**“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** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 ha 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: 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 | Alta |
| Tramite GUI è possibile specificare le dimensioni della griglia di gioco, inizialmente vuota, e costruire in maniera dinamica, interattiva ed arbitraria lo schema di blocchi contigui su cui definire l’operazione aritmetica e il risultato da ottenere. Per fornire all’utente la massima libertà nella costruzione della griglia, i blocchi di celle contigue possono essere costruiti delimitando un lato alla volta e l’apposito riquadro in cui inserire risultato e simbolo aritmetico diventa visibile soltanto nel momento in cui il blocco risulta essere completamente delimitato. | | | |

| ID | SERVIZIO | IMPORTANZA | COMPLESSITA’ |
| --- | --- | --- | --- |
| #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 (binario?) in una cartella di salvataggio che può essere scelta dall’utente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| #3 | Inserimento dinamico di numeri all’interno della griglia di gioco | Alta | Bassa |
| Dal momento della sua creazione la griglia di gioco rimane interattiva, per cui è sempre possibile interagire con le singole celle per inserire e cancellare dei numeri. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| #4 | Controllo dinamico del soddisfacimento dei vincoli | Media | Media |
| 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, ma non corrette.  (TODO si esclude il controllo dell’operazione aritmetica: bisognerebbe tenere traccia di ogni singolo blocco) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| #5 | Calcolare le soluzioni del gioco e mostrarle a schermo | Alta | Alta |
| L’utente può richiedere la visualizzazione della soluzione in ogni momento del gioco: non è richiesto che la soluzione sia unica, pertanto il gioco può avere nessuna, una o più soluzioni. L’utente può specificare il numero massimo di soluzioni da visualizzare e navigare sullo spazio delle soluzioni trovate per visualizzarle singolarmente. | | | |

***B.2 Requisiti non Funzionali***

1. memorizzazione non solo della griglia di gioco iniziale ma dinamica (inclusi gli inserimenti dell’utente)

2. pulsante per cancellare tutti gli inserimenti

***B.3 Scenari d’uso dettagliati***

*Descrivere gli scenari più comuni, più interessanti, o più complicati d’uso dei vostri servizi.*

***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   
*<IF RELEVANT, Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, and their related Sequence Diagrams >*

***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)   
<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

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>*