Everyone’s Log

21 Jan Week 1 meeting
23 Jan Doodle Poll http://doodle.com/poll/y7kfb72vh5wwrkv5
25 Jan Week 2 meeting

Process report

- The process report for your group should be between 4 and 6 pages long.
- Submission submit sdp process [filename] by **4pm on Friday 17th February**
- Report filename must be group-[g]-process.pdf where [g] is the group number.
- Structure (between 1.3 and 2 pages per point. Large illustrations and *references* will not count towards the page limits.)
    - up to 5 marks for organisational structure
    - up to 5 marks for communication mechanisms
    - up to 5 marks for risk assessment and contingency planning
- It should detail the approach that you have taken to
    - group communication,
    - task allocation,
    - progress tracking and
    - any milestones you have set yourselves (although these may just be the assessed friendly matches).
    - Early in the project you should make a plan detailing the work to be done and milestones at which progress will be judged. This may be documented in a Gantt chart. It is unlikely that you will remain faithful to the plan throughout the project, but without a plan it is difficult to judge progress. You may find it useful to make a revised version of your plan/gantt chart at key points in the project.
    - The report may also contain an assessment of the risks that you anticipate for the project, and contingency planning that you have done to guard against them.
    - You are encouraged to interact with the other group in your team as early as possible and if you have done so, you can also include information about this interaction in your process report.
    - Vesko: I have summarised Jane Hillston's videos on working in groups from SDP's Online lectures/workshops in my log. You can search 'Jane Hillston' to find it :)
- Contribution
    - What is your role? Do you share it with someone?
    - What was your plan in the beginning? (Describe as a process of consecutive steps because the plan can be used for a Gantt chart)
    - What could go wrong with your plan?
    - How would you have fixed each of the things that could go wrong?
    - What have you completed?
    - How do you track your progress?
    - How has your plan changed?
    - How did you communicate with the team?
    - How did you split the tasks with the other members of your team?
    - Have you communicated with the other group in the team? How?

Alex’s Log

I was assigned to working on the Vision aspect of the system with Vesko. The role in its entirety includes creating/adapting an old vision system that can parse a video feed and translate it into a digital world model that can be passed onto and used by the strategy controller to make decisions on what the robot will do.

Initial plan:

- Understand how to run the existing vision systems.
- Get familiar with how to use and calibrate vision systems.
- Choose vision system that can be used best with strategy.
- Understand how the vision system works(i.e. Does it use background subtraction, how does calibration work.)
- Understand how the vision system outputs data that can be used by strategy.
- Calibrate vision system for the pitch rooms.
- Discuss ideas for improving system.
- More thoroughly test system to find out where flaws are impacting robot performance.
-

One thing that could go wrong with our plan is that we could embark on using one vision system, find out that it isn’t as modular as we like, or that the strategy team would prefer to use a different system’s pre-existing code, and we would then have to switch systems and restart the process of understanding the system and how to calibrate and run it.

The vision team have a good understanding of how the vision system works, and some ideas of where the vision system could be potentially lacking, and so the next stage would be to either test the system to find out its most crucial flaws, or test new ideas to see if performance is improved.

Tracking progress ?

One change/modification of the plan is on how much time is spent on each stage, more time was spent on calibrating the vision system for the pitch rooms for the first match since making changes or improvements to the vision system would likely not work since we had limited time to test the system. Tasks such as finding flaws or exploring new ideas were put on hold until after the first match was complete.

Organising team meetings was done primarily through group instant messaging services such as slacks and facebook messenger. Communication with teammates about what individuals have learnt during the process was shared almost exclusively in person at meetings.

I primarily focused on studying and understanding how vision and strategy come together, and the approach the vision system takes in calibrating the vision, (CORRECT ME IF WRONG :]) whereas Vesko looked into OpenCV and and how to improve the system and use the system.

We have started communicating with the other group in our team to pool resources together to make a better vision system that we can both benefit from.

Background subtraction + normalisation not taken into account.

Andra’s Log

What is your role? Do you share it with someone?
My role as part of the hardware team (consisting of two people) was to aid with the robot construction (and reconstruction) process.

