

Week 21-25 | Programming Project Assessment

Hand in by: May 23, 2022 (via Moodle)

Part 1: Code and Project Documentation

Write a Java program for *one* of the given topics. It must possess a Swing-based GUI and follow an object-oriented programming paradigm. Your code must be compilable and runnable without errors.

Submission Requirements. Upload the following via Moodle on Mondays (23:59 CEST):

1. Written summary of your requirements analysis (~1 page) – **by May 9, 2022**
2. Complete class diagram for your program (PDF file) – **by May 16, 2022**
3. Complete source code for your program (ZIP file) – **by May 23, 2022**
4. Written summary of your testing strategy and results (~1 page) – **by May 23, 2022**

Part 2: Demonstration

In the week 25 exercises (**May 25, 2022**), you will demonstrate and discuss your work in a 5-minute timeslot. Think carefully about how to describe your code structure. Your lecturer will review your submitted code and project documentation and ask questions about it, so be prepared for this.

Note that questions may refer to all parts of the code or project documentation. With one or more elements of part 1 being not submitted, you will not be able to answer all questions successfully.

General Rules

1. Part 1 of the assessment must be handed in before you can begin part 2. If you perform a demonstration while your code has not been uploaded yet, the demonstration will not be counted.
2. Late submissions are subject to downgrading following the General Regulations for Assessment & Award of Lancaster University's Manual of Academic Regulations and Procedures¹.
3. Late submissions for part 1 will lead to an individual demonstration regarding part 2.
4. All code and project documentation for this assessment (= part 1) must be your own work. The demonstration (= part 2) must reflect your own work.
5. All submissions will be checked for any forms of plagiarism. The Plagiarism Framework² of Lancaster University applies.

¹<https://www.lancaster.ac.uk/academic-standards-and-quality/regulations-policies-and-committees/manual-of-academic-regulations-and-procedures/>

²<https://www.lancaster.ac.uk/academic-standards-and-quality/regulations-policies-and-committees/principles-policies-and-guidelines/plagiarism-framework/>

Topic 1: Microchess

Create a GUI-based software program that provides a game for two players who take turns in playing on a microchess board consisting of 4x5 fields.

Further specifics:

- Each of the two players has *only one* of the following figures: King, Jumper, Runner, Tower, Pawn
- Otherwise, all rules from regular chess do apply accordingly (e.g. which moves are allowed for which figure).
- The GUI provides the user a way to indicate the move to be carried out (e.g. by clicking)
- The GUI provides a visual representation of the chess board, which is updated after each move.

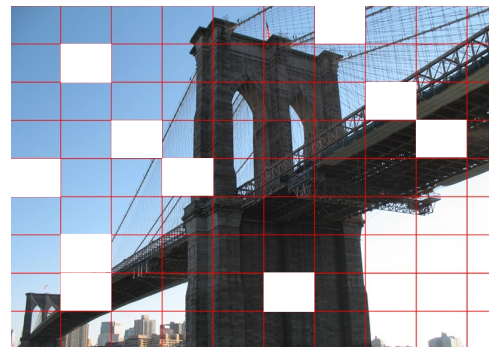


Topic 2: Picture Puzzle Game

Develop a GUI-based software program that provides a game for one player which has to place the right puzzle pieces on a pre-defined medium-sized grid to win the game.

Further specifics:

- The user decides whether at the beginning some or no puzzle pieces are displayed at the correct position.
- In each round, the user is given one random puzzle piece that he has to assign to a position.
- If the piece was assigned correctly, it stays there; otherwise it is removed automatically by the program.



Topic 3: (Simplified) MIPS Emulator

Develop a GUI-based software program that provides a (simplified) emulator for MIPS assembly language instructions – or in other words: a small brother for the well-known MARS emulator.

Further specifics:

- The user can enter up to 30 lines of MIPS assembly language code into a text component.
- The emulator accepts the following instructions and converts them into their full bytecode representation: NOP, ADD, ADDI, ANDI, SRL, SLL, BEQ, BNE
- Bytecode translations are displayed in a second text component.
- The user can ask the MIPS program to be emulated, i.e., to be executed step-by-step; at all times, the user can reset the execution.
- Program counter and registers (=register values) are displayed as part of the GUI and updated accordingly with each step.