



University  
of Glasgow | School of  
Computing Science

Honours Individual Project Dissertation

# DYNAMIC MAKING PROJECTS

**Claire Williamson**

March 24, 2023

## Abstract

- tangible learning = good - tangible learning = often inaccessible for people who would benefit from it - forays into tangible learning with people with IDs/cognitive impairments require a lot of components (people who know about cognitive impairments, people who know about tangibles, ) - creating a tool that can be used to lessen these components so these groups could benefit from tangible learning - sentence describing said tool - sentence describing the results Tangible learning has been found to be an efficient way of improving students skills Every abstract follows a similar pattern. Motivate; set aims; describe work; explain results.

“XYZ is bad. This project investigated ABC to determine if it was better. ABC used XXX and YYY to implement ZZZ. This is particularly interesting as XXX and YYY have never been used together. It was found that ABC was 20% better than XYZ, though it caused rabies in half of subjects.”

# Education Use Consent

I hereby grant my permission for this project to be stored, distributed and shown to other University of Glasgow students and staff for educational purposes. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Signature: Claire Williamson    Date: 24 March 2023

# Contents

# 1 | Introduction

## 1.1 Motivation

- lots of disabled people
- tangible learning: has many benefits, especially for people that 'traditional' learning isn't well suited for (be sure to clarify that the blame here lies on 'traditional' learning not the person)
  - which includes many disabled people
- however there's lots of barriers for tangible learning for disabled people
  - list them

## 1.2 Aim

- lower the above barriers so disabled people can participate (more easily) in Making and thus gain the benefits from it
- do this by creating a (???) (platform? website? tool?) that allows them to get Making project tutorials that are/in a way that is accessible to them
- and that also allows them to improve their skills over a customised course of projects (<- generalise this more)

## 2 | Background

What did other people do, and how is it relevant to what you want to do?

### 2.1 Guidance

- Don't give a laundry list of references.
- Tie everything you say to your problem.
- Present an argument.
- Think critically; weigh up the contribution of the background and put it in context.
- **Don't write a tutorial**; provide background and cite references for further information.

## 3 | Analysis/Requirements

What is the problem that you want to solve, and how did you arrive at it?

### 3.1 Guidance

Make it clear how you derived the constrained form of your problem via a clear and logical process.

## 4 | Design

How is this problem to be approached, without reference to specific implementation details?

### 4.1 Guidance

Design should cover the abstract design in such a way that someone else might be able to do what you did, but with a different language or library or tool.



## 5 | Implementation

What did you do to implement this idea, and what technical achievements did you make?

### 5.1 Guidance

You can't talk about everything. Cover the high level first, then cover important, relevant or impressive details.

## 6 | Evaluation

How good is your solution? How well did you solve the general problem, and what evidence do you have to support that?

### 6.1 Guidance

- Ask specific questions that address the general problem.
- Answer them with precise evidence (graphs, numbers, statistical analysis, qualitative analysis).
- Be fair and be scientific.
- The key thing is to show that you know how to evaluate your work, not that your work is the most amazing product ever.

### 6.2 Evidence

Make sure you present your evidence well. Use appropriate visualisations, reporting techniques and statistical analysis, as appropriate.

## 7 | Conclusion

Summarise the whole project for a lazy reader who didn't read the rest (e.g. a prize-awarding committee).

### 7.1 Guidance

- Summarise briefly and fairly.
- You should be addressing the general problem you introduced in the Introduction.
- Include summary of concrete results (“the new compiler ran 2x faster”)
- Indicate what future work could be done, but remember: **you won't get credit for things you haven't done.**

# A | Appendices

Typical inclusions in the appendices are:

- Copies of ethics approvals (required if obtained)
- Copies of questionnaires etc. used to gather data from subjects.
- Extensive tables or figures that are too bulky to fit in the main body of the report, particularly ones that are repetitive and summarised in the body.
- Outline of the source code (e.g. directory structure), or other architecture documentation like class diagrams.
- User manuals, and any guides to starting/running the software.

**Don't include your source code in the appendices.** It will be submitted separately.