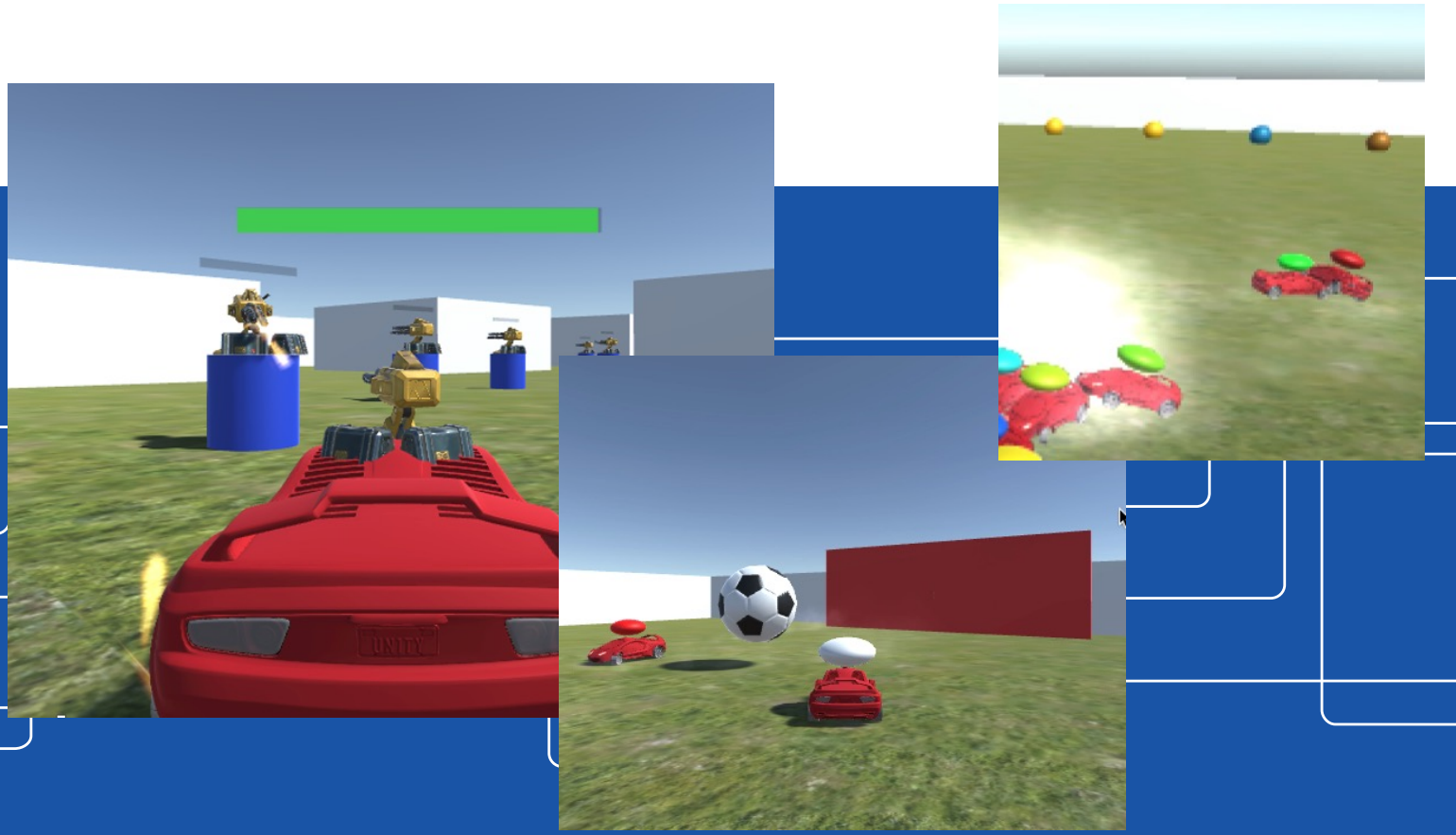
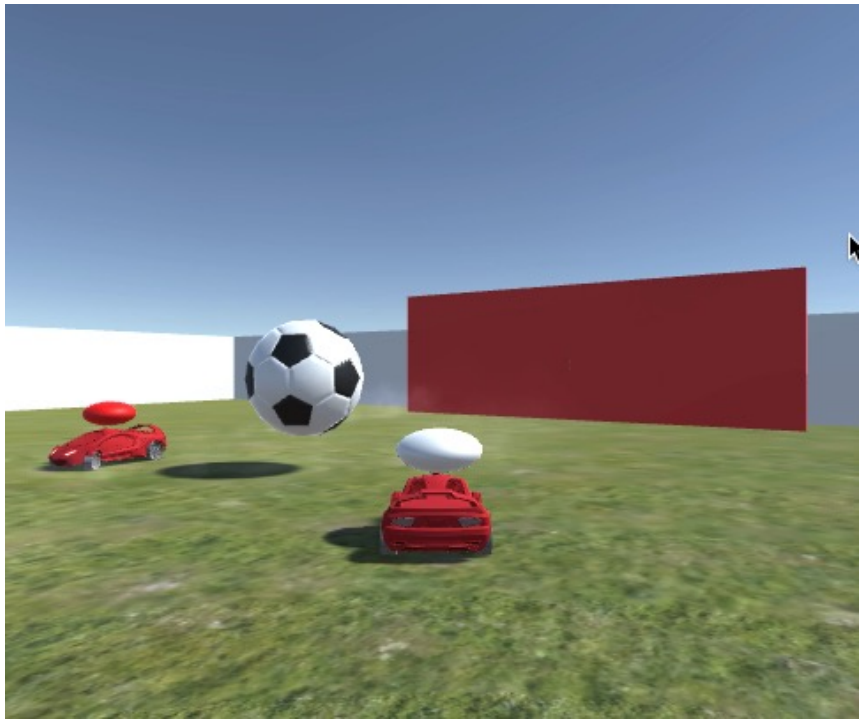
Artificial Intelligence and Multi Agent Systems DD2438

Lecture 1 – Welcome!



Outline of lecture

- Administrative stuff
  - Course content
  - Examination
  - Assignments
  - Schedule
  - Teachers
- Demo of assignment
- Motion models
- Motion planning (next lecture)



Course content

In this course, we will study problems such as:

- AI for **computer games** where multiple opponents or team mates cooperate, such as FIFA 22 or Call of Duty.
- AI for multi robot **search and rescue**
- AI for **autonomous vehicles**
- AI for **multi robot soccer**
- Cooperative path planning and task assignment

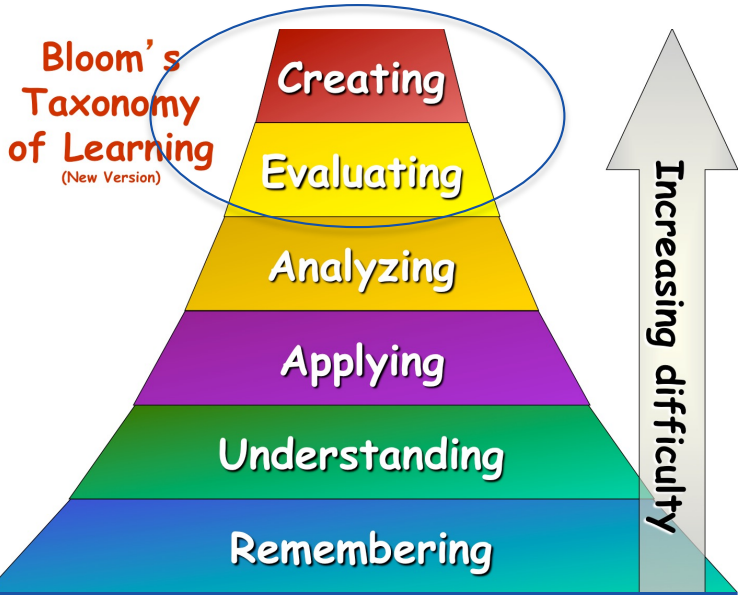
All info: https://canvas.kth.se/courses/37561 (Canvas)

