SkiFree Remake

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

Allievo: Christian Monga

Docente responsabile: Geo Petrini

Scuola: CPT Trevano

Sezione: Informatica

Classe: I3BB

Inizio Progetto: 09.09.2022

Fine Progetto. **23.12.2022**

## 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?

## Scopo

Lo scopo del progetto è quello di ricreare il vecchio gioco SkiFree della microsoft con una migliore risoluzione e qualità.

# Analisi

## Analisi del dominio

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

* Background/Situazione iniziale
* Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
* Come viene risolto attualmente il problema? Esiste già un prodotto simile?
* Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
* Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
* Esistono convenzioni/standard applicati nel dominio?
* Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
* …

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Visuale del gioco |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gioco deve essere visualizzato in full screen |
| **Sotto requisiti** | |
| **001** | Deve essere in 2D |
| **002** | Deve essere in 128bit |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Tre Modalità |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Devono essere presenti tre modalità differenti all’interno del gioco |
| **Sotto requisiti** | |
| **001** | Modalità slalom |
| **002** | Modalità tree slalom |
| **003** | Modalità freestyle |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | La mappa è infinita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I movimenti sono dettati dal mouse |
| **Sotto requisiti** | |
| **001** | La mappa viene generata automaticamente |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | La mappa contiene degli ostacoli |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Vengono generati casualmente all’interno della mappa |
| **002** | Gli ostacoli devono essere:   * Alberi (quelli piccoli vengono bruciati quando saltati ad alta velocita) * Sassi * Seggiovie * Altri sciatori o snowboarders * Trampolini * Lupi |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | I movimenti dello sciatore vengono dettati dal mouse |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | I movimenti devono essere fluidi |
| **002** | Se lo sciatore è messo orizzontalmente si ferma |
| **003** | Con il click del tasto sinistro lo sciatore salta |
| **004** | Se si tiene il tasto sinistro premuto in volo lo sciatore fa il mortale |
| **005** | Se atterra male o colpisce gli ostacoli si schianta e si ferma |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Jeti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Compare uno jeti ogni 2500m percorsi |
| **002** | Lo jeti vuole mangiare lo sciatore |
| **003** | Lo jeti è più veloce dello sciatore |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | I dati dell’utente vengono memorizzati e mostrati |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Deve essere salvato il tempo di gioco della partita |
| **002** | Deve essere salvata la distanza percorsa |
| **003** | Deve essere mostrata la velocità attuale |
| **004** | Devono essere mostrato il punteggio |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Il gioco finisce quando lo sciatore viene mangiato |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Lo jeti mangia lo sciatore |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Menù di pausa |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Cliccando “esc” si apre il menù di pausa |
| **Sotto requisiti** | |
| **001** | Blocca il gioco |
| **002** | Si può chiudere l’applicazione |
| **003** | Si può tornare all’inizio |

