|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task:** | | **2** | | |
| **Task Title:** | | **Project** | | |
| **Task Code:** | | **AT2 PRJ Task 1** | | |
|  | |  | | |
| Assessment type (): | | | | |
|  | Questioning (Oral/Written) | |  | Portfolio |
|  | Practical Demonstration | |  | Project |
|  | 3rd Party Report | |  | Other – Please Specify |

|  |
| --- |
| The base requirements this assessment task include:   * IDE or editor for developing Python programs (only PyCharm supported by the college) * Access to Office 365 & Microsoft Word * Access to the program template (.py) and the readme.md instructions * Task list and task template   Use of some of these items may not occur in this part of the assessment task. |
| Assessment Due NOTICE: While this assessment is due on the second last week of the course. You **MUST** review elements of this submission with your lecturer before submission to pass this assessment.  This assessment is due on the second last week of the term on which it was administered.  Refer to Blackboard for most accurate dates, which may alter due to unforeseen circumstances. |
| Instructions The project consists of three parts:   1. This journal and general project instructions 2. A template python program that you can use as a basis of your development work 3. A readme.md file (markdown) that contains development specific instructions as well as sections that you need to complete.   Follow the steps outlined in each of these individual components and submit all three to successfully complete this assessment.  You must follow the instructions in and complete the tasks for all three components to complete this course. You may need to submit additional files |
| Important If you are using a different IDEs or a different structure for your application, then assistance with those tools and forms may be limited. Discuss with your lecturer before straying too far off the path. |
| Scenario You are currently working as a junior software developer at words-are-us, a Perth-based company that develops content and apps to build engagement.  You have been contracted to gain a deeper understanding of the popular tool Wordle and develop a prototype app that emulates the original wordle algorithm but provides these capabilities from the command line. As a junior developer, you have access to your peers, the senior developer (played by your lecturer), and occasional access to the customer (usually also played by your lecturer). |
| Specific Instructions This file is to be submitted as a journal that demonstrates the following aspects of your competency:   * Your ability to prioritise your own work * Obtain feedback to validate and enhance your design decisions * Confirm that the application meets specifications * Evaluate and reflect on decisions you made in the process of development   Save the file as:   * XXX\_ICTPRG302\_Proj\_Journal.docx   Replacing XXX with your initials.  For example, Adrian Gould would use AG\_ICTPRG302\_Proj\_Journal.docx for his submitted filename.  Upload any code as a PyCharm project in a zip-file. Remove the virtual environment (**venv** or **.venv**) from the zip-file before uploading it to Blackboard. |
| Answering Questions When a step includes a question, you must attempt to answer it. A word count is sometimes given, but most answers require at most 1-3 paragraphs.  All answers must be in complete sentences unless indicated. You must use your own words unless otherwise specified. |
| Sources of Information In industry, it is good practice to keep track of where information was obtained. This is especially true if it is a written document, or even code.  If you answer any questions using information from web sites, please include the site name and URL (Web site address) after the answer. Likewise, include the title and author for books and magazine articles. For example:   * RS Electronics Ltd: <https://au.rs-online.com/> * Slack API Documentation, Users List Method: <https://api.slack.com/methods/users.list>  Code Storage and Issue Tracking An industry-standard tool for managing program source is git. GitHub is a free service that provides hosting for git repositories. You may choose to use this system for source control, task management, or both. However, it is not required for this project.  You may also use OneDrive within your college Office365 to store a backup of your code or keep a copy on a USB thumb drive. |
| *This space left intentionally blank.* |

| **Weekly** | **Reflections** | Words 50-100 |
| --- | --- | --- |
| 00 | Familiarisation At any stage during this assignment, you may consult the stakeholder(s) or their representative(s).  Today, you were introduced to the business requirements of the project. You would have been given an opportunity to interact with competitor products. For example, if you are performing the Wordle project, you would have got to try the commercial version of the game and discuss how the game works (system behaviour) it in class.  Bellow, note down the following:   * 1-2 lines describing the business motivation * 5-6 lines describing the system behaviour * An example of at least **two** python constructs you learned previously and one line explaining how they could be relevant to the problem. For example, for loops, Boolean expressions, file reading, etc. |  |
