

Documento dei requisiti, versione 1.0

Prefazione

Questo documento e' stato redatto per il **Professore Migliardi Mauro** e per i **committenti** con i quali ho collaborato a questo progetto: **Guglielmin Niccolo' e Zaccarin Angelo**.

Attualmente il progetto e' alla sua prima versione: 1.0

Eventuali modifiche di una certa importanza, come l'introduzione di nuovi requisiti (dato che e' plan-driven ... speriamo di no), porteranno a nuove versioni.

Ogni volta che utilizzerò nel documento una parola potenzialmente non comprensibile inserirò il simbolo (G) per indicare che il suo significato e' chiarito nel glossario.

Introduzione

Sviluppatore: **Cappellotto Lorenzo, matricola 1188257**, corso ingegneria informatica generale.

Questo sistema e' nato dalla necessita' di valutare gli studenti del corso ingegneria del software attraverso progetti svolti a casa singolarmente.

Il sistema riguarda l'elaborazione e la stampa di dati relativi a un semplice gioco inventato dal prof. Migliardi.

Il sistema riceverà in input determinati dati e parametri. In seguito andrà a creare una mappa di gioco a forma di griglia e vi posizionerà determinate pedine come richiesto dai committenti. Ognuna di queste pedine avrà valori di attacco, difesa e moltiplicatori a seconda dell'ambiente(G) della casella su cui la pedina è posta. Infine, una volta costruita la mappa di gioco, il sistema andrà a stampare determinati output relativi alle caselle e alle pedine scelti dai committenti.

Glossario

Ambiente: tipologia di casella in cui il pezzo si trova e riferimento temporale giorno/notte

Definizione dei requisiti utente

Assieme ai committenti ho specificato i seguenti requisiti funzionali:

- 1 La mappa di gioco ha la forma di una griglia di forma rettangolare, divisa in $M \times N$ caselle (M numero colonne, N numero righe).
- 2 L'utente deve specificare, prima di ogni esecuzione, le dimensioni della griglia attraverso un'unica riga di testo all'interno del file "mappa_tipologie.txt" (vedi appendice-1 per esempio di file completo).
- 3 Deve essere presente all'interno del sistema un riferimento temporale. Esso può assumere i seguenti valori ["giorno", "notte"]. Il riferimento temporale non è assoluto, ma ogni calcolo deve essere eseguito dato un riferimento.
- 4 Ad ognuna delle caselle sarà associata una e una sola tipologia tra le seguenti ["pianura", "bosco", "montagna"].

5 L'utente puo' specificare, prima di ogni esecuzione, la tipologia di determinate caselle scelte attraverso gruppi di linee di testo all'interno del file "mappa_tipologie.txt" (vedi appendice-1 per esempio di file completo).

6 Per tutte le caselle non specificate dall'utente il sistema deve scegliere una tipologia in modo casuale tra quelle possibili.

7 Definisco pezzo un oggetto posizionabile all'interno della mappa. Ad ogni pezzo e' associata una delle seguenti tipologie ["orco", "elfo", "nano"].

8 Ad ogni tipologia di pezzo devono essere associati due valori, uno di attacco e uno di difesa determinati dall'utente. Inoltre ad ogni tipologia di pezzo e' associata una lista di modificatori di combattimento che dipende dalla tipologia di casella su cui il pezzo e' posto e il riferimento temporale scelto. In seguito ho inserito le due tabelle che descrivono i valori definiti dai due committenti.

TIPOLOGIA PEZZO	orco	elfo	nano
ATTACCO	4	5	2
DIFESA	4	2	5

TIPOLOGIA PEZZO				orco	elfo	nano
AMBIENTE	pianura	GIORNO	ATTACCO	-50%	0%	0%
			DIFESA	-50%	0%	0%
		NOTTE	ATTACCO	50%	0%	0%
			DIFESA	50%	0%	0%
	bosco	GIORNO	ATTACCO	-50%	100%	0%
			DIFESA	-50%	0%	0%
		NOTTE	ATTACCO	50%	100%	0%
			DIFESA	50%	0%	0%
	montagna	GIORNO	ATTACCO	-50%	0%	0%
			DIFESA	-50%	0%	100%
		NOTTE	ATTACCO	50%	0%	0%
			DIFESA	50%	0%	100%

9 L'utente deve poter inserire, prima di ogni esecuzione, determinati pezzi all'interno della griglia che forma la mappa attraverso gruppi di linee di testo all'interno del file "mappa_pezzi.txt" (vedi appendice-2 per esempio di file completo).

