

2.1.1 Efficacia dei Quiz nell'apprendimento

All'interno dei *serious games*, uno dei tipi molto diffusi è quello dei giochi basati su quiz. Questo stile di gioco è stato ampiamente studiato per la sua efficacia nell'apprendimento, in particolare è stato trovato più efficace rispetto a simulazioni e avventure [6].

2.2 Le sfide dell'insegnamento della Storia del '900

Come descritto nel Capitolo 2, lo studio della storia è ritenuto poco importante dagli studenti. A questo è unito il problema del programma scolastico molto fitto, che porta ad non avere il tempo di approfondire gli avvenimenti storici più contemporanei, soprattutto nella seconda metà del XX secolo [7].

Per questi motivi, l'insegnamento della storia è un ambito in cui si potrebbero ottenere dei benefici significativi dall'uso di strumenti tecnologici moderni, quali i *serious games*.

Capitolo 3

Strumenti utilizzati

Per sviluppare efficientemente il Serious Game **NOME** è stato necessario selezionare tecnologie attuali, capaci di soddisfare i requisiti del progetto e in grado di garantire robustezza e scalabilità dell'applicazione.

Questo approccio è stato adottato sia durante le fasi di progettazione sia durante lo sviluppo del gioco.

3.1 Strumenti di progettazione

Per aiutare nella gestione del lavoro, durante ogni fase del progetto, è stata utilizzata Trello [8], una piattaforma online per la creazione e mantenimento di *Kanban board*. Le *Kanban board* sono state utilizzate per tenere traccia delle attività da svolgere, di quelle in corso e di quelle completate, permettendo una gestione visiva e dinamica del progresso del lavoro.

Per la creazione dei vari diagrammi UML è stato utilizzato il sito web diagrams.net, scelto per la completezza dello strumento unita alla sua facilità di utilizzo.

Invece, per creare il diagramma di Gantt è stato utilizzato il programma gratuito e multipiattaforma GanttProject [9].

3.2 Strumenti di sviluppo

3.2.1 Game engine e linguaggio di programmazione

Unity [10] è stato scelto come *game engine* su cui sviluppare **NOME**. Questa scelta è stata fatta prendendo in considerazione l'accessibilità e, soprattutto, la facilità di utilizzo, la completezza e la compatibilità del motore di gioco.

Avendo scelto Unity, come *game engine*, il linguaggio di programmazione utilizzato nello sviluppo di **NOME** è stato C#.

C# è un linguaggio di programmazione moderno, che ha permesso di sviluppare l'intera logica del gioco, dalla gestione delle scene alla gestione dei file utilizzati come quiz. Grazie ad esso, in combinazione con Unity, possiamo garantire che **NOME** sia stabile e compatibile su diverse piattaforme, come Windows, Mac e Linux.

3.2.2 Editor di testo e controllo di versione

Unity non ha un editor di testo incorporato, quindi è stato utilizzato Visual Studio Code [11] per scrivere il codice, che è stato poi incorporato nel gioco tramite Unity Editor, strumento che permette un'intuitiva gestione della grafica utilizzata nel progetto.

Come controllo di versione del gioco è stato utilizzato Git [12], grazie al quale è stato possibile mantenere l'integrità del codice, avere una registrazione dell'evoluzione del gioco e soprattutto salvaguardare il progresso svolto anche in caso di incidenti o errori. Per permettere di utilizzare Git su diversi computer, e come forma di *backup* del progetto, è stata scelta la piattaforma gratuita GitHub [13]. La scelta di GitHub è stata fatta in quanto è una piattaforma molto popolare, con ogni funzionalità utile per la gestione di questo progetto e una ottima integrazione con l'editor scelto, Visual Studio Code.

3.2.3 Risorse grafiche e sonore

All'interno del gioco **NOME** sono presenti molti elementi grafici diversi. Questi sprite sono stati presi da OpenGameArt [14], un sito web dove è possibile trovare

molte risorse grafiche con licenze che ne permettono il libero uso, da FlatIcon [15], risorsa fondamentale per trovare icone con licenza gratuita, e dove necessario sono state applicate modifiche alle immagini originali tramite il software Gimp [16], un programma gratuito completo per la modifica di immagini.

Per il progetto è stato utilizzato un font personalizzato chiamato Coolvetica, trovato sul sito web dafont.com e avente anch'esso una licenza ad uso gratuito [17].

Infine la musica di background utilizzata è stata composta appositamente per questo progetto utilizzando FLStudio [18][19].

Capitolo 4

Progettazione

Per lo sviluppo di **NOME** è stato scelto un approccio agile, cioè in cui vengono rispettati i principi del *Agile Manifesto*. Questo significa che il lavoro è stato concentrato sulla collaborazione, sulla flessibilità dei requisiti e sul continuo sviluppo di software funzionanti, anche quando incompleti.

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.” (AgileManifesto.org) []

Per progettare il software mantenendo un modello comprensibile e scalabile, sono stati creati vari documenti di progettazione, tra cui il *Software Development Plan*, il *Software Requirements and Design* e il *Software Test Plan*. Inoltre per descrivere l'architettura del software sono stati modellati vari diagrammi UML, tra cui il diagramma dei casi d'uso e il diagramma delle classi.

4.1 Organizzazione del progetto

Il progetto è stato svolto singolarmente da Sasha Cattaneo, con l'aiuto e la consulenza dei professori di Ingegneria del Software, Gargantini Angelo Michele e Bonfanti Silvia.

