l'altro personaggio. In questo caso anche il widget, come la freccia, viene disabilitato e quindi reso invisibile senza effettivamente rimuoverlo da GNomon. È quindi possibile che a causa di click poco precisi ci sia una disputa (widget rossi) tra un elemento visibile ed uno non visibile. Questo problema è però molto più raro e meno fastidioso del precedente.

- `endStageAnim.cs`: questo script rileva quando uno stage viene completato e riproduce un'animazione.
- `GameController.cs`: script utilizzato per verificare se entrambi i personaggi sono in posizione finale e, in caso positivo, concludere il livello gestendo i punti e procedendo con la scena di gioco successiva.
- `LevelText.cs`: script che modifica il testo dei pulsanti contenuti nella scena *Levels* a seconda dello stage selezionato. In pratica se lo stage selezionato è il primo, il numero dei livelli andrà da 1 a 10, se lo stage selezionato è il secondo, andranno da 11 a 20 e così via.
- `LockerControl.cs` e `LockerControleLevels.cs`: script che visualizzano l'icona del lucchetto sui pulsanti che servono per selezionare livelli o stage che per il momento sono ancora bloccati e dunque non accessibili.
- `SelectLevel.cs`, `SelectLevel.cs`: script utilizzati per la gestione dei pulsanti nelle scene in cui bisogna selezionare lo stage ed il livello a cui si vuole giocare.
- `SelectMonster.cs`: script che gestisce il corretto funzionamento della selezione del mostro nella scena *Settings*. Questo script mostra i personaggi quando viene premuto il pulsante con la freccia verso il basso, permette la scelta del mostro e impedisce che entrambi i giocatori selezionino lo stesso.

7.4.3 Personaggi

Come già mostrato in Figura 5.2 i personaggi utilizzabili sono sei mostri. Questi personaggi sono degli oggetti di gioco costituiti da diversi componenti tra cui un *RectTransform*, un *RigidBody*, un *CircleCollider2D* che ne approssima la forma per rilevare le collisioni e lo script `PlayerController.cs`

per controllarne il comportamento. Questi elementi sono mostrati in Figura 7.12.

I personaggi sono caratterizzati dal tag *Player* (tag esistente di default in Unity) e appartengono al layer *BlockingLayer*. Lo *SpriteRenderer* ha come *SortingLayer* il valore *Units* con ordine 3 per mostrare i mostri sempre al di sopra degli altri elementi mostrati all'interno del labirinto.

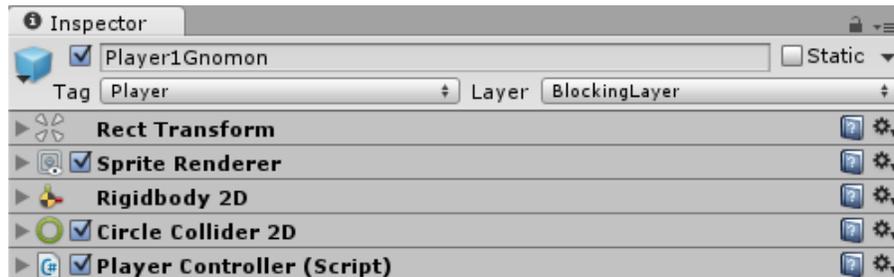


Figura 7.12: Componenti dei personaggi.

Movimento

Il movimento dei personaggi avviene modificando la proprietà *velocity* del componente *RigidBody* specificando la direzione e la velocità. Questo viene fatto nel metodo `movePlayer(Direction direction)` nello script `PlayerController.cs` associato ad entrambi i personaggi. A questo metodo viene passato un parametro indicante la direzione del movimento da compiere. Nel (listato 7.2) viene mostrato questo metodo.

```
public void movePlayer(Direction direction)
{
    moving = true;
    //this because is not possible to detect collision between a
    //collider of an object with kinematic
    //rigidbody (the player) and a collider of a gameObject
    //without rigidbody (borders)
    this.gameObject.GetComponent<Rigidbody2D>().isKinematic = false;

    lastMove = direction;
    dir = direction;
    transform.GetComponent<Rigidbody2D>().freezeRotation = true;

    if (direction == Direction.right)
    {
        transform.GetComponent<Rigidbody2D>().velocity =
            Vector2.right * velocity;
    }
    else if (direction == Direction.left)
```

```
{
    transform.GetComponent<Rigidbody2D>().velocity =
        Vector2.left * velocity;
}
else if (direction == Direction.up)
{
    transform.GetComponent<Rigidbody2D>().velocity =
        Vector2.up * velocity;
}
else if (direction == Direction.down)
{
    transform.GetComponent<Rigidbody2D>().velocity =
        Vector2.down * velocity;
}
}
```