Intended Learning Outcomes

After completing the course the students should be able to:

- develop intelligent multi-agent systems — Assignments
- use a number of **important tools and technologies**
- **assess the value of** existing solutions — Independent work
- manage and participate in a **larger project** — Project planning and reporting
- **present their results** orally and in writing
- **write a basic scientific paper**

At end of Assignments

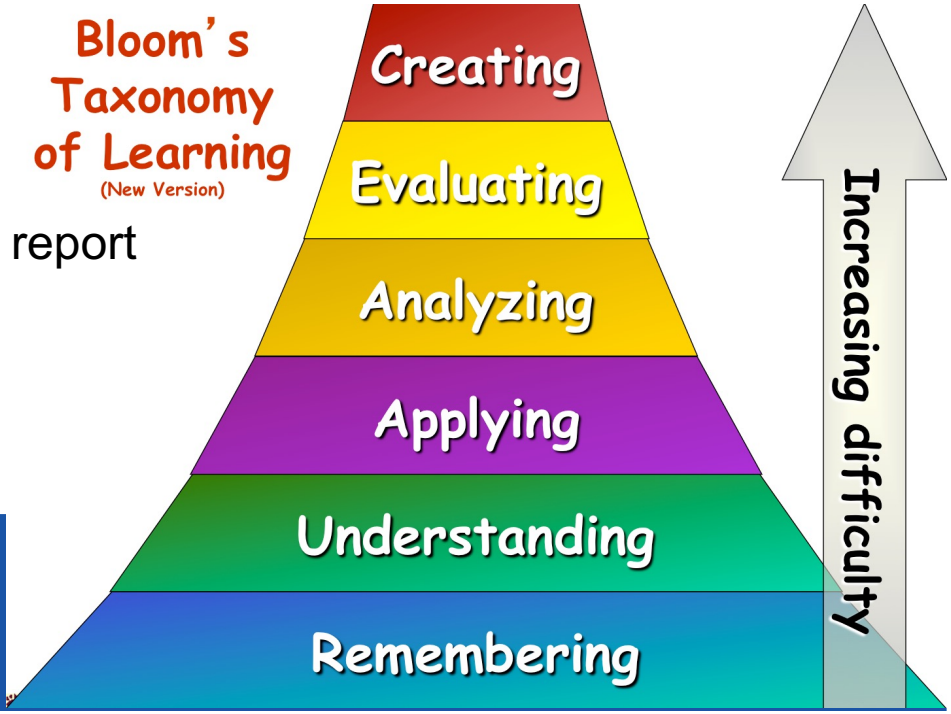


Examination

- Course grade: P/F
- Duration: periods 3 & 4
- Examination: 4 assignments (small projects, teams of 2)
- Assignments to be carried out in a simulation environment
- In this course we learn from each other!
  - Be active, ask questions, discuss
  - Start early
  - Present well

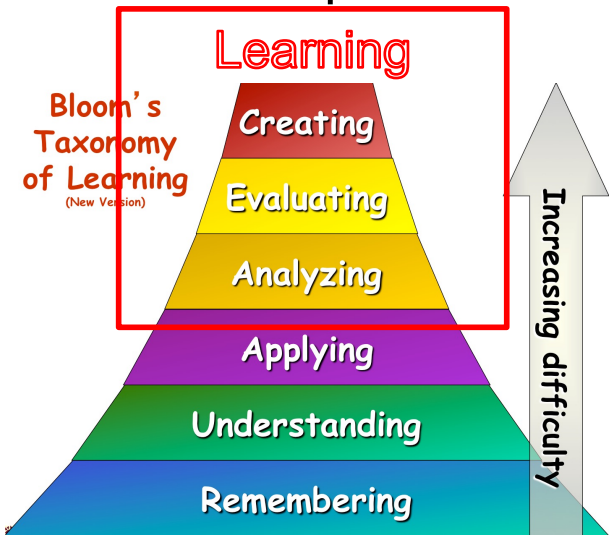
- If you
  - Have no presentation
  - Have not updated the progress report
  - Severely ignore instructions
- You get a warning
- 3 warnings => Fail the course

If this is your teammate, tell me and I will immediately split the group (and you will both work alone)

