POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

Sviluppo di un Serious Game basato su Escape Room per l'apprendimento della programmazione orientata agli oggetti

Relatori
Prof. Riccardo Coppola
Dott. Tommaso Fulcini

Candidato
Francesco Giacalone
matricola: 290267

Anno accademico 2023-2024

Ringraziamenti

Con grande emozione e gratitudine, desidero esprimere i miei ringraziamenti a tutte le persone che hanno contribuito al raggiungimento di questo importante traguardo, che segna la conclusione del mio percorso di studi.

Innanzitutto , desidero ringraziare i miei relatori, Riccardo Coppola e Tommaso Fulcini , per il notevole aiuto e la grande disponibilità dimostrata nel corso di questo percorso di tesi. Un sentito ringraziamento va anche al mio collega Matteo Biffoni , con il quale ho collaborato per la realizzazione di parte del progetto .

Un ringraziamento speciale va alla mia famiglia e ai miei genitori, Gaetano e Franca, , che hanno sempre creduto in me e mi hanno sempre sostenuto .

Vorrei ringraziare Tonino che fin dalla mia infanzia ha alimentato la mia passione per i videogiochi, una passione che è diventata una parte integrante della mia vita e che mi ha ispirato a intraprendere questo percorso accademico, sperando che possa diventare la mia futura professione.

Grazie anche ai miei amici con cui ho condiviso sogni ,paure e momenti indimenticabili.

Grazie a Sergio compagno di interminabili discussioni filosofiche che hanno illuminato le nostre notti estive , discutendo appassionatamente su come cambiare il mondo.

Infine, vorrei ringraziare tutte le persone che hanno partecipato alla ricerca , dedicandomi il loro tempo e il loro prezioso pensiero critico.

Sommario

Negli ultimi anni, l'avanzamento della tecnologia digitale ha aperto nuove prospettive nell'ambito dell'insegnamento, dando vita a metodologie innovative come i Serious Game. Questi ultimi, spesso associati a videogiochi educativi, mirano a utilizzare in modo significativo le risorse ludiche per scopi didattici piuttosto che puramente ricreativi, secondo la definizione di Sawyer.

Questa tesi propone di sviluppare un Serious Game innovativo basato su un'ambientazione narrativa coinvolgente per facilitare l'apprendimento dei concetti fondamentali della programmazione a oggetti. L'obiettivo è offrire agli studenti un'esperienza educativa interattiva e coinvolgente, specialmente a coloro che si avvicinano per la prima volta alla programmazione.

Per supportare lo sviluppo del gioco, è stata condotta un'approfondita ricerca sullo stato dell'arte dei Serious Game con focus sulla tematica del Coding, analizzando le meccaniche principali, i punti di forza e le debolezze comuni. Sulla base di questa ricerca, è stato ideato e realizzato un Serious Game in stile Escape Room, cercando di superare i limiti emersi dalla letteratura esistente.

Il giocatore si troverà immerso in un mondo governato da un'intelligenza artificiale, basata sulla programmazione a oggetti, che ha imprigionato gli esseri umani in carceri tecnologiche. Gli utenti dovranno acquisire competenze sulla programmazione a oggetti , raccogliendo indizi e risolvendo enigmi , per farsi strada all'interno del carcere e tentare l'evasione. Ogni stanza affrontata durante l'avventura fornirà conoscenze specifiche sulla programmazione a oggetti.

L'implementazione del gioco si propone di raggiungere diversi obiettivi educativi, tra cui la chiara definizione di classe e oggetto, la comprensione della relazione tra metodi e attributi, e la definizione della visibilità degli attributi. L'obbiettivo principale è sfruttare la trama coinvolgente del gioco per fornire un contesto pratico e stimolante per l'apprendimento, rendendo

la comprensione dei concetti di programmazione a oggetti una conseguenza dell'esperienza di gioco.

La dimensione educativa del progetto si integra con innovative meccaniche di gioco, che combinano le tradizionali interazioni di un escape room virtuale con un sistema ispirato alla programmazione a blocchi grafica. Questo sistema consente, ad esempio, di richiamare metodi specifici su oggetti di gioco, offrendo un approccio intuitivo alla manipolazione degli elementi virtuali. Un elemento distintivo è la possibilità per il giocatore di creare direttamente nuovi oggetti nel contesto del gioco, purché possieda la relativa classe. Ciò è reso possibile attraverso un meccanismo di stampaggio che sfrutta una stampante 3D di oggetti virtuali. Inoltre, la creazione della suddetta classe è un processo interattivo, richiedendo al giocatore di raccogliere e connettere in modo accurato i vari metodi e attributi della classe, promuovendo così una comprensione pratica e interattiva dei concetti di programmazione a oggetti.

Indice

Elenco delle tabelle

Elenco delle figure

Introduzione

Negli ultimi anni, nell'ambito accademico e dell'istruzione , c'è stato un crescente interesse nella ricerca di metodi di insegnamento efficaci ed innovativi . In questo contesto, i serious game si sono affermati come strumenti educativi promettenti, capaci di trasmettere conoscenze complesse in modo coinvolgente e interattivo. I serious game sono giochi progettati specificamente per scopi educativi, mirando a rendere l'apprendimento più accessibile, stimolante e memorabile. Secondo la definizione di Clark C.Abt , "i Serious Game sono giochi con uno scopo educativo esplicito e ben strutturato, non pensati principalmente per il divertimento, pur senza escluderlo."

La presente tesi si concentra sullo sviluppo di un serious game dedicato all'insegnamento del paradigma della programmazione a oggetti a un pubblico giovane, principalmente studenti delle scuole medie e superiori, con limitata o nessuna esperienza in materia.

Prima di presentare il serious game proposto, è essenziale sottolineare che la ricerca condotta per questo progetto ha coinvolto un'analisi approfondita dello stato dell'arte dei serious game per la programmazione. Tale analisi ha permesso di individuare le migliori pratiche e le sfide esistenti nel settore, offrendo una solida base teorica e metodologica .

Il gioco proposto si inserisce nel genere delle escape room, caratterizzato da enigmi e sfide da risolvere per raggiungere un obiettivo finale. La trama ruota attorno alla storia di un prigioniero intrappolato in un carcere gestito da un'intelligenza artificiale programmata con il paradigma della programmazione a oggetti. Il protagonista, il prigioniero, dovrà apprendere i concetti fondamentali della programmazione a oggetti per tentare di evadere dalla prigione.

Gli argomenti educativi vengono introdotti attraverso una serie di minigiochi specifici, progettati per offrire una comprensione chiara e intuitiva dei concetti di base della programmazione a oggetti. Ogni minigioco è stato concepito per essere coinvolgente e divertente, rendendo l'apprendimento una naturale conseguenza del gioco.

Nei capitoli successivi, saranno approfonditi gli obiettivi, il design e l'implementazione del gioco , insieme ai risultati ottenuti .

Il Capitolo 1 introduce i serious game, analizzando l'origine e l'evoluzione nel tempo, ed evidenziando punti di forza e criticità. Inoltre, viene eseguita un'analisi dello stato dell'arte dei serious game incentrati sulla programmazione , analizzando le meccaniche principali con cui vengono introdotti gli argomenti educativi.

Il Capitolo 2 presenta il design del serious game proposto in questa tesi, delineando obiettivi, trama di gioco, e meccaniche principali , oltre a fornire un elenco delle tecnologie utilizzate e una descrizione concettuale dell'implementazione di base .

Il Capitolo 3 espone le schermate dell'implementazione del videogioco e valuta l'esperienza utente tramite il Gamex , un questionario di 27 domande volte a valutare la user expierence in contesti di gamification .

Infine, il Capitolo 4 discute la valutazione dell'esperienza utente , i punti deboli e i possibili sviluppi futuri.

Capitolo 1

Stato dell'arte

1.1 Serious e applied games

1.1.1 Cosa sono i Serious Game

Il gioco può essere considerato come un'esperienza di intrattenimento che coinvolge sia l'aspetto mentale che fisico. Si basa su regole specifiche e ha un risultato verificabile, come ad esempio una condizione di vittoria o sconfitta. Inoltre, il gioco richiede l'applicazione di abilità specifiche da parte dei partecipanti [17] . I giochi si distinguono per il loro carattere di finzione e sono inseriti in una cornice spazio-temporale chiamata "Cerchio Magico". In questo contesto, prevalgono regole specifiche che differiscono da quelle del mondo reale [6] . Il divertimento derivante dal tentativo di colpire un bersaglio con una palla potrebbe rappresentare un'esperienza interessante, ma isolatamente questa attività non può essere categorizzata come un gioco. Tuttavia, attraverso l'introduzione di specifiche regole, si trasforma da un'attività informale a un vero e proprio gioco. Ad esempio, stabilendo un campo da gioco di dimensioni specifiche (ad esempio, 105x65 metri) e identificando bersagli rettangolari (7x3 metri), limitando il modo in cui la palla può essere colpita (ad esempio, solo con i piedi), e istituendo un criterio quantificabile per l'assegnazione dei punti (come la squadra che fa entrare il pallone nella porta avversaria guadagna un punto), si configura questa attività come gioco del calcio. Fin dall'antichità, oggetti come palloni, dadi e carte sono stati utilizzati per creare giochi attraverso la definizione di regole specifiche. Con l'avvento dei sistemi informatici nell'era moderna, come computer, smartphone, console, ecc., il medesimo principio è stato applicato. Utilizzando tali dispositivi con regole specifiche e delineando criteri quantificabili per attività

digitali, sono stati creati i cosiddetti giochi digitali o videogiochi.

Secondo la definizione dell'enciclopedia Treccani [54], " il videogioco è un gioco gestito da un dispositivo elettronico che consente di interagire con le immagini di uno schermo " . Il termine "videogioco" solitamente si riferisce a un software , ma può essere implementato anche tramite dispositivi hardware specifici , come i cabinati delle sale gioco .Il termine "videogiocatore" o "gamer" è utilizzato per indicare chi fruisce di un videogioco .Il videogiocatore utilizza una varietà di periferiche di input per interagire con il videogioco . Queste includono i tasti e i controller nel caso dei cabinati, il gamepad nelle console di gioco, il mouse e la tastiera quando si gioca direttamente su un computer. Il videogioco ha avuto origine a partire dagli anni cinquanta del Novecento, inizialmente nei contesti di ricerca scientifica e nelle università americane. Tuttavia, il suo sviluppo commerciale significativo è iniziato negli anni settanta .

Il settore del gaming sta vivendo una crescita costante, secondo i dati riportati da FinanceCommunity [20] , il mercato videoludico nel 2022 valeva 336 miliardi di dollari e si prevede raggiungerà i 522 miliardi entro il 2027. Parallelamente, il numero di giocatori continua a crescere, passando dagli attuali 3,2 miliardi a circa 3,6 miliardi di persone entro il 2025. Questo settore è al centro di un costante sviluppo tecnologico, e l'innovazione è alimentata anche dalle startup del gaming, le quali stanno attirando un crescente interesse degli investitori. Nel solo 2022, tali startup hanno raccolto oltre 13 miliardi di dollari da fondi di Venture Capital .

L'interesse crescente nel settore videoludico non solo è alimentato dal continuo sviluppo tecnologico, che porta a sistemi sempre più realistici ed immersivi, ma anche dalle abilità dei game designer di coinvolgere profondamente gli utenti, trasportandoli in uno stato di intenso coinvolgimento emotivo e distogliendoli dal mondo reale. Questo stato mentale, noto come "flow", implica un completo assorbimento nell'attività in corso e comporta una totale concentrazione, l'attivazione di automatismi, la discriminazione degli stimoli incoerenti con l'attività e una percezione di controllo totale. Il risultato è un'ottimizzazione delle prestazioni dell'utente. In sostanza, il "flow" rappresenta la capacità del gioco di generare una sensazione di benessere e automatismo nell'utente, trasportandolo dalla realtà effettiva a quella digitale. [8]

Come evidenziato dal grafico, il "flow" è un equilibrio delicato che richiede una giusta combinazione tra il livello di difficoltà di un compito e le abilità del giocatore. Un livello di difficoltà eccessivamente alto rispetto alle capacità del giocatore può generare ansia, rabbia, frustrazione e persino rinuncia.

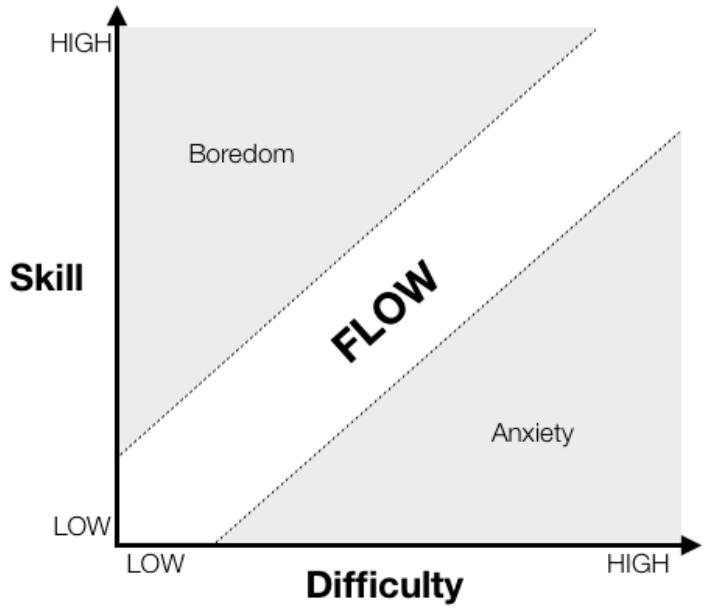

Figura 1.1. Flow Diagram

D'altra parte, un livello di difficoltà troppo basso rispetto alle capacità può causare noia e apatia. Il corretto bilanciamento di questi due fattori è essenziale per suscitare nel giocatore motivazione, divertimento e uno stato di coinvolgimento profondo. Quando le sfide proposte sono in sintonia con le abilità del giocatore, si crea un ambiente di gioco ottimale che stimola l'esperienza di "flow" e massimizza il coinvolgimento del giocatore. [28] .

I giochi, che siano digitali o tradizionali (per esempio i giochi da tavolo), hanno la capacità di immergere i giocatori nel suddetto stato mentale di "flow". Questo stato non solo contribuisce al puro divertimento, ma può essere impiegato per raggiungere obiettivi più seri. Questo principio viene sfruttato anche in contesti non ludici per favorire l'apprendimento, l'efficienza lavorativa e la concentrazione.

Alcuni giochi tradizionali sono stati adoperati con finalità più serie rispetto al semplice intrattenimento. Ad esempio, il gioco da tavolo Monopoly è stato concepito con l'intento di fungere da strumento didattico per illustrare gli effetti negativi dei monopoli sull'economia. Similmente, i giochi sportivi, come il calcio, possono essere praticati non solo a scopo ludico, ma anche per migliorare la forma fisica e promuovere la salute. [17] .

Partendo da questo presupposto, è possibile estendere lo stesso concetto ai

giochi digitali, sfruttando lo stato mentale di flow, al fine di perseguire obiettivi più ampi rispetto al semplice divertimento. I giochi digitali che incorporano questa capacità sono comunemente definiti *Serious Game* o iApplied Game.

Il termine *serious game* fu utilizzato per la prima volta da Clark C. Abt, uno sviluppatore di giochi militari su computer, nel 1971. Nel suo libro "Serious Games" , [4], Abt fornisce anche una definizione di giochi educativi, indipendentemente dal supporto (digitale o real-world). Egli li descrive come "giochi con uno scopo educativo esplicito e ben strutturato, non pensati principalmente per il divertimento, pur senza escluderlo." Oggi, il termine *Serious Game* è ampiamente associato a videogiochi con finalità educative, come suggerisce la diffusa definizione di Sawyer: "qualsiasi utilizzo significativo delle risorse dei giochi informatici il cui principale scopo non è l'intrattenimento". [41].

Negli ultimi anni, sono state proposte molteplici definizioni per descrivere i Serious Game. Nonostante le diverse origini di ciascuna definizione, il significato fondamentale è che i Serious Game sono giochi che possono essere utilizzati per ragioni diverse dal mero divertimento. [52].

L'apprendimento non è l'unico obiettivo caratterizzante dei serious games. Questi possono essere suddivisi in categorie in base ai loro obiettivi caratterizzanti. Ad esempio, gli exergame incoraggiano le persone a diventare fisicamente attive e a sostenere uno stile di vita sano, mentre gli advergame vengono utilizzati per scopi di marketing o di reclutamento, aumentando la consapevolezza dei giocatori su determinati argomenti. Gli obiettivi caratterizzanti dei serious games odierni comprendono anche il cambiamento del comportamento nello stile di vita, la diagnosi medica, la gestione aziendale, il supporto alle decisioni, lo sviluppo delle abilità sociali, l'analisi dei meccanismi causali, la creazione e difesa di argomentazioni, lo sviluppo di strategie di risoluzione dei conflitti, l'eccitazione della fantasia, l'elevazione dell'impegno civico, la promozione dei valori etici, la persuasione e il reclutamento per cause, campagne politiche e molto altro ancora. [17].

1.1.2 Differenza tra Serious Game e Gamification

I serious game devono inoltre essere distinti dalla gamification. Secondo l'enciclopedia Treccani la gamification è: "l'utilizzo di meccanismi tipici del gioco e, in particolare, del videogioco (punti, livelli, premi, beni virtuali, classifiche), per rendere gli utenti o i potenziali clienti partecipi delle attività di un sito e interessarli ai servizi offerti" [53].

Ad esempio," Starbucks Rewards [48] è un programma fedeltà dell'azienda Starbucks che premia i clienti con stelle per gli acquisti effettuati presso Starbucks. Quando i clienti accumulano stelle, possono sbloccare premi come bevande, cibi e prodotti gratuiti. Il programma prevede anche sfide e opportunità di bonus per incoraggiare i clienti a visitare più spesso e a fare acquisti più consistenti."

Pertanto, il risultato della gamification non è un gioco, ma piuttosto l'incorporazione di elementi ludici in contesti non ludici per motivare e coinvolgere gli utenti. Preso alla lettera, il termine gamification significa "fare un gioco di qualcosa che non è un gioco". In particolare, concetti e/o elementi basati sui giochi vengono utilizzati per "gamificare" applicazioni non di gioco esistenti [17].

1.1.3 Serious Game History

La nascita dei Serious Game può essere fatta risalire alle simulazioni di guerra dell'esercito prussiano nei primi anni del XVIII secolo [25].Tuttavia , le prime simulazioni militari con l'utilizzo di computersi si registrarono negli Stati Uniti negli anni cinquanta, presso la Johns Hopkins University [34].

Per esempio, Abt , [4], menziona un gioco per l'addestramento degli ufficiali sviluppato nel 1961. Un esempio più recente è rappresentato da America's Army, lanciato nel 2002. Si tratta di un videogioco militare che immerge i giocatori in situazioni di combattimento realistiche. È stato creato dall'esercito americano con l'obiettivo di sostenere il reclutamento dei giovani, presentando armi realistiche e consentendo ai giocatori di vestire le uniformi dei soldati di fanteria statunitensi. Inoltre, i giocatori di maggior successo possono ricevere un invito dall'ufficio di reclutamento dell'esercito. In realtà, già negli anni sessanta, l'esercito americano aveva istituito un'agenzia denominata "Joint War Games Agency" con l'obiettivo di sviluppare giochi per scopi militari [17].

Durante gli anni 2000, si è verificato un significativo aumento nella diffusione di vari tipi di giochi educativi, soprattutto quelli progettati per il pubblico più giovane. Nel 1999, LeapFrog Enterprises ha lanciato il LeapPad, un dispositivo che combinava un libro interattivo con una cartuccia, consentendo ai bambini di giocare e interagire con un libro cartaceo. Sfruttando il successo dei tradizionali sistemi di gioco portatili come il Nintendo Game Boy, nel 2003 hanno lanciato anche il sistema di gioco portatile chiamato Leapster. Questo sistema , basato su cartucce , offriva giochi in stile arcade integrati con contenuti educativi [18].

Nel periodo compreso tra il 1980 e il 1990, sono stati sviluppati da due a quaranta nuovi serious game all'anno. Successivamente, tra il 1990 e il 2002, il numero è aumentato fino a raggiungere i 60-80 giochi all'anno. Negli anni successivi, si è assistito a un ulteriore incremento, con un numero variabile tra i 70 e i 240 serious game prodotti annualmente, registrando un aumento significativo dopo il 2007 [17]. Fino al 2010, i serious game hanno registrato una evoluzione notevole, includendo economie reali come in Second Life. In questo contesto, gli utenti hanno la possibilità di creare attività reali che offrono beni e servizi virtuali, utilizzando i dollari Linden, che sono scambiabili con la valuta statunitense [32].

L'innovazione successiva è arrivata nel 2010 con Kinect di Microsoft, in cui il corpo umano diventa il principale dispositivo di interazione. La console è dotata di una fotocamera e di un sensore di profondità a infrarossi che rileva le articolazioni del corpo umano in tempo reale. Questo tipo di interazione si è rivelato eccellente per l'esercizio fisico, ed è stato impiegato anche nelle scuole americane per gli allenamenti di danza [17]. I mercati principali per i serious games sono il Nord America, il Giappone, la Corea del Sud e l'Europa. Mentre negli Stati Uniti i bambini erano considerati i principali destinatari, in Giappone ed Europa lo sviluppo mirava già al mondo degli adulti. Ad esempio, Brain Training del Dr. Kawashima è stato un popolare gioco sviluppato di Nintendo in Giappone , per allenare la mente con calcoli aritmetici e sfide mnemoniche. Una specialità europea sono i giochi per l'arte e la cultura, con l'obiettivo di aumentare la conoscenza del patrimonio culturale nei paesi europei. Gli esempi includono Versailles 1685 e Vikings[17].

1.1.4 Aspetti positivi e negativi

Come accennato in precedenza, i Serious Game possono rappresentare uno strumento estremamente utile, soprattutto in campi come l'insegnamento, lo sviluppo di abilità e la responsabilizzazione degli individui. Numerosi sono i fattori che rendono i serious game uno strumento interessante e differente da altri, tra i più significativi si citano:

- LearnbyDoing

  Coinvolgono gli utenti in uno stato di apprendimento attivo, a differenza delle forme di insegnamento tradizionali in cui lo studente assume un ruolo prevalentemente passivo (lezioni, documentari, ecc.). Un coinvolgimento attivo stimola il pensiero critico. [23] .

- Fun factor

  Il divertimento costituisce il cuore dei serious game, consentendo agli utenti di affrontare argomenti "seri" in modo divertente.

- Narrazione

  La narrazione di una storia interessante e coinvolgente contribuisce ad aumentare il livello di divertimento, interesse e motivazione dell'utente verso specifici argomenti.

- Motivazione

  Sfruttando un coinvolgimento attivo, i serious game possono generare un livello di motivazione negli utenti su specifici argomenti superiore a quello ottenuto con i metodi classici.

- Feedback

  I serious game forniscono feedback immediato alle azioni degli utenti e sono in grado di valutare i progressi in modo anonimo, poiché la valutazione è effettuata da una macchina. Questa caratteristica riduce lo stress e l'imbarazzo degli utenti, poiché non devono affrontare la valutazione di un altro essere umano. Nei serious game, uno stato di vittoria/sconfitta è progettato per far riflettere al giocatore sulle sue scelte. Uno stato di sconfitta può generare dissonanza cognitiva (o conflitto) all'interno di un individuo, contribuendo così all'esperienza di apprendimento.[6] .

- Ambienti controllati

  I serious game forniscono un ambiente sicuro per l'apprendimento, consentendo agli utenti di sperimentare attraverso tentativi ed errori. Ad esempio, i serious game militari consentono ai soldati di essere addestrati per situazioni di guerra senza rischiare vite umane o causare feriti.

Nonostante i numerosi vantaggi dei serious game, è importante riconoscere che presentano anche alcuni svantaggi, sia intrinseci che derivanti dall'implementazione e dal design. L'etichetta stessa di "Serious Game" può costituire un paradosso, poiché l'essenza fondamentale del gioco suggerisce intrattenimento e leggerezza. Questo dualismo, dove il termine "serio" potrebbe contrastare con l'aspettativa di divertimento, potrebbe portare a una demotivazione dei giocatori. " I giocatori potrebbero percepire un gioco serio come un debole tentativo di avvolgere qualcosa che non è piacevole in una bella scatola " [17].

Il fatto che qualcosa sia designato come un gioco non implica automaticamente che risulti divertente . Se l'obiettivo principale dell'intrattenimento viene trascurato, l'esperienza di gioco potrebbe risultare negativa.

Alcuni degli aspetti negativi, condivisi con i videogiochi tradizionali, includono problemi dal carattere prettamente mentali come la dipendenza e l'isolamento sociale, oltre a problemi fisici quali affaticamento degli occhi, mal di testa e possibili infortuni nei giochi che coinvolgono movimenti del corpo. Ulteriori problemi specifici possono derivare dal design e dall'implementazione del serious game. La creazione di un serious game di qualità non è un compito semplice, soprattutto considerando i costi elevati associati. [23].

In secondo luogo, la complessa relazione tra il game design e l'educational design può portare alla realizzazione di giochi che non rispecchiano gli obiettivi prefissati. Ad esempio, nel serious game PollutionRunner [22] , (un Endless Runner 3D), lo scopo è sensibilizzare gli utenti sull'inquinamento dell'aria. L'ambiente di gioco è costruito in base ai dati dei sensori meteorologici più vicini: più è inquinato l'ambiente, più è difficile giocare, con l'apparizione di nuvole di smog, visuale offuscata, ecc. In questo caso, dunque, più l'aria è inquinata, più aumenta la difficoltà del gioco, rischiando di creare un coinvolgimento nell'utente nel giocare e trasmettere un messaggio errato di correlazione tra inquinamento e divertimento.

1.1.5 Problemi attuali nello sviluppo di Serious Game

Progettare un Serious Game efficace è una sfida complessa. " La narrativa, il gameplay e un ambiente virtuale accattivante (design e grafica) sono elementi chiave per il successo di un serious game "[6].

Attualmente, la sfida più significativa che il settore deve affrontare è la disconnessione tra il game design tradizionale e il design educativo [52] .

La principale difficoltà nel raggiungere un design ottimale risiede nella natura interdisciplinare della progettazione di un serious game. Tale processo richiede il contributo di esperti provenienti da diverse aree, tra cui graphic design, product design, programmatori, animatori, scrittori e designer audio. Integrare la progettazione educativa con il game design è una sfida significativa che implica la collaborazione di esperti provenienti da campi diversi. Inoltre, questo coinvolgimento multidisciplinare comporta anche costi di sviluppo elevati.

In modo più specifico, il successo dell'industria dei videogiochi dimostra chiaramente che i game designer sono capaci di creare esperienze molto coinvolgenti. Tuttavia, il coinvolgimento, così come definito dal concetto di flow, potrebbe non essere sufficiente per garantire l'efficacia di un gioco a fini di apprendimento. D'altra parte, gli esperti del settore educativo possiedono una solida comprensione di come progettare esercizi su un determinato contenuto accademico, ma potrebbero non avere le competenze necessarie per tradurre questi contenuti in attività di gioco coinvolgenti. Trovare un equilibrio tra apprendimento e divertimento rappresenta quindi una sfida significativa.

