

Prova Finale Ingegneria del Software

Alessandro Margara

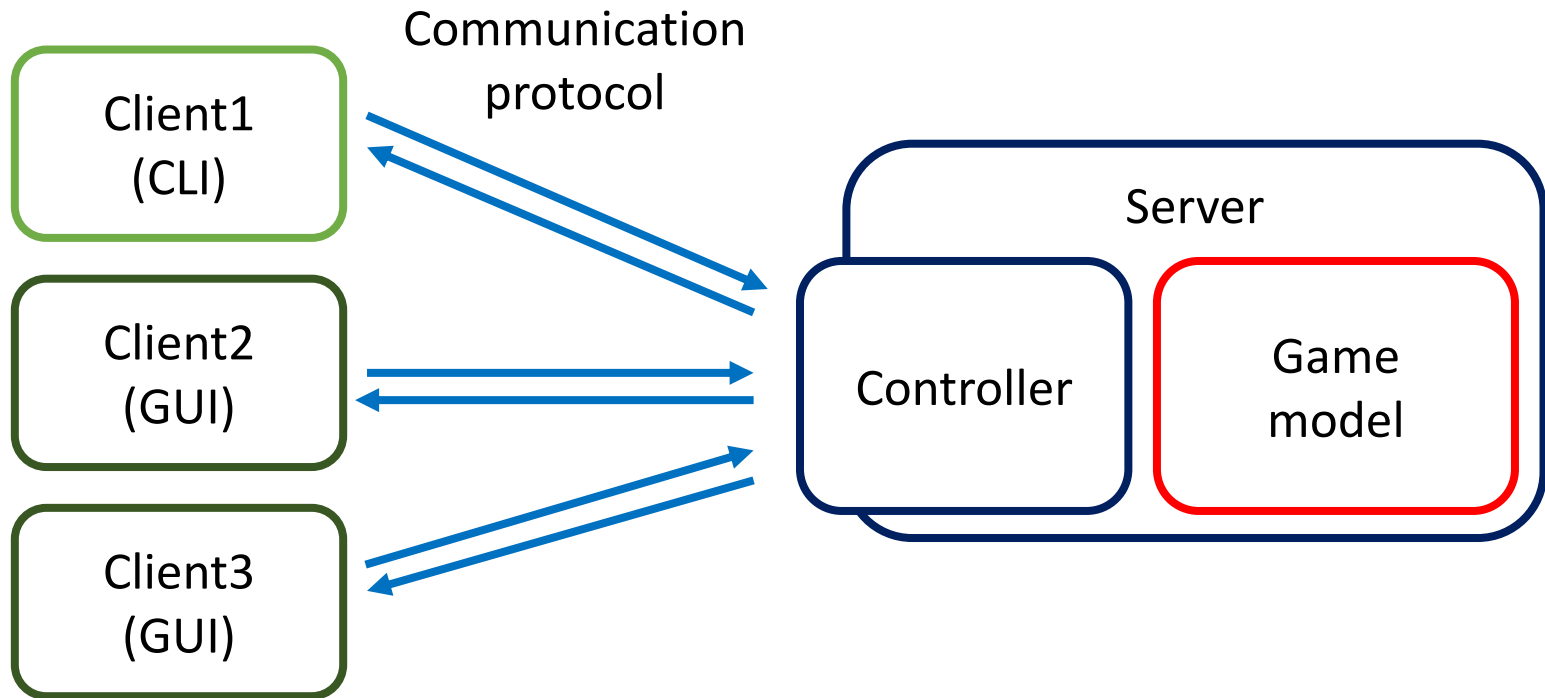
Team

- Titolare: Alessandro Margara
- Esercitatore: Gian Enrico Conti
- Responsabili di laboratorio: Gian Enrico Conti
Barbara Schinaia
- Tutor di laboratorio: Stefano Fossati
Ettore Zamponi

Obiettivi

- Mettere in pratica quanto imparato durante il corso di ingegneria del software
- Focus: processo di sviluppo del software
 - Progettare, verificare, documentare ed implementare un'applicazione software mediante un approccio di tipo object-oriented
- Cosa: applicazione multi-utente, distribuita, con interfaccia grafica
 - Software che emula un gioco da tavolo

Progetto



Il gioco



- <https://www.craniocreations.it/prodotto/my-shelfie>

Progetto

- Il documento con i requisiti dettagliati e le scale di voti verrà diffuso entro il primo laboratorio

Strumenti

- Ambienti di sviluppo (IDE): IntelliJ
- Software di gestione di progetti: maven
- Software di version control: git
- Software per testing: JUnit