Anche se il lavoro è stato svolto in solitaria si è cercato di mantenere un approccio scalabile ad un lavoro in team, per questo è stato mantenuto un definito piano di progettazione ed sono stati usati GitHub, tool fondamentale per garantire il coor-

dinamento tra diversi programmatori, e Trello, anch'esso strumento molto utile a organizzare il lavoro senza sovraccaricare i vari membri di un team.

4.2 Pianificazione del progetto

Il primo passo è stato di scrivere un *Software Development Plan*, un documento scritto per impostare le prime linee guida sulla progettazione di **NOME**.

Il progetto è stato pianificato in fasi lunghe una o due settimane, con ognuna uno specifico obiettivo da raggiungere. Inoltre durante questa fase sono stati scelti gli strumenti utilizzati poi durante il corso di tutto il progetto.

Riprendendo il capitolo 3, il gioco è stato sviluppato utilizzando il *game engine* Unity, per gestire l'intero gioco, e il linguaggio di programmazione C#, per scriverne la logica.

N. Settimana	Fase	Obiettivo
36-37	Sviluppo piano di lavoro	<i>Software Development Plan</i>
38-39	Design del progetto (diagrammi e requisiti)	<i>Software Requirements and Design</i>
40	Inizio sviluppo gioco: Homepage e funzionalità quiz	<i>Alpha 1.0</i>
41	Continuo sviluppo: CRUD quiz	<i>Alpha 2.0</i>
42	Fine sviluppo gioco: rifinitura grafica	<i>Release Candidate</i>
43	Pianificazione test	<i>Software Test Plan</i>
44	Test del software	<i>Release To Manufacturing</i>
45	Scrittura finale tesi	Tesi

Tabella 4.1: Fasi dello sviluppo del serious game

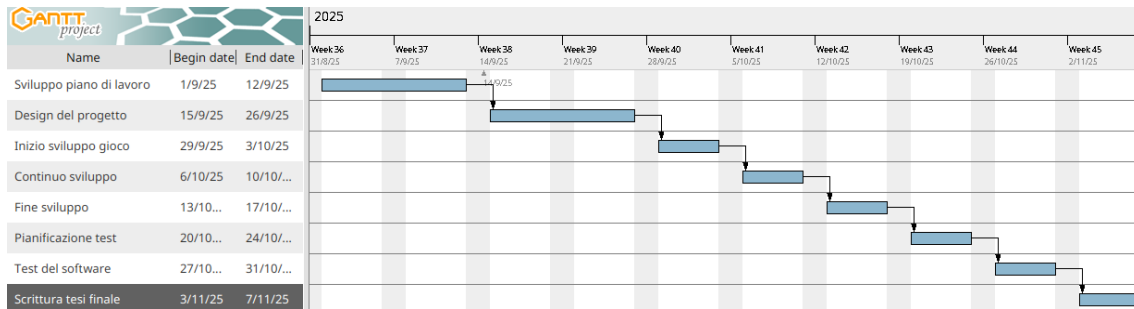


Figura 4.1: Diagramma di Gantt dello sviluppo del serious game

4.3 Analisi dei requisiti

Un passo fondamentale delle prime fasi è stata l'analisi dei requisiti, il cui obiettivo è individuare le funzionalità e i vincoli del progetto.

Per lo sviluppo di **NOME** questi requisiti, divisi tra funzionali e non funzionali, sono stati raccolti nel documento *Software Requirements and Design*. Questo ha permesso di mantenere un approccio ordinato e scalabile da un progetto sviluppato singolarmente ad uno progettato con un team.

4.3.1 Attori del sistema

Per comprendere meglio le esigenze degli utenti, sono state identificate due tipologie principali di attori con necessità diverse:

1. **Insegnante:** Vuole un'interfaccia chiara, che permetta di gestire i quiz (creazione, modifica, eliminazione, *export* e *import*) senza avere competenze di programmazione. Vuole un sistema per condividere i quiz molto semplice da usare.
2. **Studente:** Vuole un feedback immediato sulle risposte (giusto/sbagliato), conoscere il punteggio finale, avere una sfida stimolante (es. la sfida giornaliera) ed avere una meccanica di controllo dei propri progressi.

4.3.2 Requisiti funzionali

I requisiti funzionali descrivono i servizi che il sistema deve fornire agli utenti. Sono stati divisi in base alle aree di funzionalità.

Modalità di gioco

- **Selezione Contenuti:** Il sistema deve permettere la navigazione in un catalogo di quiz e l'avvio di una sessione di gioco con il quiz scelto;
- **Daily Challenge:** Il sistema deve gestire la generazione di un quiz giornaliero, limitando l'accesso al quiz giornaliero ad un tentativo;
- **Streak:** Deve essere tracciata e mostrata la serie di quiz giornalieri completati con successo;
- **Gestione del Feedback:** Il gioco deve mostrare un feedback chiaro, che distingua tra una risposta corretta ed una sbagliata. In particolare, il feedback negativo non deve essere demotivante per l'utente.

Gestione dei quiz

- **Operazioni CRUD:** Il sistema deve gestire le operazioni di *create*, *read*, *update* e *delete* dei quiz;
- **Import/Export:** Il sistema deve permettere l'importazione e l'esportazione dei quiz in modo semplice.

4.3.3 Requisiti non funzionali

I requisiti non funzionali descrivono i criteri di qualità del progetto.