Per affrontare questa sfida, uno studio è stato condotto [52], coinvolgendo 75 insegnanti in un corso di game design. L'obiettivo principale dello studio era individuare le difficoltà che gli insegnanti incontrano durante la progettazione di un serious game. La valutazione della qualità dei progetti è stata basata sull'analisi della presenza di elementi di gioco e delle loro interazioni. I risultati indicano che, nonostante alcuni progetti fossero soddisfacenti, l'inserimento degli insegnanti nel game design è, nel complesso, impegnativo, così come lo è il trasferimento delle competenze pedagogiche nel contesto del game design.

L'analisi della qualità dei progetti, focalizzata sugli elementi di gioco e sulle loro interazioni, evidenzia che, nonostante le competenze nell'ambito educativo, gli insegnanti hanno incontrato difficoltà nell'integrare in modo efficace gli elementi di gioco più critici, come meccaniche, risorse e ostacoli, nei loro progetti. Le principali difficoltà sono emerse nella definizione di ostacoli capaci di migliorare l'esperienza di apprendimento del giocatore e nella creazione di risorse che possano essere impiegate per superare tali ostacoli all'interno del gioco. Inoltre, hanno riscontrato notevoli difficoltà nel collegare in modo coerente questi elementi di gioco al fine di sostenere l'apprendimento.

È fondamentale sottolineare un ulteriore limite attuale, ovvero la mancanza di strumenti di valutazione adeguati per analizzare i serious games e la conoscenza ancora insufficiente del loro impatto sui giocatori. In particolare, le questioni relative alla qualità della progettazione concettuale formale dei serious games in relazione ai loro scopi rimangono per lo più inesplorate. [36].

Lo scopo dei serious games si riflette direttamente nel fine del gioco e nel suo argomento, ma anche nelle intenzioni dei designer e nel loro obiettivo di avere un impatto specifico sui giocatori. In breve, se un serious game non riesce ad influenzare il giocatore nel mondo reale e nel contesto della vita quotidiana, perde il suo scopo fondamentale.

E' essenziale riconoscere che i giocatori portano le proprie intenzioni e

scopi all'esperienza di gioco e potrebbero intendere un gioco diversamente da quanto previsto dai progettisti. In questo senso, lo scopo non può mai essere "trasferito" direttamente come previsto nel suo design, ma nel gioco la struttura influisce sullo spazio di possibilità. Di conseguenza, l'intenzione esplicita e lo scopo del gioco devono essere considerati in tutti i componenti della loro progettazione [36] .

Come già sottolineato, la creazione di un serious game richiede una conoscenza approfondita del dominio di apprendimento e un chiaro obiettivo educativo finale per conseguire i risultati desiderati. Tuttavia, questa conoscenza potrebbe non essere sufficiente per sviluppare un serious game di successo. Inizialmente, è essenziale che il giocatore percepisca il gioco come un'esperienza ludica, dove l'apprendimento è una conseguenza naturale delle azioni di gioco. In assenza di questa percezione, il gioco rischia di essere considerato noioso anziché un'attività divertente e coinvolgente.

Per mantenere alto il livello di divertimento, l'articolo [7] propone una metodologia di design che dovrebbe essere implementata a ogni livello del gioco, consentendo così la distribuzione dei contenuti didattici in tutto il contesto di gioco. Secondo questa metodologia, ai giocatori vengono assegnate diverse missioni da completare per ogni livello, affrontando contemporaneamente diversi meccanismi di apprendimento per assimilare i concetti associati a ciascun livello del gioco. In sintesi, questa metodologia suggerisce due componenti principali per la progettazione e lo sviluppo di un serious game: un gioco principale con missioni e una serie di meccanismi di apprendimento. Questi ultimi sono legati al gioco principale ma sono autonomi e vengono affrontati in parallelo al gioco principale, semplificando l'inclusione di contenuti didattici mediante la definizione di meccanismi di apprendimento indipendenti.

1.2 Stato dell'arte di serious games per coding

Per sostenere il design e lo sviluppo del progetto, è stata condotta una ricerca sullo stato dell'arte attuale dei serious game nell'ambito della programmazione. Questo studio ha abbracciato una vasta gamma di giochi con obiettivi che spaziano dall'insegnamento dei concetti basilari di programmazione a quelli che approfondiscono argomenti più avanzati della programmazione ad oggetti.

- CodinGame

  Si tratta di un sito che offre diversi giochi con una struttura comune. Da una parte, è presente un editor in cui è possibile scrivere codice nel linguaggio di programmazione desiderato. Dall'altra parte, vi è un gioco in cui l'obiettivo è redigere il codice corretto per superare il livello. Quando il codice viene scritto e si preme "play", vengono eseguiti dei test, mostrando l'effetto che il codice ha sul gioco. Ad esempio, potrebbe esserci un'astronave che si muove in orizzontale e si abbassa di un livello ad ogni passaggio. Con la presenza di montagne che ostruiscono il suo cammino, il giocatore deve scrivere il codice per sparare alla montagna più alta durante il passaggio, così da evitare collisioni e raggiungere il suolo con successo.

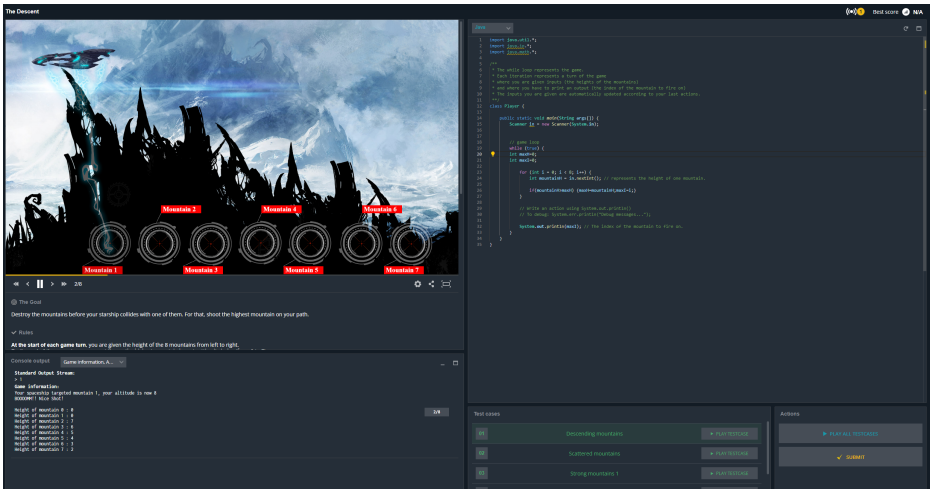

Figura 1.2. THE DESCENT , uno dei vari giochi di CodinGame

Il gioco fornisce un codice base con i comandi essenziali, ad esempio indicando che la funzionalità di sparo avviene tramite l'uso di println con un intero che rappresenta l'indice della montagna. L'utente è quindi sfidato a completare il codice e renderlo funzionante. Questo approccio è presente in diversi esempi simili sul sito, ciascuno con situazioni e difficoltà diverse. [13]

- Lost in Space

  Con meccaniche di scrittura e compilazione del codice simili a quelle di CodinGame, possiamo individuare Lost in Space, un Serious Game descritto nell'articolo "Building a Scalable Game Engine to Teach

Computer Science Languages" [44].

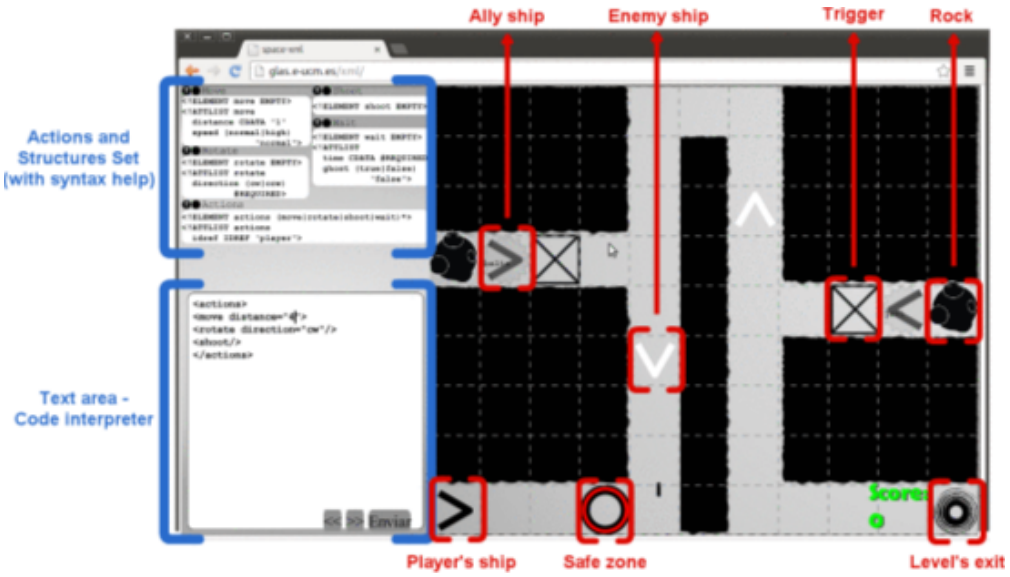

Figura 1.3. Lost in Space , gameplay.

Lost in Space è un serious game progettato per apprendere linguaggi di programmazione. L'obiettivo del gioco è far arrivare la propria navicella spaziale nella casella finale di gioco , rappresentata da un buco nero, evitando nemici, rocce e altri ostacoli. L'unico modo per muoversi e compiere azioni è scrivere del codice, il quale sarà successivamente compilato e interpretato in istruzioni di gioco per far muovere la navicella.

Il gioco si avvale di un interprete 1.4, capace di tradurre istruzioni da un comune linguaggio di programmazione in azioni di gioco. Questa caratteristica consente di aggiungere al gioco qualsiasi linguaggio si desideri apprendere, poiché le azioni sono già programmate; è sufficiente creare un nuovo interprete per il nuovo linguaggio.

- Java Tower Defense

  Si tratta di un Serious Game per dispositivi mobile descritto nell'articolo "A mobile-device based serious gaming approach for teaching and learning Java programming" [30] .

  Il gioco si presenta come un tower defense, richiedendo ai giocatori di scrivere il codice per vari GameObject. Ad esempio, gli utenti possono trovarsi ad affrontare una missione in cui devono creare una torre e posizionarla sul campo utilizzando il codice Java. La scrittura del codice

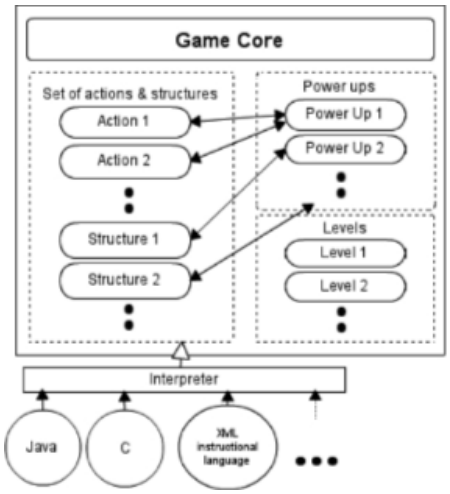

Figura 1.4. Lost in Space , interprete.

Figura 1.5. JavaTowerDefense

non avviene attraverso un tradizionale editor, ma mediante l'utilizzo di bottoni contenenti parole chiave preimpostate.

- AlgoGame

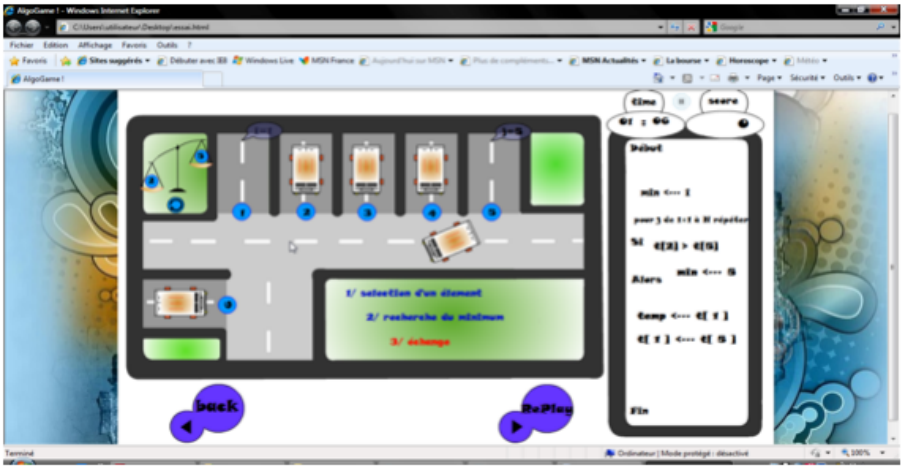

Figura 1.6. AlgoGame , Selection Sort level

AlgoGame [14] , si propone di insegnare i principi di programmazione e gli algoritmi fondamentali, partendo dai concetti di If-Then-Else e Loops, fino ad arrivare agli algoritmi di sorting. Nell'esempio illustrato in figura, viene analizzato l'algoritmo Selection Sort. L'obiettivo è ordinare dei furgoni nelle aree di parcheggio in base al loro peso. È disponibile un'area di sosta dedicata per effettuare lo swap dei furgoni. Ciascuna azione compiuta genera un codice sulla destra, e il giocatore deve completare il livello entro un limite di tempo.

- PlayLOGO 3D

  Si tratta di un Serious Game progettato per bambini delle scuole elementari, con l'obiettivo di insegnare i concetti chiave del linguaggio LOGO. Questo gioco [40] offre un'esperienza coinvolgente in cui i giocatori controllano remotamente dei robot. Il gameplay consiste nel redigere il codice necessario per far muovere il proprio robot, che si trova in una sfida diretta con il robot di un altro utente. Il gioco adotta una meccanica a turni, permettendo a ciascun giocatore di effettuare la propria mossa in sequenza. La vittoria viene assegnata al giocatore che per primo riesce a raggiungere il robot avversario e invia il comando di collisione.

Figura 1.7. PlayLOGO 3D

- OOP Codes

  Nell'articolo "OOP Codes: Teaching Object-Oriented Programming Concepts Through a Mobile Serious Game" [46] , viene descritto OOP Codes , un Serious Game per dispositivi mobile. OOP Codes è costituito dalla storia principale , che si articola in 7 livelli , ogni livello si focalizza su un argomento della programmazione ad oggetti. Il gioco principale si presenta come un platform 2D, consentendo interazioni con vari oggetti e l'attivazione di artefatti che avviano minigiochi.

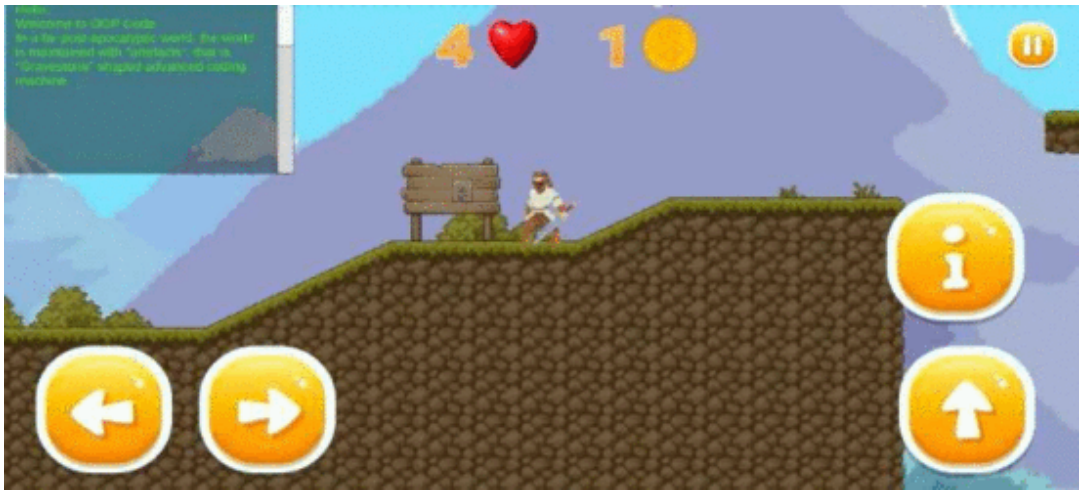

Figura 1.8. OOPCode , platform gameplay

I minigiochi costituiscono il nucleo della componente educativa di OOP Codes. In alcuni di essi, il giocatore deve riordinare le istruzioni, in altri deve prevedere l'output di un determinato segmento di codice, mentre in ulteriori sfide deve inserire pezzi di codice mancanti. Infine, sono presenti minigiochi strutturati come quiz a scelta multipla.

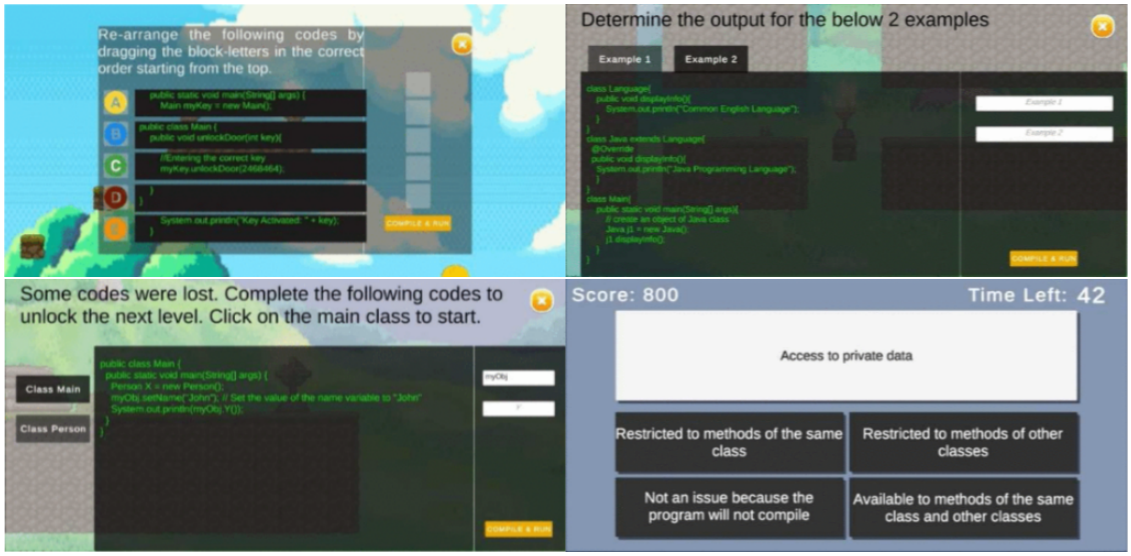

Figura 1.9. OOPCode , minigiochi

I risultati ottenuti vengono registrati su un database, al quale l'insegnante può accedere tramite un'applicazione dedicata per valutare l'andamento degli studenti.

- OOg

  L'articolo "Stealth assessment in serious games to improve OO learning outcomes" [3] , introduce il Serious Game OOg .

  Il gioco è progettato per insegnare la programmazione ad oggetti. Coinvolge l'analisi di storie presentate sotto forma di testo, da cui l'utente deve creare una corrispondenza in linguaggio ad oggetti, identificando classi, attributi, ecc.

Figura 1.10. OOg , identificazione di una classe

Figura 1.11. OOg , identificazione degli attributi

- Odyssey Of Phoenix

  L'articolo "A propriety game based learning mobile game to learn object-oriented programming — Odyssey of Phoenix" [57], introduce Odyssey

Of Phoenix, un gioco 2D, con meccanica punta e clicca , ideato per insegnare la programmazione ad oggetti senza tuttavia presentare esercizi di programmazione.

Figura 1.12. OdysseyOfPhoenix , gameplay

La protagonista Kyle deve raccogliere dei pezzi per riparare la sua astronave. Tramite un logbook ha accesso a degli indizi sui pezzi (quantità necessarie , posizione ,ecc ..), che danno l'idea di quello che sono i concetti di classe e oggetti nella programmazione. Il concetto di ereditarietà si manifesta nella creazione delle parti dell'astronave. Ad esempio, alcune parti come il motore principale e il motore secondario appartengono alla stessa classe base "motore". Questo consente la condivisione di risorse comuni, ma solo per le parti in comune, poiché il motore principale può richiedere elementi aggiuntivi rispetto a quello secondario, e così via. L'applicazione di questo meccanismo evidenzia il concetto di ereditarietà.

Figura 1.13. OdysseyOfPhoenix , logbook e inventario

- Dream Coders

  L'articolo "Experience with Dream Coders: developing a 2D RPG for teaching introductory programming concepts"[12] , introduce il Serious Game Dream Coders.

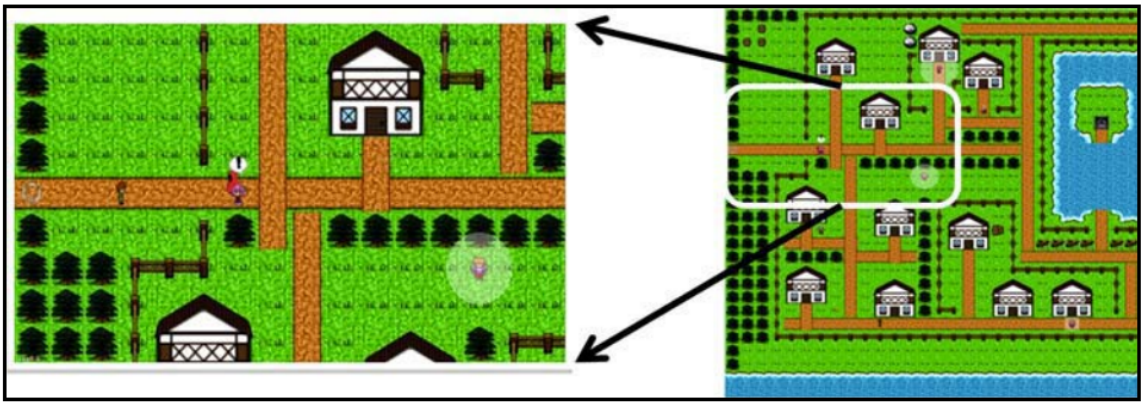

Figura 1.14. DreamCoders , ambientazione

Il gioco ha come obiettivo l'insegnamento di concetti di programmazione basilari, tra cui istruzioni condizionali, cicli, file I/O e array. Si tratta di un gioco RPG in 2D che consente ai giocatori di navigare ed interagire con l'ambiente. La storia principale segue un alunno che si addormenta

durante una lezione di programmazione, e il suo sogno costituisce l'ambientazione del gioco. L'obiettivo finale è risvegliare l'alunno dal sogno, incarnando l'eroe del gioco.

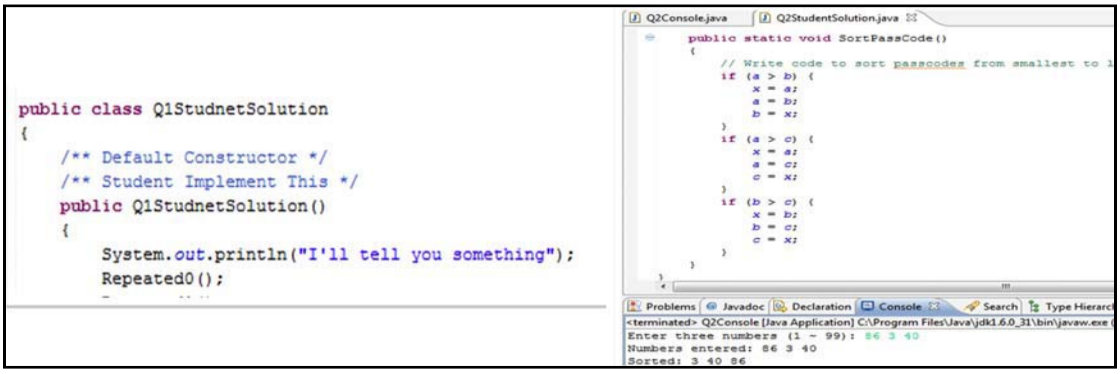

Figura 1.15. DreamCoders , IDE

Per completare le missioni, il giocatore fa uso di un IDE di programmazione classico, dove può scrivere il codice necessario per superare il livello. Tuttavia, il gioco non è in grado di riconoscere automaticamente la correttezza del codice scritto, quindi è necessario utilizzare la console dell'IDE per verificare l'esattezza delle istruzioni.

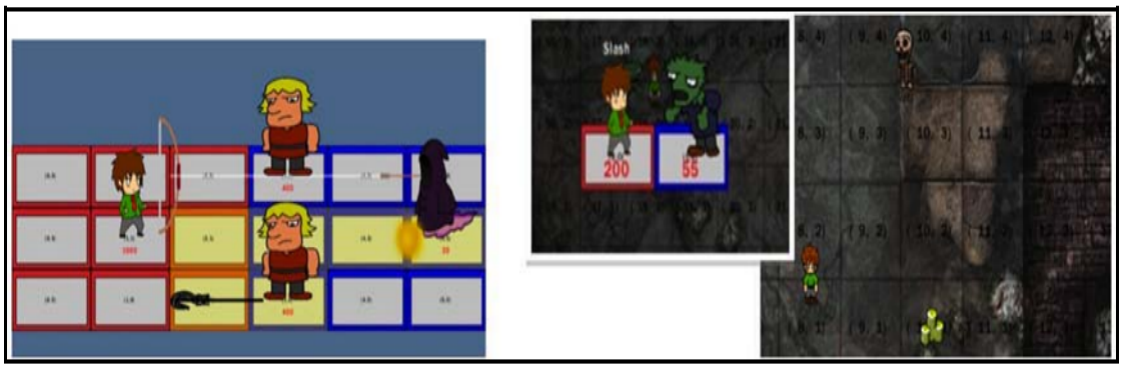

Figura 1.16. DreamCoders , Combat System

Il gioco include anche un semplice sistema di combattimento per aumentare il coinvolgimento dell'utente.

- Gaming With OOP Learn

  Si tratta di un gioco 3D platform per dispositivi mobile descritto nell'articolo "Gaming With OOP Learn: A Mobile Serious Game to Learn Object-Oriented Programming" [10] .

