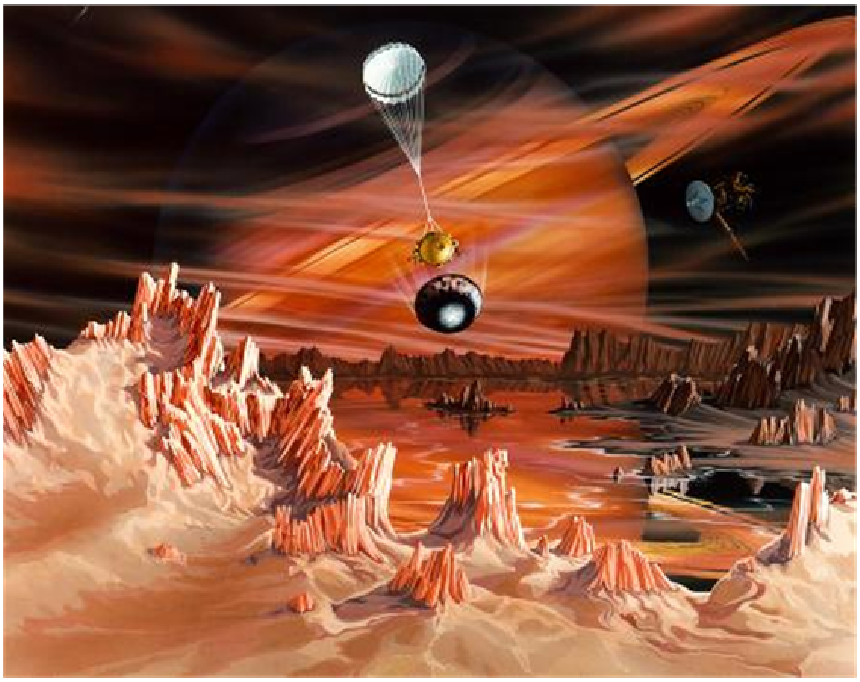
Project Manual Bachelor (Year 1)

Project 1-2

A Titanic Space Odyssey!



Period 1.4, 1.5, and 1.6
Academic year 2020-2021
BA: Data Science and Artificial Intelligence
Department of Data Science and Knowledge Engineering
Faculty of Science and Engineering
Maastricht University

Courses:

- Calculus
- ICT and Knowledge Management
- Data Structures and Algorithms
- Numerical Mathematics
- Software Engineering
- Logic

1 Project description

The central topic of project 1-2 is to bring a manned mission to land safely on Titan (one of Saturn's moons) and return back to Earth.

