

Norme di progetto

Filippo Sabbadin

Indice

| | |
|--|----------|
| 1. Introduzione | 1 |
| 1.1. Scopo del documento | 1 |
| 1.2. Scopo del prodotto | 1 |
| 1.3. Glossario | 1 |
| 1.4. Riferimenti | 2 |
| 1.4.1. Riferimenti normativi | 2 |
| 1.4.2. Riferimenti informativi | 2 |
| 2. Processi primari | 3 |
| 2.1. Processo di fornitura | 3 |
| 2.1.1. Scopo e descrizione | 3 |
| 2.1.2. Attività | 3 |
| 2.1.3. Rapporto con il tutor aziendale | 3 |
| 2.1.4. Documentazione | 4 |
| 2.1.5. Analisi dei requisiti | 4 |
| 2.1.6. Piano di progetto | 4 |
| 2.1.7. Piano di qualifica | 4 |
| 2.1.8. Specifica tecnica | 4 |
| 2.1.9. Manuale utente | 5 |
| 2.2. Strumenti utilizzati | 5 |
| 2.3. Sviluppo | 5 |
| 2.3.1. Scopo e descrizione | 5 |
| 2.3.2. Attività | 5 |
| 2.3.2.1. Analisi dei requisiti | 6 |
| 2.3.2.2. Progettazione | 6 |
| 2.3.2.3. Codifica | 7 |
| 2.3.2.4. Modellazione | 7 |
| 2.3.2.5. Testing | 7 |
| 2.3.3. Regole di sviluppo | 7 |
| 2.3.4. Strumenti usati | 8 |
| 3. Processi di supporto | 9 |
| 3.1. Documentazione | 9 |
| 3.1.1. Scopo e descrizione | 9 |
| 3.1.2. Attività | 9 |
| 3.1.3. Struttura dei documenti | 9 |
| 3.1.3.1. Intestazione | 9 |
| 3.1.3.2. Indice | 10 |
| 3.1.3.3. Corpo del documento | 10 |
| 3.1.4. Documenti del progetto | 10 |

| | |
|--|-----------|
| 3.1.5. Elenchi puntati | 10 |
| 3.1.6. Diagrammi UML | 10 |
| 3.1.7. Formato date | 10 |
| 3.1.8. Composizione tipografica | 10 |
| 3.1.9. Strumenti | 10 |
| 3.2. Gestione della configurazione | 10 |
| 3.3. Gestione della qualità | 10 |
| 3.3.1. Scopo e descrizione | 10 |
| 3.3.2. Attività | 11 |
| 3.4. Verifica | 11 |
| 3.5. validazione | 11 |
| 4. Processi organizzativi | 12 |
| 4.1. Miglioramento | 12 |
| 5. Metriche e standard per la qualità | 13 |
| 5.1. Funzionalità | 13 |
| 5.2. Prestazioni | 13 |
| 5.3. Affidabilità | 13 |
| 5.4. Sicurezza | 13 |
| 5.5. Usabilità | 13 |
| 5.6. Manutenibilità | 13 |
| 5.7. Portabilità | 13 |

Lista di immagini

Lista di tabelle

1. Introduzione

1.1. Scopo del documento

Il presente documento ha lo scopo di definire le norme di progetto che verranno seguite durante lo sviluppo del videogioco.

Le norme di progetto sono un insieme di regole e linee guida che devono essere seguite durante lo sviluppo del progetto, al fine di garantire la qualità del prodotto finale e la sua manutenibilità nel tempo.

Esse comprendono le convenzioni di codifica, le linee guida per la documentazione, le procedure di test e le pratiche di gestione del progetto.

1.2. Scopo del prodotto

Il progetto consiste nello sviluppo di un videogioco con il motore di gioco [Godot*](#) di tipo [platformer*](#), in cui il giocatore controlla un personaggio che si muove in un ambiente tridimensionale.

Il gioco si basa su temi di [Intelligenza Artificiale*](#) e [Machine Learning*](#), sviluppando meccaniche ed elementi del livello basati su questi argomenti.

Il gioco comprende 3 livelli, ognuno con un tema diverso, un menu principale che viene visualizzato non appena il gioco viene avviato, e un menu di pausa che può essere visualizzato in qualsiasi momento durante il gioco.

