Documento dei Requisiti del Sistema Gestore Disegni 2D

Matteo Gruppi (243420)

Ion Postica (241194)

|  |  |  |  |
| --- | --- | --- | --- |
| Versione | Data | Descrizione | Autore |
| 1.1 | 25-10-2016 |  | Matteo Gruppi |
| 1.2 | 22-11-2016 | Inserito paragrafo 5.2,1 | Ion Postica |
| 1.3 | 20-09-2017 | Aggiunte funzioni | Ion Postica |

1. Introduzione

Lo scopo di questo documento è di raccogliere, analizzare e definire le esigenze ad alto livello di astrazione degli utenti e delle feature di sistema di 2DesignManager.

Il sistema che si andrà a sviluppare è un applicazione per la gestione di disegni 2D. Il sistema prevede una interfaccia, realizzata utilizzando la libreria grafica Java Swing, che dialoghi con l’utente, che gli permetta di caricare, modificare e/o creare un disegno 2D.

2. Glossario

* **Java Swing**: framework utilizzato per la gestione della interfaccia grafica del programma
* **Figura semplice**: Qualsiasi figura piana, quindi rettangoli, triangoli cerchi e via discorrendo.
* **Figura composta**: Insieme disposto ordinatamente di numero variabile di figure semplici
* **Grid**: griglia da cui si può individuare facilmente la posizione e le misure delle figure nel disegno.
* **DataBase**: Archivio dove vengono salvate le Figure insieme a tutti i loro dettagli.

3. Gli Utenti e i Sistemi Esterni

Vi è un unico tipo di utente, come un’unica modalità di utilizzo. Questo poiché l’utilizzatore ha pieno accesso a tutte le funzioni integrate nel progetto 2DesignManager.

//Non vi è, momentaneamente, alcun sistema esterno che possa interagire con il progetto, bensì è facilmente deducibile come sia necessario che il progetto interagisca con altri sistemi.//

**3.1 Descrizione degli Utenti**

* Attore 1: Sono gli utenti utilizzatori

**3.3 Contesto d'uso**

Il sistema sarà destinato ad un utilizzo amatoriale di gestione grafica.

**3.4 Esigenze**

Ciascuno degli utenti e dei sistemi esterni individuati ha varie esigenze da soddisfare. Tra le più importanti:

* Attore 1: Use Case X, Use Case Y, …
* Attore 2: Use Case Z, Use Case Y …
* …

4. Il sistema

Il sistema ha lo scopo di fornire un’interfaccia per la gestione dei disegni 2D. L’utente deve poter creare un disegno 2D partendo da zero o caricare una base dal sistema. Il disegno 2D contiene un insieme di figure. Una figura può essere semplice o composta. La figura semplice è un cerchio, un quadrato, un rettangolo, un rombo, un triangolo o una delle altre figure geometriche piane. Una figura composta è l’insieme di altre figure geometriche semplici. Infine l’utente potrà salvare il disegno in versioni successive. Ogni versione di un disegno viene salvato in un file contenuto in una cartella specifica del sitema.

**4.1 Caratteristiche**

* Il sistema sarà utilizzato da
* Lo scopo del sistema è completa gestione di figure in due dimensioni
* Il sistema appartiene alla categoria delle applicazioni di ambito Educativo
* I benefici derivanti dal suo uso sono principalmente:
  + Facile fruibilità di gestione di figure piane
  + Metodo intuitivo di salvataggio
  + Storico dei cambiamenti eseguiti

5. Funzionalità del sistema

In questa sezione vengono descritte le funzionalità scoperte con associati gli attributi che le riguardano.

**5.1 Attributi delle funzionalità**

Di seguito si presenta una tabella con la descrizione degli attributi usati per caratterizzare le funzionalità.

* Id: l’identificatore delle funzionalità
* Nome: il nome della funzionalità
* Priorità: l'importanza della funzionalità: utile, importante, critica
* Sforzo: Lo sforzo richiesto per l'implementazione: basso, medio, alto
* Rischio: Le probabilità che qualcosa vada male nello sviluppo: basso, medio, alto

