Documento dei requisiti

PIETRO VALENTE

Prefazione

I lettori previsti per questo documento sono tutti coloro, persone o organizzazioni, che hanno in qualche modo a che fare col sistema ed hanno quindi un interesse legittimo.

In particolare, i responsabili del cliente (client managers), gli utilizzatori finali del sistema, gli ingegneri del cliente, i dirigenti appaltatori (contractor managers), gli architetti del sistema e soprattutto gli sviluppatori del software, sia della versione attuale, sia eventualmente futura.

Per visualizzare un riepilogo della discussione dei requisiti fare riferimento alla sezione *"Appendice"*.

Versione del software: 1.0.

Sommario

Introduzione

Il programma simula la disposizione dei pezzi sulla griglia di gioco e il suo compito è quello di calcolare e fornire su richiesta dell'utente:

1. Il numero di pezzi presenti sulla mappa per ciascuna tipologia
2. La casella con il maggior valore di difesa di giorno
3. La casella con il maggior valore di difesa di notte
4. La casella con il maggior valore di attacco di giorno
5. La casella con il maggior valore di attacco di notte
6. La casella con il maggior numero di pezzi dello stesso tipo

Il sistema necessita di un file di input scritto correttamente, la cui struttura è specificata nel requisito "Software file di input", e della Java Virtual Machine di versione 1.7, essendo scritto in linguaggio java.

Il sistema, ad oggi, è stato progettato per funzionare autonomamente, senza l'ausilio o la connessione ad altri sistemi. L'obbiettivo di questo software sarà però quello di essere integrato con un gioco da tavolo in sviluppo che verrà poi commercializzato.

Glossario

File di input	File di formato ".txt" scritto dall'utente, contenente le informazioni relative ai pezzi di gioco.
Main	È il punto di inizio per l'esecuzione del programma ma anche la parte principale, in cui vengono riunite tutte le sotto-parti. In particolare, è una zona dove le varie sezioni interagiscono tra di loro.
Cartella principale	Cartella "Sorgenti", in cui sono presenti sia tutti i file sorgente, ovvero scritti in linguaggio programmazione; sia il file di input.
rFile	Numero delle righe della matrice di gioco.
nColonne	Numero delle colonne della matrice di gioco.
fileSc	È lo Scanner del file, ovvero lo strumento che ci permette di estrarre le informazioni.
valDef	Funzione che calcola il valore di difesa di una casella attenendosi ai valori di difesa dei pezzi presenti nel documento "Homework1" e ai possibili modificatori.
valAtt	Funzione che calcola il valore di attacco di una casella attenendosi ai valori di attacco dei pezzi presenti nel documento "Homework1" e ai possibili modificatori.
Matrice completa	Matrice con tutte le caratteristiche stabilite per dimensione e tipologia celle, con i pezzi di gioco inseriti.
Utente	Utilizzatore finale del sistema.

Definizione dei requisiti dell'utente

SOFTWARE FILE DI INPUT

FUNZIONE	Verificare l'esistenza e correttezza del file di input.
DESCRIZIONE	Controlla l'esistenza del file, verifica che sia scritto correttamente e memorizza alcune informazioni di questo.
INPUTS	Righe del file.
FONTE	File di input.
OUTPUTS	Informazioni sul file di input.
DESTINAZIONE	Utilizzo nel main.
AZIONE	Attraverso un ciclo viene richiesto all'utente il nome del file di input. Successivamente tramite confronti viene verificato che il file è scritto con triplette di righe X, Y, Z. X e Y, che rappresentano la riga (X) e la colonna (Y) del pezzo, devono essere numeri interi positivi e minori di 2147483647. Z rappresenta il tipo e deve essere una parola tra elfo, nano, orco. Se anche il controllo del file va a buon fine, vengono memorizzate informazioni relative al nome del file, nColonne, nRighe, fileSc.
PRE-CONDIZIONI	Nessuno, è il primo requisito del sistema.
POST-CONDIZIONI	Il file è effettivamente valido e le informazioni sono state registrate.
ECCEZIONI	Se il file non viene trovato oppure se il formato del file è sbagliato (quindi non ".txt"), si continua a chiedere il nome di questo all'utente, con la possibilità di terminare il programma. Se i dati di input nel file sono sbagliati viene segnalata la riga del problema e termina il programma. Se il file di input è vuoto, condizione valida, nRighe e nColonne vengono impostate ad 1 invece che a 0.