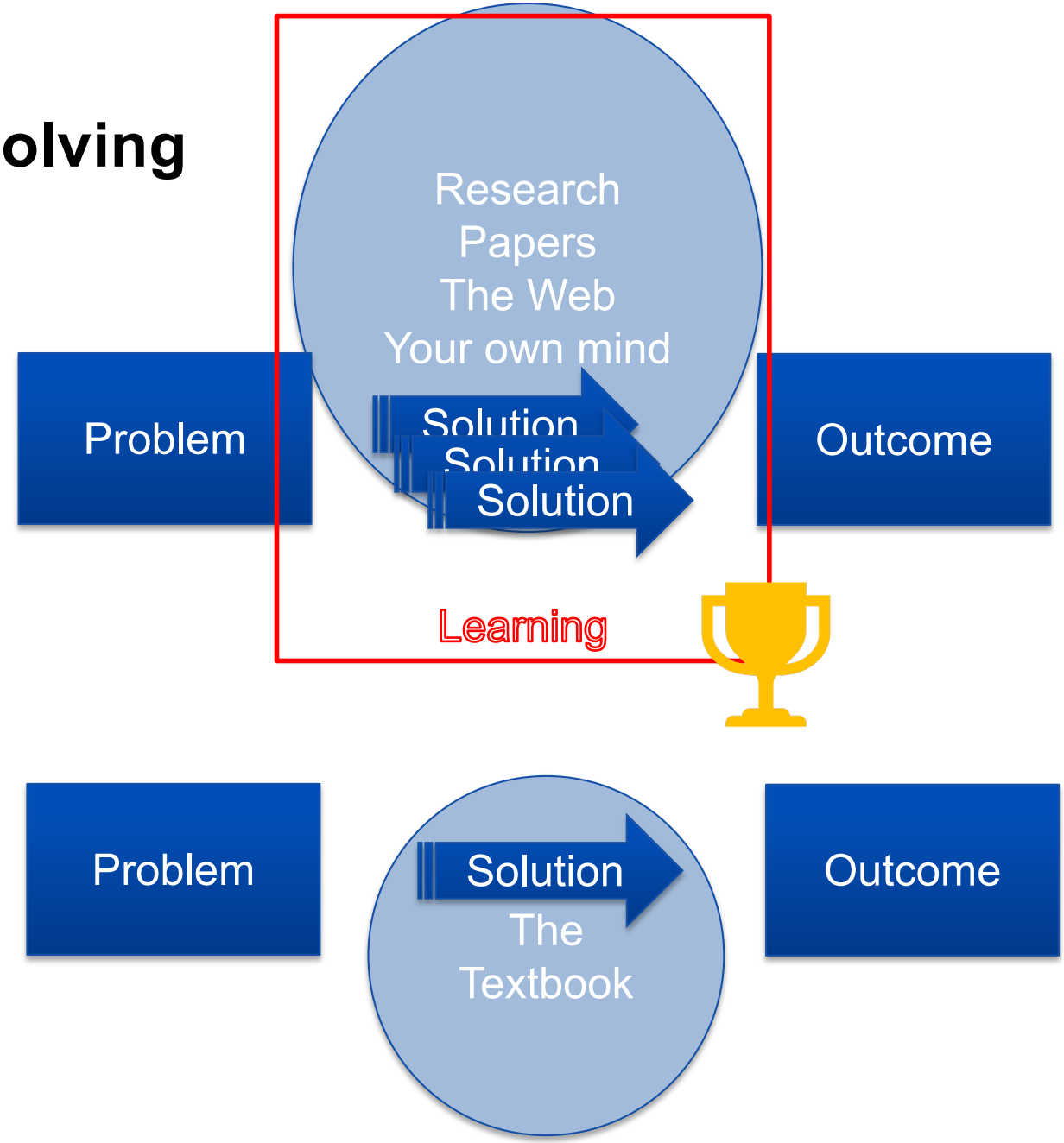


Problem solving

- Real problem



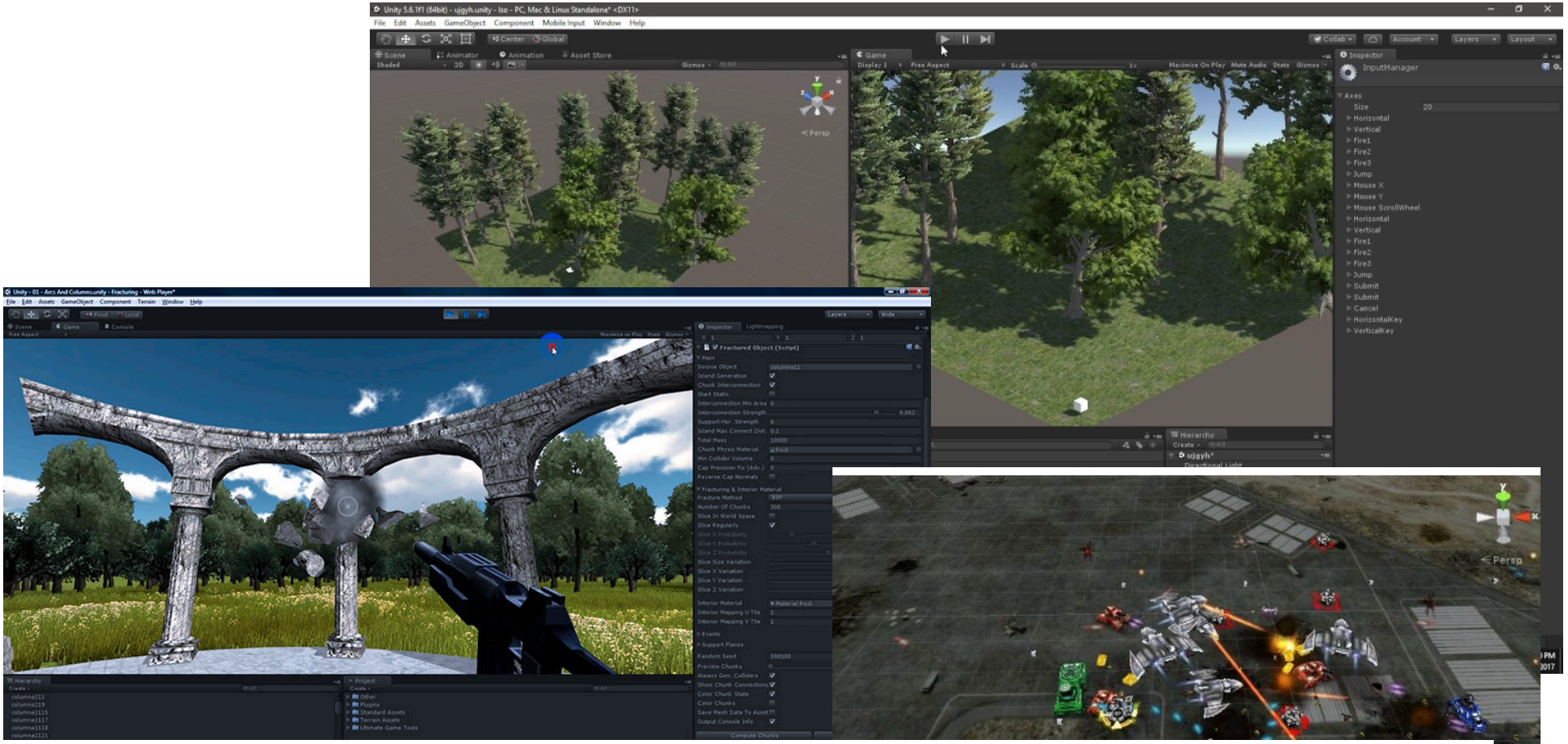
- Textbook problem



What advice would you like to give to future course participants?

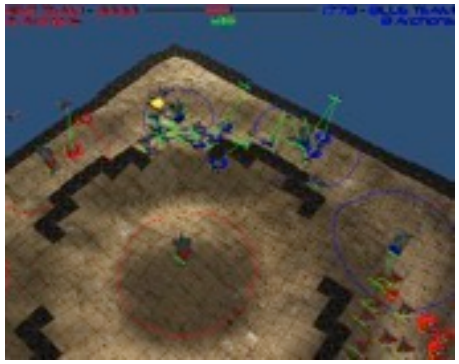
- Don't be afraid to spend some time on interesting project, and organize yourself properly to NOT do everything at the last moment. **Start in time** and scope well.
- **Make sure you plan**, and research each task to get an understanding of the workload. **Don't be afraid to be ambitious**, because failed results are sometimes better (and acceptable) than boring and expected results.
- **Start early and really make sure you understand** and correctly implement the models. (make sure your newtonian mechanics checks out).
- **Plan your time wisely**.
- To **start working** with the assignments **early**.
- **Start projects earlier** to prevent working through nights to finish on time
- **Read more papers - and read them closely**.

Unity3d will be used for simulations



What advice would you like to give to future participants? (I worked: 30-32 timmar/vecka)
Start on time, read a lot of papers and try to have fun implementing your solutions.
if you haven't used Unreal Engine before, maybe you should go for Unity.
Start early on all the projects, they take much more time than you think. Also, try to comunicate well with your teammates and divide the work properly.
Start working early with P1, it is time consuming. Setup the code so it can be used in the other assignments. DO NOT use Unreal unless you're comfortable with it.
Explore own ideas and do not settle for a simple well known solution

The four assignments



1. Single Vehicle Motion models and Motion Planning
2. Multi Agent Coordination and Task allocation
3. Dynamic Multi Agent Interaction
4. Agree on 5 different projects, with >2 teams on each:
    1. Real time strategy game (Star Craft, BattleCode.org)
    2. Soccer/Sports (Fifa, Robocup Soccer)
    3. Search & Rescue robotics (Robocup Rescue)
    4. Action game (FPS, Call of Duty),
    5. Autonomous Driving

Course timeline

In total, the course covers 20 weeks with an average workload of 20h/week.

