**1. Introduzione**

* Nome del progetto: Fall Simulator
* Obiettivo del progetto: Il progetto ha l’obiettivo di simulare il moto di caduta libera lungo curve di corpi su cui agisce la sola gravità; calcolando i parametri del moto tra cui di particolare interesse il tempo totale di caduta.
* Motivazione e contesto: La questione che ha dato origine a Fall Simulator è il Problema della Brachistocrona (Johann Bernoulli, 1696),uno storico problema di ottimizzazione che richiede di trovare la curva che minimizza il tempo per percorrere lo spazio tra due punti che un corpo in caduta sotto l'effetto della gravità impiega per percorrere lo spazio tra due punti.Il problema si risolve matematicamente con la teoria del Calcolo delle Variazioni, ottenendo che la soluzione ottimale è la curva cicloide, ovvero la traiettoria tracciata dal moto di un punto di una circonferenza che rotola. Il progetto mira a mostrare accuratamente la meccanica del moto in questione e analizzarne i parametri, grazie a calcoli numerici precisi che ricreano il modello del fenomeno fisico.
* Stakeholder: Persone interessate all' analisi simulativa di un modello di fenomeno fisico legato a uno storico problema di fisica matematica. Figure educative che vogliono insegnare il Problema o il fenomeno fisico in modo visivo e chiaro.
* Sintesi del progetto: Il software permette di selezionare due punti nello spazio (con la possibilità di includere punti intermedi) e generare una curva che passa per questi punti. Successivamente, anima una sfera lungo questa curva per studiare il suo comportamento. FallSimulator propone inoltre curve preimpostate, come parabole, rette e cicloidi, per analisi rapide.

**2. Modello di Processo**

Per sviluppare Fall Simulator viene seguito il modello di ciclo di vita della prototipazione evolutiva, addentrandosi presto nei dettagli e nelle problematiche per rivelare fin da subito le questione che richiedono il maggiore sforzo, le quali verranno affrontate in modo più pianificato, con l'obbiettivo di aggiustare e fare evolvere il prototipo verso il software finale. L'approccio allo sviluppo è di tipo Agile piuttosto che basato sulla pianificazione, in particolare nella gestione del processo.

**3. Organizzazione del progetto**

Il team di sviluppo del progetto FallSimulator è composto da due sviluppatori, Marco Locatelli e Nava Nicolò, che collaborano su tutti gli aspetti del progetto. Nicolò ha il focus sull' organizzazione del processo, la documentazione la gestione del progetto,e il testing del sistema; mentre Marco è focalizzato sull'analisi del dominio, la progettazione, modellazione e implementazione del software. Entrambi collaborano e revisionano il lavoro dell'altro.

* Revisore: Il progetto verrà valutato dal professore Gargantini, docente di ingegneria del software.
* Strumenti e comunicazione: Il team utilizza GitHub per il controllo di versione e la gestione del codice. La maggior parte degli incontri e dei confronti avviene di persona, facilitando una comunicazione diretta e collaborativa, oltre a mezzi di comunicazione rapidi e informali.

**4. Standard, linee guida e procedure**

* Convenzioni di codifica: Il team segue le convenzioni ufficiali di Oracle per Java, incluse regole come:
* Classi in PascalCase.
* Variabili e metodi in camelCase.
* Costanti in MAIUSCOLO\_CON\_SOTTOLINEATURE.
* Strumenti di sviluppo: Il progetto utilizza Maven per la gestione delle dipendenze e il controllo del ciclo di build. Il repository è gestito su GitHub per il controllo di versione.
* Gestione della documentazione: Anche se la documentazione non è un focus primario, il team utilizza Javadoc per mantenere una documentazione minima del codice. Nicolò si occupa principalmente della documentazione di progetto.
* Code review: Le modifiche al codice vengono sottoposte a revisione tramite GitHub, per garantire la qualità del codice e prevenire errori.
* Controllo di versione: Il team utilizza GitHub per gestire le versioni del progetto e tenere traccia delle modifiche.

**5. Attività di gestione**

* Monitoraggio del progresso: Il team utilizza Daily Stand-up Meeting per confrontarsi quotidianamente sui progressi e identificare eventuali ostacoli. Inoltre, dopo ogni Sprint, viene svolta una Sprint Review per esaminare il lavoro completato e pianificare le attività successive.
* Stabilire le priorità: Le funzionalità principali, come l'animazione e il calcolo del tempo, hanno la priorità. Una volta completate, il team procede con l'aggiunta di altre funzionalità, come le curve preimpostate e le ottimizzazioni.
* Gestione dei problemi: I problemi vengono risolti attraverso una comunicazione diretta all'interno del team. Se il problema non può essere risolto internamente, viene richiesto aiuto al professore Gargantini.