Organization

You are to be organised in groups of 4, with the fewest exceptions of groups of 3.

In order to provide inter-group and intra-group feedback, we will release anonymous google form to provide such feedback.

A W3C-like Working Group

You are organized in groups of 4 people, with some exceptions of 3 people.

Each group elects a W**orking Group Leader (WGL)**: that person is responsible of voicing the group's concerns with the **Working Group (WG)**.

The class also elects one **Working Group Coordinator (WGC)**, who is responsible of making the various meetings of the WG productive.

WG Practicals

The Working Group is thus constituted by the assembly of all WGLs plus the WGC.

After a while, we will devote classes to WG meetings, where you will discuss the current state of the DPC, draft changes, discuss modifications, etc.

Ideally, the second part of the semester one class per week will be class, and one class per week will be a WG meeting.

Running a WG is not simple, it is a strenuous task that you don't know how to do, and it will take time to make the WG meetings productive. For example, you may have to devise commonly-shared code: these tasks are also discussed in the WG meetings, and together you decide who should implement (and test) this code.

Working Group Leaders (WGL)

Each group elects its leader, who will represent the group at each WG meeting. They are responsible for voting changes to the DCP, and they are the only ones that can vote.

Working Group Coordinator (WGC)

The class must elect one working group Coordinator. The Coordinator is responsible of drafting each new version of the DCP. Despite the fact that they come from a group, they must act super-partes.

This role requires some extra work and this is reflected in the grading with a bonus to the total result. For example, the WGC has to supply a summary of each WG meeting. They can delegate that summary to scribes, since they have much on their plate, or whatever, but the summaries are mandatory. Typically, scribes also get rewards for their extra work.

The WGC has access to a YouTube playlist with videos from their predecessors from previous years, each sharing 3 tips. It is expected that the WGC compiles one such video before next year's class too.

Working Group Process

Ideally, once the project description is out and the faulty DPC is known, each group starts implementing them and discovers faults in the specifications. Ultimately, the WG will produce an alpha version and then a beta version, i.e., a final version of the DPC that will be frozen.

Freezing the Specification

Approaching the end of the semester the DPC become *frozen* (see the dates below). That is, no further modifications are possible (except for typos and the like) unless there are serious pitfalls in its design.

In order to unfreeze the the DPC, the WGC must have the consensus of the WGL (≥ 51%) and they must contact me. If the request sounds reasonable, the DPC is unfrozen for 1 meeting, then it is fixed and then frozen again.

Note that it is in your best interest to not unfreeze a DPC since its changes means changes to your code. Also, too many unfreezes will reflect poorly on the evaluation of the WGL and of the WGC. However, you have all the time and capabilities to reach a stable DPC in time, so that no unfreezing is necessary.

WG Evaluation

The WG is expected to work in good faith. Failure to do so (for example by providing specs that are nonsensical or clearly flawed) will incur significant penalties in the final score.

Tips on running the WG

There can be different ways for the WG to be effective. Some examples include trying to misuse the spec and see what bad interactions can happen. State diagrams can also be effective at understanding what is supposed to happen when different functionalities of the DPC interact with each other.

Sharing Code

Each group's project need to use Drones that have been developed by other groups. The interaction with other groups' code requires to set up a GitHub repository, whose usage is hinted below, and described in class.

The code pushed by different groups needs to follow some naming conventions, which are also described below.

Finally, here we also describe the Faire process, which binds each group to other groups' Drones implementations.

Class Code Repository

Each group's Drone, plus any shared code will be hosted on the group public GitHub repository. It is the responsibility of each group to build such a repo, and to add me, my GitHub handle is squera.

You need to add a file to your project in `.cargo/config.toml` with the repo URL, and then edit `Cargo.toml` and include all the repos whose Drones you rely on.

Code Naming

Each of the Drones your group publishes must be named after its group.