### 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

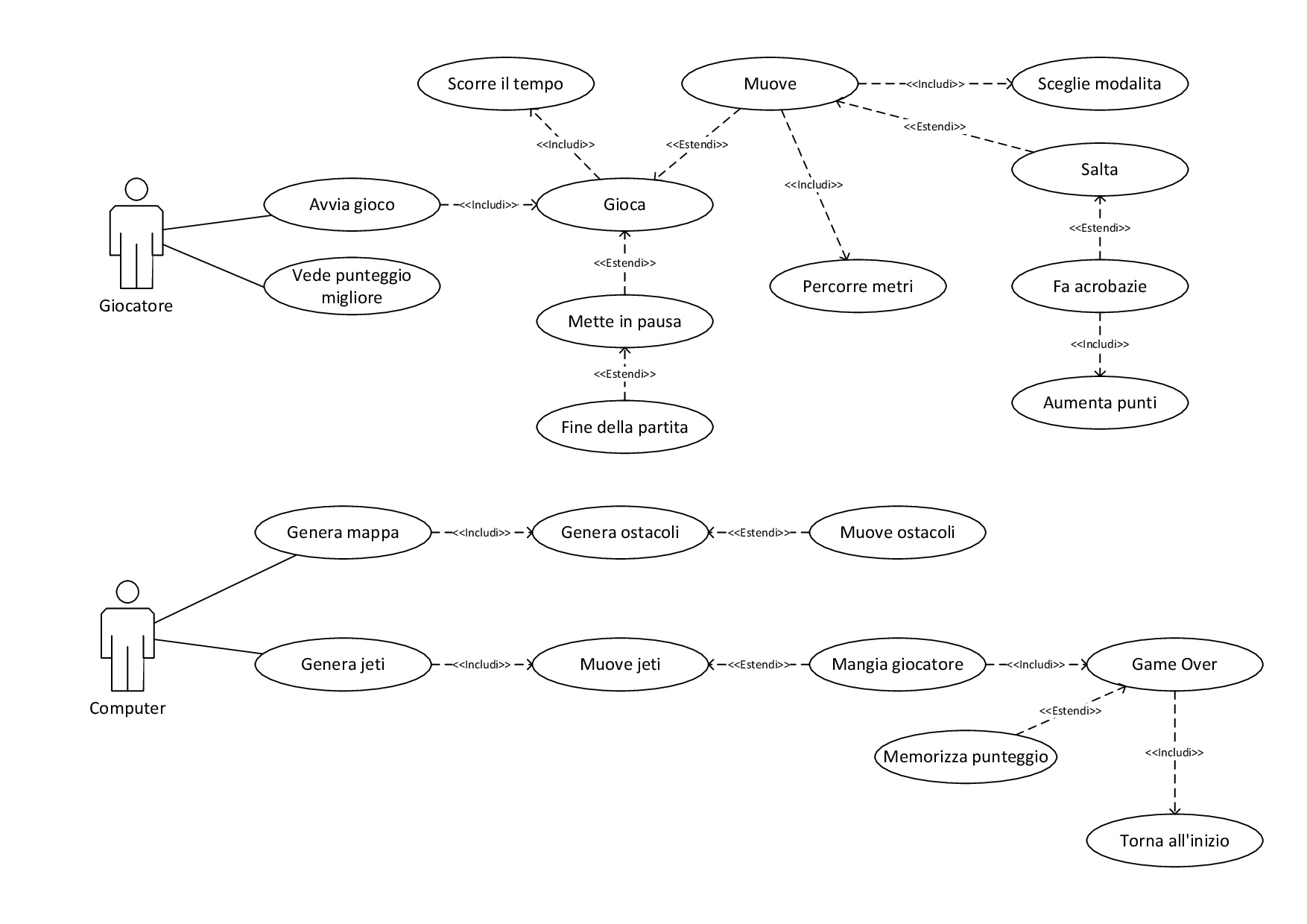
I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto. 