MATRICE SOFTWARE

FUNZIONE	Creare la matrice di gioco.
DESCRIZIONE	La matrice di gioco dovrà partire dalla cella (0,0) e la sua dimensione sarà determinata in modo da contenere tutte le caselle dove sono presenti pezzi. La dimensione dipenderà quindi dalla disposizione dei pezzi, in particolare sarà determinata dalla posizione del/dei pezzo/i con colonna e riga massima. Per quanto riguarda la tipologia delle celle, ogni colonna avrà celle dello stesso tipo e si alterneranno in sequenza colonne pianura/bosco/montagna.
INPUTS	nRighe e nColonne.
FONTE	Software file di input.
OUTPUTS	Fornire la matrice di gioco.
DESTINAZIONE	Utilizzo nel main.
AZIONE	Utilizzando i valori nRighe e nColonne viene creata la matrice di celle. Successivamente verrà assegnata ad ogni cella la sua tipologia.
PRE-CONDIZIONI	Software file di input è stato eseguito senza rilevare errori.
POST-CONDIZIONI	Viene creata la matrice con dimensione corretta e con le celle di tipologia stabilita.
ECCEZIONI	Se non ci sono pezzi, la dimensione della matrice sarà di 1 cella, non di 0, coerentemente alle informazioni fornite dal software file di input. Di conseguenza la cella sarà di tipologia pianura.

FUNZIONE RIEMPI-MATRICE

FUNZIONE	Inserire i pezzi nella matrice di gioco.
DESCRIZIONE	I pezzi saranno inseriti nella matrice alle coordinate specificate.
INPUTS	1. Matrice di gioco. 2. fileSc.
FONTE	1. Matrice software. 2. Software file di input.
OUTPUTS	Matrice contenente tutti i pezzi di gioco.
DESTINAZIONE	Software file di input.
AZIONE	Partendo dalla prima riga del file, per ogni tripletta incontrata, questa viene estratta attraverso fileSc. Si utilizzano le informazioni per inserire il pezzo nella matrice di gioco alla coordinata specificata nelle prime due righe X, Y e del tipo specificato nella riga Z. Per conoscere la tipologia di X, Y, Z fare riferimento a “Software file di input”. Essendo già stato eseguito il software file di input (requisito necessario per l’esecuzione di matrice software), sicuramente i dati estratti dal file saranno del formato corretto.
PRE-CONDIZIONI	Esecuzione di matrice software.
POST-CONDIZIONI	Matrice riempita con i pezzi di gioco, quindi completa.
ECCEZIONI	Se in una casella ci sono già 5 pezzi, viene segnalato e si salta l’inserimento del pezzo in questione.

RICHIESTE SOFTWARE

FUNZIONE	Fornire le informazioni previste dal software.
DESCRIZIONE	Viene fornito un menù in cui si può scegliere una delle seguenti informazioni da visualizzare, oppure digitare “exit” per far terminare il programma: 1. Il numero di pezzi presenti sulla mappa per ciascuna tipologia 2. La casella con il maggior valore di difesa di giorno 3. La casella con il maggior valore di difesa di notte 4. La casella con il maggior valore di attacco di giorno 5. La casella con il maggior valore di attacco di notte 6. La casella con il maggior numero di pezzi dello stesso tipo Una volta selezionata l’informazione, questa viene calcolata e fornita all’utente, che verrà reindirizzato al menù.
INPUTS	Matrice di gioco completa.
FONTE	Funzione riempi-matrice.
OUTPUTS	Fornire le informazioni richieste.
DESTINAZIONE	Utilizzo nel main.
AZIONE	Per calcolare le informazioni si utilizzeranno i dati presenti nelle celle della matrice e le funzioni valDef e valAtt.
PRE-CONDIZIONI	Esecuzione della funzione riempi-matrice.
POST-CONDIZIONI	Terminazione del programma.
ECCEZIONI	Per le richieste 2-6 se vi sono più caselle con lo stesso valore/numero, verranno mostrate tutte, se non ce n’è neanche una viene segnalato.

Architettura di sistema

Inizialmente il software file di input verifica l'esistenza e la correttezza dei dati, memorizzando informazioni tra cui il numero di colonne e righe che deve avere la matrice.

Con queste informazioni si procede con il matrice software, con cui viene creata la matrice della dimensione giusta e le celle vengono inizializzate della tipologia stabilita.

Successivamente viene utilizzata la funzione riempi-matrice che ha il compito di inserire tutti i pezzi nella matrice.

A questo punto, avendo la matrice completa, si può utilizzare il richieste software che interagirà con l'utente, fornendo le informazioni richieste.

