



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

# BitStory: Serious Game for Education in Computer Science

RELATORI

Prof. **Fabio Palomba**

Dott.ssa **Giusy Annunziata**

Università degli Studi di Salerno

CANDIDATO

**Raffaele Aurucci**

Matricola: 0512110832

Anno Accademico 2022-2023

*Questa tesi è stata realizzata nel*

sesa<sup>lab</sup>  
SOFTWARE ENGINEERING  
SALERNO

*"La creatività non è altro che un'intelligenza che si diverte" (Albert Einstein)*

## **Abstract**

Nell'ambito della didattica, sta emergendo un crescente interesse per le attività svolte mediante la gamification - una pratica che consente di inserire meccaniche di gioco in contesti non ludici - al fine di motivare e tenere impegnati gli studenti. La gamification è a sua volta integrata in molti altri contesti, tra cui i Serious Game - giochi progettati per offrire un'esperienza educativa e al contempo intrattenere il giocatore. Lo stato dell'arte mostra che i Serious Game rappresentano delle valide alternative ai classici compiti per casa, esercizi in aula o attività laboratoriali. Tuttavia, sono ancora poche le realtà scolastiche che adottano questo tipo di attività. La principale problematica dell'istruzione moderna è legata alla carenza di impegno e di motivazione degli studenti attuali - i cosiddetti nativi digitali. A questa si aggiungono, metodologie di insegnamento obsolete e uno scarso interesse verso il mondo dell'informatica, e più in generale verso le discipline STEM (Science, Technology, Engineering and Mathematics); di conseguenza, nel tempo è anche maturato un notevole divario di genere in questi settori. Partendo da tali motivazioni e per soccombere a tali lacune, è stato realizzato BitStory, un Serious Game nato per incentivare l'utilizzo di nuove metodologie didattiche, educare ed avvicinare i bambini delle scuole primarie all'informatica. Il gioco, permette di controllare un avatar in un mondo popolato da vari personaggi: pionieri della computer science. Questi, raccontano le loro storie, spiegano concetti e propongono sfide a tema. Oltre alle sfide, il gioco integra al suo interno altri elementi di gamification, in particolare: punteggi e premi. Lo sviluppo ha richiesto un'attenta analisi della letteratura, per comprendere quali elementi rendessero coinvolgenti i giochi educativi e quali caratteristiche presentassero i Serious Game per bambini. Sono stati stabiliti i punti di forza e i punti di debolezza di alcuni lavori presenti allo stato dell'arte, definendo così i requisiti di BitStory, il quale sfrutta i punti di forza dei lavori già noti e ne soccombe le limitazioni. Dunque, è stata sviluppata una demo del gioco, il cui potere motivazionale è stato valutato sulla base delle ricerche effettuate.

---

## Indice

---

<b>Elenco delle Figure</b>	<b>iii</b>
<b>Elenco delle Tabelle</b>	<b>iv</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Motivazioni e obiettivi . . . . .	1
1.2 Metodo e risultati ottenuti . . . . .	2
1.3 Struttura della tesi . . . . .	3
<b>2 Background e stato dell'arte</b>	<b>4</b>
2.1 Gamification: apprendere con divertimento . . . . .	4
2.2 Esplorando i Serious Game . . . . .	8
2.3 Fondamenti per un gioco educativo coinvolgente . . . . .	10
2.4 Lavori correlati: videogiochi per l'informatica . . . . .	16
2.4.1 Nabbovaldo e il ricatto del cyberspazio . . . . .	16
2.4.2 while True: learn() . . . . .	19
2.5 BitStory: motivazioni alla base dello sviluppo . . . . .	21
2.6 Tecnologie impiegate e ragioni di utilizzo . . . . .	22
<b>3 Game Development</b>	<b>25</b>
3.1 La modellazione 3D . . . . .	25

---

3.2	Le animazioni dei personaggi . . . . .	29
3.3	Lo sviluppo in Unity . . . . .	30
3.3.1	La scena del mondo di gioco . . . . .	30
3.3.2	La scena del menu principale . . . . .	32
3.3.3	La scena del minigioco di Ada Lovelace . . . . .	34
3.3.4	Le scene di mockup . . . . .	35
3.3.5	Le musiche e i suoni . . . . .	36
3.4	Metodo di valutazione delle prestazioni . . . . .	37
<b>4</b>	<b>BitStory</b>	<b>38</b>
4.1	Introduzione al gioco . . . . .	38
4.2	I minigiochi . . . . .	42
4.3	La ricerca applicata al Game Design . . . . .	51
4.4	Risoluzione alle limitazioni dei lavori correlati . . . . .	56
4.5	Risultati delle prestazioni di gioco . . . . .	57
<b>5</b>	<b>Conclusioni</b>	<b>58</b>
5.1	Panoramica del lavoro svolto . . . . .	58
5.2	Discussioni . . . . .	59
5.3	Sviluppi futuri . . . . .	60
	<b>Bibliografia</b>	<b>61</b>

---

## Elenco delle figure

---

2.1	Immagine tratta dalla serie Renovatio Quest [1] . . . . .	9
2.2	I quattro elementi base di un gioco [2] . . . . .	11
2.3	Alcuni dei minigiocchi da affrontare durante l'avventura [3] . . . . .	17
2.4	Approfondimenti storici e spiegazioni delle componenti di gioco [4] .	19
3.1	Rendering del mondo di gioco realizzato con Eevee [5] . . . . .	26
3.2	Rendering dei personaggi realizzato con Eevee [5] . . . . .	26
3.3	Auto-Rigger di un personaggio effettuato in Mixamo [6] . . . . .	29
3.4	Elementi della UI del mondo di gioco inclusi nella demo [7] . . . . .	32
3.5	Elementi della UI del menu principale inclusi nella demo [7] . . . . .	33
3.6	Elementi della UI del minigioco di Ada Lovelace inclusi nella demo [7]	35
4.1	Il minigioco di Ada Lovelace . . . . .	43
4.2	Il minigioco di Claude Shannon . . . . .	44
4.3	Il minigioco di Horst Feistel . . . . .	45
4.4	Il minigioco di Tim Berners-Lee . . . . .	46
4.5	Il minigioco di Dennis Ritchie . . . . .	47
4.6	Il minigioco di Alan Turing . . . . .	48
4.7	Il minigioco di John McCarthy . . . . .	49
4.8	Il minigioco di George Boole . . . . .	50

---

## Elenco delle tabelle

---

2.1	Termini correlati al concetto di gamification [8] . . . . .	7
2.2	Game motivators per un videogioco educativo [8] . . . . .	14
4.1	Risultati del benchmark della demo di gioco [7] . . . . .	57



# CAPITOLO 1

---

## Introduzione

---

### 1.1 Motivazioni e obiettivi

Oggi appare evidente che le metodologie di insegnamento adottate nei contesti scolastici siano diventate obsolete. Gli studenti attuali sono considerati nativi digitali e la loro carenza di impegno e di motivazione rappresentano la principale problematica legata al processo di apprendimento. A tutto questo, si aggiunge uno scarso interesse verso il mondo dell'informatica e in generale verso le discipline STEM (Science, Technology, Engineering and Mathematics); solo il 24% delle lauree totali in Italia. E se poi consideriamo il divario di genere, ormai da tempo presente in questi settori, in particolare nell'informatica, dove si contano solo il 16,6% di donne laureate nel 2022, è indubitabile che c'è la necessità di cambiare qualcosa, a partire dall'istruzione!

Nel corso degli ultimi anni, si è cercato quindi di rafforzare la didattica inserendo meccaniche di gamification, come premi, punti, classifiche, etc., alle attività svolte in aula. La gamification, vista come processo di progettazione, è integrata in molti altri contesti, come Applied Game, Trasformational Game, etc., ma più in particolare, nell'ambito didattico, è bene parlare dei Serious Game. Questi, sono giochi progettati per offrire un'esperienza educativa e al contempo intrattenere, sfruttando l'immersività, una qualità necessaria per motivare gli studenti e facilitare l'apprendimento.

La stato dell'arte mostra come i Serious Game siano stati utilizzati come alternativa ai classici compiti per casa, esercizi in aula o attività laboratoriali. Tuttavia, sono ancora poche le realtà scolastiche che adottano queste nuove metodologie didattiche.

Affiché gli studenti possano intraprendere una carriera nell'informatica, c'è la necessità di partire dal basso, in particolare dall'infanzia, stimolando sin dalle scuole elementari, curiosità e interesse verso questo mondo. Bisogna dare loro l'opportunità di orientarsi e conoscere questa disciplina, e in generale le materie STEM, incitandoli soprattutto in attività di problem solving. In questo modo, gli studenti potranno comprendere se questa è la strada giusta per loro, abbattendo ogni tipo di stereotipo. Da queste assunzioni, nasce la necessità di creare uno strumento che faccia conoscere, ma soprattutto educare, i bambini all'informatica.

## 1.2 Metodo e risultati ottenuti

In questa tesi è presentato lo sviluppo di BitStory [7], un Serious Game realizzato per supportare la didattica dei bambini delle scuole elementari. Il gioco permette di controllare un avatar in un mondo popolato da diversi pionieri dell'informatica. Essi, raccontano le loro storie, spiegano concetti e propongono sfide a tema. Il gioco, oltre alle sfide, integra altri elementi di gamification, in particolare, punteggi e premi. Ogni sfida e ogni personaggio sono legati ad una specifica tematica dell'informatica. I temi trattati sono i seguenti: *Algoritmi e Strutture Dati*, *Sicurezza*, *Sistemi Operativi*, *Sviluppo Web*, *Reti di Calcolatori*, *Teoria degli Automi*, *Machine Learning* e *Logica Booleana*. Per realizzare BitStory è stato necessario fare un'attenta analisi della letteratura per comprendere: (i) i fondamenti di un gioco; (ii) gli elementi che rendono i giochi educativi coinvolgenti; (iii) le caratteristiche da considerare nei Serious Game per bambini. Dopo questa analisi, sono stati esplorati alcuni Serious Game correlati, prendendo spunto dai loro punti di forza e cercando di colmare i loro punti deboli. Il processo di sviluppo ha previsto varie fasi: la creazione dell'aspetto grafico sia in Blender che in Inkscape, la creazione delle animazioni in Mixamo e l'implementazione del gioco in Unity. Ciò ha permesso di realizzare una demo, il cui potere motivazionale è stato valutato sulla base delle ricerche condotte. Infine, è stato svolto un benchmark per valutare le prestazioni di gioco su differenti calcolatori.

## 1.3 Struttura della tesi

La tesi è strutturata nel seguente modo:

- **capitolo 2** - spiega il concetto di gamification e delle sue integrazioni in vari ambiti, tra cui i Serious Game. Si approfondisce il contesto dei Serious Game nell'istruzione e si riportano i fondamenti necessari alla realizzazione di giochi educativi coinvolgenti. Successivamente, sono trattati alcuni Serious Game correlati al lavoro di tesi, ponendo il focus sugli aspetti positivi e limitanti di ognuno. Infine, viene posta attenzione alle motivazioni e alle tecnologie alla base dello sviluppo di BitStory.
- **capitolo 3** - illustra le varie fasi dello sviluppo di BitStory, ovvero il Game Development: dalla modellazione 3D, alle animazioni dei personaggi, sino all'implementazione del gioco. Dopodiché, descrive un metodo utilizzato per valutare le prestazioni di gioco.
- **capitolo 4** - spiega gli elementi di Game Design di BitStory: il mondo di gioco, i personaggi, le azioni, le regole, la storia, i minigiochi, etc. Dopodiché, viene effettuata una valutazione del potere motivazionale dei risultati ottenuti, basandosi sulle ricerche condotte nel capitolo 2. Viene inoltre spiegato in che modo sono state risolte le limitazioni dei lavori correlati. Infine, sono discussi i risultati delle prestazioni di gioco.
- **capitolo 5** - fornisce una panoramica del lavoro svolto e mostra alcune osservazioni emerse durante il lavoro. Infine, elenca gli sviluppi futuri che potranno essere svolti.

## CAPITOLO 2

---

### Background e stato dell'arte

---

#### 2.1 Gamification: apprendere con divertimento

Gli studenti attuali sono considerati nativi digitali, poiché cresciuti con l'utilizzo delle tecnologie digitali. In questo contesto, gli insegnanti si trovano di fronte a grandi sfide per adattare l'insegnamento alle diverse modalità di apprendimento degli studenti. Tra gli approcci e le tecniche educative utilizzate per migliorare la motivazione e l'interesse degli studenti, la **gamification** può essere una strategia vincente [9].

Secondo Karl Kapp, autore del libro *"The Gamification of Learning and Instruction"*, la gamification rappresenta il processo di utilizzo di elementi di gioco e tecniche di game design in contesti non ludici per coinvolgere e motivare le persone, promuoverle all'apprendimento e aiutarle a raggiungere i loro obiettivi [10].

La carenza di **impegno** e di **motivazione** degli studenti rappresentano le principali problematiche dell'istruzione moderna, in quanto influiscono negativamente sulla partecipazione attiva degli studenti al processo di apprendimento [9].

Come descritto da Laine e Lindberg nel loro articolo, possiamo definire l'impegno come il livello di coinvolgimento che lo studente mostra verso il processo di ap-

prendimento, mentre la motivazione come la ragione per cui lo studente rimane impegnato in un'attività di apprendimento [8].

L'impegno, oggetto di studio della psicologia e della scienza comportamentale, si divide in due macrocategorie, ovvero *l'impegno motivato estrinsecamente* e *l'impegno motivato intrinsecamente* [11]. Il primo concetto collega l'impegno a fattori esterni, come ricompense in denaro, voti, evitamento di punizioni, etc., quindi le azioni sono volte al raggiungimento di un obiettivo esterno. Il secondo concetto collega l'impegno a fattori interni, come l'interesse personale, la ricerca di sfide, la collaborazione, etc., che portano l'individuo ad uno stato di piacere durante l'esecuzione di un'attività. Egli può sperimentare anche ciò che viene chiamato *stato di flow*, in cui raggiunge un livello di concentrazione completo, tale da non accorgersi dello scorrere del tempo [12].

Secondo lo studio condotto da Kyriakova et al. [9], gli elementi di gioco che tipicamente svolgono un ruolo fondamentale nella gamification sono:

- **utenti** - tutti i partecipanti ad una determinata attività, possono essere studenti, dipendenti, clienti, etc;
- **sfide** - prevede l'esecuzione di uno o più task atti al raggiungimento di determinati obiettivi;
- **punti** - accumulati come conseguenza dell'esecuzione di sfide;
- **livelli** - il livello di esperienza dall'utente, determinato dal sistema di accumulo dei punti;
- **premi** - ottenuti come riconoscimento del completamento di determinate azioni;
- **classifiche** - lista dei giocatori, ordinata in base ai rispettivi punteggi ottenuti nelle sfide, livelli o altri criteri;

In ambito didattico la gamification può venire applicata in vari modi, ad esempio si può pensare di inserire elementi di gioco all'interno di attività da svolgere in classe (eg. esercizi alla lavagna) oppure sfruttare i videogiochi educativi, come *Minecraft Education* [13], il quale stimola la creatività, il problem solving, la collaborazione e l'insegnamento delle materie STEM, tra cui l'informatica mediante il coding a blocchi.

È stata adottata la gamification anche in contesti universitari, in particolare a supporto dei corsi di informatica in Brasile, come raccontano Gresse et al. [14], autori di *Scrumia*, un gioco educativo carta e penna. Il gioco ha come scopo principale quello di rafforzare la comprensione di SCRUM, una metodologia agile di gestione dei progetti software, mettendo in pratica ciò che gli studenti imparano teoricamente durante i corsi di Ingegneria del Software.

In *Scrumia* gli studenti vengono divisi in gruppi da 6, in cui i membri assumono i ruoli di *product owner*, *scrum master* e *stakeholder*. Essi si troveranno a lavorare per un'ipotetica azienda nel contesto di un'iniziativa di portare in vacanza le persone della Terra verso il pianeta *Scrumia*. L'esecuzione del gioco consiste nella realizzazione dei prodotti richiesti dai clienti, come barchette di carta, aeroplani, cappelli etc., ai fini di massimizzare il guadagno aziendale in termini di profitto e punti felicità. Gli studenti avranno a disposizione 60 minuti in cui dovranno: (i) stimare i punti e il profitto dal catalogo delle user stories; (ii) pianificare lo sprint; (iii) eseguire lo sprint, formato da kick-off meeting iniziale e 3 periodi di tempo composti da esecuzione e meeting giornalieri; (iv) rilasciare i prodotti; ogni gruppo somma il guadagno ottenuto dai progetti accettati e si stabilisce il vincitore; (v) debriefing in cui si valuta l'esperienza complessiva nell'uso di SCRUM.

Questo studio sottolinea come l'impegno e la motivazione degli studenti sia stato elevato durante l'esecuzione del gioco, accompagnandoli nel processo di apprendimento, in particolare la componente collaborativa, la competizione, la competenza richiesta, la sfida e l'immersione sono state quelle maggiormente apprezzate.

Data la popolarità del concetto di gamification col tempo sono nati molti altri concetti, come *Serious Game*, *Applied Game*, *Educational Game*, *Transformational Game*, *Exergame*, *Ludic Interface*, *Game-based Learning*, etc., che possono sembrare integrativi al concetto iniziale, ma in realtà la gamification è considerata un processo di progettazione, mentre gli altri termini sono ritenuti dei risultati finali, delle qualità o delle attività [8].

Nella tabella 2.1 sono descritti in breve vari termini correlati al concetto di gamification, con riferimenti a diversi prodotti realizzati in alcuni contesti.

**Tabella 2.1:** Termini correlati al concetto di gamification [8]

Termine	Descrizione
Serious Game	Giochi progettati per offrire un’esperienza educativa e al contempo intrattenere il giocatore (eg. Renovatio Quest [1], Minecraft Education [13]).
Applied Game	Giochi progettati per applicare nella realtà ciò che viene appreso durante l’esperienza di gioco (eg. Flight Simulator [15])
Educational Game <sup>1</sup>	Giochi progettati a puro scopo educativo, atti a facilitare l’apprendimento di concetti didattici o di formazione professionale (eg. Code.org [16], Duolingo [17]).
Transformational Game	Giochi progettati per dare un impatto positivo e trasformativo sulla vita delle persone, stimolando la crescita personale, lo sviluppo psicologico o il cambiamento comportamentale. Possono affrontare tematiche come la gestione dello stress, l’autostima, l’empatia, etc. (eg. Cancer Space [18]).
Exergame	Giochi progettati per combinare l’esercizio fisico con l’esperienza interattiva (eg. Ring Fit Adventure [19]).
Edutainment	Contenuti che combinano elementi educativi con elementi di intrattenimento per insegnare o trasmettere conoscenze in modo divertente. Può essere applicata in vari contesti come documentari e musei interattivi, programmi e giochi educativi, etc..
Ludic Interface	Interfacce utente progettate tenendo conto dei principi di game design, al fine di rendere più stimolante e divertente l’interazione con un sistema, un’applicazione, un sito web, etc. (eg. Nike Run Club [20]).
Game-based Learning	Attività che coinvolgono l’uso dei giochi come strumenti principali per l’apprendimento.
Digital Game-based Learning	Attività che coinvolgono l’uso di giochi digitali come strumento di apprendimento.

<sup>1</sup> i giochi educativi sono un sottoinsieme dei Serious Game, per questo motivo nella letteratura vengono spesso utilizzati in modo interscambiabile [21].

## 2.2 Esplorando i Serious Game

In questa sezione sarà esplorato il mondo dei **Serious Game**, l'area di interesse di questo lavoro di tesi, partendo dalle origini, passando per le motivazioni, fino ad arrivare alle relative applicazioni nell'ambito dell'istruzione. I Serious Game si distinguono dai giochi tradizionali poiché il loro obiettivo principale non è solo quello di intrattenimento, ma di fornire conoscenze, abilità o valori a fini educativi. Questa tipologia di giochi può essere utilizzata in vari contesti come l'istruzione, la formazione professionale, la salute e il benessere, il marketing, etc. [22].