What was your plan in the beginning? (Describe as a process of consecutive steps because the plan can be used for a Gantt chart)
Our main goal during the first week was to build a robot that is both modular and sturdy, that has at least a functional kicker built into it. But to do so we had to:

- firstly decide which wheels to use, which meant collaborating with the software team as well, in order to pick the code base that they would like to work with
- decide on the motors to be used and whether they would need gearing or not
- considering the spatial layout of the Lego blocks so that the design would be modular
- beginning to build the robot
- fitting the main motors for the wheels
- understanding how the circuitry works, so that we can consider the layout on the “long-term”
- designing a kicker
- fitting the kicker and a motor for it
- fitting the circuitry
- checking that everything works
- implementing some basic communication commands for the Arduino (such as go, turn, stop, reverse, kick)

What could go wrong with your plan?
Apart from technical difficulties, such as cables or connections not working, other things that could have gone wrong have to do with:

- gearing not providing enough torque or speed; changing the gearing could imply a change in the robot layout as a whole
- not appreciating the spatial layout available correctly and end up building a robot that is too large or bulky; this would mean that the robot would most likely need to be disassembled and reassembled, which is rather time consuming
- different motors providing different powers to each wheel; the consequence of this being that the robot cannot move in a straight line if the motors are both operating at the same power
- the weight of the robot; if the robot is too heavy then more power is required in order to get it moving
- the required power in order to run the kicker - higher powers would drain the battery levels faster
- the strength of the kicker (or rather the lack of strength of the kicker)

How would you have fixed each of the things that could go wrong?

- different motors: for the first match we used different powers for each motor, so that it would run in a straight line when asked to. However this is not a long term fix, our plan is to implement an algorithm that would automatically either perform course correction or that at least calculates the required powers for each motors
- the weight of the robot can be countered by using smaller and lighter motors.
- the initial kicker required full power in order to kick, and the kick did not have the strength required, due to friction and the weight of the kicker. The way we dealt with this challenge was by coming up with a kicker design that requires less strength and takes up more space; different approaches would have been to gear the existing motor or use a more powerful one, however we realised that no matter the case we would need more space in order to fix these issues. We decided on a new design as this approach would not add to the weight of the robot as well, but instead makes it lighter.

What have you completed?
By the first match we have managed to achieve our goals, namely building the robot with a kicker and implementing the commands.

How do you track your progress?
Given that we have worked in day-long sessions, we would set goals at the beginning of the session for the particular day and attempt to finish them by the end of the day. Additionally, we found that spending the last 30 minutes of the work session thinking about what we should do next and when we should meet has given us a consistency in the sense of knowing what immediate tasks we have to achieve in order to keep progressing with the project.

How has your plan changed?
Our plan has evolved and we have added increasingly more tasks to it, as our understanding of the project increased. Although we have managed to complete our initial goals, the layout of the individual components did not allow for the addition of an extra motor for a grabber or any other sensors. Hence for the second match, we started considering grabber designs and managed to fit one to the robot. The robot has maintained its modularity but we are currently considering ways of making it lighter, so that it would be faster as well.

How did you communicate with the team?
Facebook, Slack, Asana, weekly meetings.

How did you split the tasks with the other members of your team?
We have adopted a work flow similar to pair programming, meaning that while one of us was assembling the robot or trying to fit a new part on it, the other one would have a complementary responsibility, such as:
- looking for parts that were needed at the moment
- thinking of a more efficient way of distributing the space available within the robot
- considering the modularity of the design

- looking into the circuitry and considering how we would fit more motors/sensors if they were needed
- considering what should be implemented during the next sessions, in order to predict any potential problems

Have you communicated with the other group in the team? How?
We have communicated with the other group in the team whilst being in the lab. We shared ideas about kickers, grabbers, motors and helped each other understand how the code base as well as how the circuitry works.

Anthony’s Log

What is your role? Do you share it with someone?
My role is being in charge of communication. So I necessarily have to communicate with both the Software and the Hardware teams.

What was your plan in the beginning? (Describe as a process of consecutive steps because the plan can be used for a Gantt chart)
My plan at the beginning was to port the previous years code (specifically Group 7s code as the TCP-esque communication protocol they used was better suited to the type of robot we were developing) over, and tweak it to work with our particular system. Then, once we had a working system, make optimisations and improvements to make the system more reliable.