Listing 7.2: Movimento del personaggio

Quando il personaggio è in movimento si setta a `true` il valore della variabile `moving` dello script `PlayerController.cs`. Questa variabile è utile per sapere se mostrare le frecce del personaggio che utilizza `GNomon` (quando è fermo) oppure per nasconderele se è in movimento. Lo stesso script rileva le collisioni dei personaggi contro i bordi del labirinto o contro l'altro personaggio e li ferma quando necessario azzerando la velocità del *RigidBody*. In questi casi, per non fare fermare il personaggio attaccato all'altro o al bordo, si fa muovere leggermente in direzione contraria all'ultimo movimento (listato 7.3) così da evitare che resti in collisione con altri elementi.

```
if (this.moving)
{
    int offset = 6;
    this.transform.GetComponent<Rigidbody2D>().velocity = new
        Vector2(0, 0);
    if (dir == Direction.right)
        transform.Translate(-offset, 0, 0);
    else if (dir == Direction.left)
        transform.Translate(offset, 0, 0);
    else if (dir == Direction.up)
        transform.Translate(0, -offset, 0);
    else if (dir == Direction.down)
        transform.Translate(0, offset, 0);
}
```

Listing 7.3: Movimento all'indietro dopo una collisione

Si controlla se il personaggio a cui è collegato lo script è effettivamente quello che si sta muovendo. In caso affermativo, si riduce la velocità a zero e, una volta individuata la direzione del movimento (variabile `dir`), si modifica leggermente la posizione. Successivamente la variabile `moving` verrà messa a `false`.

In questo script viene anche salvata l'informazione riguardante l'ultimo movimento effettuato dai giocatori nella variabile `lastMove`. Questa informazione è utile perché quando viene premuto il pulsante per mettere in pausa il gioco viene disabilitato il canvas principale contenente il labirinto, i personaggi, ...causando quindi la fine del movimento dei mostri. Quando il gioco viene ripreso, grazie alla variabile `lastMove`, se il giocatore ha la variabile `moving` vera allora viene ripetuta l'ultima mossa consentendo al personaggio di completare il movimento che aveva iniziato prima che il gioco venisse interrotto.

7.5 *Palloncini*

7.5.1 Scene

Nella Figura 7.13 sono mostrate le scene utilizzate in questo gioco.



Figura 7.13: Scene del gioco *Palloncini*.

L'unica scena diversa da quelle in comune con il primo gioco sviluppato è la *GameMode*. In questa scena il giocatore può selezionare la modalità di gioco a cui vuole accedere. Nella Figura 7.14 vengono mostrate le scene (nei riquadri) con le azioni (ovvero i pulsanti premuti, rappresentati dalle frecce) che causano il passaggio da una scena all'altra.

A differenza del gioco precedente tutti i livelli sono implementati nella stessa scena di gioco perché i palloncini e gli oggetti vengono tutti gestiti e creati dinamicamente tramite script. Anche in questo gioco si inizia dalla scena *Welcome*.

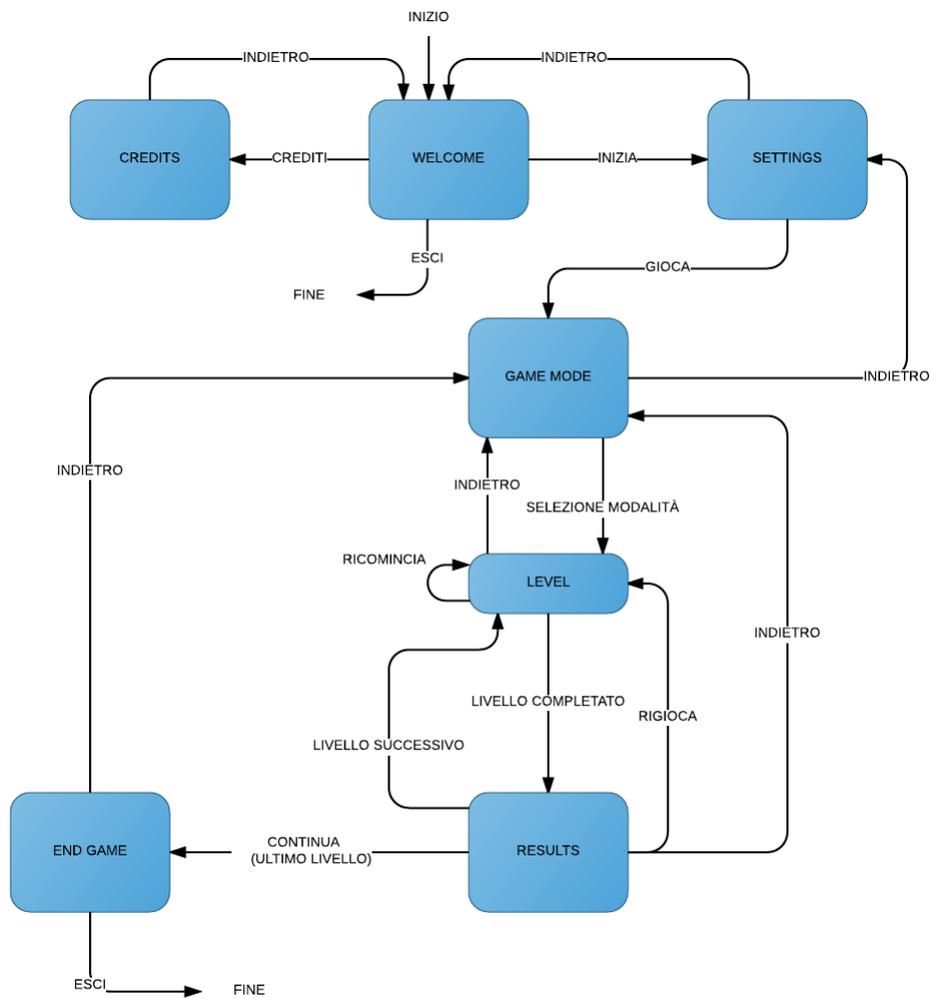


Figura 7.14: Diagramma delle scene del gioco *Il labirinto dei mostri*.

