Practical assignment 1: Game development

Number of points: 15 points

Task:

This task will allow you to apply in practice your knowledge of search-based problem-solving by developing game software. You must select a two-person zero-sum game with perfect information for this task. Students are forbidden to select the games of "Tic-tac-toe" and "Connect 4" (as well as similar games). If you have difficulties finding an appropriate game, you can take any game considered during lectures and make it more complicated or consult with the teacher. You are free to choose a programming language or environment for software development.

The software must provide the following possibilities for the user:

- to choose who starts the game: human or computer;
- to make moves and see changes in the course of the game after the user's moves and the computer's moves;
- to start a new game after the completion of the previous game.

Your software should have a graphical user interface (command-line games will not be accepted). In this case, it is not about a complex, 3D graphical interface but about the use of visual elements such as menus, buttons, text fields, icons, lists, etc.

When developing software, the student must implement the following:

- generation of a state space graph of the game (entirely or partly available);
- a heuristic evaluation function if you deal with a partly available state space graph of the game;
- a game algorithm considered in the course (Minimax, Alpha-beta pruning, or N-ply look ahead).

You need to submit the report describing the developed software. The report must contain the following:

- demonstration examples of software or user manual with the student's given explanations;
- description of the data structures used with detailed comments on what is stored in each data structure;
- a detailed description and justification of the heuristic evaluation function if such a function is implemented;
- code of the main algorithms implemented in the software (generating a state space graph of the game, assigning heuristic values to graph nodes, applying a game algorithm, finding winning paths) together with explanations. The code should be added to the report only as text. It is not allowed to add it as a set of pictures.

The software code must be:

- added to the report appendix as text;
- made available on a public website (GitHub, Google, etc.), and a link to it must be provided on the title page of the report. Using the link provided, the teacher must be able to download the student's assignment without additional registration and restrictions.

The report must be submitted in an electronic format (.docx or .pdf) using the activity "Submission of the report" available in the e-course section "Practical assignments/ Practical assignment 1".

The student must defend the developed software. The student must answer questions about software architecture, algorithms, and data structures during the defence. During the defence, the student must have all the necessary technologies and environments to demonstrate the software successfully. Information on how to apply for the defence will be published in the e-course under "Practical assignments" during the semester.

Therefore, the steps for completing the assignment are the following:

1) select a game;
2) select a programming language/environment;
3) design, implement and test the software;
4) prepare the report and submit it in the e-course;
5) arrange the defence time.

Deadlines:

The deadline for submitting the report on the developed software is published in the "Practical assignments" section of the e-course. After the specified deadline, the reports will no longer be accepted, and it will no longer be possible to defend the software.

Plagiarism checking

Please, consider that your report and the code are subjects for plagiarism checking. Therefore, any attempts to copy (in whole or partly) code created by other students/persons or available on the web or in any other information source will be considered a case of plagiarism leading to the submission of the report on the violation of academic integrity to the administration of the university and annulment of the assignment points. Annulment of the assignment points and preparation of the violation report may also occur later if the violation is revealed after the defence of the assignment.

Evaluation criteria

Evaluating the task, the teacher will consider the following:

- the quality of the report (whether the information required above is included and is meaningful);
- the quality of the software developed;
- the quality of answering questions during the defence;
- absence of a violation of academic integrity.

The assignment will be automatically failed, and the defence will not be organised if:

- the student has not submitted a report but is applying to defend the software;
- the student submitted a command line software;
- in the software, the game tree is not generated but is predefined or stored in a file;
- the student did not implement one of the algorithms considered in the study course (Minimax algorithm, Alpha-Beta pruning, or N-ply look ahead), but the computer performs moves randomly, or moves are predefined in the code, etc.
- a breach of academic integrity has been identified;
- the student has not provided a link to the code on the public website on the title page of the report;
- the student added the code in the form of pictures to the report.