

Project Plan - Santorini

1. Introduzione

Il progetto si propone di sviluppare una versione digitale con interfaccia grafica (GUI) del gioco da tavolo “Santorini” mettendo in pratica le nozioni di ingegneria del software apprese durante il corso. La versione digitale, oltre a fornire un ambiente di gioco intuitivo e facilmente accessibile, servirà come esempio concreto dei processi, delle tecniche e delle metodologie di sviluppo software studiati.

Le persone responsabili del progetto sono:

- Andrei Petrisor (1085993)
- Andrea Rusconi (1086646)
- Uriel Fumagalli (1085169)
- Antonio Radu (1085992)

TODO: da rifinire una volta finito il progetto

2. Modello di processo

Utilizzeremo il framework di sviluppo Agile, in particolare SCRUM, organizzato nei seguenti modi:

- Sprint di durata settimanale
- Daily scrum meeting brevi (WhatsApp, Discord o di persona) per monitorare l'avanzamento
- Sprint review e retrospective al termine di ogni sprint per valutare progressi, la qualità del prodotto e introdurre eventuali miglioramenti

Pietre miliari (Milestones):

- a. Creazione e validazione dei diagrammi UML di base
- b. Implementazione delle classi e metodi principali (Papyrus) (Versione **1.0.0-alpha.1**)
- c. Creazione parte server con GUI (Swing) e connessione server
- d. Implementazione regole gioco base sul server (Versione **1.0.0-beta.1**)
- e. Implementazione dei 10 “god powers” semplici (Versione **1.0.0**)
- f. (Opzionale) Implementazione leaderboard
- g. (Opzionale) Implementazione dei 20 “god powers” avanzati

3. Organizzazione del progetto

Il cliente domanda la versione digitale del gioco in scatola “Santorini”. Richiede che il programma possieda le regole complete del gioco in scatola e le prime due di queste difficoltà:

1. Gioco senza “god powers” (Difficoltà “Easy”)
2. Gioco con 10 “god powers” semplici (Difficoltà “Normal”)
3. Gioco con 20 “god powers” avanzati (Difficoltà “Advanced”). - opzionale

Inoltre il cliente vuole poter vedere una classifica nella pagina principale (Home) del gioco. - opzionale

Ruoli del team:

- Product owner: Uriel Fumagalli
- SCRUM master: Andrei Petrisor
- Developer: Uriel Fumagalli, Andrei Petrisor, Andrea Rusconi, Antonio Radu

4. Standard, linee guida, procedure

Linguaggio e Tecnologie:

- Linguaggio: Java
- GUI: Swing
- Diagrammi UML: Papyrus
- IDE raccomandato: Eclipse

Standard di Programmazione:

- Aderenza alle convenzioni Java definite da Oracle (naming convention, commenti Javadoc, formattazione del codice)

Controllo di Versione:

- Repository Git (GitHub).
- Branching: un branch principale “main” stabile, branch di feature per nuove funzionalità, pull request con code review prima del merge

Documentazione:

- Documentazione del codice con Javadoc
- Diagrammi UML aggiornati a ogni milestone principale

TODO: rifinire quando iniziamo a scrivere il programma

5. Attività di gestione

Alla fine di ogni sprint avremo una riunione dove aggiorneremo lo stato dello sviluppo e analizzeremo il lavoro compiuto nello sprint (Sprint Review), insieme a una discussione sui possibili miglioramenti applicabili allo sviluppo del progetto (Sprint Retrospective).

Requisiti:

- Funzionali obbligatori
 - Gioco base senza god powers (modalità “Easy”)
 - Gioco con 10 god powers semplici (modalità “Normal”)
- Funzionali opzionali
 - 20 god powers avanzati
 - Leaderboard giocatori
- Non funzionali
 - Performance
 - Manutenibilità, codice ben strutturato e aderente agli standard

Tempi: è richiesta la consegna del programma funzionante entro 5 giorni dalla prova d’esame

Costi: I nostri costi non saranno monetari ma misurati in ore di lavoro per sviluppatore.

6. Rischi

I potenziali rischi sono:

- Mancanza di esperienza nello sviluppo e organizzazione del software
- Mancanza di ore di lavoro che ogni membro avrà da dedicare nel progetto

- Difficoltà nella gestione del progetto
- Difficoltà nella gestione dei test

L'hardware non sarà un potenziale rischio di questo progetto grazie alla leggerezza del software.

7. Personale

Il team si compone di 4 Undergraduate Junior Developer con conoscenza base di Java, alcuni dei quali hanno maggiore familiarità con la creazione di GUI in Swing, altri con la logica di gioco

Nomi del personale:

- Andrei Petrisor
- Andrea Rusconi
- Uriel Fumagalli
- Antonio Radu

8. Metodi e tecniche

Progettazione:

- Diagrammi UML di classe, state, communication, use case e sequence mantenuti sempre aggiornati durante lo sviluppo
- Utilizzo di Kanban Board per monitorare il progresso del progetto con priorità distinte per ogni task

Implementazione:

- Rispetto degli standard di codifica e utilizzo di branching su Git, code review tra il personale per garantire la qualità del codice

Testing:

- TODO

Gestione della configurazione:

- Versionamento semantico (1.0.0-alpha.1, 1.0.0-beta.1, 1.0.0, ecc...)

TODO: rifinire quando stiamo scrivendo i test

9. Garanzia di qualità

Fattori di qualità:

- Modello di qualità esterno ed interno dello standard ISO 9126
- Nessun bug critico noto alla release finale
- Rispetto standard di codifica e nessun "warning" critico in fase di compilazione

Organizzazioni e procedure per garantire i fattori di qualità:

- Si monitorerà adeguatamente il software durante il suo processo di sviluppo
- Si assicurerà il pieno rispetto degli standard e garantire che le inadeguatezze siano evidenziate affinché vengano sistemate dai developer
- Sarà presente una code review tra pari periodica
- Test ad ogni sprint

10. Pacchetti di lavoro (workpackages)

TODO: dividerci i compiti e determinare chi fa cosa man mano

11. Risorse

Hardware: Computer/laptop con accesso a Internet e JDK installato

Software: JDK, Eclipse, Papyrus, GitHub

Comunicazione: Discord/WhatsApp per daily scrum, repository GitHub per controllo versione, eventuali incontri in presenza

12. Budget e programma

Il budget per questo progetto non sarà monetario ma definito da un quantitativo minimo di tempo utilizzato per lo sviluppo.

Questo budget è identificato in 40 ore di lavoro minime per sviluppatore.

13. Cambiamenti

Metodologia per come verranno trattati i cambiamenti:

- Tutte le modifiche proposte vengono discusse nel daily scrum
- Se approvate, vengono inserite nel backlog e pianificate nel successivo sprint
- Nessuna modifica entra dalla “porta sul retro”, saranno tutte tracciate su GitHub tramite issues

14. Consegna

La procedura di consegna per questo progetto è fissata per una data specifica, cinque giorni prima della presentazione finale, e prevede il caricamento di una versione eseguibile del software su un repository GitHub, accompagnata da tutta la documentazione e dai diagrammi utilizzati durante lo sviluppo.