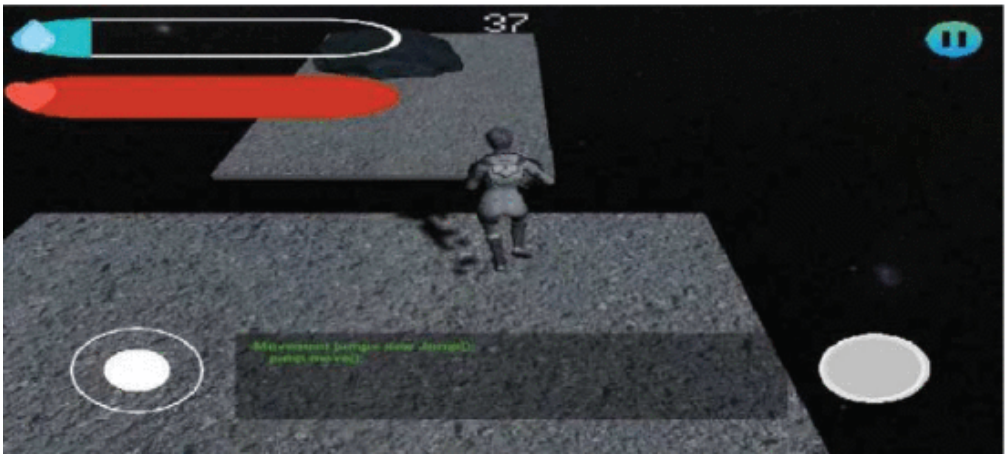

Figura 1.17. GamingWithOOPLearn , gameplay

Il giocatore controlla un personaggio impegnato nel saltare ostacoli e raccogliere risorse vitali come l'acqua per mantenersi in vita.

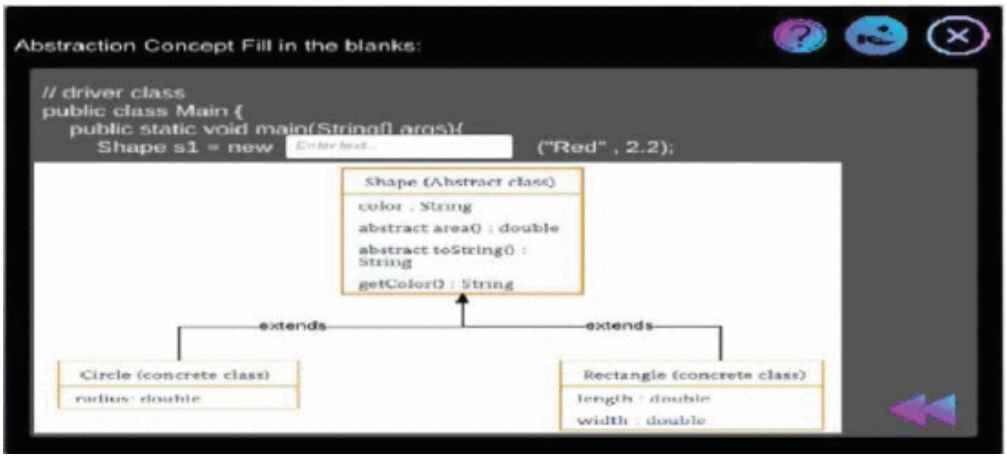

Figura 1.18. GamingWithOOPLearn , minigiochi

Durante l'avventura, il giocatore affronterà dei minigiochi incentrati sulla programmazione ad oggetti e potrà proseguire nel cammino solo completandoli correttamente. Ad esempio, potrebbe dover riempire lo spazio

vuoto di una classe Java con il codice corretto o affrontare minigiochi con meccanica drag & drop per selezionare le risposte corrette.

- POO

  Si tratta di un gioco in 2D con meccanica drag&drop per dispositivi mobile descritto nell'articolo "Teaching Object Oriented Programming Concepts Through a Mobile Serious Game" [33] , concepito principalmente per utenti alle prime armi nel mondo della programmazione.

Figura 1.19. POO , schermata iniziale

I protagonisti del gioco sono rappresentati da animali attraverso i quali vengono spiegati alcuni concetti fondamentali della programmazione ad oggetti.

Il gioco si articola in quattro livelli, ciascuno dei quali ha l'obiettivo di illustrare concetti di programmazione distinti.

Ad esempio creando un animale si spiega il concetto di oggetto. Si attribuisce all'animale una specie specifica (leone, rana, serpente, ecc.) per illustrare il concetto di classe. Successivamente, assegnando l'animale alla classe di appartenenza (mammifero, anfibio, ecc.), si introduce il concetto di ereditarietà.

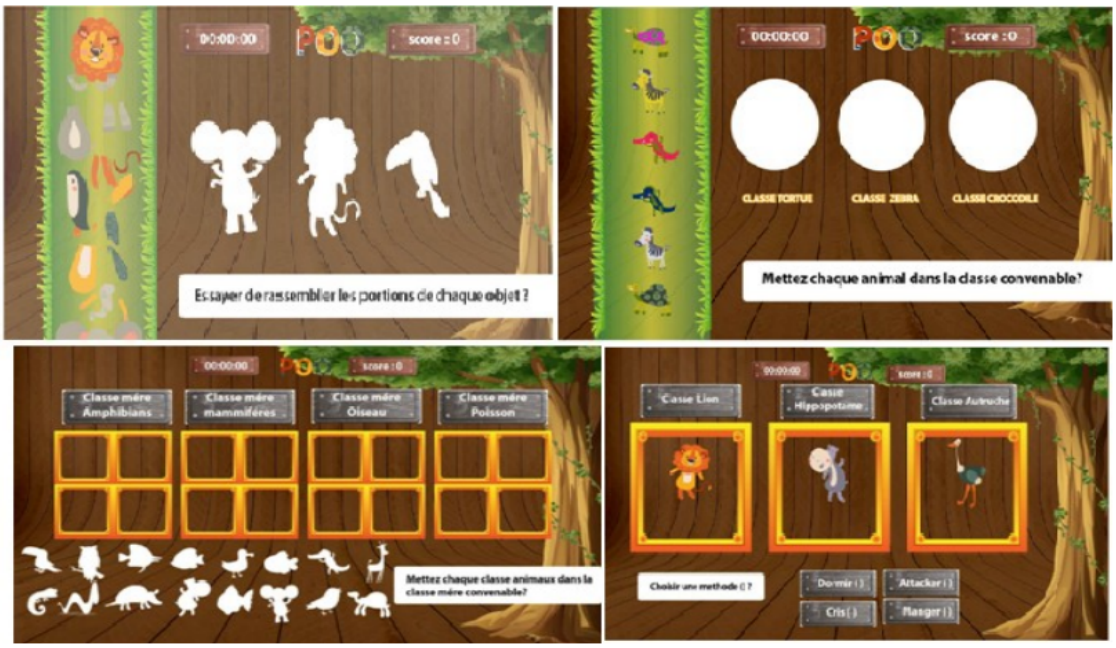

Figura 1.20. POO , livelli

- Serious Cube

  L'articolo "Serious games for learning programming languages" [35] , descrive due Serious Game (Serious Cube e 3D ACTION GAME) per imparare linguaggi di programmazione.

  L'obiettivo del gioco Serious Cube 1.21 è riempire gli spazi vuoti di un codice Java con le parole corrette. Per selezionare le parole, l'utente gira dei cubi utilizzando un controller Xbox, scegliendo il lato con la parola chiave appropriata per completare il codice.

- 3D ACTION GAME

  Gioco multiplayer 3D che propone una dinamica simile a Serious Cube, dove almeno due giocatori competono per riempire gli spazi vuoti di un codice Java con le parole corrette. Tuttavia, in questa versione, i giocatori devono raccogliere le parole nell'ambiente di gioco con i propri personaggi. Il vincitore è colui che riesce per primo a raccogliere tutte le parole necessarie. Per aumentare l'aspetto divertente e coinvolgente del gioco, i giocatori possono visualizzare il codice dell'avversario e utilizzare strategie per sottrargli le parole, rendendo la competizione più

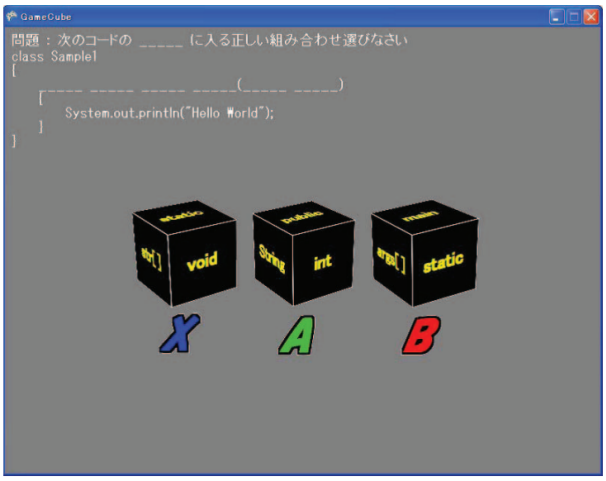

Figura 1.21. SeriousCube

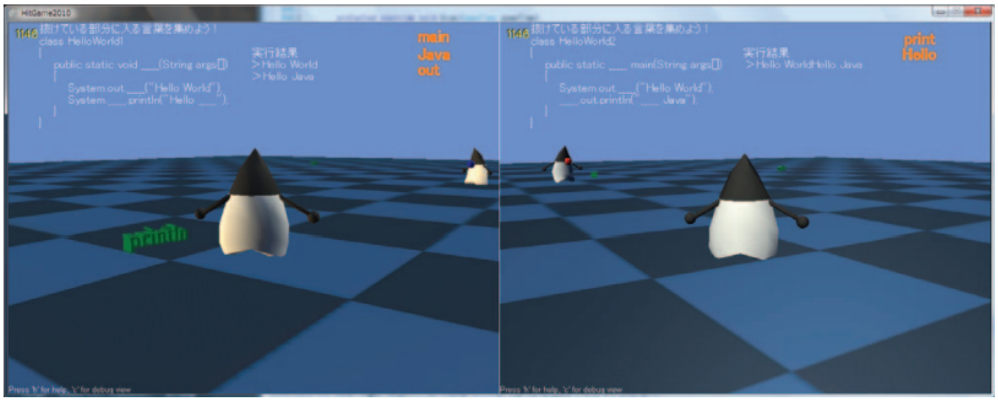

Figura 1.22. 3DActionGame

stimolante e, al contempo, offrendo un'opportunità di apprendimento attraverso l'analisi e l'interpretazione di codici altrui.

- Daily Life Programming

  L'articolo "An interactive serious game via visualization of real life scenarios to learn programming concepts" [2] , propone di insegnare concetti chiave della programmazione mediante minigiochi che utilizzano esempi tratti dalla vita quotidiana.

  Al fine di spiegare il concetto di loop, ad esempio, il giocatore dovrà far girare un ragazzo attorno a un tempio un numero specifico di volte,

Figura 1.23. Daily Life Programming , minigioco Loop

indicando solamente il numero di iterazioni del ciclo e il comparatore da utilizzare. I dati immessi influenzeranno direttamente l'effetto nel gioco, mostrando quante volte e in che modo il ragazzo girerà intorno al tempio.

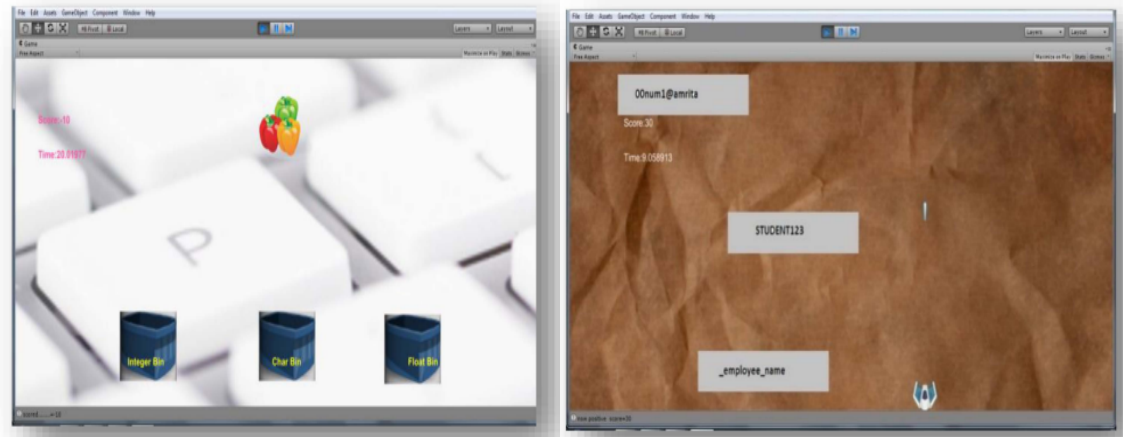

Figura 1.24. Daily Life Programming , minigiochi tipi di dato e nomenclatura variabili

In un altro minigioco, il concetto della spazzatura differenziata viene applicato per spiegare la differenza tra i diversi tipi di variabili. Invece di riciclare oggetti reali, il giocatore sarà coinvolto nell'associazione di

variabili ai contenitori appropriati, sottolineando così la corretta classificazione di variabili come int, float, o string. Un minigioco ulteriore assume la forma di uno shooter, progettato per illustrare come scrivere correttamente le variabili. L'obiettivo del giocatore sarà quello di sparare ai nomi delle variabili scritti in modo errato, promuovendo così la pratica di una corretta nomenclatura delle variabili attraverso un'esperienza di gioco coinvolgente.

- Ztech de Object-Oriented

  L'articolo "Computer Game as Learning and Teaching Tool for Object Oriented Programming in Higher Education Institution" [43] , propone Ztech de Object-Oriented , un Serious Game in grafica 2D .

Figura 1.25. Ztech , Combat System

Nella sezione di gioco, i giocatori assumono il controllo di un personaggio con l'incarico di sconfiggere i nemici, in questo caso, i terroristi. La progressione nel gioco porta al potenziamento del personaggio, rendendolo più forte e in grado di affrontare sfide più impegnative. L'obiettivo della componente di gioco è principalmente quello di stimolare l'interesse degli utenti e favorire l'apprendimento dei concetti fondamentali della programmazione orientata agli oggetti, come l'incapsulamento, l'ereditarietà e il polimorfismo. La parte educativa è introdotta attraverso l'impiego di minigiochi specifici, tra cui quiz, attività di riordinamento di segmenti di codice e altre meccaniche più innovative.

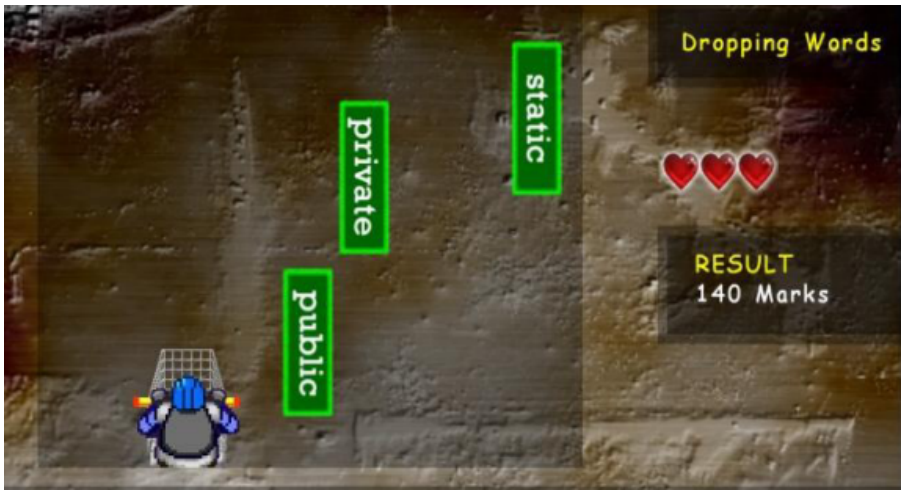

Figura 1.26. Ztech , minigioco Keyword

In uno dei minigiochi proposti, l'utente è impegnato a raccogliere le keyword del linguaggio di programmazione insegnato utilizzando un carrello virtuale. Durante questa attività è necessario selezionare accuratamente le keyword corrispondenti al linguaggio di programmazione in esame. Ogni errore commesso, come la mancata raccolta di una keyword corretta o la selezione di una non appropriata, comporta la perdita di una vita.

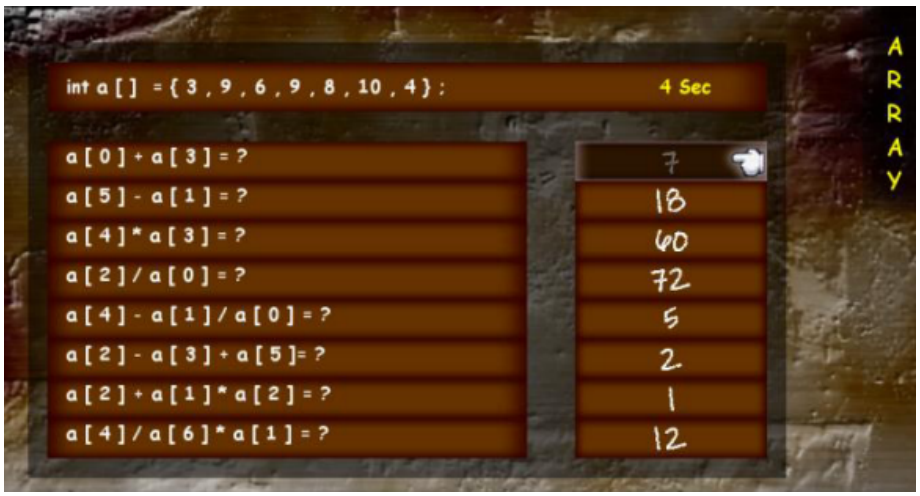

Figura 1.27. Ztech , minigioco array

Un altro minigioco consiste in un quiz a tempo, in cui l'utente è chiamato

a rispondere correttamente a domande che riguardano operazioni con array.

- Program your robot

  L'articolo "A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming" [31] , introduce il Serious Game Program your robot .

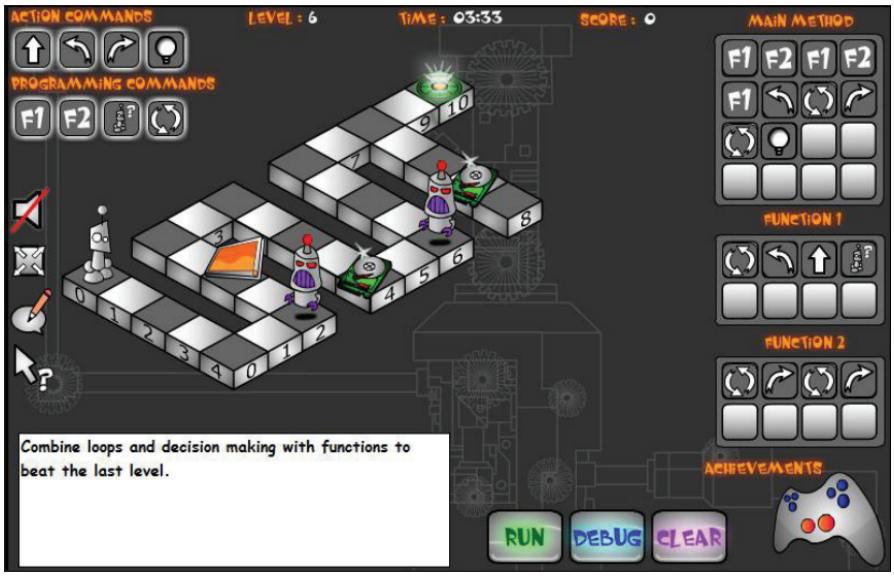

Figura 1.28. ProgramYourRobot

Il gioco propone agli utenti di assistere un robot nella fuga da una serie di piattaforme, costruendo un piano di fuga chiamato algoritmo risolutivo. Affrontando vari livelli, i giocatori devono sviluppare i loro algoritmi, impartendo comandi al robot per raggiungere un punto di destinazione denominato "teleporter" e superare il livello. Man mano che i giocatori avanzano, la complessità aumenta con l'espansione delle piattaforme. Durante il gioco, incontrano oggetti casuali che possono essere catturati dal robot, offrendo ricompense. L'obiettivo principale è motivare i giocatori a progettare algoritmi riutilizzabili, incoraggiando l'uso di funzioni, loop e istruzioni condizionali anziché inserire tutti i comandi nel metodo principale. Questo approccio mira a far comprendere ai giocatori l'importanza di separare la logica in modelli ripetibili, promuovendo la creazione di soluzioni più efficienti e riutilizzabili nel tempo.

- VR-OCKS

  Spostando l'attenzione sulla tecnologia VR , l'articolo "VR-OCKS: A virtual reality game for learning the basic concepts of programming" [42] , presenta un prototipo innovativo chiamato VR-OCKS. Un sistema di realtà virtuale (VR) progettato per insegnare i concetti di base della programmazione.

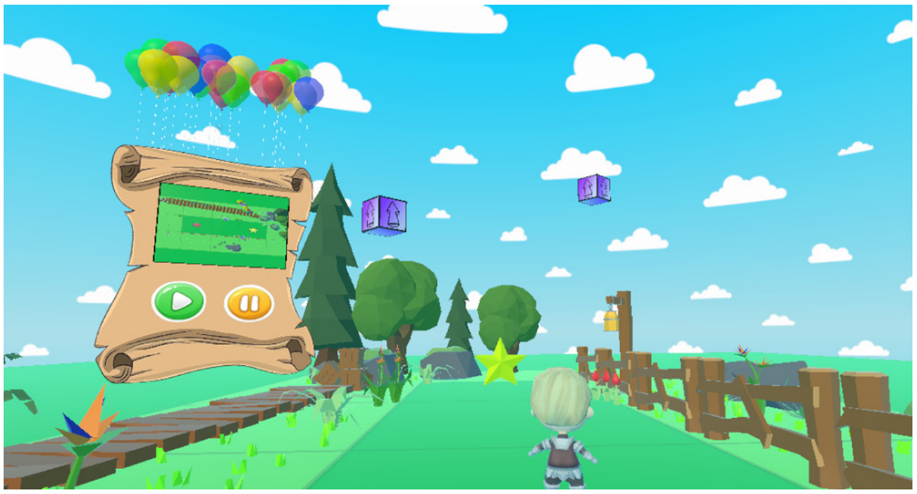

Figura 1.29. VR-OCKS

Ispirandosi a linguaggi visivi come Scratch o Kodu, VR-OCKS offre un'esperienza immersiva in cui gli utenti possono risolvere puzzle tridimensionali. Il sistema richiede l'uso di comandi di base, come avanzare e girare, insieme a concetti più avanzati, come iterazioni e selezioni condizionali, per affrontare sfide sempre più complesse. L'approccio mira a coinvolgere bambini e adolescenti nel mondo della programmazione, sfruttando il fascino e le potenzialità della realtà virtuale.

La meccanica di VR-OCKS è focalizzata sulla risoluzione di puzzle tridimensionali attraverso l'uso di comandi di base rappresentati da blocchi chiamati "blocchi azione". Il giocatore inizia il percorso con diversi ostacoli e deve selezionare e posizionare i blocchi azione in un ordine corretto per definire un algoritmo che consenta al personaggio di raggiungere l'obiettivo designato. Dopo aver definito l'algoritmo, il giocatore può testarlo premendo un pulsante "Esegui" per far muovere il personaggio; se raggiunge la posizione di obiettivo in modo corretto, il livello è completato

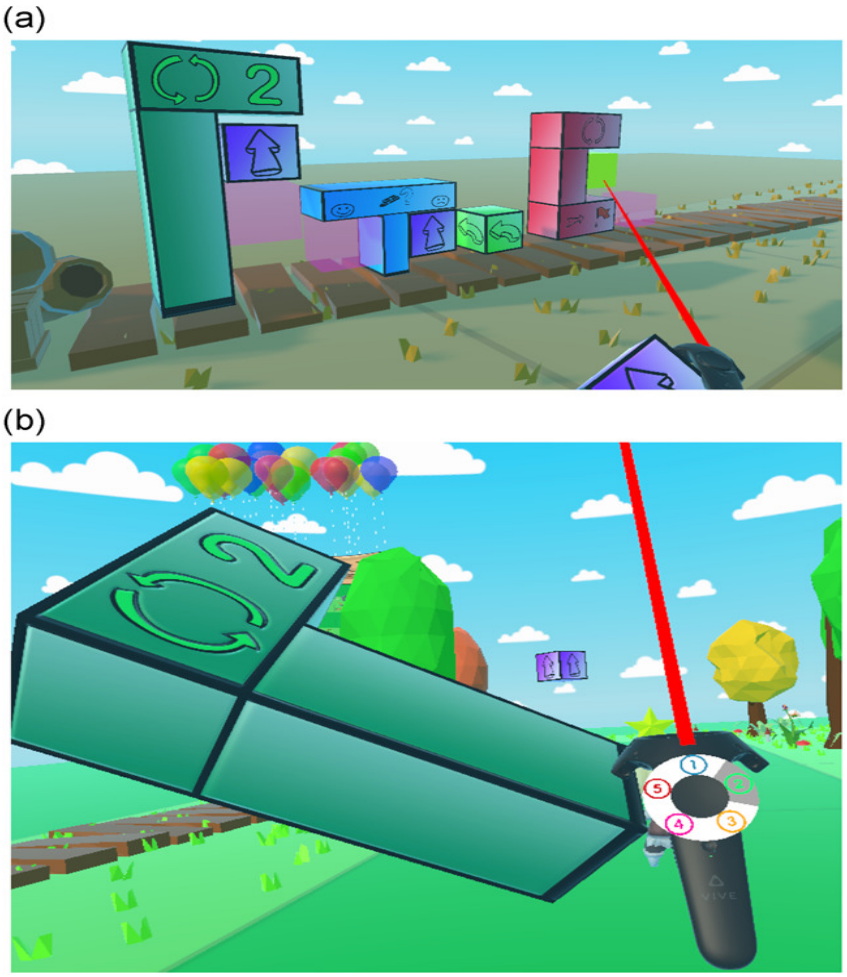

Figura 1.30. VR-OCKS , puzzle

- 3D Virtual Programming Language

  Nell'articolo "Towards a 3D Virtual Programming Language to Increase the Number of Women in Computer Science Education" [39] , si propone un linguaggio di programmazione virtuale tridimensionale .

  L'approccio innovativo si basa sull'utilizzo della tecnologia immersiva VR per insegnare i concetti di programmazione attraverso la programmazione basata su blocchi. Nell'immagine fornita, si può osservare la rappresentazione grafica dei concetti di classe e oggetti. Sul lato sinistro, è presente un codice Python, mentre a destra si visualizza la classe in azzurro, gli oggetti in blu, i metodi in verde e gli attributi in

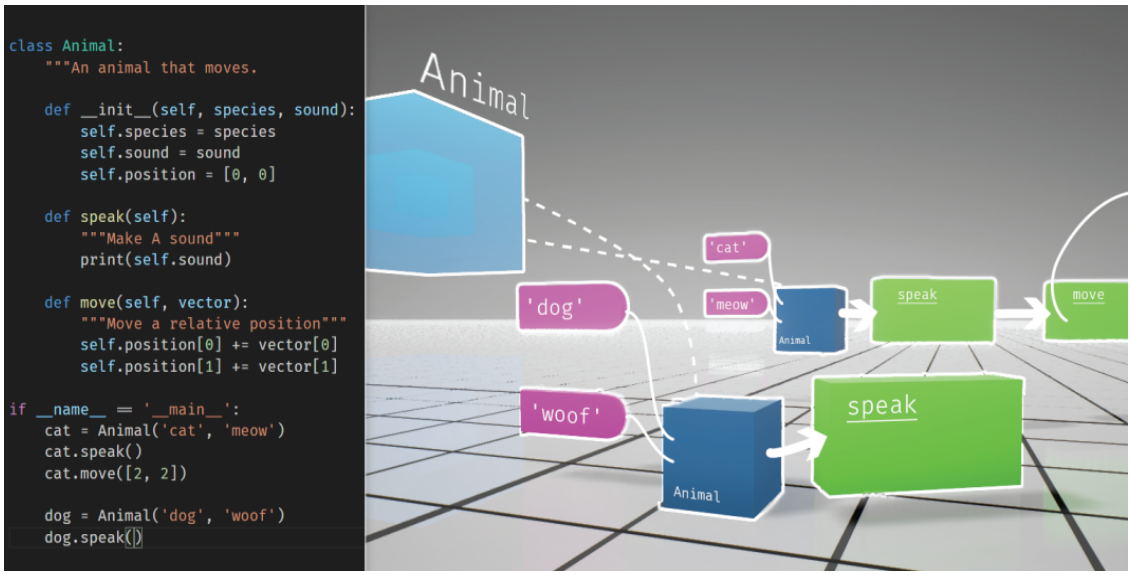

Figura 1.31. 3D-VPL

rosa. Questa rappresentazione visiva offre agli utenti un modo coinvolgente e intuitivo per comprendere i fondamenti della programmazione, sfruttando la potenza e l'immersività della realtà virtuale.