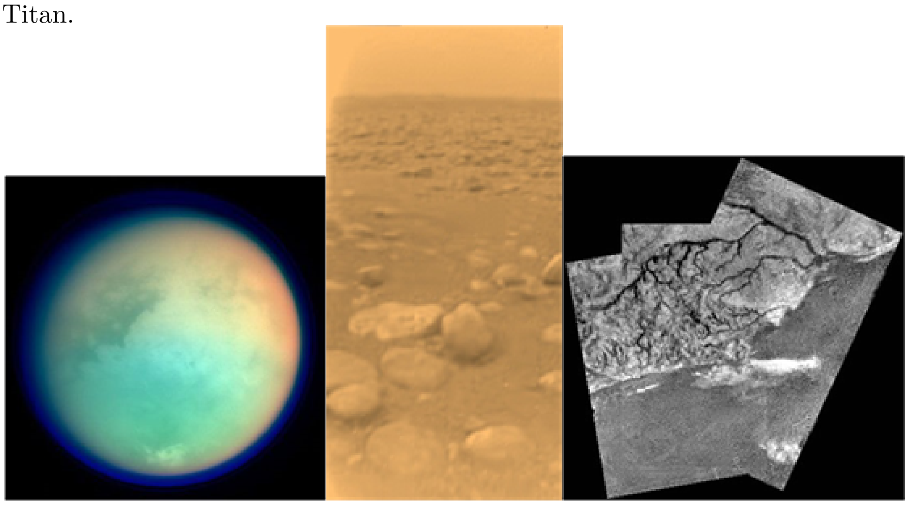
1.1 Background

Titan is the twentieth-most distant moon of Saturn and sixth-farthest among those large enough to assume a spheroid shape. Titan was the first known moon of Saturn, discovered in 1655 by the Dutch astronomer Christiaan Huygens. Titan is the largest moon of Saturn, the only moon known to have a dense atmosphere, and the only object other than Earth for which clear evidence of stable bodies of surface liquid has been found. Frequently described as a satellite with planet-like characteristics, Titan's diameter is roughly 50% larger than Earth's moon and it is 80% more massive. It is the second-largest moon in the Solar System, after Jupiter's moon Ganymede, and it is larger by diameter than the smallest planet, Mercury (although only half as massive). See [LR02] for more information.

1.2 Objective

You are on a mission to bring a manned mission to Titan ... and back!

In order to do so you need to develop and launch a spacecraft and bring it safely into orbit around Titan. You then need to land safely on Titan in a custom-built landing module. Finally, you leave orbit from Titan to bring the crew safely back to Earth. In preparation for this mission, an exploratory mission is planned: just like in Jules Verne's novel "*De la terre à la lune, trajet direct en 97 heures 20 minutes*" *("From the Earth to the Moon: A Direct Route in 97 Hours, 20 Minutes")*, from 1865, we will try to shoot a probe not by a rocket, but with a canon from earth: try to find a suitable angle and initial velocity such that the probe hits some spot on Titan.



1.3 Methods

For this project you need the following ingredients:

- Data on actual solar and planetary orbital positions and velocities

- Suitable physical constants like the gravity-constant
- Newton's laws of dynamics and mechanics and their mathematical descriptions
- Newton's law of universal gravity and the laws of Kepler and their mathematical descriptions

2 Phases of the Project

The project consists of several design and developing tasks, organised in three main phases. The main phases are defined:

2.1 Exploratory mission

In the first phase, you need to acquire the relevant data needed to build a model of the sun, the planets, and their moons, and then write a physics engine capable of computing the paths of the celestial bodies and the trajectories of your spaceships. You will then plan two missions to Titan.

The first mission is an exploratory mission to find a suitable landing site on Titan. You will launch a space probe using a ballistic missile (like a bullet from a canon) from Earth in such a way that it will land exactly (somewhere) on Titan. So here, the only forces on the probe are the gravitational forces; the probe has no engines so cannot exert a thrust. This approach was suggested in the novel [Ver1865].

Tasks:

1. Acquire data on the solar and planetary masses, and current positions and velocities. This data can be found on the NASA Horizons system [HORIZONS] and also in the file `solar_system_data-2020_04_01.txt` on the Student Portal.
2. Make a mathematical model of the solar, planetary and lunar orbits and the spacecraft motion.
3. Implement a physics engine for computing the paths of the celestial objects.
4. Make a visualization which plots paths of the spaceship and terrestrial objects.
5. Plan an exploratory single ballistic shot mission to Titan.
   (a) The probe must be launched on 1st April 2020.
   (b) The probe must be launched at a point on the surface of the Earth.
   (c) The intial velocity of the probe (relative to the Earth) must be at most $60\,\mathrm{km\,s^{-1}}$.
   (d) The probe must reach Titan before 1st April 2021.

Solver Requirements

1. Your solar system model must use the same (x, y, z) coordinate system used by NASA Horizons, with the coordinate origin being the Solar System Barycenter (SSB).
2. You must model the Sun, the 8 planets (note that Pluto is not a planet), the Earth's moon Luna, and Titan.

3. You must use standard SI units of kilograms, metres, and seconds. (In particular, distances must be in metres, not kilometres or astronomical-units.)

4. You must give the launch position and velocity objects relative to the Earth's position and velocity (in seconds since launch).

5. You must give the time of the closest approach of the probe to Titan.

6. You must use the Euler solver with a fixed step-size, which you must state.

7. You must output the complete trajectory of your space probe (i.e. the position at each time-stamp).

Testing

1. You should test your physics engine with various step-sizes. Is there a minimum step-size needed for reasonable results? How do you think the accuracy improves as you decrease the step-size? You may wish to compare your results against the true positions of the planets.

