

---

# Laboratorio di Programmazione Internet: progetto a.a. 2022/2023

---

Angelo Di Iorio  
Roberto Amadini  
Marco Ferrati

# SPACCA - Sfidare Parenti e Amici Con Carte Assurde

- SPACCA è un'applicazione stand-alone in Java FX per **giocare a carte**:
  - in modalità **multiplayer**, con **giocatori umani e robot**
  - in **partite** singole o **tornei**

# Architettura SPACCA

- SPACCA è un'applicazione JavaFX stand-alone:
  - l'amministratore lancia l'applicazione e gestisce le carte e tutte le configurazioni
  - l'applicazione memorizza i dati in locale, su uno o più file, e li carica/modifica in modo opportuno
  - i giocatori giocano utilizzando la stessa istanza dell'applicazione
- NOTA: è una (iper) semplificazione, un'applicazione reale dovrebbe essere distribuita ed eseguita su un server

# Ambiente di gioco

- L'applicazione è pensata per essere usata in un ambiente controllato (simile ad una sala da bowling):
  - Amministratore crea partita e aggiunge i giocatori → genera un codice per la partita
  - I giocatori entrano nella partita usando quel codice
  - I giocatori si alternano seguendo le indicazioni dell'applicazione (senza imbrogliare!)



# Workflow Partita

- L'amministratore si autentica (tramite username e password)
- L'amministratore crea partita -> genera un codice per la partita e aggiunge i nomi dei giocatori
- L'amministratore fa logout o chiude l'applicazione
- I giocatori entrano nella partita usando il codice (dopo aver ri-eseguito l'applicazione, se non ancora aperta)
- I giocatori si alternano seguendo le indicazioni a video
- In ogni momento i giocatori possono sospendere la partita e ripartire dallo stesso punto, rientrando con lo stesso codice
- A fine partita il punteggio viene storicizzato (su file) e mostrata la leaderboard

# Regole del gioco SPACCA

- Le regole esatte del gioco le decide il gruppo ma ci sono alcuni vincoli OBBLIGATORI:
  - Gioco di carte, eventualmente supportato da una board
  - Almeno 2 giocatori
  - Giocatori umani e/o robot (anche stupidi!)
  - Non deve essere un gioco esistente (ma si può prendere spunto)

# Carte SPACCA

- Anche i dettagli delle carte sono a discrezione del gruppo:
  - Numero di carte
  - Punteggio e caratteristiche
  - Argomento ed eventualmente personaggi
  - Resa grafica
  - ...
- Obbligatorio includere almeno 2 carte “imprevisto o probabilità”, ad esempio “scarta tutte le altre”, “vinci la partita se...”, “scambia carte con avversari”



# Giocatori in SPACCA

- Una partita in SPACCA può includere giocatori umani e/o robot
- NON è ammesso implementare un solitario, sono ammesse partite in cui tutti i giocatori sono umani ma anche tutti robot (in questo caso l'interfaccia dovrà permettere di far giocare le carte ed andare avanti nella partita)
- I Robot possono essere:
  - *Stupid*: scelgono le mosse in modo random ma rispettano le regole del gioco
  - [Opzionale] *Smart*: implementano una logica più sofisticata, i cui dettagli sono a scelta del gruppo
- E' ammesso quindi anche avere solo robot "stupidi" ma molto apprezzato implementare robot più sofisticati



# Modalità: partita singola e torneo

- SPACCA può essere usato sia per partite singole che per tornei
- Ad entrambi si accede con un codice creato dall'amministratore
- Le regole del torneo sono a discrezione del gruppo (gironi, eliminazione diretta, play-off, ecc.)
- L'applicazione permette di avanzare nel gioco e nel torneo, e di salvare lo stato della partita/torneo per ripartire dallo stesso punto

# Workflow torneo

- L'amministratore si autentica tramite username e password
- L'amministratore crea il torneo:
  - genera un codice per quel torneo
  - aggiunge i nomi dei giocatori e le partite
- I giocatori riaprono l'applicazione sulla stessa macchina ed entrano nella partita usando il codice
- Il programma mostra chi deve giocare e in quale partita
- Ogni partita segue le regole e i vincoli visti nelle slide precedenti
- Anche un torneo può essere interrotto e ripristinando usando il codice del torneo
  - [opzionale] Il gruppo può scegliere se permettere l'interruzione e il ripristino di partite singole in uno stesso torneo

# Riassumendo: ruolo giocatore

## ■ Giocatore:

- E' identificato da uno username
  - Per semplicità: se un amministratore aggiunge lo stesso nome a partite diverse, questo viene riconosciuto come stesso utente (per la leaderboard)
- Non si autentica ma entra nella partita/torneo con un codice
- Gioca partite e tornei, e vede i risultati
- Può vedere la leaderboard in qualunque momento

# Riassumendo: ruolo amministratore

## ■ Amministratore:

- Si autentica con username e password
- Crea una partita o un torneo
- Aggiunge utenti (nuovi o già esistenti) alle partite/tornei
- Crea un nuovo utente o modifica il profilo di un utente
- [Opzionale]: crea/modifica carte e parametri del gioco
- Elimina partite/tornei in sospeso

# Gestione base di dati e file

- SPACCA memorizza carte, partite e tornei su filesystem
- Ogni gruppo sceglie il formato dati da usare e le librerie per fare il parsing
- Necessari ovviamente controlli di consistenza sui dati
- Ogni gruppo sceglie la strategia che preferisce per gestire gli errori purché ben documentata e consistente
- NOTA: anche questa è una semplificazione, per un problema reale avremmo usato un database
- [Opzionale] E' ammesso usare un database o un ORM