|  | I’m a junior developer contracted to develop a prototype app of the popular game Wordle in order to build engagement for a company.   The game will have the user try to guess a 5 letter word within 6 attempts. The attempts must be valid English words. When an attempt is made, the user will be told whether letters in their word is correct and/or in the correct position. If they can’t guess the word in 6 tries then they lose. Two previous python constructs that I think would be useful for this task are Boolean expressions and file reading. Boolean expressions can be used to compared the word input from the user to the Word of the Day and file reading could be used to access from a list of words that could be Word of the Day. |  |
| 01 | Creating an algorithm Complete at least two versions of the algorithm (lecturer determines the algorithm). Present your initial version to one of your peers and the senior developer. Document how your thoughts changed as a result of the interaction.  Your algorithm is **not** documented here (it should be in the readme.md) |  |
|  | I completed a flowchart and some pseudocode in order to illustrate my thought processes. My initial attempt at doing pseudocode was contained language that was too casual and after some advice from the lecturer, I cleaned it up to its current form:  Output(‘welcome message”)  Input Word\_of\_the\_Day  Output(‘You have 6 attempts to guess the word’)  attempts = 6 While attempts > 0 then:  Input(word)  If Word\_of\_the\_Day = word then:  Output(‘congratulation’)  Else:  Call (display\_correct\_letters)  endif  attempts = attempts -1  end while  If attempts = 0 then:  Output(‘You loose’)  endif  I began with the flowchart and, interesetingly, I did the attempts backwards to how everyone else did. I started with 6 attempts counting down to 0 but, for programming, it made more sense to have attempts start at 0 and count up. I changed it to be more logical in the end. |  |
| 02 | Edge cases What are some edge cases that you have considered that could impact the design of your algorithm? Discuss with your lecturer and peers and write your reflection here |  |
|  | Edge cases are guesses that contain double letters. Example would be a Word of the Day like WORLD with a guess of BOOKS. Will the first O be given just an X or will the second O get an O too? Will repeated letters be checked properly and give the user the correct information? |  |
| 03 | Project planning What tool have you used to manage your tasks and why? Include a screen shot of your initial task list. |  |
|  | We used draw.io to help create the flowcharts and GitHub to organise our project. Microsoft Excel was used to create a task list to help manage time effectively. |  |
| 04 | Simple testing and inspection How do you intend on testing your code? Give an example of an existing test case and a test case that you will generate. |  |
|  | I tested my code by copying and moving functions that are currently being worked on, into its own IDLE file. Separating the code made it easier to ensure that there wasn’t any unintended interference from other functions. When testing it this way I had to remember to include “cheats” like asking for a secret word input for comparison purposes. Those will be removed later.    When it came to scoring, there were 3 main test cases that we used in class:  1. Guess letters were all different, secret word letters were all different e.g. drain and float.  2. Guess letters contained a double letter, secret word letters were all different e.g. hello and petal.  3. Guess letters were all different, secret word contained a double letter e.g. spray and array. |  |
| 05 | Dealing with uncertainty What are some issues that you are still uncertain about? How do you intend to get clarity? Ensure you discuss with your lecturer |  |
|  | At this point, I’m still unsure about how to compare the word of the day with the guess word. I understand that you need to ‘break’ the words into separate letters and comparing each letter but, code-wise, it’s going to take a bit of research.  I intend to get clarity by asking our lecturer any questions I may have. |  |
| 06 | Catch up This section is not assessed but is strongly recommended  Write any challenges and achievements you are having |  |
|  | 19/05/22 Today we developed a function in order to ask the user to input a file name and then get it. We made another function to import the words within that file into a list. We looked over how to randomly select a word from that list for our Word of the Day and finally we made a function to welcome the player to our game. It included rules and objectives and a personalised message with the player’s name.  26/05/22 Today we completed the functions to get the text files as well as randomly select the word of the day.  09/06/22 The scoring system is now fully functioning after a lot of fiddling around and research. |  |
| 07 | Code review Review your code with the senior developer and write down any changes you need to make because of the review. Add them to your task list and include a screenshot. |  |
|  | *Space for your answers* |  |
| 08 | Application review Review the working of the application with the client (likely, your lecturer). Write down one thing you got right and one thing you got wrong and explain why you think what worked and what didn’t.  Were there any non-functional issues? Crashes, unexpected output? |  |
|  |  |  |
| 09 | You DID IT!!! Well done on completing your first software development project!  You’ve come a long way. Don’t forget to submit all elements |  |