2. For testing, you may omit Mercury, Venus, Uranus and Neptune, and (for performance reasons) the Moon and Mars.

Implementation Requirements Your code will be automatically tested to check that your calculations are correct. To facilitate this, you must conform to the API specified on the StudentPortal in the `*Interface.java` files.

Minimum Requirements To achieve a passing grade in this phase, we expect you to:

- Have elementary graphics in 2D (minimal components: the Sun, the planets (you may zoom in to omit Uranus and Neptune), the Moon, Titan and your probe.
- Have a working physics engine giving accurate results.
- Find initial launch parameters which yield a reasonably close approach to Titan.
- Be able to demonstrate that everyone was involved sufficiently in the design and (especially) programming.

  An overview of who did what needs to appear in the presentation.

To obtain a higher grade, you will need to work above these minimal requirements, such as:

- Advanced graphics features (such as zooming-in to follow the spacecraft at launch and landing).
- A fully-correct physics engine.
- An exact landing on Titan.
- A clean design and well-organised code.
- An especially well-structured and clear presentation.

2.2 Space mission to Titan

In the second mission, a fully-fledged multiple-stage rocket is to be launched from Earth in such a way that it enters a geostationary orbit around Titan. But now you have the possibility to use your rocket engine on the way to perform corrections to the trajectory. Take into account that each such intervention requires mass to be burned by the rocket engine, mass that has to be brought from Earth. The force resulting from the thrust of the engine can be found in the reference [NASA] Though it would undeniably be more easy and less expensive to organize a one-way mission to Titan, generally speaking, astronauts like to return home! Therefore, you should also design the return journey that brings the astronauts back to Earth from Titan.

To achieve the best possible planning, you must implement advanced differential equations solvers to plug-in to your physics engine. To make sure nothing goes wrong with the mission, you must implement unit tests of your code.

Tasks:

1. Implement higher-order differential equation solvers, including a 4th-order Runge-Kutta solver and a Verlet solver, and plug these in to your physics engine.

2. Test your solvers with different step-sizes, and compare the accuracy of the different solvers with different step-sizes. Comment on how the results of your tests compare with the theoretical order of the methods. (You may test your solvers on problems other than solar-system simulation.)

3. Realize a return mission from Earth to Titan and back. You must enter and leave orbit around Titan, at a height of between 100km and 300km. You should aim to minimise the fuel costs, while still complete the mission in a reasonable time.

4. Write simple JUnit test cases and report code coverage statistics (aim for 100% coverage)

Implementation Requirements You differential equations solvers will be automatically tested on the space probe mission from Phase 1. Details will be given on the Student Portal.

Minimum Requirements To achieve a passing grade, you must

- Implement both requested higher-order differential equation solvers, which should give accurate results.

- Plan a mission which (at a minimum) performs a close fly-by of Titan and returns near to Earth.

2.3 Landing on Titan

In this phase, you mainly work on the part of the mission in which you actually land on Titan. You should also continue to improve the planning of the rest of the mission, as started in Phase 2.

Land on Titan at specific coordinates using a landing module. You start from the spaceship that is in orbit around Titan that you have found; this means that you will start with a nonzero horizontal velocity relative to the surface of Titan. You have to consider the rotation of the landing module and the atmosphere of Titan to make a safe landing. The wind direction and speed may vary during landing. You therefore investigate an open and a closed loop controller.

Tasks:

1. Write a physics engine to simulate the motion of the landing module. The engine should be able to handle rotations of the landing module as well as horizontal and vertical motion.

2. Devise controllers for landing on Titan. You should try two strategies:

   (a) An *open-loop* controller, in which you determine in advance the input functions $(u(t), v(t))$ such that the landing module lands safely starting stationary from a given initial position (x_0, y_0).
   (b) A *feedback* controller for the landing module. The feedback controller determines the thrust based on the current position and velocity. In other words, we can write

   $$u \ = \ g_u(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}) \text{ and } v \ = \ g_v(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})$$

   Your new controller must be able to land given a moving initial condition.