Calendario

| Data | Attività | Argomento da presentare |
|-------|---------------|--|
| 21/02 | Intro | Introduzione al corso |
| 28/02 | Esercitazione | Tools (IntelliJ, Maven, Git, JUnit) |
| 07/03 | Esercitazione | Presentazione progetto + architettura (distributed MVC) |
| 14/03 | Lab 1 | Design (UML) del model |
| 21/03 | Lab 2 | Design (UML), implementazione e testing di model (e controller) |
| 28/03 | Esercitazione | Protocollo di comunicazione |
| 04/04 | Lab 3 | Implementazione e testing di model e controller |
| 11/04 | Break | Vacanze Pasqua |
| 18/04 | Lab 4 | Definizione del protocollo di comunicazione (UML sequence diagram) |
| 25/04 | Break | Vacanze 25 aprile |
| 02/05 | Esercitazione | GUI |
| 09/05 | Lab 5 | Implementazione del protocollo di comunicazione |
| 16/05 | Lab 6 | GUI |
| 23/05 | Lab 7 | GUI |
| 30/05 | Lab 8 | Progetto completo |

Frequenza

- La frequenza è obbligatoria!
- Almeno 3 persone per gruppo presenti a ogni lab
- Massimo 3 assenze individuali
- Deroghe solo per giustificati e documentati motivi
 - Lavoratore a tempo pieno
 - Erasmus / scambi internazionali
 - Motivi di salute

Gruppi

- Gruppi formati da esattamente 4 persone
 - In caso di problemi a trovare colleghi, potete usare Slack
- I gruppi dovranno essere formati e comunicati entro martedì 28 febbraio, compilando il seguente documento
 - <https://tinyurl.com/ingsoft23-groups>
 - Per richiesta di gruppi inter-sezione aggiungete un commento nella colonna *note*
 - Studenti da massimo due sezioni
 - Referente del gruppo deve appartenere alla sezione
- Entro martedì 7 marzo provvederemo a creare gruppi con eventuali studenti rimanenti

Discussione in Lab

- I gruppi verranno ascoltati a turno da responsabili, tutor e docente
 - Fare domande
 - Mostrare avanzamento lavoro
- Tutti i gruppi verranno ascoltati a ogni laboratorio
 - Forniremo un calendario di massima prima di ogni laboratorio

Slack

- Forum ufficiale delle 3 sezioni per discutere con colleghi e responsabili
 - Per chiarire dubbi relativi a specifiche, strumenti, ...
- Potete contribuire alla discussione di dubbi
 - La partecipazione verrà considerata positivamente durante la valutazione
- L'interazione con docente, responsabili e tutor al di fuori dell'orario di laboratorio deve avvenire esclusivamente attraverso questo strumento

Slack

- Link di invito
 - <https://tinyurl.com/ingsoft23-slack>
- Regole
 - Iscrivorsi con la propria mail istituzionale
 - Usare i canali predefiniti per porre domande
 - Ogni domanda costituisce un thread
 - Rispondere, commentare, aggiungere contributi a quella domanda all'interno del suo thread
 - Non postare una nuova domanda prima di aver verificato che non sia già coperta da altre
 - Docenti e responsabili provvederanno a eliminare domande duplicate

Peer review

- Ci saranno due momenti in cui ogni gruppo revisionerà il progetto di un altro gruppo
 - Design UML e architettura
 - Protocollo di comunicazione
- I docenti comunicheranno l'associazione tra gruppi
- Una revisione consisterà in due elenchi
 - Suggerimenti per il gruppo revisionato
 - Idee / aspetti che il gruppo revisionante ha appreso e potrebbe integrare nel proprio progetto
- I documenti di revisione faranno parte della documentazione inclusa nel progetto

Valutazione

- Voto registrato alla fine del laboratorio (luglio)
 - Il progetto ***deve*** essere sviluppato durante questo semestre
 - La prova ***non*** può essere recuperata
 - ***Non*** esistono appelli durante l'anno: se non viene superato occorre rifrequentare l'anno successivo

Valutazione

- Il progetto andrà consegnato entro una data che indicheremo nel documento di requisiti (fine giugno)
 - Attenzione! Dopo la fine dei laboratori non avrete più supporto: finite prima!
- Discussione orale a inizio luglio, basata su una demo del progetto con presentazione, analisi del codice e della documentazione
 - Tenendo conto della partecipazione “in itinere” da parte della squadra
 - Il progetto si presenta con almeno due PC connessi in rete
- Voto in 30-esimi, come per ogni altro corso
 - Contribuisce, con gli altri corsi di tipo prova finale, all’incremento del voto di laurea ($I = 1 + V/6$)

