Student Thesis and Project Guidelines

Visualization and MultiMedia Lab

Department of Informatics, University of Zürich

If you are working on any student project such as a Bachelor or Master thesis, or other project at the Visualization and MultiMedia Lab (VMML), you are asked to follow the guidelines below with respect to carrying out your work.

Work Environment

Your work can in most cases be carried out on your own laptop and in your preferred work space. However, the VMML has a lab, room BIN 1.D.02, with a couple of work places, and student projects can be carried out in this lab if space is available. In particular, if special equipment or computers from the VMML are used, then the work should be carried out in the lab for the majority of the time, if not specified otherwise. We expect regular interactions with the mentoring assistant, at least on a monthly basis.

Coding Rules

Your implementation and source code should follow standard good-practice coding guidelines as well as the recommendations given by the mentor of the project, which is usually one of the research assistants of the VMML group.

Note that your **code and documentation** must regularly be committed to a GIT repository, which can be set up by the project mentor. This is mandatory and has to be done at least on a weekly basis. Failure to do so may affect the grade of projects involving significant programming and software development as this is standard practice in software engineering.

Note that the code as-is in the git main branch at the date of the submission of the thesis will be the basis for grading. However, editing the code afterwards, e.g. for the final presentation is okay and committing the changes is encouraged.

Presentations

During your project you are asked to give two presentations of about 30 minutes each to the members of the VMML research group, and any other involved parties, about the state of your thesis or project.

Half way through the project, a progress midterm presentation should be given to assess the state of the ongoing work, reviewing the achieved progress, intermediate results and the remaining open tasks.

After the completion of your project and the submission of your thesis or report you have to give a final presentation on the achieved solutions, their technical realization, and experimental results.

Deliverables

The deliverables of a thesis or project always include:

- The structured and documented code (e.g. in a GIT repository)
- The thesis manuscript or project report (one **textile-bound** paper print)
- The latex project of the thesis including all figures and the bibtex file
- The midterm and final presentations (PPT or Keynote files)

For most projects resulting in any kind of software or application that has a visual output, i.e. such as real-time 3D rendering, interactive data visualization or rendered images, the deliverables also include corresponding visual examples in the form of:

- A video demonstrating the use cases and capabilities of an interactive system, OR
- An organized and commented collection of visual examples demonstrating the capabilities of an (offline) rendering system OR
- A ready-to-use, deployable build for web based projects (e.g. a zip file of the public folder in most cases).

For projects focused on dataset collection, the deliverables include:

- A well-documented dataset: organized in a clear folder structure, with consistent naming conventions and file formats. Include both raw data and any processed/cleaned versions, if applicable.
- Documentation and usage guide: a `README.md` file describing the content, structure, size of data and how to access, load, and use the dataset. Include licensing and citation information if applicable.
- A short demo video (optional but recommended): explaining how the dataset was collected and what it contains.

Written Report

With your thesis or project you must deliver a written report in a typical research paper like structure: including an introduction and motivation, the problem statement, covering related work, discussing any preliminary (mathematical or technical) background information and algorithms, and in particular a detailed description of your technical algorithmic solution, with only a short description of the implementation, as well as experimental results.

The main focus should be on a scientific treatment and analysis of the problem statement and its algorithmic realization in your project, not a detailed description of your source code. A short

presentation of the software architecture and its main components is adequate in the main part of the report. Furthermore, in the main part of the text, your solution is best presented by text description, illustrations, mathematical formulations as well as pseudocode or algorithms and critical functions, rather than explicit source code. Additional code details can be provided in an appendix.

The use of correct and consistent mathematical formulations and equations is highly encouraged, where applicable, as well as the use of informative figures and diagrams to support algorithms, 3D geometry and rendering problems, data processing flow(s) and systems architecture.

For more details on the expected content and form see also the documents linked on the VMML Student Projects and Thesis web page.[1] In particular, closely follow the instructions in the *Student_Project_Report* template.

Formatting

With respect to structure, typesetting and formatting strictly follow the guidelines indicated in the *Student_Project_Report* template. You must write your report in LaTeX using the template provided on the VMML Student Projects and Thesis web page.[2]

References

Your bibliographic references should follow in style the formatting guidelines given in the report typesetting and formatting guidelines of the *Student_Project_Report* template.