7.5.2 Script

Nella Figura 7.15 sono mostrati gli script utilizzati per lo sviluppo del gioco *Palloncini*. Si nota che ci sono soltanto due script che non sono presenti anche nell'altro gioco:

- `BalloonController.cs`: questo è uno script che viene associato come componente ad ogni palloncino presente nel gioco. Nel metodo `Start()` viene solamente chiamato il metodo pubblico `makeFloat()` che serve per imprimere una forza ai palloncini e farli muovere nello spazio di gioco applicando una velocità che varia in base all'impostazione modificabile dall'utente.

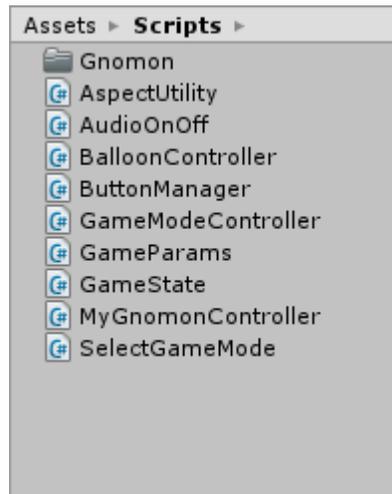


Figura 7.15: Script del gioco *Palloncini*.

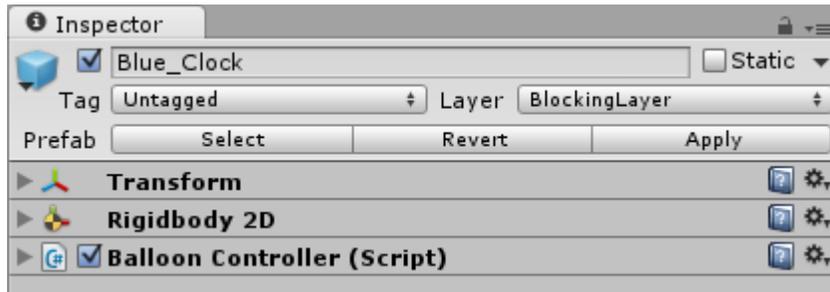
- **GameModeController.cs**: questo script è utilizzato per gestire le diverse modalità di gioco: si occupa di preparare la scena a seconda della modalità scelta gestendo sagome, colori e sequenze. Inoltre gestisce GNomon per quanto riguarda la selezione dei palloncini e si occupa del loro comportamento una volta selezionati.
- **GameModeSelect.cs**: script che permette la selezione della modalità di gioco. Definisce il comportamento dei pulsanti presenti nella scena *GameMode*.

7.5.3 Palloncini

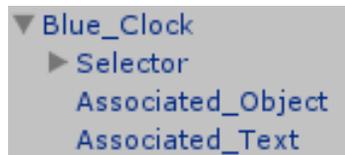
I palloncini sono degli oggetti di gioco che sono strutturati come mostrato nella figura sottostante (Figura 7.16).

In Figura 7.16a, si possono vedere i componenti che caratterizzano l'oggetto. Si ha un *Transform*, un *RigidBody2D* per emulare un comportamento fisico reale e uno script (**BalloonController.cs**) che controlla il movimento del palloncino. Il *RigidBody2D* ha il campo **GravityScale** uguale a zero. Questo valore elimina la forza di gravità presente nella scena di gioco, evitando di attirare verso il basso i palloncini.

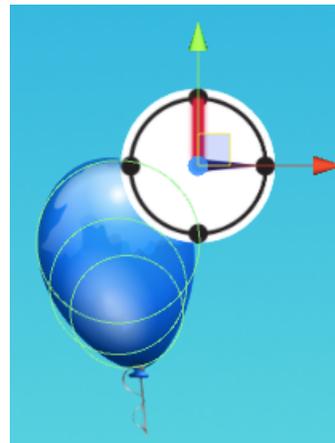
In Figura 7.16b è mostrata la struttura dell'oggetto contenente il palloncino e l'orologio di GNomon e in Figura 7.16c la rappresentazione. Il *GameObject* **Blue.Clock** contiene un elemento di nome *Selector* che è l'orologio GNomon al cui interno ha l'*Indicator* che è la lancetta. Inoltre ha **Associated.Object**, ovvero il palloncino vero e proprio, e un **Associated.Text**, per mostrare un numero o una lettera nelle modalità di gioco che lo richiedono. Comi si può vedere in Figura 7.16d, il palloncino ha tre componenti *CircleCollider2D* (sono i tre cerchi verdi visibili all'interno del palloncino blu). Vengono usati tre di questi elementi in modo da poter approssimare al meglio la sagoma del palloncino, per avere una superficie di collisione più realistica.



