

# Computational Intelligence in Games (CIG)

Sanaz Mostaghim

Chair of Computational Intelligence

Otto von Guericke University Magdeburg

# Lecturers

## Course:



Prof. Dr.-Ing. habil. Sanaz Mostaghim  
Chair of Computational Intelligence  
Room: G29 – 024  
Office hours: on request per email  
Email: [sanaz.mostaghim@ovgu.de](mailto:sanaz.mostaghim@ovgu.de)

### Research topics:

- Computational Intelligence (CI)
- Evolutionary Algorithms, Multi-Objective Optimization, Decision-Making
- Swarm Intelligence, Collective Decision-Making, Artificial life, CI in computer games
- Swarm Robotics, Evolutionary Robotics

## Tutorials:



Dr. Christoph Steup  
Chair of Computational Intelligence  
Room: G29 – 014  
Office hours: on request per email  
Email: [steup@ovgu.de](mailto:steup@ovgu.de)

Lars Wagner

[Lars.wagner@st.ovgu.de](mailto:Lars.wagner@st.ovgu.de)

# Courses at the chair of Computational Intelligence

## In WS:

- Intelligente Systeme, Bachelor (5 CP)
- Swarm Intelligence, Master (6 CP)
- Bayes Networks, Master (6 CP) --> Professor Kruse

## In SS:

- Computational Intelligence in Games, Bachelor (5 CP) und Master (+ Extra Work 6 CP)
- Evolutionary Multi-Objective Optimization, Master (6 CP)
- Fuzzy Systems, Master (6 CP) --> Professor Kruse
- Introduction to Software Engineering for Engineers (6 CP) --> Dr. Steup

## WS und SS:

- Seminar Computational Intelligence, Swarm Intelligence, Swarm Robotics
- Software Projects in SwarmLab – Flying Swarm, Rolling Swarm, Driving Swarm
- Team Projects in SwarmLab – Flying Swarm, Rolling Swarm, Driving Swarm

# Time and locations of this course

Lectures will be online (Videos will be uploaded every week).  
You access the vides on the Mediasite of the university.

All relevant information about this course are available on the Webpage:  
<http://www.ci.ovgu.de/Teaching/>



The screenshot shows the header of the website. On the left, there is a logo of Otto von Guericke University Magdeburg featuring a portrait of Otto von Guericke, followed by the text "OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG" and "CHAIR OF COMPUTATIONAL INTELLIGENCE". On the right, there are links for "Sitemap", "Imprint", and "Contact", a search bar with the placeholder "Search phrase" and a magnifying glass icon, and a small flag icon. Below the header is a navigation bar with links: "NEWS | TEAM | TEACHING | RESEARCH | PUBLICATIONS | SWARMLAB | ROBOCUP | STUDENTS | PROJECTS" and "DIRECTLINKS". Below the navigation bar is a breadcrumb trail: "Home > Teaching > SS2021 > Computational Intelligence in Games (Re...".

## Computational Intelligence in Games (Reinforcement Learning and beyond)

### Description of the course

This course addresses the basic and advanced topics in the area of computational intelligence and games. This course has three parts:

**Part one** addresses the basics in Evolutionary Game Theory (EGT). In this part you will learn about simple games such as scissors/rock/paper and the main focus on the strategies for playing games.

**Part two** is about learning agents and we focus on reinforcement learning mechanisms. There are three questions for games:

How can we use the information from a search mechanism to learn?

How can we use reinforcement learning to find a better strategy?

How can we use reinforcement learning as a search mechanism?

**Part three** contains the advanced topics in games and artificial intelligence such as how can we program an agent who can pass a Turing test? how can we consider physical constraints of a spaceship while moving in an unknown terrain? etc.

# The course is offered to the students:

Bachelor students - 5 cp

and

- MSc CV, WPF Informatik
- MSc INF, WPF Informatik
- MSc IngINF, WPF Informatik
- MSc WIF, WPF Informatik
- MSc DKE, WPF Applications

→ Master students have to do extra work to get 6 cp

# Time and locations of the tutorials

We will use two platforms (Information will be provided on the webpage):

**Zoom** for attending the tutorials

**Moodle** to upload the assignments and access a forum

Using the Zoom portal, you will attend one of the following tutorials at the given time.