10 Deve essere possibile inserire fino ad un massimo di 5 per casella.

11 Il sistema deve calcolare e stampare a terminale il numero di pezzi presenti sulla mappa per ciascuna tipologia e il totale nel formato "Totale:#totale, Orchi:#orchi, Nani:#nani, Elfi:#Elfi".

12 Il sistema deve calcolare e stampare le coordinate della casella con il maggior valore di difesa dato un riferimento temporale (giorno o notte), in caso vi siano piu' caselle verra' stampata solo la prima occorrenza.

13 Il sistema deve calcolare e stampare le coordinate della casella con il maggior valore di attacco dato un riferimento temporale (giorno o notte), in caso vi siano piu' caselle verra' stampata solo la prima occorrenza.

14 Il sistema deve calcolare e stampare le coordinate della casella con il maggior numero di pezzi dello stesso tipo, in caso vi siano piu' caselle verra' stampata solo la prima occorrenza.

15 Il sistema deve stampare una matrice di caratteri che rappresenti la tipologie di tutte le caselle della mappa. I caratteri vengono cosi formattati: ['P':"pianura", 'B':"bosco", 'M':"montagna"].

16 Il sistema deve eseguire il controllo degli errori e stampare un errore a video in caso se ne riscontri uno.

17 Il sistema deve verificare la presenza dei file di input prima di ogni esecuzione.

18 Il sistema deve verificare la validita' dell'input prima di ogni esecuzione.

Assieme ai committenti ho elicitato i seguenti requisiti non funzionali:

19 Il sistema non deve essere eseguito in modo concorrente.

20 L'input deve poter variare tra due esecuzione diverse ma non puo' variare durante l'effettiva esecuzione.

Assieme ai committenti abbiamo concordato che:

Il sistema deve essere sviluppato usando il linguaggio di programmazione JAVA

Il sistema e' stato testato su Fedora 31.

Il documento dei requisiti e il codice devono essere completi e funzionanti per il giorno 03-05-2020.

Architettura del sistema

Come specificato dal Prof. Migliardi, ho incluso in questa sezione i dettagli di architettura derivati dai requisiti, lasciando i dettagli implementativi al codice.

Dato il diagramma delle classi nella sezione "modelli del sistema" inserisco qui una descrizione delle classi usate e il motivo della loro creazione.

Moltiplicatore: e' un classe usata solamente per descrivere i moltiplicatori di attacco e difesa da attribuire alla tupla {tipologia-Casella, tipologia-Pezzo, riferimento-Temporale}. Per ridurli di numero, in caso ve ne siano di simili e' possibile inserire come tipologia-Casella o riferimento-Temporale il simbolo '*' per indicare "qualsiasi caso".

Pezzo: questa classe permette di descrivere un pezzo come richiesto dai committenti, ovvero oggetti dotati degli attributi {tipologia, attacco, difesa, moltiplicatori} dove i moltiplicatori sono una lista di oggetti e non una tabella data la loro quantita'.

PezzoFactory: data la natura dei pezzi in cui ogni oggetto della stessa tipologia ha tutti i rimanenti valori uguali, ho deciso di costruire una classe che utilizzi una combinazione dei pattern Factory e Singleton per permettere la specifica di un singolo Pezzo per ogni tipologia e la restituzione di essi in base alla singola tipologia di pezzo richiesta. (Pezzo e' una classe concreta e non un interfaccia, ma ho associato la restituire di un'istanza in base a un parametro con il pattern Factory ugualmente).

Casella: questa classe rappresenta ogni cella della mappa, e' stata costruita per facilitare la gestione della tipologia delle caselle e l'inserimento dei pezzi, rispettando il limite di 5 pezzi. Inoltre, permette di ottenere i valori di attacco e difesa della casella di giorno/notte, il numero di pezzi per tipo e il massimo numero di pezzi dello stesso tipo.

