

Task ID	Created At	Completed At	Last Modified	Name	Section/Column	Assignee	Assignee Email	Start Date	Due Date	Tags	Notes	Projects	Parent Task	Priority
1.2006E+15	16/7/21		16/7/21	Nice-to-haves	To do					code		Terminal Application		Low
1.2006E+15	16/7/21		16/7/21	Better selection of words for 1 player mode						code			Nice-to-haves	
1.2006E+15	16/7/21		16/7/21	Ability to have spaces and hyphens in 2 player mode						code			Nice-to-haves	
1.2006E+15	16/7/21		16/7/21	Keep list of words guessed						code			Nice-to-haves	
											<p>Your source control repository should be zipped and submitted via the Canvas LMS platform.</p> <p>The Software Development Plan must be submitted as a plain-text file using markdown</p> <p>Your application, exports from project management software, screenshots, and any other part of your submission should be in your source control repository. Any file you submit should have a filename which reflects what the file is, contain your name, and the requirements code it relates to.</p> <p>For example: bill-lastname-T1A2-13-project-mgt-checklist1.jpg</p>			
1.2006E+15	10/7/21		10/7/21	Submit ZIP file	To do				17/7/21	project management		Terminal Application		High
											<p>Implement all features in the software development plan you have designed. You must utilise a range of programming concepts and structures using Python 3 such as:</p> <ul style="list-style-type: none"> <li>- variables and variable scope</li> <li>- loops and conditional control structures</li> <li>- error handling</li> </ul> <p>Apply all style and conventions for the programming language consistently to all code produced.</p> <p>Creates an application which runs without error and has features that are consistent with the development plan.</p>			
1.2006E+15	10/7/21	16/7/21	16/7/21	Implement Application	Complete				16/7/21	project management	<p>Design a help file which includes a set of instructions which accurately describe how to use and install the application.</p> <p>You must include:</p> <ul style="list-style-type: none"> <li>- steps to install the application</li> <li>- any dependencies required by the application to operate</li> <li>- any system/hardware requirements</li> <li>- a written explanation of the different features of the application</li> </ul>	Terminal Application		High
1.2006E+15	10/7/21	16/7/21	16/7/21	Help File					16/7/21	project management	The help file should be 100 - 200 words.		Implement A	High
1.2006E+15	10/7/21	16/7/21	16/7/21	Developer Operations	Complete				16/7/21	project management	<p>Utilise source control throughout the development of the application by:</p> <ul style="list-style-type: none"> <li>- making regular commits (at least 20 commits) with a commit message that summarises the changes</li> </ul>	Terminal Application		High
1.2006E+15	10/7/21	15/7/21	15/7/21	Source Control					16/7/21	project management		Developer Operations		High

						<p>Utilise a project management platform to track the development of the application by:</p> <ul style="list-style-type: none"> <li>- having features itemised and broken down into checklists</li> <li>- setting deadlines, duration or a time-frame for each feature or task</li> <li>- prioritising tasks</li> </ul> <p>You must submit screenshots, or a valid export from a project management application to satisfy this requirement. If you are using a non-standard project management application, please discuss with your trainer whether they can access the file. If in doubt, screen capture your project management application and submit screen captures.</p>		Developer OJ High
1.2006E+15	10/7/21	15/7/21	15/7/21 Project Management			<p>Utilise operating system features to facilitate the building and testing of the application by:</p> <ul style="list-style-type: none"> <li>- creating a wrapper script which turns the application into an executable</li> <li>- and/or creating a script which performs the tests you have specified in your design document</li> </ul> <p>A good mark can be achieved by executing either of these two options in a sophisticated fashion. For maximum marks, attempt both.</p>		
1.2006E+15	10/7/21	16/7/21	16/7/21 Operating System features			<p>You must submit a script file in with your source control repository to satisfy this requirement.</p> <p>Design TWO tests which check that the application is running as expected.</p> <p>Each test should:</p> <ul style="list-style-type: none"> <li>- cover a different feature of the application</li> <li>- state what is being tested</li> <li>- provide at least TWO test cases and the expected results for each test case</li> </ul> <p>An outline of the testing procedure and cases should be included with the source code of the application and written in markdown.</p>		Developer OJ High
1.2006E+15	10/7/21	16/7/21	16/7/21 Test Application	Complete	16/7/21 project man:	2 player mode: Test that a long word with repeated letters displays properly as those letters are guessed.	Terminal Application	High
1.2006E+15	10/7/21	15/7/21	15/7/21 Test 1 - reveal word as guessed		13/7/21 project man:		Test Applicat	High
1.2006E+15	10/7/21	16/7/21	16/7/21 Test 2 - play again		16/7/21 project management		Test Applicat	High
1.2006E+15	12/7/21	16/7/21	16/7/21 Ask player if they want to play again (Y/N)	Complete	14/7/21 code		Terminal Application	Low
1.2006E+15	12/7/21	16/7/21	16/7/21 Print snowman status & word status	Complete	14/7/21 code		Terminal Application	High
1.2006E+15	12/7/21	15/7/21	16/7/21 After every guess, clear the screen and show an updated screen	Complete	14/7/21 code		Terminal Apj Print snowm	High
1.2006E+15	12/7/21	15/7/21	16/7/21 After every correct guess, print the snowman's status and the word with the correct letters filled in	Complete	14/7/21 code		Terminal Apj Print snowm	High
1.2006E+15	12/7/21	16/7/21	16/7/21 If all letters are guessed correctly, print a "you won" screen	Complete	14/7/21 code		Terminal Apj Print snowm	High
1.2006E+15	12/7/21	15/7/21	16/7/21 After every incorrect guess, print the snowman with one more body part missing and the word with the correct guesses filled in	Complete	14/7/21 code		Terminal Apj Print snowm	High
1.2006E+15	12/7/21	16/7/21	16/7/21 After seven incorrect guesses, print a puddle and the completed word	Complete	14/7/21 code		Terminal Apj Print snowm	High
1.2006E+15	12/7/21	15/7/21	15/7/21 Get letter guesses from player	Complete	13/7/21 code		Terminal Application	High
1.2006E+15	12/7/21	15/7/21	15/7/21 Ensure guess is either one letter or equal to the length of the answer	Complete	13/7/21 code		Terminal Apj Get letter gu	High