University of Macedonia

In questa sezione, vengono presentati alcuni progetti di tesi sviluppati presso l'Università Macedone, focalizzati sui Serious Game nel campo della programmazione. È possibile esplorare e scaricare i giochi illustrati attraverso il seguente collegamento : [49]

- An extension of objectKarel

  Ambiente tridimensionale in cui è necessario programmare il percorso di un robot per raggiungere il cancello del terminale e avanzare al livello successivo. I livelli del gioco sono predefiniti, ognuno con un obiettivo specifico. Ad ogni livello, vengono introdotte nuove difficoltà, come nuovi blocchi da spostare nelle posizioni corrette per attivare meccanismi, aprire porte, eccetera. Tutte queste azioni sono eseguite programmando in Java attraverso un apposito editor testuale. I concetti di

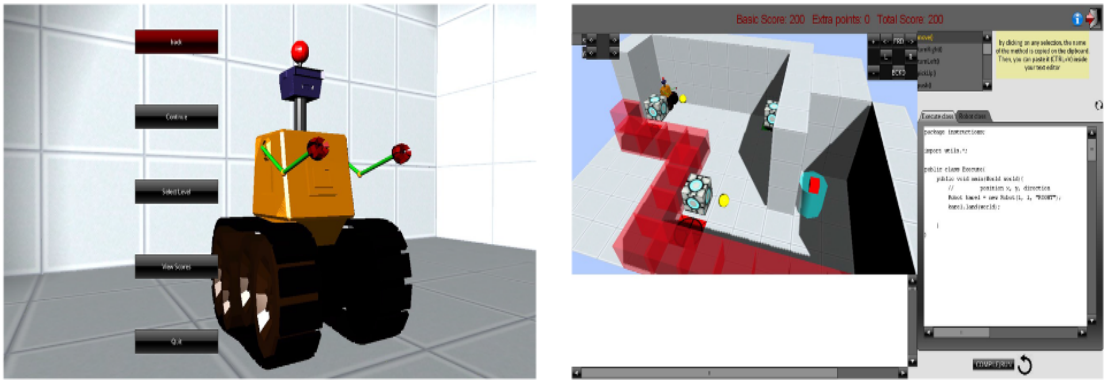

Figura 1.32. ObjectKarel

Object-Oriented Programming (OOP) trattati dal gioco includono classi e oggetti, ereditarietà, polimorfismo e overriding.

- JavabotWars

  Sparatutto in prima persona (First Person Shooter) ambientato in un mondo in guerra tra robot ed esseri umani. Il protagonista del gioco è un soldato trasformato in cyborg a seguito di un incidente, diventando l'unica speranza per vincere la guerra. Poiché è un cyborg, i suoi circuiti interni sono programmati in Java e deve imparare il linguaggio nel corso del gioco per poter sopravvivere.

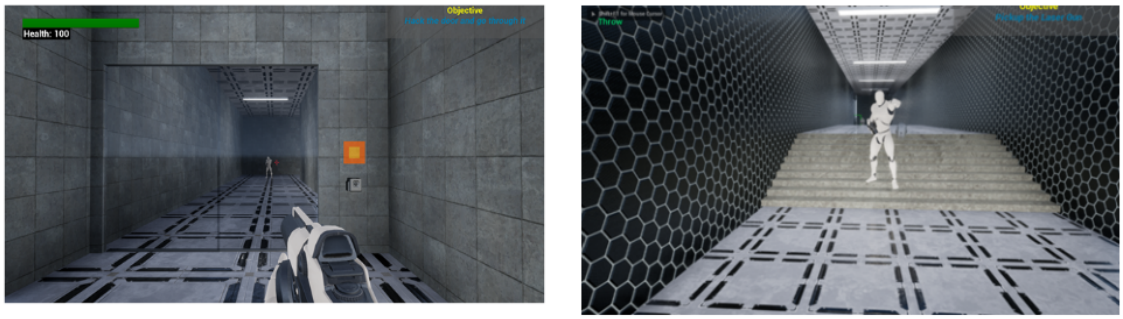

Figura 1.33. JavabotWars , gameplay

La componente educativa è introdotta attraverso minigiochi, le cui meccaniche principali consistono nel riempire gli spazi vuoti di un codice

Java con le parole corrette, determinare l'output di un codice e ordinare frammenti di codice. È importante notare che un errore in questi minigiochi provoca la morte del personaggio.

Figura 1.34. JavabotWars , minigiochi

- Py-rate Adventures

  Il gioco e la sua implementazione sono descritti anche nell'articolo "PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming With Python" [45]. Py-rate Adventures è un gioco progettato per apprendere i concetti di base della programmazione in Python. Il protagonista, un pirata, deve evitare i nemici, rappresentati da teste di scheletri, saltare sulle piattaforme evitando di cadere, in uno stile classico di gioco 2D platform. Lungo il percorso, sono presenti dei forzieri che forniscono nozioni teoriche sui concetti di programmazione.

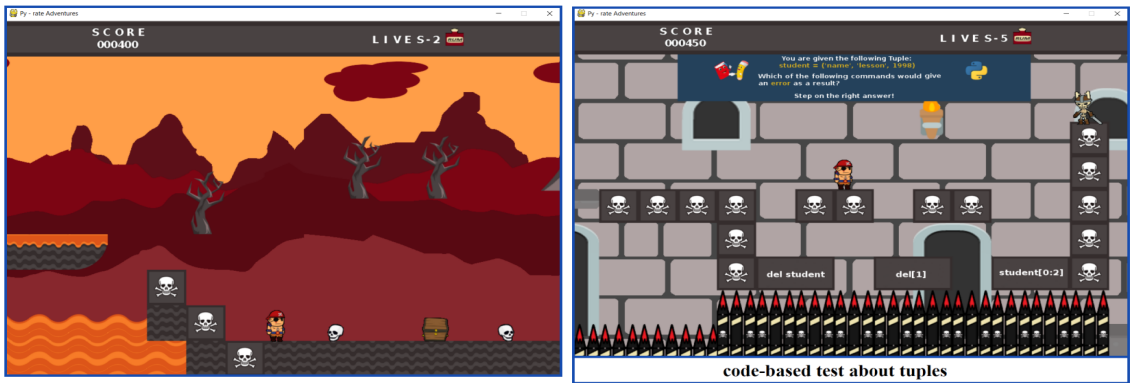

Figura 1.35. Py-rate Adventures , gameplay

Le conoscenze acquisite attraverso i forzieri vengono testate mediante minigiochi. Un esempio di questi è illustrato nell'immagine a destra, dove l'utente deve selezionare la risposta corretta alla domanda in evidenza. Tuttavia, anziché effettuare una selezione tramite un semplice clic del mouse, è necessario spostare il personaggio sulla piattaforma corretta. Una variante di questo approccio coinvolge l'associazione di ogni risposta a una bottiglia di rum, richiedendo al giocatore di acquisire la bottiglia giusta.

- Rise of the Java Emperor

  Il gioco è progettato per introdurre i concetti di base della programmazione in Java agli studenti più giovani, senza prerequisiti specifici, pertanto è particolarmente adatto a giocatori senza conoscenze pregresse di programmazione Java o a coloro che desiderano mettere alla prova le proprie competenze del linguaggio. Il gioco e la sua implentazione sono anche descritti nell'articolo "Investigating the Perceived Player Experience and Short-term Learning of the Text-based Java Programming Serious Game "Rise of the Java Emperor"" [55].

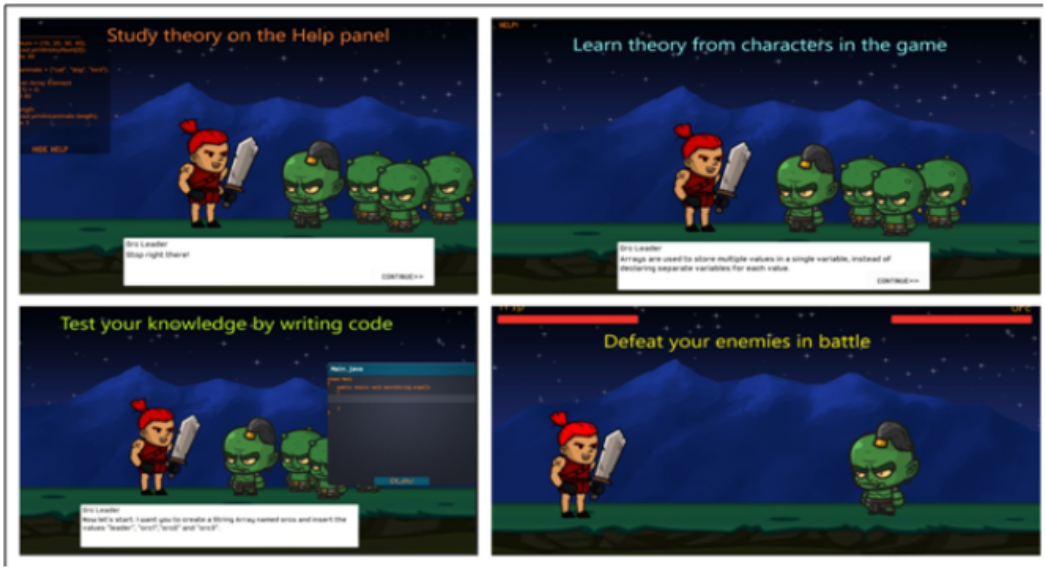

Figura 1.36. Rise of the Java Emperor

Il gioco si presenta come un'avventura grafica, priva di un vero e proprio gameplay. La trama narra la storia di un cavaliere attraverso dialoghi con personaggi che spiegano i principi di Java. Alla fine dei dialoghi, compare

una schermata con del codice predefinito e delle caselle vuote; l'obiettivo è riempire le caselle vuote con il codice richiesto per completare il livello. Ad esempio, in una situazione in cui ci si trova di fronte a un lago e bisogna attraversarlo creando una nave, viene fornita la classe base e bisogna implementare le parti mancanti per costruire la nave.

- Java Offsprings

Figura 1.37. Java Offsprings

Java Offsprings è un escape game online con l'obiettivo di educare i partecipanti sulle basi del linguaggio di programmazione Java. Il gioco comprende dieci livelli, ciascuno dei quali copre una parte diversa della teoria Java attraverso vari compiti. Nel corso del gioco, i giocatori devono cercare indizi per procedere, trovando oggetti che, se utilizzati nei posti e nei modi corretti, conducono alla scoperta della chiave per il livello successivo. Ogni chiave contiene la teoria Java relativa al livello corrente, e i giocatori possono visualizzarle in modo asincrono grazie a un inventario che tiene traccia degli oggetti e delle chiavi raccolte. Durante ogni livello, per ottenere determinati oggetti, i giocatori devono completare domande legate a Java. Inoltre, attraverso un editor di testo, devono scrivere il codice richiesto per avanzare nella sfida.

Figura 1.38. Java Offsprings, indizi

- jAVANT-GARDE

  L'implementazione del gioco è descritta nell'articolo"jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming With Java" [26]. Il gioco è un platform 2D con protagonista un robot. Le meccaniche principali seguono gli standard dei platform, avanzando attraverso il livello saltando sulle piattaforme, e perdendo una vita se si cade nel vuoto. Inoltre, sono inclusi minigiochi educativi, come quiz o domande, in cui una risposta errata può comportare la perdita di una vita. Sono inoltre presenti postazioni nozionistiche ,con visualizzazione delle nozioni opzionale, che consentono ai giocatori di apprendere nozioni di Java/OOP specifiche per il livello.

  Oltre ai quiz, sono presenti vari tipi di minigiochi, come riempire gli spazi vuoti di un codice Java con le parole mancanti. Alcuni minigiochi sono integrati nel gameplay principale, come ad esempio quello che spiega la differenza tra i vari tipi di variabili (int/double, ecc.). In questo caso, i giocatori devono scegliere la piattaforma appropriata su cui saltare per rappresentare correttamente il tipo di variabile. Se si sceglie la piattaforma sbagliata, si perde una vita. Inoltre, ci sono anche quiz integrati nel gameplay principale, dando la possibilità di rispondere alle domande attivando leve apposite con il personaggio anziché con un classico click del mouse.

  La fine di ogni livello consiste nella scrittura effettiva di codice. Si aprirà un editor in cui è possibile scrivere il codice richiesto e compilarlo. Scrivendo del codice errato, si perderà una vita. Una volta scritto il codice corretto, si può entrare in un portale e passare al livello successivo.

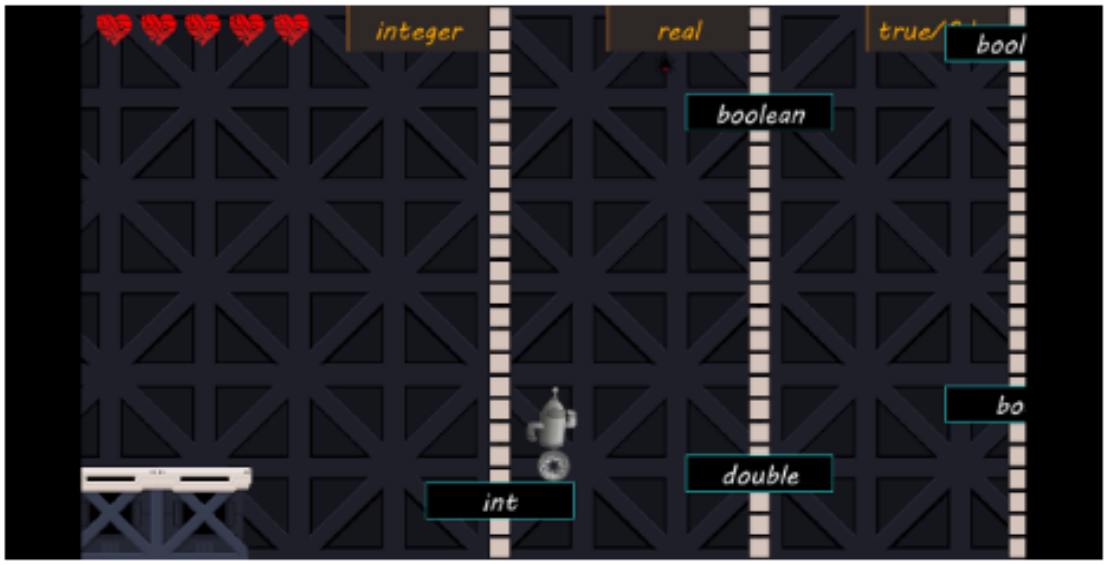

Figura 1.39. jAVANT-GARDE , salto sulle piattaforme corrette in base al tipo di variabile

Figura 1.40. jAVANT-GARDE , editor di fine livello

1.3 Meccaniche principali minigiochi didattici per programming

Grazie a un'indagine condotta sugli studenti, focalizzata sullo sviluppo di serious game per apprendere Java, discussa nell'articolo "A mobile-device based serious gaming approach for teaching and learning Java programming" [30], emerge che il 50% degli studenti intervistati preferisce giochi con un forte storytelling, mentre l'altro 50% predilige minigiochi. Di conseguenza, si è deciso di concentrare l'attenzione nella ricerca di meccaniche di minigiochi innovative e utili per l'apprendimento della programmazione a oggetti. Risulta evidente che la maggior parte dei minigiochi adotta meccaniche classiche per la maggior parte degli argomenti educativi, tra cui:

- Drag&Drop
- Riempire gli spazi vuoti di un codice
- Quiz a scelta multipla
- Determinare l'output di un segmento di codice
- Ordinare dei pezzi di codice

La ricerca ha portato all'individuazione di approcci innovativi per ciascun argomento educativo. È importante sottolineare che non si è riuscito a definire una meccanica per ogni singolo argomento e che si è considerata una distinzione tra le meccaniche adatte a un target basso, rappresentato dagli studenti che si avvicinano per la prima volta al mondo della programmazione senza conoscenze pregresse, e un target medio, rappresentato da coloro che hanno già familiarità con le basi della programmazione. Di seguito è riportata la tabella con le meccaniche innovative per gli argomenti educativi:

Tabella 1.1: Meccaniche di gioco per argomento educativo

Argomento	Target Basso	Target Medio
Tipi di dato	• La meccanica di gioco proposta coinvolge l'utente nella selezione e nell'inserimento di ogni elemento in contenitori corrispondenti alle tipologie di dati, come Int, Float e String. Ad esempio, se l'elemento è il numero 2.43, l'utente deve depositarlo nel contenitore Float. Durante l'attività, il sistema monitora il tempo impiegato dall'utente per completare l'operazione. In caso di inserimento in un contenitore errato, il punteggio subisce una riduzione. [2] • La meccanica consiste nel raccogliere le keyword del linguaggio di programmazione insegnato utilizzando un carrello. Durante questa fase, è importante raccogliere le keyword corrette, e si perde una vita se l'utente non raccoglie alcuna keyword o ne raccoglie una non corretta. Ad esempio, se stiamo insegnando il linguaggio Java, le keyword potrebbero includere "int", "double", "if", "else", ecc. [43]	• La meccanica di gioco proposta coinvolge una spiegazione interattiva della differenza tra i vari tipi di dati (come int, double, ecc.). L'utente deve scegliere tra le piattaforme scorrenti quella appropriata su cui saltare. In caso di scelta errata, si perde una vita. Per esempio, se si sta insegnando come descrivere variabili per rappresentare un numero reale, tre piattaforme diverse scorrono: boolean, double e String. L'utente deve saltare sulla piattaforma corrispondente al tipo di dato double per evitare di cadere nel vuoto e perdere una vita. [26]

Continuation of Table 1.1		
Argomento	Target Basso	Target Medio
Variabili	• La dinamica del gioco è volta a verificare se l'utente ha appreso le regole per denominare le variabili. In questo contesto, l'utente controlla una navicella spaziale e deve sparare ai nomi delle variabili che sono errati o non conformi alle regole stabilite. L'obiettivo è eliminare o evitare tali nomi scorretti per accumulare punti e progredire nel gioco. [2]	• Il Target dovrebbe già aver sviluppato tali competenze
Array	• Un quiz a tempo dove è necessario rispondere correttamente a domande riguardanti operazioni con array. [43]	• Il Target dovrebbe già aver sviluppato tali competenze
Costrutti condizionali	• In un contesto di storytelling, ad esempio quando il giocatore deve acquistare una macchina in concessionaria, gli viene assegnato il compito di prendere decisioni riguardo alle condizioni dell'acquisto. Il giocatore deve selezionare quali valori saranno confrontati nella condizione e quali operatori verranno utilizzati. Ad esempio, considerando il costo della macchina come X, il giocatore dovrà decidere se possiede abbastanza soldi per effettuare l'acquisto. [2]	• Il Target dovrebbe già aver sviluppato tali competenze

Continuation of Table 1.1		
Argomento	Target Basso	Target Medio
Loop	• Nel contesto del gioco, l'utente è incaricato di far girare un personaggio attorno a un tempio per un numero specifico di iterazioni. L'interazione avviene attraverso la scelta del numero di iterazioni del ciclo e del comparatore da utilizzare, il tutto attraverso un menu a tendina. In base ai dati immessi dall'utente, verrà visualizzato l'effetto sul gioco, determinando quante volte e in che modo il personaggio girerà attorno al tempio. [2]	• Meccanica di programmazione a blocchi grafica , utilizzata per controllare il movimento di un personaggio. Gli utenti possono trascinare e collegare blocchi while con condizione di uscita o blocchi di cicli con un numero specifico di iterazioni. Questa meccanica è particolarmente utile per interagire con loop, costrutti condizionali e funzioni nel processo di programmazione del movimento del personaggio nel contesto del gioco. [42]
Funzioni	• L'obiettivo è far raggiungere ad un robot la casella di fine livello attraverso l'inserimento di istruzioni di movimento, come "avanti", "gira a destra". Le istruzioni vengono organizzate all'interno di funzioni, partendo dal main. Tuttavia, dato lo spazio limitato nel main, è necessario utilizzare altre funzioni per implementare ulteriori funzionalità.[31]	• Non sono state trovate meccaniche significative per tale target.

Continuation of Table 1.1		
Argomento	Target Basso	Target Medio
Classi	• Per acquisire una comprensione della nozione di classe, si classificano gli animali in gruppi .Questa pratica di organizzare gli animali in gruppi basati su caratteristiche comuni rappresenta il concetto di classe,raggruppando insieme animali con simili attributi. [33] • In Odyssey Of Phoenix ,tramite un logbook si ha accesso a degli indizi sui pezzi da recuperare (quantità necessarie , posizione ,ecc ..), che danno l'idea di quello che sono i concetti di classe e oggetti nella programmazione.[57]	• Analizzando storie in formato testuale, è possibile creare una corrispondenza mediante l'uso del linguaggio ad oggetti. Identificando classi, attributi, metodi, e così via, si può tradurre il contenuto della storia in un contesto di programmazione orientata agli oggetti. [3]
Oggetti	• Comprendere la nozione di oggetto attraverso la costruzione di animali, mediante per esempio una meccanica in stile puzzle , in cui ogni animale è considerato come un oggetto. [33]	• Una meccanica tower defense in cui si scrive il codice per vari GameObject , ad esempio, la creazione di una torre e il suo posizionamento sul campo. Questo coinvolge la definizione della classe Torre, la creazione di un'istanza della torre e l'utilizzo di metodi con attributi specifici attraverso codice Java. [30] • Il concetto di classe e oggetto può essere rappresentato graficamente, come rappresentato in 3D-VPL, dove a sinistra si ha un codice Python e a destra la visualizzazione grafica di classi ,oggetti , metodi e attributi. [39]

Continuation of Table 1.1		
Argomento	Target Basso	Target Medio
Ereditarietà	• Comprendere la nozione di ereditarietà, trascinando ed assegnando gli animali in base alla rispettiva parent class. Ad es.l'oggetto di classe "CANE" viene assegnato alla specifica parent class "MAMMIFERO". [33] • In Odyssey Of Phoenix, il concetto di ereditarietà è espresso, ad esempio, nella creazione delle parti dell'astronave. Alcune parti, come il motore principale e il motore secondario, appartengono alla stessa classe base "motore". Ciò consente di condividere le stesse risorse per la costruzione, limitatamente alle parti comuni. È importante notare che il motore principale può richiedere elementi aggiuntivi rispetto a quello secondario e così via. L'applicazione di questo meccanismo evidenzia chiaramente il concetto di ereditarietà, dove una classe base può essere estesa per creare sottoclassi con caratteristiche specifiche.[57]	• Non sono state trovate meccaniche significative per tale target.
classi astratte e interfacce	• Competenze troppo elevate per questo Target	• Non sono state trovate meccaniche significative per tale target.
Interfacce funzionali	• Competenze troppo elevate per questo Target	• Non sono state trovate meccaniche significative per tale target.
Eccezioni	• Competenze troppo elevate per questo Target	• Non sono state trovate meccaniche significative per tale target.

Capitolo 2

Design

2.1 Selezione degli elementi teorici da trattare

Argomenti Educativi

L'obiettivo principale di questo progetto è introdurre i meccanismi fondamentali della programmazione a oggetti a studenti che si avvicinano per la prima volta al mondo della programmazione. Al fine di semplificare il percorso di apprendimento, si è deciso di escludere argomenti considerati troppo complessi, come polimorfismo, interfacce, ereditarietà, e così via. Gli argomenti di base della programmazione, come loop, costrutti condizionali, e strutture dati, sono anch'essi esclusi , poiché l'obiettivo non è insegnare a programmare in modo esaustivo, ma piuttosto fornire un'idea concreta del paradigma della programmazione a oggetti. In questo contesto, l'approccio didattico si concentra sulla presentazione dei concetti astratti del paradigma a oggetti , evitando di fornire implementazioni dettagliate di metodi o di altri costrutti. I metodi per esempio vengono introdotti come blocchi di calcolo astratto , che dati gli attributi a cui fanno riferimento ed eventuali valori di input , producono un output secondo determinate regole specifiche. Più nel dettaglio , considerando la classe "Chiave" con l'attributo "NomePorta" e il metodo "ApriPorta()", l'attenzione non è posta sull'implementazione specifica del metodo, ma piuttosto viene presentato come un blocco di calcolo astratto, che quando utilizzato, legge il nome della porta e se corrisponde alla porta che si vuole aprire , dopo alcuni calcoli il metodo riuscirà nel suo intento di apertura della suddetta porta .

Di seguito gli argomenti principali trattati:

- Classi
- Attributi
- Metodi
- Oggetti
- Regole di visibilità

Obiettivi

Lo scopo principale del gioco è quello di insegnare in maniera innovativa gli argomenti della programmazzione a oggetti presentati. Data la giovane età del pubblico di destinazione e la loro limitata esperienza con la programmazione, l'obiettivo non è spiegare ogni dettaglio sui concetti descritti, ma piuttosto fornire un'idea reale e concreta di cosa siano e come funzionino. Ci poniamo dunque vari obbiettivi da raggiungere con l'implentazione del gioco :

- Fornire una definizione chiara di classe e illustrarne il processo di costruzione
- Definire il concetto di oggetto
- Definire i metodi come blocchi di calcolo astratto, che dato un eventuale input, producono un output in base all'implentazione interna e agli attributi a cui fanno riferimento.
- Definire la relazione tra metodi e attributi
- Definire il concetto di visibilità degli attributi (private /public) . Più nel dettaglio, definire che un attributo di una classe deve essere dichiarato come public per essere richiamato da un metodo di un'altra classe.

