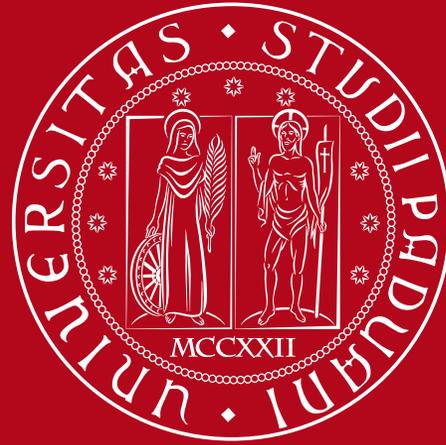


1222 \* 2022  
**800**  
ANNI



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Software Engineering  
project for the final course exam

luca boldrin

# Topics

Regole generali

Deliverables

Valutazione

Argomento del progetto

# REGOLE GENERALI

- Il progetto dovrà essere realizzato in Gruppo **composto da 4 persone**. Registrare il Gruppo in: <https://docs.google.com/spreadsheets/d/1spQG1z8OZ22IOZ1LgtwA8hRvBS9HyNILFp0txXfV50c/edit?usp=sharing> (ovviamente registrarsi anche su uniweb)
- Il Gruppo si deve registrare entro le date di consegna del progetto
- Utilizzare il linguaggio **Java** - Effettuare unit testing (utilizzando **JUnit**).
- La documentazione dovrà essere prodotta come gdoc, pagine wiki all'interno del Progetto, etc.
- Per i diagrammi UML si può usare un tool (websequencediagram, plantuml, miro, [draw.io](https://draw.io), ....)
- Si consiglia di utilizzare IntelliJ o Eclipse come IDE (si può usare maven per le dipendenze)

# CONSEGNA

- Alla consegna, inviare in ogni caso una **mail a [luca.boldrin@unipd.it](mailto:luca.boldrin@unipd.it)**
  - **in CC tutti i componenti del Gruppo**
- Creare Il progetto su Github (una [guida](#)). Invitare l'utente "luca-unipd" al GitHub.
- Scadenze:
  - 1 appello: 19 giugno 2023 - presentazione progetto e iscrizione entro **11 giugno alle 24:00**
  - 2 appello: 13 luglio 2023 - presentazione progetto e iscrizione entro **5 luglio alle 24:00**
  - 3 appello: 12 settembre 2023 - presentazione progetto e iscrizione entro **4 settembre alle 24:00**
  - 4 appello: 5 febbraio 2024 - presentazione progetto e iscrizione entro **28 gennaio alle 24:00**

Successivamente alla consegna ogni gruppo verrà convocato per discutere il Progetto in un orario preciso.

# CONSEGNA

## Verifiche finali prima della consegna

- rileggere attentamente le specifiche di progetto
- E' stato consegnato tutto quello che era richiesto nelle specifiche di progetto?
- I diagrammi sono completi? Nel diagramma delle classi tutte le associazioni hanno un nome e cardinalità? Le classi hanno gli attributi necessari?
- Il codice rispecchia la documentazione prodotta?
- Per eseguire il codice basta fare checkout del repository e seguire le istruzioni nel manuale di installazione?

# Topics

Regole generali

Deliverables

Valutazione

Argomento del progetto

# DELIVERABLES

6 deliverables fisici:

1. un **documento di specifiche** , → 4-5 pagine
2. un **documento di design** → 4-5 pagine
3. il **codice** (su github)
4. un **documento di system test** → 4-5 pagine
5. un **report di unit test**
6. un **manuale"** → 2-3 pagine

# DELIVERABLES

**documento di specifiche** , → 4-5 pagine

1. Use cases (diagrammi)
2. Descrizione tabellare di ciascun use case

# DELIVERABLES

**documento di design** → 4-5 pagine

- domain model, se è utile con una descrizione testuale
- System sequence diagram
- design class model
- Internal sequence diagrams

# DELIVERABLES

## **codice** (su github)

- Codice, files di compilazione, etc.
- opportunamente commentato
- Leggibile e compilabile con un IDE
- All'interno del codice ci devono essere anche le classi di test (junit)

# DELIVERABLES

## Documento di system test → 4-5 pagine

- Definizione dei system test case
  - di solito corrispondono 1-1 agli use cases. Vengono testati manualmente dall'interfaccia utente.
- System test report: è un doc scritto a mano. Template suggerito  
<https://www.ibm.com/docs/en/elm/7.0.3?topic=sections-test-case-template-reference> – Colonna SAFe

# DELIVERABLES

## **unit test** report

- Gli unit test case sono già nel codice, come classi di test. **Di solito corrispondono 1-1 alle classi software significative.**
- **il report è quello generato automaticamente da junit**

# DELIVERABLES

un **manuale**" →2-3 pagine

- Una descrizione ad alto livello del Progetto (1 pag max)
- le istruzioni su come installare e lanciare il software.
- Indicazioni su ambienti di esecuzione, vincoli su version java, etc.
- Un'indicazione delle principali funzioni riutilizzate da librerie esistenti (secluse quella banali, log4j, java.utils....)

