Documento di specifica dei requisiti

Analisi dei requisiti del progetto I Homework, Corso di Ingegneria del SW

Versione	Data	Modifiche apportate
1.0	20/04/2020	Versione iniziale del documento di specifica dei requisiti
1.1	23/4/2020	Aggiunta di nuove specifiche definite dai committenti a seguito della verifica del documento (pag. 3)
1.2	27/4/2020	Apportate alcune modifiche alla formattazione del documento

1. Introduzione

1.1 Obiettivi del documento

Questo documento ha come obiettivo la specifica dei requisiti, funzionali e non, di un sistema batch di elaborazione di dati statistici. In questo documento vengono inoltre descritti l'ambito di utilizzo, le funzionalità e i vincoli definiti dai committenti che caratterizzano il nuovo sistema che dovrà essere sviluppato. Questo documento, infine, ha lo scopo di dirigere il design e l'implementazione del sistema stesso in un linguaggio di programmazione object-oriented. Il procedimento seguito durante la fase di ingegnerizzazione dei requisiti ivi riportati è interamente descritto in un apposito documento allegato.

1.2 Project summary

Analisi svolta da: *Giacomo Camposampiero*
Committenti: *Riccardo Tumiati*, *Francesco Bordignon*

1.3 System overview

Il sistema, come già definito precedentemente, è un sistema di elaborazione dati batch. Lo sviluppo del sistema è stato commissionato nell'ambito di un progetto universitario e i suoi fini sono puramente didattici. L'ambito in cui si colloca il sistema è quello dei giochi fantasy e il suo scopo è, per l'appunto, quello di estrapolare dati significativi a partire da input forniti dall'utente in un formato definito in accordo con i committenti. Il sistema non necessita di interazioni con l'utente (ad esclusione dell'avvio del programma) o con altri software.

1.4 Document overview

Il seguito del documento fornirà informazioni più dettagliate riguardo al sistema stesso e sarà organizzato nelle seguenti sezioni:

- *sezione 2*: requisiti funzionali; in questa sezione sono riportati, in formato tabulare, tutti i requisiti funzionali che sono stati individuati nel corso dell'analisi; ognuno di questi requisiti descrive una funzionalità che il software offre all'utente
- *sezione 3*: requisiti non funzionali; in questa sezione è presente una lista di tutti i requisiti non funzionali (vincoli che il programma deve rispettare, direttive dei committenti, etc.) ricavati dall'analisi dei requisiti
- *sezione 4*: system architecture; in questa sezione viene riportata una descrizione ad alto livello di una possibile implementazione del sistema
- *sezione 5*: context model; sezione in cui viene approfondito l'ambiente nel quale dovrà operare il sistema e le interazioni con esso
- *sezione 6*: use case model; in questa sezione sono riportati alcuni esempi pratici di interazioni tra l'utente e il sistema
- *sezione 7*: system evolution; in tale sezione sono descritte eventuali evoluzioni del sistema dovute a evoluzione dell'hardware, delle necessità dell'utente, etc.
- *sezione 8*: glossario; in questa sezione verrà esposto il significato di alcune terminologie proprie del linguaggio informatico utilizzate in alcuni passaggi del documento