2.2 Design del videogioco

Trama di gioco

Il mondo è sottomesso a un'entità di intelligenza artificiale, la quale ha assunto il controllo sugli esseri umani imprigionandoli in strutture penitenziarie

ad alta tecnologia. Questa intelligenza artificiale è stata completamente sviluppata attraverso un linguaggio di programmazione a oggetti. Tuttavia, al fine di mantenere segreti i suoi punti deboli, ha cancellato ogni traccia della documentazione del linguaggio da ogni angolo del mondo. L'unica possibilità di fuggire da questo carcere tecnologico è acquisire conoscenza sulla programmazione a oggetti, individuando indizi e risolvendo enigmi che permettano di aprire la strada all'interno del penitenziario fino ad arrivare al nucleo dell'intelligenza artificiale. Gli indizi e la documentazione necessari sono stati precedentemente lasciati da altri prigionieri che hanno tentato invano l'evasione , o dai robot di manutenzione che decidono spontaneamente di aiutare il protagonista . Una volta giunti al cuore dell'intelligenza artificiale, è essenziale applicare tutti i concetti appresi per distruggere definitivamente l'IA. Ogni stanza affrontata durante questa avventura fornisce conoscenze specifiche su vari argomenti della programmazione a oggetti, contribuendo così a formare una comprensione completa necessaria per abbattere l'intelligenza artificiale che tiene il mondo prigioniero.

Meccanica Escape Room

La meccanica principale si basa sull'esperienza tipica delle escape room. Il giocatore ha il controllo in prima persona del protagonista di gioco , il quale compito è fuggire dalla prigione risolvendo i vari enigmi di ogni stanza , avvalendosi degli indizi a sua disposizione. Questi indizi possono essere di varia natura: teoria di programmazione , attributi, metodi, classi, oggetti ; che a loro volta possono essere utilizzati in altri rompicapo. Una volta ottenuto un indizio , è possibile visualizzarlo tramite l'inventario. Il gioco presenta i classici oggetti di interazione delle escape room , come per esempio le cassaforti, quello che cambia è il modo in cui si interigisce con questi ultimi. La cassaforte non viene solamente vista come un classico oggetto in cui inserire una combinazione di numeri tramite una manopola, bensì viene introdotta come un oggetto di programmazione , che presenta un attributo codice e i metodi SetCodice() e ApriCassaforte() . Il giocatore potrà dunque settare il codice corretto tramite il metodo SetCodice() e successivamente applicare il secondo metodo per aprirla. Questo approccio aggiunge uno strato di coinvolgimento e connessione tra gli oggetti e i concetti di programmazione, offrendo un'esperienza più interattiva e didattica all'interno del contesto dell'escape room virtuale.

Teoria di programmazione

L'illustrazione dei concetti teorici della programmazione, come la definizione di classi e metodi, è veicolata attraverso indizi teorici specifici rinvenuti nel gioco , a seguito del corretto completamento di alcuni puzzle , o ottenuti come ricompensa alla fine di un dialogo con un altro personaggio. Questi indizi vengono successivamente aggiunti all'inventario del giocatore, rendendoli consultabili in qualsiasi momento. La lettura attenta di questa parte teorica agevola l'utente nel comprendere e risolvere i vari minigiochi e puzzle presenti nel contesto dell'escape room virtuale.

Classi

La creazione delle classi è un processo coinvolgente e interattivo . Gli attributi e i metodi, raccolti come indizi all'interno dell'escape room, sono componenti chiave di questa costruzione. Un minigioco 2D offre l'opportunità di assemblare la classe, collegando tra loro metodi e attributi richiesti. Tuttavia, è necessario fare attenzione a evitare che i collegamenti vengano interrotti da ostacoli,rappresentati da blocchi elettrici, o visti dai sensori dell'IA. Il giocatore ha a disposizione un tempo limitato per completare la costruzione della classe.

Una volta che la classe è stata assemblata con successo, il giocatore riceve il relativo indizio, aggiunto automaticamente all'inventario, contenente la composizione della classe. Usufruendo della composizione della classe creata , tramite una stampante 3D di oggetti , è possibile generare gli oggetti associati a quella classe. Tuttavia, è importante notare che l'utilizzo degli attributi e dei metodi utilizzati nella creazione della classe implica la perdita di questi elementi come oggetti singoli. Per recuperarli, il giocatore deve distruggere la classe, il che porta al reinserimento automatico degli indizi (Metodi e Attributi) nell'inventario. Questa meccanica aggiunge una dimensione strategica al gioco, costringendo i giocatori a bilanciare attentamente l'uso dei loro attributi e metodi per avanzare nel gioco.

Oggetti

Gli oggetti di programmazione sono lo strumento tramite cui il giocatore può interagire con il mondo , richiamando i metodi di cui dispongono. Gli oggetti possono essere trovati all'interno della scena ed inseriti nell'inventario per usufruirne in un secondo momento ; possono essere anche oggetti fissi come la cassaforte , che permette l'interazione di apertura richiamando i metodi

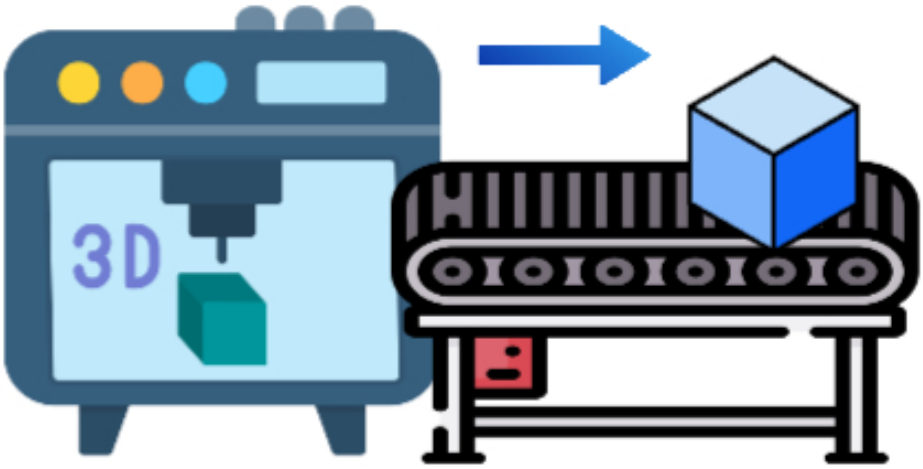

Figura 2.1. Stampante 3D

richiesti , senza la possibilità di essere raccolta e inserita nell'inventario ; oppure possono essere generati direttamente dal giocatore tramite una stampante 3D di oggetti , solo nel caso in cui si disponga della relativa classe. L'esperienza di creazione degli oggetti tramite la stampante 3D conferisce una dimensione tangibile e coinvolgente alla generazione di istanze di classe. Una volta ottenuta una classe , indistintamente se costruita direttamente dal giocatore o raccolta come indizio di scena , il giocatore ha l'opportunità di creare un oggetto corrispondente attraverso un processo interattivo. Accedendo alla stampante , il giocatore può visualizzare le classi presenti nel suo inventario e scegliere quella di cui si vuole creare un oggetto . Selezionata la classe , il giocatore ha accesso ad un minigioco , volto ad inizializzare la nuova istanza dell'oggetto con valori predefiniti per i suoi attributi . Una volta superato il minigioco, la stampante 3D produce l'oggetto desiderato. Il giocatore può quindi raccogliere l'oggetto appena stampato e inserirlo nell'inventario. Successivamente è possibile visualizzare lo stato dell'oggetto nell'inventario , esaminando i valori degli attributi associati e i metodi che implementa , nonchè il nome stesso dell'oggetto , della classe , e una breve descrizione sull'utilità dell'oggetto in relazione al contesto di escape room.

Metodi

Gli oggetti prodotti dalla stampante possono essere integrati in puzzle dedicati.All'interno di ogni livello sono presenti delle postazioni con cui è possibile

interagire , visualizzando gli oggetti di programmazione presenti nell'inventario e selezionando quello che si vuole posizionare all'interno della postazione. Una volta collocato un oggetto , è possibile esaminare lo stato degli attributi e invocare i metodi associati . I metodi fanno accesso agli attributi dell'oggetto secondo le relazioni dichiarate in fase di costruzione della classe, e accettano eventuali input esterni. Una volta invocato, un metodo genera un output visibile nel meccanismo del puzzle. L'output di un metodo potrebbe essere l'apertura di una porta , oppure in altri casi si possono visualizzare alcune informazioni dell'oggetto tramite dei monitor collegati alla postazione. L'obiettivo del giocatore è dunque quello di porre gli oggetti corretti sulle varie postazioni , e risolvere l'enigma a cui sono collegate queste ultime , chiamando i metodi con i valori richiesti su ogni oggetto. Infine, possono esserci oggetti che implementano metodi che richiedono l'accesso ad attributi di altri oggetti posizionati nel puzzle , in questo modo è possibile introdurre il concetto di Regole di visibilità .

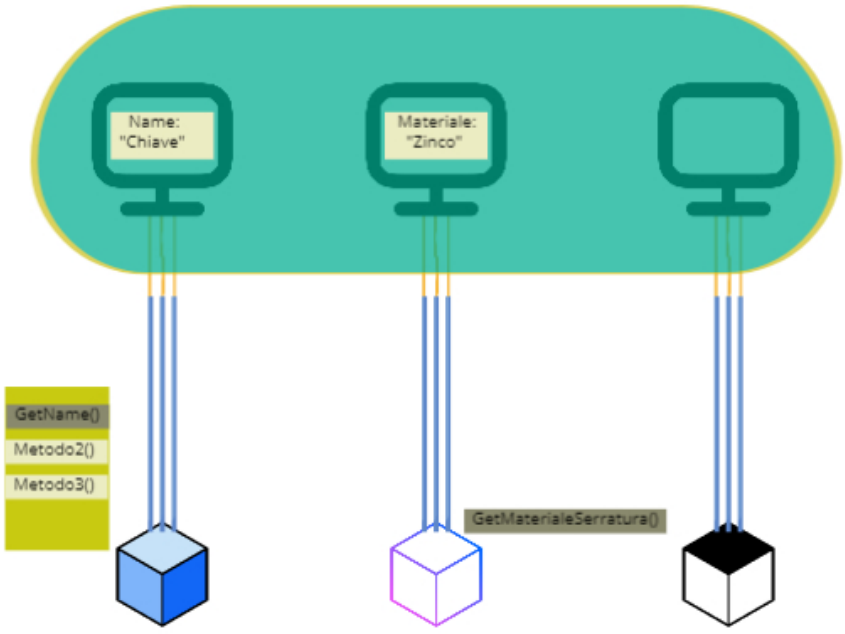

Figura 2.2. Puzzle Metodi

Regole di visibilità

La visibilità degli attributi svolge un ruolo fondamentale per la risoluzione di alcuni puzzle. Nel caso in cui un metodo faccia riferimento ad attribbuti di un altro oggetto , è necessario che questi attributi siano dichiarati come public , altrimenti il metodo non riuscirà a leggerli. Gli attributi possono essere dichiarati public o private in due modi distinti. Nel minigioco della creazione delle classi, è possibile definire la visibilità degli attributi , semplicemente cliccando sul bottone associato . Se, in un momento successivo del gioco, ci si rende conto che un attributo dovrebbe essere pubblico anziché privato, bisogna accedere al terminale di generazione delle classi , selezionare la classe specifica e successivamente tramite l'opzione modifica classe è possibile modificare la visibilità di ogni attributo.

2.3 Design degli elementi di gaming

Il gioco presenta tre meccaniche principali : EscapeRoom , ClassCreator , 3D ObjectPrinter. La trama si svolge tramite la meccanica dell' EscapeRoom , in cui il giocatore è chiamato a risolvere gli enigmi di gioco per proseguire nella storia. In questa modalità sono presenti gli elementi di gaming che contraddistingono le EscapeRoom virtuali: un sistema di inventario per gli oggetti raccolti , interazioni con gli oggetti di scena , possibilità di analizzare nel dettaglio una parte della stanza tramite un cambio di camera. La meccanica di EscapeRoom permette la connessione con le altre due , tramite la creazione di classi e oggetti , rispettivamente con il ClassCreator e il 3D ObjectPrinter. Queste ultime due sono dunque le meccaniche principali della parte educativa del gioco.

ClassCreator

Il processo di costruzione delle classi coinvolge un minigioco, dove gli attributi e i metodi raccolti durante l'avventura devono essere collegati accuratamente tra loro. Per accedere alla creazione di una classe è necessario che l'utente disponga del progetto di classe , e di tutti i metodi e attributi richiesti dal progetto. Il progetto di classe è un indizio , che il giocatore può recuperare nella scena di escape , che ha le informazioni sui metodi e attributi da utilizzare e su come collegarli . Ad esempio, si consideri la classe "Studente" con l'attributo "data di nascita" e il metodo "GiorniAlCompleanno()", che utilizza l'attributo "data di nascita" per calcolare quanti giorni mancano al

compleanno dello studente. Per avviare il minigioco è necessario disporre del progetto della classe Studente , e aver raccolti tutti i metodi e attributi richiesti. Una volta avviato il minigioco , l'obiettivo è collegare i metodi agli attributi , come definito nel progetto di classe , facendo attenzione ad evitare ostacoli che possano interferire con la linea di connessione. È possibile definire la visibilità degli attributi, interagendo con un bottone apposito sopra il nome dell'attributo stesso . Durante il collegamento, è necessario attraversare ostacoli come sentinelle dell'IA , raffigurate da telecamere , incaricate di verificare che nessuno stia creando nuove classi. Altri ostacoli, come blocchi elettrici, rappresentano una sfida aggiuntiva, poiché toccarli comporterebbe la conclusione negativa del minigioco. Una volta completati tutti i collegamenti è possibile compilare la classe. Se tutti i collegamenti sono stati eseguiti correttamente, la classe viene generata con successo; in caso contrario, si manifesta un errore, richiedendo al giocatore di riconsiderare la disposizione degli attributi e dei metodi .

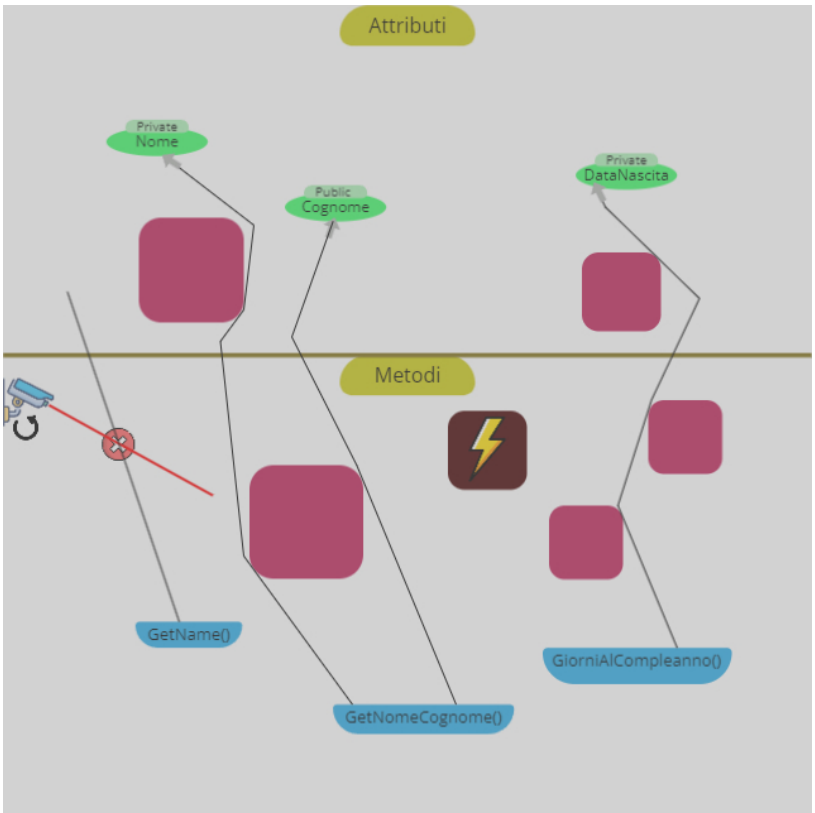

Figura 2.3. Class Creator

3D ObjectPrinter

L'idea del minigioco per la creazione degli oggetti è ispirata al puzzle utilizzato nel videogioco Marvel's Spider-Man [29]. In questo puzzle, l'obiettivo è far arrivare l'energia da un punto iniziale a uno finale, creando un percorso omogeneo attraverso cui farla scorrere. I giocatori hanno a disposizione un numero limitato di pezzi per costruire il percorso, i quali possono essere ruotati di 90°. La corrente ha un voltaggio iniziale che può essere manipolato da specifici pezzi del percorso, aggiungendo o sottraendo voltaggio. L'obbiettivo è garantire che il voltaggio finale corrisponda a quello desiderato per completare con successo il puzzle.

Figura 2.4. Marvel's Spider-Man , minigioco

Nel contesto di questo progetto , si adotta questo concetto per creare il circuito di stampa dell'oggetto. Il percorso da creare viene definito "percorso del costruttore", offrendo una comprensione pratica del concetto di costruttore di una classe. Ad ogni attribbuto è assocciato un percorso di costruzione, pertanto l'obiettivo è costruire il percorso collegando il punto iniziale con quello finale per assegnare all'attributo il valore richiesto.

In poche parole si sostituisce il concetto di "energia" usato in SpiderMan

con il valore dell'attributo. Gli attributi possono essere numeri interi , booleani o stringhe. Durante la costruzione del percorso, sono presenti blocchi che aumentano o diminuiscono il valore dell'attributo nel caso degli interi , blocchi che aggiungono un carattere alfanumerico nel caso delle stringhe , o blocchi che cambiano il valore logico nel caso di un attributo di tipo booleano. La sfida consiste nel costruire un percorso , che colleghi il blocco iniziale con quello finale , garantendo che il valore dell'attributo in esamina sia uguale al valore target di riferimento della classe. Dopo aver correttamente completato i percorsi di tutti gli attributi, l'oggetto sarà pronto per essere creato e stampato dalla stampante 3D. Il processo di costruzione di un oggetto richiede dunque il posesso della relativa classe . Interagendo con la stampante sarà infatti possibile selezionare la classe dall'elenco di classi presenti nell'inventario . E' inoltre richiesta l'assegnazione di un nome all'oggetto da creare , nel caso in cui sia già presente un oggetto con lo stesso nome , la stampante avviserà l'utente nel trovare un nome diverso da quelli degli oggetti già esistenti.

2.4 Livelli di gioco

Questa sezione esplora i vari livelli di gioco, ognuno caratterizzato da un contesto narrativo distintivo e obiettivi educativi specifici. Ogni livello è stato progettato per offrire un'esperienza coinvolgente e formativa, integrando in modo creativo la narrazione con gli elementi educativi del progetto.

1° Livello

Stanza di prigionia. Il giocatore inizia l'avventura in una cella della prigione dell'IA. Questo livello svolge la funzione di introdurre il contesto narrativo e gli oggetti di programmazione. Questi ultimi permettono di interagire con l'ambiente esterno tramite i metodi che implementano. Il protagonista dovrà dunque risolvere un'enigma per trovare la chiave della cella e passare al livello successivo.

2° Livello

Mensa. Usciti dalla cella della prigione ci si ritrova nella mensa dei detenuti. In questo contesto viene introdotta la stampante 3D e la relazione tra classi e oggetti. Il giocatore si troverà dunque nella condizione di dover risolvere

un'enigma per recuperare una classe , che gli sarà successivamente utile per creare l'oggetto necessario alla fuga dalla mensa.

3° Livello

Stanza delle caldaie. In questo livello il protagonista si finge un operatore della prigione e si trova costretto a sistemare un sistema di valvole per continuare la sua fuga. In questo contesto viene introdotta la creazione delle classi . Il giocatore dovrà recuperare metodi e attributi utili per creare la classe valvola , e successivamente potrà applicare i concetti precedentemente appresi per creare l'oggetto valvola e risolvere l'enigma legato alla stanza.

4° Livello

Sala dei creatori. Il protagonista riesce a raggiungere la sala dei creatori dell'IA , l'ultimo passo prima di accedere al nucleo di controllo dell'intelligenza artificiale. In questo livello vengono applicate le meccaniche introdotte precedentemente e viene anche introdotto il concetto di visibilità degli attributi. Il giocatore si troverà costretto a modificare la visibilità degli attributi della classe da lui creata , per potere risolvere correttamente l'enigma della stanza.

5° Livello

Nucleo di controllo IA. Quest'ultimo livello di gioco non introduce nuovi argomenti , ma mira a verificare che l'utente abbia appresso correttamente i concetti dei livelli precedenti. Ci si trova nella sala di controllo dell'IA , qui il protagonista ha la possibilità di spegnere definitivamente il computer dell'intelligenza artificiale e poter completare la sua fuga. Per farlo dovrà risolvere l'enigma legato alla stanza , usufruendo di tutte le nuove conoscenze apprese , ma avrà un tempo limitato per farlo , poichè l'IA che durante l'avventura si trovava in uno stato dormiente di aggiornamento , allo scadere del timer si sveglierà e metterà fine alla fuga del giocatore.

NOTA: questo livello è stato solamente ideato nel design , ma non è stato implementato nel progetto relativo a questa tesi.

2.5 Selezione tecnologica e dettagli implementativi

Questa sezione si concentra sulla scelta delle tecnologie impiegate nel processo di sviluppo del gioco, includendo un elenco dei software utilizzati e degli asset impiegati. Inoltre, verrà fornita una descrizione dell'implementazione del gioco, esaminando i processi e le metodologie adottate per sviluppare e integrare le varie componenti del progetto.

2.5.1 Software

Unity 3D

L'implementazione del gioco è stata realizzata tramite il game engine Unity 3D [51], un motore grafico multipiattaforma sviluppato da Unity Technologies, progettato per la creazione di applicazioni interattive e videogiochi. La decisione di utilizzare Unity è stata motivata da diversi fattori. Inizialmente, la scelta di avvalersi di un ambiente di sviluppo già consolidato ha consentito di accedere a una vasta gamma di strumenti e plugin creati da altri utenti e sviluppatori, permettendo di concentrare le risorse sul design dell'applicazione. Inoltre, la presenza di una community ampia e attiva associata a questo motore ha garantito risposte tempestive a ogni interrogativo emerso nel corso dello sviluppo. In particolare è stata adotatta la versione Unity 2022.3.14f1 , basando la scelta sulla sua qualifica LTS, che sta per "Long Term Support". La scelta di una versione LTS è motivata dalla garanzia di stabilità nel tempo, dalla ricezione di correzioni di bug, patch di sicurezza e supporto prolungato. Optare per una versione LTS, come la 2022.3.14f1, ha semplificato l'integrazione di plugin e risorse, riducendo il rischio di problemi di compatibilità e assicurando un ambiente di sviluppo affidabile per l'intero ciclo del progetto. Inoltre, la decisione di utilizzare l'ultima versione LTS disponibile all'inizio del progetto mira a garantire una base tecnologica stabile e sostenuta nel tempo.

Vectr

Vectr [56] è un'applicazione di grafica vettoriale accessibile attraverso un'interfaccia web. Vectr è noto per la sua capacità di creare e modificare immagini vettoriali attraverso l'utilizzo di equazioni matematiche, garantendo la scalabilità senza perdita di qualità. Questo strumento è stato utilizzato

per ridimensionare immagini senza compromettere la loro qualità, un aspetto rilevante per mantenere la chiarezza visiva nel contesto del progetto. Inoltre, Vectr è stato impiegato per la creazione di immagini e icone personalizzate, sfruttando la sua intuitiva interfaccia e versatilità per ottenere risultati visivi coerenti e di alta qualità.

Mixamo

Mixamo [5] è un servizio di animazione online, attualmente di proprietà di Adobe, che offre una vasta selezione di animazioni predefinite per personaggi 3D. Questa piattaforma è stata progettata per semplificare il complesso processo di animazione, fornendo agli sviluppatori, animatori e designer la possibilità di integrare movimenti realistici nei loro modelli senza dover creare ogni singolo frame manualmente. Mixamo è stato utilizzato per animare i personaggi inclusi nel progetto. Grazie alla ricca libreria di animazioni fornita da Mixamo, è stato agevolato il processo di applicazione di una diversificata gamma di movimenti, azioni e gesti ai modelli 3D utilizzati. L'utilizzo di Mixamo ha semplificato in modo significativo l'animazione, consentendo di integrare nel lavoro sequenze di movimenti di alta qualità .

2.5.2 Assets , Models & Sounds

Nel processo di sviluppo del gioco, sono stati utilizzati diversi asset dello store di Unity, ognuno dei quali ha contribuito in modo significativo all'estetica e alle funzionalità del progetto. Tra questi si citano : Sci-Fi Construction Kit [27], utilizzato per creare l'ambientazione di gioco; Cartoon FX Remaster Free [37] , utilizzato per gli effetti visivi; Starfield Skybox [11], utilizzato per creare un ambiente di gioco che trasmettesse l'illusione di trovarsi nello spazio; Quick Outline [38], questo asset ha permesso di generare i contorni sugli oggetti di gioco interagibili; Sci-fi GUI skin [1] e 80 Sci-Fi Irregular Frame In One [15] , utilizzati per le interfaccie GUI; StarterAssets - FirstPerson [50] , standard asset di unity , utilizzato per implentare il controllo in prima persona. Il font testuale utilizzato è stato scaricato dal sito dafont [16].

I modelli 3D utilizzati nel progetto sono stati acquisiti sia dagli asset menzionati sia da Sketchfab. [47] . Quest'ultimo è una piattaforma online per la visualizzazione e la condivisione di modelli 3D interattivi. Alcuni specifici modelli , come la stampante 3D , sono stati creati e animati dallo studente Matteo Biffoni , che ha contribuito all'implementazione di una parte del progetto.

La maggior parte delle icone utilizzate sono state scaricate dal sito web flaticon [21] . I suoni e le canzoni utilizzate sono state rispettivamente scaricate dai siti freesound [24] e bensound [9] .

2.5.3 Dettagli Implementativi

Nella seguente sezione, verrà presentata un'analisi della struttura del sistema implementato, focalizzandosi sull'architettura delle classi e sulle loro relazioni all'interno del gioco. Si forniranno diagrammi che illustrano in modo chiaro e conciso le classi principali del sistema, così come le loro associazioni e dipendenze. Questa rappresentazione visiva sarà accompagnata da una descrizione testuale delle principali entità e delle loro interazioni, fornendo un quadro completo dell'organizzazione e del funzionamento del sistema implementato.

Il ClassDictionary è la classe responsabile della gestione della composizione delle classi nel gioco. Il suo attributo principale è una lista di ClassValue, nella quale è possibile inserire informazioni direttamente dall'editor di Unity. Per ogni elemento della lista, è possibile specificare un nome, una descrizione e il prefab dell'oggetto che rappresenta la classe. Inoltre, è presente una lista di AttributeValue, che rappresentano gli attributi della classe e sono caratterizzati da visibilità e metodi associati. I metodi sono definiti da un nome e da un tipo dell'enum MethodType, che indica la tipologia del metodo e permette di eseguire azioni diverse: ad esempio, un metodo setter può cambiare i valori degli attributi, mentre un getter restituirà il valore degli attributi. Inoltre, un interactor controllerà che gli attributi soddisfino determinate condizioni, come quelle richieste da un enigma. Il metodo FindClass del ClassDictionary consente di cercare una classe nella lista di ClassValue utilizzando il suo nome; in caso di esito positivo, la classe viene restituita sotto forma di dizionario. Questa trasformazione è utile per semplificare i calcoli che avvengono in varie situazioni, come ad esempio nei minigiochi.