Module 1 (4 weeks, 100h of work)

w3 Tue 2023-01-17 13:00 - 15:00 in Q34
Fri 2023-01-20 13:00 - 15:00 in U41
w4 Tue 2023-01-24 13:00 - 15:00 in Q36
w5 Tue 2023-01-31 13:00 - 15:00 in Q36
w6 Tue 2023-02-07 13:00 - 15:00 in U21

Module 2 (4 weeks, 100h of work)

w7 Tue 2023-02-14 13:00 - 15:00 in W25
w8 Tue 2023-02-21 13:00 - 15:00 in U31
w9 Tue 2023-02-28 13:00 - 15:00 in U41
w10 Exam week
w11 Exam week
w12 Mon 2023-03-20 13:00 - 15:00 in U21

Module 3 (4 weeks, 100h of work)

w13 Mon 2023-03-27 13:00 - 15:00 in U21
w14 Mon 2023-04-03 13:00 - 15:00 in U21
w15 EASTER
w16 Mon 2023-04-17 13:00 - 15:00 in U21
w17 Mon 2023-04-24 13:00 - 15:00 in U21

Module 4 (4 weeks, 100h of work)

w18 Tue 2023-05-02 13:00 - 15:00 in U21
Wed 2023-05-03 13:00 - 15:00 in U21
w19 Mon 2023-05-08 13:00 - 15:00 in U21
w20 Mon 2023-05-15 13:00 - 15:00 in U21
w21 Mon 2023-05-22 13:00 - 15:00 in L51
w22 Tue 2023-05-30 (Backup and Report deadline) 14:00 - 16:00 in L51
Thu 2023-06-01 (Backup and Report deadline) 14:00 - 16:00 in L51

In the lectures we will

- Discuss ideas
- Compare solution candidates
- Have short presentations of algorithms/ideas

At least one team member should attend each lecture and be ready to

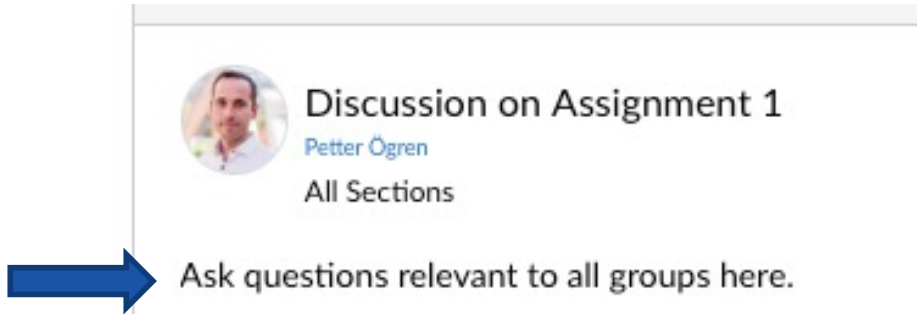
- report progress
- demonstrate solutions
- try to convince others of benefit of chosen solution!

Note: This is a project course → Few lectures, Lots of Independent work
But you are always welcome to discuss with me (email to set time)

Teachers

- Petter Ögren
- Professor in Computer Science at RPL
- PhD in 2003
- 9 years as Senior Scientist at Swedish Defence Research Agency
- 11 years at KTH
- Contact:
  - petter@kth.se (answer important to only you)
  - Canvas (answer important for all)

- Mart Kartašev
- PhD student at RPL
- Extensive experience in software development



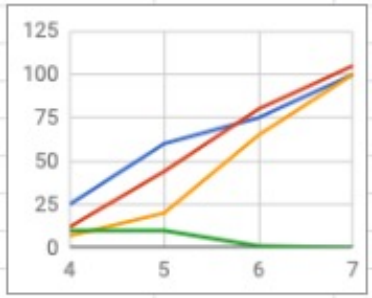
Assignment Groups

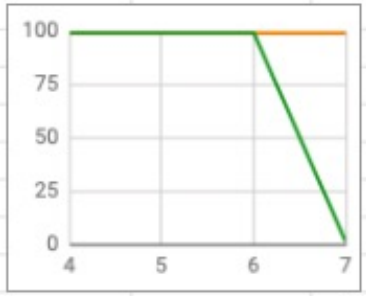
- In the assignments you will be working in **groups of 2**
- At least **one from each group must attend** each meeting/lecture
- Groups are randomized by Canvas
  - Check for **schedule clashes** (only reason for group changes)
- Each group has a **Group Manager (see canvas)** he/she is **responsible** for
  - Choosing project model (Scrum, Agile, Classical ...) and describe it on report pages
  - Updating **Progress-report-page** on
    - Planed effort
    - Progress in terms of time spent, risk estimates etc.
  - someone **presenting** at meeting
- Aim for consensus decisions, but if you cannot agree the leader decides

Progress Report (link in Canvas)

Fill in data the day before each meeting and include in submission (last page)

	Note: Effort is in % out of the 200h (for 2 persons working 2.5 weeks full time)				
	Let the number of each week be given by the effort spent between the meeting in that week and the ... meeting.				
		Week number		Final Meeting	Report Handin
Group 0		**4**	**5**	**6**	**7**
(example)	**Planned Effort (accumulated):**	25	60	75	100
	Actual Effort (accumulated):	12	44	80	105
	Actual Progress:	7	20	65	100
	Risk of Failure:	10	10	1	0
	Comments on Risk:	(only needed if >5%)			
	Comments on Progress:	(only needed if large gaps between first 3 curves)			
	Project management:	(what approach are you following and how is it working)			
		Week number		Final Meeting	Report Handin
Group 1		**4**	**5**	**6**	**7**
(Name 1)	**Planned Effort (accumulated):**	99	99	99	99
(Name 2)	**Actual Effort (accumulated):**	99	99	99	99
	Actual Progress:	99	99	99	99
	Risk of Failure:	99	99	99	1
	Comments on Risk:				
	Comments on Progress:				
	Project management:				



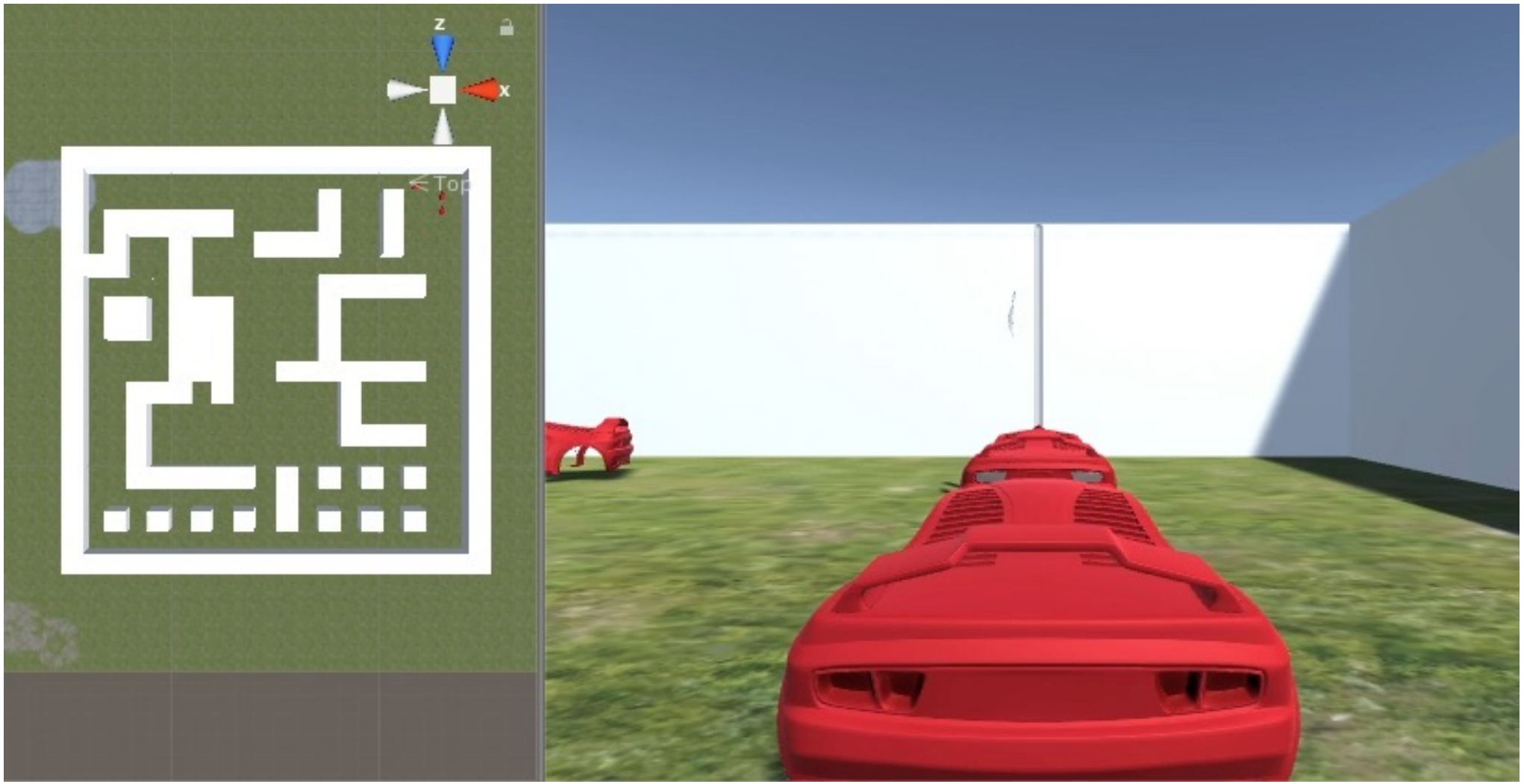


If you were tech consultants, you would charge X SEK for your Assignment 1 report and code

X = 250 000

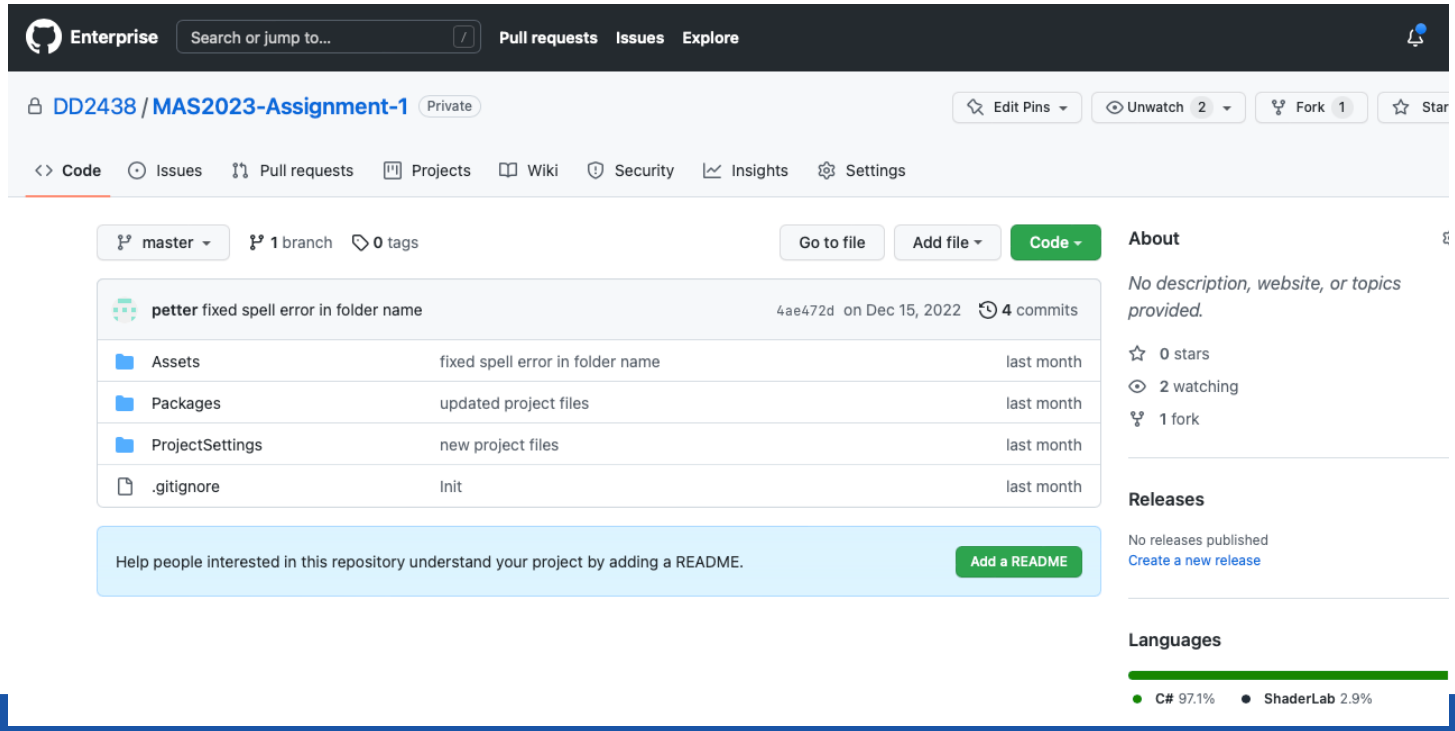
Assignment 1

- Create AI algorithms for running a "racing game" **car** and **drone** through a maze in the shortest possible time
- Demonstrate your algorithms on final meeting (new problems similar to A, B, C)
- Write a report describing your design (1 week after final meeting)

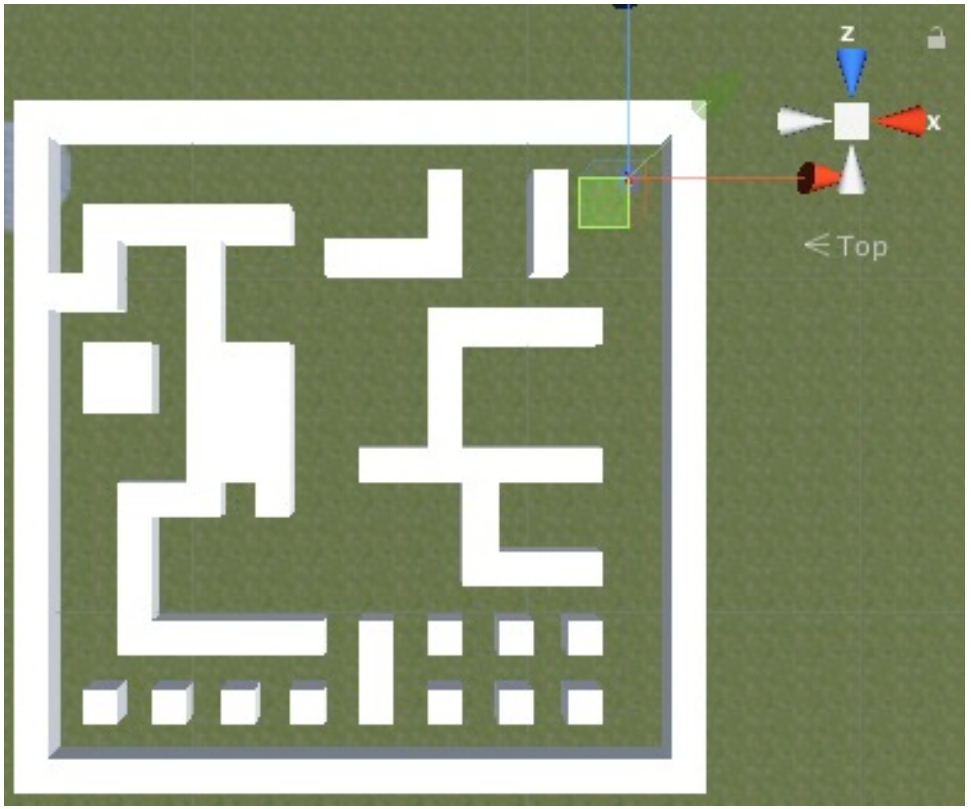


Files can be found on:

