Robotic Sensorimotor System Testing Platform (RSSTP)

Final Report

Version 0.1

🦾

|  |  |
| --- | --- |
| Name | Student number |
| Filip Georgiev | 2208085 |
| Khalil Chakal |  |
| Mourad Boulfrad | 2207592 |
| Roshan Fernando |  |
| Shabbir Hasan | 2208301 |

Executive abstract

Table of Contents

[Executive abstract 2](#_Toc117610113)

[Acronyms and Definitions 4](#_Toc117610114)

[1 Introduction 4](#_Toc117610115)

[2 Customer 4](#_Toc117610116)

[3 Project description 5](#_Toc117610117)

[3.1 Project Background 5](#_Toc117610118)

[3.2 Project Goal 5](#_Toc117610119)

[3.3 Project requirements 5](#_Toc117610120)

[4 Realised requirements 6](#_Toc117610121)

[4.1 Internal logic 6](#_Toc117610122)

[4.2 External logic 6](#_Toc117610123)

[5 Used software development method 6](#_Toc117610124)

[6 Testing 7](#_Toc117610125)

[6.1 Process 7](#_Toc117610126)

[6.2 Unit tests 7](#_Toc117610127)

[6.3 Acceptance testing 7](#_Toc117610128)

[7 Project timeline 8](#_Toc117610129)

[7.1 Milestones 8](#_Toc117610130)

[7.2 Phases 8](#_Toc117610131)

[7.2.1 Phase I 8](#_Toc117610132)

[7.2.2 Phase II 8](#_Toc117610133)

[7.2.3 Phase III 8](#_Toc117610134)

[7.2.4 Phase III 8](#_Toc117610135)

[7.2.5 Phase IV 9](#_Toc117610136)

[7.2.6 Phase V 9](#_Toc117610137)

[7.2.7 Phase VI 9](#_Toc117610138)

[8 Feedback to the course organizers 10](#_Toc117610139)

[9 References 10](#_Toc117610140)

[10 Appendix 11](#_Toc117610141)

[10.1 Original project pitch 11](#_Toc117610142)

[10.2 Software structure 12](#_Toc117610143)

Acronyms and Definitions

|  |  |
| --- | --- |
| Agent | The test subject (e.g., a robot arm) |
| MVP | Minimum Viable Product |
| PE | Perception Engineering |
| RSSTP | Robotic Sensorimotor Software Testing Platform |
| TDD | Test-Driven Development |
| UBICOMP | Center for Ubiquitous Computing |
| VR | Virtual Reality |
|  |  |

# Introduction

How does a person learn to traverse their surroundings? How does an animal? When all senses of a living being (e.g., sight, hearing, smell, etc.) are combined, they create a mesh of information, which can be used by said being to obtain a feeling about their environment and consequentially navigate through it. However, what happens when an agent is stripped of most of their senses and is only left with minimal sensory feedback? What is the best algorithm, which can be used to learn one’s environment when there is little to no information to utilize?

In the Perception Engineering (PE) research group of The Center for Ubiquitous Computing (UBICOMP) research unit in the University of Oulu, possible answers for the last question stated above will be explored. It is planned for research to be done, so that optimal algorithms, which e.g., teach a robotic hand to move in an environment with different obstacles, can be found. Before this can be done, however, a testing platform, in which these algorithms can be evaluated and compared needs to be created.

In this document, the design and creation of such a platform, named Robotic Sensorimotor System Testing Platform (RSSTP), are outlined. In the following chapters more information about the customer, the project itself and the realised requirements will be given. This will be followed by an explanation of the software development and testing methods used during the project. Afterwards the project timeline will be discussed, and the document will be concluded with the feedback given to the Software Project course organizers. Any additional (and necessary information) regarding this project will be found in the [*Appendix*](#_Appendix) chapter.

# Customer

The customer for this project is the UBICOMP research unit and more specifically the PE group, which focuses on research related to Virtual Reality (VR) and robotics-related problems, similar to the one described in this document, as for example sensing, sensor fusion, planning, learning and control [1].

The people who will act as representatives for the customer and in so as supervisors and contact persons for this project are Dr. Vadim Weinstein ([vadim.weinstein@oulu.fi](mailto:vadim.weinstein@oulu.fi)) and Dr. Kale Timperi ([kalle.timperi@oulu.fi](mailto:kalle.timperi@oulu.fi)).

# Project description

## Project Background

In UBICOMP research is being done in the field of robotics, so as to determine the best possible way to teach a robot to learn about its surroundings with minimal sensory feedback (e.g., without the use of visual sensors). The nature of this task is theoretical, so there are no hard constraints set on the type of robot or type of environment, in which learning will happen. Furthermore, the task is seen as an unsupervised problem and hence algorithms which will try to create a model in real-time based on the minimal data received will be created. However, before these algorithms can be investigated and created a testing platform in which learning can be simulated and its results can be observed and evaluated needs to be designed and implemented. The RSSTP project was established based on these requirements and its original description can be seen in [Appendix 1](#_Original_project_pitch).