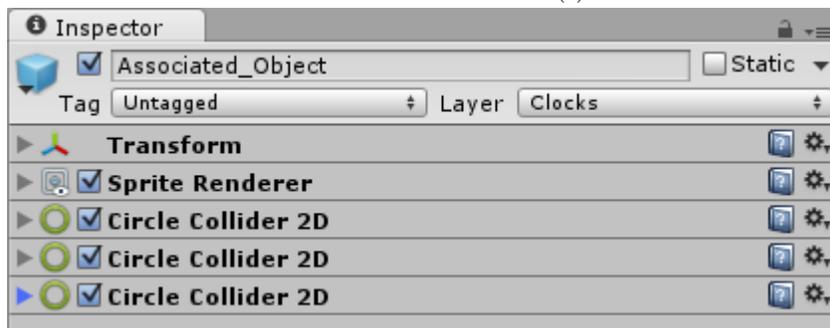
(a) Componenti del palloncino.



(b) Struttura palloncino



(c) Palloncino blu.



(d) Associated.Object.

Figura 7.16: Script del gioco *Il labirinto dei mostri*.

Movimento

Alla creazione di ogni palloncino il movimento inizia col il metodo `makeFloat()` della classe `BalloonController.cs`. (listato 7.4).

```
public void makeFloat()
{
    float speedX = (float)Random.Range(-1500, 1500); //direction on
        X axis
    float speedY = 0; //direction on Y axis
    float val = GameState.balloonSpeed; //speed value set by the
        player

    //For balloons in other scenes (not the game scene)
    if (GameState.actualScene != "Level")
        value = 2;

    if (val == 0)
        speedY = 0;
    else if (val == 1)
        speedY = (float)Random.Range(100, 500);
    else if (val == 2)
        speedY = (float)Random.Range(1000, 1500);
    else if (val == 3)
        speedY = (float)Random.Range(1800, 2600);
    else if (val == 4)
        speedY = (float)Random.Range(2800, 3500);
    else if (val == 5)
        speedY = (float)Random.Range(3800, 5000);

    if (speedY != 0)
        transform.GetComponent<Rigidbody2D>().AddForce(new
            Vector2(speedX, speedY)); //move

    transform.GetComponent<Rigidbody2D>().freezeRotation = true;
        //avoid rotation
}
```

Listing 7.4: `makeFloat()` nello script `Gnomon.cs`

In base alla velocità impostata dall'utente (settata nella variabile `GameState.balloonSpeed`) vengono impostati dei valori entro i quali viene selezionato casualmente un valore che rappresenta la dimensione della forza applicata all'asse Y del palloncino. Questa forza è sempre positiva per far

muovere i palloncini verso l'alto. La forza relativa all'asse X invece è un numero causale che può avere anche valori negativi, per far muovere il palloncino in entrambe le direzioni, destra e sinistra. Se la velocità selezionata dall'utente è uguale a zero il palloncino non viene fatto muovere.

Ogni volta in cui viene selezionato un palloncino, esso si muove verso il riquadro in alto contenente le sagome. Per effettuare questo movimento è stata utilizzata la funzione `Vector3.Lerp(Vector3 from, Vector3 to, float t)`. Questo metodo è fornito dalla classe `Vector3` della libreria `UnityEngine` e implementa l'interpolazione lineare tra due vettori. Questa funzione viene utilizzata per spostare un oggetto dalla posizione di origine (primo parametro) fino alla posizione di destinazione (secondo parametro). Il terzo parametro, `t`, varia nel range `[0,1]`.

Quando `t` vale zero la funzione ritorna il punto di origine, se vale 1 ritorna il punto finale e per tutti i valori intermedi ritorna dei valori che si trovano sul segmento che unisce l'origine con la destinazione. Grazie a questi valori si può far muovere un oggetto lungo il segmento che unisce due punti.

Quando un palloncino viene selezionato si imposta come primo parametro la sua posizione attuale e come parametro di destinazione la posizione della sagoma a cui dovrebbe andare a sovrapporsi. Il tempo di volo impostato è sempre di due secondi. Si salva in una variabile il tempo di inizio (`startTime`) e poi si può far iniziare il movimento (impostando la variabile `lerping` uguale a `true`).

```
//prepare stuff for Lerp function
startTime = Time.time;
journeyLength = 2f;
origin = go.transform.position; //the destination is set before
objToLerp = go;
lerping = true; //start to lerp
```

Listing 7.5: Modifica codice `Gnomon.cs`

La funzione `Vector3.Lerp()` deve essere chiamata all'interno della funzione `Update()` che viene chiamata per ogni frame finché la variabile `lerping` è vera. Ogni volta, prima di chiamare questa funzione, si calcola la percentuale di completamento del movimento (`fracJourney`), come mostrato nel (listato 7.6). Si prende il tempo attuale a cui si sottrae il tempo di inizio e si divide il risultato per la durata totale del viaggio (impostata precedentemente a 2 secondi). A questo punto viene chiamata la funzione di lerp e si sposta il palloncino (`objToLerp`).

```
void Update()
{
    if (lerping) //if movement is activated
    {
        distCovered = Time.time - startTime;
        fracJourney = distCovered / journeyLength;
```

```

objToLerp.transform.position = Vector3.Lerp(origin, dest,
    fracJourney); //change position of the balloons
if (fracJourney >= 1.0f) //if travel is completed
{
    //movement completed
    ...
    //do stuff
    ...
    lerping = false; //stop movement
}
}

```

Listing 7.6: Modifica codice `Gnomon.cs`

Quando un palloncino viene selezionato correttamente inizia quindi a muoversi verso la sagoma e una volta arrivato a destinazione sostituisce la sagoma emulando il suono di un palloncino che scoppia. Se invece il palloncino selezionato non è corretto, si muove fino alla sagoma, viene riprodotto un segnale acustico di errore e vengono nuovamente effettuati tutti i passaggi per la preparazione e l'utilizzo della funzione `Vector3.Lerp()` invertendo i valori di origine e destinazione, facendo quindi ritornare il palloncino nella sua posizione originale.

