**Code of Conduct**

In a Code of Conduct you discuss with each other what you expect from each other and from the collaboration. Everyone participates in this and supports the agreements that you draw up together. A Code of Conduct is a flexible document. If after some time it appears that certain agreements are not realistic or applicable, then it is important to discuss this in the group and adjust the agreements if necessary.

**Assignment description:**

In your own words, describe what you need to do as a group in this course.

**Since we will be working together for a couple of months to create a software project, it is very important to have a concrete weekly schedule / deadline as a group to ensure we have a finished product at the end. To achieve this, we need regular communication with each other to keep track of our own individual tasks and constantly review each other’s code.**

**Target or ambition level:**

What grade are you working for?

**We would like to get an 8, but the main goal is to work on the project as a group and improve ourselves. We’re also looking to develop something as close to completion as possible, so we want to get everything in the backlog done, in addition to a bug-free experience for the user.**

**Products:**

What should you deliver at the end? On which platform do you share which documents (Discourse/Miro/MS Teams)? What standards must the work submitted meet?

**A working energy quiz prototype (does not have to be a polished, fully fledged application). In addition to the Mattermost platform, we have our separate Discord server with to-do lists, announcements, and meetings, as well as online resources such as tutorials, Java documentation or useful web pages. Some important documents are the weekly backlog check-up and the final HCI report. We post or update the documents mainly through Google Docs as it allows shared viewing and editing easier than any other platform we’re familiar with. We aim to have the documents as action-specific as possible so that there is no ambiguity or confusion when checking-up on each other’s status. Moreover, we’re also looking for consistency so everything is going to be proof-read, double checked and properly formatted.**

**Planning:**

How do you ensure that each group finishes everything on time? Did you clarify who will have a final say in the final deliverable and submits it to Brightspace *on behalf of the project group?*

**We do not have “groups” in the traditional sense because we thought it more appropriate for everyone to work on every facet of the project. Therefore, everyone does issues individually. We will use the backlog to estimate how long each feature will take to implement. From there on, issues are going to be made on GitLab for each feature or functionality needed. There may be cases where multiple people contribute to a merge request as it needs some quick fixes or two people collaborate because the issue is more complicated or two issues deal with the same aspect of the project and are similar in scope and we want to reduce redundancy and conflicts. Using that and frequent meetings, we can prioritize the necessary issues and have a proper work flow. For deliverables, we will discuss with each other when we think the work is done, and if we all agree, Elias will submit the necessary files on Brightspace for the team.**

**Behavior:**

How do you treat each other in the group? How do you handle disagreements within your group? Could your guide or student assistant be involved in reaching consent? What do you do if someone is late during a group meeting?

**Everyone should be treated fairly, respectfully and in a friendly manner, that is to say, their opinions should always be taken into consideration and their work appreciated or constructively criticized. If disagreements arise and a consensus can’t be reached the matter can be taken to a vote to see what the majority of the group thinks about the issue(s) at hand. In the case of something more subjectively oriented such as design or whatnot then compromises could be reached and if something is more “technical” then the TA could be brought forward to clear some things up. If someone is late to a group meeting then they should be contacted. If they eventually answer and such incidents are rare for said individual then there really is no problem, especially if they have some sort of reason. If they do not answer at all or if the incident is not isolated then there are some problems. The rest of the team should bring the issue up with him and discuss the importance of his attendance on time and maybe reach a compromise on the meeting schedule.**

**Communication:**

In what ways do you communicate with each other as a group and among yourselves? (in the studio/MS Teams/Miro/Discourse)

**We have 3 weekly meetings: mandatory on-campus meeting on Tuesday and online meetings on Thursday and Sunday. Discord is used for the weekly meetings and generally talking to each-other but for the latter we may also use WhatsApp as it’s usually more convenient.**

**Commitment:**

How do you determine the quality of each group's work, so that each group delivers the same quality?

**Regarding assignments, everyone looks over what we’ve written and proofs it. Everyone contributes with comments to any merge request. These comments may be about the direction of the merge request (alternatives for how the interface looks, for example or solutions to problems) or about the quality of the code, which is assessed in the following way: cleanness (how easy it is to read, whether there’s duplicate / redundant code or not), functionality (whether it works as intended and implements the features we decided on), correctness (whether any bugs arise from it) and efficiency (as we do not want the client to have a slow experience with the game).**

How do you measure the commitment of the chairs and minute takers?