2. Requisiti funzionali

2.1 Lettura input e inizializzazione delle strutture dati

Funzione	Lettura dell'input dai file e inizializzazione delle strutture dati
Descrizione	Lettura delle informazioni necessarie all'inizializzazione e al popolamento della mappa dopo l'avvio del programma. Istanziazione della mappa sulla base delle informazioni precedentemente lette dai file.
Input	Informazioni sulla struttura della mappa (tabella che rappresenta la griglia della mappa in cui sono definiti i tipi di terreno e, intrinsecamente, le sue stesse dimensioni) e informazioni riguardo alle pedine presenti nella mappa (coordinate del pedina e tipo di personaggio)
Sorgente dell'input	1. *mappa.txt* - il file si trova nella stessa cartella in cui viene eseguito il programma 2. *pedine.txt* - il file si trova nella stessa cartella in cui viene eseguito il programma
Output	Strutture dati che rappresentano la mappa secondo quanto definito dai file di input
Destinazione	Successive elaborazioni da parte dell'applicativo
Azione	Il programma accede al file *mappa.txt* e ne legge il contenuto. Una volta completata la lettura, nel caso in cui non si siano riscontrati errori, viene istanziata la mappa. Il programma successivamente accede al file pedine.txt e ne legge il contenuto. Per ogni gruppo di linee corrispondente ad una pedina, tenta di inserire tale pedina all'interno della mappa. Al termine della lettura del file, se l'inserimento di tutte le pedine è andato a buon fine l'esecuzione prosegue, altrimenti il programma si blocca.
Requisiti funzione	❖ in una cella della mappa possono essere inserite al massimo 5 pedine ❖ i gruppi di linee nel file *pedine.txt* non devono essere separati da una riga vuota ❖ i tipi di pedine e di terreno devono essere espressi mediante opportuni codici [vedi sezione 3] ❖ i file mappa.txt e pedine.txt devono rispettare precise formattazioni [vedi sezione 3]; il caso in cui esse non siano rispettate deve essere opportunamente gestito: si rimanda la trattazione nel dettaglio alla funzione "Gestione di errori nei file di input" ❖ ulteriori requisiti che influenzano, direttamente o indirettamente, questa funzione sono espressi dai requisiti funzionali 3.1, 3.3, 3.4, 3.5, 3.7, 3.8, 3.9, 3.10 della sezione 3
Pre-condition	I file di input devono trovarsi nella stessa cartella in cui è eseguito il programma
Post-condition	Nessuna
Effetti collaterali	Eventuali errori nella lettura dei file potrebbero determinare un'interruzione del normale flusso di esecuzione del programma.

2.2 Gestione di errori nella lettura dei file di input

Funzione	Gestione di errori nella lettura dei file di input
Descrizione	Gestione specifica di eventuali inconsistenze riscontrate nell'apertura o nella lettura dei file utilizzati come input al programma
Input	Eccezioni lanciate dal programma durante la lettura dei file in input o condizioni di inconsistenza del contenuto dei file rilevate mediante appositi controlli
Sorgente dell'input	Modulo di lettura dei file del programma stesso
Output	Messaggio di errore personalizzato che esprime il tipo di errore riscontrato *e il numero di linea nel file nel quale è presente questo errore*[1].
Destinazione	Modulo di gestione dell'output dei risultati
Azione	1. **Errori nell'apertura dei file *mappa.txt* e *pedine.txt***: l'esecuzione del programma deve essere immediatamente bloccata e deve essere restituito all'utente un messaggio che indica quale file non è stato possibile aprire 2. **Errori di formattazione o inconsistenze nel file *mappa.txt***: l'esecuzione del programma deve essere immediatamente bloccata e deve essere restituito all'utente un messaggio che indica il tipo di errore (e.g. cella mancante, tipo di terreno non valido, etc.) *e il numero di riga del file di input che ha generato l'errore*[1] 3. **Errori di formattazione o inconsistenze nel file *pedine.txt***: l'esecuzione del programma deve proseguire fino al termine della lettura di tutto il file; al termine della lettura il normale flusso di esecuzione del programma deve essere interrotto e devono essere restituito all'utente un messaggio contenente una lista di tutti gli errori riscontrati nella lettura del file (e.g. inserimento di più di 5 pedine nella cella (x,y), tipo di pedina non valido, etc.) *e le rispettive righe del file in cui sono stati riscontrati gli stessi*[1]
Requisiti funzione	❖ si rimanda ai requisiti 3.1, 3.4, 3.5, 3.7, 3.8, 3.9 e 3.10 della sezione 3 per un approfondimento delle specifiche definite dall'utente in termini di convenzioni e struttura che i file di input devono rispettare
Pre-condition	Devono essere definiti adeguati controlli nel modulo di lettura dell'input al fine di intercettare eventuali errori logici o di formattazione dei file sulla base delle specifiche fornite dai committenti
Post-condition	Nessuna
Effetti collaterali	Nessuno