- Read through previous years code
- Choose an appropriate codebase (reliant on design of robot)
- Set up RF communications and basic program (ping/pong)
- Create interface for packing messages into a transmittable format
- Create receiving interface on the Arduino for unpacking messages
- Connect received messages to their appropriate outputs (motors, etc)
- Add code for dealing with out of sequence messages or lost packets

What could go wrong with your plan?
- The RF communications may not work with the Arduino and SRF stick
- The SRF stick could cease functioning
- The Arduino board could stop responding to messages
- There could be functionality in their system that is unnecessary and redundant in ours

How would you have fixed each of the things that could go wrong?
- Flash and reconfigure the stick and board manually to ensure they're both set up correctly
- Try to reconfigure the stick and run a simpler program on the Arduino to test the response, and if all else fails (as it did) request a new SRF stick.
- Soft-reset the Arduino using the switch on the board, and try again. If that doesn't work we can remove the Arduino from all other connected devices (motors, sensors, etc) and try again (this stops possible serial interference). And absolute worst-case scenario, request another board.
- Think if there is a close analogue in our system to the redundant functionality, and if so reuse the code for that. If not, simply remove the functionality entirely.

What have you completed?

I've refactored the PC side of the code to be more readable and cleaner, and can send correctly formatted packets to the Arduino, and the Arduino responds as expected the majority of the time.

How do you track your progress?
I've been tracking my progress primarily through Asana, and also by writing README files both for use by others and also to consolidate my knowledge and ensure that I haven't missed any core functionality by mistake.

How has your plan changed?
I quickly discovered that the previous year's PC-side code was very hastily put together (due to them changing the format of their protocol at a late stage) and so I ended up writing the code from scratch using their code as a very rough template, rather than copy-pasting and tweaking constants. For the Arduino side I was able to pretty much use the old code unmodified, except of course for the motor and sensor pins which were changed as needed by the Hardware team.

How did you communicate with the team?
By talking in person to the Software and Hardware teams and making sure the code makes sense to them (as they will have to use it) and that I'm not making any silly errors.

How did you split the tasks with the other members of your team?
Since I'm the only person on Comms, I didn't communicate with other people on my 'team', but I quite often did group programming with the Hardware team to help identify bugs in their code and mine.

Have you communicated with the other group in the team? How?
Not especially as of yet. Just a few questions back and forth when dealing with similar errors due to us using the same groups codebase as our template.

Danial’s Log

What is your role?

I’m an in charge of strategy of the robot

Do you share it with someone?

I’m working with Filipe to make sure the robot would work on the matches

What was your plan in the beginning? (Describe as a process of consecutive steps because the plan can be used for a Gantt chart)

1. Since, we cannot immediately start working on the strategy as other parts are not finished, we decided that we help the other roles as much as we can.
2. Once they are done, we could start implementing basic strategies like go to the ball, defend the goal etc.
3. When we have time, we would start implementing for advanced strategies like passing or blocking shots.

What could go wrong with your plan?

If the other roles took too long to finish their parts, we will start later than scheduled thus leaving us with less time to complete our parts

How would you have fixed each of the things that could go wrong?

- We could implement strategies excluding any parts that require interaction with unfinished roles
- We start planning for the strategies on pen and paper first so that we could implement them immediately once able

What have you completed?

Currenntly, the robot is working fine and is able to move to the ball and position itself for defense. Now we are starting to move into the more advanced strategies while smoothing out previous strategies.

How do you track your progress?

Nil

How has your plan changed?

In addition to previous plans, we also have to take into consideration of the low quality camera provided to us. Other than the standard strategies, we also have to find out a way to correct the robot if the camera feed is bad or the world model isn’t working as intended.

How did you communicate with the team?

Asana (for objectives to complete)
Slack (for discussion with mentor)
Facebook (for meeting and general discussion)

How did you split the tasks with the other members of your team?

We split the strategy to offense (when we have the ball or could go for the ball) which is worked on by Filipe and defense (blocking, and positioning) which is worked on by me

Have you communicated with the other group in the team? How?

Currently, we only have basic discussion on strategies because we are still trying to get our robots to function properly. But we have considered the possibility of either focussing on defense or offense completely so that we have a more specialised function.