**TODO: more information about why the research is being done / motivation behind it**

## Project Goal

Originally, the RSSTP project was planned to last one year. However, as the course in which it is currently being implemented (Software project) is only one semester, the goal has been set to be time-appropriate in the form of a minimum viable product (MVP). This MVP must include the design and creation of the testing platform itself and the functionality of simulating how a given algorithm affects the learning capabilities of one robotic arm, which has two joints and minimal sensory feedback. Additional project requirements are set in the section below.

## Project requirements

* The MVP platform must inform the user of the results, but this does not need to be done with a visual simulation. The evaluation of the results will be for the user to interpret, although in the future this might become part of the testing platform.
* The chosen technological stack must have the ability to interface with Python.
* The environment in which learning will happen must contain obstacles, which will hinder the agent’s movement.
* The agent should be allowed to receive minimal information as for example whether it reached its destination or not, but it should not store exact data about its surroundings (e.g., obstacle locations).
* The supplied software structure seen in [Appendix 2](#_Software_structure) should be used to create the platform.

# Realised requirements

 Algorithm gets sensory data(0/1) each transition,  
 but sensory feedback (1) only when it reaches a set point (green light)

The software structure doc

Python chosen as soft language.

## Internal logic

## External logic

# Used software development method

The SCRUM methodology, which is part of the Agile philosophy was used in this project. Even though it is required by the Software Project course, another reason for using Scrum was that it would keep the supervisors acquainted with the work done by the students. This was essential for this project due to its theoretical nature. By utilizing SCRUM any misunderstandings of the requirements of the project could be dealt with in a timely manner.

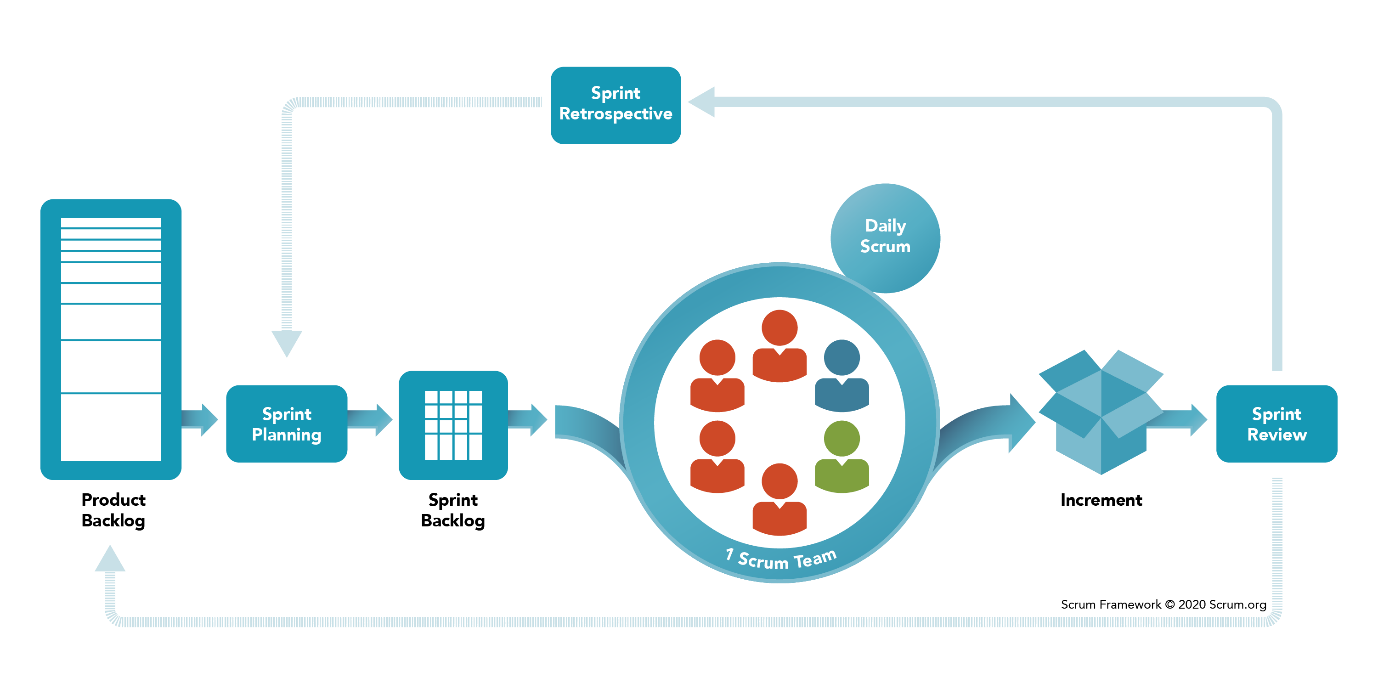


Figure 1 SCRUM Framework. Source: [SCRUM website](https://www.scrum.org/resources/what-is-scrum)

To ensure that all parties are kept up to date weekly hourly meetings on Thursday at 09:00 were scheduled. Furthermore, biweekly sprints were created, which would correspond to the phases outlined in the [Project Timeline](#_Project_timeline) section. The sprint demo, retrospective and planning would be done at the end of each sprint in the same Thursday meeting, which would in this case be scheduled for two hours. An additional meeting would be held by the students only each Tuesday at 18:00, in which they could update each other on their progress and help with issues that have arisen.

The online software tool “Trello” was used to create the SCRUM board and keep track of tasks. There would be six general cards, in which tasks could be put.

1. Backlog card, where all the future tasks would be stored.
2. To Do card, which contains all the tasks that need to be done in the current sprint.
3. Blocking card, which contains tasks that can not be done at the current time, as there is a different task that needs to be done first or some other blocker.
4. Doing card, which contains the tasks currently being done by the students.
5. Review card, which contains the finished tasks that need to be reviewed by the students or supervisors, before they can be considered done.
6. Done card, which contains all the tasks which have been done throughout the sprint. Multiple done cards were used throughout the different sprints.

**TODO: Add picture at the end of the project**

# Testing

## Process

The Test-Driven Development (TDD) process was used during the project. As testing was required by the course and the time allocated to complete the project was limited it seemed like a logical conclusion to invert the normal development cycle and start with tests first. This would mean that the Unit tests for the software would be created first, followed with the software which would have to adhere to the tests. Not only would this ensure that unit tests are created in time, but it would also provide additional benefits. For example, it would force the students to understand their tasks before they have implemented them, as they would need to know what a certain function (or unit) would need to do to be considered successful before they could create a test.

## Unit tests

As python was chosen for the development language, the “pytest” Python testing tool was used to create the testing suite and following tests.

**TODO: …**

## Acceptance testing

Acceptance tests were also done in the form of full platform algorithm runs. This would mean that after the software was developed, algorithms provided by the supervisors would be run through the testing suite and hence its functionality would be tested.

**TODO: overview at the end of the project**

# Project timeline

The timeline of the RSSTP project starts on the 21st of September. On this date, the first meeting with the clients was had. The timeline ends on the 13th of December, when the Final Presentation was had. The project was conducted in 2022.

## Milestones

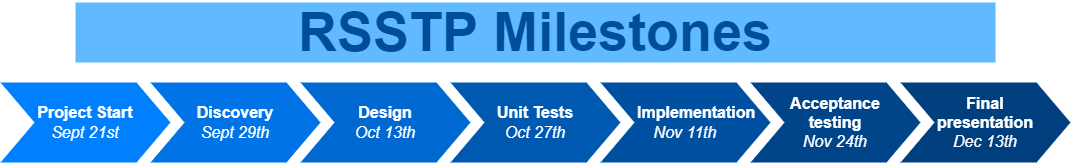


Figure 2 RSSTP Milestones

The Milestones for this project can be seen in Figure 1. Below, an explanation for each one will be given with the respective phases in which it is reached.

## Phases

Each phase of the project contains the work done to reach the next milestone in the timeline. The duration, description and the milestone reached upon completion of the phase are listed below for each phase.

### Phase I

**Duration:** 2022/09/07 – 2022/09/21

**Milestone:** *Project start*

**Description:** Managing to find a project and getting approval from the teacher/TAs was the first task for the project. The milestone was completed after the first client meeting.

### Phase II

**Duration:** 2022/09/21 – 2022/09/29

**Milestone:** *Discovery*

**Description:** The second milestone was completed once the project goal was better understood. This was done by setting up multiple meetings with the clients and discussing what the project entailed.

### Phase III

**Duration:** 2022/09/29 – 2022/10/13

**Milestone:** *Design*

**Description:** Completing the “Design” milestone was achieved when the team designed the software platform with the help of block, class and state diagrams.

### Phase III

**Duration:** 2022/10/13 – 2022/10/27

**Milestone:** *Unit Tests*

**Description:** As the project would be implemented following the TDD method, the unit tests for each software component had to be created. Once this was done, the “Unit Tests” milestone was reached.

### Phase IV

**Duration:** 2022/10/27 – 2022/11/11

**Milestone:** *Implementation*

**Description:** Based on the design and unit tests, the software components of the RSTTP were implemented. The “Implementation” milestone was completed once this was done.

### Phase V

**Duration:** 2022/11/11 – 2022/11/24

**Milestone:** *Acceptance testing*

**Description:** To ensure that the testing platform works as expected, the entire software suite was tested by using it as intended and running different example algorithm through its pipeline. Completion of this milestone meant that the software worked as intended in the requirements.

### Phase VI

**Duration:** 2022/11/24 – 2022/12/13

**Milestone:** *Final presentation*

**Description:** The final step of the project was to create a final presentation on the project. During this phase any leftover tasks/bugs were also resolved and the milestone was finished once the presentation was given.

# Feedback to the course organizers

# References

|  |  |
| --- | --- |
| [1] | "Perception Engineering (PE)," [Online]. Available: https://ubicomp.oulu.fi/research/pe/. [Accessed 24 October 2022]. |

# Appendix

## Original project pitch

Text

Description automatically generated

## Software structure

Text, letter

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated