

## **Module: CMP-7000A – APPLICATIONS PROGRAMMING**

### **Assignment: GUI Programming + Technical Design Document**

**Set by:** Debbie Taylor : Debbie.taylor@uea.ac.uk  
**Date set:** 11 November 2020  
**Value:** 60%

**Date due:** 20 January 2021 15:00 [week 1 Semester 2]  
**Returned by:** 24 February 2021  
**Submission:** Blackboard

### **Learning outcomes**

- Design and program a Graphical User Interface in a modern programming language
- Analyse software of others and yourselves, with a view to increasing usability
- Create a Technical Design Document that could be used in most industry settings.

### **Specification**

#### **Overview**

The aim of this assignment is to build on the skills obtained during the first assignment by creating a functioning Graphical User Interface for your console game. Additionally, the process will involve creating a full technical design document (maximum 20 pages) showing your design process from beginning to end, discussing your chosen agile methodology, requirements, UML diagrams and testing.

This will enable you to learn the full impact of software engineering analysis, its importance on design and programming, and gain skills in implementing your own designs.

#### **Description**

You are required to create a functional GUI and produce a maximum 20 page technical design document showing the software engineering processes used for building this system.

1. Produce a full requirements document for your GUI which shows your decision making for the following steps:
  - a. Why you chose your agile methodology, what stakeholders would use this, Persona, Empathy Map, MoSCoW, User Stories, etc.
  - b. Design a UML Use Case Diagram, Textual Use Case, Sequence Diagram and Class Diagram, plus any others you may think would be useful. These all need to be based on your GUI game system.
  - c. Full testing methods, results and changes made, based on unit testing and user feedback. Describe the strategies/techniques used and show evidence of testing, at design and system level e.g. Unit testing, Lo-Fi etc...

- d. Complete a full evaluation and analysis of your design and implementation. This will involve a discussing the system limitations, potential future enhancements, how much of your assignment 1 code was re-used, what other systems you looked at to get inspiration, etc.
  - e. The final page should concentrate on self-reflection. What did you learn? Where did you encounter problems? What would you do differently if you had to do this again? etc.
  - f. Ensure you use relevant references and citations within your bibliography and document. These are excluded from the maximum 20 pages.
2. Program a GUI implementation of your game. You may use Python tkinter or another library but it must have Python as its base code.
  - a. The implementation will need to provide a user friendly interface to process the following:
    - i. Allow a user to relay and receive instructions via the GUI instead of the command line
    - ii. Allow a user to request current and previous location information and display this with both text and images. The user should be able to choose which order they want the information displayed e.g. oldest to newest or newest to oldest
    - iii. Allow a user to request character information and display this with both text and images
    - iv. Allow a user to request current collectables and display them in alphabetical order
    - v. Allow a user to request general progress
    - vi. Optional customisation /additional features
  - b. The programming should make appropriate use of components, layout etc. and be as functional as possible. It should be easy to use but doesn't have to look professional
  - c. Make appropriate use of data types, code layout, comments, testing evidence, packages etc.
  - d. You should test as you go as you will need to show this within your technical design document.
3. Record a maximum 7 minute video of your GUI working for the above processes, as well as a sample of your code. You do not have to include any Design Document information within this video

### **Relationship to formative assessment**

This work builds on the formative work you have completed during the lab sessions and also on the skills learned during Assignment 1. You are encouraged to approach your lab leaders for feedback during the lab sessions as some of the work required for this assessment is similar to those covered in the labs.

### **Deliverables**

You must zip up your GUI code, video and PDF of your Technical Design document into a single folder. Then submit the zipped folder to blackboard by 3pm Wednesday 20 January 2021 (week 1 Semester 2). It should use the following format studentID.zip e.g. *10000000.zip*

## **Resources**

Lecture notes and previous lab sessions are highly relevant to this work. Lab leaders will be available during lab sessions to help with general queries.

A basic example of a previous Technical Design Document has been uploaded to Blackboard. This covered a different topic (music player) but as the principles behind software engineering remain relevant for all topics, please feel free to refer to this when creating your game's Design Document. Please however be aware of the plagiarism rules below.

### **A note on use of additional sources and plagiarism**

Please note that while use of texts, or online sources, is encouraged in order to learn design and programming principles, use of functions, lists, etc; they are not a substitute for completing the work yourself. It is not appropriate to find solutions or part solutions to assignments, and submit them as your own work. Neither is it allowed to post questions on online forums requesting help or solutions to specific assignment tasks. To do either (copying/requesting) would be in breach of the university's regulations on plagiarism and collusion (General Regulation 18).

In the instances where you do use code (or any other work) copied from any source, you must acknowledge the source (e.g. including a comment with the URL and author alongside the copied sections) If in doubt, approach the coursework setter to discuss what is appropriate.

## Marking scheme – Technical Design Document and GUI Video: 60%

<b>1<sup>st</sup> Marker Name</b>	<b>Student Number</b>
<b>Date:</b>	
<b>2<sup>nd</sup> Marker Name:</b>	<b>Date:</b>

Marking Details	Mark %	Marking Comments
Design Document : Requirements Agile methodology, Stakeholders, Persona, Empathy map, MoSCoW, User stories etc...	20%	
Design Document : UML Diagrams: Use Case, Textual, Sequence, Class, etc...	20%	
Design Document : Testing Evidence Usability, strategies, techniques, examples, etc...	10%	
Design Document : Evaluation and Analysis Limitations, enhancements, code re-use etc...	10%	
Design Document : Self-reflection What did you learn? Problems encountered? What would you do differently in future? , etc.	10%	
Design Document : Report Structure and Referencing Appropriate structure, use of English, spelling, grammar, bibliography reference material, etc.	5%	
GUI Video : Implementation and aesthetics Appearance, usability, consistent approach, additional functionality	15%	
GUI Video : Programming code Appropriate naming of components and functions, data types, code layout, comments, classes/class hierarchies, loops, etc...	10%	

Extra Comments:

Total Score