Esempio di documentazione

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

Nome allievo:

* + Luca Crivelli

Docente:

* + Ingrid Paola Anna Cereda

Data del progetto

* + 12.09.2025-19.12.2025

Scuola

* + SAM Trevano informatica 3AC

## Scopo

Lo scopo del progetto è quello di sviluppare una demo di un gioco horror 2D, e a questo scopo, imparare a usare il motore di gioco 3D jMonkeyEngine basato principalmente su java ma in 2D e quindi approfondire ulteriormente il linguaggio di programmazione java e imparare cose nuove.

# Analisi

## Analisi del dominio

Il progetto nasce con l’obiettivo di sviluppare una demo di gioco horror 2D utilizzando **jMonkeyEngine**, così da approfondire l’uso del motore e migliorare le competenze in Java. Attualmente, la maggior parte dei giochi horror 2D viene realizzata con motori come Unity o GameMaker, mentre l’impiego di jMonkeyEngine in questo ambito è meno comune: ciò rende il progetto interessante come sperimentazione tecnica.  
Gli utenti principali saranno appassionati di giochi indie horror e sviluppatori/studenti che vogliono comprendere le potenzialità di jMonkeyEngine nel game development. La demo mira a offrire un’esperienza semplice ma immersiva, basata su grafica 2D, controlli intuitivi e atmosfere cupe tipiche del genere.

## 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-01** | |
| **Nome** | Movimento |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Nessuna |
| **Sotto requisiti** | |
| **001** | Si necessita la creazione del personaggio |
| **002** | Si dovrà poter muovere solo a destra a sinistra e saltare |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Attacca |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Nessuna |
| **Sotto requisiti** | |
| **001** | Si necessita la creazione del personaggio |
| **002** | L’attacco potrà essere sferrato solo a destra o a sinistra |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Sistema di morte |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Nessuna |
| **Sotto requisiti** | |
| **001** | Si necessita la creazione del personaggio |
| **002** | Si necessita la creazione di nemici |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Salvare la partita |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Questo requisito è meno importante poiché al momento il gioco è piccolo ma se si riesce a espandere un po’ diventa di priorità massima |
| **Sotto requisiti** | |
| **001** | Si necessita la creazione del personaggio |
| **002** | Ci dovranno essere dei punti di salvataggio |
| **003** | Sarà necessario il requisito 9 |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Caricamento dei salvataggi |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Non sarà possibile cariare un vecchio salvataggio |
| **Sotto requisiti** | |
| **001** | Si necessita del requisito 4 e va legato al requisito 3 |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Carica stanze |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Le stanze sono già prefissate |
| **Sotto requisiti** | |
| **001** | Si necessita la creazione del personaggio e i movimenti del personaggio |
| **002** | Si necessitano la creazione delle stanze |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Genera nemici |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | I nemici verranno generati al cambio di stanza la priorità è inferiore rispetto alla creazione del Boss finale |
| **Sotto requisiti** | |
| **001** | Si necessita la creazione dei modelli dei nemici e il loro sistema di movimento e attacco |
| **002** | Si necessita il REQ 06 |

|  |  |
| --- | --- |
| **ID: REQ-8** | |
| **Nome** | Creazione Boss Fight |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Se ci sarà tempo verrà aggiunta la colonna sonora e una scena di introduzione del Boss |
| **Sotto requisiti** | |
| **001** | Creazione modello e movimento del Boss |
| **002** | Si necessita il REQ 06 |

|  |  |
| --- | --- |
| **ID: REQ-9** | |
| **Nome** | Interagibilità |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I nemici verranno generati al cambio di stanza |
| **Sotto requisiti** | |
| **001** | Si necessita del REQ 1 |
| **002** | Si necessita il REQ 06 |

|  |  |
| --- | --- |
| **ID: REQ-10** | |
| **Nome** | Creazione Menu |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | nessuna |
| **Sotto requisiti** | |
| **001** | Creazione pulsante NUOVA PARTITA |
| **002** | Creazione pulsante CONTINUA per continuare una partita |
| **003** | Creazione pulsante QUIT per chiudere il gioco |

|  |  |
| --- | --- |
| **ID: REQ-11** | |
| **Nome** | Rigenerazione vita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Rigenera tutta la vita in un colpo solo |
| **Sotto requisiti** | |
| **001** | Punto di rigenerazione legato eventualmente al REQ-4 |
| **002** | Creazione sistema di vite |

### 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 1: 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.

# 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
* …