[1] Requisito aggiunto nella versione 1.1 del documento

2.3 Calcolo della casella avente maggior valore di attacco/difesa di giorno/notte

Funzione	Calcolo della casella avente maggior valore di attacco/difesa di giorno/notte
Descrizione	Elaborazione svolta dall'applicativo sulla mappa che ha come scopo la ricerca della casella della mappa avente il maggior valore di attacco/difesa di giorno o di notte
Input	Strutture dati che rappresentano la mappa e le pedine su di essa disposte, costruite a partire dai dati forniti in input
Sorgente dell'input	Modulo di lettura dei file in input del programma
Output	Coordinate della casella/e avente/i maggior valore di attacco/difesa di giorno/notte
Destinazione	Gestione della restituzione dell'output all'utente
Azione	Il programma deve visitare tutte le caselle della mappa e, al termine, essere in grado di determinare qual è la casella con il maggior valore di attacco/difesa nelle diverse situazioni temporali. L'applicativo deve naturalmente tenere conto di eventuali bonus/malus derivanti dalle condizioni (riferimento temporale e terreno) in cui si trovano le pedine.
Requisiti funzione	❖ in caso di risultati multipli (ovvero più caselle con lo stesso valore massimo di attacco/difesa) il programma deve restituire tutti i risultati dell'elaborazione ❖ si rimanda ai requisiti 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.10, 3.11 e 3.12 della sezione 3 per approfondire aspetti legati a questa particolare funzione
Pre-condition	La mappa deve trovarsi in uno stato consistente
Post-condition	L'elaborazione non deve modificare lo stato della mappa

2.4 Calcolo del numero delle pedine presenti sulla mappa per ciascuna tipologia

Funzione	Calcolo del numero delle pedine presenti per ciascuna tipologia di pedina
Descrizione	Elaborazione svolta dall'applicativo sulla mappa che ha come scopo il calcolo del numero totale di pedine per ciascuna tipologia di pedina {elfo, nano, orco}.
Input	Strutture dati che rappresentano la mappa e le pedine su di essa disposte
Sorgente dell'input	Modulo di lettura dei file in input del programma
Output	Numero totale di pedine per ciascuna tipologia di pedina
Destinazione	Gestione della restituzione dell'output all'utente
Azione	Il programma deve visitare tutte le caselle della mappa e, al termine, essere in grado di determinare il numero totale di pedine per ogni tipologia di pedina
Requisiti funzione	❖ per "tipologia" nel testo della consegna si intende "tipologia di pedina" ❖ in caso di risultati multipli (ovvero più caselle con lo stesso valore massimo di attacco/difesa) il programma deve restituire tutti i risultati dell'elaborazione ❖ si rimanda ai requisiti 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.10, 3.11 e 3.12 della sezione 3 per approfondire aspetti legati a questa particolare funzione
Pre-condition	La mappa deve trovarsi in uno stato consistente
Post-condition	L'elaborazione non deve modificare lo stato della mappa

2.5 Calcolo della casella avente il maggior numero di pedine dello stesso tipo

Funzione	Calcolo della casella avente il maggior numero di pedine dello stesso tipo
Descrizione	Elaborazione svolta dall'applicativo sulla mappa che ha come scopo l'identificazione della/e cella/e avente/i il maggior numero di pedine dello stesso tipo.
Input	Strutture dati che rappresentano la mappa e le pedine su di essa disposte, costruite a partire dai dati forniti in input mediante i file
Sorgente dell'input	Modulo di lettura dei file in input del programma
Output	Coordinate della/e casella/e aventi maggior numero di pedine dello stesso tipo
Destinazione	Gestione della restituzione dell'output all'utente
Azione	Il programma deve visitare tutte le caselle della mappa e, al termine, essere in grado di individuare e restituire le coordinate della/e casella/e aventi il maggior numero di pedine della stessa tipologia.
Requisiti funzione	❖ in caso di risultati multipli (ovvero più caselle con lo stesso valore massimo di attacco/difesa) il programma deve restituire tutti i risultati dell'elaborazione ❖ si rimanda ai requisiti 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.10, 3.11 e 3.12 della sezione 3 per approfondire aspetti legati a questa particolare funzione
Pre-condition	La mappa deve trovarsi in uno stato consistente
Post-condition	L'elaborazione non deve modificare lo stato della mappa
Effetti collaterali	Nessuno

