

MagicPortal

Titolo del progetto: MagicPortal
Alunno/a: Lorenzo Berther, Sebastiano DeBertoldi, Simone De Marchi, Kamil Siddiqui
Classe: Info I3A
Anno scolastico: 2024/2025
Docente responsabile: Michel Palucci

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
2	Analisi	4
2.1	Analisi del dominio	4
2.2	Analisi e specifica dei requisiti	4
2.2.1	Spiegazione elementi tabella dei requisiti:	7
2.3	Use case	7
2.4	Pianificazione	7
2.5	Analisi dei mezzi	8
2.5.1	Software	8
2.5.2	Hardware	8
3	Progettazione	8
3.1	Design dell'architettura del sistema	8
3.2	Design dei dati e database	9
3.3	Design delle interfacce	10
3.3.1	Design Login	10
3.3.2	Design Register	11
3.4	Design procedurale	11
4	Implementazione	14
5	Test	14
5.1	Protocollo di test	14
5.2	Risultati test	15
5.3	Mancanze/limitazioni conosciute	15
6	Consuntivo	15
7	Conclusioni	15
7.1	Sviluppi futuri	15
7.2	Considerazioni personali	15
8	Glossario	15
9	Indice delle figure	15
10	Bibliografia	16
10.1	Bibliografia per articoli di riviste:	16
10.2	Bibliografia per libri	16
10.3	Sitografia	16
11	Allegati	16

1 Introduzione

1.1 Informazioni sul progetto

In questo capitolo raccogliere le informazioni relative al progetto, ad esempio:

- Allievo/i e docente/i coinvolti nel progetto e rispettivi ruoli,
- scuola, sezione, materia/e,
- date di inizio e termine di consegna,
- ...

1.2 Abstract

È una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all'utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

- **Background/Situazione iniziale**
- **Descrizione del problema e motivazione:** Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
- **Approccio/Metodi:** Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche...)? Quale è stata l'entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
- **Risultati:** Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

As the size and complexity of today's most modern computer chips increase, new techniques must be developed to effectively design and create Very Large-Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.

1.3 Scopo

Lo scopo del progetto (scopi didattici/scopi operativi). Dovrebbe descrivere il mandato, ma non vanno ricopiate le informazioni del quaderno dei compiti (che va invece allegato).

2 Analisi

2.1 Analisi del dominio

Il videogioco è destinato a qualunque fascia d'età, richiede solamente di registrarsi creando un account. Sarà giocabile da PC con tastiera o controller. L'obiettivo principale è quello di creare un puzzle-game in Unity...

2.2 Analisi e specifica dei requisiti

ID: REQ-001	
Nome	Gestione utente
Priorità	1
Versione	1.0
Sotto requisiti	
001	Creare Database 'MagicPortal'
002	Creare le tabelle 'User' e 'Friend'

ID: REQ-002	
Nome	Effettuare un login
Priorità	1
Versione	1.0
Note	Dipende dal requisito REQ-001 (Gestione Utente)
Sotto requisiti	
001	Dovrà essere possibile eliminare gli utenti
002	Deve spuntare una tastiera virtuale (per controller)

ID: REQ-003	
Nome	Effettuare una registrazione
Priorità ID: REQ-003	1
Versione	1.0
Note	Dipende dal requisito REQ-001 (Gestione Utente)
Sotto requisiti	
001	Controlli per non creare utenti identici a quelli già esistenti
002	Deve spuntare una tastiera virtuale (per controller)

ID: REQ-004	
Nome	Leaderboard web
Priorità	1
Versione	1.0
Note	Dipende dal requisito REQ-001 (Creazione DB)
Sotto requisiti	
001	Creare la tabella 'Leaderboard'
002	Creare una pagina web che prenda gli highscore degli utenti
003	Nel gioco deve esserci un collegamento alla pagina

ID: REQ-005	
Nome	Enigmi
Priorità	1
Versione	1.0
Note	Creare gli enigmi e rendere possibile inserirli nelle stanze. Vedere meglio i vari enigmi qui: Design dell'architettura del sistema

