



Aalto University
School of Science

Studio 2 Project

CS-C2120, Programming studio 2

CS-C2105, Programming studio A

17.01.2024

Learning goals

- The first larger *personal* software project, which starts from scratch, i.e., requires planning and implementation. You will learn about
 - Refining specification
 - Problem analysis
 - Designing software
 - Implementing and testing software
 - Documenting a project and software
 - And get some experience of project management, including time management.

Schedule

- Topics published on Mon 22.1
 - Choosing a topic, DL Wednesday 31.1
 - Preparing the *project plan*,
 - General part, DL 7.2
 - Technical part, DL 12.2
 - Project is implemented in 2 week sprints
 - You meet your personal assistant about every 2 weeks, between the sprints.
 - In total, 4 times, unless something else agreed
 - Project submission DL Thursday, 25.4
 - Project demos early May
-

General plan, Wed 7.2

- Refining the specification
 - Description of all functionality
 - Description of user interface
 - How the programs gets its input data?
 - What does it produce as output of various functionalities
 - What error situations could happen with user input or data? (e.g., missing or errorneous data, bad formatting, malicious user, ...)

*The main goal of writing the general plan is to make it clear for yourself **What** your program should do, **how** it will be used and **what problems** may appear when the program is used.*

Technical plan, DL Mon 12.2

- Technical plan
 - Files and data format in files
 - Program class structure
 - Example use cases
 - Specific algorithms
 - Data structures / collections used
- Project working plan
 - Schedule, initial plans for the sprints
 - Spring plans will be revised later based on the project progress
 - Testing plan (system testing & unit testing)

*The main goal of writing the technical plan is to make it clear for yourself **How** your program would be implemented and how the project would proceed.*

Support

- Exercise sessions continue during Period 4 on Mondays (14-16) and Thursdays (12-14).
 - Zulip
 - Each topic has nominated assistant(s)
 - Follows and evaluates your project
 - You can discuss the project topic and its interpretation / requirements in spring meetings
 - Programming problems to be resolved in exercises and Zulip (not by email)
-

Sprint meetings

- Roughly every 2 weeks a brief informal meeting with your assistant
 - Reviewing your progress
 - Resolving major problems (not simply programming problems)
 - Setting plans for the next sprint
 - In the first sprint meeting, you get feedback on the project plan
-

Project document

- Personal information
- Overall description
- User's guide
- Program structure
- Specific algorithms
- Data structures
- Files and file formats
- Testing report
- Known bugs and weaknesses
- Best and weakest points
- Reflection on project progress
- Final self-evaluation
- References

Final submission and demo

- Submission includes
 - Project document
 - Source code
- Assistant will review the submission before the demo session.
- Demo is a 15-30 minutes session with the assistant, where you:
 - demonstrate the program,
 - answer assistant's questions about your source code, such as,
 - What is this part of code doing
 - What would you change, if a new feature would be needed or some feature is revised
 - Why have you made this kind technical solution
 - Question may concern the document, as well.
- Final grades are published only after demos.
 - Assistant can request revisions of the project and provide extension for submitting it (=> grade is lower)
 - A working project submitted at least a week before DL, can bring bonus to your final grade.

Project topics

- 20+ different topics available from several themes:
 - Games
 - Graphics
 - Simulations
 - Statistics
 - Simple text data bases
 - Applications
-

Project topics cont.

- Most topics have different levels
 - Easy (grades 1-3)
 - Moderate (grades 2-4)
 - Challenging (grades 3-5)
 - Moderate / demanding require graphical user interface

Own topic

- Suggestions to Lauri Malmi by email by Jan 29th
 - At most 1 page description
 - Should not be old topic used in this course
 - Should have complex enough problem domain for class structure design
 - Should not be too trivial (but not too challenging either)
- Acceptance message and possible refinements to topic will be sent back to you by Jan 30th,
 - Only after this you can select "own topic" from A+ project topic selection assignment.

Let us look at some projects

- Some tuning of topics will be done before publishing them