Castello di Trevano VR

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

2.3 Use case 6

2.4 Pianificazione 6

2.5 Analisi dei mezzi 6

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 7

3.1 Design dell’architettura del sistema 7

3.2 Design dei dati e database 7

3.3 Design delle interfacce 7

3.4 Design procedurale 7

4 Implementazione 8

5 Test 8

5.1 Protocollo di test 8

5.2 Risultati test 9

5.3 Mancanze/limitazioni conosciute 9

6 Consuntivo 9

7 Conclusioni 9

7.1 Sviluppi futuri 9

7.2 Considerazioni personali 9

8 Bibliografia 9

8.1 Bibliografia per articoli di riviste: 9

8.2 Bibliografia per libri 9

8.3 Sitografia 10

9 Allegati 10

# Introduzione

## Informazioni sul progetto

* Allievo: Alberto Perdomo Carpio
* Docente responsabile: Michel Palucci
* Perito 1: -
* Perito 2: -
* Data inizio: 03.02.2025
* Data di fine: 03.04.2025
* Luogo di sviluppo: Scuola Arti e Mestieri di Trevano
* Sezione scolastica: Informatica 4° anno

## Abstract

Nowadays, unfortunately, information about the history of our country and its numerous historical monuments is increasingly ignored and overlooked. This is truly a shame, considering the importance of preserving and valuing our cultural heritage. For this reason, the idea was conceived to develop an innovative application that not only informs users but also engages them in a fun and interactive way.

In particular, the concept is to create a virtual room, a digital environment containing various objects with which users can interact. Each object will be accompanied by historical information, allowing users to discover interesting facts and details about our heritage. The use of a virtual world was inspired by the goal of making the experience more engaging and captivating, sparking users’ curiosity and encouraging an educational experience that goes beyond simply reading information.

## Scopo

Lo scopo principale di questo progetto è quello di diffondere informazioni riguardanti luoghi storici, partendo dal Castello di Trevano, cercando però di rendere l’attività interessante e coinvolgente. L’obiettivo è incentivare l’utente a giocare, ma allo stesso tempo fornire una panoramica di fatti e informazioni storiche in modo divertente ed educativo.

L’utente si troverà all’interno di una stanza virtuale, pensata per essere interattiva, dove potrà esplorare e interagire con vari elementi, come ad esempio dipinti o oggetti storici. La stanza sarà strutturata in modo da permettere un movimento dinamico, rendendo l’esperienza più immersiva e realistica. Per guidare l’utente, sarà presente un tutorial che spiegherà come muoversi all’interno dello spazio e interagire con gli oggetti e le informazioni contenute nella scena. Questo permetterà agli utenti di vivere un’esperienza fluida e accessibile, pur mantenendo il focus sull’aspetto educativo.

Combinando elementi educativi con un ambiente virtuale coinvolgente, il progetto mira a creare un’esperienza di apprendimento che sembri più un’avventura interattiva che una semplice esplorazione passiva della storia. Grazie a questo approccio immersivo, l’utente avrà l’opportunità non solo di scoprire il Castello di Trevano e la sua storia, ma anche di esplorare altri elementi storici correlati che potrebbero non essere stati conosciuti altrimenti.

# Analisi

## Analisi del dominio

* L'applicazione è stata sviluppata con l'obiettivo di avere una finalità educativa e può essere utilizzata da utenti di tutte le età.
* Attualmente esistono già degli applicativi che ti permettono di apprendere informazioni storiche sotto forma di gioco, ma non ne esistono riguardanti il castello di Trevano.
* Qualsiasi persona può utilizzare il prodotto siccome la soglia d’utenza non è definita e dunque gli utenti possono variare per età e conoscenze sulla tecnologia.
* È anche richiesta una funzionalità che permetta di cambiare le informazioni in maniera dinamica, senza dover ricompilare il programma.
* Si dovrà poter implementare una funzionalità che permetta all’utente di potersi muovere nell’ambiente virtuale stando fermo nel mondo reale.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Interfaccia grafica user-friendly. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’applicativo deve poter essere utilizzato in maniera intuitiva da utenti di qualsiasi età, anche da chi non conosce l’hardware utilizzato. |
| **Sotto requisiti** | |
| **001** | L’applicativo deve possedere un’interfaccia grafica intuitiva e facile da utilizzare |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Gestione degli errori. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | In caso di errore l’applicativo deve informare l’utente a riguardo con una notifica. |
| **Sotto requisiti** | |
| **001** | In caso di errore deve comparire una comunicazione che permetta all’utente di capire che sta facendo qualcosa di sbagliato. |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Informazioni backend. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Le informazioni che mostra l’applicativo devono essere inerite all’interno di un database. |
| **Sotto requisiti** | |
| **001** | Le informazioni devono provenire da un database. |
| **002** | In caso di modifica delle informazioni, l’applicativo si aggiorna di conseguenza senza dover ricompilare il progetto. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Funzione di mobilità dell’utente. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente si deve poter muovere all’interno del mondo virtuale. |
| **Sotto requisiti** | |
| **001** | L’utente si deve poter muovere utilizzando il teletrasporto o i joystick. |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Interazione con oggetti. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter interagire con gli oggetti presenti. |
| **Sotto requisiti** | |
| **001** | L’utente deve poter prendere gli oggetti. |
| **002** | L’utente deve poter lanciare gli oggetti. |
| **003** | L’utente deve poter ricavare informazioni dagli oggetti. |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Video tutorial. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter avere a disposizione un video che gli spieghi come utilizzare l’applicativo. |
| **Sotto requisiti** | |
| **001** | L’utente una volta avviato l’applicativo deve poter visualizzare un video tutorial. |
| **002** | Una volta all’interno del gioco, con un menu deve poter visualizzare un mini video per le funzionalità dell’applicativo. |

**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.

## Use case

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

## 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 : Esempio di diagramma di Gantt. |

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

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che HW sarà disponibile durante lo sviluppo?

# Progettazione

Questo capitolo descrive esaustivamente 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.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

## Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## 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.

# 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.

# Test

## 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:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **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:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. 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 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## 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.

## 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.

# Consuntivo

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

# 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.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Bibliografia

## 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.

## 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.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

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

# 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
* …