You have to register for tutorials (next slide):

Group 1: Monday 11:00 - 13:00 even weeks Group 1 starting from 19.04.2021

Group 2: Friday 11:00 - 13:00 even weeks Group 2 starting from 23.04.2021

Group 3: Monday 11:00 - 13:00 odd weeks Group 3 starting from 26.04.2021

Group 4: Friday 11:00 - 13:00 odd weeks Group 4 starting from 30.04.2021

We have both programming and written exercises.

# Tutorials - Registration

To attend a tutorial group, you will need to register for the tutorials in LSF  
(Deadline 18<sup>th</sup> April).

After you log in with your student account, you can apply for the tutorials and give preferences for each of the four groups. **Please select all the groups that fit with your schedule, and then give preferences for each of them.**

We will assign the free spots in the groups based on the preferences you gave for each group. Important: There is a **limited space** for each group.

Note that the **final assignment** of the free spots will be done **after the application deadline is over**, and you will be **informed via email** which group you are in and if you can participate in this course or not.

# Tutorials - Participation

There will be **written** and **programming** assignments during the semester, published on the homepage of the course.

Participation in the tutorials consists of:

- Preparing answers to the assignments at home,
- **Submitting them on the Moodle webpage** of the course. Format of the submission is explained on the Moodle submission platform.
- Attending the tutorials and presenting your solutions to your fellow students

At each tutorial, we will ask you to **present your solution for those assignments** that you prepared and submitted. Please note that the file that you submit will be used for your presentation.



# Tutorials - Remarks

Each week, one assignment sheet will be discussed. You only submit and present assignments in the respective week where these assignments are scheduled to be discussed.

You should only submit and present **in your own tutorial group**.

Visiting the other tutorial groups is possible, but it should **remain an exception** (e.g. because of illness or other important appointments) and should be announced beforehand.

In case of illness, you should **notify your TA** about your absence beforehand. You can then attend another tutorial group in that week.

If you have a certificate of illness from your physician, the respective assignment sheet will not be counted when calculating your percentage of solved assignments in the end.

# Tutorials - Remarks

## Remarks for solving assignments and volunteering for presentation

- If you have difficulties, you can ask your fellow students or your TA for help.
- Please only volunteer for assignments if you are prepared to present your solution.
- Your solutions during the presentations do not have to be 100% correct. It should be visible that you **made a serious attempt to solve the assignment**.
- If you submit an assignment, but can not present a solution when asked to do so, we reserve the right to nullify your points for this assignment sheet. If it happens more often, we might exclude you from the exam.

# Exam requirements

You are allowed to write the exam, only if you volunteer for at least 2/3 of all assignments and work on the programming competition assignment.

Each student is required to finish a semester programming assignment to participate in the exam (not the ones on the exercise sheets). Master students will need to finish an extra assignment to qualify for the 6th CP.

