**“Ingegneria del Software”**

**2020-202****1**

**Docente: Prof. Angelo Furfaro**

**KenKen**

|  |  |
| --- | --- |
| **Data** | <gg/mm/aaaa> |
| **Documento** | Documento Finale – D3 |

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| **Nome e Cognome** | **Matricola** | **E-mail address** |
| Giulio Talarico | 200881 | giuliotal@gmail.com |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Sommario

**Non è stata trovata alcuna voce d'indice.**

## List of Challenging/Risky Requirements or Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Challenging Task** | **Date the task is identified** | **Date the challenge is resolved** | **Explanation on how the challenge has been managed** |
| Controllo del soddisfacimento del vincolo aritmetico all’interno dei blocchi dello schema di gioco | 26-06-2021 | 27-06-2021 | La griglia di gioco tiene traccia dei blocchi che ne costituiscono lo schema: per verificare che il vincolo è soddisfatto si effettuano tutte le possibili permutazioni delle cifre all’interno ogni blocco, che costituiscono gli operandi. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

A. Stato dell’Arte

L’inventore del KenKen, il matematico Tetsuya Miyamoto, crea i suoi puzzles a mano dal 2004. Con la diffusione del gioco al di là dei confini del Giappone, in particolare in America, dove i principali giornali iniziavano a pubblicare i puzzles di Miyamoto, la creazione artigianale comincia ad essere insufficiente per gestire la domanda sempre crescente di giochi di KenKen, pertanto si arriva alla produzione di un software proprietario per la generazione automatica: [*Kenerator*](http://www.kenkenpuzzle.com/about/kenerator)*.*

*Kenerator* viene sviluppato con la collaborazione di David Levy, campione di scacchi ed esperto di intelligenza artificiale, sotto la stretta supervisione di Miyamoto, ed è oggi il software utilizzato per la generazione automatica di giochi ufficiali di KenKen sia sul web, sul sito <http://www.kenkenpuzzle.com/>, che su dispositivi mobili, tramite l’app KenKen Classic.

Il processo di generazione avviene secondo diversi criteri: livello di difficoltà, numero di celle singole da includere all’interno della griglia di gioco, quale insieme di cifre e quali operazioni aritmetiche utilizzare e se mostrarne i simboli all’interno delle celle, per incrementare il livello di difficoltà.

Una volta verificato che tutte le celle sono disposte in maniera corretta, quindi che non vi siano cifre ripetute all’interno di una stessa riga o all’interno di una stessa colonna, e che il puzzle abbia una sola soluzione, un apposito modulo del software tenta di risolvere il puzzle fingendo di essere un utente in modo da determinarne la difficoltà: se essa corrisponde alla difficoltà richiesta il puzzle viene accettato, altrimenti viene scartato e il processo di generazione automatica ricomincia.

*Kenerator,* sebbene costituisca lo stato dell’arte nella categoria di software per la generazione e la risoluzione di giochi di KenKen, non è l’unico. Esempi degni di nota sono [*kenken-maker*](https://github.com/calebsander/kenken-maker)di Caleb Sander e [KenKen solver](https://tirl.org/software/kenken/)di Charlie DeTar.

Il primo, il cui codice sorgente è pubblicamente disponibile su github, segue un’interessante iter per la generazione: si parte dalla costruzione di una griglia valida, che rispetti dunque i vincoli di non ripetizione delle cifre all’interno della stessa riga o della stessa colonna, tramite scambi e trasposizioni di righe a partire da una configurazione di base, dopodiché si costruiscono in maniera randomica i vari blocchi con le operazioni aritmetiche da soddisfare e si tenta di risolvere il puzzle ottenuto per determinarne la difficoltà.

Il secondo, utilizzabile via web, permette non solo la generazione e la risoluzione automatica di giochi di KenKen, ma fornisce anche all’utente la possibilità di costruire una griglia di gioco manualmente.

Un ulteriore esempio sui generis è [KenKen Solver](https://kenkensolver.herokuapp.com/) di Kenneth Myers: esso si differenzia dai software precedentemente analizzati in quanto utilizza tecniche di *computer vision* per risolvere dei giochi di KenKen a partire da un’immagine della griglia di gioco, fornita dall’utente via web.