2.6 Gestione dell'output del programma

Funzione	Gestione dell'output del programma
Descrizione	Gestione e restituzione all'utente di tutti gli output prodotti (errori nella lettura dei file, risultati dell'elaborazione, etc.) durante l'elaborazione dal programma
Input	Messaggio da restituire all'utente
Sorgente dell'input	Altri moduli del programma
Output	Messaggio restituito all'utente
Destinazione	Secondo le specifiche date dai committenti, la destinazione dell'output deve essere il video
Azione	L'applicativo, nel momento in cui si rende necessario restituire un qualsiasi tipo di informazione all'utente, deve gestire in maniera appropriata questa necessità; nel caso specifico, l'output deve essere visualizzato a schermo, secondo quanto stabilito
Requisiti funzione	❖ si rimanda ai requisiti 3.11 e 3.12 della sezione 3 per approfondire aspetti legati a questa particolare funzione
Pre-condition	Nessuna
Post-condition	Nessuna
Effetti collaterali	Nessuno

3. Requisiti non funzionali

3.1 La mappa di gioco consiste in una griglia rettangolare composta da MxN celle; ognuna di queste celle ha una tipologia (*pianura*/*bosco*/*montagna*); le tipologie di cella sono mutualmente esclusive.

3.2 Le coordinate della mappa partono da 0 (la prima cella in alto a sinistra ha quindi coordinate (0,0)).

3.3 La mappa possiede un riferimento temporale; il riferimento temporale è una caratteristica intrinseca alla mappa: tutte le celle della mappa hanno quindi lo stesso riferimento giorno/notte. Il riferimento è impostato di default a “giorno”.

3.4 Su ciascuna cella della mappa è possibile posizionare delle pedine; in ciascuna cella è possibile posizionarle al massimo 5; ogni pedina è caratterizzata da un tipo (elfo/nano/orco).

3.5 I valori di attacco/difesa (A/D) sono definiti per tipologia di pedina: **E** 5/2, **N** 2/5 e **O** 4/4.

3.6 I pezzi sono caratterizzati da modificatori di combattimento (+A%, +D%): elfi bosco +0%/+100%, nani montagna +100%/+0%, orco giorno -50%/-50%; orco notte +50%/+50%.

3.7 Le caratteristiche della mappa vengono fornite mediante un file apposito. I file di input devono essere denominati mappa.txt (file che contiene le caratteristiche della mappa) e pedine.txt (file che contiene le caratteristiche delle pedine).

3.8 Il file di input che contiene le informazioni sulle pedine è strutturato in gruppi di linee, un gruppo per ogni pedina. I gruppi di linee del file pedine.txt non sono tra loro separati da una riga vuota. I gruppi di linee sono strutturati nel seguente formato:

`X`	*coordinata x in cui è posizionata la pedina, $X \in [0, M-1]$*
`Y`	*coordinata y in cui è posizionata la pedina, $Y \in [0, N-1]$*
`tipo`	*tipo di pedina, tipo $\in \{E, N, O\}$*

3.9 Il file di input della mappa deve essere strutturato come una tabella, in cui ogni cella corrisponde ad il tipo di terreno nella griglia alle coordinate corrispondenti; la dimensione della mappa può essere ricavata dalla stessa tabella; le celle sono tra loro separate da uno spazio; è riportato di seguito un esempio di file mappa.txt fornito dai committenti.

```
P P P P B B M
P P P B B M M
P B B B M M M
```

esempio di file mappa.txt [dim. MxN=7x4]

3.10 I codici associati ad ogni tipo di terreno sono: **P** (pianura), **B** (bosco), e **M** (montagna). I codici associati ai tipi di personaggi sono invece: **E** (elfo), **N** (nano) e **O** (orco).