ID: REQ-006	
Nome	Mappa isometrica casuale
Priorità	1
Versione	1.0
Sotto requisiti	
001	Selezionare le stanze pre-create casualmente, inserendo all'interno di esse un puzzle tra quelli pre-creati
002	Creare id per il dungeon
003	Generare casualmente gli NPC che proteggono gli oggetti

ID: REQ-007	
Nome	Salvataggio partite
Priorità	1
Versione	1.0

ID: REQ-008	
-------------	--

Nome	Tutorial testuale
Priorità	1
Versione	1.0
Note	Creare una pagina con la descrizione dei comandi

ID: REQ-008	
Nome	Aggiungere amici
Priorità	1
Versione	1.0
Note	Dipende dal requisito REQ-001 (Creazione DB)

ID: REQ-009	
Nome	Giocare con tastiera e controller
Priorità	1
Versione	1.0
Note	In caso di controller, generare una tastiera virtuale, per soddisfare requisito REQ-002

ID: REQ-010	
Nome	Personalizzazione skin
Priorità	2
Versione	1.0
Sotto requisiti	
001	Creare uno shop in cui acquistare le skin
002	A fine del dungeon dare delle monete

ID: REQ-012	
Nome	Tutorial interattivo
Priorità	3
Versione	1.0

2.2.1 Spiegazione elementi tabella dei requisiti:

ID: identificativo univoco del requisito

Nome: breve descrizione del requisito

Priorità: indica l'importanza di un requisito nell'insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

Versione: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l'ultima versione, mentre le vecchie dovranno essere inserite nei diari.

Note: eventuali osservazioni importanti o riferimenti ad altri requisiti.

Sotto requisiti: elementi che compongono il requisito.

2.3 Use case

I casi d'uso rappresentano l'interazione tra i vari attori e le funzionalità del prodotto.