Il termine "Serious Game" in ambito puramente digitale ha origini dalla fondazione nel 2002 della *Serious Game Initiative* di David Rejeski e Ben Sawyer negli USA [23]. Inizialmente i Serious Game furono utilizzati come strumento per l'addestramento del personale dell'esercito, come dimostra la serie di videogiochi *America's Army*, sviluppata dall'esercito statunitense [24].

Come dichiarano De Gloria et al. [25] i Serious Game non si rivolgono soltanto ai power gamers, ovvero i videogiocatori che dedicano enormi quantità di tempo nel tentativo di essere competitivi e dominare il gioco, ma ad un pubblico molto più ampio. Ciò è dovuto al fatto che negli ultimi anni il numero di giocatori è drasticamente aumentato [26]. Dunque, questo ha permesso di dare maggiore attenzione all'impiego e al crescente sviluppo dei Serious Game.

Uno studio condotto da Ryan et al. precisa come l'impegno motivato possa essere trovato nel mondo dei videogiochi, grazie alle loro capacità di intrattenere per lunghi periodi di tempo [27]. Ciò accade poiché al centro di ogni gioco c'è un sistema di ricompensa nascosto che premia il giocatore per convincerlo a continuare a giocare. Se il gioco ha un gameplay, una storia ed altri elementi appaganti, allora l'endorfina viene rilasciata nel corpo, favorendo la felicità e la concentrazione.

A questo punto l'apprendimento diviene un pensiero secondario, rendendolo di fatto più semplice e di conseguenza il giocatore conserverà più informazioni nella memoria a lungo termine [22].



Nell'ambito dell'**istruzione** i Serious Game possono essere utilizzati come strumento rafforzativo alla didattica, come testimonia la serie di videogiochi *Renovatio Quest* [1] di Enea Montoli, docente presso il liceo statale di Milano. In questa serie di videogiochi, composta da giochi di ruolo, multilivello puzzle-platform e da (video)libro game, la didattica abbraccia il divertimento videoludico per permettere agli studenti di svolgere i classici esercizi di fisica e matematica in maniera alternativa. Il giocatore di fatto si troverà catapultato in una narrazione e durante la sua avventura dovrà affrontare problemi di matematica e fisica per risolvere puzzle o aiutare altri personaggi, così da ottenere punti e aumentare il proprio livello.



**Figura 2.1:** Immagine tratta dalla serie Renovatio Quest [1]

Altro Serious Game, questa volta a supporto dell'insegnamento dei corsi di Algoritmi e Strutture Dati è stato realizzato in una Università dell'Iran. Il gioco consiste nell'ordinare delle scatole nello spazio mediante un robot, applicando ad ogni livello un diverso algoritmo di ordinamento (eg. bubble sort), in più verranno poste delle domande allo studente durante il gioco per verificare le sue conoscenze. Alla fine di ogni livello verrà quindi premiato. Dalle indagini condotte si evince che il gioco si presta bene all'apprendimento e vada a motivare gli studenti [22].

Alcuni studi hanno individuato aspetti negativi nell'uso dei Serious Game nell'educazione scientifica. In particolare, si è osservato che quando un gioco richiede un carico cognitivo elevato, ciò può comportare un maggiore sforzo da parte del giocatore, con conseguenti effetti negativi sull'apprendimento [28]. Per questo motivo è di fondamentale importanza studiare la progettazione di un gioco educativo, stabilendone il target di riferimento e gli elementi essenziali in suo possesso.

## 2.3 Fondamenti per un gioco educativo coinvolgente

Come racconta Jesse Shell, autore del libro *"The Art of Game Design"*, *il gioco non è un'esperienza, ma crea un'esperienza*, ed è per questo motivo che bisogna capire quali sono gli **elementi essenziali** che definiscono l'esperienza che si vuole creare e renderli parte integrante durante tutto il **processo di progettazione** di un gioco [29].

Naturalmente quest'ultima è tutt'altra che banale, sono richieste abilità in animazione, antropologia, architettura, business, cinematografia, comunicazione, scrittura, economia, ingegneria, storia, gestione, matematica, programmazione, musica, psicologia, grafica, etc., ma l'abilità più importante è **l'ascolto**, ovvero bisogna saper ascoltare se, il gioco, il pubblico di riferimento, il cliente e il proprio team [30].

Dal punto di vista del **target**, lo scopo è chiaramente quello di progettare un gioco dedicato ai bambini delle scuole primarie. Questa è l'età della ragione, in cui essi sono in grado di leggere, ragionare e di risolvere problemi, nonché l'età in cui decidono autonomamente quali giochi preferire, rifiutando ciò che scelgono i genitori [31].

Altra componente da non sottovalutare è il **genere**. A tal fine sono stati definiti cinque elementi che uomini e donne amano vedere nei giochi. Nel caso degli uomini prevale la padronanza del gioco, la competizione, la distruzione, la presenza di puzzle e l'apprendimento attraverso gli errori. Nel caso delle donne viene posta maggiore attenzione alle emozioni, ai riferimenti col mondo reale, all'accudimento, al dialogo e agli enigmi verbali e infine all'apprendimento attraverso gli esempi [31].

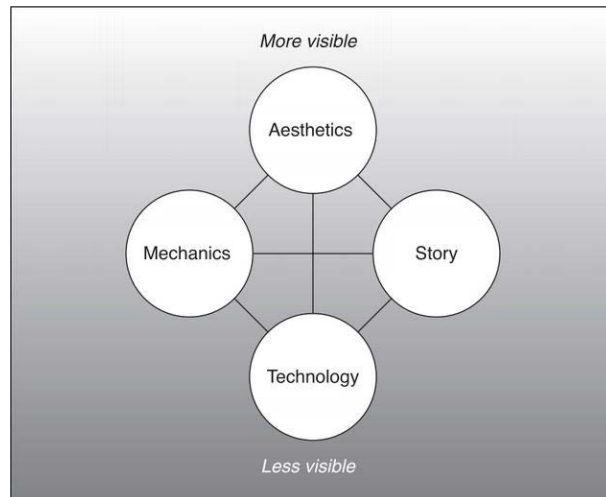
Di seguito sono elencati i quattro elementi di base che compongono un gioco, ognuno dei quali è concettualmente correlato con gli altri [2]:

**Meccaniche.** Si riferiscono alle regole e alle interazioni che determinano il funzionamento del gioco. In sostanza è come descrivere l'obiettivo del gioco, le azioni che il giocatore può compiere per raggiungerlo, nonché le risposte del gioco a tali azioni.

**Storia.** Si riferisce alla sequenza di eventi che mira a creare il contesto del gioco. Le storie possono essere lineari, ovvero seguire un flusso unico, o ramificate, ossia lasciare che le azioni del giocatore determinino il flusso della storia. Un gioco può anche non avere alcuna storia e lasciare che le ambientazioni, i personaggi non giocanti e gli oggetti di gioco mirino a creare una storia implicita.

**Tecnologia.** Si riferisce agli strumenti e ai sistemi che rendono possibile l'interazione col gioco. Può comprendere hardware, dispositivi di input, etc., stabilendo chiaramente le capacità e le limitazioni del gioco.

**Estetica.** Si riferisce all'aspetto, al suono e alle sensazioni generali che il gioco è in grado di trasmettere.



**Figura 2.2:** I quattro elementi base di un gioco [2]

Lo studio svolto da Laine e Lindberg ha contribuito a definire gli **elementi motivazionali** (*game motivators*) e i **principi di progettazione** (*design principles*) fondamentali che rendono coinvolgenti i giochi educativi [8].

Gli **elementi motivazionali** identificati sono descritti brevemente nella tabella 2.2. Tuttavia questi elementi possono sembrare in un primo momento astratti, di conseguenza sono state elaborate le seguenti **classi** di principi di progettazione:

- **Sfida** - design principles volti alla creazione di sfide stimolanti, tra questi:
  - Fornire sfide a livelli di difficoltà regolabili;
  - Privilegiare le sfide semplici rispetto a quelle complesse;
  - Fornire tempo sufficiente per risolvere le sfide;
  - Suscitare curiosità con sfide interessanti e imprevedibili;
  - Consentire la ripetizione delle sfide;
  - Utilizzare modalità di input adatti ai giocatori e ai contesti di destinazione

- **Controllo** - design principles volti alla capacità del giocatore di compiere azioni e le modalità di input con il quale può relazionarsi col gioco, tra questi:
  - Utilizzare input adatti ai giocatori e ai contesti di destinazione;
  - Utilizzare controlli sensibili e accurati;
  - Utilizzare controlli coerenti;
  - Utilizzare controlli familiari, sicuri e confortevoli;
  - Favorire un'interazione semplice;
  - Libertà di scelta e di controllo nel gameplay;
- **Creatività** - design principles volti a evitare esperienze di gioco univoche, promuovendo l'espressione personale, tra questi:
  - Consentire ai giocatori di creare contenuti di gioco;
  - Fornire modi creativi per risolvere le sfide;
- **Esplorazione** - design principles volti all'esplorazione del mondo di gioco e dei suoi contenuti, tra questi:
  - Libertà di esplorazione e sperimentazione;
  - Fornire più percorsi/opzioni e un ordine dinamico degli eventi;
- **Equità** - design principles volti ad assicurare che il gioco possa essere goduto in modo equo da tutti i giocatori, tra questi:
  - Garantire simili opportunità di successo a prescindere dall'esperienza;
- **Feedback** - design principles volti a definire le modalità con cui il gioco risponde alle azioni del giocatore, tra questi:
  - Fornire istruzioni e/o tutorial;
  - Fornire un feedback immediato, positivo e utile;
  - Fornire un feedback chiaro mediante diversi canali;
  - Fornire accesso ai dati sulle prestazioni del giocatore;

- **Obiettivi** - design principles volti a creare giochi coinvolgenti, tra questi:
  - Creare obiettivi chiari, significativi e raggiungibili;
- **Learning** - design principles volti a supportare l'apprendimento, tra questi:
  - Fornire contenuti e attività di apprendimento pedagogicamente fondanti;
  - Fornire sfide cognitive;
  - Incorporare strumenti di valutazione;
  - Scegliere il livello di trasparenza per i contenuti didattici;
  - Fornire il tempo necessario per riflettere nei momenti di gioco più intensi;
- **Profilo e proprietà** - design principles necessari alla gestione dei dati di gioco, tra questi:
  - Usare un profilo/avatar che il giocatore può possedere;
- **Rilevanza e attinenza** - design principles che rendono il gioco più allettante rispetto al contesto o alle esperienze del giocatore, tra questi:
  - Mettere in relazione il gameplay con i contesti del mondo reale;
- **Risorse ed economia** - design principles per la gestione delle risorse, tra questi:
  - Consentire la raccolta di beni virtuali;
- **Gioco sociale** - design principles volti a promuovere la socializzazione, chiave fondamentale per l'apprendimento, tra questi:
  - Fornire opportunità di concorrenza;
  - Offrire opportunità di collaborazione;
- **Narrazione e fantasia** - design principles volti a creare giochi più intriganti nella narrazione e nella fantasia, tra questi:
  - Fornire un contesto fantasy;
  - Offrire esperienze di gioco di ruolo;

**Tabella 2.2:** Game motivators per un videogioco educativo [8]

Motivators	Descrizione
Sfida	La sfida si presenta quando al giocatore viene posto un compito adatto alle sue capacità cognitive. Strettamente legata allo <i>stato di flow</i> [12].
Competenza	La competenza si riferisce all'acquisizione di abilità o capacità mediante il superamento di sfide e il raggiungimento di obiettivi.
Competizione	La competizione nasce quando il giocatore deve competere contro altri giocatori, se stesso o collaborare in gruppi per affrontarne altrettanti.
Controllo	Il controllo si riferisce alla libertà data al giocatore per influenzare il mondo di gioco. Si può dare libertà nei dialoghi, nel modo di affrontare le sfide, nei luoghi da visitare, etc. Legato allo <i>stato di flow</i> [12].
Curiosità	La curiosità può essere innescata dal desiderio di esplorare ogni angolo del gioco, dal scoprire il funzionamento dell'IA, dall'audio, etc.
Emozioni	Le emozioni emergono quando c'è tensione per un evento e dalla conseguente presenza di thrill. Possono fare leva sulle qualità morali intrinseche del giocatore.
Fantasia	La fantasia si riferisce a situazioni fisiche e sociali che non sono presenti nel mondo reale.
Feedback	Il feedback si riferisce a tutto ciò che il giocatore compie nel mondo di gioco, come la risposta ai comandi di gioco, le risposte ai dialoghi con personaggi non giocanti, la distanza da un obiettivo, etc.
Immersione	L'immersione si riferisce allo stato in cui il giocatore si trova profondamente coinvolto nel gameplay, tanto da perdere la percezione del tempo e dell'ambiente circostante.
Novità	La novità si riferisce a nuove meccaniche di gioco, una nuova tecnologia, etc.
Regole	Le regole disciplinano le attività di gioco. Devono essere chiare, flessibili e sincronizzate con gli obiettivi e il feedback.
Relazioni con il mondo esterno	Le relazioni con il mondo esterno emergono dal collegamento tra il mondo reale e il mondo di gioco.
Interazione sociale	L'interazione sociale si riferisce alle comunicazioni sincrone o asincrone con altri giocatori.
Utilità	L'utilità si riferisce a ciò che il gioco offre oltre al divertimento.

Una revisione della letteratura svolta da Yanti et al. [32] ha definito le caratteristiche dei Serious Game che influiscono maggiormente sull'apprendimento dei bambini:

1. **facilità d'uso** - la facilità d'uso si riferisce all'accessibilità data dal gioco. La progettazione deve prevedere comandi semplici, istruzioni chiare e agevolazioni verso l'apprendimento, riducendo al minimo la frustrazione del giocatore. Si ricollega alle classi *Sfida, Controllo, Equità, Feedback, Obiettivi, Learning* e *Profilo e proprietà* dei principi di progettazione descritti precedentemente;
2. **interattività** - l'interattività si riferisce alla capacità del gioco di coinvolgere i giocatori, in modo da renderli parte attiva dell'esperienza. Si ricollega alle classi *Esplorazione, Rilevanza e attinenza, Risorse ed economia, Gioco sociale* e *Narrazione e fantasia* dei principi di progettazione descritti precedentemente;
3. **visualizzazione** - la visualizzazione si riferisce all'aspetto visivo del gioco, come la resa degli ambienti, l'aspetto dei personaggi, le informazioni visualizzate a schermo, etc.;
4. **immersione** - l'immersione, come già detto, si riferisce allo stato in cui il giocatore si trova in uno stato di concentrazione tale da perdere concezione di se e del mondo circostante. Si ricollega a tutte le classi dei principi di progettazione descritte precedentemente;

Infine, dallo studio appena citato, si evince che gli *ambienti virtuali tridimensionali* sono la tecnologia maggiormente utilizzata nei Serious Game per bambini, in grado di stimolare l'impegno, la motivazione e il divertimento. A seguire vi sono gli ambienti bidimensionali, le applicazioni web, il sensore kinect e la realtà virtuale.

## 2.4 Lavori correlati: videogiochi per l'informatica

Allo stato dell'arte sono stati realizzati diversi Serious Game per promuovere l'avvicinamento dei bambini e dei ragazzi al mondo dell'informatica e in generale verso quelle che sono le materie STEM [33][34][35].

Di seguito vengono presentati alcuni Serious Game sviluppati negli ultimi anni che incorporano un po' tutto ciò che è stato detto fino a questo momento.

### 2.4.1 Nabbovaldo e il ricatto del cyberspazio

Il Serious Game *Nabbovaldo e il ricatto del cyberspazio*, sviluppato da Grifo Multimedia su iniziativa del CNR (Consiglio Nazionale delle Ricerche), è un gioco ideato con l'obiettivo di stimolare l'interesse dei bambini e dei ragazzi verso la tematica della sicurezza informatica [3]. Il gioco è stato pensato sia per essere fruito singolarmente dai ragazzi ma anche come strumento di insegnamento per i docenti durante le ore di laboratorio [36]. *Nabbovaldo e il ricatto del cyberspazio* propone diversi **minigiochi**, incentrati sul tema della cybersecurity e dell'informatica in generale, le cui categorie sono le seguenti:

- **giochi di logica** - questa tipologia di giochi stimola le capacità di *problem solving* (eg. inserire alcune componenti, come pc, modem, stampante, etc. all'interno di uno spazio a blocchi predefinito, figura 2.3a);
- **giochi di memoria** - questa tipologia di giochi stimola la *memoria a breve termine* (eg. memorizzare una sequenza e ripeterla, figura 2.3b) o la *memoria verbale* (eg. ricostruire un messaggio con parole disordinate, figura 2.3c);
- **giochi arcade** - questa tipologia di giochi stimola la *prontezza dei riflessi* (eg. schiacciare i worm prima che si moltiplichino, entro un tempo limite, evitando di colpire i gatti, figura 2.3d);

I **punti** possono essere incrementati in base al modo in cui si gioca. Infatti, quando il gioco è adottato in un contesto scolastico, è possibile che nasca una concorrenza amichevole tra gli studenti, che potrebbe stimolarli a impegnarsi di più, al fine di riuscire a giocare meglio e ottenere più punti.



Il **mondo di gioco** si sviluppa in piccole zone accessibili dalla mappa principale, nell'ordine che si preferisce. Ciò contribuisce nel dare al giocatore una certa libertà esplorativa. Vi sono inoltre dei legami tra il mondo reale e quello digitale.

Ogni zona presenta diversi **personaggi** che svolgono il ruolo di generici utenti del web (eg. i troll, gli influencer, i flammer, gli scammer, etc.). Quando si interagirà con quest'ultimi, saranno attivati i **dialoghi**, sia semplici che interattivi.

Altro elemento interessante è la **Nabbopedia**, una raccolta di termini legati alla computer science (eg. ransomware, adware, bug, server, etc.). Può essere completata raccogliendo tutti i frammenti sparsi per le varie zone della mappa.

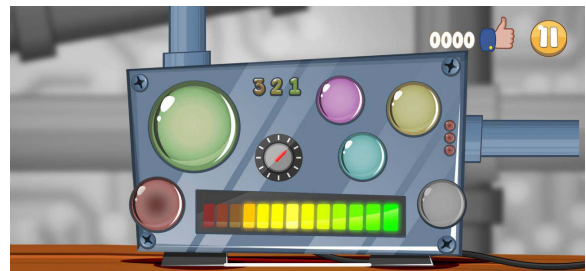
La **storia** segue un flusso unico dall'inizio alla fine e si sviluppa principalmente attraverso i dialoghi. Mentre, per quanto riguarda l'**estetica**, il gioco presenta uno stile grafico prettamente bidimensionale. Invece, dal punto di vista **sonoro**, include musiche rilassanti ed effetti audio coerenti con il tema.

L'**interfaccia utente** è relativamente semplice e contiene solo alcuni elementi essenziali (eg. punti, pulsante menu, etc.).

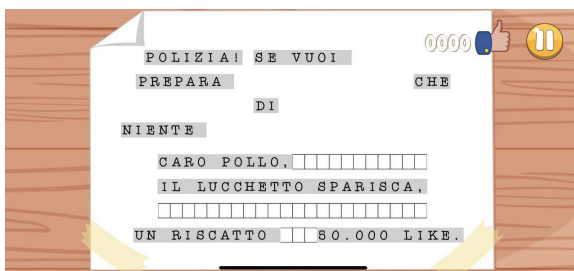
Infine, dal lato **tecnologico**, il gioco è stato sviluppato mediante **Unity** [37], ed è pensato sia per dispositivi mobile che desktop.



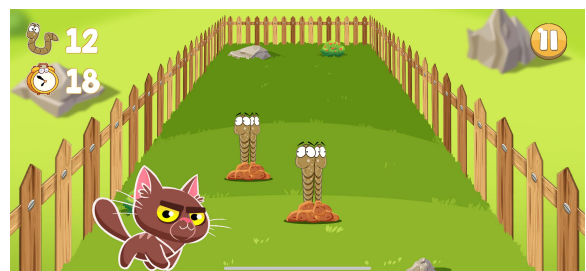
(a) Installazione componenti



(b) Pairing: memorizzazione sequenze



(c) Posta: ricostruzione messaggio



(d) Whack a worm

**Figura 2.3:** Alcuni dei minigiochi da affrontare durante l'avventura [3]

Tra le principali **limitazioni**, il punto debole risiede nei minigiochi: cybersecurity e informatica fungono da solo sfondo. Si nota la non volontà di toccare tematiche più vicine alla computer science (eg. sfida dove applicare un certo algoritmo) o della sicurezza (eg. sfida dove decrittare un messaggio secondo le regole di un cifrario). Spunti che avrebbero potuto arricchire maggiormente la parte educativa del gioco.