Filipe’s Log

- What is your role? Do you share it with someone?
    - I am part of the strategy team and one of the team managers of the team. I share my strategy duties with Danial and my team managerial duties with Andra.
- What was your plan in the beginning? (Describe as a process of consecutive steps because the plan can be used for a Gantt chart)
    - As part of the strategy team my goal was to “glue” all the parts of the system together. I started by trying to understand how the old codebase worked and from then on went ahead and try and figure what out what parts of the old system people were using and what parts they were discarding. As the first friendly match deadline came, my job was to ensure we had a basic strategy we can use.
    - As one of the team managers my goal was to simply make sure everyone can easily communicate with others and that everyone has something to do.
- What could go wrong with your plan?
    - At the start nothing much since most of it is learning what everyone else is doing. Strategy wise the worst it could happen is for the robot to be stuck due to a faulty algorithm. To test the strategy everything aspect of the robot would have to be ready, so if there was some problem while getting to the minimal working stage, it would be rough to have enough time to test the robot.
- How would you have fixed each of the things that could go wrong?
    - Well to begin with ensure that the algorithm wouldn’t go wrong through lots of testing. If something were to go wrong, there would be some mechanism to halt the robot.
- What have you completed?
    - I have a decent understanding at every aspect of the project. The robot is also able to do basic commands with some delays.
- How do you track your progress?
    - By communicating with other teams and seeing how well their robots are doing. Later on there will be goals that I will introduce that I will try to meet.
- How has your plan changed?
    - Not any major changes, the plan for strategy is quite adaptable depending on the other sections of the robot.
- How did you communicate with the team?
    - In person, through Messenger/Slack.
- How did you split the tasks with the other members of your team?
    - For the first friendly it was mostly everyone did what they deemed best. For the second task I am more focused on the Offensive strategy for the robot.
- Have you communicated with the other group in the team? How?
    - Yes (in person) to discuss progress and any issues we/they are having.

Lazar's Log

1. What is your role? Do you share it with someone?
    a. I'm part of the hardware team, made up of 2 people. We have the responsibility to build a functioning robot so that the other systems of the project may operate through it.
2. What was your plan in the beginning? (Describe as a process of consecutive steps
    a. Get some ideas from last year designs.
    b. Understand how the different motors and circuits connect with one another
    c. Learn about gearing and motor power
    d. Design a compact kicker
    e. Build a basic version of the robot to check for compatability issues. Setup the communications to work with the design.
    f. Compete in the first set of matches.
3. What could go wrong with your plan?
    a. There could be problems with the gearing of the different motors, since we need them to provide enough torque while still fitting inside the robot.
    b. The kicker design may not have enough power to kick far enough.
    c. We might not have enough room for all motors and circuits, while staying within the given constraints.
    d. Getting all the different moving parts strongly attached to the hull of the robot may prove difficult.
4. How would you have fixed each of the things that could go wrong?
    a. Through studying the different gear sizes and measuring the lengths of the insides of the robot and with some trial and error, we would be able to find the ideal gear sizes that would connect the motors to their respective mechanisms.
    b. Because we want the kicker to be compact, we don't have a lot choice. Since it can either be piston-like or a small rubber band kicker. If the initial piston-like kicker (http://i.imgur.com/bRilX3m.png) proves to not provide enough force to push the ball, we'll resort to a more powerful rubber-band kicker design. http://i.imgur.com/juq2Kls.png
    c. Because we'll be using 16x16 plates for the base, there might not be enough room for all motors, sensors and circuits inside its hull.
    d. Since the NXT motors have very few mounting holes, it is tough to make a configuration that can both contain the motors and have them well secured to the frame. (http://i.imgur.com/2xo6PD1.jpg). So we used the Lego Digital Designer to create a 3D model to determine how the motors will be secured.
5. What have you completed?
    a. We've made 3 versions of the overall design, the initial was mainly for measurements and the second was a bit more thought out. The third version is the one we used for the first set of matches and it resembles the second one a

lot, but instead of piston-like kicker it has a rubber band one instead. We also managed to fit all circuits inside the robot and most of the wires.

6. How do you track your progress?
   a. I have a general plan on what will have to be done and while working, I try to examine the problems that will arise and mentally add them to my personal plan on how to continue working. Usually at the end of the day we make a plan for the next time when we'll meet up.
7. How has your plan changed?
   a. From planning on having a piston kicker we replaced it with a rubber band one, which changed the internals of the final design substantially. We wanted to have a ball catcher, but in the end we removed it before the first matches as it was too big to fit in the provided box. After the matches we heavily changed the design of the robot so that the kicker would be better mounted and we also installed a ball catcher underneath.
8. How did you communicate with the team?
   a. Through messenger and slack to discuss ideas we've had during the day or to see when it would be convenient to meet up. We would also communicate verbally when in the lab.
9. How did you split the tasks with the other members of your team?
   a. Because the hardware team consists of 2 people, but we are working on one robot, someone would be working on building, while the other would be handing the needed parts for the construction. They would also give input on how to fit the different components together and give ideas on how to improve the design.
10. Have you communicated with the other group in the team? How?
    a. The hardware people of the groups are often in the lab at the same time, so we sometimes exchange ideas about mechanisms.