Mappa: e' una classe fondamentale, permette di costruire la griglia di caselle date le dimensioni e le tipologie, consente di aggiungere i pezzi e soddisfare tutte le richieste di calcolo dei committenti.

Main: questa parte del sistema si occupa della gestione dei file e dell'input. Permette, usando la classe Mappa prima descritta, di stampare a terminale tutte le funzioni richieste.

Specifica requisiti di sistema

Funzione	Lettura dimensioni griglia di gioco
Descrizione	Il sistema deve richiedere l'inserimento da parte dell'utente delle dimensioni della mappa di gioco M,N attraverso il file "mappa_tipologie.txt"
Input	Un'unica riga di testo
Fonte	La prima riga di testo del file "mappa_tipologie.txt"
Output	Due numeri interi positivi M,N
Destinazione	Main control loop
Azione	<p>Il sistema deve leggere la prima riga del file "mappa_tipologie.txt".</p> <p>Se la riga corrisponde a due numeri interi separati da uno spazio li leggo come "M N" (M numero colonne, N numero righe), altrimenti il sistema termina segnalando l'errore:</p> <p>"ERRORE! IMPOSSIBILE LEGGERE LE DIMENSIONI DELLA MAPPA".</p> <p>Una volta letti, controlla che i numeri siano positivi, in caso contrario il sistema termina segnalando l'errore:</p> <p>"ERRORE! LE DIMENSIONI DELLA MAPPA NON POSSONO ESSERE NEGATIVE"</p>
Requisiti	Che il file esista e che sia un file di testo leggibile
Pre-condizione	Questa deve essere la prima lettura fatta
Post-condizione	Le dimensioni della mappa diventano i due numeri interi M,N letti dal file
Effetti collaterali	Nessuno

Funzione	Lettura della tipologia di determinate caselle
Descrizione	Il sistema deve permettere all'utente di specificare la tipologia di determinate caselle da lui scelte
Input	Gruppi di righe di testo
Fonte	Questi gruppi sono contenuti tutti all'intero del file "mappa_tipologie.txt"
Output	Una lista di elementi costituiti da: due numeri interi positivi X,Y e una stringa di testo tra le seguenti ["pianura","montagna","bosco"] corrispondente alla tipologia della casella
Destinazione	Main control loop
Azione	<p>Il sistema deve leggere un numero non definito di gruppi da 3 righe di testo dal file "mappa_tipologie.txt", saltando la prima (che contiene le dimensioni della mappa).</p> <p>Una volta lette, deve verificare che esse corrispondano a due righe</p>

	<p>contenenti un unico numero intero positivo e una terza riga contenente un'unica stringa tra quelle sopra citate.</p> <p>Se le righe non seguono questo formato il sistema termina segnalando l'errore: "ERRORE! IMPOSSIBILE LEGGERE LA SPECIFICA DELLE CASELLE".</p> <p>Se almeno uno dei due numeri interi X,Y non e' positivo il sistema termina segnalando l'errore:</p> <p>"ERRORE! LE COORDINATE DELLA CASELLA DEVONO ESSERE POSITIVE".</p> <p>Se almeno uno dei due numeri interi X,Y e' >= alle rispettive dimensioni della tabella M,N il sistema termina segnalando l'errore:</p> <p>"ERRORE! LE COORDINATE DELLA CASELLA NON POSSONO SUPERARE LE DIMENSIONI DELLA MAPPA".</p> <p>Se la tipologia non e' una stringa di testo tra quelle sopra citate il sistema termina segnalando l'errore: "ERRORE! LA TIPOLOGIA DI CASELLA INSERITA NON ESISTE".</p> <p>Sottolineo che ogni casella puo' avere una e una sola tipologia associata. Se inserisco due volte la stessa casella, solo l'ultima specifica sara' valida</p>
Requisiti	Che il file esista e sia un file di testo leggibile
Pre-condizione	Il sistema deve eseguire questa funzione dopo la lettura della mappa di gioco
Post-condizione	Il sistema deve avere la lista delle caselle specificate con le rispettive tipologie
Effetti collaterali	Nessuno