Di seguito un sunto sui principali **aspetti positivi** e **limitanti** del gioco:

#### Aspetti positivi del gioco

- Minigiochi di differente tipologia in grado di offrire varie sfide cognitive;
- Mondo di gioco esplorabile liberamente, con legami tra mondo reale e digitale;
- Sistema di punteggio che premia la bravura del giocatore;
- Personaggi non giocanti (NPC) relazionati a utenti generici del web;
- Sistema di dialogo, interattivo e non, con gli NPC;
- Frammenti sparsi per la mappa a fini educativi e collezionistici;
- Storia che si sviluppa principalmente attraverso i dialoghi;
- Interfaccia utente semplice e intuitiva;
- Controlli basati su tocco o click, trascinamento e scorrimento;
- Musiche rilassanti ed effetti audio coerenti con il tema;
- La durata del gioco (4-5 ore) è giusta;

#### Aspetti limitanti del gioco

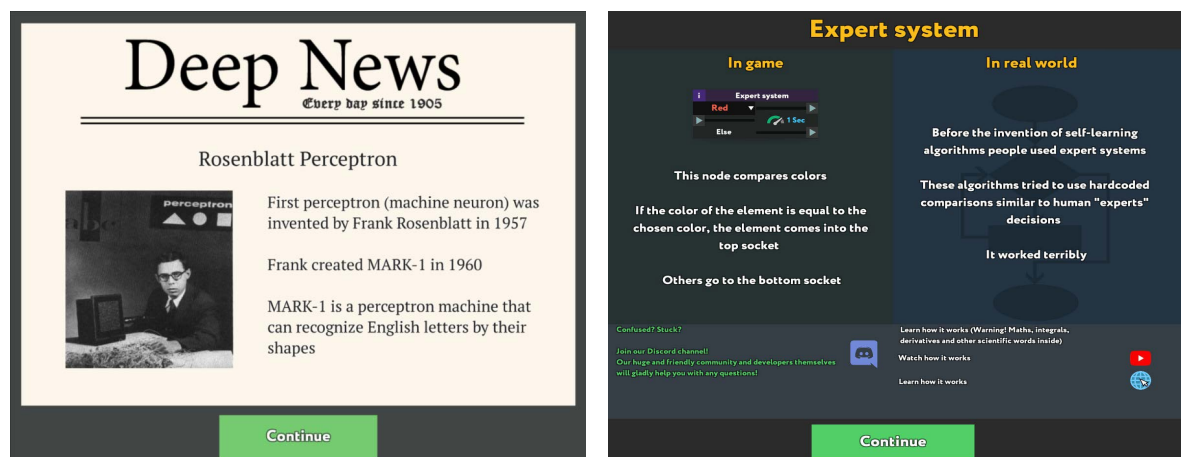
- Minigiochi dove la cybersecurity e l'informatica fungono da solo sfondo;
- Impossibilità di rigiocare una sfida una volta superata;
- Assenza di un registro relativo ai punteggi ottenuti in ciascun minigioco;
- Assenza di premi appaganti in seguito al compimento di determinate azioni;

### 2.4.2 while True: learn()

Il Serious Game *while True: learn()*, sviluppato dal team indipendente Luden.io, è un puzzle game realizzato con l'intento di introdurre i bambini, ma anche gli adulti, verso le tematiche dell'intelligenza artificiale, come il machine learning e il deep learning, e in generale verso il mondo della programmazione [4].

Il gioco presenta una serie di **task**, la cui difficoltà cresce gradualmente, in cui bisogna trasferire dei dati in input verso i corrispettivi flussi di output, utilizzando i nodi, ossia dei blocchi logici, che possono simulare costrutti come l'*if-then-else*, modelli di machine learning come il *decision tree*, modelli di deep learning come il *perceptron*, etc. Una volta realizzato il diagramma di flusso bisognerà avviare il test e verificare che gli input forniti finiscano nei flussi di output corretti, entro il tempo limite, utilizzando al più un certo numero di nodi. Al termine di ogni task, se superato, viene assegnato come premio una **medaglia**, la quale può essere d'oro, d'argento o di bronzo, a seconda del tempo impiegato dal proprio algoritmo.

Componenti decisamente interessanti sono gli **approfondimenti storici** (eg. figura 2.4a) e le **brevi spiegazioni** (eg. figura 2.4b), entrambi associati a ciò che realmente rappresentano i diversi nodi nel mondo della computer science, includendo inoltre collegamenti ad articoli scientifici o video approfondimenti presenti sul web.



(a) Frank Rosenblatt: il primo neurone artificiale

(b) Sistema esperto: tutorial e approfondimenti

**Figura 2.4:** Approfondimenti storici e spiegazioni delle componenti di gioco [4]

Il gioco dispone di un'area dedicata ai **risultati**, un elenco delle azioni portate a termine all'interno del gioco (eg. aver completato tutti i task con una medaglia d'oro). Dal lato **estetico** il gioco ha uno stile grafico bidimensionale. La colonna sonora è composta da musiche che favoriscono la concentrazione, ed effetti audio in stile tech. L'**interfaccia utente** è semplice e intuitiva, presentando durante i task diverse informazioni, come i nodi a disposizione, il tempo, le medaglie, etc.

Dal punto di vista **tecnologico**, il gioco è stato sviluppato attraverso **Unity** [37], ed è pensato sia per pc che per console.

Nel complesso *while True: learn()* è ottimo per stimolare o migliorare le proprie capacità di **problem solving**. Vi sono buone possibilità di far appassionare bambini e non solo al mondo della computer science. I cenni storici e le spiegazioni legate al contesto reale arricchiscono l'offerta, anche se il tutto sembra essere un po' troppo **sbrigativo**, richiedendo approfondimenti online. Il sistema di premi è ben studiato ed è richiesto un certo **impegno** per portare a termine il gioco al 100%.

Di seguito un sunto sui principali **aspetti positivi** e **limitanti** del gioco:

#### Aspetti positivi del gioco

- Task in grado di stimolare la logica e il ragionamento matematico;
- Cenni storici e componenti di gioco in relazione con il mondo dell'informatica;
- Sistema di premi che sollecita al miglioramento della soluzione proposta;
- Registro dei risultati raggiunti durante il gioco;
- Interfaccia utente semplice e intuitiva;
- Controlli basati su mouse (tastiera facoltativa) e adattamento a controller;
- Musiche che favoriscono la concentrazione ed effetti audio in stile tech;

#### Aspetti limitanti del gioco

- Cenni storici e spiegazioni troppo sbrigative, necessità di approfondimenti;
- Numero di task molto ampio, alla lunga il gioco potrebbe risultare ripetitivo;
- Task impegnativi, necessario il supporto della community o di una guida;
- Difficile fare una stima sulla durata del gioco, dipende da persona a persona;

## 2.5 BitStory: motivazioni alla base dello sviluppo

Le motivazioni alla base dello sviluppo del Serious Game *BitStory* [7] sono diverse, visto e considerato che questo progetto mira a raggiungere molteplici obiettivi:

**Supportare la didattica.** Si vuole realizzare uno strumento che possa rafforzare e supportare la didattica, visti i problemi descritti nella sezione 2.1. L'applicazione degli elementi di *gamification* alle meccaniche di BitStory, creerà un'esperienza coinvolgente e allo stesso tempo educativa. Inoltre, si è visto nella sezione 2.2 come l'impegno motivato possa essere trovato nel mondo dei videogiochi, in quanto facilitano l'apprendimento e la conservazione di informazioni in memoria.

**Avvicinare i bambini all'informatica.** Dato che il target di BitStory sono i bambini delle scuole primarie, si vuole avvicinare quest'ultimi al mondo dell'informatica sin dalle prime fasi di crescita, in modo da stimolare curiosità e interesse verso questo mondo. Ciò permetterà di aprire le porte a possibili future carriere nel settore, supportando le aziende IT in quelle che saranno le sfide tecnologiche del domani.

**Abbattere gli stereotipi di genere.** Attraverso BitStory si vuole far comprendere a chiunque che l'informatica è pronta ad accogliere tutte le menti creative, indipendentemente dal genere. Infatti, ad oggi sono ancora poche le donne impegnate in questo settore e il divario inizia proprio dall'istruzione. Secondo un report Istat [38], nel 2021 in Italia solo il 24% dei giovani, tra i 25 e i 34 anni, possiede una laurea nelle discipline STEM, con una quota del 33,7% in favore degli uomini, contro solo il 17,6% delle donne, praticamente la metà! Inoltre, dai dati di AlmaLaurea, si scopre che nel 2022, solo il 16,6% dei laureati in informatica appartiene al genere femminile.

**Raccontare la storia dell'informatica.** Si vuole presentare la storia dell'informatica in una chiave di lettura semplice e chiara, attraverso i racconti dei personaggi che hanno segnato per sempre questo mondo, come Ada Lovelace, Alan Turing, Dennis Ritchie, etc., in modo che i bambini possano avere dei modelli positivi da seguire.

**Fornire sfide educative.** Mediante il gioco, si vuole porre ai bambini delle sfide educative ispirate ad alcuni argomenti o problemi noti della computer science, in modo da stimolare la logica, il ragionamento matematico e la memoria.

## 2.6 Tecnologie impiegate e ragioni di utilizzo

In questa sezione si fornirà una panoramica delle tecnologie utilizzate per lo sviluppo di BitStory, fornendo una descrizione di ciò che sono e le ragioni dietro al loro utilizzo.

**Blender [39].** Un software libero e multiplatforma, rilasciato sotto la *GNU General Public License (GPL)* e sviluppato da Blender Foundation. Blender è ampiamente utilizzato nell'ambito della computer grafica 3D ed è in grado di gestire tutta la pipeline 3D: *modellazione, rigging, animazione, simulazione, rendering, compositing e motion tracking*, includendo anche *l'editing video* e la *creazione di giochi* mediante il Blender Game Engine (BGE), utilizzando blocchi logici o scripting Python; tale engine non è più ufficialmente supportato nelle nuove versioni del software.

Inoltre, Blender ha un'ampia comunità di sviluppatori che utilizzano le API del software per lo scripting Python ai fini di personalizzare l'applicazione o realizzare add-on. I motivi principali per cui è stato preferito Blender sono i seguenti:

- **gratuito** - non c'è necessità di alcuna licenza, anche per fini commerciali;
- **supporto** - disponibilità di documentazione ufficiale online ben fornita, oltre che a una vasta gamma di tutorial e una community sempre attiva;
- **portabilità** - esportazione dei modelli realizzati in vari formati compatibili con le diverse piattaforme;
- **estendibilità** - presenza di add-on ufficiali e non, installabili all'interno del software;
- **semplicità** - curva di apprendimento relativamente rapida, grazie a un'interfaccia semplice che consente di realizzare modelli 3D sin dai primi utilizzi;

**Unity [37].** Un motore grafico multiplatforma, con licenza proprietaria e sviluppato da Unity Technologies. Oltre ad essere un motore grafico, Unity consente di utilizzare anche un ambiente di sviluppo integrato (editor) per la progettazione di contenuti interattivi. Col tempo Unity ha ottenuto una grande popolarità nel mondo del Game Development e consente non soltanto lo sviluppo di videogiochi, ma

anche la visualizzazione di contenuti architettonici e la creazione di animazioni in tempo reale. Uno dei punti di forza di Unity è dovuto proprio al fatto che i giochi prodotti possono essere eseguiti su una vastità di piattaforme anche molto diverse tra loro, come pc, console, dispositivi mobile, dispositivi per la realtà virtuale (VR) e la realtà aumentata (AR), inclusi i browser web utilizzando il plugin Unity Web Player. All'interno dell'editor di Unity è possibile trovare una grossa mole di oggetti utili alla realizzazione della scena di gioco, come *3D objects*, *2D objects*, *UI elements*, *materials*, *colliders*, *animators*, *lights*, *camera*, etc. Inoltre, è presente lo *Unity Asset Store* che permette di ottenere contenuti digitali, in maniera gratuita o a pagamento, da poter utilizzare durante la progettazione della scena di gioco. Infine, Unity consente di gestire il comportamento degli oggetti definiti all'interno della scena mediante scripting in C# o mediante blocchi logici in Unity Visual Scripting. Le ragioni per cui è stato impiegato Unity sono le seguenti:

- **gratuito** - non c'è necessità di alcuna licenza per progetti realizzati a scopo non commerciale oppure per progetti a pagamento che non facciano superare una certa soglia di reddito annuo;
- **supporto** - disponibilità di documentazione ufficiale online ben curata, oltre che a una vasta gamma di tutorial, risoluzioni a problemi già noti e una community sempre attiva;
- **portabilità** - è possibile distribuire i propri progetti in maniera compatibile con le diverse piattaforme a seconda delle specifiche esigenze;
- **integrazione** - attraverso l'editor si possono importare all'interno del proprio progetto modelli 3D, animazioni, texture, materiali, etc., realizzati attraverso altri software (eg. Blender) o mediante l'importazione da Unity Asset Store;
- **familiarità** - per chi ha un minimo di esperienza di programmazione in Java o C++ troverà molto facile programmare gli script in C#, essendo un linguaggio con una sintassi essenzialmente ispirata a quest'ultimi;
- **semplicità** - curva di apprendimento relativamente rapida, grazie a un'interfaccia dell'editor molto semplice che consente agli utenti di realizzare giochi basilari in tempi piuttosto brevi;

**Mixamo [40].** Un servizio online gratuito di animazione di personaggi 3D, sviluppato originariamente da Mixamo Inc. e successivamente acquisito da Adobe. Mixamo offre agli sviluppatori una vasta libreria di personaggi pronti all'uso e animazioni realizzate attraverso la tecnica del *motion capture*. Inoltre, permette di realizzare il *rigging automatico* [6] dei personaggi 3D importati, nonché di scegliere le animazioni da applicare, personalizzando eventualmente alcuni campi (eg. larghezza delle braccia) e di esportare il nuovo modello con la relativa animazione nel formato desiderato. Quindi, le ragioni per cui è stato impiegato Mixamo sono le seguenti:

- **gratuito** - non c'è necessità di alcuna licenza, anche per fini commerciali;
- **portabilità** - è possibile importare i propri personaggi con le relative animazioni realizzate con Mixamo in diverse piattaforme, come Unity o Blender;
- **tempi ridotti** - le tempistiche di sviluppo vengono ridotte dal rigging automatico e dalla disponibilità di animazioni già pronte ad essere utilizzate;
- **personalizzazione** - fase di rigging e di animazione personalizzabili;
- **semplicità** - con pochi step è possibile animare facilmente i propri personaggi;

**Inkscape [41].** Un editor di grafica vettoriale open source e multipiattaforma, rilasciato sotto la *GNU General Public License (GPL)* e sviluppato da The Inkscape Team. Il software offre molti strumenti per la realizzazione di illustrazioni, disegni e grafiche vettoriali. Spesso è usato per la creazione di loghi, icone, design di UI, etc. Inoltre, supporta vari formati per l'esportazione, come SVG, PNG, PDF e molti altri. Le ragioni per cui è stato impiegato Inkscape sono le seguenti:

- **gratuito** - non c'è necessità di licenza, anche per fini commerciali;
- **portabilità** - è possibile importare le proprie grafiche su qualsiasi piattaforma compatibile con i diversi formati di esportazione;
- **supporto** - disponibilità di documentazione ufficiale online, vari tutorial e una community molto attiva;
- **semplicità** - curva di apprendimento relativamente rapida, grazie all'interfaccia intuitiva e alla suite di strumenti essenziali;



## CAPITOLO 3

---

### Game Development

---

In questo capitolo saranno discussi gli aspetti di Game Development che hanno guidato la realizzazione della demo di BitStory [7], i cui sorgenti sono disponibili su GitHub. Inoltre, sarà descritto un metodo per valutare le prestazioni di gioco.

#### 3.1 La modellazione 3D

La fase iniziale dello sviluppo è stata dedicata alla creazione dei modelli 3D, in particolare sono stati realizzati il **mondo di gioco** e i **personaggi**. Per tale scopo è stato utilizzato il software **Blender** [39], descritto nella sezione 2.6.

Lo **stile artistico** scelto è chiamato *3D Low Poly* [42], ovvero uno stile che predilige modelli tridimensionali caratterizzati da un numero limitato di poligoni per rappresentare forme e superfici. Ciò si traduce in linee pulite e spigoli netti, garantendo una maggiore efficienza di elaborazione anche da parte di computer non dotati di scheda grafica dedicata. A questo si aggiunge una maggiore espressione creativa nella scelta dei colori, dove il più delle volte sono state sperimentate palette di differente tipologia, favorendo colori vivaci e contrasti forti. Questo permette di donare alle scene che sono state sviluppate un tocco giocoso e di facile lettura.

Nella figura 3.1 è visibile il **rendering** del mondo di gioco nel suo stato finale, realizzato attraverso l'engine Eevee [5]. L'area di gioco è invece definita dalla scrivania e da tutto ciò che essa solleva. Per questa ragione, sono stati creati dei muri invisibili posti ai lati esterni della stessa, in modo che non possa essere varcato il confine.



**Figura 3.1:** Rendering del mondo di gioco realizzato con Eevee [5]

I personaggi andranno ad occupare una generica posizione dell'area di gioco, per cui hanno un'altezza simile al palmrest del portatile. A tale scopo è stata adattata la scala della stanza e dei personaggi. Il rendering di quest'ultimi è visibile in figura 3.2.



**Figura 3.2:** Rendering dei personaggi realizzato con Eevee [5]

Il focus è stato concentrato perlopiù sui vestiti, sulle forme del viso, dei capelli e su alcune parti del corpo, in quanto una volta creato un singolo personaggio (base iniziale) si è reso necessario apportare solo delle leggere modifiche alle forme dei modelli e ai colori delle texture. Mentre, le mani sono realizzate in maniera simile ai personaggi *Lego*. Si noti che i personaggi sono posti nella posizione neutra **A-Pose**, una posizione iniziale che permette di creare le animazioni conservando la simmetria.

Gli strumenti particolarmente utilizzati in **Edit mode**, la principale modalità in cui si eseguono le fasi di modellazione degli oggetti 3D, sono di seguito elencati:

**Move [43]**. Utilizzato per spostare gli oggetti oppure i loro vertici, segmenti o facce, lungo gli assi X, Y o Z.

**Rotate [44]**. Utilizzato per ruotare gli oggetti oppure i loro vertici, segmenti o facce, lungo gli assi X, Y o Z.

**Scale [45]**. Utilizzato per cambiare le proporzioni degli oggetti oppure dei loro vertici, segmenti o facce, lungo gli assi X, Y o Z.

**Measure [46]**. Utilizzato per misurare distanze o angoli all'interno delle scene.

**Extrude Region [47]**. Utilizzato per modificare la geometria degli oggetti, rispetto a un vertice, un segmento o una faccia, lungo gli assi X, Y o Z. Questo strumento è stato utile, ad esempio, per la creazione della scocca della tastiera, ottenuta inserendo un plane nella scena, che successivamente è stato estruso lungo l'asse Z.

**Inset Faces [48]**. Utilizzato per creare dei riquadri intorno alla facce degli oggetti, con dimensioni regolabili. Questo strumento è stato utile, ad esempio, per la realizzazione dei monitor, ottenuti singolarmente inserendo un riquadro nella faccia di un plane estruso. Tale riquadro è stato poi estruso verso l'interno, ottenendo così uno schermo.

**Bevel [49]**. Utilizzato per creare angoli e bordi smussati o arrotondati, alterando la geometria degli oggetti. Questo strumento si è rilevato utile per smussare i bordi di vari oggetti, come le scocche dei computer, del mouse, delle tastiere, etc.