Ogni volta in cui viene selezionato un palloncino, `GNomon` viene messo in pausa per consentire che i movimenti descritti in precedenza vengano effettuati senza creare confusione. Quindi per un breve intervallo di tempo le lancette degli orologi smettono di ruotare. Alla fine del movimento del palloncino selezionato, `GNomon` viene fatto uscire dalla pausa e si può riprendere il gioco.

7.5.4 Modalità di gioco

In base alla modalità di gioco scelta, la scena viene allestita mostrando il riquadro in alto, il numero ed il colore dei palloncini, le sagome a cui essi devono andare a sovrapposti ed eventualmente numeri e lettere al loro interno.

Tutte queste operazioni vengono effettuate al caricamento della scena *Level*, all'interno della funzione `Init()` dello script `GameModeController.cs`.

La gestione della selezione del palloncino attraverso `GNomon` invece viene effettuata nello script `GnomonController.cs` che gestisce ogni modalità attraverso un'apposita funzione. Queste funzioni sono `manageColors()`, `manageWords()`, `manageSequences()`, `manageDimensions()` e `manageNumbers()`.

A queste funzioni viene passato un parametro che identifica il vincitore risultante dalla selezione tramite `GNomon`. Tramite questo indice si risale al *GameObject* contenente il palloncino ed il widget `GNomon` selezionato. In seguito viene effettuato un controllo per verificare se il palloncino selezionato rispetta la sequenza data dalla modalità di gioco scelta. In caso positivo il palloncino viene fatto muovere verso l'alto, viene rimosso da `GNomon` e viene aggiunto un punto al giocatore. In caso di errore invece, il palloncino viene fatto muovere verso le sagome in altro, ma poi tornerà

indietro, non viene rimosso da GNomon e viene sottratto un punto al giocatore.

In entrambi i casi viene anche scritta sul file di log una riga per indicare l'avvenuta selezione e il suo esito.

Dopo ogni selezione viene controllato se ci sono altri palloncini in gioco. Se non ce ne sono viene terminato il livello dopo 3 secondi di attesa, in modo da poter dare ancora la possibilità di poter osservare la sequenza completata in alto.

Capitolo 8

Valutazione

8.1 Il pulsante utilizzato

Il pulsante *single-switch* sfruttato per permettere ai bambini con disabilità motorie di utilizzare i giochi sviluppati è il “*BigRed Switch*”¹ (Figura 8.1) della ablenet.inc. Si tratta di un pulsante



Figura 8.1: BigRed Switch.

avente una superficie di attivazione circolare di 12,5 cm di diametro. Questo pulsante è di tipo meccanico. Per attivarlo, infatti, è necessario imprimere una leggera pressione (156 g) sulla superficie colorata. Ad ogni pressione viene riprodotto un feedback acustico e tattile.

La superficie colorata può essere sostituita con superfici di diversi colori: rossa, blu, gialla o verde. Questo pulsante viene utilizzato collegandolo attraverso un cavo mono e un jack da 3,5 mm che va inserito in un'interfaccia che ne consente il collegamento con il PC.

¹<https://www.ablenetinc.com/catalog/product/view/id/123/category/12/>, ultima visita 20/02/2016

L'interfaccia utilizzata per i test effettuati è la “*HelpiBox 16*”² (Figura 8.2) della Helpicare. Si tratta di un'interfaccia molto semplice che non necessita di alcun driver per essere installata.



Figura 8.2: Helpibox 16.

Viene riconosciuta automaticamente dal sistema operativo (Windows 98 o superiori) e si connette al PC tramite una porta USB.

Questo dispositivo riceve in ingresso il BigRed Switch la cui pressione, a seconda dell'ingresso a cui è collegato, emula sul PC la pressione di un tasto della tastiera o del mouse.

Per i test effettuati è stato necessario emulare la pressione della barra spaziatrice (ingresso SPACE dell'interfaccia).

8.2 Test

I test sono stati effettuati sul videogioco *Il labirinto dei mostri*. Al fine di poter valutare al meglio il gioco è stata fornita una versione in cui fin dall'inizio tutti gli stage e tutti i livelli sono sbloccati. Si può quindi accedere ad un livello o stage senza dover necessariamente completare i precedenti. Questo perché, in caso qualche livello fosse risultato troppo difficile o fosse sorto qualche bug, si sarebbe potuto procedere comunque senza compromettere l'esito dell'intera prova.

8.2.1 Partecipanti

Le prove sono state svolte facendo giocare a coppie bambini con età compresa tra i 5 e i 12 anni. Solamente in due casi, a causa della mancanza di un compagno di gioco, uno dei due partecipanti è stato un adulto (famigliare del bambino).

