Non Euclidian Project

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: 5

2.3 Use case 5

2.4 Pianificazione 5

2.5 Analisi dei mezzi 5

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 6

3.1 Design dell’architettura del sistema 6

3.2 Design dei dati e database 6

3.3 Design delle interfacce 6

3.4 Design procedurale 6

4 Implementazione 7

5 Test 7

5.1 Protocollo di test 7

5.2 Risultati test 8

5.3 Mancanze/limitazioni conosciute 8

6 Consuntivo 8

7 Conclusioni 8

7.1 Sviluppi futuri 8

7.2 Considerazioni personali 8

8 Glossario 8

9 Bibliografia 9

9.1 Bibliografia per articoli di riviste: 9

9.2 Bibliografia per libri 9

9.3 Sitografia 9

10 Allegati 9

# Introduzione

## 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,
* …

## 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.*

## Scopo

Lo scopo del progetto è di creare un gioco che si avvali dell’utilizzo di portali per creare una ambiente non euclideo nel quale lo scopo del giocatore è orientarsi per raccogliere delle statue sparse nella mappa e tornare al punto di partenza

# Analisi

## Analisi del dominio

* Questo gioco è inteso per essere giocato da una fascia di età compresa tra i 3 e i 99 anni
* Il gioco è inteso per girare unicamente sulla piattaforma Windows 10-11

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?
* …

In base alla lista dei requisiti e all’analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell’interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Viewports con telecamera |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano viewport che renderizzino una telecamera |
| **Sotto requisiti** | |
| **001** | Si necessita una maschera di login |
| **002** | Si dovranno poter immettere nuovi allievi |
| **003** | Dovrà essere possibile la ricerca di allievi |

### Portali

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Viewports con telecamera |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano viewport che renderizzino una telecamera |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Telecamera che segue i movimenti del giocatore |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitana che la telecamera segua i movimenti del giocatore per dare l’impressione che non ci sia alcun portale |
| **Sotto requisiti** | |
| **001** | Viewports con telecamera |

### Audio

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Colonna sonora |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve avere una colonna sonora che giri un loop infinito per tutta la durata della partita |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Effetti sonori |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve avere degli effetti sonori per passi e raccolta oggetti |

### Gameplay

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Giocatore funzionante |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve muoversi avanti, indietro, desta, sinistra (WASD) e girare la telecamera, inoltre il movimento deve seguire la direzione della telecamera |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Livello di prova iniziale |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve contenere un livello iniziale che mostri all’utente in modo semplice il funzionamento del gioco |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Menu di pausa |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve avere la possibilità di mettere in pausa il gioco in ogni momento andando nel menu di pausa |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Menu iniziale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il giocatore quando avvia il gioco deve vedere il menu iniziale con la possibilità di iiziare una nuova partita, continuarne una già iniziata o entrare nelle impostazioni |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Raccogliere oggetti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve poter raccogliere degli oggetti contrassegnati da materiali definiti |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Far cadere oggetti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve avere la possibilità di far cadere gli oggetti raccolti |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Algoritmo di collegamenti randomici |
| **Priorità** | 4 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve contenere un algoritmo che mischi i collegamenti tra i portali per disorientare l’utente |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Salvare la partita |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve avere la possibilità di salvare la partita in qualsiasi momento |

|  |  |
| --- | --- |
| **ID: REQ-021** | |
| **Nome** | Gestione gravità |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma deve essere in grado di cambiare la direzione della spinta gravitazionale in modo da far accedere il giocatore a più percorsi |

#### Impostazioni di gioco

|  |  |
| --- | --- |
| **ID: REQ-013** | |
| **Nome** | Risoluzione |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve avere la possibilità di modificare la risoluzione della schermata di gioco |

|  |  |
| --- | --- |
| **ID: REQ-014** | |
| **Nome** | Volume audio ed effetti |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve avere la possibilità di modificare il volume della colonna sonora e degli effetti sonori separatamente |

|  |  |
| --- | --- |
| **ID: REQ-015** | |
| **Nome** | Layout tastiera |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il giocatore deve avere la possibilità di reimpostare i tasti della tastiera |

#### Tutorial

|  |  |
| --- | --- |
| **ID: REQ-016** | |
| **Nome** | Spiegare obbiettivo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve spiegare l’obbiettivo tramite una finestra di dialogo e a voce |

|  |  |
| --- | --- |
| **ID: REQ-017** | |
| **Nome** | Spiegare come si gioca |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve spiegare al giocatore come muoversi e girare la telecamera tramite una finestra di dialogo e a voce |

### Modelli 3D

|  |  |
| --- | --- |
| **ID: REQ-018** | |
| **Nome** | Set di statue |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve contenere dei modelli artistici che saranno gli obbiettivi da raccogliere |

|  |  |
| --- | --- |
| **ID: REQ-019** | |
| **Nome** | Set di stanze |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve contenere un set di stanze che saranno poi usate per creare la struttura |

|  |  |
| --- | --- |
| **ID: REQ-020** | |
| **Nome** | Collegamenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve contenere un set di collegamenti che saranno usati per collegare le stanze |

### 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

## Pianificazione

|  |
| --- |
| Figura 1: Esempio di diagramma di Gantt. |

Per la pianificazione è stato scelto lo schema Agile scrumban

### Milestones

|  |  |  |
| --- | --- | --- |
| Nome | Scadenza | Descrizione |
| Portale Funzionante | 16.02.2024 | Si necessita per la scadenza un prototipo di portale funzionante |
| Strutture terminate | 16.02.2024 | Per la scadenza si necessita un prototipo di mappa generato a mano privo di portali |
| Portale pronto per il gioco | 22.03.2024 | Si necessita per la scadenza di un portale funzionante adatto alla mappa |

## Analisi dei mezzi

### Software

Godot V4.\*: Game engine free e open source per lo sviluppo di giochi in 2D e 3D

GDscript: Linguaggio basato su python per lo sviluppo di giochi in godot

### Hardware

Hardware per sviluppo: 3 PC desktop HP elite desk Windows 10

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.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# 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
* …