3. The landing module is buffeted by wind as it approaches. Make a simple stochastic model for the wind, and see which of your controllers land successfully in this case. Perform experiments to determine the probability of a successful landing depending on wind strength for each controller.

Additionally, you may need to correct mistakes in your planning from Phase 2, and may wish to improve your mission by using advanced differential equation solvers, optimising fuel costs and travel time, and/or planning a fly-by of Jupiter.

Additional Tasks:

1. If testing your higher-order differential equations solvers from Phase 2 indicates they have bugs, you *must* correct these in Phase 3.

The following tasks are optional, but completing one or more of these may improve your final grade:

2. Implement adaptive differential equation solvers, which automatically take more steps during launch, entering/leaving orbit, and performing manoevers.

3. Optimise the fuel use and/or time of your mission.

4. Improve the results of the previous phase and plan a mission to Titan that passes Jupiter on the way to Titan.

Minimum Requirements To achieve a passing grade, you must

- Implement a physics engine for the landing module, attempt a landing using both open-loop and closed-loop controllers, and achieve a successful landing with at least one controller.

- Have at least one *completely correct* higher-order solver, as verified by the automated testing.

3 Physics and solving differential equations

Differential equations The solar system is an example of a dynamical system whose motion is governed by a system of *differential equations*

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)). \tag{†}$$

Here, $\mathbf{y}(t)$ is an n-dimensional vector describing the *state* of the system at time t. The notation $\dot{\mathbf{y}}$ denotes the time-derivative of $\mathbf{y}$, so $\dot{\mathbf{y}} = d\mathbf{y}/dt$. Since from different states, the subsequent motion also differs, to obtain a unique evolution of the system, we also need to specify an *initial condition*, which is the state at some given time: $\mathbf{y}(t_0) = \mathbf{y}_0$.

For many systems occuring in physics, the motion is determined by a *second-order* differential equation

$$\ddot{\mathbf{x}}(t) = \boldsymbol{f}_a(t, \mathbf{x}(t))$$

where $\mathbf{x}$ is the *position* vector (also denoted $\mathbf{p}$), and $\ddot{\mathbf{x}} = d^2\mathbf{x}/dt^2$ is its second derivative, the *acceleration* $\mathbf{a}$. Here, the acceleration is a function $\boldsymbol{f}_a$ of position and time *only* (this is the case for systems which conserve energy). Since different initial velocities yield different trajectories (e.g. dropping a ball is different from throwing it!) the state $\mathbf{y}$ comprises both the position vector and the *velocity* vector $\mathbf{v}$, so $\mathbf{y} = (\mathbf{x}, \mathbf{v})$. Since $\mathbf{v} = d\mathbf{x}/dt = \dot{\mathbf{x}}$, the equations of motion can be written as a first-order system

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t); \qquad \dot{\mathbf{v}}(t) = \boldsymbol{f}_a(t, \mathbf{x}(t)).$$

From this, the system can be expressed in the form (†):

$$\dot{\mathbf{y}}(t) = (\dot{\mathbf{x}}(t), \dot{\mathbf{v}}(t)) = (\mathbf{v}(t), \boldsymbol{f}_a(t, \mathbf{x}(t))) = \mathbf{f}(t, (\mathbf{x}(t), \mathbf{v}(t)) = \mathbf{f}(t, \mathbf{y}(t)).$$

The Solar System and your Spacecraft The motion of the sun, planets, moons, asteroids and your space probe are governed by Newton's laws of motion and the force of gravity. For each of these celestial bodies, you need to know its mass m_i, its position $\mathbf{x}(t) = (x_1(t), x_2(t), x_3(t))$ and its velocity $\mathbf{v}(t) = \dot{\mathbf{x}}(t) = d\mathbf{x}(t)/dt = (v_1(t), v_2(t), v_3(t))$. With this model it is possible to calculate the force on the i^{th} particle with mass m at any given moment in each point of space using Newton's law of universal gravity:

$$\mathbf{F}_i^G = \sum_{j \neq i} G m_i m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{||\mathbf{x}_i - \mathbf{x}_j||^3}$$

where G is the universal constant of gravity. Note that $\mathbf{x}$ and $\mathbf{F}$ are 3-dimensional vectors.