Le coppie sono state 5, per un totale di 8 bambini e 2 adulti. Per motivi di rispetto della privacy non verranno riportati i nomi dei partecipanti delle prove ma ogni riferimento ad essi verrà fatto

²<http://www.helpicare.com/helpibox-16/>, ultima visita 20/02/2016

utilizzando codici alfanumerici. Le coppie saranno invece rappresentate da un numero e sono descritte nella tabella seguente:

Coppia	Giocatore 1	Età	Giocatore 2	Età
1	P1	adulto	P2	10
2	P3	adulto	P4	6
3	P5	12	P6	9
4	P7	10	P8	10
5	P9	6	P10	5

I partecipanti coinvolti nelle prove sono tutti bambini senza disabilità. Per questo motivo, per non dover fornire a tutti il dispositivo *single-switch*, uno dei due partecipanti delle coppie con entrambi i bambini normodotati ha utilizzato la barra spaziatrice della tastiera del computer al posto del pulsante esterno.

Lo scopo principale di questi test è quello di analizzare il comportamento dei bambini durante il gioco, osservando come interagiscono, comunicano e collaborano tra di loro.

Il secondo obiettivo è quello di capire con precisione quale sia l'età mentale adatta che i bambini devono avere per poter sfruttare al meglio le potenzialità di questo videogioco.

Queste prove sono anche state utili per portare alla luce alcuni bug che sono stati corretti.

8.2.2 Metodi e procedure

Le prove sono state effettuate a casa di uno dei due giocatori.

Ai bambini è stata data l'opportunità di giocare liberamente, uno utilizzando GNomon e l'altro la tastiera. Ai partecipanti è stata data la possibilità di scambiare la modalità di controllo durante il gioco.

Il gioco prevede 60 livelli e, a causa della sua longevità, non tutte le coppie sono riuscite a completarli tutti, un po' per le difficoltà incontrate e un po' per il limitato tempo a disposizione.

La durata delle prove è stata variabile ed è compresa tra un'ora e 3 ore e 20 minuti circa (non consecutive).

Coppia 1

Questa coppia è composta da un bambino di 10 anni e un adulto di 22. La durata di questa prova è stata di un'ora e 20 minuti e questa coppia è l'unica ad essere riuscita a completare tutti e 60 i

livelli. Sicuramente il supporto fornito dall'adulto ha accorciato notevolmente i tempi per superare i livelli più complessi. GNomon è stato utilizzato solamente dall'adulto.

Coppia 2

Questa coppia, formata da una bambina di 6 anni e suo padre è stata quella che ha incontrato le maggiori difficoltà. Infatti la bambina non è stata in grado di giocare autonomamente ma soltanto seguendo attentamente le indicazioni fornite dal padre.

Coppia 3

Coppia formata da un bambino di 12 anni e da uno di 9. La prova è durata circa un'ora e venti minuti ed hanno giocato fino al livello numero 48 compreso. GNomon è stato utilizzato dal bambino più grande.

Coppia 4

Coppia formata da due bambini, entrambi di 10 anni di età. La prova ha avuto una durata di circa un'ora e durante il gioco entrambi hanno provato ad utilizzare il dispositivo *single-switch*.

Coppia 5

Questa coppia è composta da una bambina di 6 anni e da un bambino di 5. GNomon è stato utilizzato dalla bambina più grande. È stato anche fatto provare al bambino più piccolo ma con scarsi risultati.

8.3 Risultati

Durante le prove questo gioco è stato definito come *carino* e *interessante*. Tutti i bambini sono stati soddisfatti e alcuni hanno spontaneamente detto di essersi divertiti.

L'interazione tra i giocatori per quanto riguarda l'aspetto collaborativo e di socializzazione è stata molto soddisfacente.

Per la prima coppia questo aspetto è stato molto positivo sicuramente anche grazie alla presenza di un adulto (motivo per il quale sono state riscontrate poche difficoltà nel superamento di molti livelli trovati difficili da altre coppie).

Per la seconda coppia invece non è stato possibile avere dei risultati in quanto la bambina di 6 anni non aveva ancora le capacità di giocare, se non seguendo solamente le indicazioni del padre.

Questo fatto può dipendere da diversi fattori, come l'esperienza ed il rapporto personale con i videogiochi: se un bambino non ha mai utilizzato nessuno di questi strumenti è normale che incontri più difficoltà rispetto ad un altro, magari anche con età leggermente inferiore, ma che invece ha già una certa confidenza.

Nelle coppie in cui entrambi i componenti erano bambini, a livello cooperativo è emerso che normalmente il primo bambino che trovava la soluzione, di norma il più grande, forniva istruzioni all'altro per completare il livello. In altri casi i bambini si sono scambiati informazioni regolative, del tipo “*Aspetta, ora mi muovo io.*”, oppure “*Spostati in quella direzione.*”.

In tutti i casi, comunque, la collaborazione ha funzionato molto bene, soprattutto nelle coppie in cui i due bambini si conoscevano molto bene essendo abituati a giocare insieme.

Durante il gioco sono state incontrate alcune difficoltà nel superare alcuni livelli, infatti non è stato possibile per quasi tutte le coppie completarli tutti e 60, ad eccezione della prima. Queste difficoltà sono state più invalidanti per la coppia 5, quella con i due bambini con età minore. In questo caso è stato necessario l'intervento di un adulto per aiutarli in alcuni passaggi più complessi e sono stati fatti saltare i livelli più difficili a partire dal terzo stage.

