This is my title

by

My Name

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Bachelor of computer science (SE programme)

in the

Faculty of Computer Science
of the
October University for Modern Sciences and Arts (MSA), EGYPT

Graduation Projects advisor:
Dr. ABC, Dr XYZ

(July 2023)

Abstract

Gestures have a lot of potentials as a natural interaction method that can enrich the interaction between humans and ubiquitous environments. The lack of ordinary devices in ubiquitous environments like the keyboard and mouse make researchers work on utilizing hand gestures for interaction. However, hand gestures could change according to the situation it is performed.

Acknowledgments

I am heartily thankful to Professor

Contents

List of Tables

List of Figures

Chapter 1

Introduction

You will find many topics here to put from the Proposal

1.1 Introduction guide lines

A good introduction should tell the reader what the project is about without assuming special knowledge and without introducing any specific material that might obscure the overview. It should anticipate and combine main points described in more detail in the rest of the project report. Also, importantly, it should enthuse the reader about the project, to encourage them to read the whole report. Normally it should include such things as:

- the aim(s) or goal(s) of the project
- the intended audience or "beneficiaries" of the work done and the scope of the project"
- the approach used in carrying out the project
- assumptions on which the work is based; and
- a broad summary of important outcomes.

Chapter 2

Background

2.1 General Guidelines

The purpose of the Background section is to provide the typical reader with information that they cannot be expected to know, but which they will need to know in order to fully understand and appreciate the rest of the report It should explain why the project is addressing the problem described in the report, indicate an awareness of other work relevant to this problem and show clearly that the problem has not been solved by anyone else. This section may describe such things as:

- The wider context of the project;
- The problem that has been identified
- Likely stakeholders within the problem area
- Any theory associated with the problem area
- Any constraints on the approach to be adopted
- Existing solutions relevant to the problem area, and why these are unsuitable or Insufficient in this particular case Methods and tools that your solution may be based on or use to solve the problem and so on.

The wider context of the project includes such things as its non-computing aspects. So, for example, if you are producing software or any other products, including business recommendations, for a specific organisation then you should describe aspects of that organisations business that are relevant to the project.

2.2 Example

This is Example for writing some related work you can extract them from your Proposal and SRS

The study of gesture recognition with a presentation viewer application was shown in [1]. They show an active region for starting and ending gesture interaction. Also, they point out that gestures can be useful in crowded or noisy situations, such as in a stock exchange or manufacturing environment. Head and hand gestures have been used for limited interactions as demonstrated in by Keates et al. [1]. They discussed the problem of learning gestures and showed the importance of customization. Kurze et al. [2] presented penalization of multi-modal applications as a design approach. They focus on implicit and explicit customization of systems according to a user's preferences. Kawsar et al. [2] presented customizing the proactive applications preferences in a ubiquitous environment. They present customization in many levels of artifact, action, interaction, and timing preferences.

Chapter 3

Specification - (SRS)

3.1 General Guide Lines

If you are SE student put your SRS Here

Some students can choose between those methods based on project type

1. (Application oriented) Selection of Approach
2. (Comparing Algorithm) Description of Algorithms
3. (Novel Algorithm) Problem statement

3.2 SRS Guidelines

A specification should tell the reader what the software system is required to do.

Describing what a software system does (specification) and how it does so (design) effectively usually means describing it from more than one viewpoint. Each viewpoint will convey some information about the system that other viewpoints omit.

Possible viewpoints might be:

- The business model the software supports;
- The user interface;
- The dynamic behaviour of the system;
- How data flows through the system;
- What data types are implemented in the system;
- What algorithms are implemented in the system;

- The static architecture of the system, i.e. how the code is partitioned into modules, etc.

A common approach is to first define the user or business requirements, then describe the static architecture, identify modules and groups of closely connected modules, and then to apply other views to each of these groups. Fine details, specifically details of code, should be left out. We strongly recommend that you make extensive use of diagrams, such as entity-relationship diagrams, UML diagrams, state charts, or other pictorial techniques

Chapter 4

Design

4.1 Design Guidelines

If your are SE Student just put your SDD

Some Students can choose between

1. (Application oriented) Application of Selected Approach
2. (Comparing Algorithm) Implementation
3. (Novel Algorithm) Alternative Designs and Final Algorithm

The design then gives the top-level details of how the software system meets the requirement. It will also identify constraints on the software solution, that are important in guiding decision making throughout the development process.