# Topics

Regole generali

Deliverables

Valutazione

Argomento del progetto

# VALUTAZIONE

La valutazione del progetto avverrà tenendo conto dei seguenti punti:

- La realizzazione delle specifiche funzionali;
- L'organizzazione e la leggibilità del codice;
- L'adeguatezza della documentazione allegata al progetto
- La corretta definizione dei test
- La discussione del Progetto

La **valutazione individuale** dipende dall'efficacia di ciascuno nella presentazione del Progetto e da alcune domande individuali sulla parte teorica.

# Topics

Regole generali

Deliverables

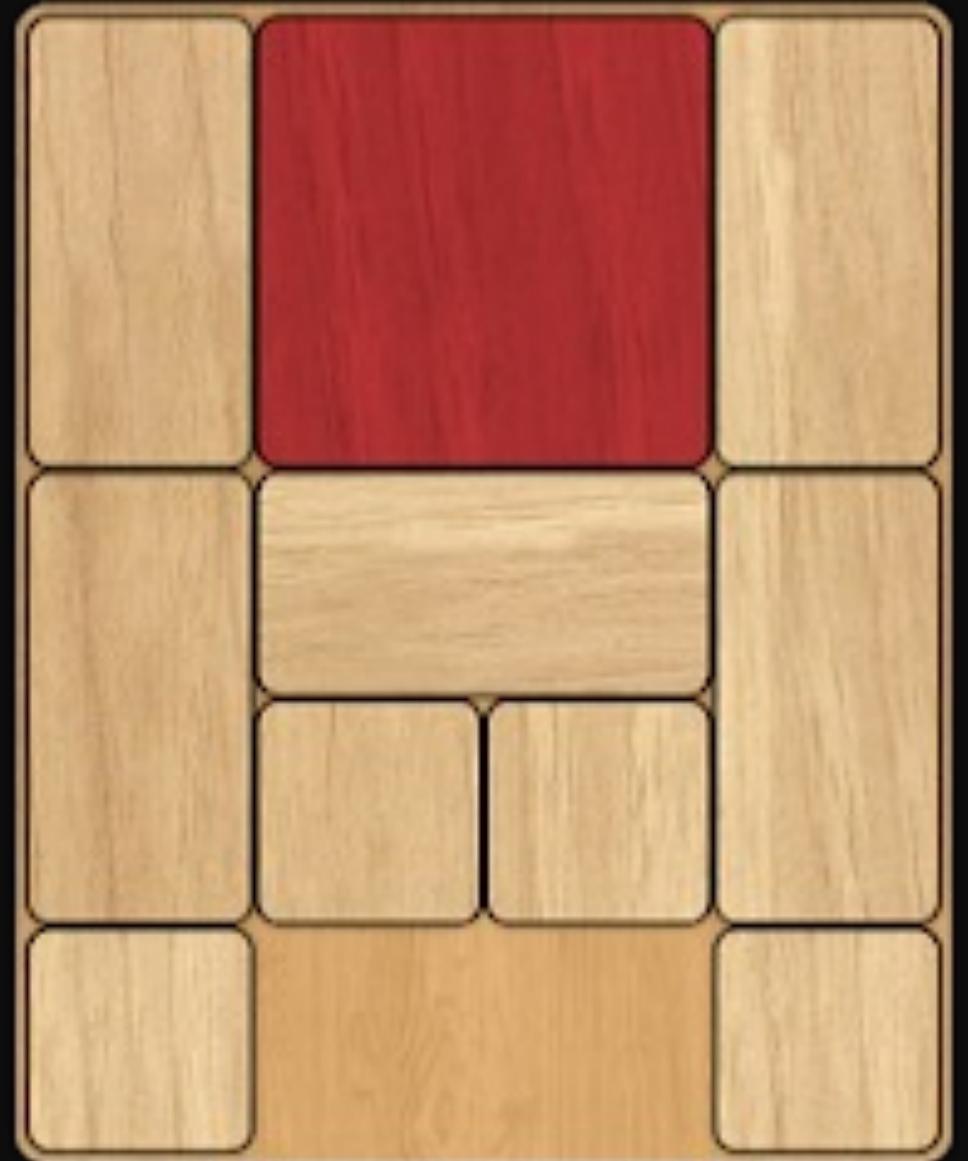
Valutazione

Argomento del progetto

# PROGETTO KLOTSKI

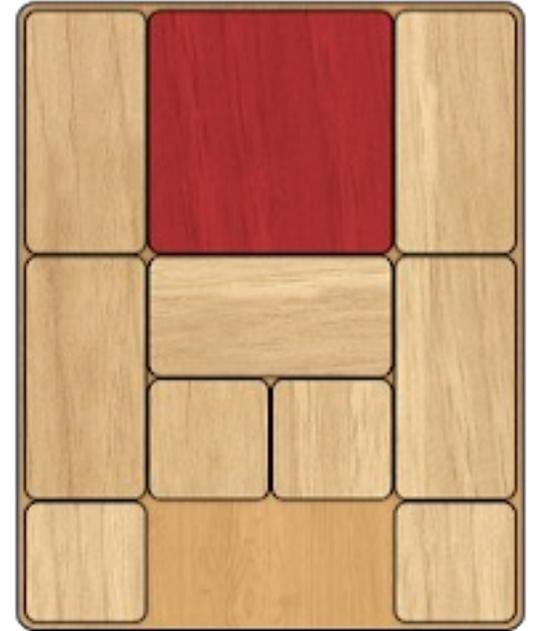
---

Sliding puzzle



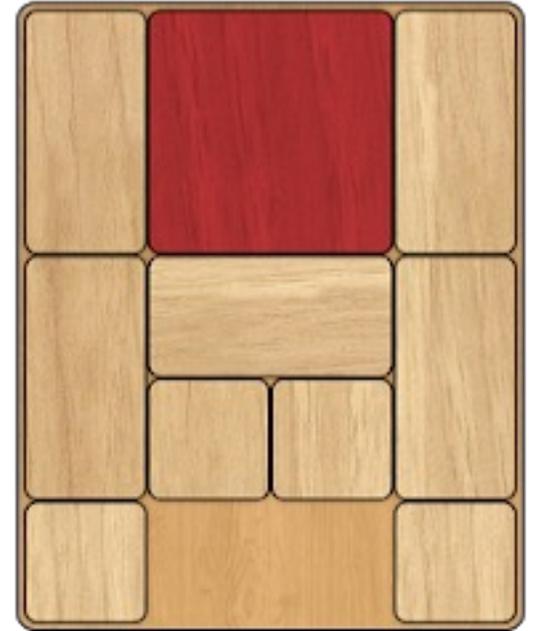
# Funzionalità

- Interfaccia che consente di muovere i blocchi nelle posizioni consentite
- Possibilità di scegliere alcune diverse configurazioni di partenza
- Funzione di reset
- Funzione di undo – fino all’inizio
- Funzione «next best move»
- Counter delle mosse effettuate
- Possibilità di salvare lo stato corrente e di ripristinarlo