**Loop Cut [50]**. Utilizzato per aggiungere in loop nuove facce alla geometria degli oggetti, mantenendo inalterate le forme originali. Questo strumento è stato utile, ad esempio, per la creazione dei connettori, del vano floppy, del trackpad, etc.

**Knife** [51]. Utilizzato per tagliare la geometria degli oggetti, al fine di creare nuovi vertici o segmenti. Questo strumento è stato utile, ad esempio, per la realizzazione delle pieghe dei fogli posti sulla scrivania e alle pareti.

**Spin** [52]. Utilizzato per duplicare gli oggetti attorno a un'asse e a un punto specifico. Questo strumento è stato utile per la realizzazione delle rotelle poste al di sotto della sedia da scrivania.

Sono stati applicati alcuni **modifier**, ossia degli strumenti che modificano la geometria di un oggetto in modo non distruttivo, lasciando inalterata la sua composizione originale. In particolare, sono stati utilizzati:

**Array** [53]. Utilizzato per creare copie di un oggetto lungo uno o più assi, ad una certa distanza (step) l'uno dall'altro. Applicato per realizzare i **tasti** delle tastiere.

**Mirror** [54]. Utilizzato per creare copie speculari di un oggetto lungo uno o più assi. Può anche consentire la creazione di oggetti simmetrici, semplificando il processo di modellazione. Applicato per realizzare le **gambe** della scrivania e del tavolino, i braccioli della sedia e per modellare i **corpi** dei personaggi in maniera simmetrica.

**Solidify** [55]. Utilizzato per aggiungere spessore agli oggetti. Applicato per realizzare vari oggetti, come: i **tubi**, la **tazza**, i **fili** di mouse e tastiere, etc.

Il **texturing** [56] è stato impostato realizzando semplici materiali di default con diversi colori di base [57], applicati alle facce degli oggetti. Mentre, le texture (eg. schermo tv) sono state create sia con materiali di default che emissivi [58], ossia in grado di emettere luce. Il colore di questi materiali è dato da specifiche immagini.

Per la gestione delle collisioni [59] sono stati creati i **collisions** [60], mediante lo strumento *Collider-Tools* [61], un add-on installabile in Blender. Questo strumento permette di creare mesh sovrapposte agli oggetti originali, le quali possono assumere forme primitive o la forma dei modelli.

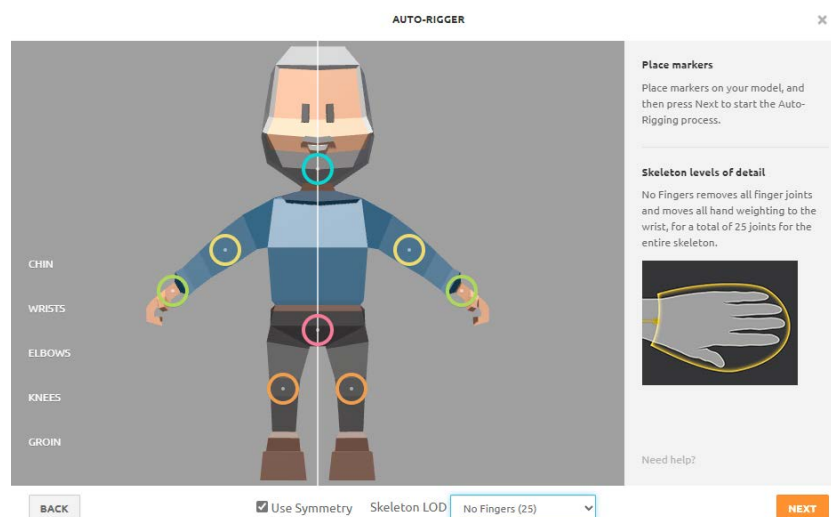
In penultima fase, sono state **ricalcolate le normali** [62], al fine di evitare problemi di rendering. Infine, sia il mondo di gioco che i personaggi sono stati esportati in formato **FBX**, in modo da poterli inserire in un nuovo progetto Unity [37], oltre che, nel caso dei personaggi, importarli all'interno della piattaforma Mixamo [40].

## 3.2 Le animazioni dei personaggi

Nella seconda fase dello sviluppo sono state scelte le animazioni dei personaggi dalla piattaforma **Mixamo** [40], descritta nella sezione 2.6. Esse includono: (i) un'animazione di **Idle**, eseguita da tutti i personaggi, incluso l'avatar del giocatore quando non è in movimento; (ii) animazioni per il movimento dell'avatar in **Avanti**, **Indietro**, **Destra**, **Sinistra**, **Diagonale Indietro Destra** e **Diagonale Indietro Sinistra**.

Eseguita la registrazione alla piattaforma, si è proseguito importando i personaggi realizzati, così da poter applicare le animazioni. Inoltre, dato che i personaggi erano sprovvisti di uno **scheletro** [6], è stata richiesta l'esecuzione del **rigging automatico**, in cui sono state definite le varie parti del corpo dei singoli modelli, ossia: il mento, i gomiti, i polsi, l'inguine e le ginocchia. Mentre, per le mani è stata selezionata l'opzione "No Fingers". L'esecuzione di questa fase è visibile in figura 3.3. In questo modo, l'algoritmo di Mixamo ha costruito per ogni personaggio uno scheletro.

Successivamente, sono state selezionate le animazioni, personalizzando vari parametri in base al personaggio, come: la velocità dell'animazione, la larghezza delle braccia durante l'animazione, il movimento sul posto o nello spazio, etc. Infine, ogni animazione ha richiesto il download del personaggio rielaborato, ossia con scheletro e clip applicata, scegliendo come formato **FBX for Unity** a **60 frame per secondo**.



**Figura 3.3:** Auto-Rigger di un personaggio effettuato in Mixamo [6]

### 3.3 Lo sviluppo in Unity

L'ultima fase dello sviluppo ha previsto la realizzazione del gioco mediante l'editor di **Unity** [37], descritto nella sezione 2.6. BitStory è stato concepito come gioco esclusivo per computer e dunque i controlli sono orientati verso l'utilizzo di mouse e tastiera. Inoltre, la demo è sviluppata per adattarsi alle proporzioni degli schermi a 16:9, ed è stata resa compatibile con diverse risoluzioni.

Il gioco complessivamente presenta varie **scene**, realizzate in quest'ordine:

1. il mondo di gioco;
2. il menu principale;
3. il minigioco di Ada Lovelace;
4. i mockup riservati ad altri minigiochi;

#### 3.3.1 La scena del mondo di gioco

La scena del mondo di gioco è stata realizzata importando la stanza realizzata in precedenza insieme a tutti i personaggi rielaborati in Mixamo. Inoltre, durante l'importazione è stato necessario estrarre i relativi materiali applicati ai modelli. Fatto ciò, si è provveduto ad aggiungere questi oggetti alla scena.

Sono stati poi aggiunti i **Mesh Collider** [63] ai vari oggetti statici presenti nell'area di gioco, sfruttando i collisions precedentemente applicati.

In seguito, sono stati inseriti nell'area di gioco (la scrivania) dei cubi sottili che rappresentano delle note da collezione. A questi elementi e ai personaggi, escluso l'avatar del giocatore, sono stati aggiunti i componenti **Box Collider** [64] e **Rigidbody** [65].

Le animazioni dei personaggi sono gestite mediante il componente **Animator Controller** [66]. Nel caso dell'avatar del giocatore si è realizzato un solo stato mediante un **Blendtree 2D** [67], il quale ha permesso di combinare le varie animazioni di movimento, inclusa quella di Idle, in modo fluido e in base alle azioni del personaggio. Gli altri personaggi sono gestiti da un semplice stato con associata la clip di Idle.

Il movimento dell'avatar del giocatore è gestito mediante il componente **Character Controller** [68], il quale ha permesso di lavorare efficacemente ai movimenti, alla gravità e alla collisione con altri oggetti dell'area di gioco. Non a caso, questo componente è particolarmente indicato per la gestione dei personaggi controllati dal giocatore nei giochi in prima o terza persona. Infatti, BitStory su questo aspetto si presenta come gioco in **terza persona**, dovuto al fatto che la **Camera** [69] aggiunta alla scena è posta poco distante dal personaggio, seguendolo durante i suoi movimenti.

La gestione delle collisioni abbinata all'utilizzo dei **Tag** [70] e dei **Layer** [71], permette di stabilire con cosa effettivamente l'avatar entra in collisione durante i movimenti. In particolare, è stato assegnato il **Layer Characters** ai personaggi e ad ognuno un Tag uguale al nome dell'individuo rappresentato. Mentre, nel caso delle note, il **Layer Notes**, i cui Tag sono relativi alle note stesse. Ciò permette di attivare nel caso dei personaggi il **pannello dei dialoghi**, nell'altro caso il **pannello delle note**.

Inoltre, è stato implementato il **salvataggio dei progressi**, utile per stabilire, ad esempio, quali note caricare nell'area di gioco in varie partite, a seconda dello stato raccolto/non raccolto. Oltre che per conservare i punteggi, risultati raggiunti, etc. Queste e molte altre funzionalità, come il movimento del personaggio, il movimento della Camera, etc., sono state realizzate mediante scripting in C#.

L'interfaccia utente (UI) del mondo di gioco è realizzata inserendo nella scena il componente **Canvas** [72], in cui è stata impostata la **Scale Mode** su *Scale With Screen Size*. Fatto ciò, sono stati aggiunti al Canvas diversi elementi:

**Progress Panel.** Un pannello sempre attivo che mostrerà i progressi fatti durante il gioco, visibile nelle figure 3.4. Esso contiene: (i) un testo per i **bit**, ossia i punti raccolti durante il gioco; (ii) un testo per le **note** raccolte, rispetto alla quantità totali nascoste; (iii) un testo per le **medaglie** ottenute rispetto a quelle totali.

**Notes Panel.** Un pannello per la nota raccolta. Esso contiene un testo per il titolo e un testo per la descrizione del concetto. Visibile nella figura 3.4b.

**Dialog panel.** Un pannello per i dialoghi. Include un testo per i nomi dei personaggi, un testo per le battute e pulsanti per il cambio scena. Visibile nelle figure 3.4c e 3.4d.

La scelta del **font** è orientata rispetto allo stile grafico utilizzato in precedenza, per cui l'idea è ricaduta su un font simile a quello utilizzato da *Minecraft Education* [13]. Il font è reso disponibile gratuitamente su GitHub dall'utente *IdreesInc* [73].



(a) Bit, note e medaglie visibili in alto



(b) Pannello che mostra la nota raccolta



(c) Pannello di dialogo con Ada Lovelace



(d) Pulsanti per accettare o rifiutare la sfida

**Figura 3.4:** Elementi della UI del mondo di gioco inclusi nella demo [7]

### 3.3.2 La scena del menu principale

La scena del menu principale è stata realizzata a partire da un duplicato della scena del mondo di gioco, in quanto si è voluto fornire una prospettiva da lontano di quest'ultimo. A tale scopo sono stati disabilitati vari script ed eliminati vari oggetti della scena originale. Quindi è stata aggiunta una nuova Camera, in posizione fissa.

È stato inserito all'interno della scena un **Empty GameObject** che non sarà distrutto durante i cambi di scena [74]. Questo oggetto è stato utilizzato per la gestione di musiche e suoni, aggiungendo ad esso dei componenti **Audio Source** [75], in grado di riprodurre **Audio Clip** [76]: dati audio dei file .mp3, .wav, etc. L'altro componente necessario è l'**Audio Listener** [77], presente di default in ogni Camera di scena.



L'interfaccia utente (UI) del menu principale, mantiene il **Progress Panel** della scena del mondo di gioco, oltre che il font e i colori. In più aggiunge i seguenti elementi:

**Pulsante Delete Data.** Elimina i salvataggi dei progressi di gioco. Si veda figura 3.5.

**Pulsante Control Info.** Abilita un pannello per i comandi di gioco. Si veda figura 3.5.

**Pulsante Start Game.** Avvicina la Camera di scena alla Camera dell'avatar presente nell'area di gioco. Quando la Camera attuale sarà sufficientemente vicina alla Camera del giocatore, avverrà il cambio di scena col mondo di gioco. Si veda figura 3.5.

**Logo BitStory.** Realizzato mediante il software Inkscape [41], descritto nella sezione 2.6. Inserito nel Canvas in qualità di **sprite 2D**. Visibile in figura 3.5a.

**Pannello Control Info.** Un pannello che contiene vari testi che mostrano i comandi di gioco e un pulsante per chiudere il pannello. Visibile in figura 3.5b.



(a) Pulsanti, logo e pannello dei progressi



(b) Il pannello dei controlli in primo piano

**Figura 3.5:** Elementi della UI del menu principale inclusi nella demo [7]

### 3.3.3 La scena del minigioco di Ada Lovelace

La scena del minigioco di Ada Lovelace, visibile in figura 3.6, è costruita utilizzando oggetti primitivi di Unity: **cubi** di varie dimensioni che formano le caselle di un percorso, una **sfera** controllata dal giocatore e dei **quad**. I **Mesh Renderer** [78] di quest'ultimi includono materiali con immagini, i quali simulano le texture al di sopra delle caselle grandi. La Camera invece è stata posta in posizione fissa, ruotata di 90° lungo l'asse X, in modo che punti dall'alto verso il basso.

Il movimento della sfera è gestito utilizzando la fisica, attraverso i componenti **Rigidbody** [65] e **Sphere Collider** [79]. Inoltre è stato annullato l'effetto di attrito e di rimbalzo mediante l'aggiunta di un **Physic Material** [80] allo Sphere Collider. Tutte le caselle del percorso hanno le componenti **Rigidbody** e **Box Collider** [64]. Mentre, alle caselle più piccole sono stati aggiunti degli oggetti vuoti come figli, spostati leggermente più in alto e con il solo Box Collider duplicato dall'oggetto padre. Questi oggetti figli, fungono da **Trigger** [64], non reagendo alla fisica e necessari per gestire l'evento in cui la sfera esce dalla collisione con una casella piccola, in modo che quest'ultima possa cadere poco dopo. Quando si ricevono dei bonus sono istanziati dei **Particle System** [81]. Al termine del gioco sono salvati i risultati ottenuti.

L'interfaccia utente (UI) del minigioco di Ada Lovelace, in modo simile alle altre scene, è realizzata aggiungendo al Canvas [72] i seguenti elementi:

**Progress Panel.** Un pannello che include vari testi: tempo rimasto, passi effettuati, high score e limite di passi. Oltre che uno sprite 2D per la medaglia (trasparente se non ottenuta, colorata altrimenti). Si veda la figura 3.6.

**Start Game Panel.** Un pannello iniziale che include testi per le regole di gioco, e pulsanti per avviare la sfida o ritornare al mondo di gioco. Si veda la figura 3.6a.

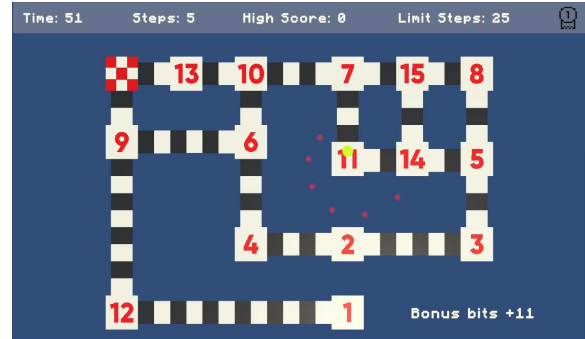
**Bonus.** Un testo mostrato quando la sfera raggiunge le caselle con i numeri, ottenendo bonus sul tempo o sui punti finali. Si veda la figura 3.6b.

**Game Over Panel.** Un pannello per il Game Over. Include un testo di Game Over e pulsanti per ricominciare la sfida o ritornare al mondo di gioco. Si veda la figura 3.6c.

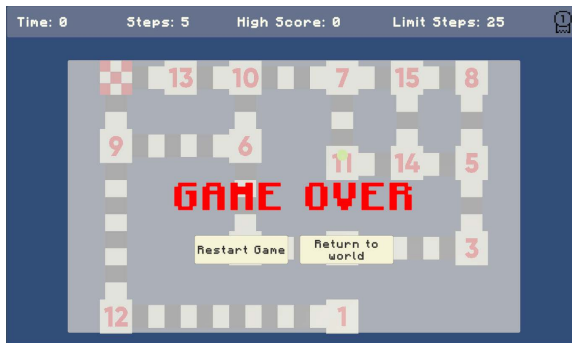
**Success Panel.** Un pannello per lo stato di vittoria. Include testi per la condizione di vincita (con medaglia, senza medaglia, etc.) e per i punti guadagnati, nonché pulsanti per ricominciare la sfida o ritornare al mondo di gioco. Si veda la figura 3.6d.



(a) Pannello che illustra le regole del minigioco



(b) Bonus ottenuti durante l'esecuzione



(c) Pannello che mostra lo stato di Game Over



(d) Pannello che mostra lo stato di vittoria

**Figura 3.6:** Elementi della UI del minigioco di Ada Lovelace inclusi nella demo [7]

Gli sprite 2D sono stati realizzati con il supporto della piattaforma **Flaticon** [82].

### 3.3.4 Le scene di mockup

Le scene di mockup sono state progettate al fine di fornire un'idea completa del gioco. Sebbene siano ancora in fase di sviluppo, in quanto mancano di una completa UI e del comparto di scripting, esse sono costruite per essere facilmente implementate. In tutto, sono presenti sette scene di mockup di minigiochi sia 3D che 2D.

Durante la realizzazione sono stati utilizzati oggetti primitivi di Unity e sprite 2D creati con **Inkscape** [41] e con il supporto della piattaforma **Flaticon** [82].

Inoltre, la scena di un minigioco ha richiesto la costruzione di un labirinto che è stato autogenerato attraverso un tool ottenuto da **Unity Asset Store** [83], e successivamente modificato in base alle esigenze e allo scopo del minigioco.

### 3.3.5 Le musiche e i suoni

Il comparto audio di BitStory include musiche e suoni gestiti attraverso l'**Empty GameObject** aggiunto alla scena del menu principale, discusso nella sezione 3.3.2. I suoni che sono stati utilizzati sono liberamente disponibili con licenza *Creative Commons Zero (CC0)* sulla piattaforma **Freesound** [84] e includono:

- **button click** - un suono riprodotto alla pressione dei pulsanti della UI oppure quando si avanza nei dialoghi nella scena del mondo di gioco [85];
- **page turn** - un suono riprodotto quando si raccoglie una nota nella scena del mondo di gioco [86];
- **step** - un suono riprodotto in loop durante i movimenti dell'avatar nella scena del mondo di gioco [87];
- **start game** - un suono riprodotto alla pressione del pulsante start game nella scena del menu principale [88];
- **bonus** - un suono riprodotto quando si ottengono dei bonus nella scena del minigioco di Ada Lovelace [89];
- **game win** - un suono riprodotto nella scena del minigioco di Ada Lovelace quando esso termina nello stato di successo [90];
- **game over** - un suono riprodotto nella scena del minigioco di Ada Lovelace quando esso termina nello stato di Game Over [91];
- **world music** - una traccia musicale riprodotta in loop nella scena del menu principale e che continua la sua esecuzione al passaggio di scena con il mondo di gioco [92];
- **minigame music** - una traccia musicale riprodotta in loop nella scena del minigioco di Ada Lovelace [93];

## 3.4 Metodo di valutazione delle prestazioni

Lo scopo di questa sezione è di definire il metodo attraverso il quale saranno valutate le prestazioni della demo di BitStory durante la sua esecuzione. A tal fine, saranno sottoposti al test diversi calcolatori, con varie componenti, più o meno recenti, i cui sistemi operativi saranno principalmente Windows 10 e Windows 11.

La valutazione si baserà sul verificare il **numero medio di frame per secondo (FPS)**, ovvero il numero medio di immagini in grado di essere elaborate e mostrate a schermo ad ogni secondo. A tal fine, sarà eseguito un benchmark attraverso l'utilizzo di **Fraps** [94], riportando i valori minimi, massimi e medi degli FPS ottenuti.

Il benchmark sarà eseguito su ogni specifico elaboratore riproducendo le medesime condizioni di gioco, ma soprattutto, facendo attenzione a chiudere tutti i task non necessari, in modo tale da avere risultati quanto più veritieri possibile.

Nella selezione dei calcolatori da sottoporre al test saranno inclusi sia computer dotati di scheda grafica dedicata, che non <sup>1</sup>, in modo da verificare quanto incida sulle prestazioni la presenza o meno di tale componente.

C'è da precisare che essendo la demo progettata per schermi a 16:9, saranno inclusi nel test solo monitor con tale aspect ratio, con una risoluzione di 1080p e un refresh rate di 60Hz. In questo modo il gioco potrà essere eseguito a tale risoluzione ed essere in grado di raggiungere i 60 FPS. Il benchmark sarà effettuato impostando il **Quality Level** su *Ultra Quality* e il **VSync Count** su *Every V Blank*, entrambi presenti di default nelle impostazioni del progetto Unity [95].

---

<sup>1</sup>i computer moderni non dotati di scheda grafica dedicata, presentano processori con scheda grafica integrata, utile per elaborazioni grafiche di base

## CAPITOLO 4

---

### BitStory

---

In questo capitolo saranno illustrati gli elementi di Game Design di BitStory [7]: mondo di gioco, personaggi, minigiochi, etc. Inoltre, sarà valutato il potere motivazionale del Serious Game basandosi sulle ricerche svolte nel capitolo 2, discusse le soluzioni alle limitazioni dei lavori correlati e mostrati i risultati delle prestazioni di gioco.

### 4.1 Introduzione al gioco

Il titolo del Serious Game nasce dall'unione del termine *Bit*, l'unità di misura di base dell'informazione, e dal termine *Story*, per delineare la storia dell'informatica.

Lo **scopo del gioco** è immergere il giocatore in un mondo ambientato nel passato, dove esplorare e conoscere alcuni dei **personaggi** che hanno delineato le origini di questo settore. Durante l'esplorazione, il giocatore avrà modo di imparare molte cose sulla computer science, mediante i **dialoghi** e le **note** collezionabili. Inoltre, potrà affrontare vari **minigiochi** proposti dai personaggi, che se superati singolarmente nella maniera corretta, ma soprattutto ottima, conferiranno al giocatore una **medaglia**.

Le sfide stimoleranno differenti aree cognitive, oltre che a definire dei chiari collegamenti con alcuni dei problemi o argomenti noti della computer science. Infine, si avrà modo di incrementare il **punteggio globale** attraverso i progressi compiuti nel gioco.

**Il mondo di gioco.** Il mondo di gioco, visibile nella figura 3.1, nel suo stato finale richiama l'estetica di una stanza da hacker degli anni '80 o '90.

Gli elementi che principalmente saltano all'occhio, sono situati sulla scrivania, la quale presenta computer e monitor vintage tipici di quel periodo, oltre che tastiere meccaniche, mouse a due tasti, etc., i cui design si ispirano alle tecnologie *Olivetti* e *IBM* dell'epoca. Mentre, le decorazioni date dalle texture, come la locandina di *Ritorno al futuro*, il poster di *Donkey Kong* e un frame di *Mickey Mouse* sullo schermo del televisore, favoriscono a creare un ambiente più inclusivo per tutte le età.

La rappresentazione di questo scenario permette di arricchire il **tema di gioco** [96], vista la scala ridotta dei personaggi e di altri elementi inseriti nell'area della scrivania. Quest'ultima idea, si può ricondurre all'immaginazione dei bambini, la quale ben si mostra durante il gioco creativo, riferendosi in parte allo *small world toys*: l'attività in cui essi creano mondi in miniatura con personaggi dalle piccole dimensioni [97].

**I personaggi.** Durante la selezione dei personaggi è stata attuata un'approfondita analisi delle diverse aree dell'informatica da ricoprire, in moda da avere sfide variegata e a tema rispetto ad ogni individuo. Seguendo l'ordine della figura 3.2, da sinistra verso destra, i personaggi storici scelti sono i seguenti:

1. **Ada Lovelace** - matematica inglese vissuta durante la metà del XIX secolo, celebre per il suo apporto alla *macchina analitica* di *Charles Babbage*. Ricordata come una delle prime programmatrici di computer al mondo [98].
2. **Alan Turing** - matematico, logico e crittografo inglese, vissuto durante la metà del XX secolo. Celebre per aver ideato il modello *macchina di Turing* e per i suoi contributi nel campo della crittoanalisi e dell'intelligenza artificiale [99].
3. **Horst Feistel** - crittografo tedesco-americano vissuto durante il XX secolo, il quale ha lavorato all'IBM e alla realizzazione di diversi cifrari, tra cui *Lucifer* ed il suo successore, ovvero il *DES* [100].
4. **Dennis Ritchie** - informatico statunitense vissuto tra la metà del XX secolo e gli inizi del XXI secolo. Precursore dell'informatica moderna e celebre per aver inventato il *linguaggio C* e il sistema operativo *Unix* [101].

5. **George Boole** - matematico e logico britannico vissuto durante la metà del XIX secolo. Noto per aver inventato il sistema di *algebra booleana* e aver gettato le basi per la logica matematica [102].
6. **Claude Shannon** - ingegnere e matematico statunitense vissuto durante il XX e XXI secolo, ricordato come il *padre della teoria dell'informazione* [103].
7. **John McCarthy** - informatico americano vissuto tra il XX e il XXI secolo, noto per aver inventato il termine *intelligenza artificiale* e il *linguaggio Lisp* [104].
8. **Tim Berners-Lee** - informatico inglese nato a metà del XX secolo, celebre per l'invenzione del *World Wide Web* [105].

Il giocatore vestirà il ruolo di un **bambino**, l'ultimo personaggio a destra della figura 3.2. Questo personaggio riflette la curiosità e la voglia di esplorare tipici dell'infanzia. Infatti, queste qualità serviranno a garantire un'esperienza di gioco stimolante [8].

**I collezionabili.** All'interno del gioco sono presenti alcuni collezionabili: (i) le **note**, ossia brevi appunti sparsi per il mondo di gioco, mediante i quali saranno descritti alcuni concetti legati all'informatica, come *algoritmo*, *sistema operativo*, *bit*, *malware*, etc; (ii) le **medaglie**, ossia premi ottenibili attraverso un perfetto superamento delle sfide proposte dai diversi personaggi. La presenza di premi può indurre il giocatore a giocare meglio, grazie al senso di appagamento dato da questo sistema [22]; (iii) i **bits**, ossia il punteggio globale che tiene conto dei progressi fatti nel gioco. Sono ottenibili superando le sfide o raccogliendo le note.

**Le azioni del mondo di gioco.** Le azioni eseguibili dal giocatore all'interno del mondo di gioco, sono le seguenti: **(i)** muovere l'avatar in direzioni diverse: dx, sx, avanti, indietro, diagonale avanti a sx, diagonale avanti a dx, diagonale indietro a sx e diagonale indietro a dx; **(ii)** ruotare l'avatar su se stesso ottenendo una visuale completa a 360 gradi intorno ad esso; **(iii)** muovere la visuale verso l'alto o verso il basso; **(iv)** raccogliere e collezionare le note; **(v)** accumulare i bits (punti); **(vi)** attivare i dialoghi con i vari personaggi; **(vii)** accedere ai differenti minigiochi proposti dai personaggi; **(viii)** salvare i progressi di gioco e tornare al menu principale;



**Le regole e i controlli del mondo di gioco.** L'elenco delle regole e i controlli imposti per la corretta esecuzione delle azioni precedentemente descritte, sono i seguenti:

1. **Movimento avatar** - utilizzare i tasti *W, A, S, D* oppure i tasti *Freccia Su, Freccia Giù, Freccia Sx, Freccia Dx*.
2. **Rotazione avatar** - trascinare il *mouse a destra o a sinistra*.
3. **Rotazione visuale** - trascinare il *mouse a destra, a sinistra, in alto o in basso*.
4. **Raccogliere note** - avvicinarsi a una nota e premere il tasto *C*. Premere nuovamente per riporla.
5. **Accumulare bits (punti)** - raccogliere le note. Ogni nota vale 10 bits.
6. **Attivare un dialogo** - avvicinarsi a un personaggio e premere il tasto *E*. Premere nuovamente per avanzare nelle battute.
7. **Accedere ai minigiochi** - attendere la fine di un dialogo e premere il pulsante *Si*. In alternativa rifiutare con il pulsante *No*. In entrambi i casi usare il *mouse*.
8. **Salvare i progressi** - premere il tasto *ESC*, il quale riporterà al menu principale.

**La storia.** Il gioco non ha una vera e propria trama, in quanto si è deciso di lasciare al giocatore la libertà di immaginare la sua storia attraverso l'ambientazione, i personaggi e i dialoghi. Il dialogo con **Ada Lovelace**, il primo personaggio incontrato, sarà importante per comprendere il contesto del gioco. Di seguito sono riportate le **battute di dialogo**, le cui informazioni sono estrapolate dal libro "*Giants of Computing*" [98]:

**ADA LOVELACE:** Ciao, sono Ada Lovelace, conosciuta anche come la prima programmatrice della storia dell'informatica. È un piacere conoscerti!

**ADA LOVELACE:** Immaginami come una "maga del computer", utilizzo numeri e istruzioni per fargli fare tutto ciò che desidero!

**ADA LOVELACE:** Durante i miei studi ho lavorato alla macchina analitica di Charles Babbage: il primo prototipo di computer meccanico.

**ADA LOVELACE:** Con essa ho realizzato un algoritmo per generare i numeri di Bernoulli: le istruzioni magiche di cui ti parlavo prima!

ADA LOVELACE: Un algoritmo è una sequenza di istruzioni eseguite da un computer per risolvere problemi o compiti specifici.

ADA LOVELACE: Magico vero?

ADA LOVELACE: In questo mondo, potrai divertirti insieme a me e a tanti altri personaggi illustri della storia dell'informatica!

ADA LOVELACE: Avrai l'opportunità di imparare e scoprire tante cose che forse ancora non conosci... consideralo un viaggio educativo nel passato.

ADA LOVELACE: Ora ti lancerò una sfida interessante. Se la superi egregiamente, riceverai una fantastica medaglia!

ADA LOVELACE: La sfida consiste nel trovare il percorso più breve da un punto A a un punto B.

ADA LOVELACE: Questa sfida è nota come "la ricerca della path di cammino minimo in un grafo".

ADA LOVELACE: Un grafo è un insieme di nodi collegati tra loro da archi.

ADA LOVELACE: Le azioni che compirai per completare questa sfida saranno il tuo algoritmo. Ma ora focalizzati sulla sfida.

ADA LOVELACE: Allora che ne dici? Cominciamo?

I dialoghi con gli altri personaggi seguono questo stesso schema di conversazione: **presentazione**, **spiegazione di concetti** in parole semplici, e infine **guida verso la sfida**. Ogni personaggio avrà un totale di 10-15 battute di dialogo ciascuno.

## 4.2 I minigiochi

In questa sezione saranno descritti i minigiochi che sono stati progettati per BitStory.

### Path di cammino minimo di un grafo dal nodo $s$ al nodo $t$ (Ada Lovelace)

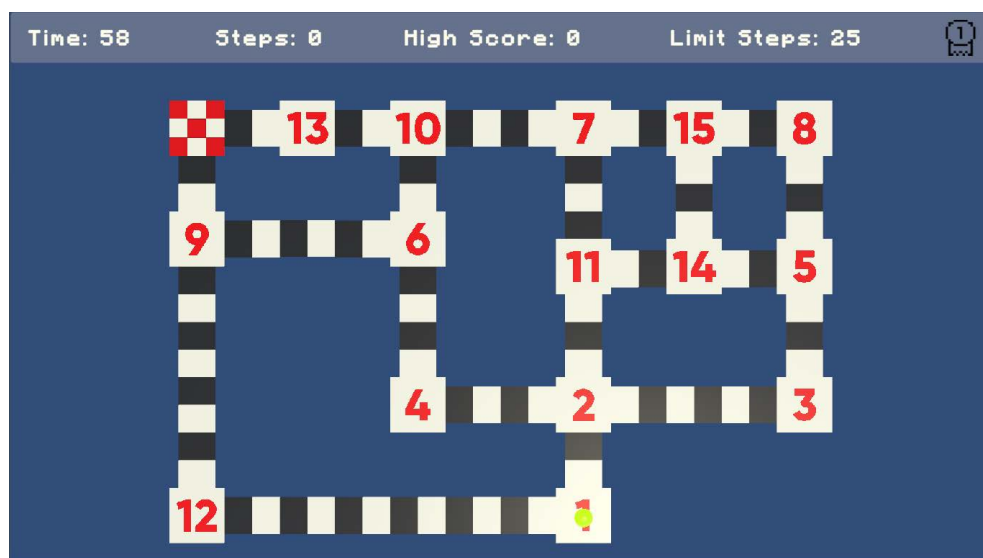
In questo gioco, ispirato al noto problema sui grafi [106], l'obiettivo è guidare una pallina dal **nodo di partenza** al **nodo del traguardo**, cercando di eseguire il **minor** n° possibile di **passi**, al fine di trovare il percorso più breve. Si veda la figura 4.1.

I nodi del grafo sono rappresentati attraverso delle **caselle grandi** con dei numeri, nonché attraverso la casella contrassegnata dalla bandiera a scacchi. Gli archi sono

invece raffigurati dalle **caselle piccole** bianche e nere, disposte a scacchiera. Il nodo di partenza è la casella dove risiede la pallina (nodo 1), mentre il nodo del traguardo è la casella marcata con la bandiera a scacchi.

Ogni casella piccola corrisponde ad un **singolo passo** eseguito dalla pallina.

Supponiamo ad **esempio** che la pallina venga spostata dal nodo 1 al nodo 3. In questo caso, il percorso richiederà 8 passi complessivi: 2 passi dal nodo 1 al nodo 2 e successivamente 6 passi dal nodo 2 al nodo 3.



**Figura 4.1:** Il minigioco di Ada Lovelace

Per evitare il **Game Over**, serve fare attenzione al **tempo** (60 secondi) e al **limite di passi** (25 passi). Inoltre, è fondamentale non far cadere la pallina nel vuoto. Ciò può accadere per due ragioni: (1) in via del tutto intenzionale da parte del giocatore e (2) poiché le caselle piccole (archi) una volta attraversate cederanno.

Il giocatore **vince** e ottiene dei **bits** (punti) quando la pallina sarà giunta al nodo del traguardo entro il tempo limite e senza superare il n° di passi consentiti.

È stata introdotta la possibilità di **rigiocare** la sfida, per incentivare il giocatore a impegnarsi e a trovare il percorso più breve, permettendogli di ottenere una **medaglia** in premio. Ad ogni tentativo l'**high score** sarà aggiornato in base ai passi effettuati. Inoltre, ogni qual volta che la pallina passerà su un nodo del grafo, potrà ottenere come **bonus** un certo n° di bits (punti) o secondi da sommare al tempo.

Il **movimento** della pallina è controllato utilizzando i tasti *W*, *A*, *S*, *D*, oppure equivalentemente tramite i tasti *Freccia Su*, *Freccia Giù*, *Freccia Sx*, *Freccia Dx*.

### Labirinto dell'informazione (Claude Shannon)

In questo gioco, ispirato dalla **teoria dei segnali** [107], lo scopo è guidare un cubo dal punto di partenza sino al punto d'uscita (casella con bandiera a scacchi) del labirinto. Si veda la figura 4.2. Il cubo rappresenta il segnale che si propaga e fa viaggiare le **informazioni**, mentre il labirinto identifica il canale di comunicazione.

Man mano che il cubo avanzerà verso l'uscita, parte dell'informazione sarà persa, correlando dunque il concetto di segnale che si attenua con l'andamento dell'oggetto. Il giocatore avrà inizialmente **25 unità di informazione** e ad ogni **passo**, costituito da una singola casella del labirinto, verrà persa 1 unità di informazione.

Lungo il percorso il giocatore avrà modo di **ripristinare** il valore iniziale delle unità di informazione perse recandosi sulle caselle marcate dal simbolo di **segnale rosso**. Quest'ultima meccanica è legata al concetto di rigenerazione del segnale.

Inoltre, si otterranno **bonus** sul tempo o bits (punti) raccogliendo le monetine rosse.



**Figura 4.2:** Il minigioco di Claude Shannon

Per evitare il **Game Over**, servirà fare attenzione al **tempo** (60 secondi) e a **non** perdere tutte le unità di informazione. Il giocatore **vince** e ottiene dei **bits** (punti), se arriverà all'uscita rispettando le precedenti condizioni. L'impegno sarà stimolato da una **medaglia** ottenibile in caso si vinca con un n° di unità di informazione  $\geq 20$ . L'**high score** sarà aggiornato, ad ogni tentativo, in base alle unità di informazione giunte al traguardo. Il **movimento** del cubo sarà gestito mediante i tasti *W, A, S, D*, oppure equivalentemente con i tasti *Freccia Su, Freccia Giù, Freccia Sx, Freccia Dx*.

### Cifrario di Cesare (Horst Feistel)

Nel contesto di questo gioco, ispirato alla **crittografia classica**, l'obiettivo è decifrare un messaggio seguendo le regole del noto *Cifrario di Cesare* [108]. Si veda la figura 4.3. Il gioco presenta **due alfabeti**: quello standard composto dai caratteri da A a Z e un secondo alfabeto cifrato. Quest'ultimo è ottenuto spostando ogni lettera dell'alfabeto standard di un numero fisso di posizioni verso destra.

Il giocatore dovrà dunque **inserire** il testo in chiaro del messaggio cifrato nell'apposita casella di testo. Per fare ciò bisognerà considerare la chiave, ovvero il numero di spostamenti subito da ciascuna lettera, e in più comprendere il significato delle parole decifrate. Infatti, il messaggio cifrato conterrà dei *question mark* (?) che faranno riflettere sul significato logico e sintattico delle parole, e nonché della frase.



**Figura 4.3:** Il minigioco di Horst Feistel

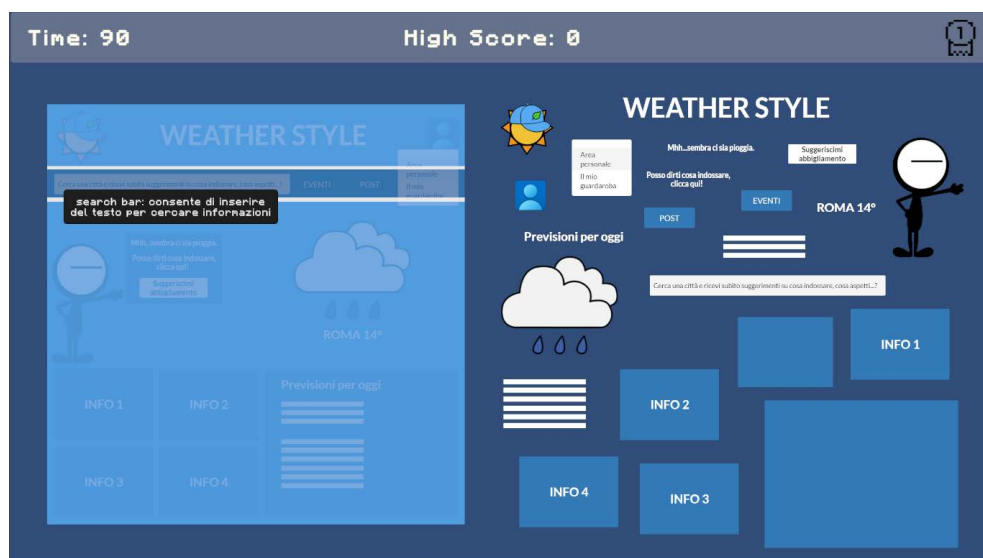
Per evitare il **Game Over**, bisognerà completare la sfida entro 4 minuti e restare nel range dei 3 tentativi. Quest'ultimi verranno conteggiati ogni volta viene premuto il pulsante **Invio**. La **vittoria** e l'ottenimento di **bits** (punti) saranno garantiti dalla correttezza del messaggio in chiaro.

Quando il giocatore digiterà una parola esatta saranno dati in **bonus** dei bits (punti). Ogni nuova partita modificherà casualmente la chiave di cifratura e aggiornerà l'**high score** in base al miglior tempo impiegato per superare la sfida.