B. Raffinamento dei Requisiti

***B.1 Servizi (con prioritizzazione)***

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **SERVIZIO** | **IMPORTANZA** | **COMPLESSITA’** |
| #1 | Configurazione manuale della griglia di gioco | Alta | Media |
| Tramite GUI è possibile specificare le dimensione della griglia di gioco, inizialmente vuota, e costruire in maniera dinamica, interattiva ed arbitraria lo schema di blocchi contingui su cui definire l’operazione aritmentica e il risultato da ottenere. | | | |

| #2 | Salvataggio/caricamento della griglia di gioco su/da file system | Bassa | Media |
| --- | --- | --- | --- |
| Si definisce un sistema di lettura/scrittura delle griglie di gioco per permettere di salvare su file system la griglia di gioco creata e ripristinare griglie di gioco precedentemente memorizzate. I file vengono memorizzati in un formato apposito in una cartella di salvataggio che può essere scelta dall’utente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **SERVIZIO** | **IMPORTANZA** | **COMPLESSITA’** |
| #3 | Inserimento dinamico di numeri all’interno della griglia di gioco | Alta | Bassa |
| Una volta specificato e confermato lo schema di gioco le celle della griglia diventano campi di testo interattivi in cui è sempre possibile inserire e cancellare dei numeri. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| #4 | Controllo dinamico del soddisfacimento dei vincoli | Media | Alta |
| In ogni momento è possibile richiedere il controllo del soddisfacimento dei vincoli della griglia di gioco corrente: le celle contenenti dei numeri che risultano essere duplicati all’interno della stessa riga o della stessa colonna vengono evidenziate come errate. Inoltre si verifica che i blocchi completi rispettino il vincolo aritmetico: in caso negativo l’intero blocco viene evidenziato come errato. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| #5 | Calcolo delle soluzioni del gioco e visualizzazione su schermo | Alta | Alta |
| L’utente può richiedere la visualizzazione della soluzione in ogni momento del gioco, che può avere nessuna, una o più soluzioni. Si può specificare il numero massimo di soluzioni da visualizzare e navigare sullo spazio delle soluzioni trovate per visualizzarle individualmente. Le soluzioni vengono calcolate automaticamente utilizzando la tecnica del backtracking. | | | |

***B.2 Requisiti non funzionali***

1. Salvataggio non solo della griglia di gioco iniziale ma anche degli inserimenti dell’utente (A che serve? Potrei in ogni caso salvare la griglia con tutti gli inserimenti e poi al caricamento se voglio la griglia vuota li cancello tutti)

2. Pulsante per cancellare tutti gli inserimenti

3. Pulsanti per annullare o ripetere l’ultima operazione effettuata sulla griglia di gioco

***B.2 Scenari d’uso dettagliati***

|  |  |
| --- | --- |
| **Creazione di un nuovo gioco** | |
| Tipo | Primario |
| Svolgimento normale | **1.** L’utente seleziona la voce del menu per la creazione di un nuovo gioco  **2.** L’utente specifica la dimensione della griglia di gioco desiderata  **3.** Il sistema presenta una griglia vuota costituita da celle individualmente selezionabili dall’utente  **4.** L’utente seleziona le celle per costruire i blocchi di uno schema di gioco fin quando ogni cella appartiene ad un blocco  TODO da estendere : l’utente seleziona celle di gioco non contingue, l’utente non seleziona almeno una volta tutte le celle della griglia di gioco, l’utente conferma la selezione e crea il blocco, tutte le celle appartengono ad un blocco per cui si può avviare la partita |
| Svolgimento alternativo | **1a.** Una griglia di gioco è stata precedentemente definita ed eventualmente riempita con dei numeri: il sistema chiede all’utente se vuole effettuare il salvataggio della griglia di gioco corrente prima di procedere alla creazione di un nuovo gioco |
| Descrizione | Questa operazione porta alla creazione di una griglia di gioco pronta per essere riempita con dei numeri dall’utente |

|  |  |
| --- | --- |
| **Salvataggio della griglia di gioco** | |
| Tipo | Primario |
| Svolgimento normale | **1.** Il sistema richiede all’utente la selezione di una directory di salvataggio  **2.** L’utente seleziona una directory e sceglie il nome del file in cui memorizzare la griglia di gioco corrente  **3.** Il sistema conferma l’avvenuto salvataggio |
| Svolgimento alternativo | **2a.**L’utente annulla l’operazione di salvataggio  **.1** Il sistema riporta l’utente davanti alla griglia precedentemente definita  **2b.** L’utente sceglie il nome di un file che esiste già  **.1** Il sistema propone la sovrascrittura del file |
| Descrizione | Questa operazione permette all’utente di memorizzare su file system una griglia di gioco precedentemente definita, compresi gli eventuali numeri con cui è stata completata |

|  |  |
| --- | --- |
| **Visualizzazione delle soluzioni** | |
| Tipo | Primario |
| Svolgimento normale | **1.** L’utente richiede tramite GUI la visualizzazione delle soluzioni del gioco corrente  **2.** Il sistema chiede all’utente il numero massimo di soluzioni da visualizzare  **3.** Il sistema blocca l’inserimento di nuovi numeri e la modifica dei numeri fino ad ora inseriti  **4.** Il sistema calcola le soluzioni e mette a disposizione dell’utente un sistema di navigazione per visualizzarle  **5.** L’utente scorre tramite GUI tra le soluzioni fornite |
| Svolgimento alternativo | **2a.**L’utente annulla l’operazione di richiesta delle soluzioni  **.1** Il sistema riporta l’utente davanti alla griglia precedentemente definita |
| Descrizione | Questa operazione permette all’utente di visualizzare dinamicamente le soluzioni del gioco, tuttavia termina implicitamente la partita corrente |