1.3. Glossario

Per facilitare la comprensione del documento, è stato creato un glossario che contiene i termini utilizzati nel documento e le loro definizioni. I termini presenti nel glossario sono colorati di blu e seguiti da un'asterisco: [esempio*](#).

Il glossario è accessibile tramite il link:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Glossario.pdf>

oppure consultando il rispettivo documento all'interno della stessa cartella.

1.4. Riferimenti

1.4.1. Riferimenti normativi

- Piano di lavoro:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Piano-di-lavoro.pdf>

1.4.2. Riferimenti informativi

- Glossario:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Glossario.pdf>

2. Processi primari

2.1. Processo di fornitura

Il processo di fornitura è strutturato in conformità agli esiti previsti dalla clausola 5.2 dello [standard*](#) ISO/IEC 12207:1997. Tale processo include la definizione di requisiti concordati, l'analisi dei rischi associati, e la pianificazione di tempi e costi.

2.1.1. Scopo e descrizione

Il processo di fornitura ha lo scopo di garantire che il prodotto software sia consegnato al cliente in modo completo e funzionante.

2.1.2. Attività

Il processo di fornitura comprende le seguenti attività:

- Analisi dei requisiti:
raccolta e analisi dei requisiti del cliente, definizione delle specifiche del prodotto.
- Confronto:
confronto tra i requisiti del [relatore aziendale*](#) e le specifiche del prodotto, identificazione delle eventuali discrepanze.
- Pianificazione:
pianificazione delle attività di sviluppo, stima dei tempi e dei costi, definizione del piano di progetto.
- Progettazione:
definizione dell'architettura del sistema, progettazione dei componenti software.
- Sviluppo:
implementazione del codice sorgente, creazione dei test unitari.
- Verifica e validazione:
esecuzione dei test di integrazione, verifica della conformità alle specifiche.
- Consegna:
consegna del prodotto al cliente, formazione degli utenti, raccolta dei feedback.

2.1.3. Rapporto con il tutor aziendale

Il rapporto con il tutor aziendale è fondamentale per garantire che il prodotto software soddisfi le aspettative stabilite. Il tutor aziendale viene contattato in due modalità:

-
- Incontri periodici:
incontri settimanali per discutere lo stato di avanzamento del progetto, raccogliere feedback e risolvere eventuali problemi.
 - Comunicazioni via email:
comunicazioni via email per condividere documenti, report di avanzamento e richieste di chiarimenti.

2.1.4. Documentazione

Il processo di fornitura deve essere documentato in modo dettagliato, in modo da garantire la tracciabilità delle attività svolte e delle decisioni prese.

2.1.5. Analisi dei requisiti

Nel documento *Analisi dei requisiti* sono stati definiti tutti gli [use case*](#) e i requisiti funzionali del progetto. Questi sono stati raccolti in collaborazione con il tutor aziendale e sono stati utilizzati come base per la progettazione e lo sviluppo del software.

2.1.6. Piano di progetto

Il *Piano di progetto* è un documento che definisce le attività da svolgere e i tempi previsti per lo sviluppo del software.

Viene descritto in dettaglio ogni periodo di sviluppo, con una retrospettiva delle attività svolte e una pianificazione delle attività future.

2.1.7. Piano di qualifica

Nel documento *Piano di qualifica* sono definite le metriche che vengono usate per garantire la qualità del prodotto software. Vengono inoltre scritte le modalità di test e verifica del software, in modo da garantire che il prodotto soddisfi i requisiti stabiliti.

2.1.8. Specifica tecnica

La *Specifica tecnica* è un documento che descrive in dettaglio l'architettura del sistema, i componenti software e le loro interazioni.

2.1.9. Manuale utente

Il *Manuale utente* è un documento che fornisce istruzioni dettagliate su come utilizzare il software e garantirne il corretto funzionamento.

2.2. Strumenti utilizzati

Gli strumenti utilizzati per lo sviluppo del software sono i seguenti:

- **Git**: sistema di controllo versione utilizzato per gestire il codice sorgente e le modifiche apportate.
- **GitHub**: piattaforma di hosting per il codice sorgente e la gestione del progetto, utilizzata per la collaborazione tra i membri del team di sviluppo.
- **Typst**: linguaggio di markup utilizzato per la composizione tipografica della documentazione.
- **Notion**: strumento di collaborazione e gestione dei progetti utilizzato per la pianificazione delle attività.

2.3. Sviluppo

2.3.1. Scopo e descrizione

Il processo di sviluppo è finalizzato alla realizzazione del software richiesto dal tutor, in base ai requisiti stabiliti durante i colloqui. Questo processo comprende tutte le attività necessarie per la creazione del prodotto, dalla progettazione all'integrazione, garantendo che il prodotto creato soddisfi i requisiti concordati. Questo processo prevede le seguenti attività:

2.3.2. Attività

- **Analisi dei requisiti;**
- **Progettazione;**
- **Codifica;**
- **Modellazione;**
- **Testing;**

Ovviamente, il prodotto software atteso deve essere un software funzionante, che soddisfi i requisiti concordati e ampiamente testato. Sarà dunque necessario seguire i requisiti concordati seguendo le linee guida fissate.

2.3.2.1. Analisi dei requisiti

Lo sviluppo inizia con l'analisi dei requisiti, durante la quale vengono identificati tutti i casi d'uso e i requisiti dell'applicazione. Tutta questa attività è documentata nel documento «Analisi dei requisiti». I requisiti devono descrivere:

- funzionalità richieste;
- funzionalità di supporto;

I casi d'uso descrivono le interazioni tra il prodotto e gli attori coinvolti, ed aiutano ad individuare ulteriori requisiti. L'applicazione presenta un solo attore:

- giocatore.

Ogni use case ha la seguente struttura:

- diagramma del caso d'uso (se lo use case non è già presente in un altro diagramma);
- attori principali;
- attori secondari (se presenti);
- descrizione;
- precondizioni;
- postcondizioni;
- scenario principale;
- generalizzazioni, estensioni ed inclusioni, se presenti.

Per i requisiti, vi sono 3 tipi:

- funzionali;
- di qualità;
- di vincolo.

2.3.2.2. Progettazione

La progettazione è l'attività di definizione dell'architettura del software dal punto di vista logico. In questa fase si decide come soddisfare i requisiti identificati durante l'analisi.

In particolare, vengono definite le unità dell'applicazione e le loro interazioni, prestando attenzione a mantenerle separate e indipendenti per garantire una maggiore manutenibilità e scalabilità del sistema. Inoltre vanno definite le responsabilità che verranno applicate in fase di codifica assicurandosi di mantenere un livello di [efficienza*](#) e [efficacia*](#) il più alto possibile. L'approccio utilizzato in questa attività sarà sia [top-down*](#), per scomporre il problema in sotto-problemi, sia [bottom-up*](#), per ragionare sui singoli sotto-problemi e integrarli in una soluzione complessiva.

L'architettura prevista a fine attività sarà a monolite, dove tutti i componenti sono integrati e compilati in una singola applicazione.

2.3.2.3. Codifica

Obiettivo della fase di codifica è applicare le decisioni prese durante la fase di progettazione. Viene eseguita in parallelo con la fase di modellazione e consiste nella scrittura del codice sorgente secondo le specifiche definite nella fase di progettazione. Durante questa attività, si applicano le convenzioni di codifica stabilite e ogni componente viene sviluppato in modo modulare.

2.3.2.4. Modellazione

La fase di modellazione viene eseguita in parallelo con la fase di codifica. Comprende le seguenti sotto-attività:

- creazione ed esportazione dei modelli 3D;
- creazione delle animazioni;
- [baking*](#) delle textures.

I modelli vengono esportati nel formato [.glb*](#) e inseriti dentro la cartella *models*, assicurandosi che rispettino le specifiche tecniche e stilistiche definite in fase di progettazione.

2.3.2.5. Testing

Durante questa fase, vengono testati singolarmente i componenti e integrati progressivamente nel sistema complessivo. Vengono poi testate le prestazioni.

Eventuali problemi riscontrati vengono risolti tempestivamente, aggiornando la documentazione tecnica ove necessario.

