

Introduzione

Informazioni sul progetto

- Docenti responsabili al controllo del lavoro: Luca Muggiasca e Geo Petrini
- Allievo responsabile della realizzazione dell'intero progetto: Jure Grgic
- Luogo e materia ove si svolge il progetto: Scuola d'arti e mestieri di Trevano, nella classe I3AC, durante il modulo 306
- Inizio del progetto: venerdì 6 Settembre 2019
- Scadenza del progetto: venerdì 20 Dicembre 2019
- Nome del progetto assegnato: Fiocchi di neve

Abstract

È stato creato un applicativo denominato "Fiocco di neve" che, quando viene eseguito, il programma presenta un frame dove è possibile fare dei tagli ad un triangolo, in modo da poter creare e mostrare un fiocco di neve utilizzando il triangolo appena tagliato. In seguito viene convertita una copia del fiocco generato, in un formato PNG o SVG. Infine viene inserito in un sito appositamente creato per questo, tutti i requisiti prestabiliti, con il link per installare il file .jar dell'applicativo "Fiocco di neve" ed infine una serie di screenshot che hanno il ruolo di una mini guida per poter informare gli utenti su come utilizzare il programma. L'applicativo è stato sviluppato con il linguaggio di java.

Scopo

Lo scopo del progetto è apprendere le conoscenze tecniche per portare avanti un progetto individualmente. Oltretutto serve per poterci ambientare già nel mondo del lavoro e per far sì che in caso di difficoltà, proviamo a trovare una soluzione da soli.

Analisi

Analisi del dominio

La scuola d'arti e mestieri di Trevano è l'ambiente di lavoro dove verrà svolto il progetto. Il programma deve essere svolto con il linguaggio java o javascript. È stato utilizzato un mac con un virtual machine per poter sviluppare l'applicativo. Non c'è stata alcuna informazione riguardo altri programmatori che hanno sviluppato questo programma, ma probabilmente vista la bassa difficoltà del programma, altri programmatori avranno già provato a creare un codice simile. L'utente che andrà ad utilizzare il seguente applicativo non necessita di nessuna specifica tecnica avanzata, necessiterà magari di una piccola guida che aiuti l'utente ad orientarsi. A differenza dell'utente, il programmatore che deve sviluppare il programma, è obbligato a conoscere delle determinate nozioni di java o javascript per poter fare il proprio lavoro. Come ogni progetto, si hanno dei requisiti da rispettare per poter svolgere un buon lavoro, e questi requisiti variano a seconda di cosa vuole il cliente.

Analisi e specifica dei requisiti

ID	Nome	Priorità	Vers	Note
Req-1	Il software deve essere scritto in JAVA	livello 1	1.0	-
Req-2	Deve esistere sito con descrizione software	livello 1	1.0	-
Req-3	Deve essere possibile scaricare il software	livello 1	1.0	-
Req-4	I tagli del triangolo devono poter essere fatti con il mouse	livello 1	1.0	-
Req-5	Deve esserci un'interfaccia grafica	livello 1	1.0	-
Req-6	L'area di lavoro deve essere un triangolo	livello 1	1.0	-
Req-7	Deve essere possibile resettare i punti registrati	livello 1	1.0	-
Req-8	Deve esistere il tasto genera fiocco	livello 1	1.0	-
Req-9	La generazione deve avvenire in tempo reale	livello 2	1.0	-
Req-10	I punti devono poter essere spostati o rimossi	livello 2	1.0	-
Req-11	Il lavoro deve poter essere salvato	livello 1	1.0	-
Req-12	Il salvataggio deve essere fatto in formato PNG o SVG	livello 1	1.0	-
Req-13	Il salvataggio deve avere dimensioni definite dall'utente	livello 1	1.0	-
Req-14	Deve essere possibile salvare i punti di taglio	livello 1	1.0	-

Spiegazione elementi tabella dei requisiti:

ID: identificativo univoco del requisito

Nome: breve descrizione del requisito

Priorità: indica l'importanza di un requisito nell'insieme del progetto, definita assieme al committente. Ad esempio poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

Versione: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Note: eventuali osservazioni importanti o riferimenti ad altri requisiti.

