**“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 di KenKen sia sul web, sul sito ufficiale <http://www.kenkenpuzzle.com/>, che su dispositivi mobili, tramite l’app KenKen Classic, sia su Android che su iOS.

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 automatica di giochi di KenKen, non è l’unico. Esempi degni di nota sono ….. TODO

Esistono altri siti web che permettono di completare interattivamente delle griglie di gioco, tuttavia i puzzles messi a disposizione sono creati a mano da esseri umani (es. <https://newdoku.com/>, <https://www.calcudoku.org/>)

B. Raffinamento dei Requisiti

*A partire dai servizi minimali richiesti, raffinate la descrizione dei servizi offerti dal vostro applicativo. Descrivete anche I requisiti non funzionali.*

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

*Descrivete in* ***dettaglio*** *i servizi offerti dal vostro Sistema, insieme a quelli che ritenete siano le soluzioni concettuali necessarie. In questa fase, non fate riferimento ad alcuna tecnologia specifica. Se volete, intervistate stakeholder e collezionate dati dal web o da altre sorgenti. Dovete acquisire una conoscenza avanzata dei problemi associate ai vostri servizi. Assegnate un ID a ciascun servizio. Prioritizzate inoltre I servizi in base a due scale: importanza alta, media, bassa. Complessità alta, media, bassa.*

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

*Elencare i requisiti non funzionali più’ importanti per il vostro Sistema*

***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. Dati e loro modellazione (se il sistema si interfaccia con un DBMS)

*Definite le sorgenti di dati a voi necessarie per realizzare I servizi di cui sopra. Modellate tali dati tramite un ER o similari. Specificate se e quali di tali dati sono gia’ forniti da applicativi esistenti.*

E. 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.>

F. Progettazione di Basso Livello

G. 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>*