# Implementazione

- Codice Java (per la grafica javaFX o altro ....)
  - eventualmente javascript
- usare, quando è il caso, i design pattern
- Per la persistenza,
  - storage su file è ok
  - DB (locale o su AWS) benissimo
- Risolutore (next best move)
  - Non serve un algoritmo intelligente, basta caricare su DB le soluzioni di **una configurazione** (non è necessario che sia in grado di rispondere in tutti i possibili casi, ma deve consentire almeno di fare un percorso dall'inizio alla fine)
  - preferibilmente come rest service (e.g. <https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-create-api.html>)
  - Ma va bene anche una classe nell'applicazione che gestisce le soluzioni
- Domain e design model accurati



# Risorse

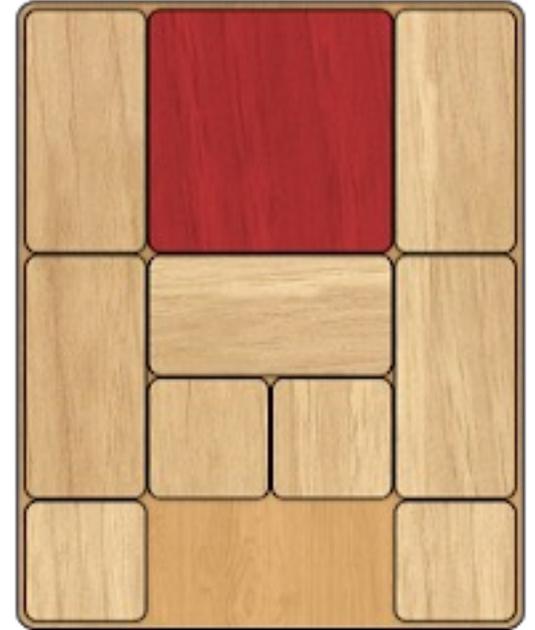
<https://github.com/SimonHung/Klotski>

<https://josephpetitti.com/klotski>

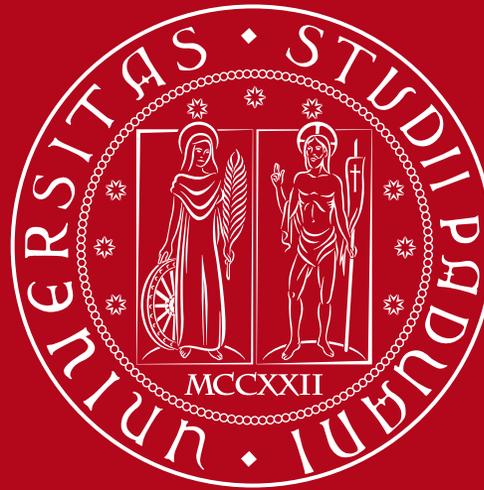
<http://simonsays-tw.com/web/Klotski/game/klotskiDemo.html>

<https://klotskisolver.sourceforge.net/>

(and many others... <https://github.com/topics/klotski>)



1222 \* 2022  
800  
ANNI



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA