Immagine che contiene scatola

Descrizione generata automaticamente

Università degli Studi di Salerno

Immagine che contiene testo, clipart

Descrizione generata automaticamente

Dipartimento di Ingegneria dell’Informazione ed Elettrica e Matematica Applicata

Corso di Laurea in Ingegneria Informatica

**Software Architecture Design 2024/2025**

**Canale I-Z**

**REVIEW REPORT**

Gruppo n. **3 – IZ**

|  |  |  |
| --- | --- | --- |
| **Cognome e Nome** | **Matricola** | **e-mail** |
| Masturzo Vincenzo Pietro Pio | 0622702589 | v.masturzo@studenti.unisa.it |
| Santonicola Federico | 0622702567 | f.santonicola10@studenti.unisa.it |
| Sessa Attilio Marco | 0622702468 | a.sessa111@studenti.unisa.it |
| Treglia Martina Rosaria | 0622702656 | m.treglia3@studenti.unisa.it |

**Anno accademico 2024-2025**

***Sommario***

[**1.** **Project Velocity** 3](#_Toc199857641)

[**2.** **Implementation issues** 3](#_Toc199857642)

[**3.** **Product Backlog issues** 3](#_Toc199857643)

1. **Project Velocity**

Durante questo Sprint, il team aveva pianificato di completare User Story per un totale di **26 Story Points**. Al termine dello Sprint, sono stati effettivamente completati **23 Story Points**. Sebbene la differenza sia contenuta, questo dato evidenzia una leggera discrepanza rispetto alla stima iniziale. Pertanto, per la pianificazione del prossimo Sprint, si terrà conto di una velocity leggermente inferiore, al fine di mantenere previsioni più aderenti alla capacità reale del team.

1. **Implementation issues**

Durante lo sviluppo del rilascio, sono state riscontrate le seguenti problematiche:

* Il nostro sistema gestisce tutte le forme tramite l’interfaccia ShapeInterface, che funge da adattatore per le classi del package javafx.scene.shape. Tuttavia, alcune classi esterne necessitano di accedere ai metodi della superclasse Node, di cui le classi JavaFX Shape sono sottoclassi. Per consentire tale accesso senza ricorrere a cast espliciti, abbiamo aggiunto il metodo toJavaFXShape all’interfaccia ShapeInterface, permettendo così a ciascuna forma di restituire la propria rappresentazione equivalente in JavaFX.
* Durante l’implementazione ci siamo accorti che la complessità del sistema sta gradualmente aumentando. In particolare, al clic del mouse nello spazio di lavoro, devono essere eseguiti diversi controlli, che variano in base allo stato corrente dell’applicazione. Considerando l’incremento delle funzionalità previsto nelle prossime Sprint, potrebbe rendersi necessario un refactoring del controller. In tal caso, l’introduzione del design pattern State potrebbe rappresentare una soluzione efficace per gestire in modo più ordinato i diversi comportamenti associati agli stati.

Eventuali bug riscontrati durante il test sono stati risolti.

1. **Product Backlog issues**

Il product Backlog sviluppato chiariva sufficientemente i requisiti della nostra applicazione, con i relativi criteri di accettazioni. Tuttavia, la User Story 17, ovvero **Operazione annulla** peccava di un criterio di accettazione necessario per testare correttamente il funzionamento dell’applicazione. Alla luce del fatto che l’annullamento non deve essere previsto se non ci sono state modifiche annullabili, è stato successivamente aggiunto il seguente criterio:

* Dato che l’utente ha avviato il programma, quando non ci sono modifiche annullabili nello spazio di lavoro, allora il pulsante ‘Annulla’ risulta disabilitato.