Categoria	Descrizione del Requisito
Usabilità	L'interfaccia utente deve essere intuitiva. Il tempo di apprendimento per un nuovo utente deve essere inferiore ai 5 minuti.
Performance	Il gioco deve essere responsivo, dall'azione dell'utente alla risposta deve passare meno di 1 secondo.
Portabilità dei dati	I dati devono essere serializzati in formato JSON , per essere leggibili e modificabili esternamente.
Portabilità del gioco	Il gioco deve essere compatibile su diverse piattaforme, tra cui Windows, macOS e Linux.
Affidabilità	Il sistema deve gestire correttamente le eccezioni durante l'importazione di file JSON malformati, evitando il crash dell'applicazione.

Tabella 4.2: Requisiti non funzionali del sistema.

4.4 Casi d'uso principali

Il diagramma dei casi d'uso seguente illustra le interazioni principali tra i due attori, i docenti e i loro studenti, e il *serious game*.

Come si vede dal diagramma, i docenti possono creare, modificare, eliminare, importare ed esportare quiz, mentre gli studenti possono selezionare e giocare ai quiz, ricevendo feedback sulle proprie risposte. Anche se non mostrato nel diagramma, anche gli insegnanti possono giocare ai quiz per testarne il funzionamento.

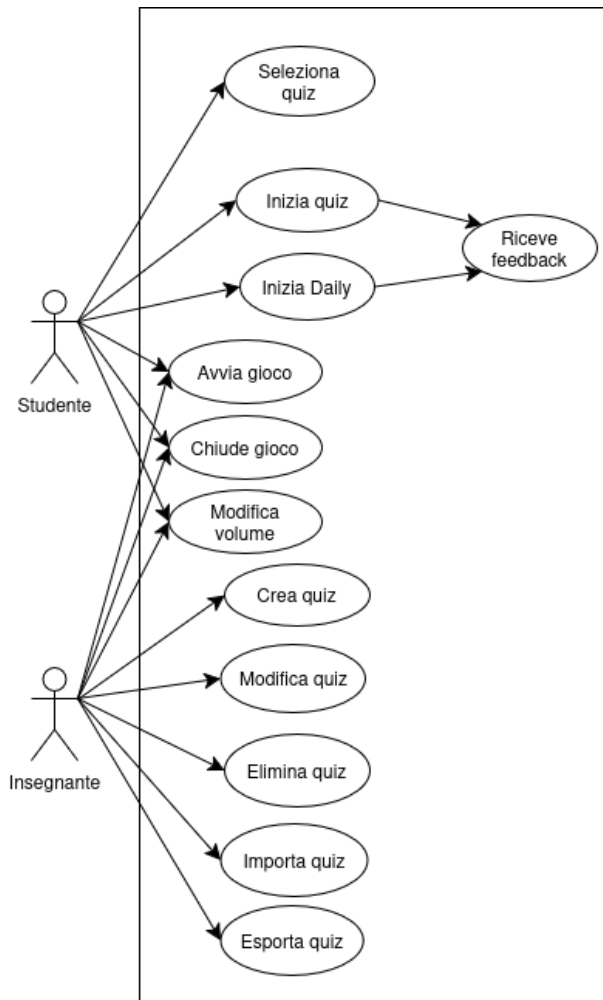


Figura 4.2: Diagramma dei casi d'uso del serious game

4.5 Analisi dei rischi

La scelta di utilizzare il *game engine* Unity ha permesso di prevenire vari problemi di compatibilità che ci sarebbero stati creando il progetto senza *game engine*.

Però questo non assicura la completa assenza di rischi, vulnerabilità e soprattutto di bug nel codice di gioco.

Per prevenire possibili futuri problemi ed errori è stato fondamentale analizzare i potenziali rischi durante la prima fase del progetto.

Questo è consistito nell'identificare e valutare i possibili rischi e, in base alla priorità definita, trovare possibili soluzioni da implementare successivamente.

Nome	Descrizione	Probabilità	Pericolosità	Prevenzione
<i>Scope creep</i>	Provare ad implementare troppi requisiti	Media	Alta	Requisiti ben definiti e chiara pianificazione degli sprint
Ritardi di sviluppo	Ritardi di sviluppo rispetto al piano scelto	Media	Media	Definizione di obiettivi intermedi raggiungibili
Corruzione dei dati di salvataggio	Perdita di dati a causa di un problema	Bassa	Bassa	Implementare backup automatici
Bug nel codice	Presenza di bug nella logica del gioco	Alta	Alta	Implementare test esaustivi durante l'intero sviluppo

Tabella 4.3: Analisi dei rischi del progetto

4.6 Architettura del software

L'architettura del sistema è stata progettata per garantire modularità e scalabilità. È stata applicata una separazione logica delle responsabilità che richiama il pattern *Model-View-Controller (MVC)*, adattato al contesto di un motore di gioco real-time.

4.6.1 Layer logici

Il sistema è suddiviso in tre layer principali:

1. *Data Layer (Model)*: Responsabile della gestione dei dati. Include classi per la rappresentazione di quiz, domande e risposte. Classi principali: *QuestionData*, *GameData* e *JsonHelper*.
2. *Logic Layer (Controller)*: Gestisce il flusso dell'applicazione e le regole di business. Coordina le interazioni tra i dati e la visualizzazione. Classi principali: *QuizManager*, *QuizCRUD* e *QuizHistoryManager*.
3. *Presentation Layer (View)*: Responsabile della visualizzazione dei dati a schermo. Classi principali: *SceneLoader*, *AnswerScript* e *PanelManager*.

4.6.2 Diagramma delle classi

Il diagramma delle classi seguente illustra le principali classi del sistema e le loro relazioni.

Le classi sono organizzate in base ai layer logici sopra descritti, evidenziando le responsabilità di ciascuna classe e le interazioni tra di esse.