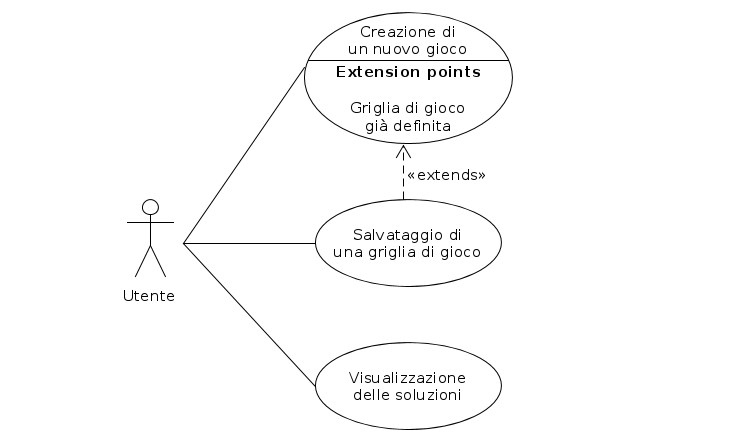
***B.4 Excluded Requirements***

*Descrivere i servizi eventualmente i esclusi, e spiegare il perchè*

***B.5 Assunzioni***

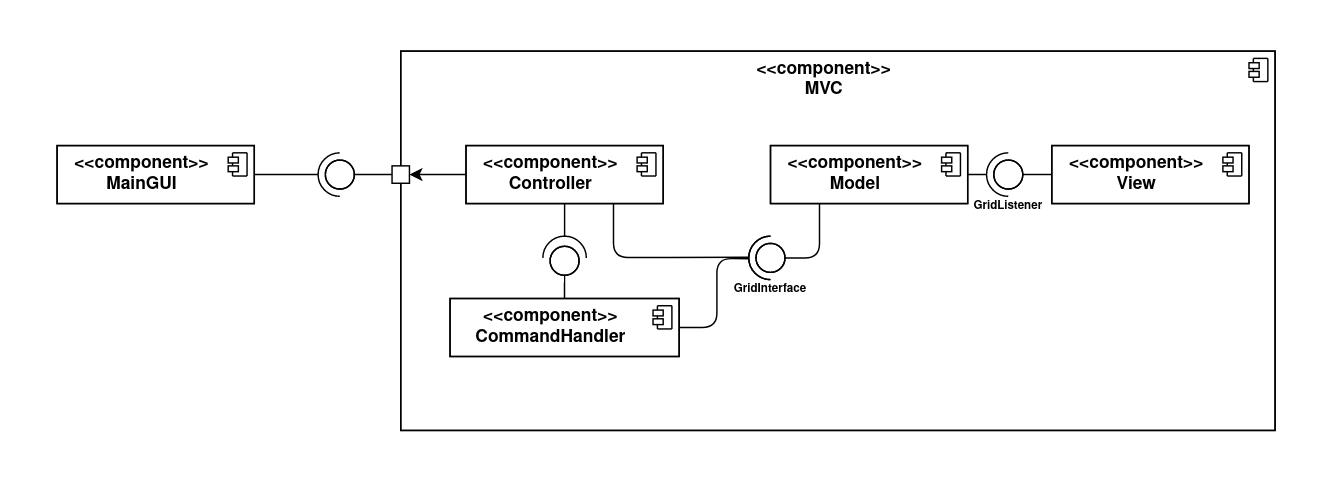
*<Briefly document, in this section, the most relevant requirement assumptions/decisions you had to made during your project>*

***B.6 Use Case Diagrams***

******

C. Architettura Software

***C.1 The static view of the system: Component Diagram***



***C.2 The dynamic view of the software architecture: Sequence Diagram***

D. Scelte Progettuali (Design Decisions)

|  |  |
| --- | --- |
| **N°** | **SCELTA PROGETTUALE** |
| 1 | Architettura MVC con design pattern Observer |
| 2 | Design pattern Command per l’interazione del controller con il model |
| 3 | Design pattern Memento per catturare lo stato interno della griglia di gioco in modo tale da poterlo ripristinare con operazioni di UNDO e REDO |
| 4 |  |
| 5 |  |

E. Progettazione di Basso Livello

F. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs) *<Report in this section how the architectural and low level design you produced satisfies the FRs and the NFRs>*

Appendix. Prototype   
*<Provide a brief report on your prototype, and especially: information on what you have implemented, how the implementation covers the FR and NFR, how the prototypes demonstrates your project correctness with respect to the FR and NFR. You may add some screenshots to describe what required above. Be ready to show your prototype during the oral examination>*