

**WHY IS THERE NO PEANUT BUTTER IN THE FRIDGE  
(I DON'T WANT TO WALK TO THE SHOPS)**

Submitted in part fulfilment for the degree of  
B.Sc. (HONS) IN SCIENCE IN COMPUTING  
TU Dublin  
Dublin, Ireland

by  
**Charlotte Smith**

supervisor  
**Dr. Matt Smith**



# Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Degree of B.Sc. (Hons) in Science in Computing in TU Dublin, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: \_\_\_\_\_ Dated: \_\_\_\_/\_\_\_\_/\_\_\_\_



# Abstract

GUIDE TO STUDENTS: An abstract is a brief overview of the project. Ideally it should fit onto a single page.

This project tackled the problem of creating a computer-based tool to model and manipulate music. The problem to be solved was how to create a computer tool that is easy to use but that also allows for non-trivial musical actions to be made on the computer. In this report a description is presented of the specification and analysis of the problem, the review of relevant research conducted, and the life cycle of the system that was developed to solve the problem.

Key features of the developed system, called “Pitch Circles”, including its basis in a respected music theory, its development in an object-oriented programming language – Java 2, and the evaluation of the system from an HCI (Human Computer Interaction) perspective, in addition to more traditional functional testing.

The result is a simple, but powerful tool for basic musical tasks, that is easy to learn while elegantly modelling the symmetric and asymmetric features of the Western Diatonic music system. With the Pitch Circles tool novices users can identify the relationship between triadic chords and different regions (scales), and use simple rules for movement around the tonal circles to derive and apply fundamental musical principles including the chromatic and diatonic circles of fifths.



# Acknowledgements

Write your “Thank You”s here!





# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Introduction</b>	<b>ix</b>
0.1 Simple figure . . . . .	ix
0.2 Other great books . . . . .	x
0.3 Example of a Table . . . . .	x
0.4 Free Online TableConvert.com . . . . .	xi
<b>Second chapter</b>	<b>xiii</b>
0.5 More results . . . . .	xiii
<b>General Discussion</b>	<b>xv</b>
<b>Software required for Symfony development</b>	<b>xix</b>
.1 Don't confuse different software tools . . . . .	xix
.2 Software tools . . . . .	xx
.3 Test software by creating a new Symfony 4 project . . . . .	xx
<b>Questionnaire</b>	<b>xxi</b>
<b>WebApplication.php</b>	<b>xxiii</b>
<b>List of References</b>	<b>xxv</b>



# Introduction

## 0.1 Simple figure

Here is a simple figure.



Figure 1: Github's famous 'Octocat'!

You can cite the figure in the text (to match the automatic figure numbering). See Figure 2 for a dog version of an octopus!



Figure 2: Lokikaze's Octodog.

(Thanks to Lokikaze for publishing this image on [Deviant Art](#))

Here is a bulleted list using minus-signs before each item:

- apples
- pears

- bananas

Here is a numbered list using 1. before each item:

1. blue
2. green
3. yellow
4. gold

here is some code inline using backticks:. A good loop is `foreach`

here are some lines of code using indentation

```
foreach($cars as $car){
    print $car->getRegistration();
}
```

To get better syntax highlighting you can triple backtick code blocks and state the language:

```
foreach($cars as $car){
    print $car->getRegistration();
}
```

You need to insert empty lines at the end of each section.....

## 0.2 Other great books

There are some great books out here, e.g. see (Adams 1979).

If you want to learn about Unity game programming try a cookbook (Smith 2018).

Another sentence typed here.

## 0.3 Example of a Table

Below see Table 1, for a list tech things, and in the Markdown source how to create such a table.

Table 1: Description of some computing techy things.

Item	Description	hrs	rate
Hugo Installation	Static HTML site generator	2	180
Database Integration	Testing of data structures	2	180
SCORM integration	Connecting to Moodle	6	180
Design and CSS	Design of web page front-end	4	180
Testing and Tuning	Ensuring that components are functional	3	180

Item	Description	hrs	rate
Subtotals			
Discount			
Total (ex. GST)			
Total (inc. GST)			

See Figure 3 for how to create tables using MarkDown and a little bit of LaTeX.

**## Example of a Table**

Below see Table `\ref{hugh_et_al}` for a l... cable.