# Qualche libreria

- Esistono diversi formati per serializzare i dati: CSV (Comma Separated Values), JSON (Javascript Object Notation), XML (eXtensible Markup Language), *custom*
- Qualche libreria utile in Java (non vincolante!):
  - JSON serializer: <https://github.com/google/gson>
  - XML serializer: <https://www.edureka.co/blog/serialization-of-java-objects-to-xml-using-xmlencoder-decoder/>
  - CSV serializer: <http://opencsv.sourceforge.net/>



# Regole e vincoli

# Requisiti obbligatori ed estensioni

- Queste specifiche sono poco vincolanti di proposito, siete invitati a personalizzare e aggiungere funzionalità
- E' richiesto tuttavia implementare **tutte le funzionalità indicate come obbligatorie** nelle slide precedenti
- **Le funzionalità aggiuntive se sensate e ben implementate migliorano la valutazione**



# Qualche idea (non vincolante) per estensioni

- Comunicazione tramite mail/telegram/... dei risultati di un torneo o di una partita → bisogna tenere traccia oltre che dello username anche di un recapito per gli utenti
- Extension pack: l'amministratore può aggiungere carte a disposizione nell'applicazione
- Multi-lingua: il gioco è in lingue diverse, configurabili dall'amministratore
- Il gioco di carte usa un tabellone, per muovere i giocatori in base alle carte e all'andamento della partita
- ...

# Gruppi

- Il progetto si svolge in gruppi da **2 o 3** persone
  - In via eccezionale si può svolgere in gruppo da 4 ma proponendo un'estensione alle specifiche e previa approvazione
  - Non sono ammessi progetti individuali, salvo casi eccezionali giustificati e approvati singolarmente
- **Queste eccezioni devono essere comunicate in anticipo e non sono valide per la sessione di gennaio/febbraio 2024**
- *Essendo un lavoro di gruppo è caldamente consigliato usare GitHub o altri repository per versionare e condividere codice*

# Discussione e consegna

- Il progetto si discute SOLO dopo aver superato la prova di laboratorio
- Alla discussione del progetto devono essere presenti **tutti** i membri del gruppo, salvo casi eccezionali
  - Es. Borsa di studio o gravi impedimenti
- Il progetto si consegna **prima** della data di discussione (~5 giorni prima) su [virtuale.unibo.it](http://virtuale.unibo.it)
- **Queste specifiche solo entro il 31/5/2024.**
- Discussioni in presenza o da remoto. Appelli: **14/06, 06/07, 25/07, 13/09**
  - *Iscrivere una sola persona per gruppo su AlmaEsami*

# Consegna

- Il progetto si consegna su Virtuale. Prima di ogni appello saranno pubblicate apposite sezioni per la consegna con le istruzioni
- Cosa consegnare
  - Sorgenti Java. [Opzionale] link e accesso al repository usato per sviluppare il codice
  - JAR eseguibile (provatelo su computer diversi!) e istruzioni/parametri per eseguirlo [Opzionale]
  - **Breve** relazione: max 3 pagine, con qualche screenshot dell'applicazione e una sequenza step-by-step di istruzioni per avviarla (in formato txt/md/pdf, no .odt o .docx)
  - Video illustrativo di 2 minuti: mostra l'applicazione in funzione ed è usato solo come backup nel caso non si riesca ad eseguire il codice consegnato
  - Alcuni dati precaricati: utente amministratore e tutti i dati che ritenete utili per mostrare il funzionamento del software

# Valutazione progetto

- Le funzionalità aggiuntive aiutano ad ottenere un voto più alto
- L'applicazione deve:
  - Soddisfare i requisiti obbligatori
  - Fornire qualche funzionalità aggiuntiva
  - Essere usabile e con una chiara gestione degli errori
  - Essere ben strutturata e flessibile
  - Funzionare
- Soprattutto: deve essere frutto del vostro lavoro, è ammesso usare librerie esterne ma dovete essere in grado di spiegare/modificare il codice all'esame
- La valutazione non è di gruppo ma singola. Lo stesso progetto può portare a voti diversi in base al contributo di ognuno

# Valutazione progetto

## VOTO FINALE $\cong$ VOTO LABORATORIO $\pm 3$

- Il voto di laboratorio è la **base** con cui presentarsi alla discussione del progetto
- Il **progetto in sé non ha un voto** proprio ma è un “*aggiustamento*” del voto di laboratorio
  - Salvo **eccezioni**, a seconda del progetto **e** della sua discussione orale il voto di laboratorio può essere:
    - **confermato**
    - **decrementato** da 1 a 3 punti
    - **incrementato** da 1 a 3 punti
- Di conseguenza, una volta discusso il progetto non è possibile rifare l'esame di laboratorio

# Domande?

- Alcune domande discusse fine a lezione:
  - Anche l'amministratore può giocare?  
*Sì, senza fare login. Esiste uno username associato all'amministratore che può essere aggiunto a partite e tornei*
  - Si possono usare carte esistenti (ad esempio, da poker) in giochi inventati?  
*Sì, l'importante è che le regole del gioco siano personalizzate e che ci siano carte imprevisi/probabilità*
  - Le carte imprevisi/probabilità possono non essere usate durante il gioco?  
*Sì, l'importante è che il gioco preveda queste carte e che sia possibile pescarle. Si potrebbe poi dare ai giocatori la possibilità di non usarle o di usarle in un secondo momento*