



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. Carmine Gravino

Statement of Work

Progetto

DashingCube

Riferimento	DC_SOW
Versione	0.1
Data	02/10/2024
Destinatario	Prof. Carmine Gravino
Presentato da	Francesco Botta, Vincenzo Beniamino Fresa
Approvato da	



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. Carmine Gravino

Revision History

Data	Versione	Descrizione	Autori
02/10/2024	0.1	Prima stesura	Francesco Botta Vincenzo Beniamino Fresa
13/10/2024	0.2	Prima revisione	Francesco Botta Vincenzo Beniamino Fresa



Statement of Work (SOW) del Progetto

DashingCube

1. Piano Strategico/Strategic Plan

Un imprenditore, notando la grande popolarità dei videogiochi, ha commissionato la creazione di un videogioco da un gruppo di nuovi specialisti nel settore.

2. Obiettivi di Business/Business Needs

L'obiettivo del prodotto è quello di inserirsi nel mercato videoludico cercando di tastare il terreno nei gusti degli attuali videogiocatori e di attirare possibili nuovi investitori. Inoltre, la creazione di un nuovo gruppo di programmatori ha lo scopo di formare nuove leve nel settore e di attrarne di nuove in futuro. L'ambiente di sviluppo utilizzato è quello di Unity.

3. Ambito del Prodotto/Product Scope

L'obiettivo del progetto è fornire un videogioco del tipo “Endless Runner” in grado di divertire l'utenza di ogni età attraverso un gioco semplice nelle meccaniche ma con elevata rigiocabilità sfruttando l'effetto psicologico di questo tipo di giochi nel creare dipendenza, vedasi il caso di Flappy Bird. Le meccaniche del gioco saranno:

- la possibilità di saltare su varie piattaforme cercando di evitare di cadere
- sistema monetario che permette l'acquisto di vari accessori e scenari
- la possibilità di registrare il punteggio ottenuto, puntando a migliorare ad ogni partita successiva
- meccanica di un potenziamento in grado di semplificare il percorso del giocatore
- livelli di difficoltà con scenari diversi ad ognuno di essi.

4. Data di Inizio e di Fine

Inizio: Ottobre 2024

Fine: Gennaio-Febbraio 2025.



5. Deliverables

- RAD, SDD, ODD, Matrice di Tracciabilità, Test Plan, Test Case Specification, Test incident Report, Test Summary Report, Manuale D'Uso, Manuale Installazione e ogni altro documento richiesto per lo sviluppo del sistema.

6. Vincoli/Constraints

Vincoli collaborativi e comunicativi.

- Rispetto scadenze delle scadenze intermedie/di fine progetto **definite nello statement of work**
- Budget/Effort non superiore a $50 \cdot n$ ore dove n sono i membri del team
- Uso di sistemi di versioning - GitHub in particolare
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo
- Utilizzo di tool di per la suddivisione dei task e attività (Trello o similare)
- Utilizzo di tool di comunicazione tracciabile (Discord)

Vincoli tecnici

Analisi e specifica dei requisiti

- Specifica di **minimo** 2 e **massimo** 4 scenari per ogni membro del team;
- Diagramma degli use
- Esattamente uno use case per ogni membro del team - i casi d'uso aggiuntivi non saranno valutati;
- Specifica dei requisiti funzionali e non funzionali relativi a uno use case selezionato.
- Specifica degli oggetti boundary, control e entity per gli use case specificati.
- Un sequence diagram ogni due membri del team - il sequence diagram deve far riferimento agli use case specificati. • Specifica di un class diagram per team - (gli object diagram non verranno valutati).
- Almeno un activity diagram per sistema esistente o sistema proposto e almeno uno statechart diagram ogni due membri del team
- Definizione di **un diagramma** di decomposizione dei sottosistemi per team, con annessa descrizione e motivazione all'uso.
- Definizione di **un deployment diagram** per team, con annessa descrizione e motivazione all'uso.



System Design

- Specifica dei design goal e analisi dei trade-off relativi ad almeno due coppie di design goal.
- Definizione dell'architettura del sistema.
- Individuazione di dove poter utilizzare due design pattern, indicandone l'obiettivo e come sarebbero implementati.
- Uso di UML;

Testing

- Ogni studente dovrà effettuare il testing di unità, tramite category partition, dei casi d'uso specificati.

7. Criteri di Accettazione/Acceptance Criteria (Criteri che, se non rispettati, portano al fallimento del progetto)

- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Adeguato utilizzo del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Adeguato utilizzo di Discord, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Adeguato utilizzo di Trello, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Documentazione adeguata. Verranno usati tool di **plagiarism detection** per identificare casi in cui gli studenti hanno copiato da progetti di anni precedenti e/o da altre fonti.
- Appropriato test di unità di un metodo sviluppato, che preveda il **rispetto dei vincoli**.
- Appropriato test di sistema di una funzionalità del sistema sviluppato, che preveda il **rispetto dei vincoli**.

8. Criteri di premialità

- Uso adeguato di sistemi di **build**;
- Uso adeguato di un processo di **continuous integration** tramite Travis;
- Uso adeguato di tool di controllo della qualità (ad esempio, **CheckStyle**);



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. Carmine Gravino

- Adozione di processi di **code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).