Figura 1 UML Use Case del progetto

## Pianificazione

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

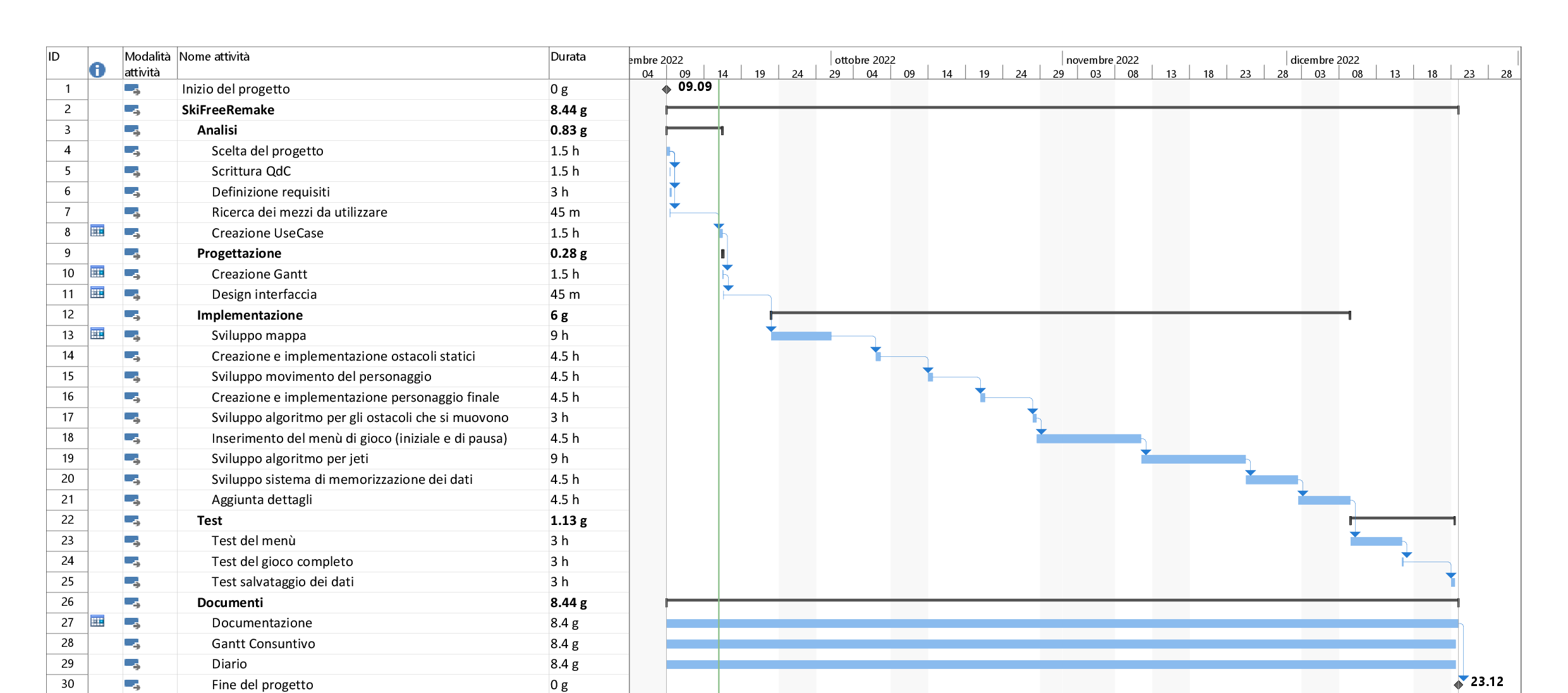


Figura 2 Diagramma di Gantt Iniziale

Per questo progetto utilizzo la pianificazione Iterativa evolutiva.

## 

## 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.

Unity 2022.1.1f1

### 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?

Il prodotto potrà essere eseguito su Windows

# 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-01 | **Nome:** | Movimenti del personaggio |
| **Descrizione:** | Controllare se i movimenti del personaggio funzionano correttamente | | |
| **Prerequisiti:** | Avvio del gioco | | |
| **Procedura:** | 1. Usare il muose 2. Posizionare il mouse sotto il personaggio e verificare che vada in avanti 3. Spostare il muose verso sinistra e verificare che il personaggio giri e che in posizione orizzontale si fermi 4. Spostare il mouse verso destra e verificare che il personaggio giri e che in posizione orizzontale si fermi | | |
| **Risultati attesi:** | Movimenti fluidi e corretti del personaggio | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-01 | **Nome:** | Movimenti del personaggio (in aria) |
| **Descrizione:** | Controllare se i movimenti del personaggio quando si trova in aria funzionano correttamente | | |
| **Prerequisiti:** | Aver verificato i movimenti base | | |
| **Procedura:** | 1. Posizionare il mouse sotto il personaggio 2. Cliccare una volta il tasto sinistro 3. Quando il personaggio si trova in volo tenere premuto il tasto sinistro | | |
| **Risultati attesi:** | Il personaggio salta e prova a fare un mortale | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-02 | **Nome:** | La mappa si genera correttamente |
| **Descrizione:** | Controllare se la generazione della mappa è fluida e corretta | | |
| **Prerequisiti:** | Aver verificato i movimenti base | | |
| **Procedura:** | 1. Posizionare il mouse sotto il personaggio 2. Andare avanti, destra e sinistra per un po’ | | |
| **Risultati attesi:** | Man mano che il personaggio si sposta la mappa genera gli ostacoli automaticamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-02 | **Nome:** | Collisione con gli ostacoli |
| **Descrizione:** | Controllare se esiste una collisione con i vari ostacoli | | |
| **Prerequisiti:** | Aver verificato i movimenti base e che la generazione della mappa avvenga in modo corretto | | |
| **Procedura:** | 1. Posizionare il mouse sotto il personaggio 2. Andare a sbattere contro gli alberi e provare il trampolino | | |
| **Risultati attesi:** | Il personaggio dovrebbe fermarsi alla collisione con gli alberi e saltare più in alto guadagnando velocità se salta sul trampolino. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-03 | **Nome:** | Controllare punteggio |
| **Descrizione:** | Controllare se quando si fa un’acrobazia vengono assegnati i punti in modo corretto | | |
| **Prerequisiti:** | Aver verificato i movimenti base e quelli aerei | | |
| **Procedura:** | 1. Posizionare il mouse sotto il personaggio 2. Saltare su un trampolino 3. Fare un’acrobazia | | |
| **Risultati attesi:** | Ogni volta che si fa un’acrobazia aumenta il punteggio | | |

## 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
* …