**GRUPPO 23**

**RELAZIONE PRIMO SPRINT**

**SPRINT REVIEW PRIMA SETTIMANA**

All’inizio del primo sprint, durante la fase di pre-game, era stata stimata una velocità di **54 SP**. Erano state definite 14 user stories, da completare entro il 28/11/21. Alcuni dei task di tali user stories non sono stati completati, per questo è stato riportato un **Technical Debt di 10** **sp**, dovuto alla mancanza del completamento dei **task** relativi alla gestione dell’aggiornamento della text area dopo le operazioni di clear, drop, dup, swap, over, alle operazioni di square root e inverted sign e alla mancanza dell’inserimento di numeri allo stack tramite il pulsante d’invio; di conseguenza **la velocità** si è ridotta arrivando a **44 sp**.

* E’ stata definita un’interfaccia **GUI** la quale permette all’utente di inserire in maniera esplicita i numeri, per poi essere visualizzati in un’area di testo. Da essa è possibile effettuare calcoli in modo agevole, attualmente solo per le operazioni basilari di somma, sottrazione, moltiplicazione e divisione; inoltre è possibile ripulire l’area di testo e svuotare lo stack in memoria attraverso l’uso di un “clear” button.
* La classe **Calculator**, gestita attraverso un pattern Singleton, permette di gestire alcune delle operazioni necessarie per lo stack e richieste dal cliente.
* La classe **ComplexNumber** consente di definire un numero complesso in forma Cartesiana, quindi, composto di una parte immaginaria e una reale, mentre la classe **ComplexOperations** si occupa di tutte le operazioni necessarie e sempre richieste dal cliente, per gestire i numeri complessi. Attualmente la radice restituisce un unico risultato, solo per k=0, nel prossimo sprint verrà gestita anche la soluzione per k > 0, dove k indica un numero compreso tra 0 e n-1, dove n indica l’indice di radice.
* La classe **Model,** definisce la struttura dati predominante, necessaria per implementare la logica della calcolatrice a stack (LIFO mode), inoltre sono stati definiti in essa tutti i metodi necessari per operare su di essa.
* Infine, è stata definita una classe **ParserString**, che consente di trasformare una stringa in input, in un numero complesso. Essa gestisce correttamene l’insieme dei numeri immaginari e inoltre sono stati gestiti tutti i casi d’inserimento da parte di un utente errati (eliminazione di spazi, segni in sospeso, inserimento di due parti reali/immaginarie ecc.). Tali diverse casistiche sono enunciate all’utente tramite dei messaggi esplicativi della causa dell’inserimento non valido.
* Sono sempre stati gestiti tutti i casi di test per tutte le classi

**Primo Sprint – Retrospective**

* **Stop (cose da smettere di fare):**

Smettere di scrivere codice senza aver consultato all’inizio della giornata lavorativa il resto del team ( prevalentemente è stato fatto a fine giornata, creando varie discussioni).

Smettere di occuparsi del codice altrui, senza coinvolgere prima l’altro membro del gruppo.

Non aggiornare Trello alla fine di ogni task.

* **Less of (cose da fare di meno):**

Smettere di occuparsi del codice altrui senza coinvolgere prima l’altro membro del gruppo

* **Keep doing (cose da continuare a fare):**

Continuare a fare commit di codice testato, e accorparlo al ramo principale.

* **More of (cose da fare di più):**

Organizzare meglio il lavoro e soprattutto i vari task.

Eseguire più spesso una revisione del codice altrui.

* **Start (cosa da iniziare a fare):**

Organizzare e dividere in modo efficiente il lavoro.

Immagine che contiene testo

Descrizione generata automaticamente

**ORGANIZZAZIONE DEL SECONDO SPRINT.**

Nel prossimo sprint si inizierà a lavorare a partire da tali task non completati, per poi riprendere il lavoro dalle user stories successive, definite nello Sprint Backlog della seconda settimana di lavoro.