Software Faire

In order to know what Drones your group will use, towards the end of the semester we'll have a Faire. During the software Faire, each group will:

- Showcase their Drones.
- Decide which other groups' Drones they are going to use.

You have to come with working versions of your Drones, show that they comply to the agreed Specification and convince other groups to commit to your code and not to other groups'.

During the Faire, each group must commit to use 10 (ten) Drones from other groups.

By the end of the Faire, we will ask and register each group's choices. These choices are final, and in the final project evaluation, your Constellation must run with those Drones.

The first three groups whose code is committed-to the most will have a grade bonus.

Customer Support

Because you are both using other people code, as well as supplying code to other people, it is expected for each group to provide some form of customer support. You are free to implement your customer support however you choose, you have to inform your customers upon the day of the Faire.

Because customer support is important, know that all projects discussed in January will only be registered in February. There are no deadlines (that we are aware of) that this delay will trigger, and this ensures a fair customer support for those coming in February as well.

Based off previous years' data, we expect most groups to come and discuss the projects between January and February.

Project Organisation

The project has some fixed dates that are presented below:

The DCP Specifications are released	~8th class
The WG is formed, the WGL and WGC are elected	~8th class
Each group starts working	
The DCP is frozen (max)	21st class
The market fair happens	23rd class

The commitments are registered	23rd class
The exam happens	

Exams and Grading

This section describes the expected contributions for the exam, relevant exam details and the grading of the project.

List of Expected Contributions

This section presents the contributions we expect from each group, and from each member of each group.

Each group must develop:

- one Drone, that is sold to other groups
- one Drone Network Initialiser, that is used to initialise the Drone Network comprised of the Drones bought from other groups

Each member must develop one functioning Individual Contribution. Individual Contributions include:

- CLN: Client. A Client application, see the Project Component section.
- one Server, that is used locally for the Clients
- SC: Simulation Controller, see the Project Component section.

For each group, there must be at least one member who develop a CLN and at least one member that develop a UI.

Details for the Exam

A typical exam is the showcase of the functionality of your project and a discussion of its parts. During a typical exam day, groups will register for a timeslot on Telegram. Each time slot is 45 minutes, which includes:

- 5 minutes to enter, set up the showcase
- 30 minutes for showcase and questions
- 5 minutes to exit

- 5 minutes for me to write an evaluation

The examination is held in my office: **Povo2, 124**

Failure to show up to a booked slot, without proper and sound notice is ground for terrible punishment.

Submitting Your Code

3 days before the exam (typically the day of the written part of the exam), you will tell me that you are submitting.

On the day of the exam, you will checkout your repo on my machine (from my account) and run it from there. You are not allowed to push any code in the meantime.

You also have to submit your Drone code and any test that will let us indeed test that part of code, since you will not include it in your Constellation.

Grading

You will be graded according to these aspects:

- the code you write. Using advanced features that have not been presented in class is grounds for higher marks.
- how your group has interacted in the WC3 style groups
- how you maintain the code, and offered support for the teams that have bought your products
- bonuses, as described below.

Grading for the project is done in /30th and it amounts to 60% of the class total grade. Thus, it amounts to up to 18 points of the total grade.

The grading of the project is divided in:

- group score (18/30, so 10 points of the total)
- individual contribution score (12/30, so 8 points of the total)

Grading Bonuses

The WGLs and the WGC have extra work to do, so they score extra points depending on the quality of their work:

- the WGC gets a bonus of min 1 max 3 points
- each WGL gets a bonus of min 0.5 max 1.5 points
- the WGC can provide honorable mentions for members that contributed to the WG, they get min 0.5 max 1 points
- the 3 groups that sell the most Drones at the Faire get a bonus of 1.5 points

Ethics Considerations

Finally, reflect on what you are doing. The software you're developing may be running on actual Drones, whose communication may be used to save lives, but also to take lives. What do you think about the code you wrote. Are you fine with any of its usage?