# PROJECT INITIATION DOCUMENT

Balancing Robot Project

Abstract

This document informs on the project about the balancing and app controlled robot.

Konghon Choo and Eveline Ververgaert

# Document history

## Revision

| Version | Status | Date | Change |
|---|---|---|---|
| **0.1** | Concept | 21-07-2017 | Setup |
| **0.2** | First draft | 26-07-2017 | Chapters |
| **0.3** | First draft | 31-07-2017 | Remaining chapters |
| **1.0** | Review | 03-08-2017 | Reviewing and completion |
| | | | |

## Distribution

This document has been sent to:

| Version | Date | Name | Function |
|---|---|---|---|
| **1.0** | 04-08-2017 | Vaughn Malkin | Lecturer |
| | | | |

# Table of Content

# 1. Project Assignment

## 1.1 Goal

The goal is to demonstrate a project with a combination of multiple disciplines working together. This projects includes a remote controlled balancing robot. The robot can be controlled from a mobile application. View the requirements document for further specification.

## 1.2 Background

This project is a part of the course Second year Special Topic with the subject robotics. The product will be used as proof of our ability to work independently on a project. Through this project we are able to share our views and work process within a project with the Otago Polytechnic - IT School. This project will combine robotics and app development to demonstrate the use of multiple disciplines to help showcase that options are endless in the world of Technology.

## 1.3 Boundaries

This project will be confined within these boundaries.

Robot:

- The robot will not be protected from hazardous environments.
- The skeleton for the robot will be built with materials available to us.
- The material choice for the robot will depend on the machines that will be available to us for the fabrication process.
- We will not take bumps on the ground into account.
- The hardware will use an ESP Wi-Fi module to communicate with the app.
- The hardware will be written in C or C++ (this will be defined in a later stage).
- The robot will be able to stay in balance at all times.
- The robot will be controlled through Android app.
- The robot will only be able to avoid obstacles if time and budget allow this.

App:

- The app will be developed for Android devices only.
- The app will only be able to run on the latest version of Android.
- The app platform will be written in Kotlin to expand our knowledge beyond the standard use of Java.
- The app can make the robot move forward, backward and sideways.

## 1.4 Strategy

During the project the team will be using the Scrum methodology. This will enable to create a product that can be demonstrated in earlier stages. Each sprint will take 2 weeks. At the end of each sprint there could be a meeting with the lecturer.

## 1.5 End product

The end product described below is not final as more could be added. The minimal end product consists of the following:

Robot:

- The robot can keep itself balanced at all times.
- The robot can move forward, backward, to the left and right.
- The robot can receive commands from the app.

App:

- The app can communicate with the hardware.
- The app allows the user to press forward, backward or sideways.

See also the requirements document for more details.

# 2. Project Organization

This chapter informs about the members and communication methods.

## 2.1 Team members

| Name | E-mail | Role | Availability (days per week) |
|---|---|---|---|
| **Konghon Choo** | CHOOK2@student.op.ac.nz | Team member | 1* |
| **Eveline Ververgaert** | VERVEH1@student.op.ac.nz | Team member | 1* |
| **Vaughn Malkin** | Vaughn.Malkin@op.ac.nz | Lecturer | By appointment |

*Every Monday the developer team will work on this project.

**Team member**

This is a member of the development team. The development team is responsible for the development of the project.

**Lecturer**

A teacher, known as lecturer, is assigned for guidance during the project. The lecturer is not a part of the development team. The lecturer will assess the project and insure the project and the development team stays on track.

## 2.2 Communication

**Meetings**

| Kind of consultation | Frequency | Present |
|---|---|---|
| **Planning** | Start of sprint | Developer team (and potentially lecturer) |
| **Review** | End of every sprint | Developer team and lecturer |

**Other communication means**

The members will communicate mainly through the meetings during the project days. May other communication resource be necessary, the team will use Telegram (with texting as backup). Furthermore, communications with the lecturer will happen via Email or face-to-face when visiting the office.

## 2.3 Decision form

During the project the development team will make decisions based upon facts and references. Therefore, any major decision will be supplemented with an argument provided in a document. When an approval is necessary it will be discussed with the lecturer.

# 3. Activities and schedule

This chapter explains which activities are planned to occur and what the overall time schedule is. The schedule is a estimation and could therefore change during this semester.

## 3.1 Layout and approach

During the project the team members will work with the Agile methodology. This will be tracked through a Trello board. The team will work with Source Control tool GitHub for code and Google Drive for documentation. During each sprint the developer team shall continuously integrate tested and reviewed code into the main branch (see also chapters 4 Quality assurance and 5 Testing).

## 3.2 Overall schedule

Each sprint consists of 2 weeks. These goals potentially change over time and will be specified during each planning sessions.

| Sprint | Potential goal(s) | Dates |
|---|---|---|
| **1** | • Project Initiation Document<br>• Requirements<br>• Hardware research | 24-07 till 04-08 |
| **2** | • Research<br>• UML Document (incl. protocol document)<br>• Ordering hardware | 07-08 till 18-08 |
| **3** | • Start on Balancing algorithm<br>• App design document | 21-08 till 01-09 |
| **4** | • App development<br>• Assemble hardware | 04-09 till 15-09 |
| **5** | • Communications between App and Robot<br>• Robot driving movements | 18-09 till 29-09 |
| **6** | Term break | 02-10 till 13-10 |
| **7** | Buffer sprint (possibly for adding extra features or to finish previous tasks) | 16-10 till 27-10 |
| **8** | • Finishing project<br>• Demo preparations | 30-10 till 10-11 |
| **9** | Buffer sprint | 13-11 till 24-11 |

## 3.3 Definition of Done

**Documentation**

This will be reviewed and needs to be accepted by the member who is not the author. The reviewer will check for grammar as well the content. The document needs to be transparent in its conclusion, decisions, references and arguments.