Si presenta una dualità nella rappresentazione delle classi all'interno del sistema. In questo contesto, la rappresentazione di una classe può assumere due forme differenti, ciascuna con i propri vantaggi e finalità specifiche. Per quanto riguarda il ClassCreator, risulta utile adottare una rappresentazione basata su una lista di attributi, dove ciascun attributo è accompagnato da una lista dei metodi che lo utilizzano. Questo approccio semplifica la verifica dei collegamenti tra metodi e attributi, particolarmente durante l'esecuzione dei minigiochi, garantendo una valutazione rapida ed efficiente. D'altro canto, gli oggetti di gioco, richiedono una rappresentazione inversa della classe.

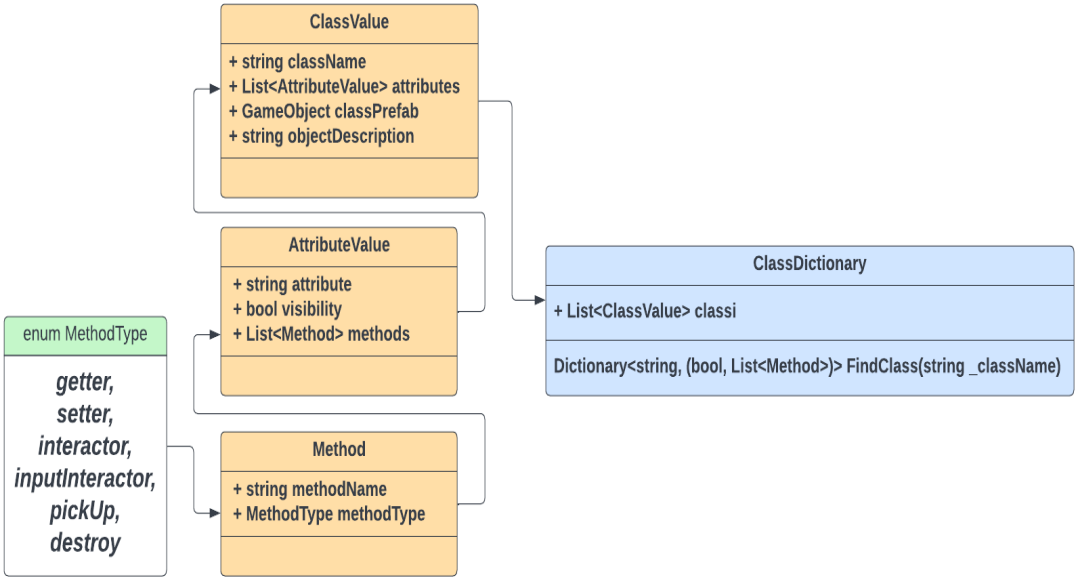

Figura 2.5. Diagramma , ClassDictionary

In questo caso, la rappresentazione consiste in una lista di metodi, ognuno dei quali associato a una lista degli attributi a cui fa riferimento. La trasformazione della classe in un dizionario consente al sistema di adattarsi con flessibilità a entrambe le rappresentazioni, riducendo al minimo la complessità delle operazioni necessarie per gestire tali variazioni.

Ogni oggetto di programmazione implementa la classe OggettoescapeValue. Se l'oggetto è stato creato tramite la stampante, i suoi valori vengono impostati dal minigioco di creazione degli oggetti attraverso le informazione reperite dal dizionario della classe; altrimenti, vengono impostati manualmente nell'editor di Unity. OggettoescapeValue è composta da una variabile booleana che indica se l'oggetto è stato creato dalla stampante, il nome dell'oggetto, il nome della classe da cui deriva, una descrizione dell'oggetto, il prefab che rappresenta l'istanza del gameobject e l'objectInteractorID che indica in quale eventuale ObjectInteractor è stato inserito.

Ogni oggetto implementa una lista di Methos (rappresentazione della classe nel caso degli oggetti), dove ogni Methos è composto da un Method e da una lista di stringhe che rappresentano gli attributi a cui il metodo fa riferimento. È inoltre presente una lista di Attribute, dove ogni Attribute è

costituito da un nome e un valore dell'attributo. La scelta di inserire un secondo riferimento agli attributi con i rispettivi valori, anziché inserire i valori direttamente negli attributi di ogni Methos, è motivata dal fatto che è più agevole in questo modo accedere agli attributi. Senza questa seconda lista, l'accesso agli attributi avrebbe richiesto un tempo elevato, specialmente nei casi in cui non fosse necessario accedere ai metodi; sarebbe stato necessario cercare nella lista di Methos e recuperare tutti gli attributi di ogni Methos con i rispettivi valori. Pertanto, si è scelto di privilegiare le prestazioni, anche a discapito di un leggero aumento del consumo di memoria.

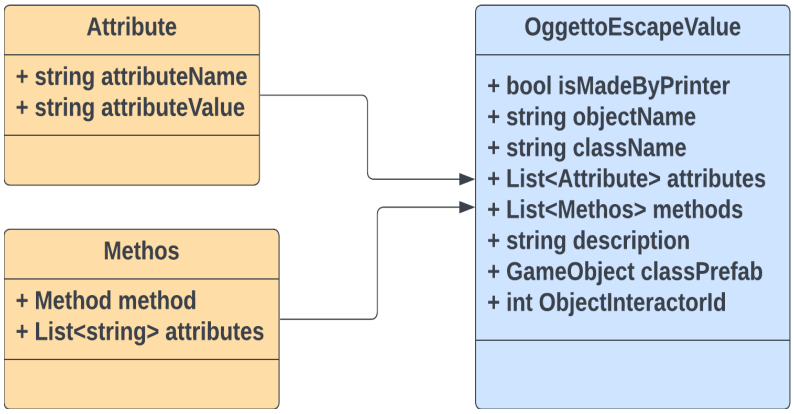

Figura 2.6. Diagramma , OggettoEscapeValue

La classe OggettoEscapeValue non eredita i comportamenti degli oggetti Unity tramite l'ereditarietà della classe Monobehaviour. Per gestire le interazioni con l'ambiente di gioco, viene creata una classe più astratta chiamata OggettoEscape, che fornisce un attributo OggettoEscapeValue e implementa altre funzionalità. Presenta inoltre anche un attributo di tipo Clue.

La classe Clue viene implementata da tutti gli oggetti che possono essere raccolti e inseriti nell'inventario. Essa contiene un enum ClueType che indica la tipologia dell'indizio (teoria, attributo, oggetto, ecc.), oltre a un nome e una descrizione dell'indizio. Quando l'utente interagisce con un oggetto di tipo Clue, questo viene inserito nell'inventario nella tipologia specificata da ClueType .

L'interazione con gli oggetti avviene attraverso l'interfaccia Interactable, che viene implementata sia da Clue che da OggettoEscape. Grazie all'utilizzo di questa interfaccia, ogni volta che il giocatore si avvicina a un oggetto di questo tipo, viene evidenziato con un contorno per indicare la possibilità di interazione. Quando il giocatore preme il tasto di interazione, il metodo

Interact dell'oggetto viene chiamato. Nel caso degli oggetti di tipo Clue, questo metodo viene sovrascritto per inserire il clue nell'inventario.

OggettoEscape include un riferimento a MethodListener, che viene aggiornato dinamicamente ogni volta che un oggetto viene posizionato su un meccanismo di puzzle di gioco in cui è possibile richiamare metodi. Il MethodListener ha il compito di ascoltare l'invocazione di questi metodi ed eseguire azioni corrispondenti. L'invocazione dei metodi viene effettuata tramite il metodo CallMethod di OggettoEscape. Ogni volta che viene chiamato un metodo sull'oggetto dall'interfaccia, CallMethod esegue diverse azioni sul MethodListener in base al tipo di metodo chiamato. Il metodo da chiamare viene passato come stringa nell'argomento della funzione.

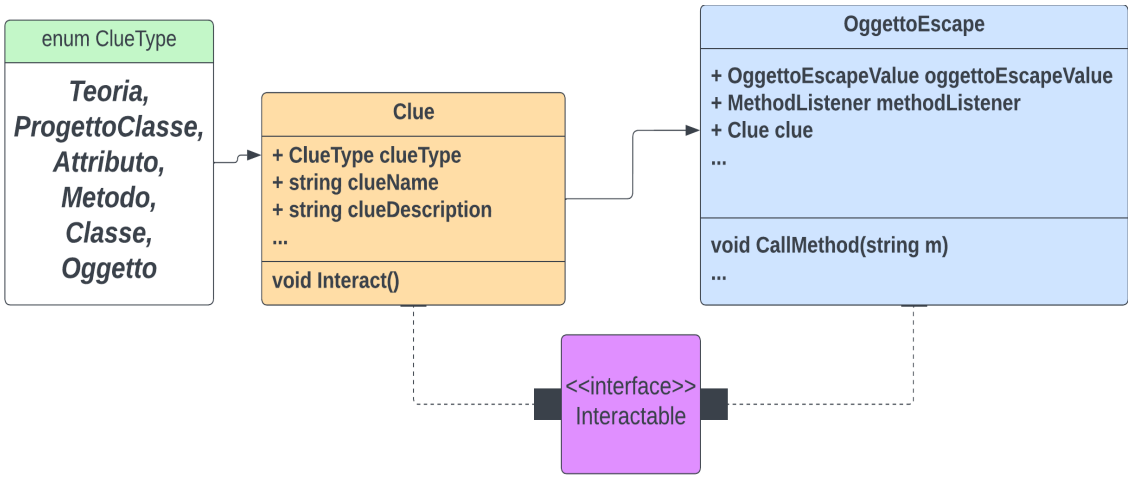

Figura 2.7. Diagramma , OggettoEscape

Ogni OggettoEscape può essere inserito in appositi oggetti di scena, come ad esempio la serratura di una porta, dove è possibile richiamare i suoi metodi. Questi oggetti di scena implementano la classe ObjectInteraction, che include un riferimento all'oggetto inserito, un ID univoco che lo distingue dagli altri oggetti della stessa classe e un riferimento a un MethodListener.

Il MethodListener verifica se l'oggetto inserito è quello richiesto dall'enigma confrontando l'attributo className con il nome della classe dell'oggetto inserito. Inoltre, il MethodListener è composto da un ID e due liste contenenti nome e valori degli attributi. La prima lista, AttributeValueListener, specifica quali attributi e valori devono essere presenti nell'oggetto per risolvere l'enigma. La lista ObjectValue, invece, rappresenta i valori degli attributi dell'oggetto inserito.

I metodi del MethodListener sono progettati per verificare la corrispondenza tra le due liste oppure, nel caso del metodo Getter, mostrare solo la lista dell'OggettoEscape.

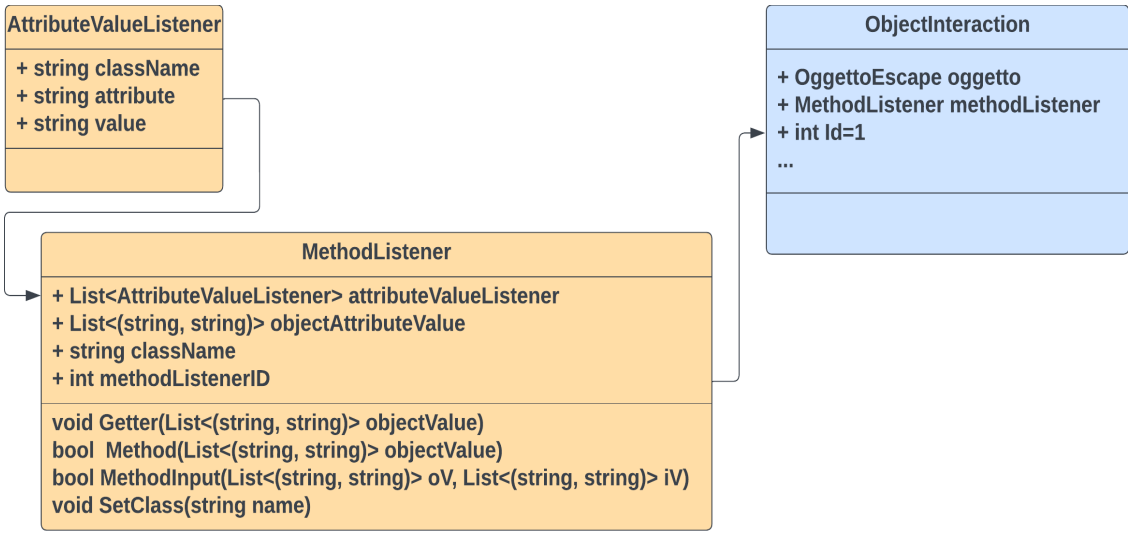

Figura 2.8. Diagramma , ObjectInteraction

Il MethodListener è implementato come un'interfaccia, consentendo così a tutti gli oggetti che implementano gli enigmi di ereditare le funzionalità di base e, eventualmente, di aggiungere ulteriori funzionalità . Ad esempio, la classe Door eredita le funzionalità del MethodListener e, in seguito alla corretta verifica dell'enigma, implementa la funzionalità di apertura della porta.

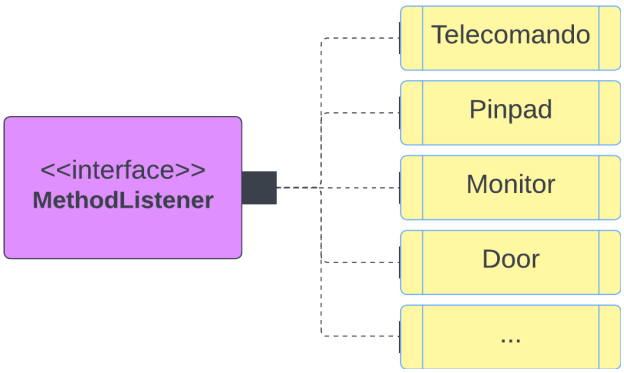

Figura 2.9. Diagramma , MethodListener interface

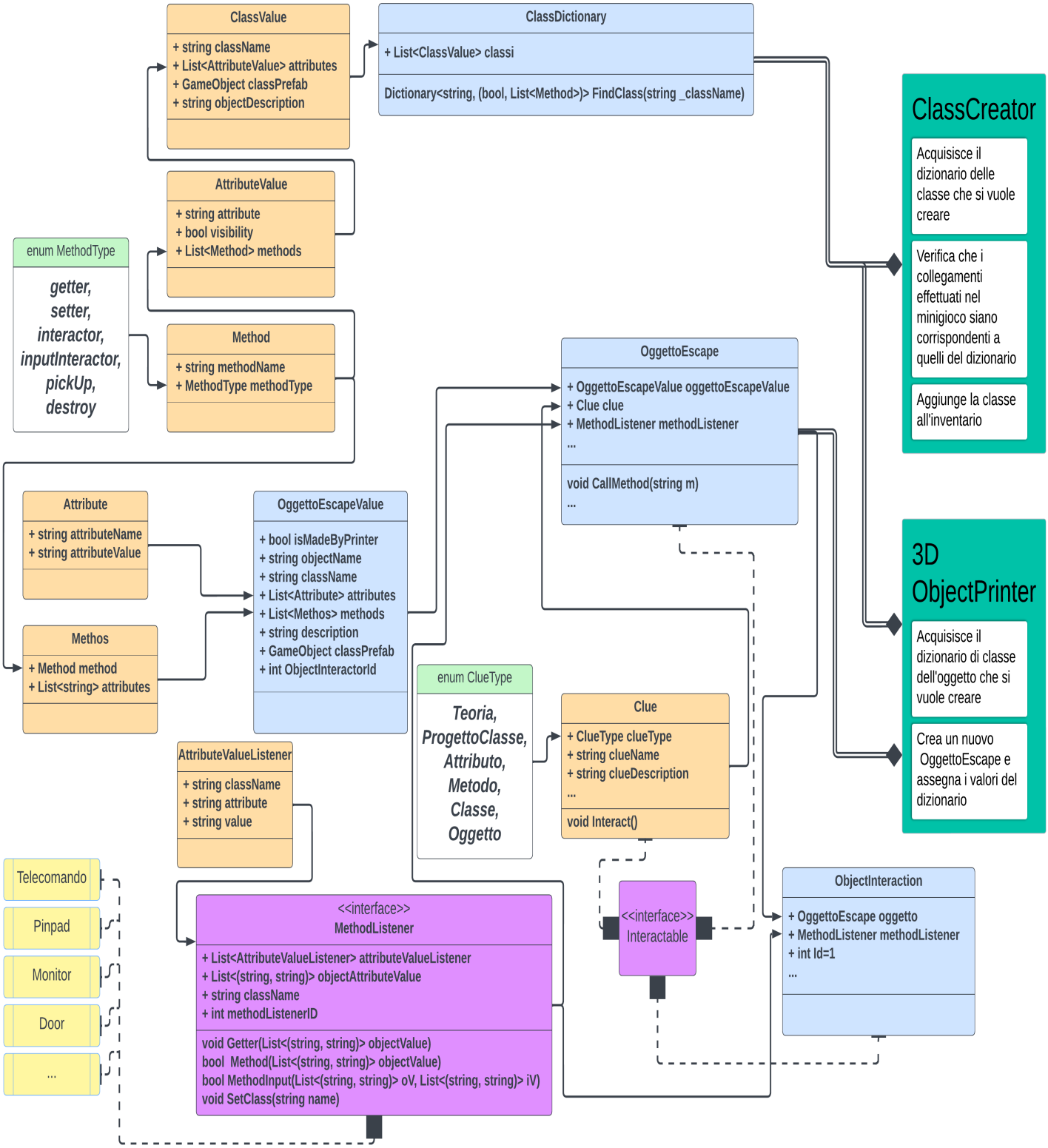

Figura 2.10. Diagramma , schema completo

Capitolo 3

Validazione e risultati

3.1 Validazione

3.1.1 Criteri di validazione

Nel corso della presente ricerca, è stato adottato il GAMEX [19] come strumento chiave di valutazione dell'esperienza di gioco . Il GAMEX è una scala di misurazione progettata per valutare l'esperienza ludica degli utenti in contesti non ludici, come ad esempio l'interazione con applicazioni gamificate. Questa scala identifica e quantifica diversi aspetti dell'esperienza di gioco, tra cui il divertimento, l'assorbimento, il pensiero creativo, l'attivazione, l'assenza di emozioni negative e la dominanza.

La sua utilità risiede nella capacità di fornire una valutazione dettagliata ed empirica dell'esperienza ludica, consentendo una comprensione approfondita delle risposte emotive e del coinvolgimento generato durante l'uso del gioco.

Al fine di validare il serious game, è stato condotto un test su un campione di 10 partecipanti, sottoponendoli all'esperienza di gioco e utilizzando il GAMEX per raccogliere dati sulle loro percezioni. Nella Tabella 3.1 sono riportate le domande del Gamex, valutate numericamente dagli utenti su una scala da 1 a 7.

È importante sottolineare che l'obiettivo principale della validazione era focalizzato sulla user experience, piuttosto che sull'efficacia educativa del gioco , pertanto i partecipanti selezionati non rappresentavano il target ideale di riferimento del serious game, in quanto possedevano già una solida conoscenza della programmazione a oggetti. La validazione mirava a esplorare le reazioni degli utenti, i livelli di coinvolgimento e gli aspetti emotivi generati dal

serious game, escludendo eventuali ostacoli derivanti dalla complessità educativa del gioco. Tale approccio ha consentito di ottenere un'analisi più accurata e centrata sull'aspetto dell'usabilità e dell'esperienza utente, evitando che la conoscenza pregressa della materia potesse influenzare negativamente il giudizio.

La valutazione dell'impatto educativo del gioco rappresenta un possibile sviluppo futuro del lavoro di ricerca.

Tabella 3.1: Domande questionario valutativo

Criterio di valutazione	Domande
Divertimento	• Giocare al gioco è stato divertente • Mi è piaciuto giocare al gioco • Mi sono divertito tanto giocando al gioco • La mia esperienza di gioco è stata piacevole • Penso che giocare al gioco sia molto divertente • Giocherei a questo gioco per il piacere del gioco stesso, non solo quando me lo chiedono
Immersività	• Giocare al gioco mi ha fatto dimenticare dove sono • Mi sono dimenticato del mio ambiente circostante mentre giocavo • Dopo aver giocato al gioco, mi sentivo come se stessi tornando al "mondo reale" dopo un viaggio • Giocare al gioco mi ha allontanato da tutto • Mentre giocavo al gioco ero completamente ignaro di tutto ciò che mi circondava • Mentre giocavo al gioco ho perso la percezione del tempo
Dominanza	• Mentre giocavo al gioco, avevo la sensazione di essere al comando • Mentre giocavo al gioco, mi sentivo influente • Mentre giocavo al gioco, mi sentivo autonomo • Mentre giocavo al gioco, mi sentivo sicuro

Continuation of Table 1.1	
Criterio di valutazione	Domande
Attivazione	• Mentre giocavo al gioco, mi sentivo attivo • Mentre giocavo al gioco, mi sentivo nervoso • Mentre giocavo al gioco, mi sentivo frenetico • Mentre giocavo al gioco, mi sentivo eccitato
Pensiero creativo	• Giocare al gioco ha scatenato la mia immaginazione • Mentre giocavo al gioco, mi sentivo creativo • Mentre giocavo al gioco, sentivo di poter esplorare cose nuove • Mentre giocavo al gioco, mi sentivo avventuroso
Possibili effetti negativi	• Mentre giocavo al gioco, mi sentivo infastidito • Mentre giocavo al gioco, mi sono sentito ostile • Mentre giocavo al gioco, mi sono sentito frustrato

3.1.2 Risultati validazione

A causa del ridotto campione di 10 utenti, si è scelto di fare affidamento esclusivamente sulla media come strumento principale per valutare i risultati.

Nella rappresentazione grafica che segue, vengono riportate le valutazioni medie di ogni utente per i diversi criteri valutativi. Sull'asse delle ascisse sono indicati gli utenti, mentre sull'asse delle ordinate sono riportati i valori medi delle risposte per ogni sezione analizzata . La media generale viene rappresentata con una linea rossa.