Sarà conferita una **medaglia** se si decifrerà il messaggio entro massimo 3 minuti.

### HTML puzzle (Tim Berners-Lee)

Questo gioco nasce con l'obiettivo di trasmettere al giocatore la conoscenza di alcuni degli elementi che costituiscono una **pagina web**. Lo scopo di questa sfida è dunque ricostruire una pagina HTML [109] sfruttando gli elementi presenti nell'area di gioco. Si veda la figura 4.4. Il giocatore dovrà **trascinare** (drag and drop) tali componenti verso i corrispettivi di colore trasparente. Inoltre, quando sarà spostato il mouse su un componente trasparente della pagina, apparirà un **tooltip**, che fornirà delle semplici informazioni a scopo educativo su tale elemento.



**Figura 4.4:** Il minigioco di Tim Berners-Lee

Il giocatore avrà a disposizione 90 secondi e, in caso questi si azzerino, si entrerà nello stato di **Game Over**. La sfida è ritenuta **superata** quando tutti gli elementi della pagina saranno disposti correttamente. In questo modo si otterranno dei **bits** (punti). Nel caso in cui si riesca a terminare il minigioco entro massimo 60 secondi, sarà conferito come premio una **medaglia**.

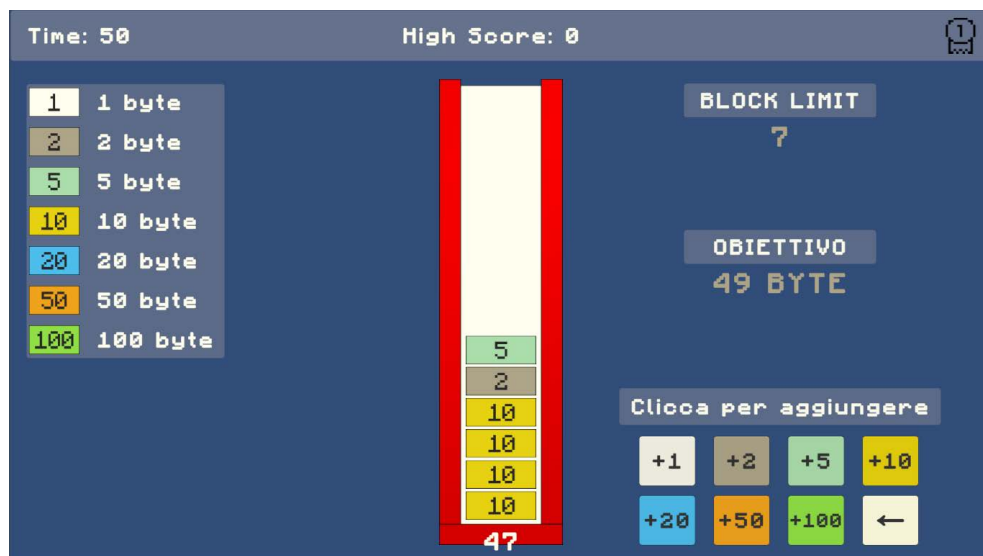
Ogni nuovo tentativo aggiornerà l'**high score** con il maggior numero di elementi posizionati all'interno della pagina.

Durante il gioco saranno casualmente forniti in **bonus** dei bits (punti), ogni qual volta sarà disposto correttamente un elemento nella pagina.

### Stack math (Dennis Ritchie)

In questo gioco, che trae ispirazione da alcuni concetti relativi alla **gestione della memoria** dei sistemi operativi [110], l'obiettivo è cercare di ridurre al minimo il n° di blocchi da allocare in memoria. Per semplificare, la memoria è rappresentata mediante uno **stack**, mentre i **blocchi** da alcuni valori numerici non troppo complessi, onde evitare laboriosi calcoli con le eventuali potenze di 2. Si veda la figura 4.5.

Il giocatore dovrà dunque raggiungere una soglia casuale di **byte** (eg. 49 byte), sommando i valori dei vari blocchi, i quali saranno aggiunti allo stack premendo gli appositi pulsanti (+1, +2, ...). Mediante il pulsante **Backspace** sarà possibile rimuovere i blocchi inseriti. Inoltre, è presente un **limite di blocchi** da utilizzare. Esso è pari al n° minimo di blocchi necessari più una costante, in modo da agevolare la sfida.



**Figura 4.5:** Il minigioco di Dennis Ritchie

Per evitare lo stato di **Game Over** bisognerà terminare la sfida entro 60 secondi e non superare il limite di blocchi. Il giocatore **vince** e ottiene dei **bits** (punti) quando avrà raggiunto la soglia dei byte indicati senza sforare il limite.

Se verranno utilizzati il n° minimo di blocchi e il tempo è superiore ai 30 secondi, verrà conferito come premio una **medaglia**.

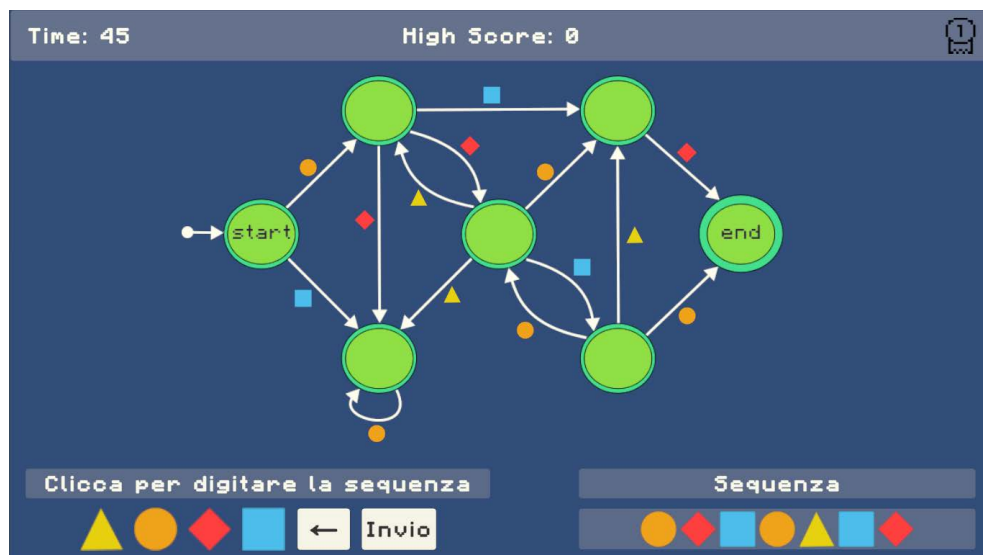
Ogni tentativo superato aggiornerà l'**high score** con il miglior tempo impiegato.

Si otterranno in **bonus** dei bits (punti), in maniera casuale quando saranno premuti i pulsanti per aggiungere i blocchi.

### Automa Geometrico (Alan Turing)

Nell'ambito di questo gioco, che si ispira alla **teoria degli automi** [111], lo scopo è cercare di trovare la sequenza più lunga di figure geometriche accettata dall'automa. L'automa è rappresentato da un diagramma a stati finiti, in cui lo stato iniziale è riconoscibile dal testo **start** e lo stato finale dal testo **end**. Si veda la figura 4.6.

Il giocatore dovrà digitare la sequenza che ritiene corretta mediante gli appositi pulsanti (**triangolo**, **cerchio**, **rombo**, **quadrato**). La pressione di quest'ultimi aggiungeranno la figura corrispondente all'area dedicata alla sequenza. Mentre, il pulsante **Invio** e **Backspace** saranno necessari rispettivamente per confermare la sequenza e cancellare le figure inserite nella stessa.



**Figura 4.6:** Il minigioco di Alan Turing

Si entrerà nello stato di **Game Over** quando il **tempo** (60 secondi) sarà scaduto oppure si digiterà una **sequenza rifiutata** dall'automa. Il giocatore **vince** e ottiene dei **bits** (punti) quando inserisce e conferma una qualsiasi **sequenza accettata**. Nel caso in cui tale sequenza sia la più lunga, verrà conferito come premio una **medaglia**.

Ogni partita modificherà la disposizione delle figure sulle transizioni dell'automa.

L'**high score** sarà aggiornato con miglior tempo impiegato per superare la sfida.

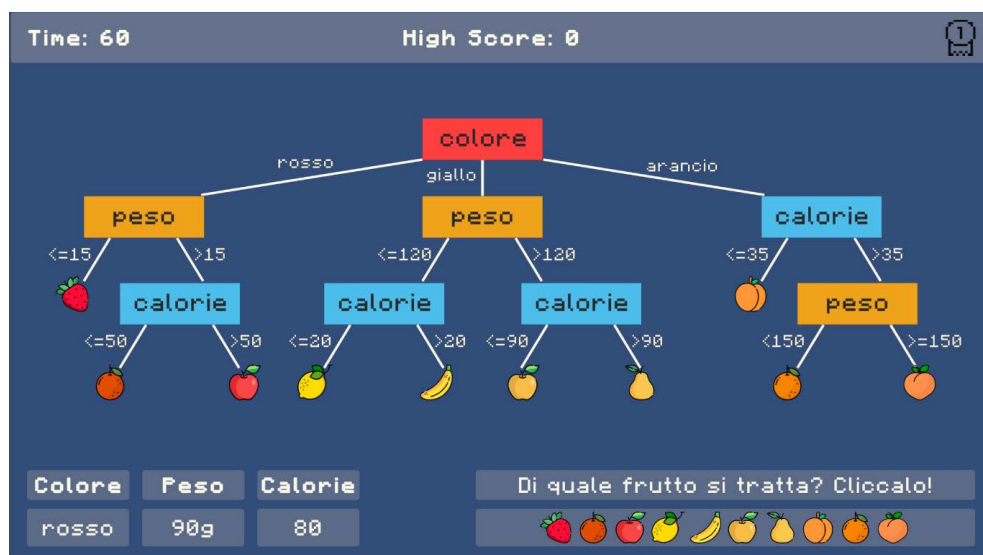
Durante la digitazione si otterranno casualmente come **bonus** dei bits, nel caso in cui si stia procedendo a creare una sequenza accettata.



### Fruit Decision Tree (John McCarthy)

In questo gioco, che prende spunto dai concetti di intelligenza artificiale, in particolare dal contesto del **machine learning**, viene mostrato al giocatore un albero decisionale in grado di classificare [112] alcuni frutti. L'obiettivo del giocatore è comprendere quale sarà l'output, ossia il frutto predetto, da tale modello, a partire dai dati di input forniti (**colore**, **peso**, **calorie**). Si veda la figura 4.7.

L'output sarà verificabile, a partire dalla radice, considerando le caratteristiche in corrispondenza dei nodi dell'albero e decidendo il cammino da seguire in base ai dati presenti sugli archi. Quando si giungerà a un nodo foglia si otterrà l'output. Per confermare la decisione, bisognerà cliccare il frutto che corrisponde alle caratteristiche date, nell'area dedicata. Ogni nuovo tentativo modificherà i dati forniti.



**Figura 4.7:** Il minigioco di John McCarthy

Si avranno a disposizione 60 secondi per effettuare la scelta, al termine dei quali si entrerà nello stato di **Game Over**. Il giocatore **vince** e ottiene dei **bits** (punti) quando effettuerà una scelta corretta. Nel caso in cui tale scelta sia per l'appunto corretta, e venga fatta entro 30 secondi, sarà conferito come premio una **medaglia**.

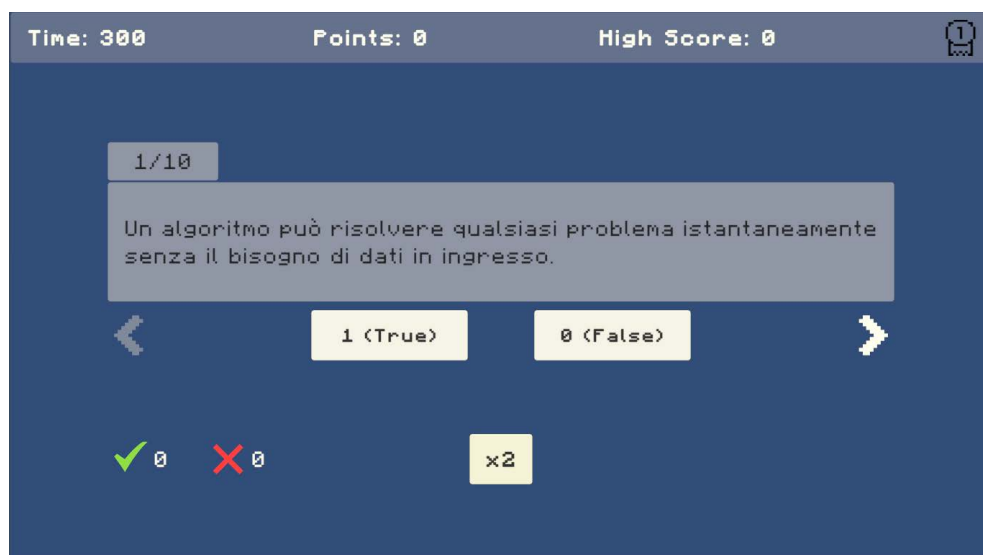
L'**high score** sarà aggiornato, ad ogni tentativo superato, in base al tempo impiegato per completare il minigioco.

Al termine della sfida, a seconda del tempo utilizzato, saranno forniti come **bonus** dei bits (punti), aggiunti a quelli già ottenibili con il suo completamento.

### Quiz 1 o 0 (George Boole)

Questo gioco, ispirato ai classici *quiz vero o falso*, ha sostanzialmente due obiettivi: (1) iniziare a far ragionare il giocatore in termini di **logica booleana**, utilizzando i valori **1** e **0**, rispettivamente per risposta vera e risposta falsa e (2) mettere insieme tutte le conoscenze acquisite durante il proseguo del gioco. Infatti, questa sfida sarà accessibile solo dopo aver completato la raccolta delle note presenti nel mondo di gioco e aver ottenuto tutte le relative medaglie nei precedenti minigiochi.

Saranno dunque proposte **10 domande** di cultura informatica, semplici e facili da comprendere, a cui bisognerà rispondere mediante gli appositi pulsanti (**1** e **0**). Si veda la figura 4.8. Una volta data la risposta non sarà possibile cambiarla, e si potrà avanzare o tornare indietro nelle domande mediante i pulsanti **Freccia**.



**Figura 4.8:** Il minigioco di George Boole

Ogni risposta corretta conferirà 1 punto, mentre ogni risposta errata sottrarrà 1 punto. Inoltre, sarà possibile attivare il **potenziamento x2** tramite l'apposito pulsante, il quale aggiungerà o sottrarrà il doppio dei punti. Si avranno a disposizione 5 minuti e saranno concessi al più 4 errori prima di entrare nello stato di **Game Over**. Mentre, almeno 6 risposte corrette garantiranno la **vittoria** e l'ottenimento dei **bits** (punti). Nel caso in cui il giocatore otterrà un punteggio  $\geq 15$ , sarà conferita una **medaglia**. Durante il gioco, per ogni risposta corretta, saranno dati come **bonus** dei bits (punti). L'**high score** sarà aggiornato, ad ogni tentativo, con il punteggio migliore.

## 4.3 La ricerca applicata al Game Design

Questa sezione si ricollegherà ai vari studi discussi nel capitolo 2, in particolare alla sezione 2.3. Queste ricerche sono state di fondamentale importanza durante il processo di progettazione di BitStory e permetteranno di valutare nel dettaglio il potere motivazionale dei risultati ottenuti.

Dal punto di vista dei **game motivators** [8] è stata fatta leva su quasi tutti gli aspetti descritti nella tabella 2.2:

- **sfida** - garantita dagli otto minigiochi appositamente progettati tenendo conto del target di destinazione, e caratterizzati da una difficoltà cognitiva moderata;
- **competenza** - mediante le sfide, i dialoghi e le note raccolte, si acquisiranno nuove conoscenze, che saranno messe alla prova nella sfida di *George Boole*;
- **competizione** - nasce nel momento in cui il giocatore tenta di superare il proprio high score nei diversi minigiochi e/o ottenere le medaglie;
- **controllo** - dato dalla libertà di affrontare le sfide in qualsiasi ordine, decidendo di accettarle subito o provarle in seguito, alternando momenti esplorativi a momenti di sfida, oltre che dai diversi modi per completare i minigiochi.
- **curiosità** - data dalla libertà esplorativa del mondo di gioco, dalla conoscenza dei personaggi, dall'ottenimento di medaglie, dalla raccolta delle note e infine dall'esperienza sonora del gioco;
- **emozioni** - i dialoghi con i personaggi e il superamento delle sfide alimentano l'inclusività del giocatore verso il mondo dell'informatica, indipendentemente dal genere, dalle abilità o altro;
- **fantasia** - alimentata dalla presenza di personaggi in scala inseriti in un mondo di gioco semi-reale, oltre che dall'ambientazione passata, dove gli individui delle diverse epoche si incontrano;
- **feedback** - caratterizzato dai diversi elementi dell'interfaccia grafica, come le regole, i progressi, i comandi, etc.;

- **immersione** - legata alle attività eseguibili, come girare per la mappa oppure svolgere i minigiochi come elemento di apprendimento;
- **regole** - descritte nel menu principale per far comprendere come manipolare il gioco, nonché prima dell'avvio e del termine di una sfida (inclusa la guida attraverso i dialoghi), allo scopo di dare indicazioni sugli obiettivi;
- **relazioni con il mondo esterno** - date dalle figure dei personaggi storici, da alcune informazioni fornite, dal legame tra le sfide e gli argomenti della computer science e infine dal mondo di gioco semi-reale;
- **interazione sociale** - possibilità di un contesto collaborativo indiretto, in cui informalmente due o più giocatori collaborano alla risoluzione di una sfida;
- **utilità** - fortemente garantita dalle conoscenze acquisite nel proseguo del gioco e dalla stimolazione di diverse aree cognitive durante la risoluzione delle sfide, come il ragionamento matematico, la logica e la memoria;

I **design principles (DP)** [8] in relazione con gli elementi motivazionali precedentemente descritti, sono di seguito elencati:

#### **Sfida**

**(DP1) Fornire sfide a vari livelli di difficoltà regolabili.** Il gioco darà la possibilità di scegliere l'ordine con cui affrontare le sfide. In certi casi, alcune potrebbero sembrare più semplici e altre leggermente più difficili.

**(DP2) Privilegiare sfide semplici rispetto a quelle complesse.** Tutte le sfide sono progettate per essere semplici, richiedendo la sola conoscenza delle regole.

**(DP3) Fornire tempo sufficiente per risolvere le sfide.** Ogni sfida richiederà un certo lasso di tempo per essere completata e in alcuni casi saranno forniti dei bonus che vanno ad incrementare il tempo a disposizione.

**(DP4) Suscitare curiosità con sfide interessanti e imprevedibili.** Le sfide sono progettate per creare diverse esperienze e perseguire vari obiettivi.

**(DP5) Consentire la ripetizione delle sfide.** Il gioco darà la possibilità di riprovare le sfide in ogni momento, sia in caso di vittoria che in caso di sconfitta.

## Controllo

**(DP6) Utilizzare input adatti ai giocatori e ai contesti di destinazione.** Il gioco può essere usufruito da computer anche non molto recenti. Non si impone la necessità di una scheda grafica dedicata. I controlli richiedono l'utilizzo di mouse e tastiera.

**(DP7) Utilizzare controlli sensibili e accurati.** Il gioco garantisce un output immediato in risposta ai diversi input da parte del giocatore.

**(DP8) Utilizzare controlli coerenti.** I controlli restano coerenti per più sessioni di gioco. Durante il passaggio mondo di gioco-minigioco, essi vengono inoltre semplificati, mantenendo la struttura iniziale.

**(DP9) Utilizzare controlli familiari, sicuri e confortevoli.** Nel caso in cui i giocatori abbiano precedenti esperienze nel mondo videoludico per computer, troveranno molta familiarità nei comandi adottati. In particolar modo per la gestione dei movimenti, la visuale in terza persona e gli utilizzi del mouse.