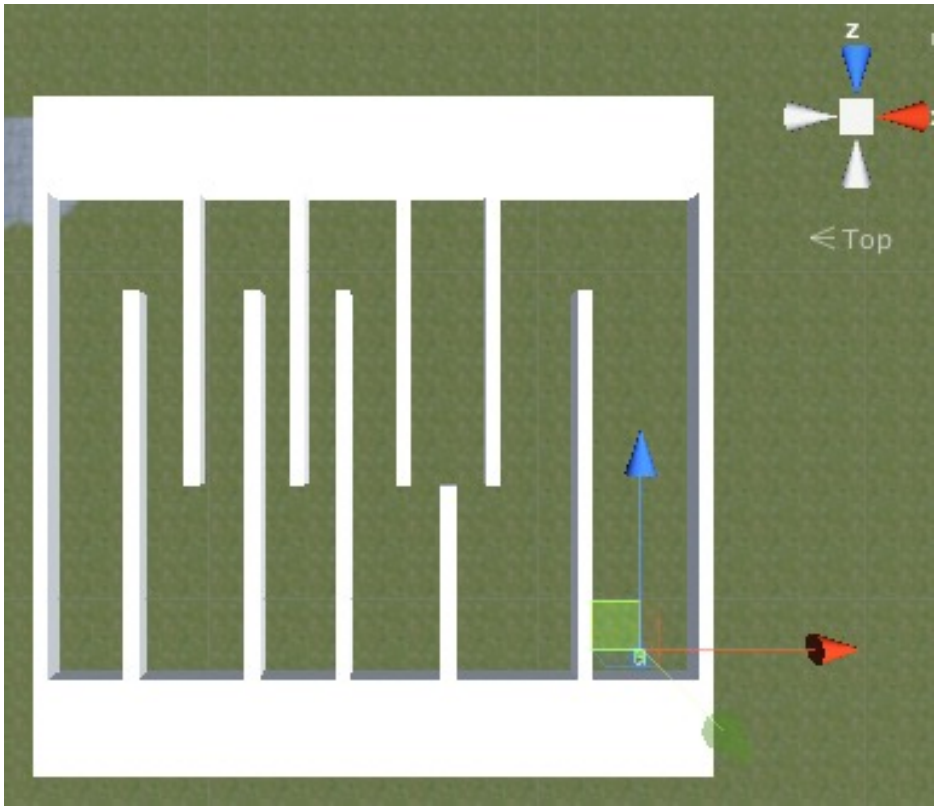
- https://gits-15.sys.kth.se/DD2438/MAS2023-Assignment-1
- Create a Fork of the repository



terrainA.json



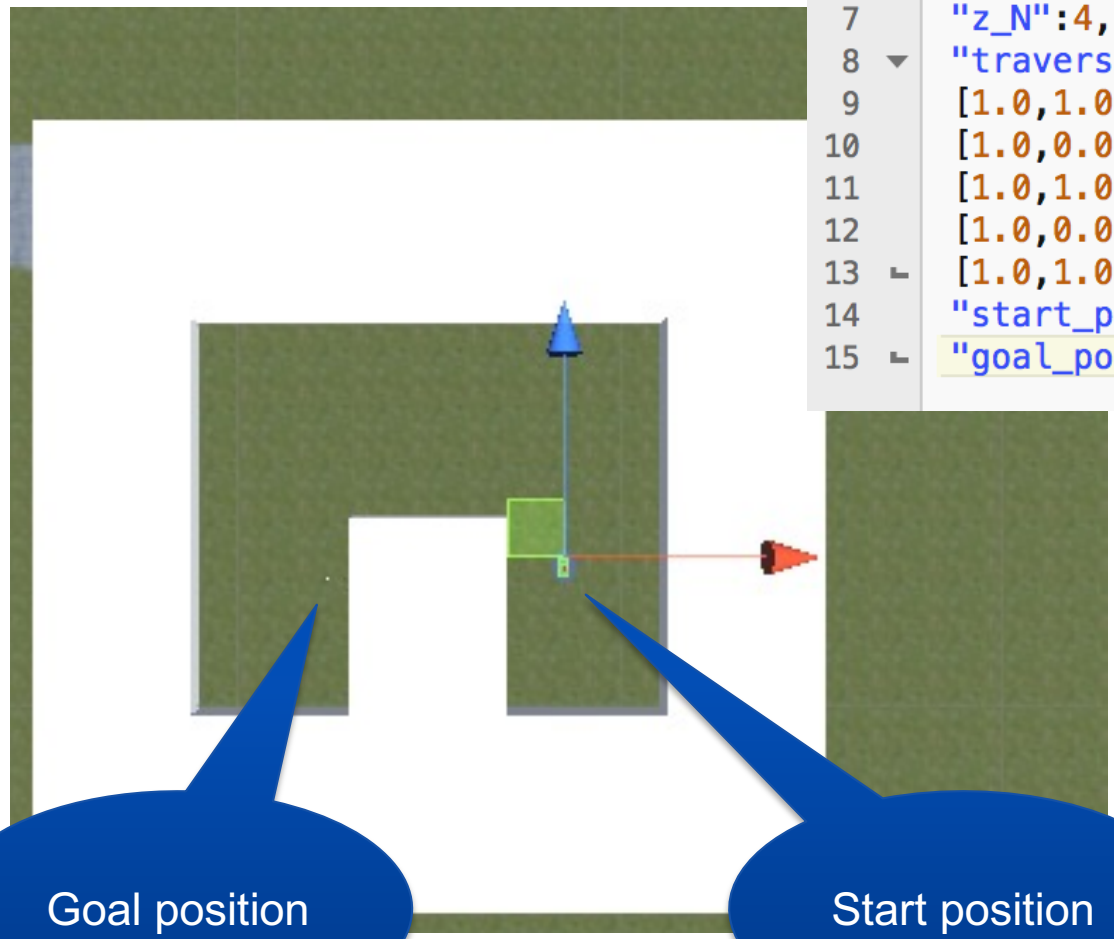
terrainB.json



~/Projects/unityDev/Assignment_1/Assets/Resources/Text/terrainC.json

```
{"file_name":"test88",
"x_low":50.0,
"x_high":250.0,
"x_N":5,
"z_low":50.0,
"z_high":250.0,
"z_N":4,
"traversability":[
[1.0,1.0,1.0,1.0],
[1.0,0.0,0.0,1.0],
[1.0,1.0,0.0,1.0],
[1.0,0.0,0.0,1.0],
[1.0,1.0,1.0,1.0]],
"start_pos":{"x":185.0,"y":0.0,"z":137.0},
"goal_pos":{"x":124.0,"y":0.0,"z":134.0}}
```

Grid matrix



Goal position

Start position

Demo of Unity3d for Assignment 1

Technical part of lecture

- **Why** study Multiagent system?
- **How** do we model them? (motion models)
- **How** do they navigate (motion planning)

Why multiagent systems?

- If a **group of smaller** robots replace **one big,** then …

- We might improve
  - Robustness
  - Flexibility
  - Performance

How to model multiagent systems?

