# Portfolio Exam 3

## Reinforcement Learning – BlackJack Player

This portfolio exam is all about a self-learning BackJack player based on Reinforcement Learning methods. The basis of Reinforcement Learning are the methods from [2]. Basic concepts of the BackJack card game, various rule variations, and card counting methods can be found in [3].

### Task P3.1

Realize a Reinforcement Learning implementation of a self-learning BlackJack player in a programming language of your choice. This implementation provides the basis for the paper from Task P3.2. It shall learn optimal policies for at least the following scenarios:

1. The *"Basic Strategy"* from [3].
2. The *"Complete Point-Count System"* from [3].
3. In addition to the basic rules, two rule variations of your choice shall be examined for their influence on the strategies from (1.) and (2.).
4. Consider improving the system from (2.) to be able to achieve higher profits on average. Note: Your system does not have to be suitable for humans. It may therefore be relatively complicated, e.g. with respect to card counting.

What profit can be expected for the different scenarios?

You can re-use your previous implementations from the Reinforcement Learning exercises. For example, the environment implementation from the Chapter 3 exercises.

The deliverable for this task is the commented source code of your implementation and all logfiles that contributed to the results in Task P3.2.

### Task P3.2

Prepare a research paper using the official IEEE template from [1] (format A4). The absolute maximum length of the paper is 6 pages. Make sure your paper has an appropriate structure and outline. Explanations for an appropriate structure of a scientific paper can be found in the Portfolio 2 document.

The practical part from Task P3.1 focuses on different aspects from the application point of view. The paper should now integrate these findings into a scientific paper. Among other things, the following scientific questions arise:

- How do the different learning algorithms behave for the task at hand? Which learning methods have specific advantages and disadvantages here?
- Can you roughly estimate the size of the state-action space for your implementation? Can one expect to achieve stable estimates $Q(\cdot,\cdot)$? If not, how do you deal with this?
- Can you explain why the rule changes you decided to make led to the policy changes you observed?

The deliverable for this task is a pdf version of your paper.

## References

[1] IEEE Paper Template. `https://www.ieee.org/conferences/publishing/templates.html`.

[2] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 2 edition, 2018.

[3] Edward O. Thorp. *Beat the Dealer*. Vintage, New York, 1966.