



Verbali Esterni

Gruppo SWEetBIT — Progetto SWEDesigner

Informazioni sul documento

Versione	1.0.0
Redazione	Sebastiano Bertolin
Verifica	Da inserire
Approvazione	Da inserire
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo SWEetBIT

Descrizione

Questo documento traccia i verbali di tutte le riunioni esterne, con il committente od il proponente, svolte dal gruppo SWEetBIT.

Indice

1 Riunione 1	2
1.1 Informazioni sulla riunione	2
1.2 Domande e Risposte	2

1 Riunione 1

1.1 Informazioni sulla riunione

- **Data:** 23/02/2017
- **Luogo:** Zuccheti - sede di Padova , via Giovanni Cittadella 7
- **Ora:** 16:30
- **Durata:** 90 min
- **Argomento:** Chiarimenti dei requisiti sul capitolato
- **Partecipanti Interni:** Davide Santimaria - Fabio Massignan - Gianmarco Salmistraro - Malick Bodian - Salvatore Pilò - Sebastiano Bertolin
- **Partecipanti Esterni:** Gregorio Piccoli

1.2 Domande e Risposte

- **Come deve essere la qualità del codice generato dai diagrammi UML ?**
Per rispondere al meglio alla domanda, ci si vuole da prima focalizzare su un altro punto, ovvero il perché sia stato suggerito il tema dei giochi da tavolo su questo capitolato.
Ebbene nel dominio dei giochi da tavolo, la ripetitività è molto elevata; Basti pensare al gioco della dama, degli scacchi ad esempio, entrambi condividono la stessa scacchiera di gioco, anche il Monopoli ha una scacchiera, che si diversifica per colori e tipi di caselle , ma è pur sempre una scacchiera. Focalizzarsi su di un specifico ambito, in questo caso i giochi da tavolo, aiuta a migliorare la qualità del codice (vista anche la possibilità del suo riutilizzo) , in quanto se si decidesse di rappresentare ogni contesto, risulterebbe difficile generare del codice che si adatti al meglio in ogni situazione.
La vera difficoltà sta nel generare codice dai diagrammi che rappresenteranno i metodi. Si suggerisce un approccio utilizzando l'activity diagram.
- **Nel nostro designer dobbiamo includere dei template?**
Si, a patto che venga scelto un dominio su cui il designer si basi.
Dare la possibilità di avere ad esempio già una scacchiera 8x8, evitando che l'utente finale debba crearsela da zero. Ovviamente i giochi da tavolo non son l'unico dominio su cui ci si può basare, la scelta potrebbe ricadere in altri settori, ma tale scelta deve esser fatta per poter avere dei diagrammi che generino codice.
Il settore fin ora citato è consigliato perchè offre molti approfondimenti in merito all'adattamento in diagrammi ed inoltre garantirà una fase di testing più piacevole.

- **Se viene modificato il codice generato, il diagramma deve aggiornarsi anch'esso?**

È un aspetto sicuramente interessante se si riuscisse ad implementare. Solitamente dopo la creazione dei diagrammi ed il rispettivo codice, le successive modifiche vengono apportate solo al codice, in quanto il diagramma ha lo scopo principale da fungere da linea guida, rappresentando solo le classi principali, tralasciando nella visualizzazione quelle di supporto. Rappresentare tutte le classi che costituiscono il progetto, potrebbe ridurre la leggibilità del diagramma; una soluzione potrebbe essere quella di celare o inserire in un layer diverso le classi di supporto.

Visualizzare solo lo scheletro della classe senza l'implementazione dei metodi può esser una soluzione alla domanda posta.

- **L'applicazione deve essere solo desktop o deve essere anche una web-application?**

È preferita la web-application, ma la scelta non è vincolante. Lo scopo principale è entrare nell'ottica di fare un progetto usando molto i diagrammi.

- **Il codice prodotto deve esser in formato Java o JavaScript_G?**

Nella fase di disegno dei diagrammi, in particolare a quelli dei metodi_G, si deve procedere in modo astratto, ovvero tracciando solo l'algoritmo necessario al metodo in esame. Procedendo con quest'ottica si può generare codice in entrambi i linguaggi. Proseguire in quest'ottica di pensare per specifiche risulta interessante ed una buona sfida, ma se si decidesse di pensare per programma, il consiglio è quello di procedere inizialmente con Java, il quale è un linguaggio più controllato e verificato.

- **L'applicazione dovrà esser disponibile su di un server oppure è sufficiente in locale?**

Non è necessario l'acquisto di uno spazio su cui ospitare l'applicazione, eventualmente si possono utilizzare dei server gratuiti a tempo limitato come ad esempio Amazon, Heroku.

- **Considerando che il numero di gruppi che aderiscono a tale capitolato è aumentato, alcuni dei requisiti opzionali son diventati obbligatori?**

I requisiti opzionali rimangono invariati, un aspetto su cui focalizzarsi è lo studio dell'UML. Il progetto consiste nel creare un disegnatore che abbia il diagramma della classe con i rispettivi diagrammi dei metodi; la soluzione di come rappresentare il collegamento tra questi diagrammi, è sicuramente il punto su cui ci si deve concentrare, proponendo anche modelli ibridi dei diagrammi.