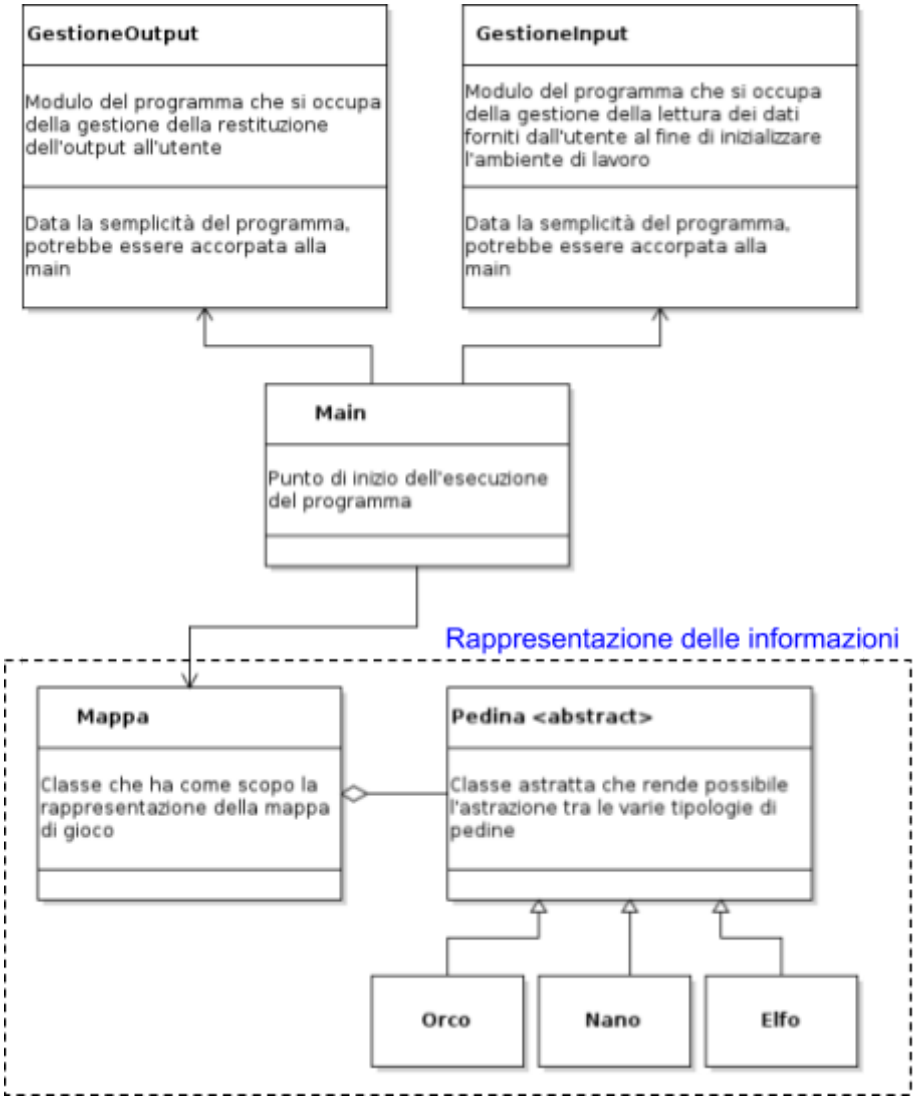
3.11 Le caselle risultato devono essere identificate nella stampa dell’output mediante le loro coordinate. L’output deve essere stampato a schermo.

3.12 Nel caso di risultati multipli, devono essere tutti restituiti come risultati validi dell’elaborazione.

3.13 Non ci sono particolari vincoli di sistema/usabilità/sicurezza/prestazioni. L’unico requisito è che il programma possa essere eseguito su un normale PC desktop.

4. System architecture

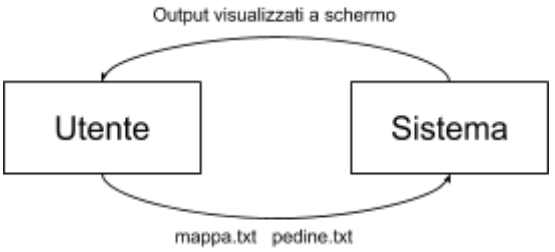
In questa sezione viene riportata una schematizzazione ad alto livello dell'architettura che si pensa di implementare nelle fasi successive dello sviluppo del software, con lo scopo di evidenziare la distribuzione delle funzioni tra le varie componenti del sistema. Una descrizione molto più esaustiva e dettagliata dell'architettura sarà successivamente stilata durante la progettazione del sistema, che avrà come output il documento di architettura.