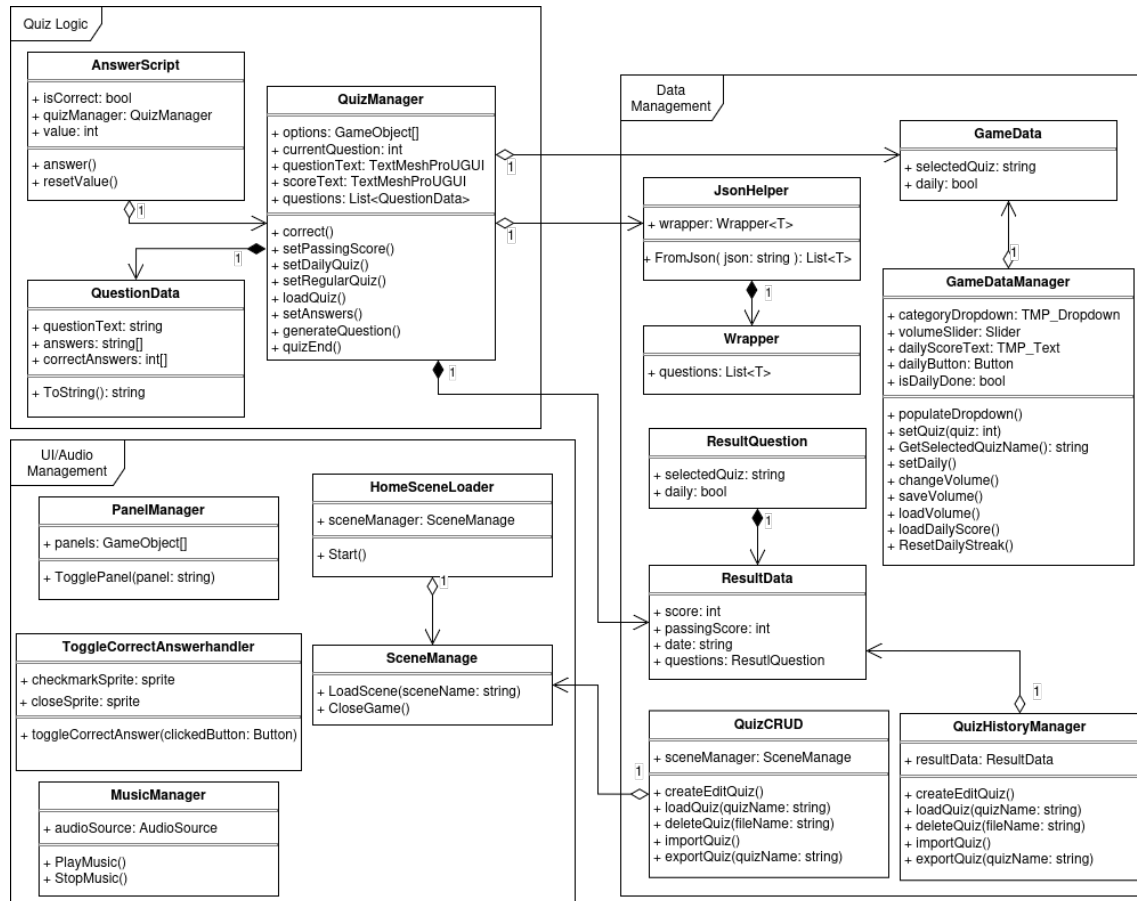


Figura 4.3: Diagramma delle classi del serious game

4.7 Testing

Per assicurarsi il corretto funzionamento di **NOME** è stato indispensabile avere una fase di testing precisa e ben definita. Queste specifiche sono state raccolte nel documento *Software Test Plan*.

La fase di test del software è stata svolta sia in parallelo allo sviluppo del gioco sia come fase indipendente alla fine della scrittura del codice.

Durante lo sviluppo sono stati utilizzati gli strumenti per il debugging disponibili in Unity. Questi sono stati sufficienti per assicurarsi la corretta implementazione di nuovi elementi grafici e audio e di semplici metodi.

Invece per testare il corretto funzionamento del flusso di gioco e di metodi più complicati è stato utilizzato *Unity Test Framework*, un *package* di Unity che implementa attributi specifici per testare elementi specifici di Unity .

Grazie a questo è stato possibile creare i test necessari ed eseguirli in tempo reale tramite la finestra disponibile nel *game engine*.