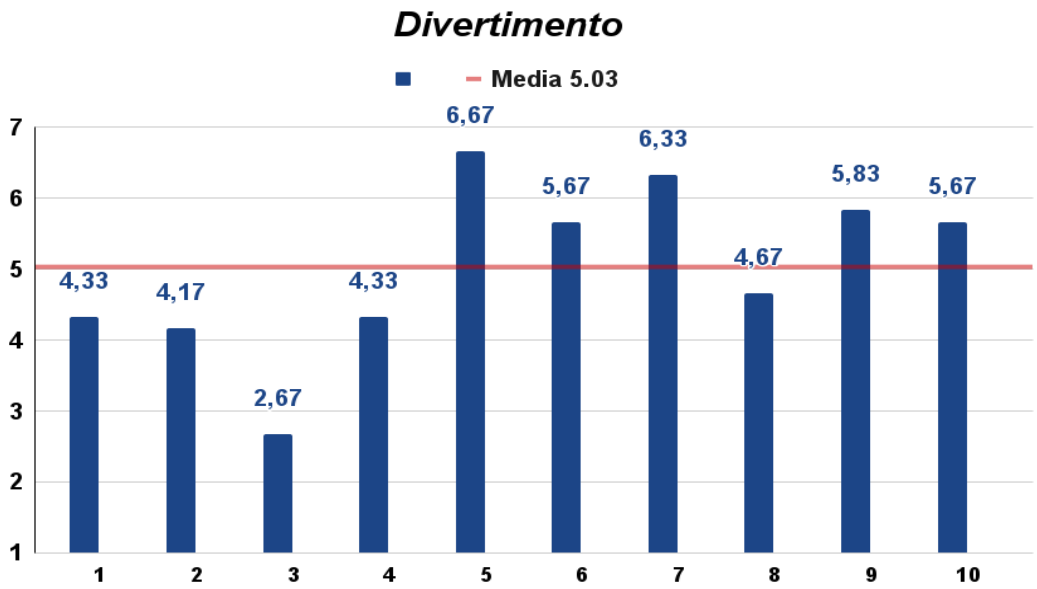

Figura 3.1. Grafico , Divertimento

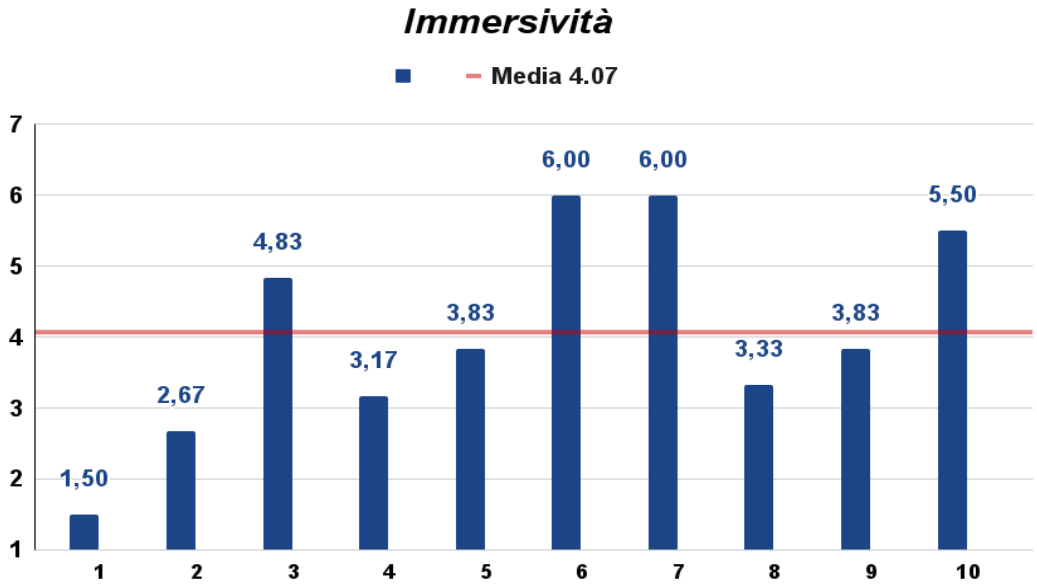

Figura 3.2. Grafico , Immersività

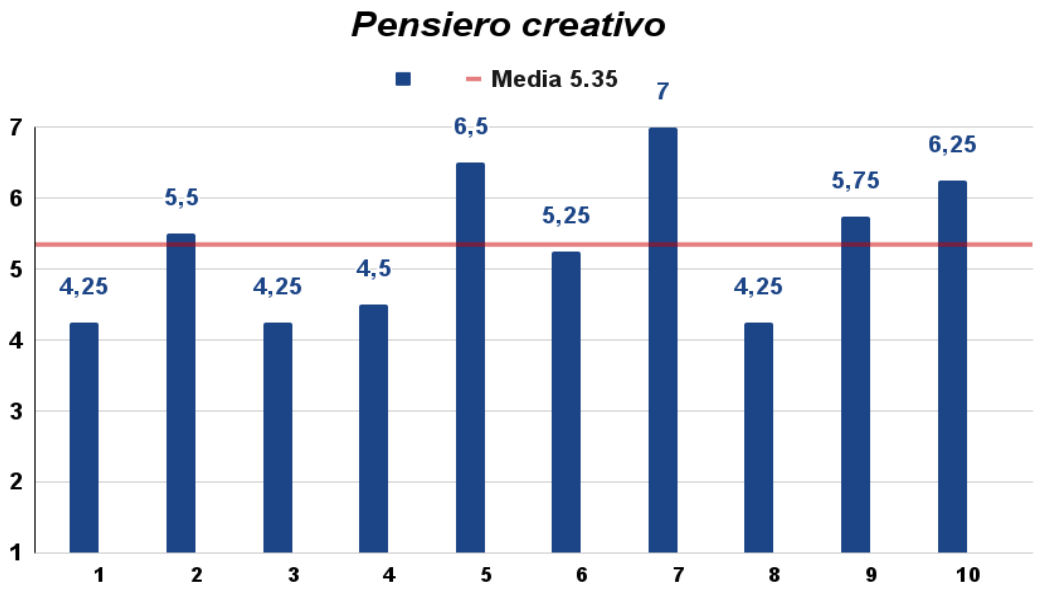

Figura 3.3. Grafico , Pensiero creativo

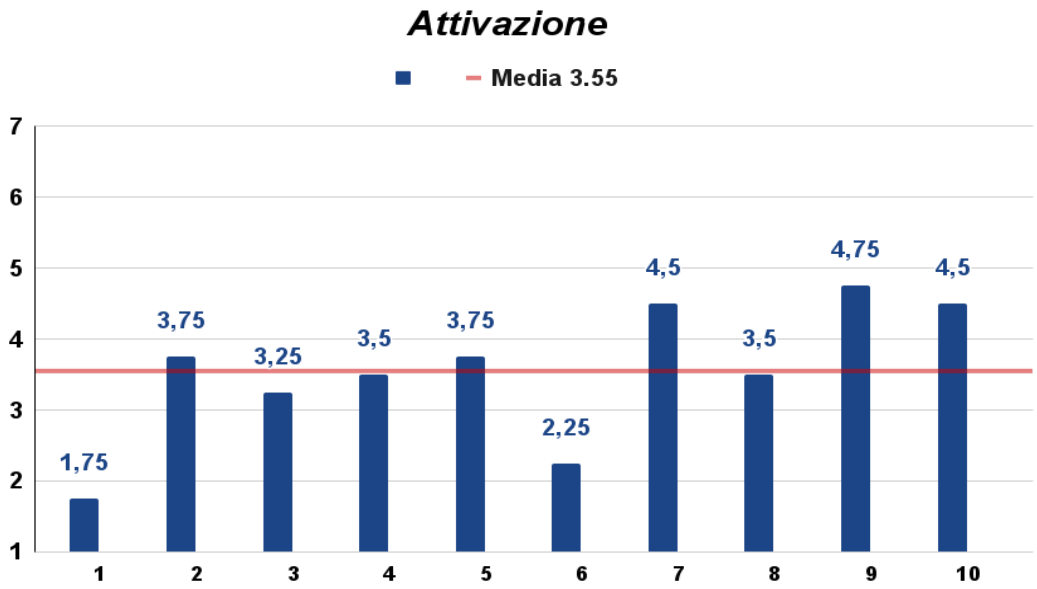

Figura 3.4. Grafico , Attivazione

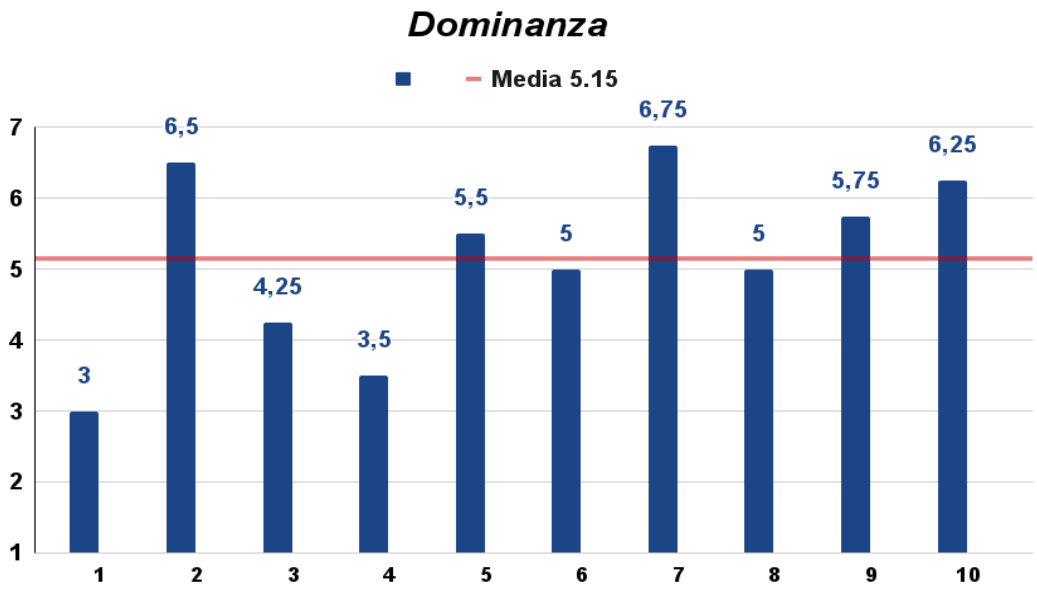

Figura 3.5. Grafico , Dominanza

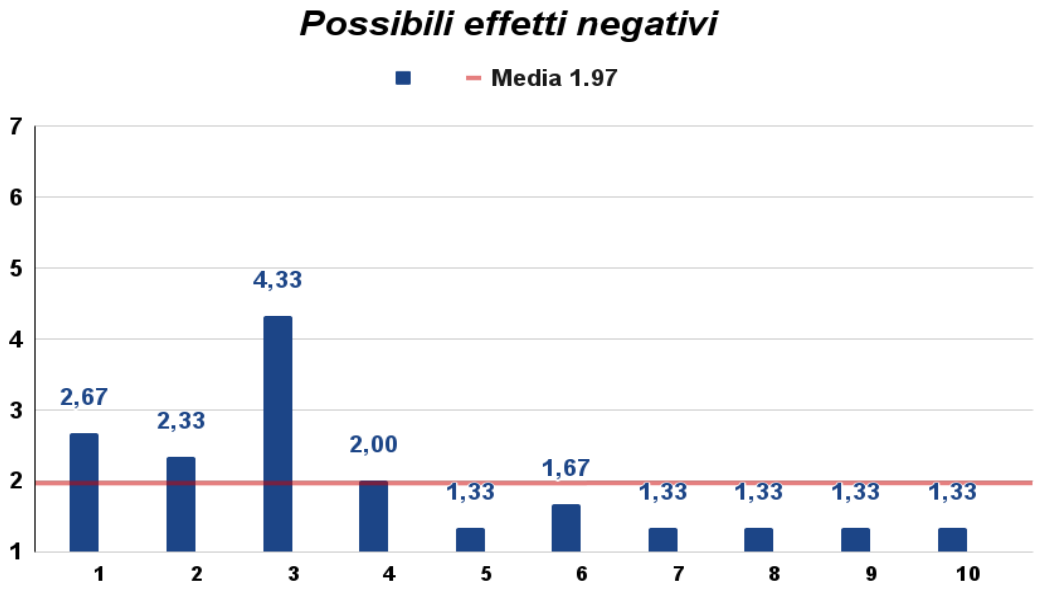

Figura 3.6. Grafico , Possibili effetti negativi

Dall'analisi dei grafici emerge che la maggioranza degli utenti che hanno valutato i possibili effetti negativi al di sopra della media sono gli stessi che hanno dato una valutazione al di sotto della media per gli altri settori. Tale coincidenza suggerisce una buona attendibilità del questionario. Questi utenti potrebbero rappresentare casi speciali che richiedono un'ulteriore indagine per comprendere meglio le loro esperienze e opinioni. In particolare, l'utente 3 ha assegnato una valutazione significativamente alta per i possibili effetti negativi e significativamente più bassa rispetto alla media per il fattore divertimento. Analizzando la recensione dell'utente 3, emerge che il FOV (campo visivo) della telecamera era per lui troppo basso e ciò ha influenzato negativamente la sua esperienza. Questo aspetto, non rilevato dagli altri utenti, potrebbe indicare un caso isolato che, vista la limitata partecipazione al sondaggio, ha comunque influito in modo significativo sulle medie complessive del questionario. A seguito di questa osservazione, sembra che il fattore divertimento sia stato valutato in modo attendibile, poiché escludendo il caso isolato dell'utente 3, le altre valutazioni oscillano intorno al valore medio. Anche per il fattore pensiero creativo, si osservano dati omogenei in cui i valori degli utenti si collocano intorno alla media senza presentare eccessi positivi o negativi. L'analisi complessiva dei risultati ottenuti in relazione agli obiettivi prefissati sarà discussa nel capitolo 4.

3.2 Illustrazioni del Prototipo Videoludico

3D Object Printer

In questa sezione si mostra la realizzazione della stampante 3D , la sua interfaccia e il minigioco di creazione degli oggetti .

Figura 3.7. 3D Object Printer

Figura 3.8. 3D Object Printer , Interfaccia per la selezione della classe per la creazione dell'oggetto e l'inserimento del nome dell'oggetto da creare.

Figura 3.9. 3D Object Printer , fase di stampa di un oggetto .

Figura 3.10. 3D Object Printer , Minigioco creazione oggetti , con attributo di tipo intero

Figura 3.11. 3D Object Printer , Minigioco creazione oggetti , con attributo di tipo stringa

ClassCreator

In questa sezione si mostra la realizzazione del ClassCreator , la sua interfaccia e il minigioco di creazione delle classi .

Figura 3.12. ClassCreator

Figura 3.13. ClassCreator , Interfaccia per la selezione del progetto di classe .

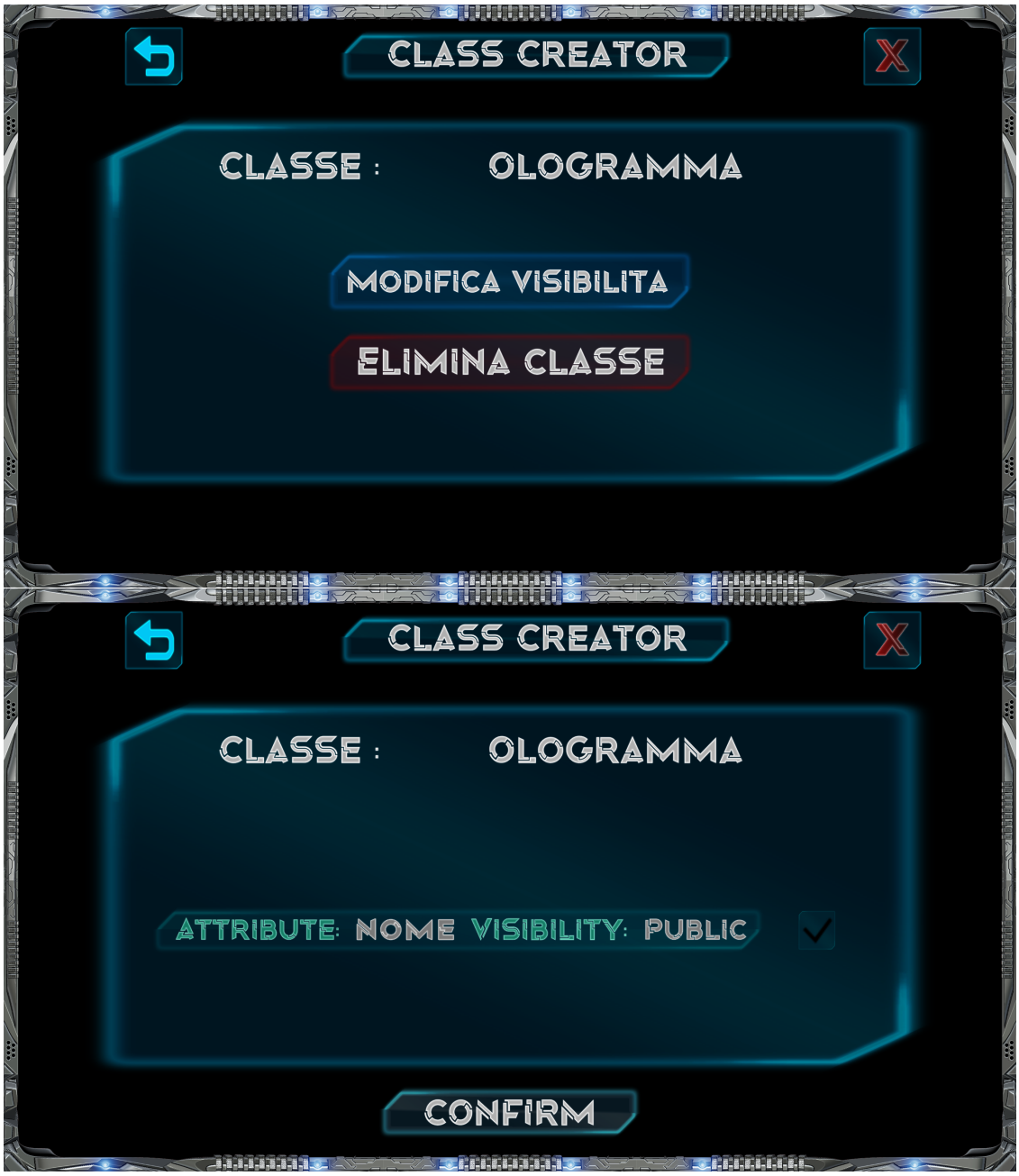

Figura 3.14. ClassCreator, Interfaccia della classe: permette l'eliminazione di una classe o la modifica della visibilità dei suoi attributi. Per cambiare la visibilità, è possibile utilizzare la casella di controllo associata all'attributo: selezionarla per impostare la visibilità come pubblica, deselezionarla per impostarla come privata.

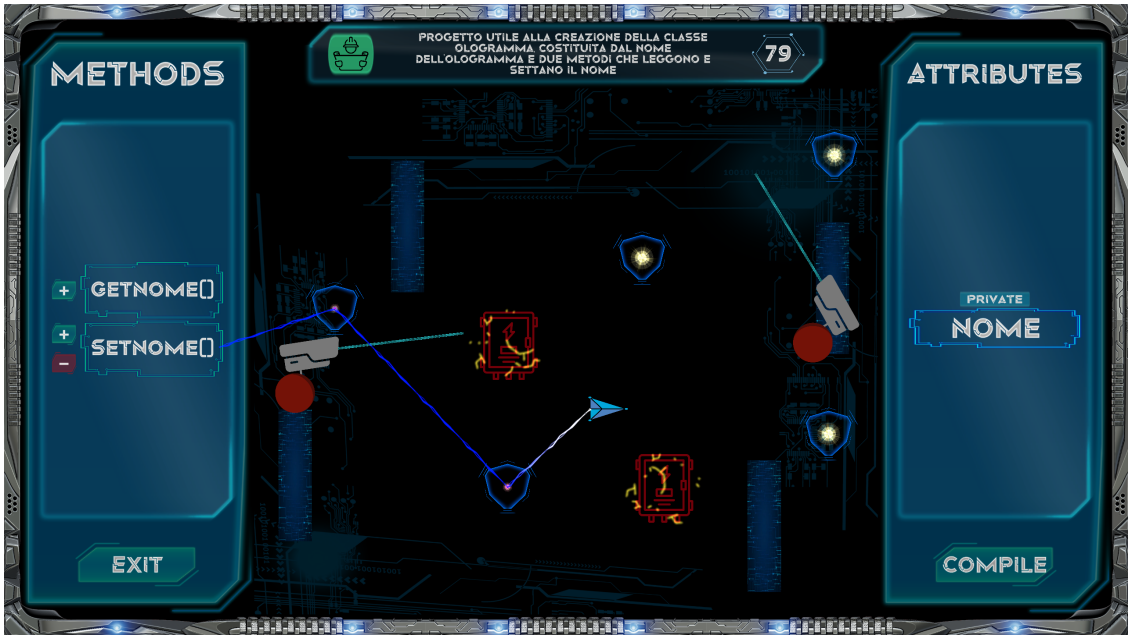

Figura 3.15. ClassCreator , Minigioco creazione classe , è necessario associare i metodi agli attributi.

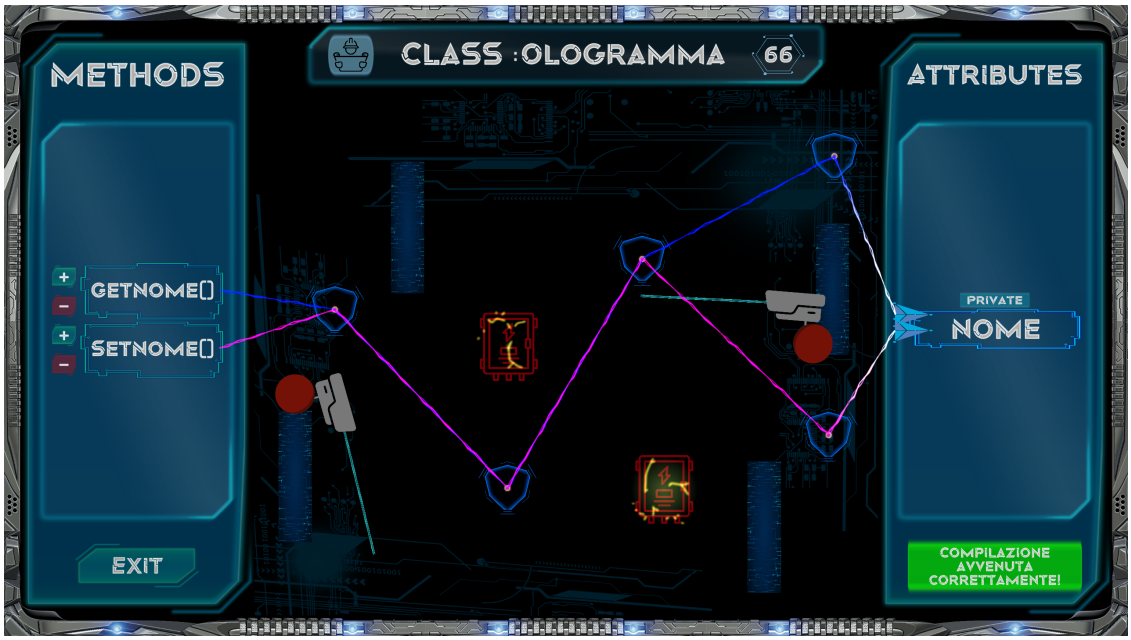

Figura 3.16. ClassCreator , Minigioco creazione classe , i collegamenti sono stati completati con successo e la compilazione ha avuto esito positivo.

Inventario

In questa sezione si mostra la realizzazione dell'inventario di gioco.

Figura 3.17. Inventario , Interfaccia che si attiva ogni volta che viene raccolto ed inserito nell'inventario un nuovo elemento. Per ogni elemento viene raffigurato il nome , la descrizione , e la tipologia (Teoria, Progetto di classe , Attributo, Metodo , Classe , Oggetto) .

Figura 3.18. Inventario , vengono rappresentati gli indizi di teoria raccolti. Selezionando un indizio è possibile visualizzarne la descrizione.

Figura 3.19. Inventario , Rappresentazione di una classe: a sinistra vengono elencati i metodi che implementa, mentre a destra sono indicati gli attributi con il loro stato di visibilità. Inoltre, vengono mostrate le relazioni tra attributi e metodi attraverso collegamenti visivi.

Figura 3.20. Inventario , Rappresentazione di un oggetto: selezionando un oggetto, è possibile visualizzare la classe da cui deriva, gli attributi con i relativi valori, i metodi che implementa e la sua descrizione .

Interfacce

Questa sezione si focalizza sull'illustrazione delle interfacce utente (UI) dedicate all'interazione con gli oggetti di scena . Le interfacce UI presentate offrono agli utenti la possibilità di selezionare gli oggetti di programmazione da inserire all'interno degli enigmi di gioco e di richiamare i metodi associati a tali oggetti.

Figura 3.21. Interfaccia di selezione oggetti.

Figura 3.22. Interfaccia dell'oggetto telecomando.

Figura 3.23. Interfaccia degli oggetti Cassaforte e TesseraOperatore.

Ambiente di gioco

Questa sezione offre uno sguardo sugli ambienti di gioco, i dialoghi e gli enigmi .

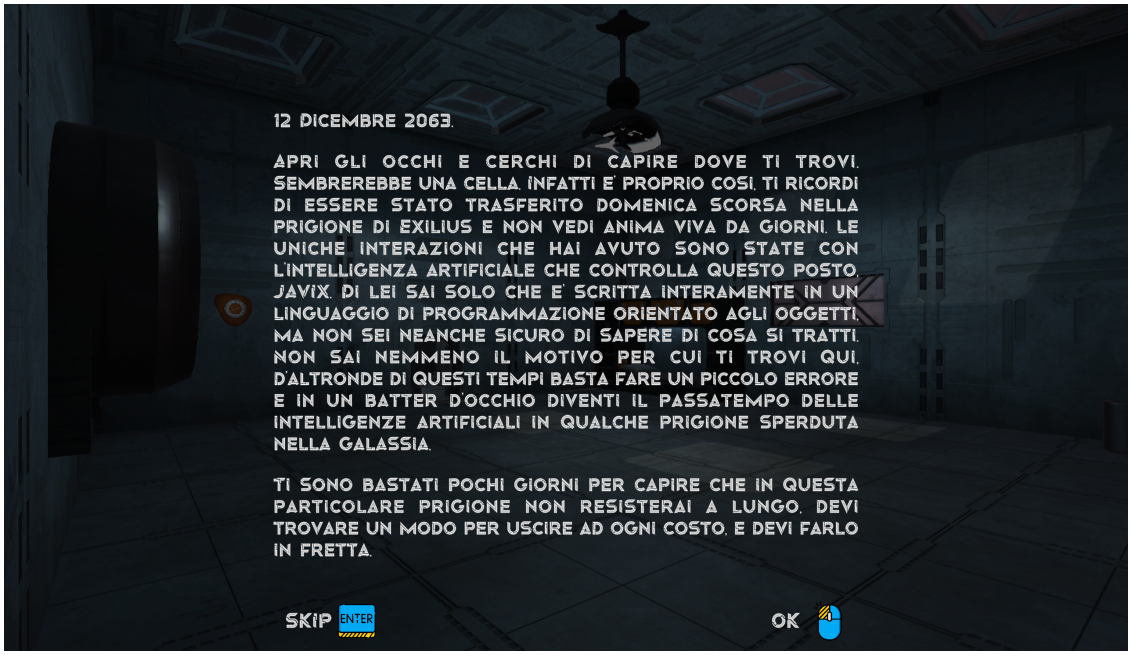

Figura 3.24. Uno dei dialoghi di introduzione presente all'inizio di ogni livello

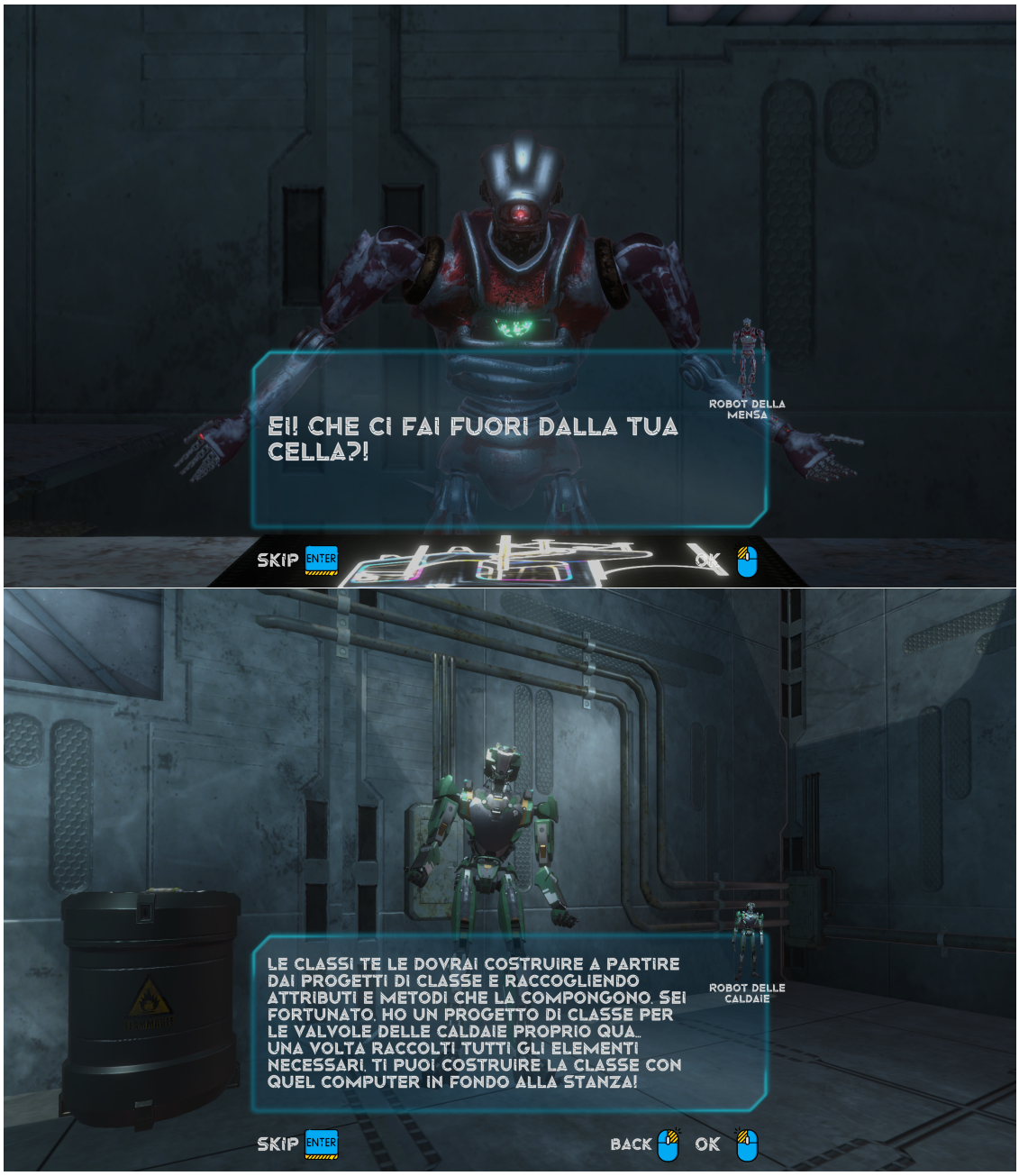

Figura 3.25. Dialoghi

Figura 3.26. Indizio , in alcuni casi in cui il giocatore impiegarà troppo tempo per risolvere un'enigma verrà mostrato un suggerimento per aiutarlo nella risoluzione.

Figura 3.27. Uno degli enigmi di gioco

Figura 3.28. Investigazione , in questa modalità è possibile indagare sugli oggetti di scena per trovare degli indizi.