Valutazione: requisiti

| Requisiti | Voto massimo |
|--|--------------|
| Regole semplificate + comunicazione + CLI | 18 |
| Regole complete + comunicazione + CLI | 21 |
| Regole complete + comunicazione + GUI | 24 |
| Regole complete + comunicazione + GUI + 1FA | 27 |
| Regole complete + comunicazione + CLI + GUI + 1 FA | 30 |
| Regole complete + comunicazione + CLI + GUI + 2 FA | 30 L |

FA = Funzionalità avanzate

Valutazione: criteri

Oltre alle funzionalità implementate, il voto finale considererà i seguenti aspetti

- Qualità della progettazione
 - Un uso appropriato di interfacce, ereditarietà, composizione tra classi
 - Uso di pattern (statici, di comunicazione e architetturali)
 - Divisione delle responsabilità
- Stabilità dell'implementazione
- Conformità alle specifiche

Valutazione: criteri

- Leggibilità del codice
 - Nomi di variabili, metodi, classi, package
 - Commenti e documentazione JavaDoc (preferibilmente in inglese)
 - Assenza di codice ripetuto e metodi di eccessiva lunghezza
- Efficacia e copertura dei casi di test
 - Il nome e i commenti di ogni test dovranno chiaramente specificare le funzionalità testate e i componenti coinvolti
- Utilizzo degli strumenti (IntelliJ, Maven, Git, ..)

Valutazione: criteri

- Conoscenza di tutto il codice da parte di tutti i componenti del gruppo
- Autonomia, impegno e comunicazione (con i responsabili e all'interno del gruppo)
- Contributi di discussione
 - Interventi su slack, qualità delle peer review

Warnings

- Qualunque gruppo che venga trovato a copiare verrà automaticamente considerato insufficiente ...
- ... e dovrà ripetere il corso l'anno successivo
- È vietato copiare blocchi di codice da altri, da altre sezioni, da soluzioni trovate su Internet
 - Sistema di controllo molto efficace:
<https://theory.stanford.edu/~aiken/moss/>

Warnings

- L'uso di librerie esterne *deve essere approvato* dai responsabili di laboratorio
- In generale, saranno accettate librerie utili a fornire funzionalità aggiuntive ...
- ... ma non librerie che automatizzino l'implementazione di intere parti dell'applicativo

Warnings

- In particolare, è vietato l'uso di tool che facilitino la creazione della GUI
- Per la GUI potete utilizzare Swing oppure JavaFX
 - Entrambi verranno presentati a esercitazione
 - Non sono consentiti tool che generino automaticamente codice Java per l'interfaccia

Prima settimana

- Prima della prossima settimana, dovrete iniziare a installare i tool (IntelliJ, Maven, Git) seguendo le istruzioni presenti nel seguente video
 - <https://web.microsoftstream.com/video/07e40f42-df34-45a3-8683-20d5f8b639ff>
- Postate dubbi e problemi su Slack
 - Responsabili, tutor e docenti risponderanno alle vostre domande
 - Leggete domande di altri studenti prima di postare duplicati
 - Se potete, rispondete ad altri studenti
 - Responsabili, tutor e docenti confermeranno le risposte valide
- Obiettivo: entro il 7 marzo tutti gli studenti avranno una corretta installazione e configurazione dei tool e potranno iniziare a lavorare
- Se siete un gruppo standard (singola sezione) potete iniziare a creare il vostro repository

GitHub

- Ogni studente deve creare un account personale GitHub student
 - Con e-mail istituzionale
 - Fatelo subito
- Una volta stabilito il gruppo di appartenenza, uno dei membri del gruppo crea un repository privato denominato
IS23-AMxx
(dove xx è il numero del gruppo)
- Invitate al repository tutti i membri del gruppo e i responsabili

GitHub

- Vi viene inoltre richiesto di creare la cartella “deliveries” nella root del repository
- In questa cartella dovrete periodicamente caricare i prodotti intermedi richiesti
 - Diagrammi UML (iniziale e finale)
 - Documentazione (e.g., protocollo di rete)
 - Documenti di peer review
 - Report di copertura casi di test

GitHub

- Warning: inserite solo commit significativi
- La qualità dei commit verrà considerata per la valutazione
 - Qualità in termini di titoli, commenti, chiarezza circa i cambiamenti effettuati
 - Evitate di committare ogni singola modifica del codice!

Maven

- Ci aspettiamo un progetto Maven
 - Compilabile
 - Testabile
 - Eseguitibile
 - Senza errori

Domande?

- Link per registrare i gruppi
 - <https://tinyurl.com/ingsoft23-groups>
- Link per accedere a Slack
 - <https://tinyurl.com/ingsoft23-slack>
- Link al tutorial di installazione strumenti
 - <https://web.microsoftstream.com/video/07e40f42-df34-45a3-8683-20d5f8b639ff>