2.4 Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

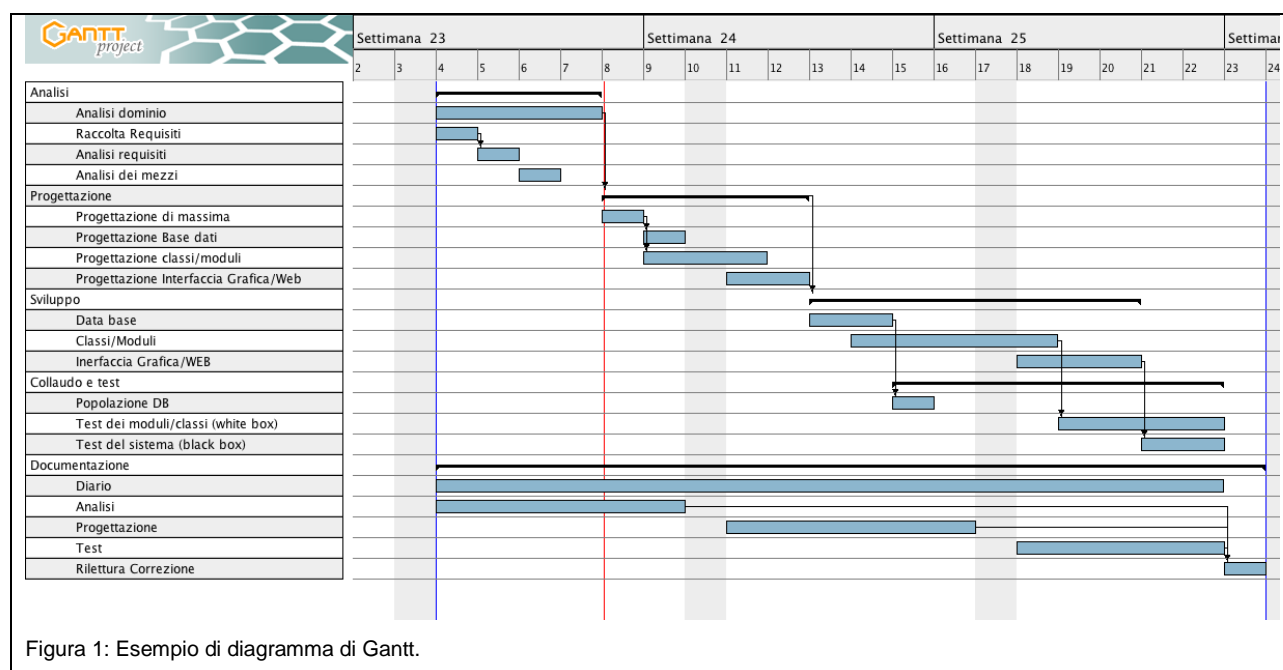


Figura 1: Esempio di diagramma di Gantt.

Se si usano altri metodi di pianificazione (p.es. scrum), dovranno apparire in questo capitolo.

2.5 Analisi dei mezzi

2.5.1 Software

- Unity 6
- MS-Office
- MS-Project
- GitHub (Desktop)
- PHP Storm
- MySQL Workbench 8.0
- Bootstarp 5.2.3
- Figma

2.5.2 Hardware

PC forniti dalla scuola e Server scolastico dove mettere il progetto e il Database

3 Progettazione

Questo capitolo descrive esaurientemente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all'esecutore di evitare fraintendimenti e imprecisioni nell'implementazione del prodotto.

