

Programming II

"Proof of Performance"

Complete **Task 1** in order to pass the proof of performance ("Leistungsnachweis").

Note the following rules:

- Use Python for the implementation.
- Write a report of about 4 pages.
- Groups of up to 4 students can make a joint submission.
- Submission must include the names and student IDs of all group members.
- **Submit your report (as PDF) and code (as ZIP) via iLearn until July 04, 2023.**

Successful completion of **Task 2** carries a 10% bonus for the exam. The same rules as for Task 1 apply, except that you do not have to write a report.

Task 1 - Tic Tac Toe

Implement the game of Tic-Tac-Toe on the command line for two players:

- You are welcome to use online tutorials, e.g. this one¹, for help.
- Develop the source code based on object-oriented programming.
- Design your software according to the model view controller (MVC) or model view presenter (MVP) architectural design pattern.
- Implement a feature to save the current game state in a file. Also, implement a feature to load a game state when opening the game.
- Add a second mode in which you play against a game AI. You can come up with your own heuristics for your game AI or implement the minimax algorithm².
- Write unit-tests for your business logic including your game AI. Utilize a test coverage tool and achieve at least 90% line coverage of your business logic.
- Use a profiler to identify slow parts of your business logic including your game AI. Try to speed them up.
- Follow the Python PEP8 coding style guide. Use a linter to ensure adherence.
- Use `https://mygit.th-deg.de` to manage the codebase.

¹<http://robertheaton.com/2018/10/09/programming-projects-for-advanced-beginners-3-a>

²<http://robertheaton.com/2018/10/09/programming-projects-for-advanced-beginners-3-b>

The report should consist of the following parts:

1. Introduction
2. Architecture: Describe the design of your program by means of UML diagrams.
3. Serialization and deserialization: Explain how you save and load game states.
4. Game AI: Describe how your game AI works.
5. Tests: Describe your test approach and how you achieved at least 90% line coverage of your business logic.
6. Speed up: Explain how you identified slow parts of your business logic and your game AI and how you optimized them.
7. Conclusions and prospects: Summary and what else you could do in order to further improve your program.

Task 2 - Chess (Optional)

Similarly to the requirements of Task 1, implement the game of Chess on the command line for two players. Implement alpha-beta pruning³ for your game AI.

³See: S. Russel and P. Norvig, "**Artificial Intelligence: A Modern Approach**", Prentice Hall, third edition, 2009.