Funzione	Scelta della tipologia delle caselle non specificate dall'utente
Descrizione	Il sistema deve scegliere la tipologia delle caselle non specificate dall'utente
Input	Nessuno
Fonte	Nessuna
Output	Una matrice di grandezza N,M contenente la tipologia di tutte le caselle della mappa.
Destinazione	Main control loop
Azione	<p>Il sistema deve costruire una matrice di tipologie e popolarla inserendo per ogni casella non specificata dall'utente una ed una sola tipologia tra quelle possibili ["pianura","montagna","bosco"].</p> <p>La funzione che decide quale tipologia assegnare ad ogni casella garantisce che ognuna avra' probabilita' uniforme 1/3 di essere scelta.</p>
Requisiti	Il sistema deve disporre di una funzione random
Pre-condizione	Il sistema deve eseguire questa funzione dopo la lettura della mappa di gioco

Post-condizione	La mappa di gioco viene costruita, le strutture dati popolate ed e' possibile posizionare pezzi su di essa
Effetti collaterali	Nessuno

Funzione	Posizionamento di pezzi all'interno di determinate caselle
Descrizione	Il sistema deve permettere all'utente di posizionare pezzi all'interno di caselle a scelta
Input	Gruppi di righe di testo
Fonte	Questi gruppi sono contenuti tutti all'interno del file "mappa_pezzi.txt"
Output	Una lista di elementi costituiti da: due numeri interi positivi X,Y e una stringa di testo tra le seguenti ['orco':'O','elfo':'E','nano':'N'] corrispondenti alla tipologia di pezzo
Destinazione	Main control loop
Azione	<p>Il sistema deve leggere un numero non definito di gruppi da 3 righe di testo dal file "mappa_pezzi.txt".</p> <p>Una volta lette, deve verificare che esse corrispondano a due righe contenenti un unico numero intero positivo e una terza riga contenente un'unica stringa tra quelle sopra citate.</p> <p>Se le righe non seguono questo formato il sistema termina segnalando l'errore: "ERRORE! IMPOSSIBILE LEGGERE LA SPECIFICA DEI PEZZI".</p> <p>Se almeno uno dei due numeri interi X,Y non e' positivo il sistema termina segnalando l'errore:</p> <p>"ERRORE! LE COORDINATE DELLA CASELLA DEVONO ESSERE POSITIVE".</p> <p>Se almeno uno dei due numeri interi X,Y e' >= alle rispettive dimensioni della tabella M,N il sistema termina segnalando l'errore:</p> <p>"ERRORE! LE COORDINATE DELLA CASELLA NON POSSONO SUPERARE LE DIMENSIONI DELLA MAPPA".</p> <p>Se la tipologia non e' una stringa di testo tra quelle sopra citate il sistema termina segnalando l'errore:</p> <p>"ERRORE! LA TIPOLOGIA DI PEZZO INSERITA NON ESISTE"</p>
Requisiti	che il file esista e sia un file di testo leggibile
Pre-condizione	Il sistema deve eseguire questa funzione dopo la lettura della mappa di gioco
Post-condizione	La mappa viene completata, da questo momento la mappa sara' non vuota e le funzioni della mappa non restituiranno valori nulli.
Effetti collaterali	Nessuno

Funzione	Calcolo del numero di pezzi presenti sulla mappa per ciascuna tipologia
Descrizione	Il sistema deve calcolare il numero di pezzi presenti all'interno della

	mappa
Input	Nessuno
Fonte	Nessuna
Output	Una stringa formattata contenente i dati richiesti
Destinazione	Questa funzione deve appartenere alla classe “Mappa”
Azione	Il sistema deve calcolare il numero di pezzi presenti all’interno della mappa, in seguito proseguire a costruire la stringa contenente i dati richiesti nel formato “Totale:#, Orchi:#, Elfi:#, Nani:#”
Requisiti	Nessuno
Pre-condizione	Il sistema deve aver già’ letto tutti i dati e costruito le sue strutture dati
Post-condizione	Nessuna
Effetti collaterali	Nessuno