Note that **Wikipedia** and any other general purpose encyclopedia cannot directly be used as a scientific reference. However, Wikipedia and other online or printed encyclopedia are good starting points for initial information gathering and pointers to adequate reference works such as scientific journals, articles and books.

Use of (generative) AI

Persons must disclose any use of generative AI in their project, code and report. This includes information on what they have used generative AI for, i.e. coding and/or writing text, and how they have ensured compliance with good scientific practice. If no system with generative AI was used, this must also be declared.

In particular, also explain how the output of generative AI has been verified to be correct. ChatGPT and other systems are not authoritative systems and do not guarantee any correctness, thus their output cannot be assumed to be correct.

In an appendix of the written report, the necessary details of how and where (i.e. text chapter or section, code modules and functions) (generative) AI has been used, as well as how it has been checked for consistency and correctness should be indicated.
Typical acceptable use of (generative) AI includes:

[1] https://www.ifi.uzh.ch/en/vmml/teaching/student-projects.html
[2] https://www.ifi.uzh.ch/en/vmml/teaching/student-projects.html

- Analyzing errors and warnings with LLMs to find the causes of bugs in your code
- Per line or small block auto-complete features like for example using GitHub copilot
- Using it to find useful functions in libraries (e.g. let LLM find or generate some specific code, but don't simply copy it, instead check the functions before porting them into your own code base)
- Using LLMs for grammar checking and correction of your own written text
- Assist in background research (e.g. get some introductory overview of a topic and help with terminology of the area (needs validation from verified sources))

Do not use (generative) AI in general to:

- Generate (copy) large code segments or text from LLMs (only exceptional cases as indicated above)
- Generating tables, charts, and figures for experimental results directly from your data

Evaluation

Your thesis or project will be evaluated based on the following criteria and the given expectation for achieving a good score:

Category		Expected achievements for a good grade
Quality of deliverables	Meeting objectives	All required objectives were met.
	Quality of implementation and experimental results	A solid and complete implementation was produced which can be used with little to no effort.
Originality of content	Level of own ideas added	New ideas extend already existing work.
	Level of contribution with respect to related work	Contribution is outperforming existing methods in many cases.
Theoretical understanding	Algorithms, data structures; Mathematical concepts and rigor; Design concepts and principles	Most of both basic and advanced concepts were correctly learned, applied and explained.
	Independence in learning new concepts and techniques	The student managed to accomplish the work on their own, with occasional help from the supervisor.
Research methodology	General	All important components of the research were present (RQs correctly formulated, measures and analyses performed).
	Literature research	All aspects of the problem were researched and discussed by referencing multiple papers in a structured fashion.
	Completeness and relevance	The majority of the papers are recent and/or relevant, properly motivated and linked to the project.
	Validity of experimental results	Basic experimental design is present, validated measures were applied and results were interpreted correctly.
Manuscript	Language quality (typos, sentence structure, ...), terminology	Minor typos, otherwise correct language use (spelling, sentence structure, etc.)
	Structure, use of illustrations	All sections are present, no important information is missing. Clear storyline, coherent story and structure. Good use and integration of illustrations.
	Citation style and bibliography	Citations complete, bibliography not missing key information, but some inconsistent entries or formatting.
Presentation	Quality of slides and speech (supports story, clarity, terminology, timing)	All the slides were very clear, used the correct terminology and concepts were introduced appropriately. Speech is clear and well timed.
	Timing, quality and completeness of structure	The presentation contains the standard elements in a scientific presentation (i.e. motivation, problem addressed, related work, solution, evaluation/analysis, conclusions). In all the slides the level, of detail is adequate (i.e., it is easy to identify what was done, how it was done, and why).
	Ability to answer questions	All questions were answered satisfactorily.

For Bachelor and Master thesis, **Quality of deliverables**, **Theoretical understanding** and the **Manuscript** are of key importance. For Master (group) projects, there is a higher emphasis on the **Quality of deliverables**.