**Agree beforehand on who does what and make sure that everyone agrees with it. We can see if the quality of their agendas / notes is good enough (before meetings) and we make a list of improvements so that the chairperson / note-taker has better documents prepared next time as well as better points to be made during the meeting.**

**Meetings:**

How often will you meet as a group? What preparation is needed for the meetings?

**3 times a week. Depends on the meeting. Usually, we’d either need the agenda ready or announcements made on discord to inform everyone of what we’re talking about. Moreover, everyone should also be familiar with the week’s progress so we can have proper discussions about the ongoing work.**

**Decision-making:**

How do you make decisions? By majority vote or by consensus?

**If we can reach a consensus, we’ll of course go by it. However, if we can’t reach it, we will go by majority vote.**

**Dealing with conflicts:**

How do you handle conflicts within the group?

**Firstly, we talk about the conflict in a meeting. Of course, this depends on the nature of the conflict as well, as some conflicts (small disagreements) could be easily solved outside of meetings. If we cannot work it out, then we contact our TA.**

**Guidance:**

What do you expect from the teacher's and/or student assistant’s guidance? What do you want feedback on, on the content or on the collaboration?

**Since the TA’s are only here to assess our teamwork, we don’t expect them to know all about the technical details of the project. That being said, we expect the TA to gives us a rough idea of our progress as a group so we know whether we need to up our pace or balance our issues better. As for technical issues, we can always look online for answers. However, if that doesn’t yield sufficient results, we could ask one of the lecturers on Mattermost.**

**Consequences:**

What are the consequences if a participant in the group does not keep the agreements?

**Work it out as a team before going to the TA. The consequences for the team will be that the rest of the team will have to work harder which in turn disrupts the team cohesion. For the person in question, they might get removed from the group if deemed necessary by the TA.**

**Success factors:**

What makes your team a dream team?

**The most difficult thing to achieve in a team is constant, clear, transparent communication with each other. If we can achieve this and maintain throughout the project, we can indeed be a dream team. Moreover, we also need to interact with each-other’s work for everyone’s skills to be used to the fullest. That is the reason why we put an emphasis on checking others’ issues and reviewing them or helping each-other when needed.**

**Gitlab rules:**

What have we agreed on with regards to using GitLab?

**General format for the issue descriptions:**

* User story: {user\_story}
* Definition of done: {checklist}
* (Optional) Idea for implementation: {idea}

**Commits & merge requests**

All merge requests have to be approved by two people different from the uploader.

Officially one person commits and another specifically designated person reviews but, in practice, other people review the commit too.

Minimum 10 comments for merge requests unless they deal with fixes or small improvements.

For each issue, create an empty merge request and branch and list it as work in progress.

When we actually merge issues, also add the commit id in the description.

**Agenda & notes naming convention**

The file names for notes and agendas should follow the following format:

Agendas: mm-dd\_agenda.md

Notes: mm-dd\_notes.md

**Testing application before merge request:**

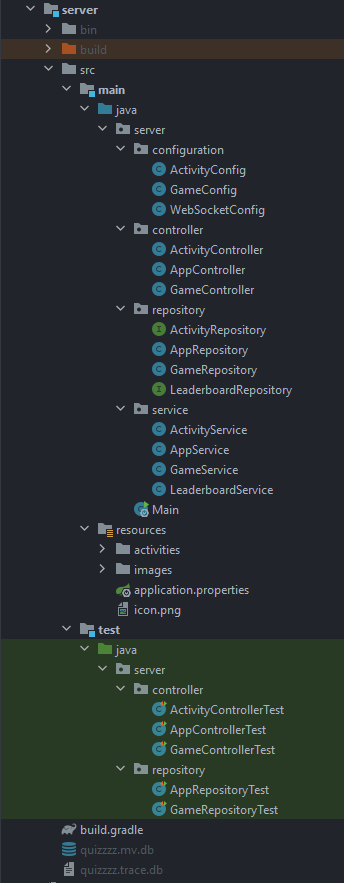
During development of the code, check regularly whether the application and all its related functionalities are working; the WebSockets and API, especially, are sensitive to changes.

Before requesting an official review, resolve any merge conflicts that would require the reviewer to have to change the code again.

**Code structure and organization:**

What have we agreed on with regards to organizing our code?

**The structure for the server code:**

  
  
  
  
 **/configuration** – f.e. WebSocketConfig

**/controller** – REST controller, the API layer

**/repository** – JPA repository

**/service** – the business logic

**/resources/activities** – activity .json files and images

***/test/java*** *should copy the structure of the* ***main/java*** *package*

*Activity database file should be in the* ***server*** *folder.*