Specificazione dei requisiti del sistema

Il programma è scritto in linguaggio java, quindi necessita della Java Virtual Machine di versione 1.7.

Il programma sarà strutturato in classi pubbliche, ognuna di queste presente in un file diverso, come stabilito dalle regole Java.

Le classi saranno: Main, Cella, Matrice, InputFIle, Richieste. Le ultime tre racchiudono i compiti specificati nei requisiti, la funzione riempi-matrice è stata implementata come metodo all'interno della classe InputFIle.

La classe matrice conterrà un array bidimensionale di classe "Cella". La classe Cella conterrà informazioni riguardo al numero di pezzi, elfi, nani e orchi e il riferimento ad un enum rappresentate la tipologia della cella. Inoltre, conterrà due metodi valDef e valAtt in grado di calcolare il valore di attacco/difesa della cella, passandogli come intero il riferimento temporale (se giorno 0, se notte 1).

Il Main sarà la classe effettivamente eseguibile ma rimarrà ridotta a livello di righe di codice per favorirne la leggibilità.

Modelli di sistema

Fare riferimento al documento allegato "Documento di architettura".

Evoluzione del sistema

È previsto un rilascio singolo definitivo della versione il 3 Maggio 2020, senza rilascio di aggiornamenti successivi.

Appendice

Cronologia interviste:

I requisiti sono stati discussi con i due committenti durante una videochiamata su Zoom il giorno 15/04/2020 e nei giorni successivi su un gruppo Whatsapp.

Inizialmente si è parlato del requisito legato al file di input. Entrambi i committenti sono stati d'accordo sulla struttura e sulla gestione di eventuali errori: il file deve essere scritto con triplette di righe, nelle prime due righe ci devono essere due numeri che identificano la posizione del pezzo, mentre nella terza la tipologia di pezzo che può essere solo una fra: "elfo, nano, orco". Se il file non rispetta questa struttura, il programma deve lanciare un messaggio d'errore e terminare la sua esecuzione. Il file deve essere di formato ".txt" e deve essere inserito nella cartella "Sorgenti", quella principale, se così non fosse lo si segnala viene chiesto di reinserire il nome del file.

Successivamente, un requisito discusso è stato relativamente alla griglia. Entrambi i committenti erano d'accordo sul far partire la griglia di gioco dalla casella (0,0). Si è poi discusso della dimensione che questa doveva avere, il commitente1 pensava fosse meglio richiederla ogni volta all'utente, mentre il committente2, pensava fosse meglio adattare la dimensione in base alla disposizione dei pezzi, per eliminare anche gli errori legati all'inserimento di un pezzo in una cella esterna alla griglia. Su consiglio anche del programmatore si è scelta la seconda opzione ed entrambi i committenti sono stati d'accordo anche sulla disposizione della tipologia delle celle: ogni colonna doveva avere la stessa tipologia e si sarebbero alternate colonne pianura, bosco, montagna.

Un altro requisito che è stato trattato riguarda l'inserimento dei pezzi nella griglia di gioco, in particolare, nella gestione di un inserimento in una cella che conteneva già altri 5 pezzi. Il committente2 avrebbe voluto che il programma segnalasse il problema e chiedesse all'utente di indicare un'altra cella in cui inserire il pezzo. Il committente1 pensava fosse meglio saltare l'inserimento per rendere più fluido il programma e poiché la perdita dell'inserimento non era poi così importante. Si è scelto infine per la soluzione del committente1.

L'ultimo requisito riguarda le richieste del programma. Nel caso in cui ci siano più caselle che soddisfano le richieste 2-6, il committente2 voleva che tutte le possibili caselle venissero segnalate, mentre il committente1 pensava fosse sufficiente segnalarne solo una di queste. Si è scelta la soluzione del committente2 perché più utile ai fini del gioco.

È stata sollecitato dal programmatore se fosse necessario creare un file ".txt" nella cartella principale per tenere i valori di attacco/difesa dei vari pezzi e anche dei modificatori, per rendere più facili le eventuali modifiche. I committenti hanno ritenuto questa operazione non necessaria, prevedendo di non modificare questi valori in futuro.

Per concludere si è parlato della finalità del software, che è stato progettato per ora indipendente da altri sistemi software, in quanto dovrà essere affiancato a un gioco da tavolo, ma in futuro i due committenti non escludono di rendere disponibile una versione completamente digitale.

Committente1: Andrea Seghetto

Committente2: Edoardo Barba