3.1 Design dell'architettura del sistema

3.2 Design dei dati e database

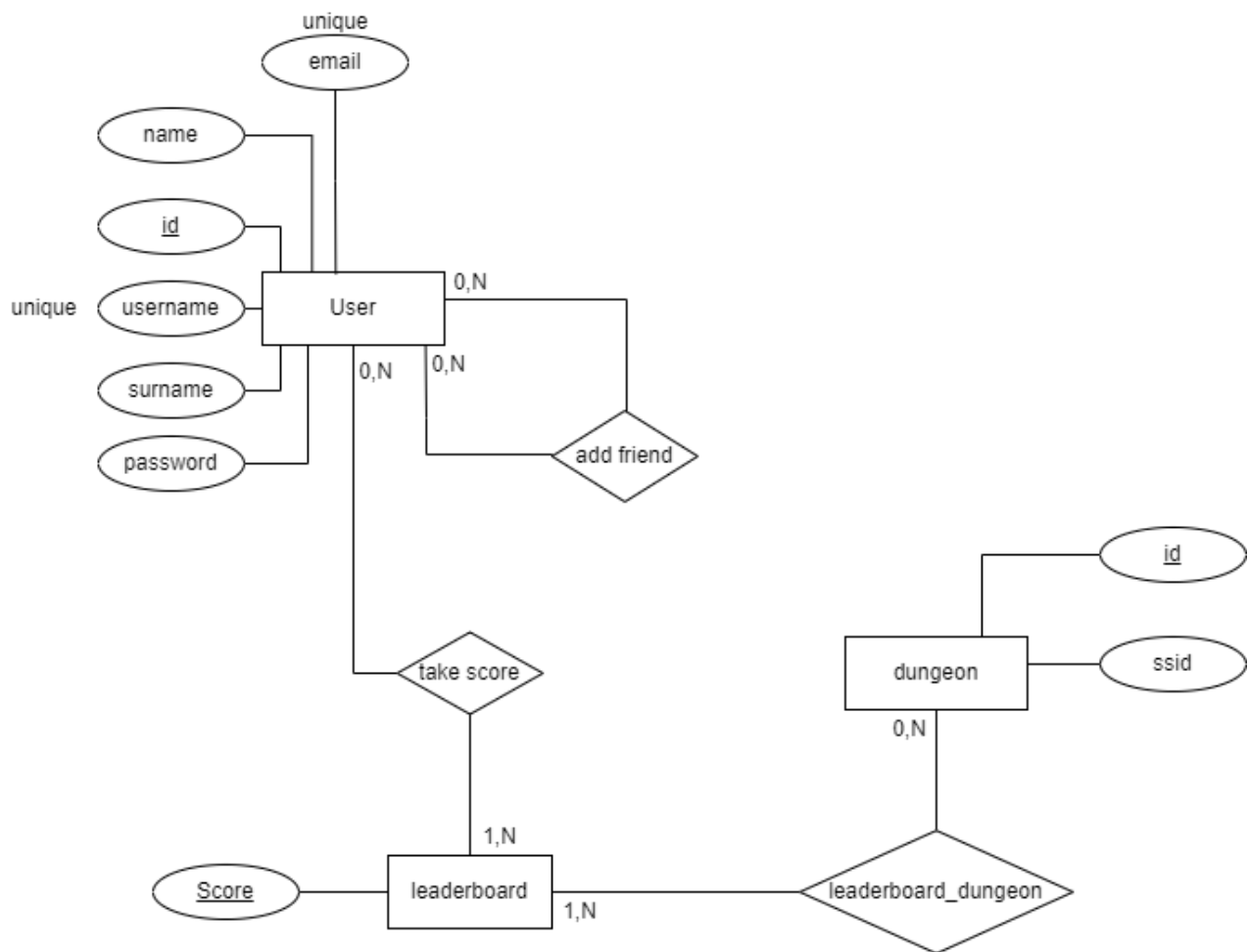


Figura 2: Schema E-R

User	
PK	<u>id int NOT NULL AUTO INCREMENT</u>
	username varchar(64) UNIQUE NOT NULL
	password varchar(256) NOT NULL
	email varchar(256) NOT NULL

Leaderboard	
PK	<u>score int NOT NULL</u>
FK	user_id int
FK	dungeon_id int

Friend	
PK	<u>player_id int NOT NULL</u>
PK	<u>friend_id int NOT NULL</u>

Dungeon	
PK	<u>id int NOT NULL AUTO INCREMENT</u>
	ssid int

Figura 3: Schema logico

3.3 Design delle interfacce

3.3.1 Design Login

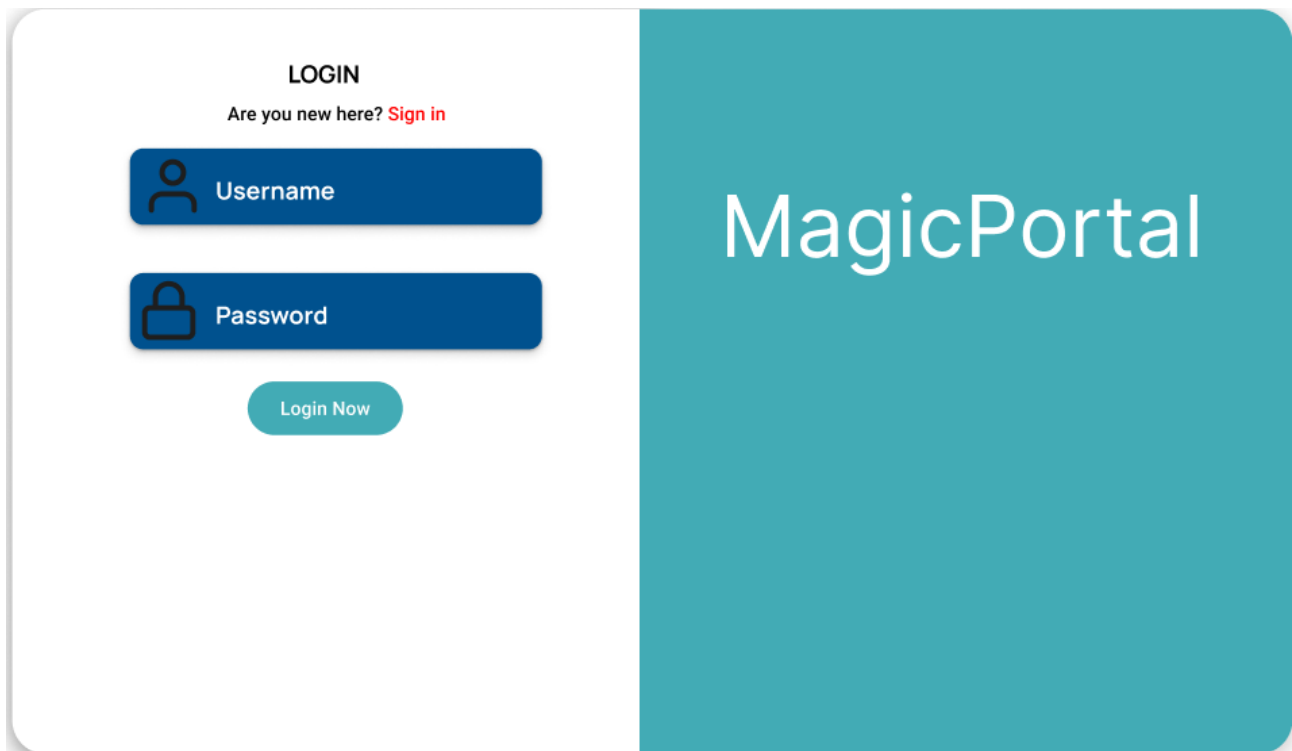


Figura 4: Design interfaccia di Login

3.3.2 Design Register

SIGN IN

Username

Email

Password

Conferma Password

Register Now

MagicPortal

Figura 5: Design interfaccia di Registrazione

3.3.2.1 Design della mappa

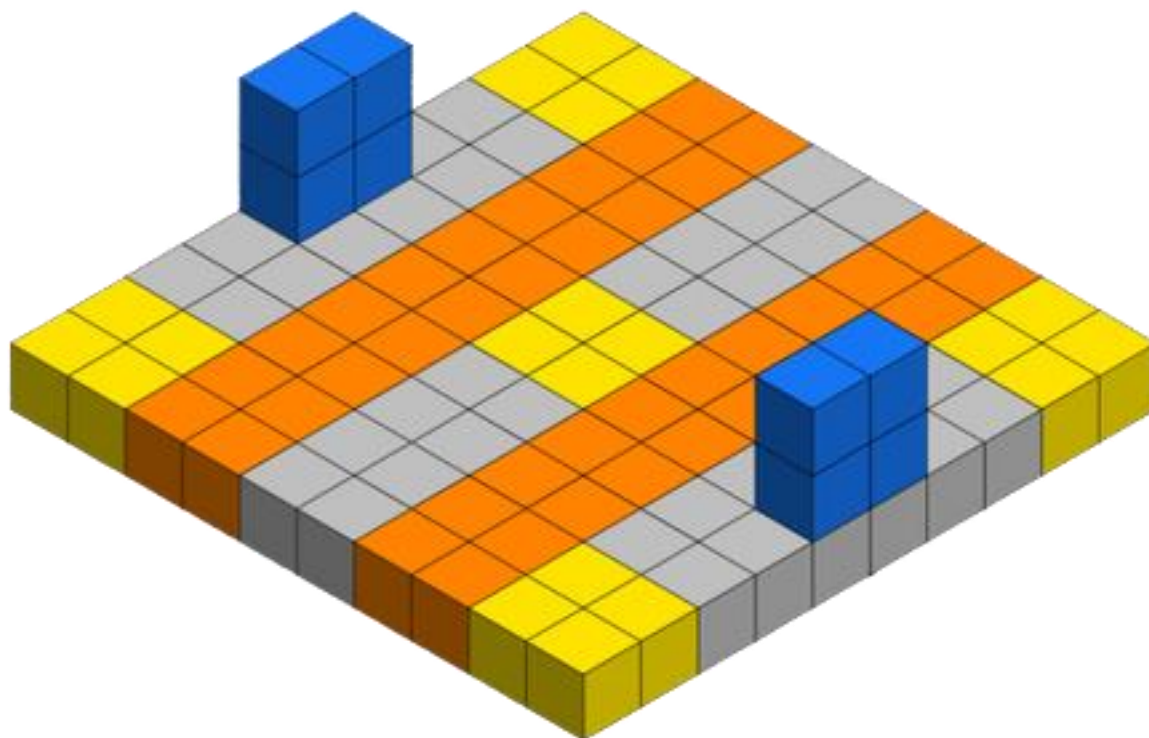


Figura 6: Design di una stanza

3.4 Design degli Enigmi e ostacoli

3.4.1 Percorso con pedane finte

In questo percorso il giocatore dovrà percorrere un percorso lungo 4 pedane, dove alcune pedane saranno finte. Se l'utente ci passerà sopra, verrà riportato all'inizio della stanza, perdendo eventuali oggetti che possedeva, come la chiave per la porta d'uscita, e la pedana sparirà, lasciando un buco.