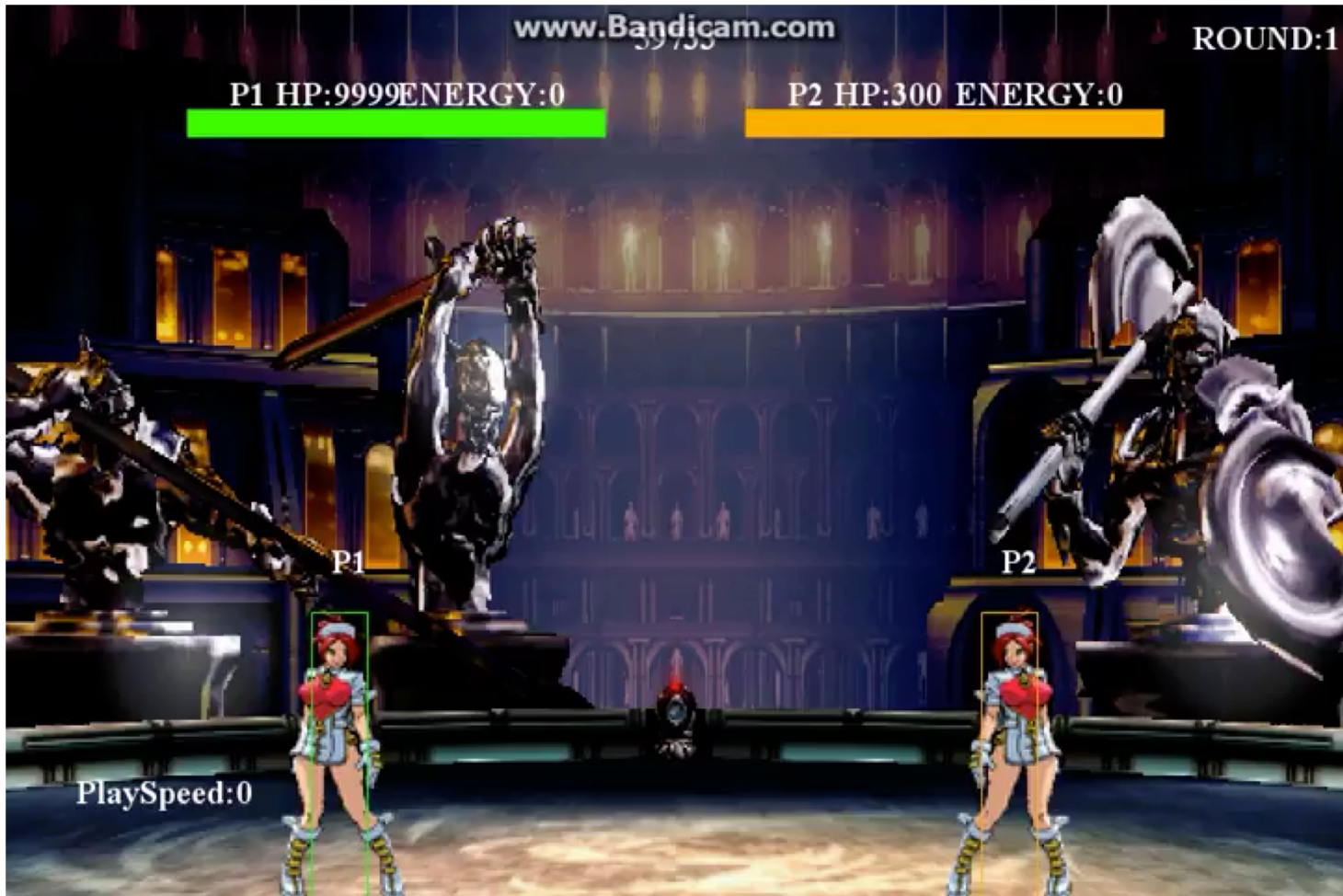
In the programming assignment you will apply algorithms from the lecture to an actual research competition game.

- You can work in groups of 2.
- Also, we require you to write a short report.
- At the end of the lecture we will have an internal competition to determine the best solution. The best participants will have time to shortly describe their algorithm.

Submission deadlines will be announced in the exercise classes.

