Preliminary design document

Group Name

Team Information

pic	pic	pic	pic	pic	pic
First (Nickname) Last	First (Nickname) Last	First (Nickname) Last	First (Nickname) Last	First (Nickname) Last	First (Nickname) Last
Initials	Initials	Initials	Initials	Initials	Initials

Part 1: Mission and scope

Mission statement

Mission

What is the overarching mission of this team? You should write in one sentence.
What is the need that is being addressed? Do not focus on technical specifics yet.

Project scope

Are you going to rewrite Duckietown from scratch? Probably not. You need to decide what are the boundaries in which you want to move.

What is in scope

What do you consider **in** scope? (e.g. having a different calibration pattern)

What is out of scope

What do you consider **out** of scope? (e.g. hardware modifications)

Stakeholders

What other pieces of Duckietown interact with your piece?
List here the teams, and a possible contact person for each team.

Part 2: Definition of the problem

Problem statement

Time to define the particular problem that you choose to solve.

Suppose that we need to free our prince/princess from a dragon. So the mission is clear:

Mission = we must recover the prince/princess.

Now, are we going to battle the dragon, or use diplomacy?

If the first, then the problem statement becomes:

Problem statement = We need to slain a dragon.

Otherwise:

Problem statement = We need to convince the dragon to give us the prince/princess.

Suppose we choose to slain the dragon.

Assumptions

At this point, you might need to make some assumptions before proceeding.

Does the dragon breath fire?
What color is the dragon? Does the color matter?
How big is this dragon, exactly?

Approach

All right. We are going to kill the dragon. How? Are we going to battle the dragon? Are we trying to poison him? Are we going to hire an army of mercenaries to kill the dragon for us?

Functionality-resources trade-offs

The space of possible implementations / battle plans is infinite.
We need to understand what the trade-offs will be.

Functionality provided

How do you measure the functionality (what this module provides)?
What are the "metrics"?

> *Example:* numbers of dragons killed per hour

Note that this is already tricky. In fact, the above is not a good metric. Maybe we kill the dragon with an explosion, and also the prince/princess is killed. A better one might be:

> *Example:* numbers of royals freed per hour
> *Example:* probability of freeing a royal per attempt

It works better if you can choose the quantities so that functionality is something that you maximize to maximize. (so that you can "maximize performance", and "minimize resources").

Resources required / dependencies / costs

How do you measure the resources (what this module requires)?

> *Example:* numbers of knights to hire
> *Example: total salary for the mercenaries.*
> *Example:* liters of poison to buy.
> *Example:* average duration of the battle.

It works better if you think of these resources as something to minimize.

Performance measurement

How would you **measure** the performance/resources above? If you don't know how to measure it, it is not a good quantity to choose.

Example: we dress up Brian as a Dragon and see how long it takes to kill him.

Part 3: Preliminary design

Modules

Can we decompose the problem? Can you break up the solution in modules?
Note here we talk about logical modules, not the "physical" architecture (ROS nodes).

Interfaces

For each module, what is the input, and what is the output?
How is the data represented?

Note we are not talking about ROS messages vs services vs UDP vs TCP etc.

Preliminary plan of deliverables

What needs to be designed? What needs to be implemented? What already exists and needs to be revised?

Specifications

Do you need to revise the Duckietown specification?

Software modules

Here, be specific about the software:is it a ROS node, a Python library, a cloud service, a batch script?

Infrastructure modules

Some of the modules have been designated as "infrastructure".

Part 4: Project planning

First steps for the next phase

Now, make a plan for the next phase.

Data collection

What data do you need to collect?

Data annotation

Do you have data that needs to be annotated? What would the annotations be?

Relevant Duckietown resources to investigate

List here Duckietown packages, slides, previous projects that are relevant to your quest

Other relevant resources to investigate

List papers, open source code, software libraries, that could be relevant in your quest.

Risk analysis

What could go wrong?
How to mitigate the risks?