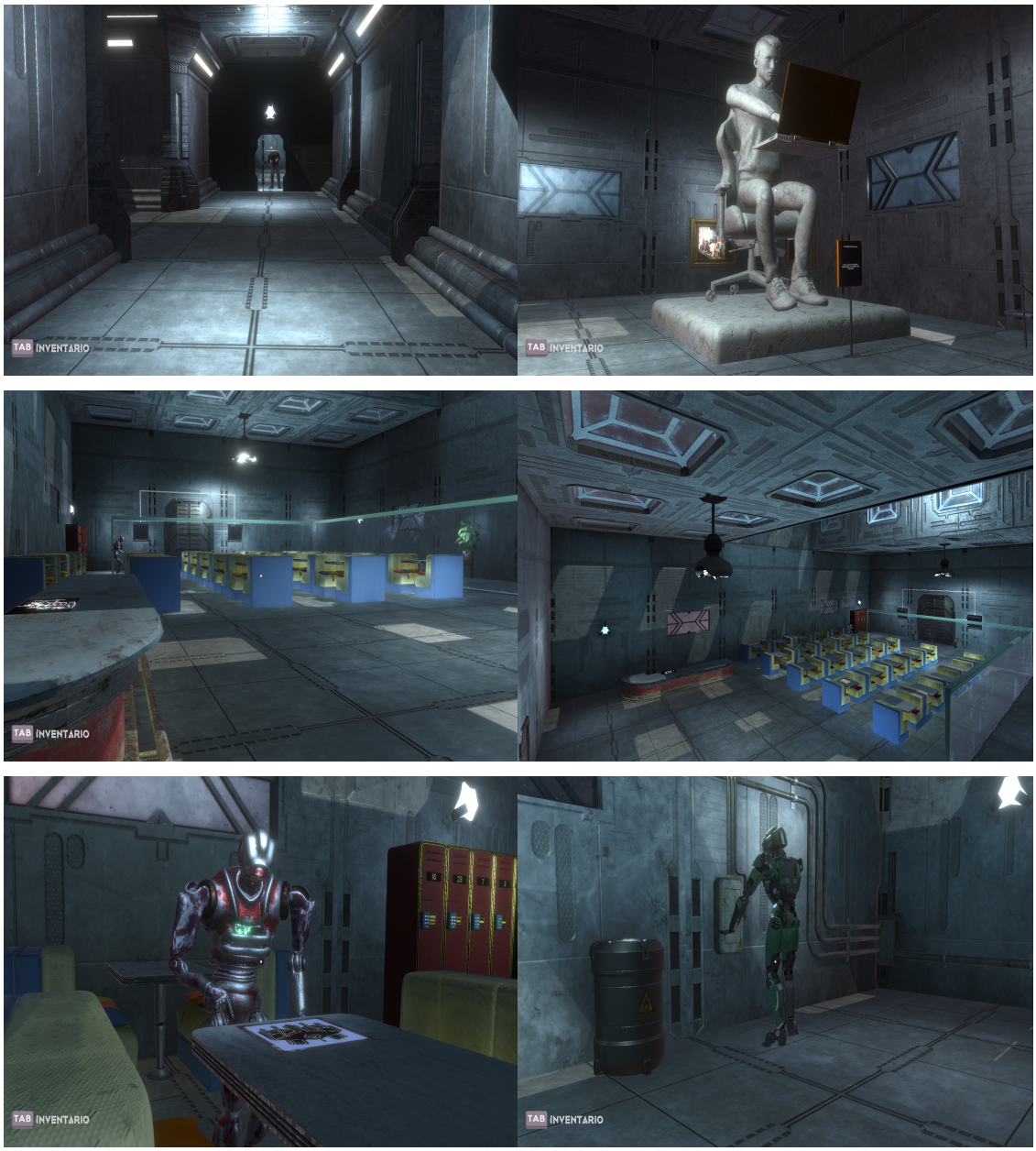

Figura 3.29. Scene di gioco

Figura 3.30. Scene di gioco

Capitolo 4

Conclusioni

4.1 Risultati ottenuti

Alla luce dei risultati emersi dall'analisi dei dati raccolti, è possibile trarre alcune conclusioni significative riguardo all'esperienza utente del serious game sviluppato per l'insegnamento della programmazione a oggetti.

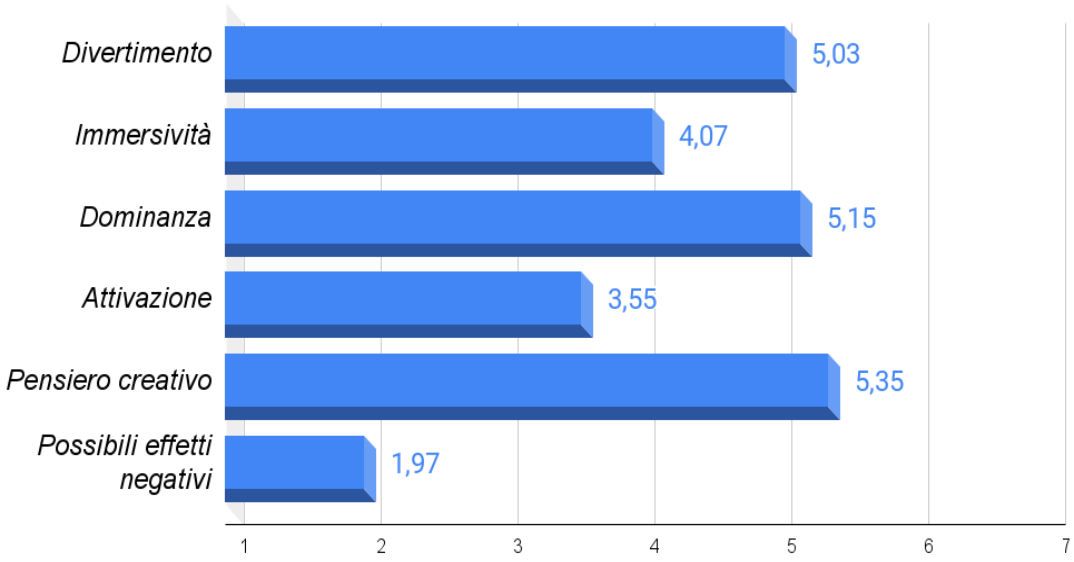

Figura 4.1. Grafico , Risultati Gamex

I dati raccolti hanno evidenziato una valutazione complessivamente positiva da parte dei partecipanti. I risultati sono stati raggruppati per sezione e valutano le medie delle valutazioni degli utenti per ogni sezione. Le valutazioni sono risultate elevate, superando il punteggio 5, per i criteri di divertimento, dominanza e pensiero creativo. Questi risultati indicano che il gioco è stato in grado di coinvolgere e intrattenere gli utenti, stimolando al contempo la loro creatività e il pensiero critico.

È fondamentale sottolineare che i possibili effetti negativi hanno ricevuto una valutazione relativamente bassa, indicando un'esperienza complessivamente positiva per la maggior parte dei partecipanti.

I punteggi per i criteri di immersività e attivazione sono risultati leggermente inferiori rispetto agli altri, indicando possibili margini di miglioramento. Analizzando le recensione degli utenti è emersa la necessità di alcuni miglioramenti e ottimizzazioni per rendere l'esperienza di gioco più fluida e accessibile. Questi interventi potrebbero contribuire a un aumento dell'immersività e dell'attivazione dei giocatori, migliorando complessivamente la qualità dell'esperienza di gioco. Gli ambiti di miglioramento identificati verranno discussi nella sezione relativa agli sviluppi futuri.

È importante evidenziare che il gioco è stato testato su diversi computer, inclusi quelli di fascia bassa senza scheda video dedicata, e ha garantito un'esperienza di gioco senza intoppi, garantendo fluidità e accessibilità a tutti gli utenti.

In conclusione, i risultati ottenuti da questa fase di validazione, offrono una visione complessivamente positiva dell'esperienza di gioco. I punteggi medi elevati nei criteri di divertimento, dominanza e pensiero creativo indicano un coinvolgimento significativo degli utenti e una buona capacità del gioco di intrattenere e stimolare il pensiero critico. Considerando che questa fase di valutazione si è concentrata principalmente sull'esperienza utente, i risultati ottenuti suggeriscono una soddisfacente riuscita del gioco nel raggiungere i suoi obiettivi principali.

4.2 Limitazioni della ricerca

Una delle principali limitazioni di questa ricerca riguarda la dimensione del campione coinvolto nella fase di validazione. Il numero limitato di partecipanti potrebbe influenzare la generalizzabilità dei risultati ottenuti , inoltre il gioco potrebbe contenere alcuni bug non identificati a causa del limitato

campione di utenti che lo ha testato. Un'altra importante limitazione riguarda il tipo di validazione , che si è concentrata esclusivamente sull'esperienza utente, senza considerare l'impatto educativo del gioco. Questo approccio potrebbe non cogliere appieno l'efficacia del gioco nel trasmettere i concetti educativi previsti.

Infine, va considerato che il progetto sviluppato costituisce solamente un prototipo del gioco finale. Di conseguenza, la sua durata potrebbe risultare limitata e alcuni giocatori potrebbero riscontrare difficoltà nell'assorbire tutti i concetti presentati nel breve periodo di tempo disponibile. La sequenza di presentazione dei concetti potrebbe essere troppo veloce, non consentendo ai giocatori di acquisire una piena comprensione e familiarità con le nuove nozioni.

4.3 Sviluppi futuri

Nonostante il raggiungimento degli obiettivi prefissati, il presente lavoro lascia spazio a possibili sviluppi futuri che potrebbero arricchire ulteriormente il gioco e migliorarne l'efficacia educativa.

Una possibile direzione per lo sviluppo futuro potrebbe riguardare l'implementazione di una valutazione più approfondita dell'impatto educativo del gioco. Questo potrebbe essere realizzato attraverso uno studio più ampio e mirato, coinvolgendo un campione più rappresentativo e valutando anche indicatori di apprendimento specifici, come il miglioramento delle competenze in programmazione a oggetti dopo aver giocato.

Sarebbe possibile ampliare il gioco con nuovi livelli (ad esempio implementare il livello 5), enigmi e meccaniche di gioco che approfondiscano ulteriormente i concetti introdotti. Una maggiore varietà di situazioni e sfide potrebbe rendere l'esperienza di gioco più coinvolgente e stimolante per i giocatori, consentendo loro di esplorare in modo più approfondito i concetti di programmazione a oggetti.

Inoltre ,durante il processo di valutazione del gioco, sono emerse diverse criticità che hanno influenzato l'esperienza di gioco degli utenti. Di seguito vengono elencati i principali problemi riscontrati e le relative raccomandazioni per il miglioramento:

- Mancanza della Possibilità di Saltare: Alcuni utenti hanno segnalato la mancanza della possibilità di saltare all'interno del gioco, ritenendo che l'aggiunta di questa funzionalità potrebbe aumentare il livello di divertimento complessivo.

- Mancanza di Informazioni sui Comandi di Gioco: Alcuni utenti hanno evidenziato la difficoltà nel comprendere i comandi di gioco, specialmente per azioni come la corsa. Si consiglia di aggiungere una schermata dedicata che illustri chiaramente i comandi di gioco, migliorando così l'accessibilità e la facilità di utilizzo.

- Tutorial dei Minigiochi confusionari: I tutorial dei minigiochi vengono attualmente presentati prima dell'inizio del minigioco stesso, creando confusione e sovraccaricando gli utenti con troppe informazioni da assimilare contemporaneamente. Si suggerisce di rivedere la modalità di presentazione dei tutorial in modo che siano accessibili in qualsiasi momento durante il minigioco e che possano essere consultati in caso di necessità.

- Complessità del Minigioco di Creazione degli Oggetti: Alcuni utenti hanno riscontrato difficoltà nel manipolare gli oggetti durante il minigioco a causa della necessità di deselezionare un oggetto prima di poter selezionarne un altro. Si consiglia di semplificare questa interazione consentendo agli utenti di selezionare direttamente un nuovo oggetto senza dover deselezionare quello corrente.

- Complessità delle Interfacce degli Oggetti: L'interazione con le interfacce degli oggetti è risultata poco intuitiva, richiedendo agli utenti di selezionare prima un metodo e poi cliccare su un pulsante per invocarlo. Si propone di semplificare questo processo consentendo agli utenti di lanciare direttamente l'invocazione del metodo semplicemente selezionandolo.

- Durata Eccessiva delle Animazioni di Cambio Camera: Le animazioni di cambio camera sono state percepite come troppo lunghe, specialmente quando la distanza dalla camera principale è breve. Si suggerisce di ridurre la durata di queste animazioni per migliorare la fluidità del gameplay e ridurre i tempi di attesa degli utenti.

In conclusione, l'espansione e il miglioramento continuo del gioco offrono numerose opportunità per arricchire l'esperienza di apprendimento . Gli sviluppi futuri potrebbero contribuire a migliorare le valutazione sul divertimento , immersività e attivazione , inoltre aiuterebbero a consolidare il ruolo del gioco come strumento efficace per l'apprendimento della programmazione a oggetti e promuovere una maggiore diffusione di competenze digitali tra i giovani.

Bibliografia

[1] 3d.rina. *Sci-fi GUI skin*. [Online; in data 11-marzo-2024]. 2017. URL: https://assetstore.unity.com/packages/2d/gui/sci-fi-gui-skin-15606.

[2] Sajana A., Kamal Bijlani e R. Jayakrishnan. "An interactive serious game via visualization of real life scenarios to learn programming concepts". In: *2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2015, pp. 1–8. DOI: 10.1109/ICCCNT.2015.7395173.

[3] Suhni Abbasi e Hameedullah Kazi. "Stealth assessment in serious games to improve OO learning outcomes". In: *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*. 2020, pp. 1–5. DOI: 10.1109/AECT47998.2020.9194170.

[4] C.C. Abt. *Serious Games*. University Press of America, 1987. ISBN: 9780819161482. URL: https://books.google.it/books?id=axUs9HA-hF8C.

[5] Adobe. *Mixamo*. [Online; in data 16-marzo-2024]. URL: https://www.mixamo.com/#/.

[6] Alice H. Aubert, René Bauer e Judit Lienert. "A review of water-related serious games to specify use in environmental Multi-Criteria Decision Analysis". In: *Environmental Modelling Software* 105 (2018), pp. 64–78. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2018.03.023. URL: https://www.sciencedirect.com/science/article/pii/S1364815217307661.

[7] André F. S. Barbosa et al. "A New Methodology of Design and Development of Serious Games". In: *International Journal of Computer Games Technology* 2014 (2014). ISSN: 1687-7047. DOI: 10.1155/2014/817167. URL: https://doi.org/10.1155/2014/817167.

[8] Gabriele Barone. *Perchè quando gioco non mi accorgo del tempo che passa?* [Online; in data 15-novembre-2023]. 2015. URL: https://www.horizonpsytech.com/2015/05/29/perche-quando-gioco-non-mi-accorgo-del-tempo-che-passa/.

[9] *bensound*. [Online; in data 16-marzo-2024]. URL: https://www.bensound.com/.

[10] Deekshita Bundhoo e Leckraj Nagowah. "Gaming With OOP Learn: A Mobile Serious Game to Learn Object-Oriented Programming". In: *2022 3rd International Conference on Next Generation Computing Applications (NextComp)*. 2022, pp. 1–6. DOI: 10.1109/NextComp55567.2022.9932243.

[11] PULSAR BYTES. *Starfield Skybox*. [Online; in data 11-marzo-2024]. 2017. URL: https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717.

[12] Jack Chang et al. "Experience with Dream Coders: developing a 2D RPG for teaching introductory programming concepts". In: *Journal of Computing Sciences in Colleges* 28 (ott. 2012), pp. 227–236.

[13] CodinGame. *CodinGame*. [Online; in data 17-novembre-2023]. 2023. URL: https://www.codingame.com/training.

[14] Wassila Debabi e Tahar Bensebaa. "Using Serious Game to Enhance Learning and Teaching Algorithmic". In: *Journal of e-Learning and Knowledge Society* 12.2 (mag. 2016). ISSN: 1826-6223. URL: https://www.learntechlib.org/p/173469.

[15] Deckweed. *80 Sci-Fi Irregular Frame In One*. [Online; in data 11-marzo-2024]. 2023. URL: https://assetstore.unity.com/packages/2d/gui/80-sci-fi-irregular-frame-in-one-248272.

[16] Ronin Design. *Origin Tech*. [Online; in data 11-marzo-2024]. 2021. URL: https://www.dafont.com/it/origin-tech.font.

[17] Ralf Dörner et al. "Introduction". In: *Serious Games: Foundations, Concepts and Practice*. A cura di Ralf Dörner et al. Cham: Springer International Publishing, 2016, pp. 1–34. ISBN: 978-3-319-40612-1. DOI: 10.1007/978-3-319-40612-1_1. URL: https://doi.org/10.1007/978-3-319-40612-1_1.

[18] Allison Druin. *Mobile technology for children : designing for interaction and learning*. Elsevier, 2009. ISBN: 9780080954097. URL: https://search.worldcat.org/it/title/460107843.

[19] René Eppmann, Magdalena Bekk e Kristina Klein. "Gameful Experience in Gamification: Construction and Validation of a Gameful Experience Scale [GAMEX]". In: *Journal of Interactive Marketing* 43 (2018), pp. 98–115. ISSN: 1094-9968. DOI: https://doi.org/10.1016/j.intmar.2018.03.002. URL: https://www.sciencedirect.com/science/article/pii/S1094996818300124.

[20] FinanceCommunity. *Gaming, il settore vale 336 miliardi di dollari e salirà a 522 miliardi entro il 2027*. [Online; in data 15-novembre-2023]. 2023. URL: https://financecommunity.it/gaming-il-settore-vale-336-miliardi-di-dollari-e-salira-a-522-miliardi-entro-il-2027/.

[21] *flaticon*. [Online; in data 16-marzo-2024]. URL: https://www.flaticon.com/.

[22] Mattia Fortunati e Antonella Galizia. "Pollution Runner: a Serious Game to Promote Awareness Towards Air Pollution". In: *2023 IEEE 19th International Conference on e-Science (e-Science)*. 2023, pp. 1–8. DOI: 10.1109/e-Science58273.2023.10254805.

[23] De Grove Frederik, Mechant Peter e Van Looy Jan. "Uncharted Waters? Exploring Experts' Opinions on the Opportunities and Limitations of Serious Games for Foreign Language Learning". In: *Proceedings of the 3rd International Conference on Fun and Games*. Fun and Games '10. Leuven, Belgium: Association for Computing Machinery, 2010, pp. 107–115. ISBN: 9781605589077. DOI: 10.1145/1823818.1823830. URL: https://doi.org/10.1145/1823818.1823830.

[24] *freesound*. [Online; in data 16-marzo-2024]. URL: https://freesound.org/.

[25] ADRIANA GALGANO. *LA GAMIFICATION NELLE ATTIVITÀ FORMATIVE*. [Online; in data 28-febbraio-2024]. 2019. URL: https://www.hbritalia.it/mondo-formazione/2019/03/15/news/la-gamification-nelle-attivita-formative-3683/.

[26] Stefanos Galgouranas e Stelios Xinogalos. "jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming With Java". In: *Simulation Gaming* 49 (lug. 2018), p. 104687811878997. DOI: 10.1177/1046878118789976.

[27] Sickhead Games. *Sci-Fi Construction Kit (Modular)*. [Online; in data 11-marzo-2024]. 2020. URL: https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-construction-kit-modular-159280.

[28] Gamindo. *Il flow nei videogiochi*. [Online; in data 15-novembre-2023]. 2023. URL: https://blog.gamindo.com/stato-di-flow-nei-videogiochi/.

[29] InsomniacGames. *Marvel's Spider-Man*. [Online; in data 16-marzo-2024]. 2018. URL: https://insomniac.games/game/spider-man-ps4/.

[30] Tobias Jordine, Ying Liang e Edmund Ihler. "A mobile-device based serious gaming approach for teaching and learning Java programming". In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. 2014, pp. 1–5. DOI: 10.1109/FIE.2014.7044206.

[31] Cagin Kazimoglu et al. "A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming". In: *Procedia - Social and Behavioral Sciences* 47 (2012). Cyprus International Conference on Educational Research (CY-ICER-2012)North Cyprus, US08-10 February, 2012, pp. 1991–1999. ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2012.06.938. URL: https://www.sciencedirect.com/science/article/pii/S1877042812026742.

[32] Linden Lab. *SecondLife*. [Online; in data 28-febbraio-2024]. URL: https://secondlife.com/.

[33] Elaachak Lotfi e Bouhorma Mohammed. "Teaching Object Oriented Programming Concepts Through a Mobile Serious Game". In: *Proceedings of the 3rd International Conference on Smart City Applications*. SCA '18. Tetouan, Morocco: Association for Computing Machinery, 2018. ISBN: 9781450365628. DOI: 10.1145/3286606.3286851. URL: https://doi.org/10.1145/3286606.3286851.

[34] Carrie McLeroy. *History of Military gaming*. [Online; in data 28-febbraio-2024]. 2008. URL: https://www.army.mil/article/11936/history_of_military_gaming.

[35] Tamotsu Mitamura, Yasuhiro Suzuki e Takahumi Oohori. "Serious games for learning programming languages". In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2012, pp. 1812–1817. DOI: 10.1109/ICSMC.2012.6378001.

[36] Konstantin Mitgutsch e Narda Alvarado. “Purposeful by Design? A Serious Game Design Assessment Framework”. In: *Proceedings of the International Conference on the Foundations of Digital Games*. FDG ’12. Raleigh, North Carolina: Association for Computing Machinery, 2012, pp. 121–128. ISBN: 9781450313339. DOI: 10.1145/2282338.2282364. URL: https://doi.org/10.1145/2282338.2282364.

[37] Jean Moreno. *Cartoon FX Remaster Free*. [Online; in data 11-marzo-2024]. 2023. URL: https://assetstore.unity.com/packages/vfx/particles/cartoon-fx-remaster-free-109565.

[38] Chris Nolet. *Quick Outline*. [Online; in data 11-marzo-2024]. 2022. URL: https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488.

[39] Francisco R. Ortega et al. “Towards a 3D Virtual Programming Language to increase the number of women in computer science education”. In: *2017 IEEE Virtual Reality Workshop on K-12 Embodied Learning through Virtual Augmented Reality (KELVAR)*. 2017, pp. 1–6. DOI: 10.1109/KELVAR.2017.7961558.

[40] Ioannis Paliokas, Christos Arapidis e Michail Mpimpitsos. “PlayLOGO 3D: A 3D Interactive Video Game for Early Programming Education: Let LOGO Be a Game”. In: *2011 Third International Conference on Games and Virtual Worlds for Serious Applications*. 2011, pp. 24–31. DOI: 10.1109/VS-GAMES.2011.10.

[41] B. Sawyer. “The "Serious Games" Landscape.” In: Presented at the Instructional Research Technology Symposium for Arts, Humanities e Social Sciences, Camden, USA, 2007.

[42] Rafael J. Segura et al. “VR-OCKS: A virtual reality game for learning the basic concepts of programming”. In: *Computer Applications in Engineering Education* 28.1 (2020), pp. 31–41. DOI: https://doi.org/10.1002/cae.22172. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.22172. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22172.

[43] Wong Yoke Seng e Maizatul Hayati Mohamad Yatim. “Computer Game as Learning and Teaching Tool for Object Oriented Programming in Higher Education Institution”. In: *Procedia - Social and Behavioral Sciences* 123 (2014). TAYLOR’S 6TH TEACHING AND LEARNING CONFERENCE 2013: TRANSFORMATIVE HIGHER EDUCATION TEACHING AND LEARNING IN PRACTICE PROCEEDINGS OF

THE TAYLOR'S 6TH TEACHING AND LEARNING CONFERENCE 2013 (TTLC2013), November 23, 2013, Taylor's University Lakeside Campus, Selangor Daruh Ehsan, Malaysia, pp. 215–224. ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2014.01.1417. URL: https://www.sciencedirect.com/science/article/pii/S1877042814014554.

[44] Ángel Serrano-Laguna et al. "Building a Scalable Game Engine to Teach Computer Science Languages". In: *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje* 10.4 (2015), pp. 253–261. DOI: 10.1109/RITA.2015.2486386.

[45] Grigorios Sideris e Stelios Xinogalos. "PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming With Python". In: *Simulation & Gaming* 50.6 (2019), pp. 754–770. DOI: 10.1177/1046878119872797. eprint: https://doi.org/10.1177/1046878119872797. URL: https://doi.org/10.1177/1046878119872797.

[46] Nishal Singh e Leckraj Nagowah. "OOP Codes: Teaching Object-Oriented Programming Concepts Through a Mobile Serious Game". In: *2021 25th International Computer Science and Engineering Conference (ICSEC)*. 2021, pp. 377–382. DOI: 10.1109/ICSEC53205.2021.9684593.

[47] SketchFab. *SketchFab*. [Online; in data 16-marzo-2024]. URL: https://sketchfab.com/feed.

[48] *STARBUCKS® REWARDS*. [Online; in data 19-marzo-2024]. URL: https://www.starbucks.com/rewards.

[49] stelios-xinogalos. *stelios-xinogalos/serious-games/programming*. [Online; in data 18-novembre-2023]. 2023. URL: https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-games/programming#TOC-Py-rate-Adventures:-a-2D-platform-game-for-learning-programming-in-Python.

[50] Unity Technologies. *StarterAssets - FirstPerson*. [Online; in data 11-marzo-2024]. 2023. URL: https://assetstore.unity.com/packages/essentials/starterassets-firstperson-updates-in-new-charactercontroller-pac-196525.

[51] Unity Technologies. *Unity 3D*. [Online; in data 16-marzo-2024]. URL: https://unity.com/.

[52] Sevasti Theodosiou e Ilias Karasavvidis. "Serious games design: A mapping of the problems novice game designers experience in designing games". In: *Journal of e-Learning and Knowledge Society* 11.3 (set. 2015). ISSN: 1826-6223. URL: https://www.learntechlib.org/p/151929.

[53] Enciclopedia Treccani. *Gamification*. [Online; in data 19-marzo-2024]. URL: https://www.treccani.it/vocabolario/gamification_%28Neologismi%29/.

[54] Enciclopedia Treccani. *Videogioco*. [Online; in data 28-febbraio-2024]. URL: https://www.treccani.it/enciclopedia/videogioco/.

[55] Dimitrios TSIOTRAS e Stelios Xinogalos. "Investigating the Perceived Player Experience and Short-term Learning of the Text-based Java Programming Serious Game "Rise of the Java Emperor"". In: *Informatics in Education* 20 (mar. 2021), pp. 153–170. DOI: 10.15388/infedu.2021.08.

[56] Vectr. *Vectr*. [Online; in data 16-marzo-2024]. URL: https://vectr.com/.

[57] Yoke Seng Wong et al. "A propriety game based learning mobile game to learn object-oriented programming — Odyssey of Phoenix". In: *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. 2017, pp. 426–431. DOI: 10.1109/TALE.2017.8252373.