**Project Proposal**

**Title:**

**Topological Navigation Editor**

**(ToNE)**

By

Laxwell Nyama

MComp Computer Science

University of Lincoln

2015/2016

Supervised by

Dr Marc Hanheide

**Table of Contents**

**Page**

1. Abstract
2. Introduction
   1. Overview
   2. Relevance
   3. Aim
   4. Objectives
   5. Literature review
3. Project Plan
   1. Overview
   2. Gantt Chart
   3. Pert diagram
4. Methodology
   1. Overview
   2. XP Programming
5. Scope and Constraints
   1. Overview
   2. Scope
   3. Constraints
   4. Risk Matrix
6. Resources
   1. Overview
   2. Development Platform
   3. Hardware requirements
   4. Software requirements
7. References

**Abstract**

This interactive design project will seek to develop an innovative graphical user interface for robot control and manipulation. In this proposal I propose to design a topological map navigation editor (GUI) to be used by human to control mobile robots that is usable and accessible.

**Introduction**

Overview

“Mobile robotics and robot navigation is a growing area of scientific research. Without navigation the creation of self-propelled, household machines, guard robots, or planet surveyors is beyond imagination”. (Richárd Szabó 2004). An easy and effective way of controlling such robots will vastly increase their applicability.

In recent years we have seen exponential growth in mobile robotics and robot use, however, the control and use of these robots requires a high level of technical knowledge. The way these robots are currently controlled is too complex and technical for an average person, hence the need to design and implement a graphical user interface that is simple, effective and easy to use for people with less technical ability.

It is however, also important that robot controlling softwares be written in several high level computer languages like C/C++ or Java, python for the simple reason of maintenance and compatibility. Above all, usability fundamental and draws from the field of Human Computer Interaction.

**Relevance**

This project is part of STRANDS with the aim of developing a real world topological navigation system editor (GUI) by improving on the existing ones to advance the robot navigation capabilities, usability and user interfaces. Autonomous Mobile Robotics is a growing scientific field and it is related to one of the modules I will be undertaking.

**Aims and objectives**

**Aim**: The aim of the project is to develop a Topological navigation editor user interface for robot systems

**Objectives**

* Establish requirements
* To identify, investigate and evaluate current systems (Topological navigation editors)
* To identify, investigate and evaluate tools and techniques
* Create and evaluate a prototype
* To design and develop the topological navigation editor user interface
* Test, analyse and evaluate the topological navigation editor
* Demonstrate the effectiveness of the artefact
* Artefact to be delivered on time

The navigation editor is design to be simple for ease of control of the robots by people of all walks of like with little to no technical ability. The editor will include features like drag and drop, the user will be able to drag and drop a node, edge and also edit these through a properties. However it is not just limited to novice users but also expert users.

**Establish requirements**

According to (Preece et al, 2011) “Whatever the initial situation and whatever the aim of the project, the users’ needs, requirements, aspirations, and expectations have to be discussed, refined, clarified and probably rescoped”. This will be the first objective to be met so I can establish what it is that is required. With requirements gathering I aim to understand as much as possible about the users, their activities and the context of that activity, so the system under development can support them in achieving their goals and it also produces a set of stable requirements that form a sound basis to start design.

My requirements gathering process comprises of sub activities, which will inform and refine one another iteratively through out the product life cycle, this is due to the fact that requirements evolve and develop as the stakeholders interact with the design through prototypes and see what is possible and how certain facilities can help them.

For this project I will gather both “functional requirements: which say what the system has to do and non-functional requirements: which says what constraints there are on the system and its development. Requirements gathering techniques like interviews, focus groups and direct observation on users will be conducted to collect enough information.

**To identify, investigate and evaluate current systems (Topological navigation editors)**

This first objective is aimed at finding out any existing Topological navigation systems. The investigation of pre-existing systems not only does it help in discovering the industry standard techniques, hardware and software requirements but will also aid in the design of my artefact. The background research and literature review will shed some light on what is needed to be done and in some measure how to do it effectively and efficiently.