**5.2 Elenco delle funzionalità**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Nome** | **Priorità** | **Sforzo** | **Rischio** |
| **FF1** | **Add Figure** | **Alta** | **Medio-Alto** | **Medio-Alto** |
| Il sistema fa scegliere all’utente i dati della nuova Figura semplice da creare. L’utente dovrà assegnare un nome alla nuova figura semplice e il numero dei suoi lati, numero superiore a 2. | | | | |
| **FF2** | **Remove Figure** | **Alta** | **Medio-Alto** | **Medio-Alto** |
| Il sistema rimuove dal Canvas la Figura che è selezionata nel elenco figure. | | | | |
| **FF3** | **Save Figure** | **Alta** | **Alto** | **Medio-Alto** |
| Il sistema salva la Figura selezionata nel elenco figure. Chiede all’utente se vuole aggiungere una nota da associare alla Figura e successivamente salva tutto nel DataBase. | | | | |
| **FF4** | **DataBase** | **Alta** | **Alto** | **Alto** |
| Apre una nuova finestra contenente l’elenco delle figure salvate con il loro ID, la loro versione, classe, numero di campi, nome file e descrizione. | | | | |
| **FF5** | **Load** | **Alta** | **Alto** | **Alto** |
| Funzionalità presente nella finestra DataBase che consente di caricare la Figura selezionata con tutti i suoi dettagli. | | | | |
| **FF6** | **Remove** | **Alta** | **Alto** | **Alto** |
| Funzionalità presente nella finestra DataBase che consente di rimuovere completamente dal DataBase la Figura. | | | | |
| **FF7** | **Choose Color** | **Media-Bassa** | **Medio** | **Medio-Alto** |
| Questa funzionalità consente di cambiare il colore della Figura selezionata. Nel caso la Figura selezionata è di tipo composta e la spunta “Aggregate Color” è selezionata, verrà cambiato il colore a tutte le figure semplici presenti nella figura composta. | | | | |
| **FF8** | **Composite Figure** | **Alta** | **Alto** | **Alto** |
| Questa funzionalità consente di creare Figure Composte. Per creare una figura composta bisogna selezionare, tenendo premuto il tasto Maiuscolo, le figure che andranno a comporre la figura composta nel menu delle figure. | | | | |
| **FF9** | **Move** | **Media** | **Medio-Alto** | **Medio** |
| Permette di muovere/trascinare all’interno del canvas la Figura che è selezionata nell’elenco Figure. | | | | |
| **FF10** | **Resize** | **Media** | **Alto** | **Alto** |
| Permette di ridimensionare all’interno del canvas la Figura che è selezionata nell’elenco Figure, trascinando il mouse all’interno della Figura. | | | | |
| **FF11** | **Rotate** | **Media** | **Medio** | **Medio** |
| Consente di ruotare la Figura selezionata nel elenco delle figure. | | | | |
| **FF12** | **ZeroDegree** | **Media** | **Medio** | **Medio** |
| Imposta la rotazione della figura a zero gradi, ossia nella posizione iniziale di creazione della figura. | | | | |
| **FF13** | **SetX** | **Media** | **Medio** | **Medio** |
| Imposta la coordinata della figura lungo l’asse x del canvas. | | | | |
| **FF14** | **SetY** | **Media** | **Medio** | **Medio** |
| Imposta la coordinata della figura lungo l’asse y del canvas. | | | | |
| **FF15** | **Grid** | **Bassa** | **Basso** | **Basso** |
| Rende visibile il sistema di misura nel canvas. | | | | |

6. Requisiti Non Funzionali

I Requisiti non funzionali non sono collegati direttamente con le funzioni implementate dal sistema, ma piuttosto alle modalità operative e di gestione. Di seguito si evidenzieranno i vincoli a cui il sistema si dovrà attenere.

**6.1 Requisiti di Prodotto**

6.1.1 Requisiti di Usabilità

Per poter usare il software bisogna aver installato sulla macchina Java, scaricandolo dal sito ufficiale <https://www.java.com/it/download/> .  
Dopodiché basta fare doppio-click sull'icona del file “2DesignManager.jar”.

6.1.2 Requisiti di Efficienza

Per usare il prodotto in modo efficiente si consiglia di disporre di una macchina con i seguenti requisiti:

Requisiti minimi:

Processore: 1 gigahertz (GHz) o più

RAM: 1 gigabyte (GB) per 32-bit o 2 GB per 64-bit

Disco fisso: 200 MB per 32-bit OS, 400 MB per 64-bit OS

6.1.3 Requisiti di Affidabilità

Non ci sono particolari requisiti di affidabilità, eccetto quello, per il software, rispondere in modo reattivo alle azioni, tramite mouse e tastiera, dell'utente.

6.1.4 Requisiti di Portabilità

Il programma può essere eseguito su qualsiasi sistema operativo a patto che vengano soddisfatti i requisiti di usabilità e di efficienza.

**6.2 Organizzativi**

6.2.1 Requisiti di consegna

Il progetto dovrà contenere il documento della specifica dei requisiti, il documento di progettazione ed infine il software assieme un manuale utente che descriva le procedure di installazione e di uso del sistema.

6.2.2 Requisiti di implementazione

Per l’implementazione del programma è necessario utilizzare il linguaggio Java e la libreria grafica Swing: <https://en.wikipedia.org/wiki/Swing_(Java)>

7. Casi d'uso

In questa sezione vengono descritti i casi d'uso che descrivono il comportamento del sistema e si presenta la vista d’insieme del sistema attraverso un diagramma dei casi di uso (alcuni diagrammi dei casi di uso). La priorità dei diversi casi di uso è individuata dai valori: basso, medio e alto.

**7.1 Elenco dei Casi di uso e Vista d'insieme**

|  |  |  |
| --- | --- | --- |
| **Id** | **Nome** | **Priorità** |
| UC1 | ... | … |
| UC2 | … | … |

**7.2 Diagramma Principale dei Casi di Uso**

**7.3 Nome Use Case 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Versione | Data | | Autore |
| 1.0 | 25-10-2016 | |  |
| Goal | |  | |
| Attori | |  | |
| Precondizioni | |  | |
| Trigger | |  | |
| Scenario Principale | | | |
| Step 1 | |  | |
| Scenario Alternativo 1 | | | |
| Step 1 | |  | |
| Scenario Eccezione 1 | | | |
| Step 1 | |  | |
| Postcondizioni | |  | |
| Use Case Richiamati | |  | |
| Note | |  | |

**Diagramma dei casi di uso**

**Diagrammi di sequenza**

8. Modello del Sistema

Il sistema può essere descritto … (Uno o più diagrammi di classe ed eventuali diagrammi di stato)