Complexity

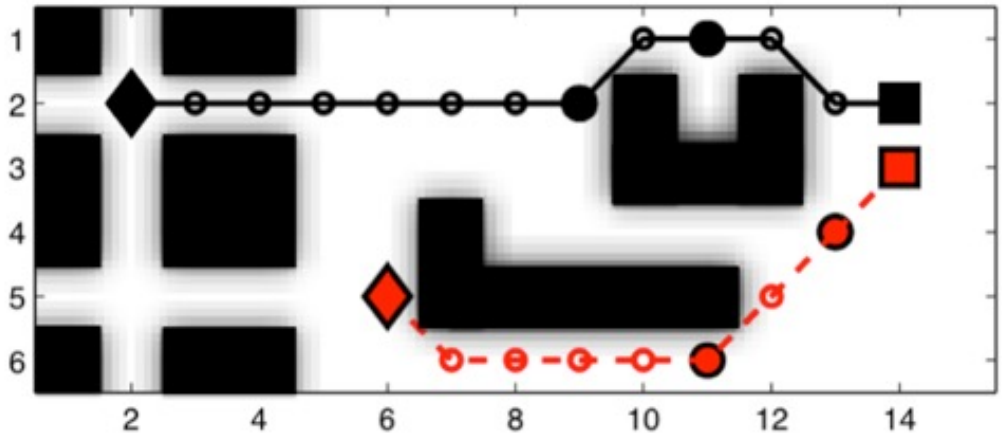
- Discrete state space
- Kinematic point model
- Dynamic point model
- Differential drive
- Kinematic Car model
- More detailed models

Discrete state space

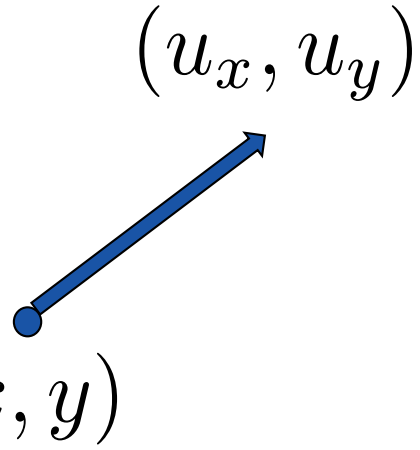
- In detail:

$$x_{k+1} = u_x,$$
$$y_{k+1} = u_y.$$

- Example: Cooperative search



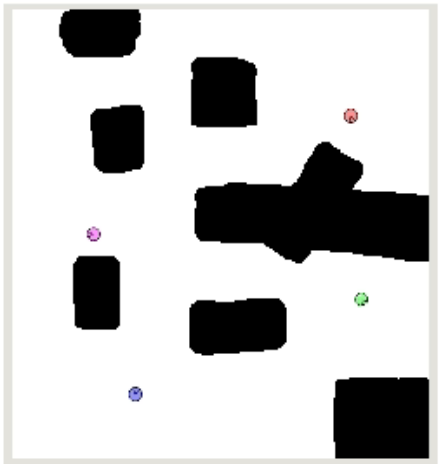
Kinematic point model



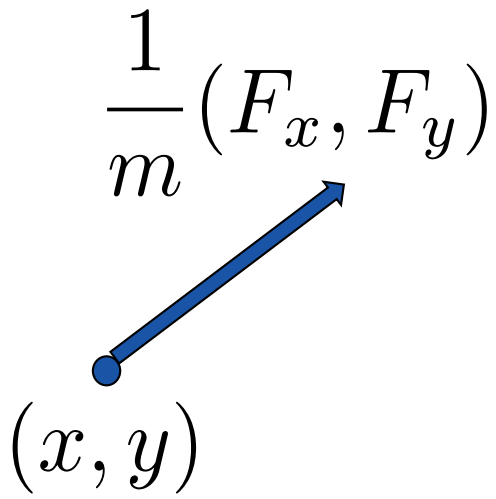
- In detail:

$$\begin{aligned} \dot{x} &= u_x, \\ \dot{y} &= u_y. \end{aligned}$$

- Example: Pursuit evasion

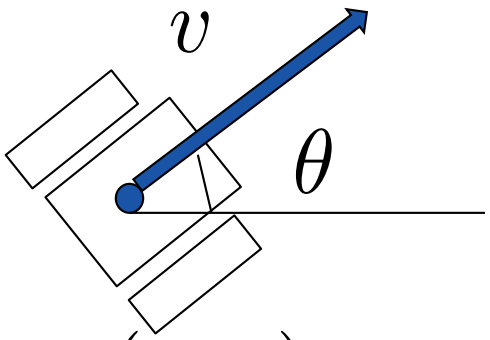


Dynamic point model



$$\begin{aligned} \ddot{x} &= F_x/m, \\ \ddot{y} &= F_y/m. \end{aligned}$$

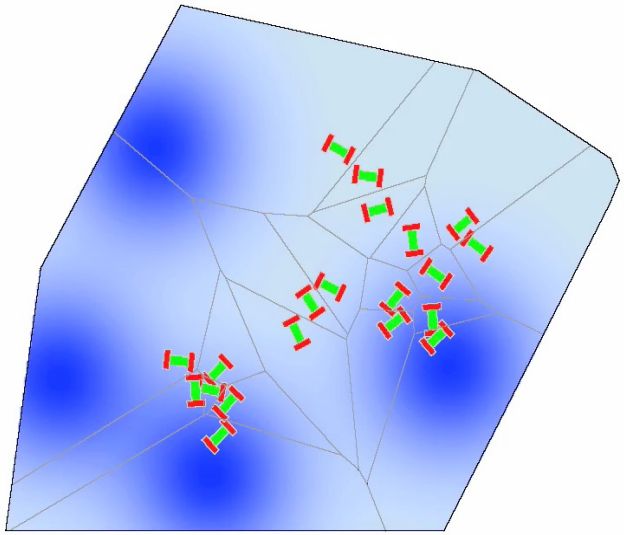
Differential drive model



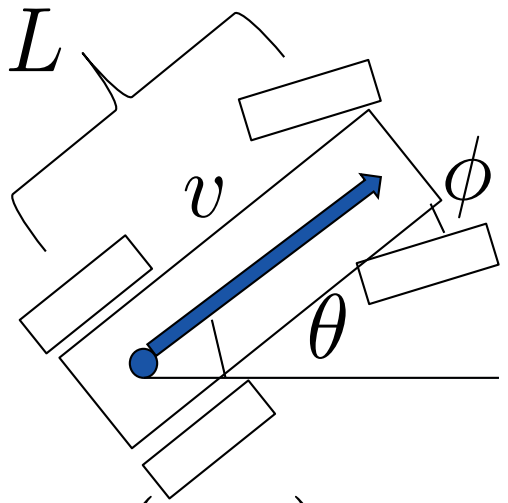
- In detail

$$\begin{aligned} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \omega. \end{aligned}$$

- Example: Coverage control



Kinematic Car model

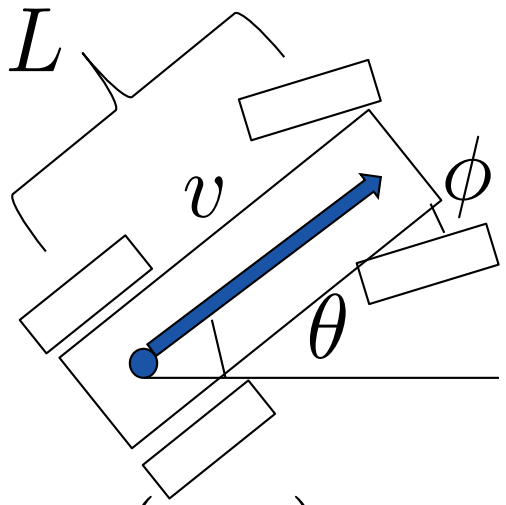


- In detail:

$$\begin{aligned} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \frac{v}{L}\tan\phi. \end{aligned}$$