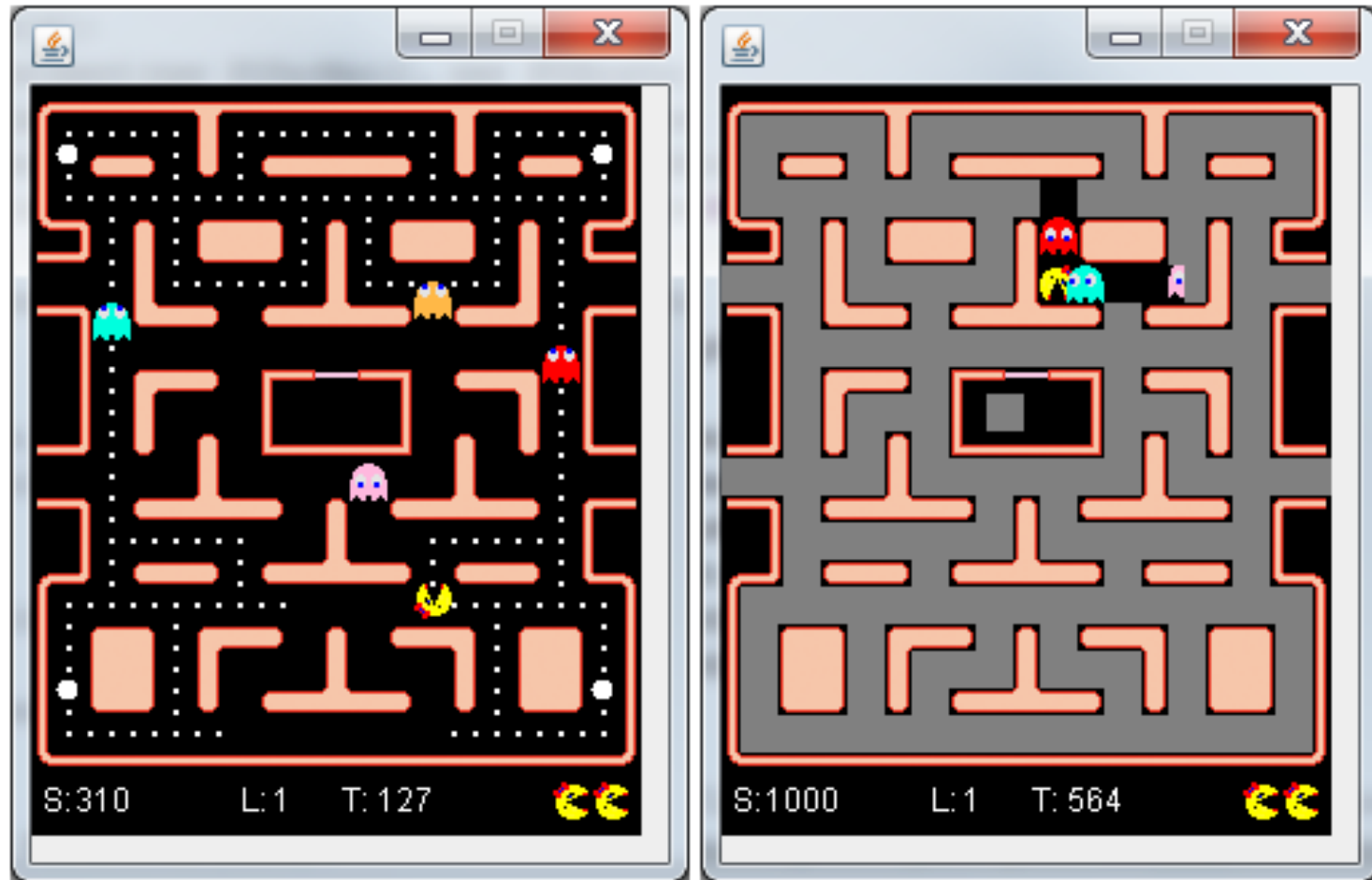
# Previous competitions - CIG 2014 & 2015

## Fighting Game Competition



# Previous competitions - CIG 2016 and 2017

## Mrs. Pac-Man Vs. Ghost Team Competition



# Previous competitions - CIG 2018, 2019 and 2020

Hearthstone: Heroes of Warcraft



# This year's competition (new)

## Strategy Game: Micro RTS

The screenshot shows a game grid with various units and buildings. Annotations include:

- Player 0 units have a blue outline**: Points to a group of units (grey circles and a white square) in the top-left.
- Selected unit is highlighted in green**: Points to a grey square building in the top-middle.
- Player 1 units have a red outline**: Points to a white square building and a grey circle in the bottom-right.

Unit and building descriptions on the right:

- Barracks**: train attack units
- Heavy**: high power but slow melee unit
- Light**: low power but fast melee unit. Heavy and Light units are very effective when micromanaged properly. Stay away from these units if you are not familiar with microRTS though.
- Ranged**: long range attack unit. Try to use this unit, as it is the most effective for human use.
- Bases**: accumulate resources and train workers
- Workers**: can harvest minerals and construct buildings
- Minerals**: harvest them with workers

At the bottom, there are three buttons: **Light**, **Heavy**, and **Ranged**. An arrow points to the **Ranged** button with the text: "Select the unit to train/construct with mouse, or using the highlighted quick key."

# Exam

1. During the lecture you will learn the theoretic background of many computational intelligence algorithms.
2. The tutorials will help you to understand how to apply them to a very constrained practical problem.
3. The programming assignment will ask you to apply and adapt the algorithms yourself.

The written exam will test your knowledge. Exam questions will test your understanding of all three previous phases.

- Exam questions will be in English.
- You may answer in English or German.