1.2006E+15	12/7/21	15/7/21	15/7/21 Ensure guess is only letters	Complete	13/7/21 code	Terminal Apj Get letter gu High
1.2006E+15	12/7/21	15/7/21	15/7/21 Convert guess to upper case	Complete	13/7/21 code	Terminal Apj Get letter gu High
1.2006E+15	12/7/21	15/7/21	15/7/21 Add guesses to a guessed list	Complete	13/7/21 code	Terminal Apj Get letter gu High
1.2006E+15	12/7/21	15/7/21	15/7/21 Check guesses are not already on the guessed list	Complete	13/7/21 code	Terminal Apj Get letter gu High
1.2006E+15	12/7/21	15/7/21	15/7/21 Get word or words	Complete	13/7/21 code	Terminal Application High
1.2006E+15	12/7/21	15/7/21	15/7/21 Check entered words are only letters, spaces or hyphens	Complete	13/7/21 code	Terminal Apj Get word or High
1.2006E+15	12/7/21	15/7/21	15/7/21 Convert to upper case	Complete	13/7/21 code	Terminal Apj Get word or High
			Print ascii art snowman and blank spaces for missing			
1.2006E+15	12/7/21	15/7/21	15/7/21 letters	Complete	13/7/21 code	Terminal Application High
1.2006E+15	12/7/21	15/7/21	15/7/21 Underscores for missing letters	Complete	13/7/21 code	Terminal Apj Print ascii art High
1.2006E+15	12/7/21	15/7/21	15/7/21 Show spaces & hyphens	Complete	13/7/21 code	Terminal Apj Print ascii art High

Write TWO status updates on your application. You must write a status update when:

- you significantly change a feature or your implementation plan at any point
- encounter a significant issue, challenge or roadblock
- are halfway through implementing the features
- prior to / after testing the application

You should submit these logs throughout the course of developing your application.

Each status report should be between 100 - 200 words.

The Development Log should be written in markdown and placed in a file named development-log.md in the root of your source control repository. Ensure you put the date of the log at the top of each log entry.

Please send your log to your educator as a markdown file as you complete them. This is so we can provide you with