`<!-- ***** start of table ***** -->`

`Table: (Describing of some computing techy things) \label{hugh_et_al}`

Item	Description	hrs	rate
Hugo Installation	Static HTML site generator	2	180
Database Integration	Testing of data structures	2	180
SCORM integration	Connecting to Moodle	6	180
Design and Coding	Design of web page front-end	4	180
Testing and Tuning	Ensuring that components are functional	3	180
Subtotals			
Discount			
Total (ex. GST)			
Total (inc. GST)			

`<!--`

**Annotations:**

- Cite table numbers just like figure numbers With `\ref{...}` and `\label{...}`
- caption
- Simple tables are simple in MarkDown  
Minus signs for row separator  
Vertical bars to separate items in each row

Figure 3: MarkDown source code for tables.

## 0.4 Free Online TableConvert.com

If you don't want to edit tables manually in MarkDown, a great free online tool that will do this for you is TableConvert. See Figure 4.

- <https://tableconvert.com/?import=example>

TableConvert Online Table (4x5)

Name	Shoe size			
joe	10			
fred	9			
joanne	5			

Markdown IDEA CSV JSON XML YAML S

AsciiDoc reStructuredText PHP Ruby ASP (As) Action

```

| Name | Shoe size | | | |
|-----|-----|---|---|
| joe | 10 | | | |
| fred | 9 | | | |
| joanne | 5 | | | |
| | | | | |
  
```

Figure 4: Online table Markdown (and LaTeX) creator.

# Second chapter

## 0.5 More results

Add your second chapter here

Here is another citation about Smith's (2018) book on Unity.





# General Discussion

Write your awesome discussion here!



# Appendices



# Software required for Symfony development

## .1 Don't confuse different software tools

Please do not confuse the following:

- Git and Github
- PHP and PHPStorm

Here is a short description of each:

- Git: A version control system - can run locally or on networked computer. There are several website that support Git projects, including:
  - Github (perhaps the most well known)
  - Gitlab
  - Bitbucket
  - you can also create and run your own Git web server ...
- Github: A commercial (but free for students!) cloud service for storing and working with projects using the Git version control system
- PHP: A computer programming language, maintained by an international Open Source community and published at **php.net**
- PHPStorm: A great (and free for student!) IDE - Interactive Development Enviroment. I.e. a really clever text editor created just for working with PHP projects. PHPStorm is one of the professional software tools offered by the **Jetbrains** company.

So in summary, Git and PHP are open source core software. Github and PHPStorm are commercial (but free for students!) tools that support development using Git and PHP.

## .2 Software tools

Ensure you have the following setup for developing Symfony software on your local machine

- PHP 7.2.5 or later (free, open source)
- Composer (up-to-date with `composer self-update`)(free, open source - a PHP program!)
- PHPStorm (with educational free account if you're a student!) - or some other editor of your choice
- MySQL Workbench (Community Edition free)
- Git (free, open source)

See Appendix ?? for checking, and if necessary, installing PHP on your computer. See Appendix 0.5 for details about other software needed for working with PHP projects.

## .3 Test software by creating a new Symfony 4 project

Test your software by using PHP and Composer to create a new Symfony 4 project. We'll follow the steps at the [Symfony setup](#) web page.

Follow the steps in Appendix ??.

# Questionnaire

We asked participants the following:

- Age group (only adults aged 18+ were consulted)
  - under 25 / 25 - 35 / 35 - 50 / over 50
- Level of computer expertise/usage
  - computer novice / use a few times a month / use daily





# WebApplication.php

Code listing for /src/WebApplication.php. This class is the decision-making logic for the application to decide how to respond to the HTTP Request message received by the web application.

```
<?php
namespace Tudublin;

class WebApplication
{
    private $mainController;

    public function __construct()
    {
        $this->mainController = new MainController();
    }

    public function run()
    {
        $action = filter_input(INPUT_GET, 'action');

        switch($action) {
            case 'jokes':
                $this->mainController->jokes();
                break;

            case 'index':
            default:
                $this->mainController->index();
        }
    }
}
```



# List of References

Adams, D. (1979) *Hitch Hiker's Guide to the Galaxy*, Pan.

Smith, M. (2018) *Unity 2018 Cookbook*, Packt.