The identification, investigation and evaluation of current systems will be conducted through different medium such as the internet, academic literature, IT journal, books, publications and newspapers for current news and interacting with real life robots. Case studies from past and present projects relating to my project will also be explored to aid me in the design and implementation of the navigation editor “for example, Green *et al* (2009)” describes how they adapted a large open source piece of software, called Bugzilla, to design and develop a ‘Problem Reporting, Analysis and Corrective Action System’ for NASA’s next generation vehicle to replace the shuttle.” (Preece et al, 2011).

**To identify, investigate and evaluate tools and techniques**

The aim of this objective is to research into the tools and techniques I am going to employ, this objective is fundamental as it facilitate on learning on how to use the tools and employ techniques I will be utilising to develop the topological navigation editor graphical user interface (GUI). Here I will be investigating ROS and STRANDS robot systems, which are some of the tools I will be using to achieve the goal of this project.

“Since the advent of graphical user interfaces in the 1980s many tools have been developed to support creative thinking, design sketching, simulation, brainstorming, mind-mapping, most aimed at visual interfaces.” (Julie and Andrew, 2003) and the challenge is to choose the best tool for the job. MongoDb and PyMongo just to mention a few are also other tools I will conduct some research on as they integral to the project development.

The tools I will use range from low-level i.e. requiring a lot of programming to high levels tools. The lowest level being the *graphical libraries* that provide hardware independence, *User interface toolkits* to structure an interface as a tree of interactive objects and widgets, whereas user interface builders provide an interactive application to create and edit those widget trees. *Application frameworks* build on toolkits and UI builders to facilitate creation of typical functions such as cut/copy/ paste, undo and help.” (Julie and Andrew, 2003). This objective will help me understand how they can be used for my project.

**Conceptual design/ Create and evaluate a prototype**

After conducting all the necessary research as stipulated by the first two objectives, here will draw up a Low-Fidelity prototype. “A Low-fidelity prototype also known as Lo-fi is one that does not look very much like the final product, it uses materials that are very different from the intended final version, such as paper and cardboard rather than electronic screens and metal.” (Preece et al, 2011).

I will employ two lo-fi prototyping techniques that is storyboarding and paper prototyping. The former will aid with showing how a user might progress through a task using the product under development while the latter will facilitate to show the design interface.

The reasons I will use lo-fi prototype is for the benefits it offers such as being cheap and quick to produce. Lo-fi prototypes are also simple and quick to modify, hence they support the exploration of alternative designs and ideas. This is particularly important during my conceptual design.

Paper prototyping involves use of paper, index cards, cardboard etc. to build up a paper representation of the system. One example of this was the “cardboard computer” (Ehn and Kyng, 1991).

**To design and develop the topological navigation editor user interface**

A user-centred approach will be applied to design the interface. “In 1985, Gould and Lewis (1985) laid down three principles they believed would lead to a ‘useful and easy to use system’:

1. Early focus on users and tasks.
2. Empirical measurement
3. Iterative design” (Preece *et al*, 2011).

These three principles are now accepted as the basis for a user-centred approach (e.g. see Mao *et al,* 2005)

Since I am aiming to design a Topological navigation editor for robot systems, in this objective I will look to employ both types of design. “Broadly speaking there are two types of designs: conceptual and physical. The former is concerned with developing a conceptual model that captures what the product will do and how it will behave, while the latter is concerned with details of the design such as screen and menu structures, icons and graphics.” (Preece et al, 2011).

The design process will itself be iterative, that means when problems are found they are fixed and more testing carried out and so on.

A user study will be conducted to come up with a conceptual design, which will be modified after a few iterations to birth the actual design to be implemented. User centred design

**Test, analyse and evaluate the topological navigation editor interface**

“Nowadays users expect much more than just a usable system, they also look for a pleasing and engaging experience. This means that it is important to carry out an evaluation.” (Preece et al, 2011).