Funzione	Calcolo della posizione con massima difesa rispetto a un riferimento temporale
Descrizione	Il sistema deve calcolare la posizione della casella con difesa massima di giorno o notte
Input	Il riferimento temporale giorno o notte
Fonte	Il riferimento deve essere scritto manualmente dal programmatore durante la chiamata a metodo
Output	Una coppia di interi corrispondente alle coordinate X,Y
Destinazione	Questa funzione deve appartenere alla classe “Mappa”
Azione	Il sistema deve essere in grado di calcolare la difesa di ogni casella rispetto ai 2 riferimenti temporali giorno o notte, trovare il massimo tra questi e restituire le coordinate X,Y della casella rappresentante il massimo. In caso vi siano piu’ caselle caratterizzate dallo stesso valore al sistema sara’ sufficiente restituire la prima occorrenza con il suddetto valore
Requisiti	Nessuno
Pre-condizione	Il sistema deve aver già’ letto tutti i dati e costruito le sue strutture dati
Post-condizione	Nessuna
Effetti collaterali	Nessuno

Funzione	Calcolo della posizione con massimo attacco rispetto a un riferimento temporale
Descrizione	Il sistema deve calcolare la posizione della casella con attacco massimo di giorno o notte

Input	Il riferimento temporale giorno o notte
Fonte	Il riferimento deve essere scritto manualmente dal programmatore durante la chiamata a metodo
Output	Una coppia di interi corrispondente alle coordinate X,Y
Destinazione	Questa funzione deve appartenere alla classe “Mappa”
Azione	Il sistema deve essere in grado di calcolare l’attacco di ogni casella rispetto ai 2 riferimenti temporali giorno o notte, trovare il massimo tra questi e restituire le coordinate X,Y della casella rappresentante il massimo. In caso vi siano piu’ caselle caratterizzate dallo stesso valore al sistema sara’ sufficiente restituire la prima occorrenza con il suddetto valore
Requisiti	Nessuno
Pre-condizione	Il sistema deve aver gia’ letto tutti i dati e costruito le sue strutture dati
Post-condizione	Nessuna
Effetti collaterali	Nessuno

Funzione	Calcolo della posizione con maggior numero di pezzi dello stesso tipo
Descrizione	Il sistema deve calcolare la posizione della casella con maggior numero di pezzi dello stesso tipo
Input	Nessuno
Fonte	Nessuna
Output	Una coppia di interi corrispondente alle coordinate X,Y
Destinazione	Questa funzione deve appartenere alla classe “Mappa”
Azione	Il sistema deve essere in grado di calcolare per ogni casella il maggior numero di pezzi dello stesso tipo e restituire le coordinate X,Y della casella contenente il massimo numero. In caso vi siano piu’ caselle caratterizzate dallo stesso valore al sistema sara’ sufficiente restituire la prima occorrenza con il suddetto valore
Requisiti	Nessuno
Pre-condizione	Il sistema deve aver gia’ letto tutti i dati e costruito le sue strutture dati
Post-condizione	Nessuna
Effetti collaterali	Nessuno

