Documento dei Requisiti del Sistema Gestore Disegni 2D

Matteo Gruppi (243420)

Ion Postica (241194)

|  |  |  |  |
| --- | --- | --- | --- |
| Versione | Data | Descrizione | Autore |
| 1.1 | 25-10-2016 |  |  |
| 1.2 | 22-11-2016 | Inserito 5.2,1 | Ion Postica |
| 1,3 |  |  |  |

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

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
* Attore 2: Sono … Useranno il sistema principalmente per …
* …

**3.2 Descrizione dei Sistemi Esterni**

* Attore 1: Sono … Useranno il sistema principalmente per …
* Attore 2: Sono … Useranno il sistema principalmente per …
* …

**3.3 Contesto d'uso**

Il sistema sarà usato per un amatore al primo approccio 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 | Caricamento Disegno 2D | Alta | Medio-Alto | Medio-Alto |
| Il sistema apre una finestra in cui verranno elencati i disegni presenti sul database. L’utente una volta selezionato il disegno desiderato dovrà scegliere la versione del disegno. | | | | |
| FF2 | Inserimento nuova figura semplice | Alta | Alto | Medio |
| L’utente dovrà assegnare un nome alla nuova figura semplice e può anche inserire una descrizione.  E’ possibile importare delle figure semplici da altri disegni. | | | | |
| FF3 | Gestione figure | Media | Alto | Basso |
| Ogni figura, semplice oppure composta può essere ruotata, rimpicciolita, ingrandita o traslata. Si può assegnare un colore alla figura semplice. In una figura composta si può mantenere i colori delle sue parti o assegnare un colore unico. | | | | |
| FF4 | Salvataggio Disegno 2D | Medio-Alta | Medio-Alto | Medio-Alto |
| Ogni disegno può essere salvato in versioni successive. Ogni versione di un disegno viene salvato in un file contenuto in una cartella specifica del sistema. Inoltre, i dati della annotazione del disegno e queli delle eventuali annotazioni delle sue figure sono salvati anche su un database. | | | | |

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

…

6.1.2 Requisiti di Efficienza

…

6.1.3 Requisiti di Affidabilità

…

6.1.4 Requisiti di Portabilità

…

**6.2 Organizzativi**

6.2.1 Requisiti di consegna

…

6.2.2 Requisiti di implementazione

…

6.2.3 Requisiti sugli Standard

…

**6.3 Esterni**

6.3.1 Requisiti di Interoperabilità

…

6.3.2 Requisiti Etici

…

6.3.3 Requisiti Legislativi

…

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**

7. 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

…

6.1.2 Requisiti di Efficienza

…

6.1.3 Requisiti di Affidabilità

…

6.1.4 Requisiti di Portabilità

…

**6.2 Organizzativi**

6.2.1 Requisiti di consegna

…

6.2.2 Requisiti di implementazione

…

6.2.3 Requisiti sugli Standard

…

**6.3 Esterni**

6.3.1 Requisiti di Interoperabilità

…

6.3.2 Requisiti Etici

…

6.3.3 Requisiti Legislativi

…

8. Modello del Sistema

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