Testing is fundamental and I am obligated to test the artefact to ascertain functionality and make sure it is fit for the purpose and also its usability (Usability testing). Once the application is completed, I will conduct a functionality test and a usability test with a focus group of users. An application with high functionality is not enough if its user interface makes it difficult for most of the users to be able to make use of its functionalities. “Computer systems with poor user interfaces can have a financial cost.” (Stone D *et al*, 2005).

The testing of the GUI will be conducted using the following criteria:

* Minimal requirements
* Interface usability
* Performance and effectiveness

My evaluation will be guided by the DECIDE framework. “The DECIDE framework provides a check list to help plan evaluation studies and to remind you about the issues you need to think about. It has the following six terms:

* **D**etermine the goals
* **E**xplore the questions
* **C**hoose the evaluation method
* **I**dentify the practical issues
* **D**ecide how to deal with ethical issues
* **E**valuate, analyze, interpret and present the data.” (Preece *et al*, 2011).

This framework also allows for iterative movement back and fourth the list after taking a first pass through each of them.

**Demonstrate the usability and effectiveness of the artefact**

After accomplishing all the objectives above, the artefact will be ready for final demonstration to the client. The demonstration will involve deploying the application on a simulation. This is to show the stakeholder the how the artefact functions, give them a walk on the application so may they may have some idea on how they will use it.

**Artefact to be delivered on time**

There are going to be some difficulties and constraints along the way as is the case with most software projects, however I still aim to deliver the artefact in the time stipulated, this can be achieved through effective planning, time management and adhering to the plan and time schedules. The use of different tools like the Gantt chart for the project will also aid in project management.

Delivering software is hard, and delivering quality software in time is even harder.

**Literature review**

Overview

In this section I present the background research undertaken in order to understand the problem domain and allow me to design the system to meet the user requirements.

The point of literature review as identified by Borg and Gall (1989, cited by Saunders et al. 1997: 39) is among other things:

* To refine research questions and objectives
* To highlight research possibilities that have either been explicitly; identified by other authors or have possibly been overlooked in the past.
* To avoid repeating the works of others
* To identify search methods and strategies

This is achieved with references to past and current literature in the field of robotics, user interfaces and Human computer interaction.

Background research

* Simulation agents

The use of simulation is well presented in Explorations in evolutionary robotics (Cliff et al., 1993; Bongard, 2013), but it is a tool that must be used with caution; the benefits in terms of speed and cost have to be balanced against issues arising from the validity of the simulation - the so called ‘reality gap’ (Jakobi et al., 1995). I will be using a simulation in a project so a bit of research in this field will help with overall undertaking of the project.