5. Context model

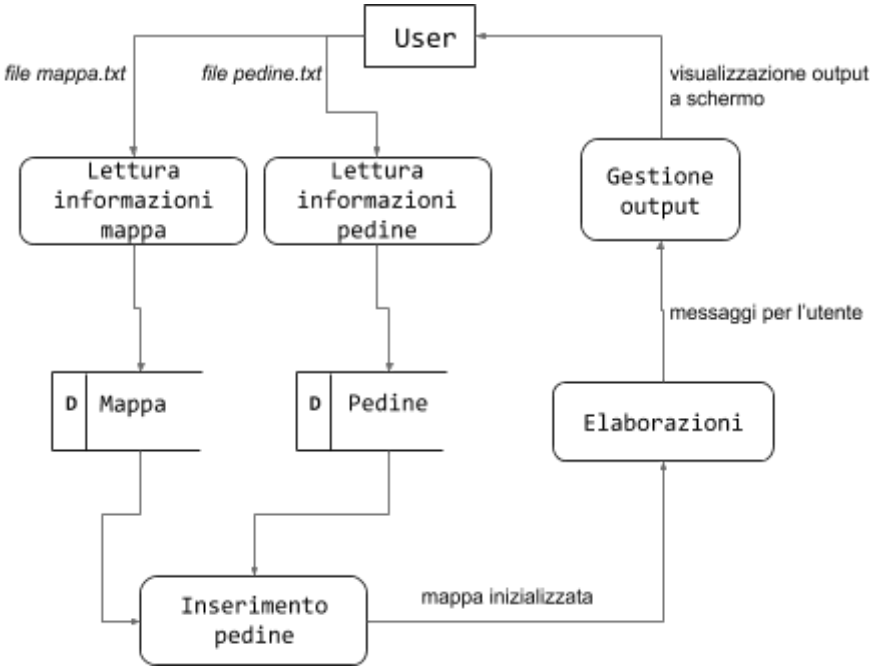
5.1 Ambiente esterno

Il contesto all'interno del quale opera l'applicativo è estremamente ridotto e le interazioni con l'ambiente esterno sono pressoché nulle. Queste ultime possono in definitiva essere riassunte mediante un semplice schema riportato qui di seguito.



5.2 Data-flow model

Sulla base dell'architettura "primitiva" definita nella precedente sezione [4], è possibile anche stilare un semplice data-flow model, allo scopo di evidenziare i flussi di informazione all'interno del sistema stesso.



6. Use case model

A causa della limitata possibilità di interazione dell'utente con l'applicativo l'unico use case definito è stato il seguente.

6.1 Avvio

Nome dello use case	Avvio del programma
Riassunto	L'unica interazione dell'utente con il programma consiste nell'avvio del programma
Flusso di esecuzione	1. Avvio del programma da parte dell'utente 2. Elaborazione dei risultati 3. Visualizzazione dei risultati a video
Flussi di esecuzione alternativi	2. (*alternativo*) Viene interrotta l'esecuzione del programma a seguito del riscontro di errori nella formattazione o nel contenuto dei file di input
Extension point	Nessuno
Pre-condizioni	I file di input devono trovarsi nella stessa cartella da cui viene lanciato il programma
Post-condizioni	Nessuna

7. System evolution

Non sono previste dai committenti, allo stato attuale, evoluzioni significative del software.

8. Glossario

Programma batch: programma che può essere eseguito senza un'interazione diretta con l'utente [fonte]
Programmazione object-oriented: paradigma di programmazione che permette di definire oggetti software in grado di interagire gli uni con gli altri attraverso lo scambio di messaggi [fonte]
Eccezione: occorrenza di diversi tipi di condizioni o eventi che alterano il normale flusso di controllo ed esecuzione di un microprocessore programmabile o di un programma [fonte]
Struttura dati: entità usata per organizzare un insieme di dati all'interno della memoria del computer [fonte]

9. Sommario