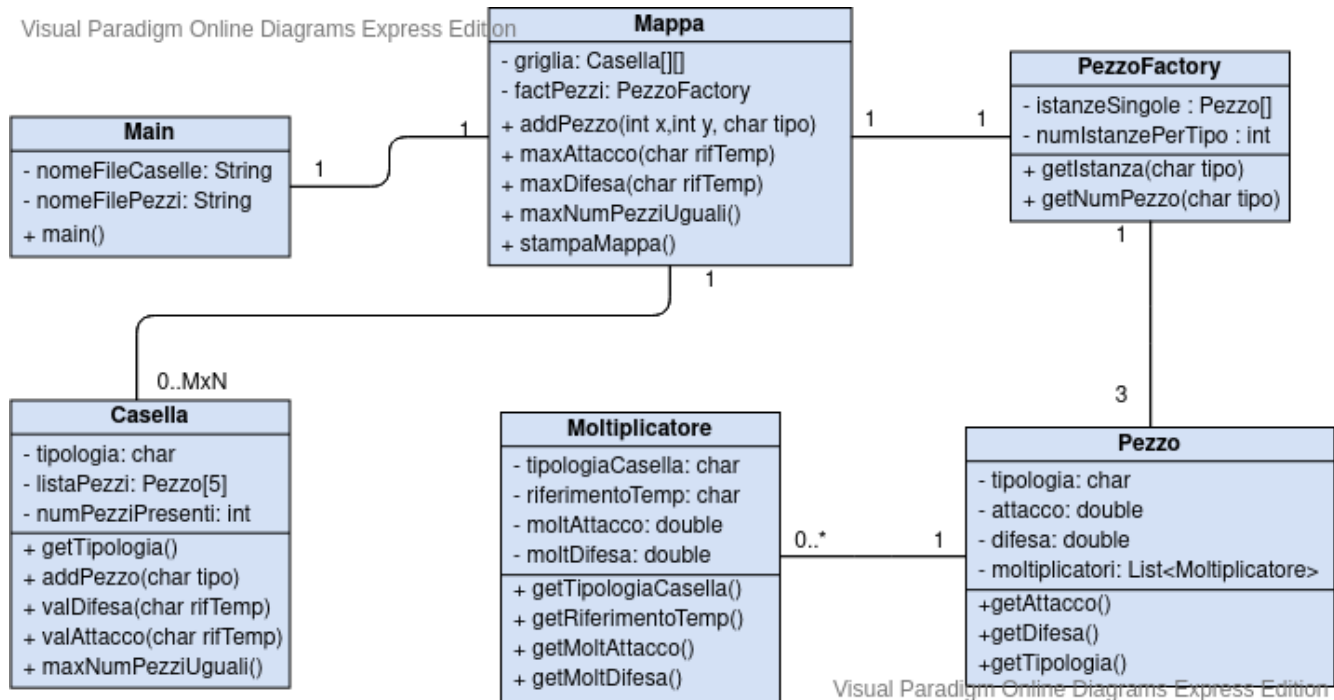
Funzione	Stampa tipologia caselle della mappa
Descrizione	Il sistema deve poter stampare la tipologia delle caselle della mappa di gioco
Input	Nessuno

Fonte	Nessuna
Output	Una matrice di caratteri
Destinazione	Questa funzione deve appartenere alla classe “Mappa”
Azione	<p>Il sistema deve poter restituire la mappa di gioco sotto forma di matrice di caratteri, quindi in particolare solo la tipologia delle singole casella e non i pezzi posizionati su di esse.</p> <p>Dato che le tipologie di caselle sono delle stringhe, i caratteri ritornati nella matrice seguiranno questa configurazione: [‘P’:”pianura”,’B’:”bosco”,’M’:”montagna”]</p>
Requisiti	Nessuno
Pre-condizione	Il sistema deve aver già letto tutti i dati e costruito le sue strutture dati.
Post-condizione	Nessuna
Effetti collaterali	Nessuno

Funzione	Memorizzazione dei pezzi e delle caselle all’interno del sistema
Descrizione	Il sistema deve memorizzare all’interno di strutture dati appropriate la tipologia delle caselle della mappa e la tipologia e la posizione dei pezzi inseriti
Input	Tipologie e posizioni
Fonte	Letture da file
Output	Le strutture dati popolate
Destinazione	Questa funzione deve appartenere alla classe “Mappa”
Azione	<p>Il sistema deve memorizzare all’interno di strutture dati appropriate i valori:</p> <p>La tipologia delle caselle della mappa deve essere una tra le seguenti [“pianura”,”montagna”,”bosco”].</p> <p>La tipologia di pezzi deve essere una tra le seguenti [“orco”,”elfo”,”nano”]</p> <p>Il sistema deve verificare che una casella non contenga più di 5 pezzi.</p> <p>Il sistema deve associare ad ogni tipologia di pezzo un valore di attacco, difesa e un numero non definito di moltiplicatori scelti dal committente.</p> <p>Non verrà creata una classe per ogni tipologia di pezzo, ma verrà usata come attributo per identificarli tra di loro</p>
Requisiti	Nessuno
Pre-condizione	Questa operazione deve essere eseguita dopo la lettura dei file di input
Post-condizione	Le strutture dati sono popolate
Effetti collaterali	Nessuno

Modelli per il sistema

Data la semplicità del progetto ho inserito solamente il diagramma delle classi in questa sezione.



Evoluzione del sistema

Questa è la prima e unica versione prodotta del documento dei requisiti. Data la durata e la difficoltà del progetto non ho proposto una guida alla modifica dei requisiti (requirement change model).