Sotto requisiti: elementi che compongono il requisito.³

Use case

utente -> Apertura dell'applicativo

utente -> Caricamento -> Triangolo visibile -> Interazione con l'applicativo

utente -> Fine ritaglio -> Conferma forma -> Visualizzazione risultato
utente -> Fine o riavvio dell'applicativo

Pianificazione

La pianificazione del progetto è stata fatta con un diagramma di Gantt. Ecco in ordine gli incarichi da svolgere:

- Analisi - Analisi dei requisiti per l'applicativo da sviluppare - Formulazione di domande al cliente per possibili fraintendimenti dei requisiti - Preparazione dell'ambiente di lavoro - Creazione del Gantt - Creazione di una repository dove tenere tutti i file - Sistemazione delle cartelle e dei file all'interno della repository - Creazione di una macchina virtuale - Installazione di Github Desktop nella macchina virtuale - Configurazione di Github Desktop con Github di rete - Sviluppo dell'applicativo - Creazione del JPanel - Creazione dell'interfaccia grafica
- Gestione dei puntini - Gestione triangolo tagliato - Gestione formato del fiocco - Test dell'applicativo (svolto in contemporanea con lo sviluppo dell'applicativo) - Integrazione - Creazione del sito - Integrazione di tutti i file sul sito

Infine gli ultimi due incarichi sono la documentazione e i diari, questi due incarichi a differenza degli altri, iniziano e finiscono in contemporanea con il progetto in generale, siccome essi vengono appunto aggiornati ogni volta che il progetto è stato modificato o ampliato, rimanendo attivi per tutta la durata del progetto.

Analisi dei mezzi

Il progetto è stato svolto con la versione 10.12.5 del macOS Sierra, appartenente all'aula 422, della scuola d'arti e mestieri di Trevano e con la versione 10.13.6 del macOS High Sierra, il mio computer personale. Per salvare i file è stato utilizzato come "magazzino" Github e Github Desktop. Per sviluppare il programma invece è stato utilizzato l'editor NetBeans versione 11.1. Il Gantt è stato creato grazie alla versione 1.9.1 di ProjectLibre.

Progettazione

Design dell'architettura del sistema

Dovranno essere presenti dei metodi che permettano la creazione dei puntini che andranno poi a formare tra loro i poligoni di ritaglio. Bisognerà adattare la posizione di ogni oggetto presente (puntini, rettangolo base, poligoni di ritaglio) in base alla dimensione del frame, e per farlo, dovrò utilizzare dei calcoli matematici con percentuali. I puntini creati verranno salvati in un apposita lista. Servirà un metodo per controllare lo stato dei bottoni. Un altro metodo servirà invece per controllare se viene creato un poligono. È necessario gestire il salvataggio dell'immagine del fiocco di neve, quindi sarà presente anche un metodo per questo. Inoltre bisognerà controllare con un altro metodo, se il mouse interagisce con un puntino già creato o no, per farlo potrebbero servire i metodi del MouseMotion e del MouseMotionListener. Infine un metodo che servirà per resettare tutti i puntini creati.

Diagramma delle classi e descrizione.

alt text

La classe principale (FioccoDiNeve), conterrà buona parte del mio codice. Per poter funzionare però dovrò implementare MouseListener e MouseMotionListener e forse anche altro a dipendenza delle necessità. Inoltre

avrò bisogno di estendere il JPanel, in modo da poter disegnare e passare il paint al frame (FioccoDiNeveFrame). Infine è presente la classe PuntoDouble, che è un aggregante della classe principale (FioccoDiNeve).

Design delle interfacce

L'interfaccia iniziale dovrebbe presentare il triangolo base da ritagliare al centro del frame. Ci deve essere un'area dove poter creare i poligoni di ritaglio (quest'area può essere distinta grazie ad un colore diverso rispetto all'area esterna ad essa). Nell'area esterna saranno presenti i pulsanti che permetteranno di fare determinate azioni, nella parte bassa saranno presenti i pulsanti di reset e generazione del fiocco di neve, a destra i pulsanti per salvar e caricare i puntini e a sinistra i pulsanti per tagliare e incollare i puntini di ritaglio. Nell'area esterna alta ci sarà il nome del programma.

alt text

Diagramma di flusso

alt text

Implementazione

Il tutto parte dalla classe FioccoDiNeve.java, questa classe utilizza FioccoDiNeveFrame.java come interfaccia grafica, dove appunto verrà disegnato il fiocco di neve. La classe FioccoDiNeve.java estende il JPanel, che permette appunto di passare a FioccoDiNeveFrame.java, grazie al metodo paintComponent; inoltre implementa MouseListener e MouseMotionListener, necessari per poter interagire con l'applicazione. I metodi di MouseListener e MouseMotionListener che vengono utilizzati sono mouseClicked e mouseDragged. Sono presenti tre liste per il salvataggio dei punti/poligoni di ritaglio: - la lista "percentages" che si occupa di salvare la posizione in percentuale dei puntini dell'attuale poligono che si sta creando, ciò viene permesso grazie ad una classe esterna: PuntoDouble.java - la lista "listeDiPunti" che contiene tutte le liste percentages create (che formano appunto un poligono di ritaglio) - la lista "polygons", dove vengono presi tutti i punti creati e uniti per formare i vari poligoni, per poi salvare qui dentro. Inoltre sono presenti altri metodi: - Il metodo "generateSnowFlake", è un semplice metodo per la generazione del fiocco di neve - Il metodo "clickedArea", anch'esso è un semplice metodo che controlla se il mouse clicca all'interno dell'area determinata (area blu) - Il metodo "isOver" che controlla se il mouse ha cliccato un determinato puntino creato in precedenza - Il metodo "closedPolygon" che controlla se il mouse ha cliccato il puntino di origine di un poligono di ritaglio - Il metodo "resetLists", resetta tutte le liste, ovvero "percentages", "listeDiPunti" e "polygons" - il metodo "saveImage", serve per salvare il fiocco di neve in formato png - il metodo "isGenerated", che controlla se il fiocco è generato o meno

