# Coursework Title

CW3 FP

Jack Bauermeister

Emma Cooper

Contents

[Coursework Title 1](#_Toc165286851)

[Project Proposal 2](#_Toc165286852)

[Project Overview 2](#_Toc165286853)

[Project Aims 2](#_Toc165286854)

[Planning 4](#_Toc165286855)

[Initial Planning 4](#_Toc165286856)

[Detailed Process Planning 4](#_Toc165286857)

[Code Implementation 4](#_Toc165286858)

[Testing 4](#_Toc165286859)

[Goals 4](#_Toc165286860)

[Abstract Testing 4](#_Toc165286861)

[System Testing 4](#_Toc165286862)

[Conclusion 4](#_Toc165286863)

## PROJECT PROPOSAL

### Project Overview

This project aims to allow the user to select an orbital through a graphical user interface (GUI) and select a visual representation of the orbital (1D, 2D, or simple rendered model), which is then displayed to the user through the GUI. The program should have stored rendered models that can be pulled from the directory and displayed and should be able to calculate and graph both the 1D and 2D representations using radical wavefunctions and distribution functions.

### Project Aims

The following aims for this project cover inputs, any GUIs, calculations, and outputs, that should be achieved through the program code. These aims can be used as guidelines for the final testing of both abstract selection of code and the overall final version.

|  |  |  |
| --- | --- | --- |
| **Selection/Function** | **Overview** | **Aims** |
| **Inputs**   * Orbitals type * Visual representation | The user should give both the orbital shape (principle quantum number, and angular momentum) and a visual representation of the orbital. | * Selection buttons for the orbital shape. * Selection buttons for the visual representation. * A button to start the generation of the graph or pull the rendered orbital from the directory. |
| **GUIs** | The GUIs should be aesthetically appealing, clear, and simple for the user to understand. The output and input GUIs should appear in separate windows. | * Labels/prompts should provide clear and understandable instructions to the user. * Any elements in the GUIs should not merge due to colour or overlap/interception. * Should have separate windows to collect the inputs and display output to the user. |
| **Calculations**   * Probability distribution of electrons in orbital | Using the inputs provided through the GUI, if required the probability distribution should be calculated and then graphed in sensible units and in a format for the user. | * Should identify the representation selected. * Should calculate the values to graphed if required. * Should create a graph in a sensible format, including title, axis titles, and units. |
| **Outputs**   * Graph/or Rendered model | Should give the visual representation selected into the output GUI for the user. | * Should produce the visual representation selected, graph, or rendering from the directory. * Should provide any variables/constants used in any calculations completed to the user through the GUI. |

A computer screen with a white screen

Description automatically generatedA screenshot of a computer

Description automatically generatedBelow are examples of what the GUIs (graphical user interface) could look like according to the aims stated.

Figure 2. An example of the output window GUI.

Figure 1. An example of the input window GUI.

## PLANNING

### Initial Planning

[Flowchart showing over process for code to follow]

### Detailed Process Planning

[Any further, more detailed planning of specific processes or functions]

## CODE IMPLEMENTATION

[Code with any additional explanations or notes, not including in code comments]

## TESTING

### Goals

[Summary of the goals the code is tested against, should be similar to the aims]

### Abstract Testing

[Testing of any functions or specific processes individually – apart from the overall code]

### System Testing

[Testing the whole of code including as functions or selection of code tested in abstract testing]

## CONCLUSION