The motion of the i^{th} body is given by the laws of Newton. The acceleration $\mathbf{a}_i$ of the body times its mass m_i is equal to the gravitational force $\mathbf{F}_i^G$ acting on it:

$$m_i \mathbf{a}_i = \mathbf{F}_i^G.$$

The acceleration is the time-derivative of velocity, so is the second-derivative of position,

$$\mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i.$$

The equation of motion for the body can be written as a *second-order* system

$$m_i \ddot{\mathbf{x}}(t) = \mathbf{F}_i^G$$

or as a system of *coupled first-order equations*

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t); \qquad m_i \dot{\mathbf{v}}(t) = \mathbf{F}_i^G.$$

In addition to the force of gravity, your manned spaceship has an engine which can be used to impart an additional force $\mathbf{F}^T$, yielding total force

$$\mathbf{F}_S = \mathbf{F}_s^G + \mathbf{F}_s^T.$$

If the engines provide a very large thrust over a short interval of time, then the effect can be well-approximated by providing an *impulse* $\mathbf{I}$ to the velocity:

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{I}}{m} \quad \text{where } \mathbf{I} \;=\; \int_t^{t+\delta t} \mathbf{F}(\tau) d\tau$$

To obtain the trajectoy of the probe or spaceship, you must solve these equations for the celestial bodies and your spacecraft. Since Euler's method and other standard solvers (with the exception of the Verlet solver) work with first-order systems, you should use the first-order formulation. The state vector $\mathbf{y}$ must contain the positions and velocities of every object in the system

$$\mathbf{y} = \big(\mathbf{x}_{\text{ship}}, \mathbf{v}_{\text{ship}}, \mathbf{x}_{\text{sun}}, \mathbf{v}_{\text{sun}}, \ldots\big) \text{ or } \mathbf{y} = \big(\mathbf{x}_{\text{ship}}, \mathbf{x}_{\text{sun}}, \ldots, \mathbf{v}_{\text{ship}}, \mathbf{v}_{\text{sun}}, \ldots\big).$$

Note that $\mathbf{y}$ does not have to be stored as a list of numbers. In Java, a custom data structure containing positions and velocities of all bodies might be more appropriate.

Solving differential equations The standard approach to solving the first-order differential equation (†) is to use a *time-stepping* approach. The value of $\mathbf{y}(t)$, the state at time t, is approximated at times $(t_0, t_1, t_2, \ldots, t_n)$ by values $(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ with $\mathbf{y}_i \approx \mathbf{y}(t_i)$. The *step size* for the i^{th} step is $h_i = t_{i+1} - t_i$. The result can be presented as a list of pairs of times and states,

$$\big((t_0, \mathbf{y}_0), (t_1, \mathbf{y}_1), (t_2, \mathbf{y}_2), \ldots, (t_n, \mathbf{y}_n)\big).$$

A simple (first-order) numerical scheme for solving the differential equation (†) is *Euler's method*. Using the approximation $\mathbf{y}(t+\delta t) \;\approx\; \mathbf{y}(t) + \delta t\, \mathbf{f}(t, \mathbf{y})$ we obtain the update rule

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \mathbf{f}(t_i, \mathbf{y}_i)$$

Landing on the moon Titan For simplicity, only consider a spaceship moving in two dimensions. The spaceship has its main thrusters mounted aft, and four smaller thrusters mounted on its side to provide rotational control.

The differential equations describing the motion are:

$$\begin{aligned} \ddot{x} &= u \sin(\theta) \\ \ddot{y} &= u \cos(\theta) - g \\ \ddot{\theta} &= v \end{aligned}$$

Here, x is the horizontal position, y the vertical position, η the angle of rotation, u the acceleration provided by the main thruster, v the torque provided by the side thrusters, and $g \approx 1.352\text{ms}^{-2}$ the acceleration due to gravity on Titan. The maximum acceleration provided by the main thruster is $u_{\max}$ and the maximum torque is $v_{\max}$.

The landing pad is located at $(x, y) = (0, 0)$. In order for a safe landing to occur, the state $y = 0$ must be reached with the variables satisfying the following conditions:

$$\begin{aligned} |x| &\leq \delta_x \\ |\theta \mod 2\pi| &\leq \delta_\theta \\ |\dot{x}| &\leq \varepsilon_x \\ |\dot{y}| &\leq \varepsilon_y \\ |\dot{\theta}| &= \varepsilon_\theta \end{aligned}$$

Take tolerance values $\delta_x = 0.1$, $\delta_\theta = 0.02$, $\varepsilon_x = 0.1$, $\varepsilon_y = 0.1$ and $\varepsilon_\theta = 0.01$.

References

[Ver1865] Jules Verne, *De la terre à la lune*, 1865.

[HORIZONS] *HORIZONS: Solar System Dynamics*, NASA Jet Propulsion Labori-tory, https://ssd.jpl.nasa.gov/horizons.cgi.

[NASA] *Rocket Thrust Equation*, Glenn Research Centre, NASA http://www.grc.nasa.gov/WWW/K12/airplane/rockth.html.

[LR02] Ralph Lorenz & Jacqueline Mitton, *Lifting Titan's Veil: Exploring the Giant Moon of Saturn*, Cambridge University Press, 2002.

Appendix

Project phases

The project assignments in block/period 1.4 and 1.5 are preparatory for the concluding part of project 1-2 which takes place in block 1.6. During week 6 of block 1.4, you will work full-time on phase 1 of the project. In block 1.5 and 1.6, you will start working on the project immediately without having a separate project opening. At the end of each phase, you will present your project work to the examiners and your tutor. In the last project phase, this presentation will additionally be in front of your fellow students. Furthermore, at the end of the last phase, you have to hand in a project report.

Assessment Moments

Presentation First Phase (PFP)/Presentation Second Phase (PSP) The intermediate presentations conclude the first and second phase of the project. The group will present in front of the examiners and the tutor. The group should bring an additional laptop with code ready to be checked and run (if the presentation takes place in person). The group needs to wait in a Zoom/Collaborate Ultra session until the examiners join that session (if the presentation takes place in online). When entering the session, the examiners should find the group sharing a screen with the presentation and ready to start. The groups have 10 minutes to present (including a short and to-the-point demonstration if needed), 10 minutes are additionally reserved for questions/feedback from the examiners. The presentation needs to follow the format of a scientific presentation and has to include a planning for phase 2 (Gantt chart) and an overview of who did what.

Pre-Examination This is the last opportunity for the group to receive feedback from the examiners. The students can receive feedback on their report draft, presentation and/or progress of their project. The students are expected to be prepared with a list of things they want to discuss. The pre-examination is not graded.

Final Presentations (FP) The final presentations will be attended by the examiners, tutor and the fellow students. Thus, the presentations will not contain that many technical details. These will be discussed in the product and report examination.

Product and Report Examination The examiners can give feedback on the submissions and test the understanding, contribution and knowledge of the group (members).

The phases are assessed according to the assessment forms that can be found on Canvas. As stated in the rules and regulations (see Canvas), "the first grade is issued after the presentation first phase and accounts for 15% of the final grade. The second grade is issued after the presentation second phase and accounts for 15% of the final grade. The third grade is issued after the final assessment at the end of the third block and accounts for 70% of the grade". More detailed information on the assessment (including individual grade reduction because of missing mandatory events) can be found in the rules and regulations uploaded to student portal. It is highly recommended to read them carefully. To pass the project, you need a weighted average of the three grades higher or equal to 6. It is not required to obtain at least a six in each of the phases. A weak performance in one of the phases can potentially be compensated in another phase.

Project coordination

The examiners of project 1-2 are: Peter Collins, Nico Roos, Kateřina Staňková, and Christof Seiler. Katharina Schüller is coordinating the project. Otti D'Huys, Chiara Sironi, Daniel Campora, and Katharina Schüller are tutoring the groups. For questions regarding the organisation, please email to k.schuller@maastrichtuniversity.nl. For other questions, contact your tutor. They will eventually forward your question to the examiners or the project coordinator. General information, information on the courses and schedules is to be found on Canvas.