Propongo comunque in questa sezione una piccola descrizione di come ho svolto il progetto:

1. I committenti, a partire dalla consegna inviata, mi hanno inviato ognuno singolarmente un documento scritto, contenente le funzioni che desideravano il sistema svolgesse e le modalità di svolgimento.
2. Ho analizzato i documenti inviati, valutato quali fossero le funzionalità in comune e i punti di conflitto tra i due documenti. Ho di conseguenza raccolto una lista di domande da porre ai committenti.
3. Abbiamo svolto un meeting di gruppo via zoom in cui ho mostrato loro l'analisi fatta a partire dai documenti scritti, ho in seguito fatto domande a voce per risolvere i punti di conflitto, chiarito i punti non dettagliati e verificato che le funzionalità richieste da un singolo committente fosse ben accette da entrambi.
4. Ho quindi completato l'analisi iniziata al punto 2 grazie alle decisioni prese durante il meeting e le soluzioni proposte per i punti di conflitto. Sono quindi arrivato a costruire una prima lista di user requirement visibile all'appendice-3.
5. Ho verificato la correttezza e completezza dei requisiti e prodotto questo documento. In seguito ho costruito anche il documento di architettura e infine sviluppato il codice.

Appendici

Appendice-1: esempio formato file “mappa_tipologie.txt”

```
5 4
1
1
bosco
1
2
bosco
1
3
bosco
2
1
montagna
3
2
pianura
1
3
bosco
EOF
```

Appendice-2: esempio formato file “mappa_pezzi.txt”

```
0
0
orco
2
1
nano
3
1
nano
1
1
elfo
1
4
elfo
```

Appendice-3: lista di user requirement prodotta dopo l’incontro con i committenti

RIFERIMENTO TEMPORALE:

Entrambi i committenti, come da consegna, prevedono l’esistenza di un riferimento temporale: [“giorno”, “notte”].

DESCRIZIONE PEZZI:

Entrambi i committenti, come da consegna, prevedono che ogni tipologia di pezzo sia caratterizzata da un valore di attacco/difesa e dei moltiplicatori (maggiori specifiche negli user requirement).

DIMENSIONE MATRICE:

Angelo: la matrice ha dimensione fissa 10x10.

Niccolo': la matrice ha dimensioni variabile inseribile da riga di comando (in questo caso fare attenzione che tutte le celle riportate nel file appartengano alla matrice).

Le proposte dei committenti erano in conflitto tra di loro.

Abbiamo concordato che la matrice deve avere dimensione variabile $M \times N$ (M numero colonne, N numero righe). Specificata all'inizio del file "mappa_tipologie.txt" attraverso una riga di formato "M N".

INPUT TIPOLOGIA CASELLE

Angelo: viene riempita casualmente ma in un file "paesaggio.txt" c'è la probabilità che a ciascuna casella venga assegnato un certo tipo.

Es:

34 (percentuale pianura)

25 (percentuale montagna)

41 (percentuale foresta)

Bisogna controllare che la somma delle percentuali sia 100, altrimenti il programma si blocca e stampa a schermo "Errore durante la creazione della mappa".

Niccolo': tutte le celle sono inizializzate con la tipologia "pianura". Lettura da file "mappa_tipologie.txt" da trovare all'interno della directory contenente i file .java del gioco . "mappa_tipologie.txt" contiene gruppi di linee di testo nel formato:

"

X

Y

TIPOLOGIA

"

"mappa_tipologie.txt" contiene per ogni cella che si vuole personalizzare (ovvero ogni cella che dev'essere di tipologia diversa da pianura) la tipologia assegnata alla stessa: questo file può contenere da 0 a $M \times N$ gruppi di X,Y, tipo indicanti la cella corrente e la sua tipologia, ogni cella ha una ed una sola tipologia.

Le due proposte dei committenti erano in conflitto tra di loro.

Abbiamo concluso cercando di soddisfare entrambe le richieste.

All'interno del file "mappa_tipologie.txt" si possono specificare la tipologia di singole caselle nella forma descritta dal committente Niccolo'.

Tutte le caselle non specificate all'interno del file verranno determinate casualmente scegliendo uniformemente tra le tre (o varie... se aumenteranno) possibili tipologie (tutte avranno prob. unif $1/|\omega|$ dove ω è l'insieme dei possibili eventi elementari).