Questo percorso occuperebbe uno spazio di colore rosso sulla mappa.

Ecco un esempio di come un'ipotetica pedana potrebbe essere e come potrebbe venir generata:

Le pedane rossi sono i blocchi da evitare, per generarli si potrebbe gestire il percorso come un array bidimensionale, prendendo la pedana in alto a sinistra come posizione [0,0] e la pedana in basso a destra come posizione [h,4], generando poi casualmente la posizione di quali pedane sono finte, prestando sempre attenzione di lasciare una via disponibile.

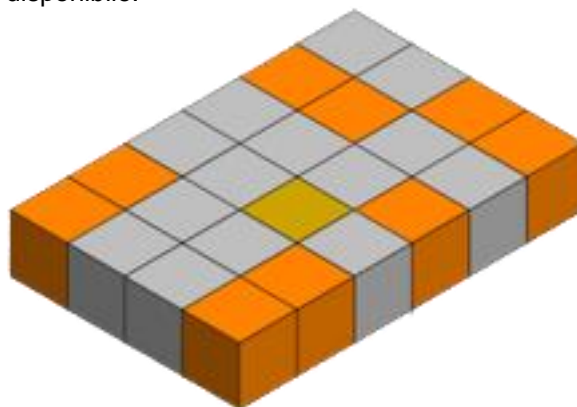


Figura 7: Design del percorso con pedane finte

3.4.2 Percorso con pedane che attivano trappole

3.4.3 Percorso con pedana invisibile e soluzione sul muro

Questo percorso è simile all'enigma "Percorso con pedane finte", ma a sua differenza le pedane sono invisibili e il percorso da percorrere è disegnato sul muro, anche qua se si sbaglia e si passa su una pedana inesistente si verrà portati all'inizio della stanza, perdendo ogni oggetto posseduto.

Questo percorso occuperebbe uno spazio di colore rosso sulla mappa.

Ecco un esempio di come un'ipotetica pedana potrebbe essere e come potrebbe venir generata:

La parte rossa è bianca sono invisibili/trasparenti. Sul muro grigio vediamo la strada corretta da percorrere (bianca).

Per crearla potremmo usare un metodo simile a quello de "Percorso con pedana finte", quindi con un array bidimensionale, e lo useremmo anche per disegnare sul muro

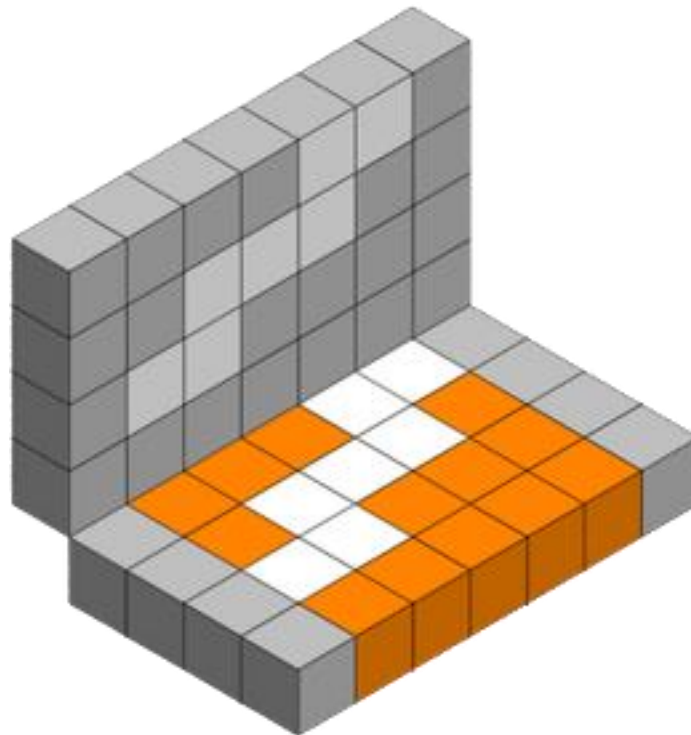


Figura 8: Design dell'enigma "Percorso con pedana invisibile e soluzione sul muro"

3.4.4 Cannoni

3.4.5 Asce oscillanti

3.5 Design procedurale

Descrive i concetti dettagliati dell'architettura/sviluppo utilizzando ad esempio:

- Diagrammi di flusso e Nassi.
- Tabelle.
- Classi e metodi.
- Tabelle di routing
- Diritti di accesso a condivisioni ...

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

4 Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l'inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d'uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

5 Test

5.1 Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l'adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

Test Case:	TC-001	Nome:	Import a card with KIC, KID and KIK keys, but not shown with the GUI
Riferimento:	REQ-012		
Descrizione:	Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI		
Prerequisiti:	Store on local PC: Profile_1.2.001.xml (appendix n.n) and Cards_1.2.001.txt (appendix n.n). PIN (OTA_VIEW_PIN_PUK_KEY) and ADM (OTA_VIEW_ADM_KEY) user right not set.		
Procedura:	<ol style="list-style-type: none"> Go to "Cards manager" menu, in main page click "Import Profiles" link, Select the "1.2.001.xml" file, Import the Profile Go to "Cards manager" menu, in main page click "Import Cards" link, Select the "1.2.001.txt" file, Delete the cards, Select the "1.2.001.txt" file, Import the cards Research the "41795924770" Card, Click the imsi card link Check the card details Execute the SQL: <pre>SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset;</pre> 		
Risultati attesi:	Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details)		

5.2 Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l'errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l'errore con eventuali ipotesi di correzione.

5.3 Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

6 Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

7 Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un'aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

7.1 Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

7.2 Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

8 Glossario

Termine	Descrizione
Godot Engine	Engine per sviluppare videogiochi 2D/3D in GDscript

9 Indice delle figure

Figura 1: Esempio di diagramma di Gantt	7
Figura 2: Schema E-R	9
Figura 3: Schema logico	9
Figura 4: Design interfaccia di Login	10
Figura 5: Design interfaccia di Registrazione	11
Figura 6: Design di una stanza	11
Figura 7: Design del percorso con pedane finte	12
Figura 8: Design dell'enigma "Percorso con pedana invisibile e soluzione sul muro"	13

10 Bibliografia

10.1 Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell'autore o degli autori, o nome dell'organizzazione,
2. Titolo dell'articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell'articolo.

10.2 Bibliografia per libri

1. Cognome e nome (o iniziali) dell'autore o degli autori, o nome dell'organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell'editore,
5. Anno di pubblicazione,
6. ISBN.

10.3 Sitografia

- URL del sito (se troppo lungo solo dominio, evt completo nel diario),
- Eventuale titolo della pagina (in italico),
- Data di consultazione (GG-MM-AAAA)
- <https://www.youtube.com/watch?v=dcl8w6JW7Y>, *Tutorial Godot 'ThinkWithGames'*, 15-01-2025
- <https://www.youtube.com/watch?v=8HvcHtauKoc>, *Tutorial Collision 'ThinkWithGames'*, 15-01-2025

Esempio:

- <http://standards.ieee.org/guides/style/section7.html>, *IEEE Standards Style Manual*, 07-06-2008.

11 Allegati

Elenco degli allegati, esempio:

- Diari di lavoro
- Codici sorgente/documentazione macchine virtuali
- Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
- Documentazione di prodotti di terzi
- Eventuali guide utente / Manuali di utilizzo
- Mandato e/o QdC
- Prodotto
- ...