Project meetings

The aim of a project meeting with the tutor is to continuously track the status of the project by looking backward and forward. Agreements made are checked, new appointments/agreements are made. Moreover, the planning will be checked. In case of deviations, an analysis of the situation will be made in order to trace the causes. Project meetings are scheduled on a fixed date and time. An agenda is set up by the group in advance for each meeting. Minutes will be taken in each of the project meetings. The project meetings are mandatory (attendance will be tracked) and participation will be assessed.

Agenda and Chairperson

The agenda template is uploaded to Canvas. This agenda may be changed, influenced by the project or specific situation. Still all main points need to be discussed during the meeting. The chairperson changes every project meeting. For each meeting, the group decides in advance to the project meeting on who is going to take this role and prepares an agenda (see example on Canvas). Preparation of the agenda includes filling out the sections (or the sections of the adjusted agenda) and uploading the agenda to Canvas in advance to each project meeting. During the meeting, the chairperson takes care that everybody can participate (e.g., invites other team members to talk about the tasks they have done), that the atmosphere in the meeting is safe and open, that decisions are made and that time constraints are not exceeded. The chairperson guides through the meeting based on the agenda.

Minutes and Secretary

The secretary takes care of the minutes. He/she summarizes the discussion and decisions of the meeting such that someone not attending the meeting can understand the findings of the meeting. The points listed in the minutes need to correspond to the points on the agenda, i.e., the headings in the minutes are the same as in the agenda. The minutes have to be provided with date, group number and the (full) names of those present and absent. A template for minutes is uploaded to Canvas as well. Furthermore, the group is responsible for uploading the minutes on time (see due date of assignment on Canvas).

Overview of tasks

Every group is expected to create an overview per phase of what every group member did during the project. It is highly recommended to start keeping track of this at the beginning of the project. It is uploaded at the end of each phase to Canvas. An example can be found in the project opening slides uploaded to Canvas. If you do not agree on an overview, please let your tutor know in advance.

Planning

Every group is expected to make a planning in form of a Gantt chart at the beginning of each phase for the complete phase. Groups of projects 1-1 and 1-2 only need to make the planning in phase 2 and phase 3. The planning should give an overview on when the group expects to achieve which milestones. It is uploaded to Canvas before the second project meeting in that phase (see Canvas for concrete deadlines). An example for a planning is uploaded to Canvas.

Schedules

The concrete schedules for the project meetings and the presentations will be uploaded to Canvas.

Rules and Regulations

The rules and regulations can be found on Canvas. They include, e.g., the consequences for missing mandatory events (project meetings, skill classes, assessment moments). Please read them carefully.

Missed Meetings

Missing mandatory events will cause a grade reduction of your final grade. The details can be found in the rules and regulations. You need to keep track on your missed meetings yourself. To address the most important consequences:

- One project meeting per phase can be missed without consequences
- If you miss more than one project meeting in a phase, you need to have a valid excuse for all meetings. If you lose your token of one free-to-miss project meeting because of oversleeping, the next project meeting will immediately lead to a grade reduction no matter if you have a valid reason for missing that one.
- If you miss a skills class, it will immediately lead to a grade reduction of 0.5 points (no matter if you have a valid reason for missing it) as it hinders you to fulfil the intended learning outcomes of the project.

- If you miss 3 or more project meetings in a phase, you will receive an NG for the project (you have to re-do the project).
- If you miss 3 or more skill classes, you will receive an NG for the project as well.
- If you end up with an NG for the skill classes or you drop below a passing grade, you can request a hardship (please send an email to dke-exams@maastrichtuniversity.nl).
- If you want to be excused for missing more than one project meetings, you can ask the Board of Examiners to excuse you (please send an email to dke-exams@maastrichtuniversity.nl). But first, you need to check, whether you have a chance to be excused by going through the hardship clause document uploaded to Canvas.