* Graphical User-Interfaces development environments

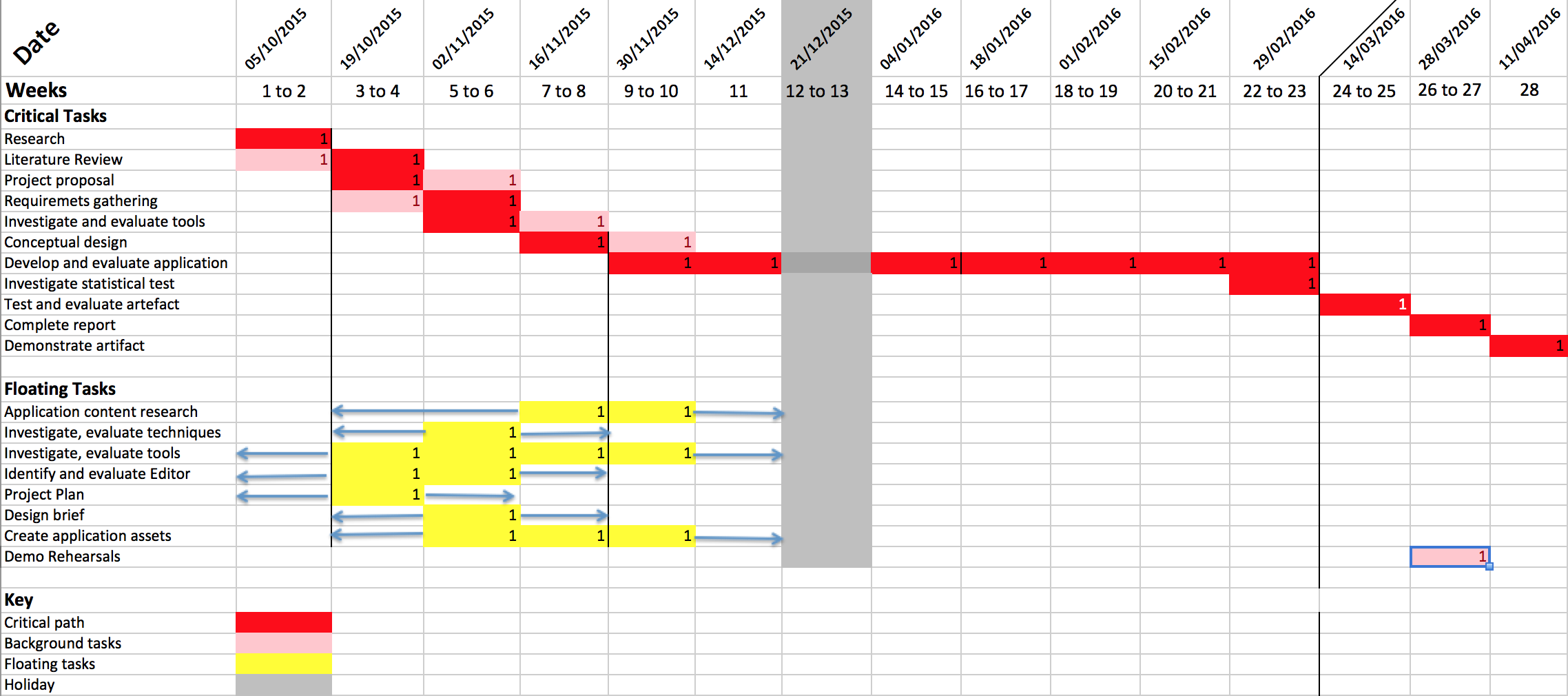
There is a need to be well versed with the development environment I will be using; however researching on other development environments will enrich my knowledge and techniques.

**Project plan**

Overview

In this section of the proposal I will present some of the tools I have commissioned to help me in organising, time management and ensuring that the project stays on track to facilitate on time delivery. I have employed the Gantt chart for the project plan and a Pert diagram for the flow of activities.

Gant chart



In the chart above the critical path is in red showing the scheduling of main activities and their precedence, which is further illustrated by the pert diagram below.

The pert diagram below further shows stages where the iterations will happen be it on the conceptual design, actual development or the testing and evaluation. These iterations are shown in loop like format.

Pert diagram

Research

Gather user requirements

Create Project Plan

User study

Conceptual Design/ Lo-FI Prototyping

Evaluate prototype

Develop widgets

Develop GUI

Implement functions

Testing and evaluating artefact

Report and Demonstration

**Methodology**

I will use one of the agile software development methodologies known as Extreme Programming (XP). Extreme Programming (XP) is a software engineering methodology, the most prominent of several agile software development methodologies. Like other agile methodologies, Extreme Programming differs from traditional methodologies primarily in placing a higher value on adaptability than on predictability. “XP also focuses on excellent application of programming techniques, clear communication, feedback, courage and respect, which allows us to accomplish things we previously could not even imagine”(Kent Beck and Cynthia Andres, 2005). These are the reasons why I have chosen to use this methodology.