Vesko’s Log

- 20 Jan - Ran the vision system software from both implementations - visionwrapper.py for Craig the Robot (CtR)
- 22 Jan - Read Gui Features in OpenCV
- 24 Jan - Read ‘Preprocessing’ folder in CtR
    - 24 Jan - Read Open CV - Camera calibration
    - 24 Jan - Read OpenCV Background Subtraction
    - 24 Jan - Read Open CV Changing Colorspaces
        - --> Object Tracking/extracting a colored object/
        - (see later chapters for removing noise)
    - 24 Jan - Read Open CV Histograms - 2: Histogram Equalization
        - --> Is there a reason to use adaptive histogram equalisation in preprocessing
- 25 Jan - Read Equipment Guide's Fast and Inexpensive Color Image Segmentation for Interactive Robots
    - The software system is composed of four main parts:
        - a novel implementation of a threshold classifier,
        - a merging system to form regions through connected components,
        - a separation and sorting system that gathers various region features, and
        - a top down merging heuristic to approximate perceptual grouping.
    - A key to the efficiency of our approach is a new method for accomplishing color space thresholding that enables a pixel to be classified into one or more of up to 32 colors using only two logical AND operations.
    - *Much of the information in an RGB image varies along the intensity axis, which is roughly the bisecting ray of the three color axes.*
    - 2.2 Contains representation of YUV conversion
    - 2.3-2.5 Two stages for implementing connected regions
- 25 Jan - Read Equipment Guide's Fast and Accurate Vision-Based Pattern Detection and Identification
    - the document, supposedly, presents the derivation of an efficient and highly accurate detection algorithm for many different patch-based patterns
    - p.4 col.2 Using vectors to establish the plate's dots location
- 26 Jan - Watched Working in Groups or Teams by Jane Hillston
  The following text is an adapted summary of the video presentation
    - We are advised to use *agile methods,* which
        - recognise that the initial search for route to solution may be suboptimal
        - aims to reduce the (time) penalties incurred by changes of route to solution
        - involves continuously evolving requirements, frequent small releases and continuous testing
    - We are advised to use *pair programming*, which means that 2 people work in a group. One writes the code and the other thinks about how the code works and considers improving the code's strategy. Thus, while writing the code, its weaknesses are revealed and fixed.
    - The need and methods of communication will change during the project

  - Role structure (and their default activities, which can be modified)
    - Project manager
      - can be taken by another student during the project
      - is a point of contact within and outside the group
      - organises meetings
      - submits reports
      - could have the role of tracking everyone's progress
    - Scrum master (a.k.a. Team leader)
      - Responsible for resource availability
      - Booking the pitch
      - Asking Gary, the mentor, etc questions which the group has
    - Product owner
      - advises if a change should be made in the route to completion
      - follows the release/final product
      - tracks everyone's progress
      - takes decisions when they need to be made
    - Developer/Programmer
      - Responsible for modelling/developing
      - and documentation
  - Our communication channels
    - Slack - formal communication with mentor, Gary, etc
    - Messenger - informal fast-reaching communication
    - Log - tracking everyone's progress
    - Asana - online resource merging a calendar and a todo list that allows up to set up deadlines for our tasks

- 8 Feb - Watched Introduction to Project Management by Jane Hillston
  - Project Management - to have a clear idea of the tasks that need to be completed,
    - break down project into smaller identifiable tasks, which specify what we should have done, what we should be doing and what we plan to do for any moment in time for the duration of the project,
    - look at the needed and the available resources,
    - consider time and resource constraints (one task must be completed before another; one task may need specialist resources)
  - First step, make a plan, which can be later modified due to large deviations such as
    - Subtasks
    - Ordering constraints between subtasks
    - Estimated completion time of each subtask (base on guess, doubled guess, previous experience)
    - Timing constraints (e.g. for the first friendly match it is useful to turn and kick)
  - Second step,
    - scheduling,
    - monitoring, and
    - controlling the complex interdependencies between subtasks
  - Specific difficulties
    - Time until completion of a piece of code is unpredictable
    - The needed parts of the code may be unidentifiable