Inoltre, durante le prove, alcuni bambini si sono lamentati perché chi utilizzava il pulsante *single-switch* era più lento rispetto all'altro giocatore e gli errori di selezione più probabili.

8.4 Conclusioni

Dalle osservazioni descritte nella sezione precedente è emerso che a livello di collaborazione il gioco si è dimostrato efficace e molto utile. In tutti i casi osservati, la collaborazione tra i due giocatori è avvenuta. La modalità con cui essa avviene varia in base a diversi aspetti:

- *Età giocatori*: se i giocatori hanno un'età diversa è molto probabile che sia il più grande a condurre il gioco. Normalmente, infatti, è il primo a trovare la soluzione del livello e nel dare indicazioni all'altro giocatore su come comportarsi.
- *Carattere e confidenza tra i giocatori*: lo stile di interazione varia molto in base al carattere dei bambini e in base a quanto essi si conoscono. È più probabile che due bambini che si conoscono bene riescano a comunicare e collaborare meglio, ma anche, a seconda del carattere di ognuno, ad avere situazioni di attrito come nel caso dei giocatori P7 e P8 che sono arrivati fino al litigio.

Dalle prove effettuate si è giunti alla conclusione che a livello collaborativo il gioco è adatto a bambini con età che va dai 5 o 6 anni in avanti.

Le richieste cognitive del gioco che hanno causato le maggiori difficoltà sono state la capacità di prevedere le mosse, di immaginare i vari movimenti e le conseguenze che queste mosse comportano. Inoltre viene richiesta una buona dose di memoria di lavoro: i bambini devono ricordarsi delle mosse già fatte in precedenza e tenere conto dei risultati ottenuti. Per questi motivi, siccome i bambini più piccoli hanno incontrato maggiori difficoltà in molti livelli, a livello cognitivo l'età per poter sfruttare la massima potenzialità di questo gioco deve essere maggiore. Per completare con successo tutti i livelli si è stimato che almeno uno dei due giocatori debba avere l'età maggiore o uguale ai 10 anni.

In caso contrario il gioco fornisce comunque buoni risultati a livello collaborativo e consente efficacemente ai bambini di socializzare, interagire e collaborare, ma per alcuni livelli sarà necessario l'intervento di un aiuto esterno.

Capitolo 9

Conclusioni

Il mondo dei videogiochi è costantemente in sviluppo e in movimento. I vantaggi e i benefici apportati da questi strumenti sono molteplici e molto importanti, soprattutto per i bambini perché offrono la possibilità di andare a sviluppare molte capacità cognitive, sociali, motivazionali ed emozionali. Nonostante questi aspetti, il rischio di lasciare qualcuno indietro è molto alto.

I bambini con disabilità motorie gravi faticano o non possono proprio utilizzare i videogiochi nello stesso modo con cui vengono utilizzati dai bambini normodotati. Questo fatto può essere la causa della perdita di molte opportunità, come l'acquisizione di determinate abilità, di stare con gli altri, di divertirsi e di crescere come tutti gli altri bambini.

Il rischio che i bambini con disabilità motoria si sentano soli ed esclusi aumenta se essi non hanno passioni e attività da condividere con i propri coetanei.

Questo lavoro mi ha offerto l'opportunità di avvicinarmi a questo mondo e alla comprensione di questi problemi di cui spesso non è facile accorgersene. Quello che ci si auspica è che questi problemi possano raggiungere la sensibilità delle maggiori aziende di sviluppo di videogiochi in modo che possano fornire una maggiore quantità di software accessibile anche da parte di chi ha gravi disabilità motorie.

Con questo lavoro si è voluto offrire ai bambini disabili questa opportunità: giocare alla pari con i propri amici con un videogioco stimolante e collaborativo che costringa i bambini ad interagire tra loro senza competizione, a parlarsi e a conoscersi passando del tempo insieme e sviluppare alcune capacità.

Questo è stato il principale scopo di questa tesi. I risultati ottenuti dimostrano che l'obiettivo prefissato è stato raggiunto e che il gioco rende possibile una socializzazione tra due bambini normodotati. Purtroppo i test con coppie formate da un bambino disabile ed uno normodotato non è ancora stato possibile effettuarli, ma si presume che i risultati siano molto simili.

9.1 Futuri sviluppi

Alcuni tra i possibili sviluppi di questo lavoro sono sicuramente le sperimentazioni effettuabili con bambini con gravi disabilità motorie. Dai risultati di queste analisi relative al gioco *Il labirinto dei mostri* si potrà osservare se si riesce a soddisfare il bisogno di compagnia dei bambini disabili, se si riesce a farli divertire, giocare e integrare più facilmente con i loro coetanei normodotati facendoli sentire meno soli e più accettati nonostante le loro disabilità. Questo si è dimostrato possibile per coppie di bambini normodotati e non c'è nulla che faccia pensare che con i bambini disabili i risultati saranno diversi.

Altre sperimentazioni, per entrambi i videogiochi sviluppati, potrebbero essere utili per valutarne la giocabilità e l'efficacia nello sviluppo dei bambini e delle componenti educative che si vogliono andare a migliorare.

Inoltre, futuri sviluppi possono esserci andando a sviluppare videogiochi che sfruttano le potenzialità di GNomon per poter offrire ai bambini disabili un portafoglio di giochi più ampio, con giochi di vario tipo, stimolanti, divertenti ed educativi.