This is a type of agile software development involves developing and delivering functionality in fragments or in incremental stages. The XP methodology allows for iterations and changes are encouraged and are seen as natural and desirable happenings that occur during the course of the projects development cycle. This is approach is beneficial and appropriate for this particular project where the exact requirements and potential difficulties are not always foreseeable due to the experimental nature of the project.

The methodology involves splitting up the different aspects of the projects functionality steps giving the developer a greater degree of control over the projects development process.

“XP builds on best practises such as unit testing, pair programming, and refactoring” (Erich Gamma, 1999). However, since this is an individual project, the practise of pair programming will not be practised, as this requires two people to undertake. “Extreme programming uses an object oriented approach as its preferred development paradigm and encompasses a set of rules and practices that occur within the context of four framework activities: planning, design, coding and testing.” (Roger S. Pressman, 2010)

**Scope and constraints**

Scope

The scope of this project will be limited due to the nature of the project and a number of constraints some of which are discussed below. A narrower scope means shorter processes and fewer resources required, however the scope will determine the process, but it also must fit project goals.

Constraints

* Ethical issues – this may arise if the user participating in the test decides to quit before testing is completed. Participants are allowed to exercise their right to withdraw from the test at any point in time as they wish, this will however have huge implications on the project completion and delivery.
* Expertise – This project is a real challenge to me as I am entering into uncharted territory. My knowledge and expertise is virtually none existent. This will be the biggest constraint in this project which might have impact on project delivery as I will consumed a lot of time getting to know the tools techniques.
* Time – The fact that I have to undertake a lot of research and learning on how to use the tools in a short space of time will impact on the project schedule.

Risks

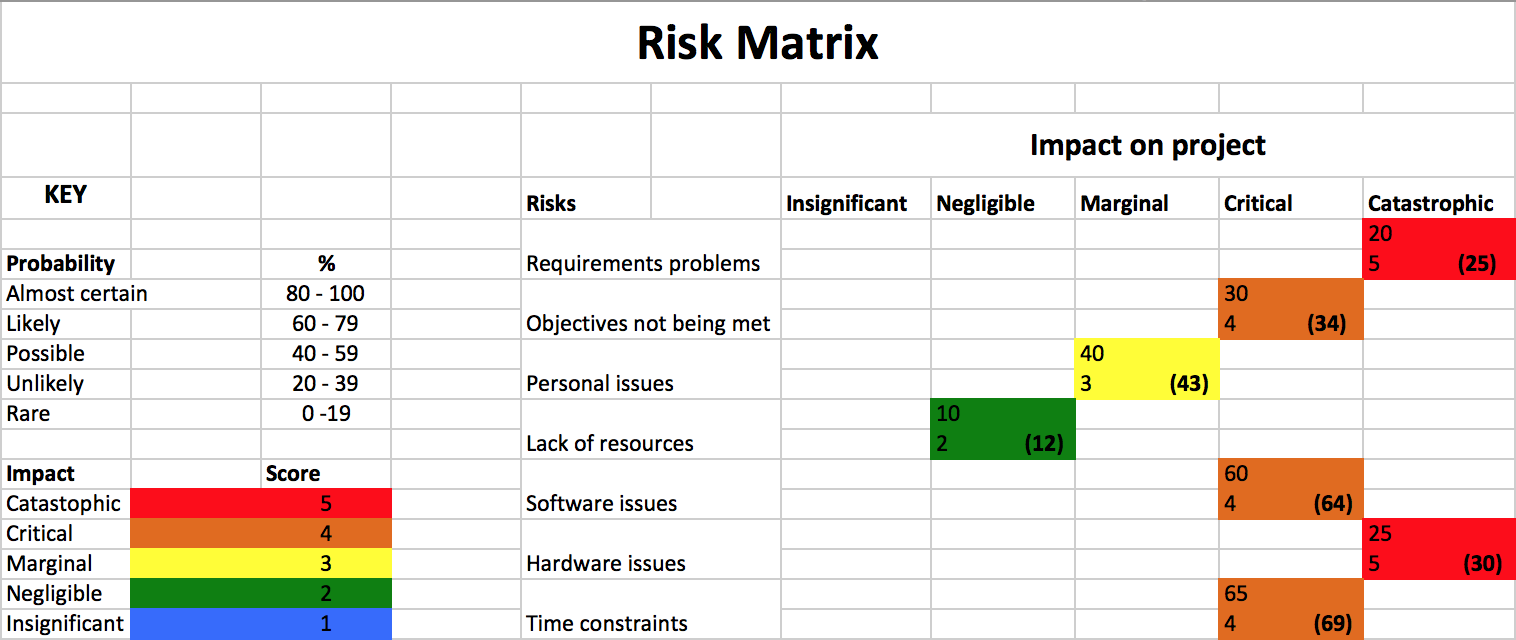
These come in different shape and forms; here I will highlight two major risks categories that may impact my project:

*“Project risks* – threaten the project plan, if a risk manifest it is likely that the project schedule will slip. Project risks identify schedule, personnel, resources, stakeholder and requirements problems.

*Technical risks* – threaten the quality and timeliness of the software to be produced, if they manifest, implementation may become difficult. Technical risks identify potential design, implementation, interface, verification and maintenance problems.” (Roger S. Pressman, 2010).

Risk Matrix

The risk matrix below shows the probability of a risk manifesting and the impact it will have on the whole project.



**Resources**

**Development platform**

I will be using the Ubuntu operating system, which runs on the Linux Kernel (Ubuntu Linux 14.04LTS 64bit (trusty)). The reason being the STRANDS project is already underway and using that platform, so for compatibility purposes I have to run the same system.

**Hardware requirements**

* Computer running on Linux kernel
* 40GB hard drive memory
* 1GB working memory
* Mobile robot that provides odomerty data and is equipped with a horizontally mounted, fixed, laser range finder (to use slam\_gmapping).

**Software requirements**

* Ubuntu Operating system (trusty)
* ROS software
* STRANDS software
* Python
* Rviz
* MongoDb
* PyMongo
* PyQt4

Conclusion

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**References**

Jenny Preece, Yvonne Rogers, Helen Sharp (2011) *Interaction Design beyond human-computer interaction 3rd Edition*. Wiley.

Julie A. Jacko, Andrew Sears (2003) *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications.* Lawrence Erlbaum Associates. London.

Roger S. Pressman, (2010) *Software Engineering: A Practitioner’s Approach Seventh Edition.* McGraw-Hill.

D, Stone., C. Jarret., M. Woodroffe., S. Minocha (2005). *User Interface* *and Evaluation*. Morgan Kaufmann Publishers.

Cliff, D., Husbands, P. and Harvey, I. (1993) ‘*Explorations in evolutionary robotics’, Adaptive Behavior* 2(1), 73–110.

Ehn, P. and Kyng, M (1991) *Cardboard Computers: Mocking it up or hands-on the future*. New jersey.

Kent Beck with Cynthia Andres (2005) *Extreme Programming Explained, Embrace Change.* Pearson Education, Inc.

Elizabeth Parlemo

[www.livesciences.com/51834-exoskeleton-helps-clumpsy-robots.html](http://www.livesciences.com/51834-exoskeleton-helps-clumpsy-robots.html) [accessed on 08 Oct 2015]

Richárd Szabó Department of General Computer Science Eötvös Loránd University, Budapest, Hungary. Topological navigation of simulated robots using occupancy grid.

<http://www.jataka.hu/rics/ars2004/ars2004.pdf> [Accessed 15 Oct 2015]

<https://docs.mongodb.org/getting-started/python/>

[Accessed 15 Oct 2015]

F. de la Paz Lopez and J. R. Alvarez-Sanchez Dept. of Artificial Intelligence – UNED (Spain)

<http://link.springer.com/chapter/10.1007%2F3-540-45723-2_55#page-1>