1.2006E+15	10/7/21	15/7/21	15/7/21 Production Log	Complete	15/7/21 project management	Terminal Application High
1.2006E+15	10/7/21	12/7/21	12/7/21 Status update 1		12/7/21 project management	Production Lt High
1.2006E+15	10/7/21	15/7/21	15/7/21 Status update 2		15/7/21 project management	Production Lt High
1.2006E+15	12/7/21	13/7/21	13/7/21 1 player or 2	Complete	12/7/21 code	Terminal Application Medium
			Create list of possible words to choose from randomly if			
1.2006E+15	12/7/21	13/7/21	13/7/21 only 1 player		12/7/21 code	1 player or 2 Medium
1.2006E+15	12/7/21	12/7/21	12/7/21 Create ascii art	Complete	12/7/21 code	Terminal Application High
1.2006E+15	12/7/21	12/7/21	12/7/21 Completed snowman		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without hat		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without scarf		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without eyes		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without nose		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without head		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without arms		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Snowman without torso		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Puddle		12/7/21 code	Create ascii : High
1.2006E+15	12/7/21	12/7/21	12/7/21 Welcome screen		12/7/21	Create ascii art
1.2006E+15	8/7/21	10/7/21	10/7/21 Write out requirements in Asana	Complete	9/7/21 project management	Terminal Application High
1.2006E+15	8/7/21	10/7/21	10/7/21 Get github backups working	Complete	9/7/21 project management	Terminal Application High
1.2006E+15	8/7/21	10/7/21	10/7/21 Send idea to Oliver	Complete	8/7/21 project management	Terminal Application High

						Brainstorm ideas What are the minimum viable features? Nice to have? Dream? What sort of errors are you likely to run into? Scaffold a rough idea of what your program flow might look like			
1.2006E+15	8/7/21	8/7/21	10/7/21	Choose application	Complete	8/7/21 project man	Get this, do this, output this	Terminal Application	High
1.2006E+15	8/7/21	8/7/21	10/7/21	Gymnastics competition roster		project management		Choose application	
1.2006E+15	8/7/21	8/7/21	10/7/21	Raffle		project management		Choose application	
1.2006E+15	8/7/21	8/7/21	10/7/21	Random password generator		project management		Choose application	
1.2006E+15	8/7/21	8/7/21	10/7/21	Hangman		project management		Choose application	
							Develop an implementation plan which: - outlines how each feature will be implemented and a checklist of tasks for each feature - prioritise the implementation of different features, or checklist items within a feature - provide a deadline, duration or other time indicator for each feature or checklist/checklist-item  You must submit this as a written document, ideally in a tabular format, and it is suggested that you enter your checklists into an appropriate project management application to assist you in completing T1A2-10.		
1.2006E+15	10/7/21	12/7/21	12/7/21	Implementation Plan	Complete	11/7/21 project man	Your checklists for each feature should have at least 5 items.	Terminal Application	High
							Design a software development plan for a terminal application  This task should be written in a single document in a report format with headings.  It is suggested that this document is placed in the root of your source control repository, and that you write it in markdown.  NOTE: we do not accept submission of Word, Pages or Open office documents. We will only accept PDFs, or markdown files for written assessments.		
1.2006E+15	10/7/21	12/7/21	12/7/21	Software development plan	Complete	11/7/21 project man	Develop a statement of purpose and scope for your application. It must include: - describe at a high level what the application will do - identify the problem it will solve and explain why you are developing it - identify the target audience	Terminal Application	High
1.2006E+15	10/7/21	12/7/21	12/7/21	Statement of Purpose and Scope	Complete	11/7/21 project man	- explain how a member of the target audience will use it	Terminal Ap	Software dev High

						<p>Develop a list of features that will be included in the application. It must include:</p> <ul style="list-style-type: none"> <li>- at least THREE features</li> <li>- describe each feature</li> </ul>	
						<p>Ensure that your features include the following language elements and concepts:</p> <ul style="list-style-type: none"> <li>- use of variables and the concept of variable scope</li> <li>- loops and conditional control structures</li> <li>- error handling</li> </ul>	
1.2006E+15	10/7/21	12/7/21	12/7/21 Features	Complete	11/7/21 project man	<p>Note: If the features you described do not require you to use all of the above it is likely that your application is not sophisticated enough. Consult with your educator to check your features are sufficient to address the assessment criteria.</p> <p>Develop an outline of the user interaction and experience for the application.</p> <p>Your outline must include:</p> <ul style="list-style-type: none"> <li>- how the user will find out how to interact with / use each feature</li> <li>- how the user will interact with / use each feature</li> <li>- how errors will be handled by the application and displayed to the user</li> </ul> <p>Develop a diagram which describes the control flow of your application. Your diagram must:</p> <ul style="list-style-type: none"> <li>- show the workflow/logic and/or integration of the features in your application for each feature.</li> <li>- utilise a recognised format or set of conventions for a control flow diagram, such as UML.</li> </ul>	Terminal Ap Software de High
1.2006E+15	10/7/21	12/7/21	12/7/21 User Interaction and Experience	Complete	11/7/21 project man		Terminal Ap Software de High
1.2006E+15	10/7/21	12/7/21	12/7/21 Control Flow Diagram	Complete	11/7/21 project man	Please provide an image file or PDF of your diagram.	Terminal Ap Software de High