**Code**

The code will be peer reviewed by the other member. During the review there will be checked for readability, simple code guidelines, test case failures/successes and the code must build without warnings and errors.

**Hardware**

When code has been built for the hardware. The developer gives the other team member a demo. This will do as proof of concept for the current hardware component.

# 4. Quality assurance

This chapter contains information on how the team will assure the quality of the products.

## 4.1 Reviews

The following table shows which moment or product/document is in need of a review, what kind of review and who the assessor is.

| Product / moment | Type |
|---|---|
| **PID** | Productreview |
| **Requirements document** | Productreview |
| **UML document** | Productreview |
| **Sprint review (/demo)** | Phase review |
| **Potential shippable product** | Productreview |

## 4.2 Approval

The following products/documents are in need of approval from the lecturer:

- PID
- Requirements document

# 5. Testing

This chapter informs about how we are going to test during the project.

## 5.1 Responsibility

Testing ensure the requirements have been met and that there are no defects, defects and bugs are a separate thing.

## 5.2 Test approach/strategy

This project will be tested with Acceptance testing and Unit testing. During the acceptance testing the robot (in cooperation with the app) will be evaluated in its environment on the provided features. For this kind of testing a document will be provide to go through all the steps. Running Unit tests will help eliminate bugs and errors in code.

## 5.3 Test environment

Unit tests run on a laptop, the preferred unit test tool has yet to be determined. For Acceptance testing the robot and the app are necessary with an even ground (preferably no carpet). If the feature obstacle avoidance has been added the environment will also include obstacles.

## 5.4 Necessities

To execute the tests the following is required:

- Test cases or the document
- Laptop with the software (for Unit testing)
- Robot
- Mobile phone with app
- Required test environment (mostly consists of an even ground)

Possibly some features may need specific extra’s for testing, which will be specified in the Acceptance Document.

# 6. Financials

This chapter will describe the needed budget and cost of this project, also the assumptions that has been made to complete this chapter.

## 6.1 Budget

### 6.1.1 Hourly spending

8 hours during the week, most of the hours will be made on Monday the day reserved for the project. There are 2 extra hours reserved for overtime if needed.

This project will have a runtime of 17 weeks excluding the holidays. With an ending date of 24-11-17.

| Hours | Description | Total |
|---|---|---|
| **8 x 17 x 2** | Normal working hours for the project team of 2 | 272 |
| **2 x 17 x 2** | Extra working hours for the project team of 2 | 68 |
| **Maximal hours spent on project** | | 340 |

### 6.1.2 Materials

See preliminary hardware research document for more specifications for some components.

- Arduino + USB cable
- Motors
- Tires
- Cables
- Battery + plug for Arduino
- Wi-Fi shield
- Sensors (to be determined)
- Android phone with Android 4.x or higher
- Misc. parts (e.g. materials for the skeleton)

### 6.1.3 Miscellaneous

- Working space
  A working space will be provided by the school.
- Hardware tools
  All the tools needed to assemble the robot are available at the school.
- Software tools
  The software applications needed to develop the code will be primarily self-provided, in case there is a need for an software pack that is not available through the normal ways.

## 6.2 Assumptions

To make an early estimate for the budget we have to make a few assumptions regarding the available hardware and tools. We assume that there will be a working space provided by school to work on the project.

# 7. Risks and dependencies

This chapter provides a risk assessment and informs on any dependencies.

## 7.1 Dependencies

During the project internet is required for research and the online scrum tool Trello as well as Google Drive for the documents. During implementation the necessary programs need to be available, among which unit tests and software developing tools.

## 7.2 Risk and alternative activities

**Risk**: Internet failure

- Solution When possible try to fix internet or use books and people for research. Or go to another location with internet availability.
- Alternative Continue on parts of the project which do not require internet.
- Impact Low
- Possibility Small possibility this will happen

**Risk**: Lecturer not available

- Solution Mail the issue/question
- Alternative Continue on other work
- Impact Medium
- Possibility Medium

**Risk**: Laptop/computer (with project on it) not available

- Solution Make do with a different computer if possible
- Alternative Work on parts that does not require a computer
- Impact High
- Possibility Low

**Risk**: Unable to login into GitHub, Google Driver or other important online platforms

- Solution Research if problem can be solved, than solve it (e.g. password forgotten)
- Alternative Get any necessary information from other team member
- Impact Low
- Possibility Medium

**Risk**: Hardware unavailable

- Solution Research what is wrong (too late delivery or broken equipment), create a solution for it.
- Alternative Work on other parts of the project
- Impact High
- Possibility Medium

**Risk**: Team member(s) unavailable

- Solution    Make sure this won't affect the planning too much. If it does have a high effect on the project, talk to the lecturer
- Alternative    If possible work at home or at other times
- Impact    High
- Possibility    Medium

**Risk**: deviate from scheme

- Solution    Talk to lecturer if not possible to work on other occasions
- Alternative    Keep working on working
- Impact    Medium
- Possibility    Medium

**Risk**: Newly added code has bad effects on project

- Solution    Quarantine that version, get all errors and bugs out, then add it again into the main project.
- Alternative    Ask for help
- Impact    Medium
- Possibility    Medium

**Risk**: Workspace unavailable

- Solution    Work temporarily elsewhere
- Alternative    Work at home
- Impact    When working with hardware: High
- Possibility    If assigned to standard work space by school: Low