        - Integration between software and hardware will be a major task because the robot my not to live up to the code's expectations, and vice versa
    - When time until completion is longer than the available time (fixed with a network diagram)
        - Crashing - change the duration of a single task (e.g. complete task faster with pair-programming)
        - Fast tracking - identify sequential tasks that can be performed in parallel (e.g. add more people to team)
    - Resources - diagrams assume infinite resources
        - Team - parallel/sequential tasks depend directly on the number of teammates available
            - Availability depends on teammates' other commitments
        - Robot
        - Availability of pitches and machines
    - Diagrams
        - Work Breakdown Structure - a tree that progressively defines what we need to achieve
            - Start with the whole project
            - Keep breaking into smaller unit-of-work tasks - documentation, software, hardware
            - Documentation - process report, user guide, technical report, presentation
            - Etc.
        - Network Diagram -
            - Each node represents a subtask and contains an estimated completion time
            - Node 1 points to node 2 if subtask 1 needs to be completed before subtask 2
            - Ability to create parallel order with nodes with multiple branches
            - Critical path - longest path from beginning to end; minimum duration of completion of the project
            - Slack - subtasks which are not on the critical path and may be delayed after the completion of the previous tasks, without affecting the overall duration of the project
        - Gantt chart - calendar view with time on the horizontal axis and order on the vertical axis
            - Not necessarily showing constraints between tasks
            - Helps monitoring progress - Where am I in the calendar?
- 9 Feb - Process report contribution
    - What is your role? Do you share it with someone?
        - I work on the vision with Alex Wilson.
    - What was your plan in the beginning? (Describe as a process of consecutive steps because the plan can be used for a Gantt chart)
        - Run the code from last year's repos
        - Familiarise myself with how the Python implementation works
        - Test correctness of World model
        - Found out we need computer specific calibrations -> calibrate
        - Continuous tests

    - Expand my knowledge by reading the OpenCV documentation
    - Preprocess the feed to remove the noise from the video feed
- What could go wrong with your plan?
    - Other deadlines may lead to unavailability of teammates and , so I asked team to share deadlines in Asana. Teammate unavailability may lead to missing deadlines/milestones
    - The last year's repos may raise errors, or have errors that go unnoticed
    - Some implicit errors may go unnoticed
    - Libraries that I need for my implementation may not be installed
- How would you have fixed each of the things that could go wrong?
    - I asked team to share deadlines in Asana
    - Fix explicit errors and familiarise myself with the code (2.2) to find the implicit errors
    - Should get output of the World model that contains all coordinates of objects in the field and check the output for consistency
    - Try to install library myself with limited permissions. Ask if the library can be submitted with the implementation. If installing fails, ask Garry if it is possible that he installs the library.
- What have you completed?

I have performed/been performing all tasks. I have implemented a library for noise reduction and object recognition. However, I have not yet implemented the library in the vision system. I have not tested the correctness of the World model.

- How do you track your progress?

I have created list of the previously specified tasks (not in Asana but report can say otherwise) and followed that list

- How has your plan changed?

Create a vision system that uses object recognition compared to color recognition. The current vision system relies on the colors on the plates but the colors are heavily affected by the lighting conditions in the pitch room

- How did you communicate with the team?
    - Organising a weekly meeting where everyone meets with the mentor, asks recent questions and updates the mentor and one another of their progress.
    - Asana for documenting SDP deadlines, other subject's deadlines (shows periods of unavailability), and milestones. Asana is a web-based application that can visualise and create deadlines in a calendar view and time-ordered list.
    - Slack for official group communication that involves our group mentor - Craig Walton and Garry Ellard.
    - Facebook Messenger for immediate communication with single members of the team and the team group.
    - Google Docs Log, which contains different parts - a common one, a part for each teammate, a process report, etc. The common part describes our meetings (and decisions). Each teammate's part contains the log for their activity.
    - GitHub repository. Commit messages describe the changes everyone makes.
- How did you split the tasks with the other members of your team?
    - Alex and me started getting familiar with the code separately, but asked each other any time we would have trouble understanding parts of it. We also shared ideas about possible ways to improve on the repository.
- Have you communicated with the other group in the team? How?

    - Once a week, I asked them for a short verbal description of the current state of their robot and their plan for further development
    - In the beginning of week 3, I asked them if their vision team would like to work with ours to create a single powerful vision system. By adding more people to the production of a single system, we intend to perform more tasks in parallel.