**(DP10) Favorire un'interazione semplice.** L'interazione richiede la memorizzazione di un ristretto numero di comandi.

**(DP12) Libertà di scelta e di controllo nel gameplay.** Il giocatore può decidere autonomamente cosa fare: esplorare liberamente, raccogliere le note o affrontare un minigioco piuttosto che un altro.

## Esplorazione

**(DP16) Libertà di esplorazione e sperimentazione.** Il giocatore potrà esplorare il mondo di gioco per conoscerne tutti i suoi elementi e i suoi personaggi.

**(DP17) Fornire più percorsi/opzioni e un ordine dinamico degli eventi.** Alcune sfide potranno essere risolte in vari modi (eg. più percorsi validi nella sfida di *Ada Lovelace*). Inoltre, il giocatore può decidere l'ordine con cui completare i minigiochi.

## Equità

**(DP20) Garantire simili opportunità di successo a prescindere dall'esperienza.** Indipendentemente dal fatto che un giocatore conosca o meno l'argomento a cui un minigioco è ispirato, esso sarà risolvibile senza alcun problema.

## Feedback

**(DP21) Fornire istruzioni e/o tutorial.** Nel menu principale vi è un pannello dedicato ai comandi di gioco, inoltre prima di ogni sfida, sono illustrate le regole di gioco.

**(DP22) Fornire un feedback immediato, positivo e utile.** Gli elementi della UI del mondo di gioco e delle sfide, forniscono feedback immediati sulle azioni svolte, come nota raccolta, bonus ottenuti, game over, sfida superata con o senza medaglia, etc.

**(DP23) Fornire un feedback chiaro attraverso canali diversi.** Oltre al feedback visivo, viene fornito il feedback uditivo, riproducendo vari suoni in base all'azione svolta.

**(DP24) Fornire accesso ai dati sulle prestazioni.** Nel menu principale e nel mondo di gioco è presente il pannello dei progressi che stabilisce le statistiche di gioco: note, punti e medaglie. Queste statistiche sono presenti anche nei minigiochi, che mostrano l'high score e una medaglia colorata se ottenuta, trasparente altrimenti.

## Obiettivi

**(DP25) Creare obiettivi chiari, significativi e raggiungibili.** Il giocatore avrà l'obiettivo di completare il gioco, ottenendo tutte le medaglie e collezionando le note. Questi obiettivi saranno scoperti attraverso il "fare" azioni nel gioco.

## Learning

**(DP28) Fornire contenuti e attività di apprendimento pertinenti e pedagogicamente fondati.** Il gioco richiederà poche abilità al giocatore, basterà infatti saper leggere, scrivere, memorizzare, contare, fare somme e conoscere le uguaglianze e le disuguaglianze. Queste abilità sono generalmente sviluppate nei primi anni di scuola elementare. Inoltre, le informazioni contenute nei dialoghi e nelle note definiscono i contenuti di apprendimento. Mentre, le sfide daranno l'opportunità di mettere in pratica questi concetti.

**(DP29) Fornire sfide cognitive.** Le sfide educative si presentano come semplici rompicapo, in grado di stimolare il ragionamento matematico, la logica e la memoria. Inoltre, l'ultima sfida di *George Boole*, presentata in forma di quiz, richiederà di sintetizzare le conoscenze acquisite e applicare il pensiero critico.

**(DP30) Incorporare strumenti di valutazione.** In base ai risultati ottenuti nelle sfide, sarà conferita una medaglia e un punteggio.

**(DP31) Scegliere un livello di trasparenza appropriato per i contenuti didattici.**

I contenuti didattici sono descritti in maniera semplice, fornendo una conoscenza generale dell'informatica, evitando troppi dettagli e spiegando i concetti essenziali.

**(DP32) Fornire il tempo necessario per riflettere nei momenti di gioco più intensi.**

Ogni minigioco fornirà più o meno tempo, lasciando alcuni momenti per pensare. Al termine di una sfida se non si ottiene la medaglia, oppure si supera un certo limite, si informerà il giocatore, in modo da spingerlo a migliorare l'eventuale soluzione.

### **Profilo e proprietà**

**(DP33) Usare un profilo/avatar che il giocatore può possedere e con cui può relazionarsi.** Il gioco renderà disponibile un profilo in cui saranno salvati tutti i progressi compiuti durante il gioco: bits (punti), note, medaglie, risultati delle sfide, etc.

### **Rilevanza e attinenza**

**(DP36) Mettere in relazione il gameplay con i contesti del mondo reale.** Il gioco presenta diversi elementi in relazione con i contesti reali, come: i personaggi storici esistiti, alcune informazioni fornite nelle note e nei dialoghi, il legame tra le sfide e gli argomenti dell'informatica e infine il mondo di gioco semi-reale.

### **Risorse ed economia**

**(DP40) Consentire la raccolta di beni virtuali** Per garantire continuità nel gioco sono stati creati diversi beni virtuali: bits (punti), medaglie e note.

### **Gioco sociale**

**(DP44) Fornire opportunità di concorrenza.** Opportunità di competere contro se stessi, per migliorare il proprio high score nelle varie sfide e/o ottenere le medaglie.

**(DP 45) Offrire opportunità di collaborazione.** Fuori dal gioco, due o più giocatori, possono collaborare informalmente alla risoluzione di una sfida.

### Narrazione e fantasia

**(DP50) Fornire un contesto fantasy.** Il gioco include personaggi in scala inseriti in un mondo semi-reale, che ripercorre gli aspetti del gioco creativo dei bambini. Inoltre, vi è un'ambientazione passata, dove gli individui delle diverse epoche si incontrano.

**(DP51) Offrire esperienze di gioco di ruolo.** Il gioco assegnerà al giocatore il ruolo di un bambino. Tale ruolo sarà collegato al profilo discusso nel DP33.

Lo studio della letteratura [32] ha contribuito a stabilire le caratteristiche che stimolano l'apprendimento nei bambini: **facilità d'uso, interattività, immersione**, ma soprattutto **visualizzazione**. Quest'ultima, giustifica il focus dato all'aspetto grafico di BitStory, alternando ambienti tridimensionali (3D) a quelli bidimensionali (2D). Inoltre, sono stati presi in considerazione alcuni aspetti che uomini e donne amano vedere nei giochi [31], che è possibile notare nel Serious Game. Nel caso degli **uomini** prevalgono: la padronanza del gioco, la competizione, la presenza di puzzle e l'apprendimento attraverso gli errori. Mentre, per quanto riguarda le **donne**: le emozioni, i riferimenti con il mondo reale, i dialoghi e gli enigmi verbali.

## 4.4 Risoluzione alle limitazioni dei lavori correlati

All'interno del capitolo 2, in particolare nella sezione 2.4, sono stati evidenziati alcuni aspetti limitanti dei lavori correlati.

Le limitazioni riscontrate nel gioco di **Nabbovaldo e il ricatto del cyberspazio**, sono state risolte nel seguente modo: (i) centralità dell'informatica nei vari minigiochi; (ii) opportunità di rigiocare i minigiochi in qualsiasi momento, siano essi superati, che non superati; (iii) salvataggio del proprio high score nei vari minigiochi; (iv) medaglie in premio per le sfide superate sotto certe condizioni.

Le limitazioni evidenziate nel gioco **while True: learn()**, sono state risolte nel seguente modo: (i) cenni storici e concetti informatici semplici, corredati da informazioni essenziali; (ii) pochi minigiochi, con giusta varietà l'uno dall'altro; (iii) minigiochi non troppo impegnativi, sono sufficienti le abilità dei primi anni di scuola elementare e la comprensione delle regole; (iv) durata totale del gioco stimata: 25-30 minuti.



## 4.5 Risultati delle prestazioni di gioco

In questa sezione saranno discussi i risultati del benchmark che è stato effettuato per valutare le prestazioni della demo di BitStory, come descritto nella sezione 3.4. La tabella 4.1 riassume i risultati dei benchmark con le relative componenti dei calcolatori su cui è stato testato il gioco.

**Tabella 4.1:** Risultati del benchmark della demo di gioco [7]

Processore	Scheda Grafica	RAM	FPS Min	FPS Max	FPS Medi
Intel Core i5-9300H	Nvidia GeForce GTX 1650 Mobile	16 GB	57	62	60
AMD Ryzen 7 3750H	Nvidia GeForce GTX 2060 Mobile	16 GB	59	62	60
AMD Ryzen 5 3500U	-	8 GB	55	61	59
Intel Celeron N2840	-	8 GB	5	15	10
Intel Celeron N3450	-	8 GB	4	12	8

Da questi risultati si evince che il gioco riesce ad essere eseguito molto bene dagli elaboratori con una scheda grafica dedicata ed anche da quelli con una buona scheda grafica integrata, come nel caso dell'AMD Ryzen 5. Mentre i calcolatori con processori di fascia bassa, come gli Intel Celeron, necessitano di un abbassamento della qualità grafica al fine di un miglioramento delle prestazioni.

#### 5.1 Panoramica del lavoro svolto

La carenza di impegno e di motivazione degli studenti attuali rappresentano la principale problematica dell'istruzione moderna. A questa, si aggiunge il mancato interesse per l'informatica e, in generale, per le discipline STEM, oltre alla percezione di un mondo ancora fortemente dominato dal genere maschile.

Per soccombere a tali limitazioni, è stato realizzato questo lavoro di tesi, in cui è presentato lo sviluppo di BitStory [7], un Serious Game dedicato ai bambini delle scuole elementari, ideato attraverso un'attenta analisi della letteratura, che ha permesso di crearne una versione demo. Il gioco dà l'opportunità di controllare un avatar in un mondo composto da computer d'epoca e popolato da alcuni personaggi storici dell'informatica. In questo mondo, sono introdotti i concetti e le origini della materia, attraverso i dialoghi con i personaggi, la raccolta di note esplicative e i minigiochi ispirati ad argomenti a tema. Oltre a fornire sfide educative, il gioco adotta altre meccaniche di gamification: punteggi e premi. Infine, il potere motivazionale dei risultati ottenuti è stato valutato sulla base delle ricerche effettuate.

In questo modo, si vuole educare ed avvicinare i bambini all'informatica, oltre che guidarli a stimolare il ragionamento matematico, la logica e la memoria.

## 5.2 Discussioni

In questa sezione saranno discusse alcune osservazioni emerse durante il lavoro.

**Serious Game VS Tool di programmazione grafica.** Una prima considerazione che è stata fatta, ancor prima di progettare BitStory, è stata decidere in che modo poter trasmettere alcune conoscenze dell'informatica ai bambini. Analizzando ciò che è attualmente presente in letteratura, sono stati trovati diversi tool di programmazione grafica, come *Scratch* e *GameMaker*. Tuttavia, come viene spiegato nell'articolo di Monika Akbar et al. [35], questi tool possono ostacolare l'apprendimento, poiché mancano di un contesto autentico, come personaggi, storie, obiettivi, etc. Per cui, lo sviluppo è stato orientato su un Serious Game che comprendesse questi elementi.

**Single Player VS Multiplayer.** Dallo studio della letteratura emerge che l'interazione sociale può favorire l'apprendimento [8]. Tuttavia, questo aspetto avrebbe comportato alla nascita di una competizione tra gli studenti, rischiando di distogliere l'attenzione dai contenuti di apprendimento e dall'esperienza generale offerta dal gioco. Pertanto, si è scelto di mantenere il gioco in modalità Single Player, consentendo al giocatore di progredire con i propri tempi, evitando qualsiasi forma di pressione. Nulla esclude però, di realizzare un contesto collaborativo indiretto, come è stato descritto nella sezione 4.3.

**Utilizzo di una teoria didattica.** Dopo aver realizzato la demo e aver testato il suo funzionamento, si è cercato di capire se i concetti illustrati nei dialoghi e nelle note fossero abbastanza chiari per il target del gioco, nonostante la forte semplificazione data a queste informazioni. E in effetti, la letteratura consiglia, nel caso dello sviluppo di giochi educativi, di utilizzare delle opportune teorie didattiche [8]. Sono state quindi condotte alcune ricerche sulle teorie che possono migliorare questi aspetti del gioco, soffermandosi sulla *Teoria Cognitiva dell'Apprendimento Multimediale* [113]. Questa teoria, si concentra sull'utilizzo dei media, come testo, immagini, audio e video per migliorare il processo di apprendimento. Spunti di riflessione che si possono tenere in considerazione in future versioni del gioco.

## 5.3 Sviluppi futuri

Secondo il lavoro svolto e alcune riflessioni derivate dalla sezione 5.2, sono proposti i seguenti sviluppi futuri:

- Innanzitutto è fissato l'obiettivo di rilasciare una prima versione completa del gioco, ultimando lo sviluppo degli altri minigiochi a partire dalle scene di mockup realizzate.
- Un'altra miglioria è l'aggiunta di un menu di pausa alla scena del mondo di gioco che includa diverse sezioni, come delle liste descrittive dei personaggi incontrati e delle note raccolte, oltre che un riepilogo dei risultati ottenuti nei vari minigiochi. L'ultima sezione che si potrebbe inserire è relativa alla personalizzazione dei comandi di gioco. Queste migliorie tengono conto del DP11 della classe Controllo e dei DP34-35 della classe Profilo e proprietà [8].
- Altra modifica potrebbe essere apportata al sistema dei dialoghi e delle note da collezione, includendo oltre al testo, delle immagini di accompagnamento o brevi video, secondo la filosofia della *Teoria Cognitiva dell'Apprendimento Multimediale* [113].
- Ulteriore aggiunta potrebbe essere quella di andare a realizzare vari livelli a difficoltà crescente per ogni minigioco, realizzando delle versioni alternative degli stessi e che facciano sempre leva sugli stimoli di varie aree cognitive.
- Un'altra sezione da inserire, al fine di rendere il gioco molto più accessibile da parte di calcolatori meno prestanti, è relativa alla regolazione manuale della qualità grafica del gioco.
- Infine, come ultimo step, si vuole sperimentare BitStory in varie scuole primarie, al fine di raccogliere utili informazioni in merito alle impressioni date dal gioco, come la curiosità, la motivazione, l'apprendimento, etc.

---

## Bibliografia

---

- [1] E. Montoli. (2020) Renovatio Quest. [Online]. Available: <http://renovatioquest.it> (Citato alle pagine iii, 7 e 9)
- [2] J. Shell, "The Game Consists of Elements," in *The Art of Game Design*. CRC Press, 2008, pp. 39–46. (Citato alle pagine iii, 10 e 11)
- [3] grifomultimedia and Registro.it. (2021) Nabbovaldo e il ricatto del cyberspazio. [Online]. Available: [https://grifomultimedia.it/portfolio\\_page/nabbovaldo\\_e\\_il\\_ricatto\\_dal\\_cyberspazio](https://grifomultimedia.it/portfolio_page/nabbovaldo_e_il_ricatto_dal_cyberspazio) (Citato alle pagine iii, 16 e 17)
- [4] Steam. (2019) while True: learn(). [Online]. Available: [https://store.steampowered.com/app/619150/while\\_True\\_learn/](https://store.steampowered.com/app/619150/while_True_learn/) (Citato alle pagine iii e 19)
- [5] Blender Foundation. (2023) Eevee. [Online]. Available: <https://docs.blender.org/manual/en/latest/render/eevee/introduction.html> (Citato alle pagine iii e 26)
- [6] Adobe. (2023) Rigging e skeletal animation: cos'è e come funziona. [Online]. Available: <https://www.adobe.com/it/creativecloud/animation/discover/rigging.html> (Citato alle pagine iii, 24 e 29)

- 
- [7] R. Aurucci. (2023) BitStory repository GitHub. [Online]. Available: <https://github.com/raffaele-aurucci/BitStory.git> (Citato alle pagine iii, iv, 2, 21, 25, 32, 33, 35, 38, 57 e 58)
- [8] T. H. Laine and R. S. N. Lindberg, "Designing Engaging Games for Education: A Systematic Literature Review on Game Motivators and Design Principles," *IEEE Transactions on Learning Technologies*, vol. 13, no. 4, pp. 804–821, 2020. (Citato alle pagine iv, 5, 6, 7, 11, 14, 40, 51, 52, 59 e 60)
- [9] K. Gabriela, A. Nedezhda, and Y. Lina, "Gamification in Education," 2014. (Citato alle pagine 4 e 5)
- [10] K. M. Kapp, "The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education," 2012, pp. 9–15. (Citato a pagina 4)
- [11] R. M. Ryan and E. L. Deci, "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions." *Contemporary educational psychology*, vol. 25, no. 1, pp. 54–67, 2000. (Citato a pagina 5)
- [12] M. Csikszentmihalyi, "Flow: The Psychology of Optimal Experience," H. USA, Ed., 2011. (Citato alle pagine 5 e 14)
- [13] Xbox Game Studios and Mojang Studios. (2016) Minecraft Education. [Online]. Available: <https://education.minecraft.net> (Citato alle pagine 5, 7 e 32)
- [14] C. G. von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA—An educational game for teaching SCRUM in computing courses," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2675–2687, 2013. (Citato a pagina 6)
- [15] Microsoft. (2020) Flight Simulator. [Online]. Available: <https://www.flightsimulator.com> (Citato a pagina 7)
- [16] H. Partovi and A. Partovi. (2013) Code.org. [Online]. Available: <https://code.org> (Citato a pagina 7)
- [17] Duolingo Team. (2011) Duolingo. [Online]. Available: <https://it.duolingo.com> (Citato a pagina 7)

- [18] J. Swarz, A. Ousley, A. Magro, M. Rienzo, D. Burns, A. Lindsey, B. Wilburn, and S. Bolcar, "CancerSpace: A Simulation-Based Game for Improving Cancer-Screening Rates," *IEEE Computer Graphics and Applications*, vol. 30, no. 1, pp. 90–94, 2010. (Citato a pagina 7)
- [19] Nintendo. (2019) Ring Fit Adventure. [Online]. Available: <https://ringfitadventure.nintendo.com> (Citato a pagina 7)
- [20] Nike. (2020) Nike Run Club. [Online]. Available: <https://www.nike.com/it/nrc-app> (Citato a pagina 7)
- [21] K. Sostmann, D. Tolks, M. Fischer, and S. Buron, "Serious Games for Health: Spielend lernen und heilen mit Computerspielen?" *GMS Medizinische Informatik, Biometrie und Epidemiologie*, vol. 6, 01 2010. (Citato a pagina 7)
- [22] B. Keivan and R. H. Maghami, "Serious Games in Computer Science Learning Goals," pp. 161–166, 2018. (Citato alle pagine 8, 9 e 40)
- [23] A. Daniela, "Serious Games – A New Perspective on Workbased Learning," *Procedia - Social and Behavioral Sciences*, vol. 204, pp. 277–281. (Citato a pagina 8)
- [24] M. Zyda, J. Hiles, A. Mayberry, C. Wardynski, M. Capps, B. Osborn, R. Shilling, M. Robaszewski, and M. Davis, "Entertainment RD for defense," *IEEE Computer Graphics and Applications*, vol. 23, no. 1, pp. 28–36, 2003. (Citato a pagina 8)
- [25] A. De Gloria, F. Bellotti, and R. Berta, "Serious Games for education and training," *International Journal of Serious Games*, vol. 1, 02 2014. (Citato a pagina 8)
- [26] DFC Intelligence. (2023) Global Video Game Audience Reaches 3.7 Billion. [Online]. Available: <https://www.dfciint.com/global-video-game-audience-reaches-3-7-billion> (Citato a pagina 8)
- [27] R. M. Ryan, S. C. Rigby, and A. Przybylski, "The Motivational Pull of Video Games: A Self-Determination Theory Approach," *Motivation and Emotion*, vol. 30, no. 4, pp. 344–360, 2006. (Citato a pagina 8)