INPUT POSIZIONE PEZZI

Angelo: non ha specificato (come da consegna?).

Niccolo': lettura da file "mappa_pezzi.txt" che si trova all'interno della directory contenente i file .java del gioco.

"mappa_pezzi.txt" contengono gruppi di linee di testo secondo lo schema:

"

X

Y

PEZZO

"

ogni casella può avere da 1 a 5 pezzi, nella riga "pezzi" di ogni gruppo X,Y pezzi vi è una singola parola dell'insieme "pezzi" (ripeto che si scrive in questo file una cella con i propri pezzi solo se quella cella ha almeno un pezzo), nel caso si vogliano inserire più pezzi sarà necessario aggiungere un altro gruppo.

Data la mancata specifica da parte del committente Angelo, abbiamo deciso di implementare la funzionalità proposta dal committente Niccolo'. Inoltre i committenti hanno deciso di non sviluppare una classe per ogni tipologia ma un'unica classe con attributo tipologia (è un requisito? o è più una questione di architettura/design?)

CONTROLLO INPUT:

Angelo: se uno dei file manca o è in un formato sbagliato il programma si blocca e stampa a schermo "Controllare che il file -nomefile- sia presente e nel formato corretto".

Se il numero di pezzi in una casella supera i 5 il programma stampa a schermo "Nella cella -x,y- ci sono troppi pezzi", tuttavia continua l'esecuzione ignorando i pezzi in eccesso.

Se un pezzo viene posizionato fuori dalla mappa il programma stampa a schermo "Un pezzo è stato posizionato fuori dalla mappa", tuttavia continua l'esecuzione ignorando i pezzi fuori dalla mappa.

Niccolo': non è necessario effettuare verifiche di formato (ovvero il contenuto dei file stessi non è coerente con le specifiche) in quanto si lascia allo sviluppatore la scelta di scrittura dei file, con totale affidabilità.

A fine dei loro interessi i controlli riguardanti l'esistenza, il formato dei file e la validità dei valori verranno fatti in modo completo, in modo da rendere sicura l'esecuzione del software. Inoltre anche le situazioni eccezionali verranno controllate. La specifica è stata rimandata allo sviluppatore.

OUTPUT:

Angelo: il programma stampa i risultati su un file "output.txt".

1)Totale:#totale - Orchi:#orchi Nani:#nani Elfi:#Elfi

2)La casella con il maggior valore di difesa di giorno (x,y) (x',y') ...

3)La casella con il maggior valore di difesa di notte (x,y) (x',y') ...

4)La casella con il maggior valore di attacco di giorno (x,y) (x',y') ...

5)La casella con il maggior valore di attacco di notte (x,y) (x',y') ...

6)La casella con il maggior numero di pezzi dello stesso tipo (x,y) (x',y') ...

Niccolo':

- 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

Dopo esserci consultati assieme i committenti hanno deciso di eseguire l'output via terminale e nel seguente formato:

1) verrà stampato sia il totale di pezzi presenti per ogni formato che il totale di pezzi di qualsiasi tipo nel formato proposto dal committente Angelo.

2-6) i punti da 2 a 6 verranno stampati nel formato: "la casella con maggiore attacco/difesa di giorno/notte e': (x,y)" stampando solo la prima occorrenza.

Hanno inoltre deciso di aggiungere un'altra stampa data la casualità della tipologia di caselle all'interno della mappa. Questa stampa riguarderà una matrice MxN di caratteri corrispondenti al tipo di caselle nella forma: P pianura, M montagna, B bosco.

Es:

PPPMB

PPMBB

MMBPP

Indice

Pag 1:	<u>Prefazione</u>
	<u>Introduzione</u>
	<u>Glossario</u>
Pag 1-3:	<u>Definizione dei requisiti utente</u>
Pag 3:	<u>Architettura del sistema</u>
Pag 4-9:	<u>Specificazione requisiti di sistema</u>
Pag 10:	<u>Modelli per il sistema</u>
	<u>Evoluzione del sistema</u>
Pag 11-14:	<u>Appendici</u>
Pag 14:	<u>Indice</u> (recursive)