- Example: Autonomous cars

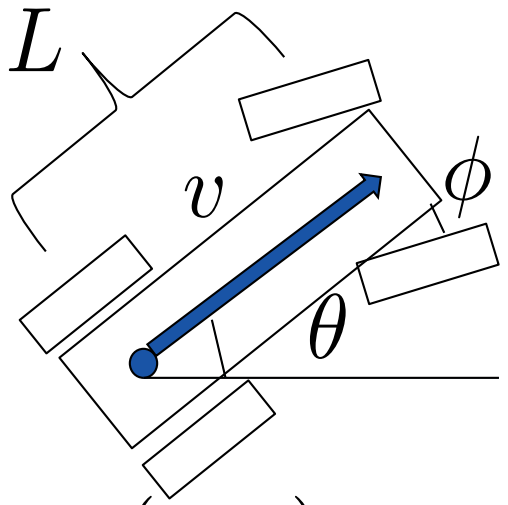
Dynamic Car model



- In detail:

$$\begin{aligned} \dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\theta} &= \frac{v}{L}\tan\phi \\ \dot{v} &= \frac{F}{m} \end{aligned}$$

Dynamic Car model with friction



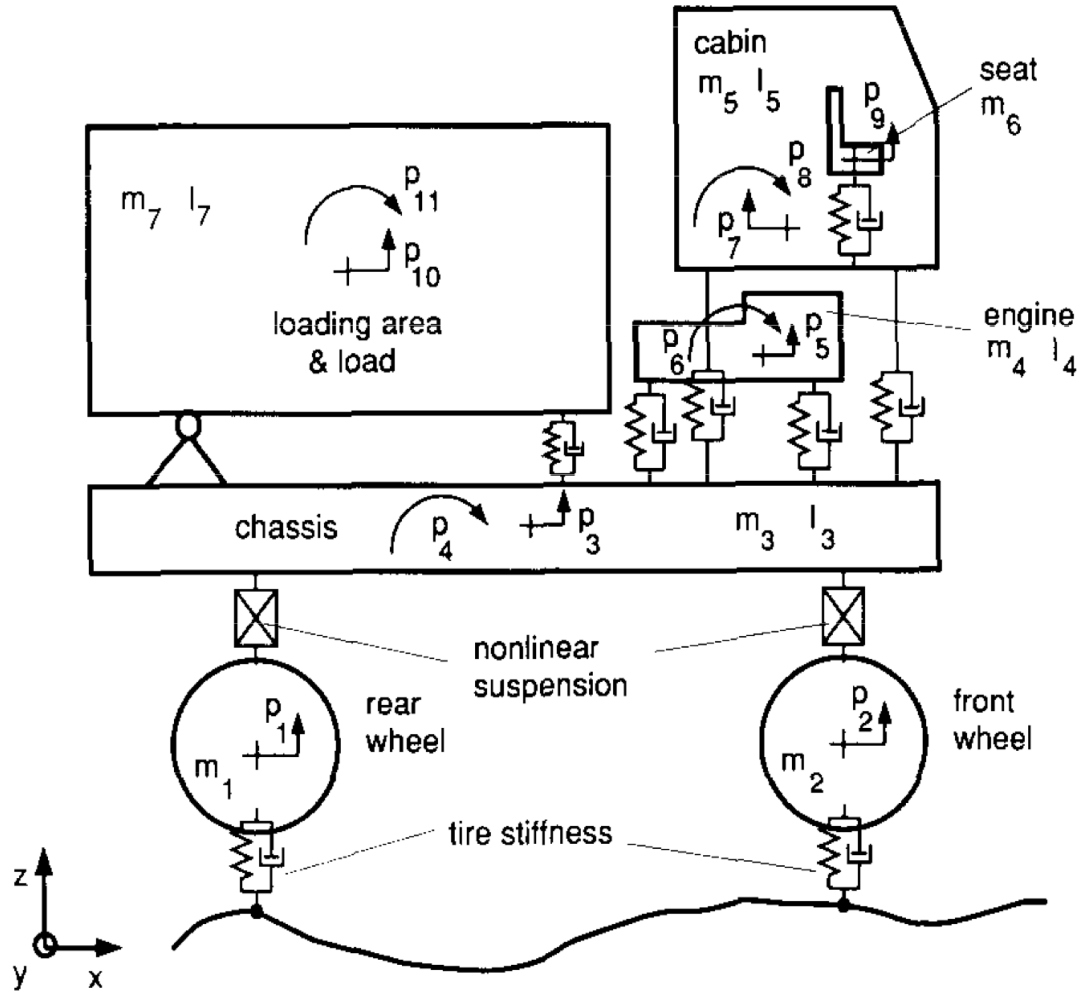
- In detail:

$$\begin{aligned} \dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\theta} &= \frac{v}{L}\tan\phi \\ \dot{v} &= \frac{F}{m} \end{aligned}$$

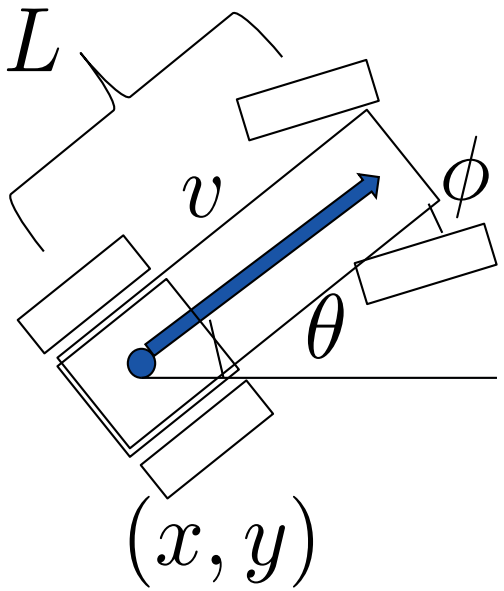
- Bound of steering angle and acceleration:
  - (car skids when inequality does not hold)

$$\sqrt{\frac{v^4}{L^2}\tan^2\phi + \frac{F^2}{m^2}} \leq g$$

More detailed models …



Summary of vehicle models



Complexity

- Discrete state space
- Kinematic point model
- Dynamic point model
- Differential drive
- Car model
- More detailed models

$$\begin{aligned} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \frac{v}{L}\tan\phi. \end{aligned}$$

Module 0: General Information

- Course timeline (and zoom links)
- Unity3d and C#
- Assignment Reporting
- Lectures
- Warnings
  3 pts
- Contact info

Module 1: Agent Models and Path Planning

- Assignment 1 description
- Motion Models
- Details on the Unity environment and how to run it
- A1 Meeting preparations
- A1 links to slides (anyone can edit)
- Meeting preparations Week 4 (slides, progress report, peer review)
  Jan 27 | 0 pts
- Meeting preparations Week 5 (slides, progress report, peer review)
  Feb 3 | 0 pts
- Meeting preparations Week 6 (slides, progress report, peer review) FINAL MEETING
  Feb 10 | 0 pts
- Meeting preparations Week 6 (Upload Your Best JSON Trajectories)
  Feb 18 | 0 pts

View Course Notifications

To Do

- Welcome to DD2...
  Jan 12 at 11:34am
- Simulation setup r...
  Jan 19 at 4:10pm
- Meeting preparati...
  Jan 27 at 8pm
- Meeting preparati...
  Feb 3 at 8pm
- Meeting preparati...
  Feb 10 at 8pm
- Meeting preparati...
  Feb 18 at 10am
- Upload A1 Report...
  Feb 18 at 11:59pm

Things to do until Friday…

- One person in each pair makes a Fork of the repo
- Get to know
  - Your teammate
  - Unity
  - C#
  - Simulation setut
- Understand the scripts controlling the vehicles
  - CarAI.cs
  - DroneAI.cs

Let’s check who you will work with…

- Talk to your team mate
  - One of you needs to fork the repo
    - https://gits-15.sys.kth.se/DD2438/MAS2023-Assignment-1
  - Decide a way to communicate
  - Discuss project management (scrum, planning, trello…)

The End