2.3.3. Regole di sviluppo

Durante lo sviluppo, sono state imposte le seguenti regole per garantire omogeneità e correttezza. Le regole sono suddivise in base all'argomento:

• Codifica:

Tutti i file script del gioco sono salvati come file [.gd*](#), e sono scritti con il linguaggio di programmazione [GDScript*](#). I nomi delle classi sono salvate con una nomenclatura [PascalCase*](#), mentre i nomi dei files e delle variabili usano [snake_case*](#).

Per maggiori dettagli sulla nomenclatura, si seguono le convenzioni della documentazione ufficiale:

https://docs.godotengine.org/it/4.x/tutorials/scripting/gdscript/gdscript_styleguide.html

• Modellazione:

Tutti i modelli sono esportati nel formato [.glb*](#). I [materiali*](#) vengono esportati senza immagini, dato che verranno rimpiazzati dal materiale presente nel gioco. Nel caso il modello presenti animazioni, vengono esportate insieme al modello.

- **Animazione:**

Le animazioni sono incluse nel modello durante l'esportazione. Per semplificare l'attività, viene usato un [rig*](#) che dispone di [Inverse Kinematics*](#). Le animazioni sono già separate prima dell'esportazione e possono essere trovate nella sezione [NLA*](#) e selezionate individualmente premendo la linea con il mouse e modificarle usando la scorciatoia Shift+TAB.

- **Textures:**

Le textures sono salvate come semplici immagini di tipo [.png*](#). Entrambe le dimensioni della texture (larghezza e altezza) devono essere una potenza di 2. Risoluzioni esempio:

- 256x256
- 512x512
- 1024x1024 (1K)
- 2048x2048 (2K)

Di norma, 1024 pixels corrispondono a 1 metro.

2.3.4. Strumenti usati

- **Visual Studio Code:** editor di testo utilizzato per la scrittura del codice sorgente e della documentazione.
- **Godot:** motore di gioco utilizzato per lo sviluppo del videogioco.
- **Blender:** software di modellazione 3D utilizzato per la creazione dei modelli, delle animazioni e delle texture del videogioco.
- **GIMP:** software di modifica delle immagini utilizzato per la modifica occasionale di [texture*](#) o e delle immagini del videogioco.
- [Gamescope*](#): strumento che fornisce un overlay durante il gioco, mostrando le varie statistiche come fps, [latenza*](#), etc...

3. Processi di supporto

3.1. Documentazione

3.1.1. Scopo e descrizione

Il processo di documentazione ha lo scopo di garantire che tutta la documentazione necessaria per il progetto sia creata e mantenuta aggiornata.

In questa sezione verranno descritte le decisioni prese e rispettate, in modo da garantire la coerenza tra i vari documenti.

3.1.2. Attività

Il processo di documentazione comprende le seguenti attività:

- **Realizzazione:** creazione o modifica di un determinato documento, in base alle esigenze del progetto, si divide in:
 - **individuazione compito:** identificazione della sezione del documento da modificare o creare.
 - **redazione:** modifica del documento;
- **Revisione:** revisione delle modifiche applicate al documento, e decisione se approvarle o modificarle ulteriormente, comprende le seguenti attività:
 - **revisione interna:**
- **Aggiornamento:** comprende le seguenti attività:
 - **push:** i documenti vengono caricati nell [repository*](#) di [GitHub*](#)
 - **compilazione:** modifica dei documenti in base alle esigenze del progetto, in modo da garantire la coerenza tra i vari documenti.

3.1.3. Struttura dei documenti

3.1.3.1. Intestazione

- **Titolo:**
- **Autore:**

3.1.3.2. Indice

3.1.3.3. Corpo del documento

3.1.4. Documenti del progetto

3.1.5. Elenchi puntati

3.1.6. Diagrammi UML

3.1.7. Formato date

3.1.8. Composizione tipografica

3.1.9. Strumenti

3.2. Gestione della configurazione

3.3. Gestione della qualità

3.3.1. Scopo e descrizione

Il processo di gestione della qualità ha lo scopo di garantire che il prodotto software soddisfi

3.3.2. Attività

3.4. Verifica

3.5. validazione

4. Processi organizzativi

4.1. Miglioramento

5. Metriche e standard per la qualità

5.1. Funzionalità

5.2. Prestazioni

5.3. Affidabilità

5.4. Sicurezza

5.5. Usabilità

5.6. Manutenibilità

5.7. Portabilità