As well as describing the system, it is important that you justify its design, for example, by discussing the implications of constraints on your solution and different design choices, and then giving reasons for making the choices you did. Typically these implications will relate to the aims of the project and to aspects of it discussed in the Background section. The design of the system will almost certainly have evolved while you were developing it. Obviously you should describe its final state but often there are good reasons for describing intermediate states, too; for example, if you want to discuss the details of the design method used or to highlight learning that you later refer to in the Reflection section. If you do this, take special care to make sure the reader does not get confused between different stages of the design.

If you are not designing a system, but testing a hypothesis for a more scientifically oriented project , specification and design sections may not be required in quite the

The specification instead becomes a description of the problem and what is required of a solution. The design becomes a description of your approach to solving the problem and your suggested solutions.

For instance, if you are designing an algorithm to solve a particular problem you would have a problem statement section and then a section describing one or more suggested algorithms to solve the problem.

Chapter 5

Implementation

5.1 General Guidelines

Students can choose between

1. Illustrate your implementation parts
2. (Application oriented) "Deliverables" from Selected Approach
3. (Comparing Algorithm) Experiment Design

The Implementation section is similar to the Specification and Design section in that it describes the system, but it does so at a finer level of detail, down to the code level. This section is about the realisation of the concepts and ideas developed earlier. It can also describe any problems that may have arisen during implementation and how you dealt with them. Do not attempt to describe all the code in the system, and do not include large pieces of code in this section. Complete source code should be provided separately. Instead pick out and describe just the pieces of code which, for example:

- Are especially critical to the operation of the system;
- You feel might be of particular interest to the reader for some reason;
- Illustrate a non-standard or innovative way of implementing an algorithm, data structure, etc..
- You should also mention any unforeseen problems you encountered when implementing the system and how and to what extent you overcame them. Common problems are:

- Difficulties involving existing software, because of, e.g., its complexity, lack of documentation; lack of suitable supporting software;

A seemingly disproportionate amount of project time can be taken up in dealing with such problems. The Implementation section gives you the opportunity to show where that time has gone

Chapter 6

Results and Evaluation

6.1 General Rules

In this section you should describe to what extent you achieved your goals. You should describe how you demonstrated that the system works as intended (or not, as the case may be). Include comprehensible summaries of the results of all critical tests that were carried out. You might not have had the time to carry out any full rigorous tests – you may not even got as far as producing a testable system. However, you should try to indicate how confident you are about whatever you have produced, and also suggest what tests would be required to gain further confidence. This is also the place to describe the reasoning behind the tests to evaluate your results, what tests to execute, what the results show and why to execute these tests. It may also contain a discussion of how you are designing your experiments to verify the hypothesis of a more scientifically oriented project. E.g., describe how you compare the performance of your algorithm to other algorithms to indicate better performance and why this is a sound approach. Then summarise the results of the tests or experiments.

You must also critically evaluate your results in the light of these tests, describing its strengths and weaknesses. Ideas for improving it can be carried over into the Future Work section. Remember: no project is perfect, and even a project that has failed to deliver what was intended can achieve a good pass mark, if it is clear that you have learned from the mistakes and difficulties. This section also gives you an opportunity to present a critical appraisal of the project as a whole. This could include, for example, whether the methodology you have chosen and the programming language used were appropriate

Chapter 7

Conclusions and Future work

7.1 General Rules

The Conclusions section should be a summary of the aims of project and a restatement of its main results, i.e. what has been learnt and what it has achieved. An effective set of conclusions should not introduce new material. Instead it should briefly draw out, summarise, combine and reiterate the main points that have been made in the body of the project report and present opinions based on them. The Conclusions section marks the end of the project report proper. Be honest and objective in your conclusions.

7.2 Future Work

7.3 General Rules

It is quite likely that by the end of your project you will not have achieved all that you planned at the start; and in any case, your ideas will have grown during the course of the project beyond what you could hope to do within the available time. The Future Work section is for expressing your unrealised ideas. It is a way of recording that I have thought about this, and it is also a way of stating what you would like to have done if only you had not run out of time1. A good Future Work section should provide a starting point for someone else to continue the work which you have begun.

Appendix A

Collected materials from ....

Please be sure to put here any materials you have collected during your graduation project.

Bibliography

[1] M. Rehm, N. Bee, and E. Andre, "Wave like an egyptian: accelerometer based gesture recognition for culture specific interactions," in *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, ser. BCS-HCI '08. Swinton, UK, UK: British Computer Society, 2008, pp. 13–22.

[2] S. T. Ayman Atia and J. Tanaka, "Smart gesture sticker: Smart hand gestures profiles for daily objects interaction," *Computer and Information Science, ACIS International Conference on*, vol. 0, pp. 482–487, 2010.