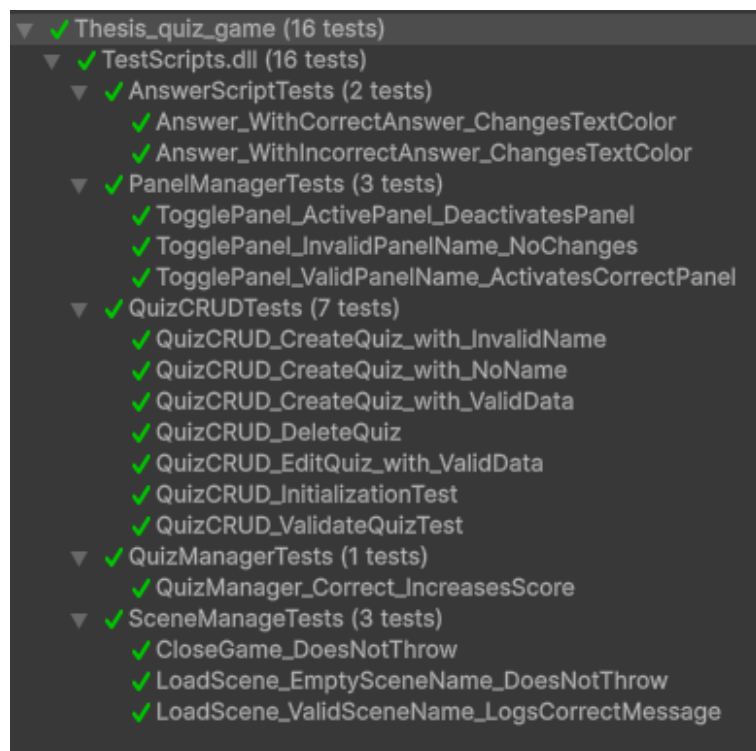


Figura 4.4: Finestra di Unity Test Framework con i test eseguiti

L'ultimo passaggio della fase di testing è stato confermare il corretto funzionamento di **NOME** al di fuori dell'ambiente di sviluppo.

La *build* finale del progetto è stata esportata e testata su tre computer diversi con installati tre sistemi operativi diversi: Linux Mint 22, Windows 10 e macOS Ventura.

Capitolo 5

Gioco sviluppato

Il *serious game* **NOME** consiste in un programma *cross-platform*, sia per computer sia per dispositivi mobili, per la gestione e l'esecuzione di quiz.

Il progetto è stato pensato per un utilizzo scolastico, focalizzato sulla storia e la grafica è stata scelta per avvicinarsi a questo tema.

L'insegnante ha accesso alla finestra di gestione dei quiz, dalla quale ha l'abilità di creare e modificare i quiz per adattarli alla parte del programma che sta coprendo in quel periodo. Successivamente può esportare i quiz creati per darli ai propri studenti. I quali, molto semplicemente, possono importarli e giocarli.

Ogni operazione viene eseguita tramite un'interfaccia grafica semplice ed intuitiva, in modo da permettere a chiunque di utilizzare il gioco senza difficoltà.

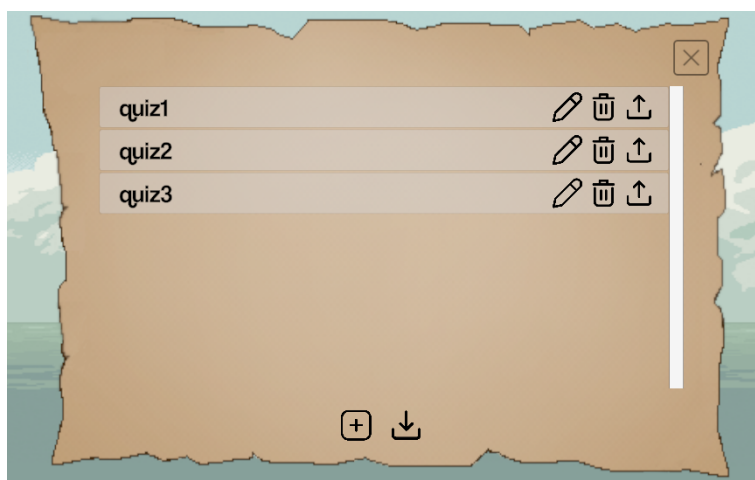


Figura 5.1: Pagina iniziale del gioco.

5.1 Modalità di gioco implementate

Sono state implementate due modalità di gioco:

- La modalità classica: nella quale l'utente, solitamente lo studente, può importare e giocare a dei quiz creati da un altro utente, solitamente il suo insegnante.

Un quiz viene considerato completato con successo se l'utente riesce a raggiungere almeno il 60% del punteggio massimo ottenibile.

- La modalità *Daily*: nella quale l'utente può giocare ad un quiz giornaliero, creato automaticamente in modo da avere un quiz diverso ogni giorno.

Se l'utente riesce a completare con successo il quiz, cioè con almeno 80% del punteggio massimo, aumenterà di uno un contatore dei giorni consecutivi in cui il quiz giornaliero è stato completato con successo. Questo contatore viene chiamato *streak*.

La *streak* viene resettata nel momento in cui il quiz giornaliero viene fallito o al primo giorno in cui non viene completato.

Entrambe le modalità di gioco si basano sullo stesso sistema di quiz, che consiste in una serie di domande a risposta multipla. Ogni domanda ha quattro risposte possibili, di cui una o più corrette. Il numero di domande di un quiz è variabile a discrezione dell'utente che lo crea.

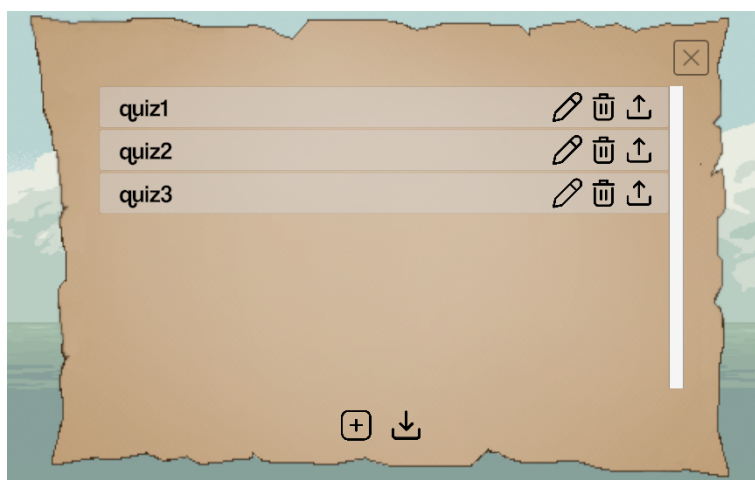


Figura 5.2: Pagina di una domanda durante l'esecuzione un quiz.

5.2 CRUD e Import/Export dei quiz

Per la gestione dei quiz è stata implementata una finestra di CRUD (Create, Read, Update, Delete) che permette all'utente di gestire i quiz presenti nel proprio dispositivo.

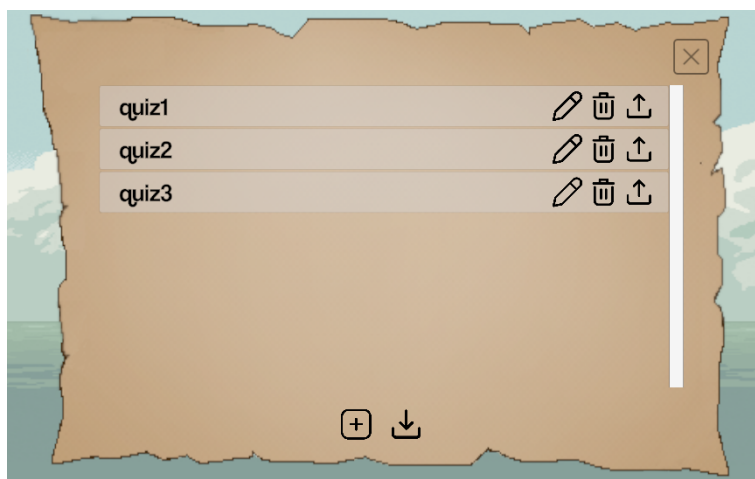


Figura 5.3: Pagina per la gestione dei quiz.

Tramite questa pagina gli insegnanti possono creare o modificare i quiz per renderli adatti ai propri studenti ed al programma che stanno insegnando in quel momento.

Nella creazione di un quiz si può scegliere liberamente il nome, il numero e il contenuto delle domande. Ogni domanda ha quattro risposte e può avere una o più risposte corrette.

La modifica di un quiz permette di modificare il numero e contenuto delle domande e le loro risposte.

Successivamente possono condividere i quiz creati, o modificati, con i propri studenti tramite le funzionalità di *import/export* dei quiz. Durante l'esportazione di un quiz, viene scelta dall'utente la cartella di destinazione. A quel punto utilizzando strumenti intermedi di condivisione, come Microsoft Teams o Google Classroom, l'insegnante può condividere il file esportato con i propri studenti. Gli studenti, una volta ricevuto il file, possono importarlo nel gioco tramite la funzionalità di importazione dei quiz.

Questo sistema permette di avere una gestione semplice e veloce dei quiz, senza la necessità di utilizzare sistemi complessi di gestione degli utenti o di autenticazione.

Inoltre lascia la libertà di scegliere il metodo di condivisione. Questo è un fattore molto importante in ambito scolastico, perché ogni scuola fa uso di piattaforme diverse per la condivisione dei materiali didattici e se questo gioco forzasse l'uso di una piattaforma specifica, potrebbe limitare le scuole in cui può essere utilizzato.

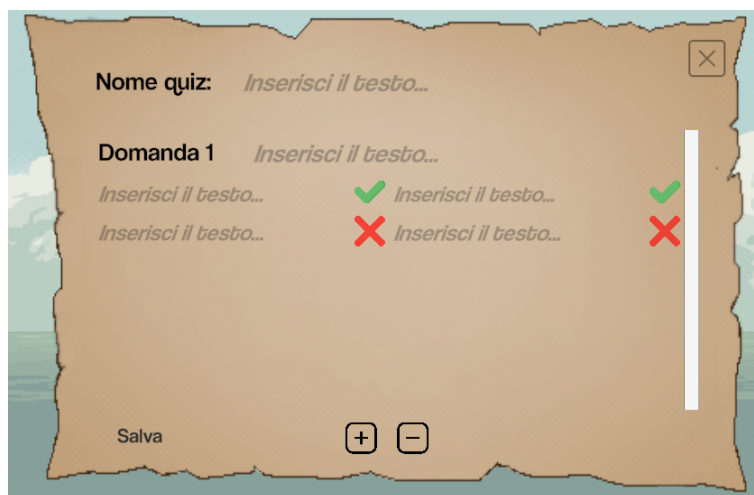
The image shows a digital interface for creating a quiz, designed to look like a piece of aged parchment with a torn edge. At the top left, it says "Nome quiz:" followed by a text input field containing the placeholder "Inserisci il testo...". In the top right corner, there is a small square button with an "X" icon. Below the quiz name, it says "Domanda 1" followed by another text input field with the placeholder "Inserisci il testo...". Underneath this, there are two rows of text input fields, each with the placeholder "Inserisci il testo...". The first row has a green checkmark icon to its right, and the second row has a red "X" icon to its right. To the right of these two rows is a vertical white bar. At the bottom left, there is a "Salva" button. In the bottom center, there are two circular buttons, one with a "+" sign and one with a "-" sign.

Figura 5.4: Schermata di creazione di un quiz.

5.3 Cronologia di gioco

Ogni quiz giocato viene salvato in un file di cronologia, che tiene traccia delle partite giocate dall'utente. Oltre al nome del quiz giocato, vengono salvati anche la data e l'ora in cui è stato giocato, il punteggio ottenuto e ogni risposta scelta, sia se corretta sia se sbagliata.

Per permettere agli utenti di tenere traccia dei propri progressi, è stata implementata una schermata in cui è possibile visualizzare questi dati in modo leggibile.

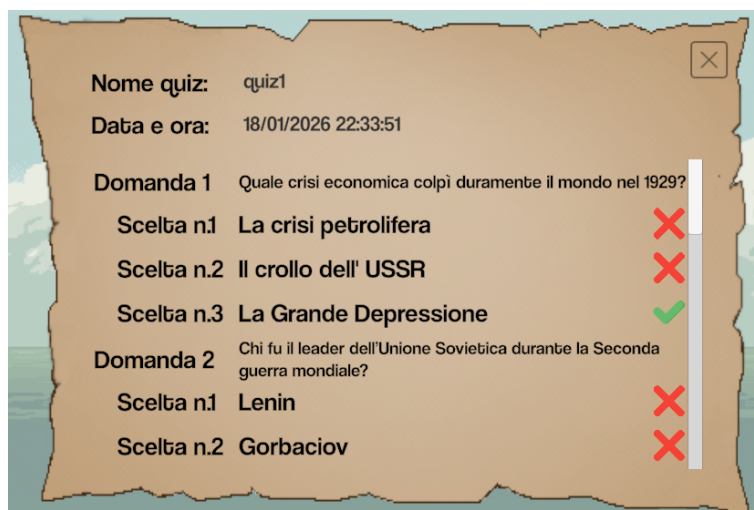


Nome	Data e ora	Risultato	
quiz1	20/01/2026 18:59:06	16 / 18	Q
quiz3	20/01/2026 18:50:40	0 / 12	Q
quiz3	20/01/2026 18:49:52	8 / 12	Q
quiz1	18/01/2026 22:33:51	10 / 18	Q

Figura 5.5: Schermata della cronologia di gioco.

Questa schermata permette di visualizzare tutte le partite giocate, ordinate in ordine cronologico, insieme al risultato ottenuto.

Cliccando sulla lente d'ingrandimento di una specifica partita, viene mostrato un riepilogo dettagliato della partita, con tutte le domande e le risposte date dall'utente, evidenziando quelle corrette e quelle sbagliate.



Nome quiz: quiz1

Data e ora: 18/01/2026 22:33:51

Domanda 1 Quale crisi economica colpì duramente il mondo nel 1929?

Scelta n.1 La crisi petrolifera ❌

Scelta n.2 Il crollo dell' USSR ❌

Scelta n.3 La Grande Depressione ✅

Domanda 2 Chi fu il leader dell'Unione Sovietica durante la Seconda guerra mondiale?

Scelta n.1 Lenin ❌

Scelta n.2 Gorbaciov ❌

Figura 5.6: Schermata che mostra i dettagli di un quiz giocato.

5.4 Gestione dei dati

Per la gestione dei file dei quiz è stato scelto di utilizzare il formato JSON, sia per i file dei quiz sia per i file della cronologia.

Questo formato è stato scelto in quanto è un formato leggibile sia da umani sia da computer, permettendo così ai professori che conoscono il formato di poter modificare i quiz anche senza utilizzare il gioco, mentre per gli altri docenti è stata creata una pagina specifica nel gioco.

Altri vantaggi di questo formato sono la leggerezza, che facilita le operazioni di condivisione dei quiz tra i docenti e i loro studenti, e la possibilità di usare degli strumenti già sviluppati per verificare l'integrità del file e per garantire che si tratti di un quiz valido.

5.5 Meccaniche pedagogiche implementate

Per rendere il gioco coinvolgente e divertente, sono state usate varie strategie di *game design*.

In primo piano c'è il *feedback* immediato dopo ogni risposta dell'utente. In questo modo lo studente capisce subito il risultato della sua scelta e può imparare dagli errori commessi. Il feedback positivo è rappresentato da un cambio del colore del testo della risposta data in verde, mentre il feedback negativo è rappresentato da un cambio del colore del testo della risposta data in rosso e da una breve vibrazione dello stesso testo.

Anche se lo studente sbaglia una risposta, non gli viene rivelata la risposta corretta. Questo per evitare di scoraggiare l'utente e per spingerlo a cercare comunque la risposta corretta.

In base al numero di errori viene però ridotto il punteggio ottenuto. Il punteggio massimo per ogni domanda è 2 punti, se l'utente sbaglia una risposta il punteggio viene ridotto a 1 punto, se sbaglia due risposte il punteggio viene ridotto a 0 punti. Il motivo del punteggio è per incentivare l'utente a rispondere con precisione, ma senza penalizzare troppo gli errori. Inoltre può motivare lo studente a ripetere il

quiz per migliorare il proprio punteggio e la ripetizione è un aspetto fondamentale nell'imparare nuove informazioni.

L'ultimo aspetto pedagogico del gioco sviluppato è la *Daily Challenge*. Questa modalità riprende l'idea dei giochi puzzle online, come ad esempio *Wordle*[20], che propongono un puzzle diverso ogni giorno ed invitano l'utente a tornare ogni giorno grazie al mantenimento della *streak*. Questa strategia funziona perché si tratta di giochi brevi, che impegnano pochi minuti al giorno, e, grazie a questo, riescono a creare una routine di apprendimento quotidiana.

Nel gioco sviluppato, la *Daily Challenge* propone un quiz diverso ogni giorno e mantiene una *streak* che indica il numero di giorni consecutivi in cui l'utente ha completato con successo il quiz giornaliero. Questo incentiva gli studenti a giocare la sfida giornaliera, portando l'apprendimento al di fuori dalla classe.

Inoltre il quiz giornaliero è accessibile solo per un tentativo al giorno, aumentandone la sfida, ed è creato automaticamente, usando la data del giorno come seme per un generatore di numeri casuali. In questo modo ogni utente riceve lo stesso quiz giornaliero, permettendo così agli studenti di confrontarsi e competere tra di loro.

