INTRODUCTION

While the web is central to much of software development today, there are many other subfields of computer science to be considered. From concurrency, to networking, to artificial intelligence, the world of software development is vast and varied.

PROJECT OUTLINE

Create a desktop application with Java and JavaFX to facilitate a two player abstract board game. The game should support play over two computers and have an AI player option.

Background and Context:

The application you will be creating will users to play a two player abstract board game through a JavaFX GUI. The two-player game will be implemented through a client-server model, which makes use of threads for concurrency and Java sockets for networking. Additionally, an AI option must be available and implemented using the minimax algorithm.

What you will do:

You are in charge of creating the desktop system using Java and JavaFX. The system is actually made up of two separate programs: The server program and the client program. The server program is responsible for listening for requests from clients and pairs them off into games. The client programs connect to the server and display the JavaFX interface of the board to each player.

The client should allow three game modes:

- Standard Mode
    - This is a simple game, played on one computer.
    - Players sit next to each other and take control of the mouse when it is their turn to play.

- Networked Mode
    - This is a game player over 2 computers.
    - Each client is matched to a second client through connection to a server.
    - This mode requires the use of threads and sockets.
- AI Mode
    - This is a game for one human player and one AI player played over one computer.
    - It makes use of the minimax algorithm.

What you will be given:

A link to a list of possible board game choices. This is a guideline, rather than a full list.

PROCESS

This section of the brief contains helpful guidance and information concerning your process. You will be technically guided through each phase of the project.

Phase 1: Planning

Screen planning

Design and plan the layout of your application's screens and navigation. Consider the layout and navigation system so that users can easily use the application and its functionality. Consider, also, how users will interact with the board. Record all assumptions, decisions, and reasoning in your rationale.

Phase 2: Implementation Phase

Game Implementation

Start by implementing the standard game. Once this is complete, use inheritance to add a networked game and an AI game. The networked mode also requires an external server socket system. Record all assumptions, decisions, and reasoning in your rationale.

Interface Implementation

Create a desktop interface following your designs from phase 1 using JavaFX. Ensure that it allows for all the functionality specified in the project outline.

Phase 3: Testing phase

Manual Testing

Test your application on a desktop machine.

Validation and Feedback

Make sure that the screens communicate information to the user effectively. Any input errors must be considered and communicated, and users should be guided through the functionality of the application.

Quality Control

Make sure the screens of your application are consistent, follow your designs, display all the information that they should, and support all the functionality that they should.

Phase 4: Final Phase

Documentation

Go back to your rationale document and polish it. This document should include all your assumptions, decisions, and reasoning. Include a class diagram of your project in the appendix.

Presentation

You must present your deliverables to your lecturer and the class.

DELIVERABLES

Project files

The project folder including all of your Java files and resources as well as your `pom.xml` manifest file and a self-executing JAR.

Rationale Document

This document should be delivered as a PDF. It must include all your assumptions, design decisions, planning and reasoning. The appendix must include a diagram.
In addition to the soft copy you submit to the Google Classroom, you must also submit a printed hard copy before presenting your project.

Plagiarism Document

You must hand in a signed plagiarism document in both hard copy and soft copy.

SUBMISSION

- Late submissions will be accepted for a maximum of 48 hours after the allocated time of the deadline, as outlined in the brief. Any late work submitted during this 48 hour grace period will be marked by the lecturer as usual, however the work submitted will be awarded a maximum of 50% as a late-submission penalty
- After 48 hours, 0% will be awarded. OW will take a zero-tolerance stance on work submitted later than 48 hours after the deadline
- Extensions may be applied for BEFORE the deadline only, and should be done via a formal application by the student, which will be recorded on an institutional list where all lecturers can see which students are applying for extensions, and why
- Submission and presentation will take place in Week 8
- Projects should be digitally submitted on Google Classroom
- All deliverables must be placed in a single zipped archive
- The zipped folder must follow the naming convention **`IDV304_xxxxxx_yyyyyy.zip`** where **`xxxxxx`** is your first name and **`yyyyyy`** is your student number

OUTCOME

By completing this project, learners will have learned to:

- Create a concurrent application with threads
- Communicate through a network with sockets
- Implement basic artificial intelligence with the minimax algorithm
- Document an application implementation
- Professionally present projects
- Manage their time

ASSESSMENT

Learners will be assessed on their:

- Data analysis abilities
- Planning
- Problem solving skills
- Decision making abilities
- Tool use
- Technical skill
- Technical understanding
- Correctness in implementation
- Usage considerations
- Device and hardware considerations
- Initiative and brief expansion
- Layout, communication and aesthetics
- Documentation
- Professionalism

**Please see the attached mark sheet for a detailed breakdown of the assessment weightings.*

SCHEDULE

WEEK	DATE	CLASS WORK	MILESTONES
1	2018-09-17 to 2018-09-21	• Standard game implementation	• Planning Phase
2	2018-09-24 to 2018-09-28	• Networking architecture • Concurrency	• Planning Phase • **Homework Assignment 1**
3	2018-10-01 to 2018-10-05	• Multi-computer networking	• Planning Phase • **Homework Assignment 2**
4	2018-10-08 to 2018-10-12	• Heuristics	• Implementation Phase • **Homework Assignment 3**
5	2018-10-15 to 2018-10-19	• Minimax	• Implementation Phase • **Progress Milestone 1**
6	2018-10-22 to 2018-11-26	• Tree depth	• Implementation Phase • **Progress Milestone 2**
7	2018-10-29 to 2018-11-02	• Documentation	• Testing Phase • **Progress Milestone 3**
8	2018-11-05 to 2018-11-09	• Submission, demonstration, and assessment	• Final Phase
9	2018-11-12 to 2018-11-16	• Submission, demonstration, and assessment	• Final Phase