Test

Protocollo di test

controllo funzionalità punti dei poligoni di ritaglio - test 01 | creazione di un punto - test 02 | spostamento di un punto - test 03 | eliminazione di un punto

controllo responsive punti, triangolo base, poligoni e fiocco di neve - test 04 | resize dei puntini - test 05 | resize dei poligoni di ritaglio - test 06 | resize del triangolo di base - test 07 | resize del fiocco di neve

controllo bottoni - test 08 | bottone salvataggio del fiocco di neve - test 09 | bottone reset dei punti - test 10 | bottone genera fiocco di neve

Risultati test

risultati controllo funzionalità punti dei poligoni di ritaglio - esito test 01 | funzionante: il punto viene creato con il click destro del mouse - esito test 02 | funzionante: il punto viene spostato trascinando il punto schiacciando la rotellina centrale del mouse - esito test 03 | funzionante: il punto viene eliminato con il click sinistro del mouse

risultati controllo responsive punti, triangolo base, poligoni e fiocco di neve - esito test 04 | funzionante: i puntini creati si spostano in maniera corretta, a dipendenza della dimensione della schermata - esito test 05 | funzionante: i poligoni di ritaglio si ridimensionano correttamente - esito test 06 | funzionante: il triangolo di base si ridimensiona correttamente - esito test 07 | non funzionante: il fiocco di neve non possiede un responsive corretto, solo alle dimensioni 1024x768 ha la giusta forma

risultati controllo salvataggio - esito test 08 | funzionante: è presente un bottone che permette il salvataggio in formato png del fiocco di neve - esito test 09 | funzionante: è presente un bottone che permette di resettare tutti i poligoni di ritaglio - esito test 10 | funzionante: è presente un bottone che permette di generare il fiocco di neve

Mancanze/limitazioni conosciute

Non sono presenti i pulsanti per il salvataggio dei punti correnti e per il caricamento di un file di punti. Non si può tagliare e incollare un puntino, solo eliminarlo, spostarlo o crearne uno nuovo. Non è presente un anteprima del fiocco di neve generato. Una volta che un poligono di ritaglio è stato generato, non si può più modificare la sua struttura. Se è presente un poligono di ritaglio incompleto mentre si schiaccia il pulsante "generate", esso non verrà preso in considerazione. Il fiocco di neve generato non è responsive, è corretto solo se messo alle dimensioni minime consentite (1024x768), per coprire questa mancanza ho settato a 1024x768 e fatto in modo che non si potesse ridimensionare la schermata.

Consuntivo

alt text

Conclusioni

Sviluppi futuri

- Si potrebbe far decidere all'utente il numero di spicchi del fiocco di neve
- Dar la possibilità all'utente di scegliere il colore del fiocco di neve
- Dare più possibilità di scelta del triangolo base (nel senso che non per forza deve essere un triangolo, ma potrebbe essere anche un quadrato o un cerchio, di conseguenza la forma finale logicamente può

variare rispetto al fiocco di neve originale)

- Usare il fiocco di neve come tiro al bersaglio
- Creare il movimento di più fiocchi di neve che cadono dall'alto al basso, come se stesse nevicando

Considerazioni personali

Questo è stato il mio primo progetto serio che abbia mai fatto. Come primo progetto è stato parecchio difficile e noioso, però allo stesso tempo soddisfacente. Ho capito che fare un progetto così "grande" da soli non è per niente semplice, preferisco fare un lavoro di gruppo in questo caso. Magari il fatto di aver trovato noioso questo progetto è dovuto al tipo di progetto che ci è stato assegnato (fiocco di neve). Però, pur se avrei preferito lavorare con un gruppo in questo caso, non mi piacciono tanto i lavori di gruppo.

Mie opinioni

Voto di realizzazione del programma (come io l'ho realizzato): 6.

Mi sono limitato a fare le basi del programma, senza andare a implementare tutti i dettagli, come per esempio il responsive per il fiocco di neve oppure il pulsante di salvataggio dei punti.

Voto documentazione: 8.

Credo di aver svolto una buona documentazione.

Voto del progetto (cioè quanto mi è piaciuto): 5.

Non è stato proprio divertente o interessante, ma neanche così brutto.

Sitografia

[Pagina web del Fiocco di neve](htt