Capitolo 6

Conclusioni

6.1 Considerazioni finali

In conclusione, la creazione di un *serious game* per la gestione e l'esecuzione di quiz in ambito educativo si è rivelata un'idea promettente e realizzabile.

Per testare l'efficacia del gioco, è stato possibile condurre un breve test in una classe di scuola media. Durante questa prova, gli studenti hanno potuto giocare a dei quiz creati dalla loro insegnante.

Il feedback degli studenti è stato generalmente positivo. Soprattutto è stata osservata una buona motivazione dovuta da una spontanea competizione tra amici.

Anche considerando le limitazioni del tesi, il risultato è stato soddisfacente e incoraggiante per futuri sviluppi.

6.2 Limitazioni

Purtroppo il test condotto è stato molto limitato, sia per il numero di studenti e insegnanti coinvolti, sia per la sua durata. Questo non permette di essere sicuri dell'efficacia del gioco in un contesto scolastico più ampio, per cui bisognerebbe condurre altre prove.

Dal punto di vista tecnico, la limitazione principale è la mancanza di varietà nei tipi di domande supportate. Attualmente, il gioco supporta solo domande a risposta multipla, anche se con la possibilità di avere un numero variabile di risposte corrette.

6.3 Sviluppi futuri

Come accennato nelle limitazioni, dei possibili sviluppi futuri possono riguardare l'aggiunta di diversi tipi di domande, come domande a vero/falso, oppure di modalità di gioco alternative.

La grafica è stata scelta per avvicinarsi al tema della storia, ma questo non implica che i quiz debbano essere necessariamente sulla storia.

Un'altra possibile espansione futura potrebbe quindi essere l'aggiunta di diverse categorie tematiche, cioè diverse materie scolastiche , che mantengano la stessa struttura e le stesse funzionalità attuali, ma un design grafico adatto alla materia scelta. Questo renderebbe il gioco più interessante, aumentando la varietà grafica senza dover modificare il funzionamento del gioco.

Bibliografia

- [1] GoStudent. *Future of Education Report 2025*. 2025. URL: <https://www.gostudent.org/static/documents/en-gb/education-report/2025.pdf>.
- [2] Redazione ANSA. *Liliana Segre riceve la cittadinanza onoraria di Venaria Reale*. 2025. URL: https://www.ansa.it/sito/notizie/politica/2025/03/26/liliana-segre-riceve-la-cittadinanza-onoraria-di-venaria-reale_d7ac8de1-02e9-4d34-a6fd-de972ceb2e4b.html.
- [3] Begoña Gros. «Digital Games in Education». In: *Journal of Research on Technology in Education* 40.1 (2007), pp. 23–38. DOI: 10.1080/15391523.2007.10782494.
- [4] Jorge Brandão e Vítor Carvalho. «“GAME QUIZ” - Implementing a serious game platform based in quiz games for the teaching of information and technology». In: *2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV)*. 2014, pp. 47–50. DOI: 10.1109/REV.2014.6784218.
- [5] Yu Zhonggen. «A Meta-Analysis of Use of Serious Games in Education over a Decade». In: *International Journal of Computer Games Technology* 2019.1 (2019), p. 4797032. DOI: <https://doi.org/10.1155/2019/4797032>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2019/4797032>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2019/4797032>.
- [6] Valentin Riemer e Claudia Schrader. «Learning with quizzes, simulations, and adventures: Students’ attitudes, perceptions and intentions to learn with different types of serious games». In: *Computers & Education* 88 (2015), pp. 160–

168. ISSN: 0360-1315. DOI: <https://doi.org/10.1016/j.compedu.2015.05.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0360131515001232>.
- [7] Gianni Oliva. *Scuola, più Muro di Berlino e meno Annibale*. 2023. URL: https://www.lastampa.it/cultura/2023/07/29/news/scuola_piu_muro_di_berlino_e_meno_annibale-12971236/.
- [8] *Trello*. Usato da 2025-10 a 2026-01. URL: <https://trello.com/>.
- [9] *Gantt Project*. Usato il 2025-10. URL: <https://www.ganttproject.biz/>.
- [10] *Unity*. Usato da 2025-11 a 2026-01. URL: <https://unity.com/>.
- [11] *Visual Studio Code*. Usato da 2025-11 a 2026-01. URL: <https://code.visualstudio.com/>.
- [12] *Git*. Usato da 2025-09 a 2026-01. URL: <https://git-scm.com/>.
- [13] *GitHub directory*. Usato da 2025-09 a 2026-01. URL: https://github.com/sasha-cattaneo/HistoryQuiz_unity.
- [14] *OpenGameArt*. Usato da 2025-10 a 2025-11. URL: <https://opengameart.org/>.
- [15] *FlatIcon*. Usato il 2025-10. URL: <https://www.flaticon.com/>.
- [16] *GIMP*. Usato il 2025-10. URL: <https://www.gimp.org/>.
- [17] *Coolvetica Font*. Usato il 2025-10. URL: <https://www.dafont.com/coolvetica.font>.
- [18] *FL Studio*. Usato il 2025-12. URL: <https://www.image-line.com/>.
- [19] Gioele Cattaneo. *Musica di background*. 2025.
- [20] *Wordle*. URL: <https://www.nytimes.com/games/wordle/index.html>.
- [21] Michele Beretta. *M. UniBG LaTeX Thesis Template [Computer software]*. URL: <https://github.com/micheleberetta98/unibg-thesis-template> (visitato il giorno 24/10/2025).