Project Manual Bachelor Year 2
Project 2.1
Play it!

2024-2025

Period 1.1, 1.2, and 1.3
Academic year 2024-2025
Department of Advanced Computing Sciences
Faculty of Science and Engineering
Maastricht University

1 Project description

The central topic of this project consists of strategy games. You are required to implement a human-human player version and a human-computer player version using object-oriented programming. For the artificial player, you will need to take into account that you have to implement a game data structure and a smart evaluation function, which you can apply to your game. Additionally, your artificial player should be adaptive and present some form of learning. The design of your software should follow the Software Engineering principles. The three phases of this project can be described as follows:

1.1 First project phase

The following lists provide the games from which you as a project group should choose one. If you have a favorite game and you want to implement this, you need to ask the examiners for permission. For each game, you should refer to the given link for the standard rules that you have to implement. All deviations need to be made clear in the project plan, presentations and the report. Note that we have categorized them according to three (difficulty) criteria: 1) two-player or multi(more than two)-player; 2) perfect or imperfect information; 3) with or without chance.

The "easiest" games are perfect-information, two-player games without chance:

- Tak (`https://en.wikipedia.org/wiki/Tak_(game)`)
- Connections (`https://boardgames.lovetoknow.com/Connections_Board_Game`)
- Yinsh (`http://www.gipf.com/yinsh/rules/rules.html`)
- Breakthru (`https://boardgamegeek.com/boardgame/335/breakthru`)
- Frysian Draughts (`https://www.frisiandraughts.com/spagina/30/2/rules---regulations.html`)
- Cage (`http://web.archive.org/web/20200701105455/http://www.marksteeregames.com/Cage_rules.html`)

The more difficult games (with one difficulty criterion) are:

- Hearts (`https://en.wikipedia.org/wiki/Hearts_(card_game)`)
- Sneakthrough (`https://ludii.games/details.php?keyword=Sneakthrough`)
- Can't Stop (`https://en.wikipedia.org/wiki/Can%27t_Stop_(board_game)`)

Finally, one "hard" games (with two difficulty criteria)

- Carcassonne (`https://images.zmangames.com/filer_public/d5/20/d5208d61-8583-478b-a06d-b49fc9cd7aaa/zm7810_carcassonne_rules.pdf`)
- Cluedo (`https://en.wikipedia.org/wiki/Cluedo`)

It is clear that the easier games are easier to implement, but of course also less challenging. It is strongly recommended that you clearly discuss this within your group and estimate your skills when choosing one of the games.

It is furthermore possible to choose a game that is not on the provided list. In that case, you have to ask the examiners for permission (Backgammon, Dice Chess, Stratego and Chinese Checkers are not allowed).

Implement - in Java - the game chosen such that two or more human players can play it "on the computer" and make sure you have a good visualization of the game. You are allowed to make use of any library you like for the GUI and visualization of your game. For other libraries, contact your examiners for permission.

Research different algorithms to design a computer player capable of playing against a human. Do a literature study of available AI techniques, decide which ones you will use for your game, and argue why they are suitable for it. You should look into different techniques from the fields of Machine Learning and Adversarial Search and choose at least one technique for each of the two fields. Please note that - unlike in previous projects where the examiners were satisfied with a vague idea of what you will do next - here, we want you to put enough effort into researching the approaches at the start of the project.

By the end of period 1, you submit a project plan. This project plan should contain as minimal requirements (i) Research questions, (ii) the methods and algorithms you plan to apply, and how you will apply them, (iii) the deliverables of your project (what will be the end results and products?), (iv) Time management, containing milestones. You provide a Gantt Chart describing when you will be doing what; how much time is required for each of these things; and which tasks depend on each other.

A risk analysis and contingency plans are recommended (but not mandatory).You think what can possibly go wrong and you come up with alternatives (this is a contingency plan).

You need to make sure that the examiners understand your approaches. UML diagrams may help you to make this clear. However, focus on the most interesting parts of your implementation rather than presenting a huge UML diagram nobody can read. Highlighting the most important parts will make everything readable. This project plan counts for 15% of the Group Grade.

Peer feedback: by the end of period 1, each group evaluates anonymously the games implemented by two other groups and fills out a feedback form. Groups can consider this feedback to improve their programs in the subsequent phase.

1.2 Second project phase

Implement at least one type of game-playing agent based on at least one of the AI techniques you examined and test it against a baseline agent. The simplest baseline agent you can use is one that chooses random actions for each game step, but you can also implement a more advanced baseline. Furthermore, perform a complexity analysis for your algorithm.

1.3 Third project phase

Improve the agent you developed earlier. If your agent was using only one among the Adversarial Search and Machine Learning techniques you selected in the project plan, make sure that in this phase you implement a hybrid agent that combines techniques from both areas. Finally, test and evaluate all the strategies you implemented.

Peer feedback: by the end of the project, each group evaluates anonymously the games and computer players implemented by two other groups and fills out a grade and a feedback form. This grade counts for 10% of the Group Grade.

Project examiners: Nico Roos, Gijs Schoenmakers, Tjitze Rienstra, Aki Härmä