Per esempio, si potrebbe andare a sviluppare il terzo gioco che è stato proposto per questa tesi: il gioco *Unisci i puntini* date le potenzialità e le componenti educative che presenta.

Bibliografia

- [1] S. Aced López, F. Corno, and L. De Russis. *Playable One-Switch Video Games for Children with Severe Motor Disabilities Based on GNomon*. Torino, Italia, 2015.
- [2] S. Aced López, F. Corno, and L. De Russis. *Clocks, Bars and Balls: Design and Evaluation of Alternative GNomon Widgets for Children with Disabilities*. ACM CHI, San José, California, 2016. In stampa.
- [3] AESVI. *Il mercato dei videogiochi in Italia (2014-2015)*. http://www.aesvi.it/cms/attach/editor/Infografica_AESVI_2014-2015.pdf. [Online; ultimo accesso 02-Febbraio-2015].
- [4] J. Angelo. *Comparison of three computer scanning modes as an interface method for persons with cerebral palsy*, volume 46. American Journal of Occupational Therapy, 1992.
- [5] Simona Caravita Annella Bartolomeo. *Il bambino e i videogiochi. Implicazioni psicologiche ed educative*. Edizioni Carlo Amatore - Fiera Publishing Group, agosto 2004.
- [6] A. Baddeley. *Working Memory*. *Science*, 255:556–559, January 1992.
- [7] S. Bellocchi, L. Giombini, and S. Contento. *L'evoluzione delle capacità di apprendimento: un'analisi delle componenti cognitive verbali e non verbali*. *Difficoltà di apprendimento*, 14, Dicembre 2008.
- [8] T. Broderick. *Nomon: efficient communication with a single switch*. St. John's College, University of Cambridge, giugno 2006.
- [9] S. Corradini. *Giochi di gruppo, integrazione e diversità*. http://www.unife.it/lettere/filosofia/filo.edu/studiare/tirocinio-formativo/materiale-seminari-teorici/a-a-2012_2013/GI0C0_stefania_Corradini.pdf. [Online; ultimo accesso 02-Febbraio-2015].
- [10] McCormack L.P. Reilly R.B. Knops H. Craddock, G.M. *Assistive Technology: Shaping the Future*. IOS Press, agosto 2003.
- [11] Perry Else. *The Value of Play*. Bloomsbury Academic, aprile 2009.

- [12] Rita L. Ernest R. Hilgard, Richard C. Atkinson. *Psicologia. Corso introduttivo*. Giunti, 1989.
- [13] James C. Kaufman Garo P. Green. *Videogames and Creativity*. Academic Press in an impiant of Elsevier, 2015.
- [14] Kenneth R. Ginsburg. *The Importance of Play in Promoting Healthy Child Development and Maintaining Strong Parent-Child Bonds*, volume 119. January 2007.
- [15] Jeffrey Goldstein. *Il gioco nello sviluppo, nella salute e nel benessere del bambino*. http://www.assogiocattoli.it/files/allegati_pubblicazioni_file_452045332.pdf, febbraio 2012. [Online; ultimo accesso 07-Ottobre-2015].
- [16] Rutger C. M. E. Engels Isabela Granic, Adam Lobel. *The Benefits of Playing Video Games*. *American Psychologist*, 69:66–78, January 2014.
- [17] R. Glase J. W Segal, S. F. Chipman. *Thinking and Learning Skills: Volume 1: Relating Instruction To Research*, volume 1. Routledge, June 2014.
- [18] Georgios Kouroupetroglou. *Assistive Technologies and Computer Access for Motor Disabilities*. IGI Global, ottobre 2013.
- [19] Georgios Kouroupetroglou. *Disability Informatics and Web Accessibility for Motor Limitations*. IGI Global, ottobre 2014.
- [20] Panier Bagat Matilde and Sasso Salvatore. *L'altra crescita. Percorsi osservativi e descrittivi dello sviluppo psicosociale delle persone con disabilità*. Franco Angeli, novembre 1995.
- [21] A. Munna. *Le funzioni attentive e la riabilitazione neuropsicologica*. <http://www.metaintelligenze.it/sito/le-funzioni-attentive-e-la-riabilitazione-neuropsicologica/>. [Online; ultimo accesso 14-Gennaio-2015].
- [22] Commissione ONU. *La Convenzione delle Nazioni Unite sui diritti delle persone con disabilità*. <http://www.governo.it/backoffice/allegati/42085-5202.pdf>. [Online; ultimo accesso 15-Ottobre-2015].
- [23] E. Pollina. *Una ricerca su come è cambiata l'interazione tra bambini e tecnologie*. http://www.corriere.it/scienze/11_gennaio_20/bambini-digitali-pollina_1b9ec506-2488-11e0-8269-00144f02aabc.shtml. [Online; ultimo accesso 02-Febbraio-2015].
- [24] Miguel Santos Rui Olivia Teles. *Tecnologie Assistive*. http://www.cp-pack.eu/index.php?option=com_content&view=article&id=75&Itemid=69&lang=it. [Online; ultimo accesso 03-Ottobre-2015].
- [25] Brian Sutton-Smith. *The Ambiguity of Play*. Harvard University Press, maggio 1997.