- [28] M. Ullah, S. U. Amin, M. Munsif, U. Safaev, H. Khan, S. Khan, and H. Ullah, "Serious games in science education. a systematic literature review," *Virtual Reality Intelligent Hardware*, vol. 4, no. 3, pp. 189–209, 2022. (Citato a pagina 9)
- [29] J. Shell, "The Designer Creates an Experience," in *The Art of Game Design*. CRC Press, 2008, pp. 9–22. (Citato a pagina 10)
- [30] —, "In the Beginning, There Is the Designer," in *The Art of Game Design*. CRC Press, 2008, pp. 1–7. (Citato a pagina 10)
- [31] —, "The Game is Made for a Player," in *The Art of Game Design*. CRC Press, 2008, pp. 97–112. (Citato alle pagine 10 e 56)
- [32] P. P. Yanti, Y. Rosmansyah, and B. Dabarsyah, "Serious Games for Children: A Systematic Literature Review," in *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2019, pp. 79–84. (Citato alle pagine 15 e 56)
- [33] S. Fonseca, S. González, B. Rodríguez, M. Seda, J. Bobonis, A. Canela, C. Rolón, J. Bosque, L. Méndez, A. Cartagena, T. Fuentes, N. G. Santiago, and A. Nieves, "Educational Nanotechnology Video Game to Inspire Middle and High School Students to Pursue STEM Related Professional Careers," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–5. (Citato a pagina 16)
- [34] J. Baldeón, I. Rodríguez, A. Puig, D. Gómez, and S. Grau, "From learning to game mechanics: The design and the analysis of a serious game for computer literacy," in *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 2016, pp. 1–6. (Citato a pagina 16)
- [35] M. Akbar, L. Dura, A. Q. Gates, A. Ortega, M. K. Roy, C. Santiago, J. G. Tellez, and E. Villa, "Sol y Agua: A Game-based Learning Platform to Engage Middle-school Students in STEM, year=2018," in *2018 IEEE Frontiers in Education Conference (FIE)*, pp. 1–9. (Citato alle pagine 16 e 59)
- [36] oltrelecolonne.it. (2021) "Nabbovaldo e il ricatto del cyberspazio", videogioco per imparare la cybersecurity. [Online]. Available: [https://www.nic.it/sites/default/files/documenti/2021/20211104\\_563274328.pdf](https://www.nic.it/sites/default/files/documenti/2021/20211104_563274328.pdf) (Citato a pagina 16)



- [37] Unity Technologies. (2023) unity.com. [Online]. Available: <https://unity.com/> (Citato alle pagine 17, 20, 22, 28 e 30)
- [38] Istat. (2023) Livelli di istruzione e ritorni occupazionali | Anno 2021. [Online]. Available: <https://www.istat.it/it/files//2022/10/Livelli-di-istruzione-e-ritorni-occupazionali-anno-2021.pdf> (Citato a pagina 21)
- [39] Blender Foundation. (2023) blender.org. [Online]. Available: <https://www.blender.org/> (Citato alle pagine 22 e 25)
- [40] Mixamo. (2023) mixamo.com. [Online]. Available: <https://www.mixamo.com/#/> (Citato alle pagine 24, 28 e 29)
- [41] Inkscape. (2023) inkscape.org. [Online]. Available: <https://inkscape.org/it/> (Citato alle pagine 24, 33 e 35)
- [42] M. Keleşoğlu and D. Gulec Ozer, “A Study on Digital Low Poly Modeling Methods as an Abstraction Tool in Design Processes,” *Civil Engineering and Architecture*, vol. 9, pp. 153–169, 10 2021. (Citato a pagina 25)
- [43] Blender Foundation. (2023) Blender Manual: Move. [Online]. Available: [https://docs.blender.org/manual/en/latest/scene\\_layout/object/editing/transform/move.html](https://docs.blender.org/manual/en/latest/scene_layout/object/editing/transform/move.html) (Citato a pagina 27)
- [44] ——. (2023) Blender Manual: Rotate. [Online]. Available: [https://docs.blender.org/manual/en/latest/scene\\_layout/object/editing/transform/rotate.html](https://docs.blender.org/manual/en/latest/scene_layout/object/editing/transform/rotate.html) (Citato a pagina 27)
- [45] ——. (2023) Blender Manual: Scale. [Online]. Available: [https://docs.blender.org/manual/en/latest/scene\\_layout/object/editing/transform/scale.html](https://docs.blender.org/manual/en/latest/scene_layout/object/editing/transform/scale.html) (Citato a pagina 27)
- [46] ——. (2023) Blender Manual: Measure. [Online]. Available: <https://docs.blender.org/manual/en/latest/editors/3dview/toolbar/measure.html> (Citato a pagina 27)

- [47] ——. (2023) Blender Manual: Extrude Region. [Online]. Available: [https://docs.blender.org/manual/en/latest/modeling/meshes/tools/extrude\\_region.html](https://docs.blender.org/manual/en/latest/modeling/meshes/tools/extrude_region.html) (Citato a pagina 27)
- [48] ——. (2023) Blender Manual: Inset Faces. [Online]. Available: [https://docs.blender.org/manual/en/latest/modeling/meshes/editing/face/inset\\_faces.html](https://docs.blender.org/manual/en/latest/modeling/meshes/editing/face/inset_faces.html) (Citato a pagina 27)
- [49] ——. (2023) Blender Manual: Bevel. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/meshes/editing/edge/bevel.html> (Citato a pagina 27)
- [50] ——. (2023) Blender Manual: Loop Cut. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/meshes/tools/loop.html> (Citato a pagina 27)
- [51] ——. (2023) Blender Manual: Knife. [Online]. Available: [https://docs.blender.org/manual/en/latest/modeling/meshes/editing/face/intersect\\_knife.html](https://docs.blender.org/manual/en/latest/modeling/meshes/editing/face/intersect_knife.html) (Citato a pagina 28)
- [52] ——. (2023) Blender Manual: Spin. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/meshes/tools/spin.html> (Citato a pagina 28)
- [53] ——. (2023) Blender Manual: Array Modifier. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/array.html> (Citato a pagina 28)
- [54] ——. (2023) Blender Manual: Mirror Modifier. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/mirror.html> (Citato a pagina 28)
- [55] ——. (2023) Blender Manual: Solidify Modifier. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/solidify.html> (Citato a pagina 28)

- [56] Adobe. (2023) Cos'è il texturing 3D? [Online]. Available: [https://www.adobe.com/ch\\_it/products/substance3d/discover/3d-texturing.html](https://www.adobe.com/ch_it/products/substance3d/discover/3d-texturing.html) (Citato a pagina 28)
- [57] Blender Foundation. (2023) Blender Manual: Principled BSDF. [Online]. Available: [https://docs.blender.org/manual/en/latest/render/shader\\_nodes/shader/principled.html](https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/principled.html) (Citato a pagina 28)
- [58] —. (2023) Blender Manual: Emission. [Online]. Available: [https://docs.blender.org/manual/en/latest/render/shader\\_nodes/shader/emission.html](https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/emission.html) (Citato a pagina 28)
- [59] Unity Technologies. (2023) Unity Manual: Introduction to collision. [Online]. Available: <https://docs.unity3d.com/Manual/CollidersOverview.html> (Citato a pagina 28)
- [60] Blender Foundation. (2023) Blender Manual: Collisions. [Online]. Available: [https://docs.blender.org/manual/en/latest/physics/rigid\\_body/properties/collisions.html](https://docs.blender.org/manual/en/latest/physics/rigid_body/properties/collisions.html) (Citato a pagina 28)
- [61] Weisl. (2023) Collider-Tools. [Online]. Available: <https://github.com/Weisl/Collider-Tools> (Citato a pagina 28)
- [62] Blender Foundation. (2023) Blender Manual: Normals. [Online]. Available: <https://docs.blender.org/manual/en/latest/modeling/meshes/editing/mesh/normals.html> (Citato a pagina 28)
- [63] Unity Technologies. (2023) Unity Manual: Mesh Collider component reference. [Online]. Available: <https://docs.unity3d.com/Manual/class-MeshCollider.html> (Citato a pagina 30)
- [64] —. (2023) Unity Manual: Box Collider. [Online]. Available: <https://docs.unity3d.com/Manual/class-BoxCollider.html> (Citato alle pagine 30 e 34)
- [65] —. (2023) Unity Manual: Rigidbody. [Online]. Available: <https://docs.unity3d.com/Manual/class-Rigidbody.html> (Citato alle pagine 30 e 34)

- [66] ——. (2023) Unity Manual: Animator Controllers. [Online]. Available: <https://docs.unity3d.com/Manual/AnimatorControllers.html> (Citato a pagina 30)
- [67] ——. (2023) Unity Manual: Blend Trees. [Online]. Available: <https://docs.unity3d.com/Manual/AnimatorControllers.html> (Citato a pagina 30)
- [68] ——. (2023) Unity Manual: Character Controller. [Online]. Available: <https://docs.unity3d.com/Manual/character-control-section.html> (Citato a pagina 31)
- [69] ——. (2023) Unity Manual: Camera. [Online]. Available: <https://docs.unity3d.com/Manual/class-Camera.html> (Citato a pagina 31)
- [70] ——. (2023) Unity Manual: Tags. [Online]. Available: <https://docs.unity3d.com/Manual/Tags.html> (Citato a pagina 31)
- [71] ——. (2023) Unity Manual: Layers. [Online]. Available: <https://docs.unity3d.com/Manual/Layers.html> (Citato a pagina 31)
- [72] ——. (2023) Unity Manual: Canvas. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/comp-CanvasComponents.html> (Citato alle pagine 31 e 34)
- [73] IdreesInc. (2023) Minecraft-Font. [Online]. Available: <https://github.com/IdreesInc/Minecraft-Font> (Citato a pagina 32)
- [74] Unity Technologies. (2023) Unity Scripting API: Object.DontDestroyOnLoad. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html> (Citato a pagina 32)
- [75] ——. (2023) Unity Manual: Audio Source. [Online]. Available: <https://docs.unity3d.com/Manual/class-AudioSource.html> (Citato a pagina 32)
- [76] ——. (2023) Unity Manual: Audio Clip. [Online]. Available: <https://docs.unity3d.com/Manual/class-AudioClip.html> (Citato a pagina 32)
- [77] ——. (2023) Unity Manual: Audio Listener. [Online]. Available: <https://docs.unity3d.com/Manual/class-AudioListener.html> (Citato a pagina 32)

- [78] ——. (2023) Unity Manual: Mesh Renderer. [Online]. Available: <https://docs.unity3d.com/Manual/class-MeshRenderer.html> (Citato a pagina 34)
- [79] ——. (2023) Unity Manual: Sphere Collider. [Online]. Available: <https://docs.unity3d.com/Manual/class-SphereCollider.html> (Citato a pagina 34)
- [80] ——. (2023) Unity Manual: Physic Material. [Online]. Available: <https://docs.unity3d.com/Manual/class-PhysicMaterial.html> (Citato a pagina 34)
- [81] ——. (2023) Unity Manual: Particle System. [Online]. Available: <https://docs.unity3d.com/Manual/class-ParticleSystem.html> (Citato a pagina 34)
- [82] Flaticon. (2023) flaticon.com. [Online]. Available: <https://www.flaticon.com/> (Citato a pagina 35)
- [83] Unity Asset Store. (2023) Maze Generator. [Online]. Available: <https://assetstore.unity.com/packages/tools/modeling/maze-generator-38689> (Citato a pagina 35)
- [84] Freesound.org. freesound.org. [Online]. Available: <https://freesound.org> (Citato a pagina 36)
- [85] MAbdurrahman. calculatorClick. [Online]. Available: <https://freesound.org/people/MAbdurrahman/sounds/425187/> (Citato a pagina 36)
- [86] MATRIXXX. PageTurn01. [Online]. Available: [https://freesound.org/people/MATRIXXX\\_/sounds/397548/](https://freesound.org/people/MATRIXXX_/sounds/397548/) (Citato a pagina 36)
- [87] ——. A Step On Dirt. [Online]. Available: [https://freesound.org/people/MATRIXXX\\_/sounds/370367/](https://freesound.org/people/MATRIXXX_/sounds/370367/) (Citato a pagina 36)
- [88] NHumphrey. Ufo. [Online]. Available: <https://freesound.org/people/NHumphrey/sounds/204417/> (Citato a pagina 36)
- [89] GameAudio. Spacey 1up/power up. [Online]. Available: <https://freesound.org/people/GameAudio/sounds/220173/> (Citato a pagina 36)
- [90] Fupicat. WinSquare. [Online]. Available: <https://freesound.org/people/Fupicat/sounds/527650/> (Citato a pagina 36)

- 
- [91] myfox14. Game Over Arcade. [Online]. Available: <https://freesound.org/people/myfox14/sounds/382310/> (Citato a pagina 36)
- [92] josefpres. 8 bit game loop 004 only organ long 120 bpm. [Online]. Available: <https://freesound.org/people/josefpres/sounds/653802/> (Citato a pagina 36)
- [93] ——. 8 bit game loop 005 only drums short 120 bpm. [Online]. Available: <https://freesound.org/people/josefpres/sounds/655080/> (Citato a pagina 36)
- [94] Beepa Pty Ltd. (2013) Fraps: Real-time video capture benchmarking. [Online]. Available: <https://fraps.com/> (Citato a pagina 37)
- [95] Unity Technologies. (2023) Unity Manual: Quality Settings. [Online]. Available: <https://docs.unity3d.com/Manual/class-QualitySettings.html> (Citato a pagina 37)
- [96] J. Shell, “The Elements Support a Theme,” in *The Art of Game Design*. CRC Press, 2008, pp. 47–56. (Citato a pagina 39)
- [97] C. Gripton and K. Vincent, “Using small world toys for research: a method for gaining insight into children’s lived experiences of ‘ability’,” *International Journal of Research Method in Education*, vol. 44, 2020. (Citato a pagina 39)
- [98] G. O’Regan, “Ada Lovelace,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 179–181. (Citato alle pagine 39 e 41)
- [99] —, “Alan Turing,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 251–257. (Citato a pagina 39)
- [100] A. Konheim, “Horst Feistel: the inventor of LUCIFER, the cryptographic algorithm that changed cryptology,” *Journal of Cryptographic Engineering*, vol. 9, pp. 85–100, 2019. [Online]. Available: <https://doi.org/10.1007/s13389-018-0198-5> (Citato a pagina 39)
- [101] G. O’Regan, “Dennis Ritchie,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 219–222. (Citato a pagina 39)

- [102] —, “George Boole,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 47–52. (Citato a pagina 40)
- [103] —, “Claude Shannon,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 233–238. (Citato a pagina 40)
- [104] —, “John McCarthy,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 183–185. (Citato a pagina 40)
- [105] —, “Tim Berners-Lee,” in *Giants of Computing: A Compendium of Select, Pivotal Pioneers*. Springer London, 2013, pp. 39–42. (Citato a pagina 40)
- [106] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, “Cammini minimi con sorgente singola,” in *Introduzione agli algoritmi*. McGraw-Hill, 2008, pp. 491–524. (Citato a pagina 42)
- [107] A. S. Tanenbaum and D. J. Wetherall, “Il livello fisico,” in *Reti di Calcolatori*. Pearson, 2018, pp. 83–180. (Citato a pagina 44)
- [108] W. Stallings, “Cifrario di Cesare,” in *Crittografia e sicurezza delle reti*. McGraw-Hill, 2007, pp. 36–39. (Citato a pagina 45)
- [109] W3Schools. (2023) HTML Tutorial. [Online]. Available: <https://www.w3schools.com/html/default.asp> (Citato a pagina 46)
- [110] A. Silberschatz, P. B. Galvin, and G. Gagne, “Gestione della memoria,” in *Sistemi operativi: Concetti ed esempi*. Pearson, 2019, pp. 375–415. (Citato a pagina 47)
- [111] M. Sipser, “Automati e linguaggi,” in *Introduzione alla teoria della computazione*. Maggioli Editore, 2016, pp. 33–65. (Citato a pagina 48)
- [112] A. Géron, “Decision Trees,” in *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly, 2019, pp. 177–190. (Citato a pagina 49)
- [113] S. Sorden, “The Cognitive Theory of Multimedia Learning,” 2023. (Citato alle pagine 59 e 60)

---

## Ringraziamenti

---

Questa tesi è frutto di una passione che ho sempre avuto sin da quando ero solo un bambino: il mondo dei videogiochi. Essi, a differenza di un libro o di un semplice film, hanno saputo trasmettermi emozioni difficilmente spiegabili a parole.

Ricordo di quando mio padre tornò da lavoro, avevo forse 3 o 4 anni, ed ebbe l'idea di portarmi in un negozio di elettronica, lì trovammo una console collegata ad un televisore, su cui era riprodotto il classico Super Mario. E no, non era la Nintendo, non era la Playstation o la Xbox, era una di quelle console da pochi euro, dove l'unica cosa che contava era il divertimento. Ci decidemmo a comprarla, e in seguito tornammo a casa, pronti per giocare! Da lì sono cambiate molte cose, ma la mia passione non è mai cambiata. Ancora oggi, di fronte a un videogioco provo le stesse emozioni di quando ero bambino, e forse, è proprio questa la magia che si nasconde dietro ad ognuno di essi, chi più, chi meno. È proprio grazie ai videogiochi, se sono riuscito ad avvicinarmi al mondo dell'informatica. Attraverso questo lavoro di tesi vorrei quindi lanciare un messaggio a tutti quei bambini che non sanno ancora cosa fare da grandi, facendogli scoprire la bellezza di questo mondo, dove l'unico limite è la fantasia!

Innanzitutto, ci tengo a ringraziare il professore Fabio Palomba per avermi concesso l'opportunità di realizzare questo lavoro e aver supervisionato questa tesi. Un ringraziamento va anche alla dottoressa Giusy Annunziata, per i numerosi consigli e suggerimenti fornitimi durante questi mesi, oltre alla sua gentilezza e disponibilità.



---

Ringrazio i miei genitori per avermi sempre detto di andare a scuola e imparare, imparare e imparare, senza mai fermarmi; è grazie a loro se ho potuto decidere autonomamente di affrontare questo percorso universitario.

Ringrazio i miei nonni, ormai da tempo non più presenti, che hanno sempre creduto in me, mi dicevano continuamente: « *Tu sì o' cap guaglione* ».

Ringrazio mio fratello, Felice, che dopo qualche esame, spiegavo com'era andata e mi diceva: « *Nun t preoccupà, tu sì tropp fort!* »

Ringrazio i miei cugini, in particolare Giuseppe, per i consigli che mi ha fornito durante questo percorso universitario, e soprattutto per il proseguo.

Ringrazio i miei zii per il supporto donatomi durante gli anni che mi hanno portato dove sono oggi, da chi mi aiutava nei compiti per casa a chi veniva a prendermi a scuola quando i miei genitori erano a lavoro.

Un ringraziamento speciale va alla mia ragazza, Dora, che mi ha sempre infondato sicurezza e fiducia in ciò che sono i miei obiettivi; una spalla su cui contare, una forza e una carica emotiva che solo lei sa darmi. È stata proprio lei a dirmi durante tutti questi anni che io ce l'avrei fatta, non smetterò mai di ringraziarti.

Ringrazio i miei amici del team *WeatherStyle*: Angelo, Francesco e Annalaura. Con voi ho trascorso la migliore esperienza universitaria, ci siamo dati forza insieme e siete stati lì ad ascoltare le mie lamentele generali.

Ringrazio il mio amico e collega Salvatore, oltre che compagno di superiori, rimasto al mio fianco durante tutto questo percorso universitario. Mi hai aiutato tanto durante i primi mesi di pandemia, quando ancora avevo difficoltà con la programmazione, oltre che essere un fedele amico a cui raccontare del più e del meno; ti devo molto.

Ringrazio il mio amico e collega Roberto, per le lunghe giornate passate insieme a studiare, come se non ci fosse un domani. Con te tutte le ansie e gli stress derivati da vari esami, soprattutto quelli di matematica, a poco a poco scomparivano.

Ringrazio il mio amico e collega Luigi, oltre che compagno di superiori, per avermi fatto avvicinare al mondo della computer grafica, sei stato di grande ispirazione.

Ringrazio alcuni degli insegnanti delle scuole elementari, medie e superiori che mi hanno permesso di crescere con la motivazione necessaria per proseguire gli studi.

E infine, anche se a volte sono fin troppo duro con me stesso, mi sento di ringraziare la persona che sono. Che questo sia solo l'inizio di un lungo viaggio!

*Questa tesi ha contribuito a piantare un albero in Camerun tramite il progetto Treedom.*

<https://www.treedom.net/it/user/sesalab/event/sesa-random-forest>