UNIVERSITY OF TECHNOLOGY SYDNEY

52685 Working with Data and Code

Assessment Item 2: Code Prototype Project (60%)

Task Description

Working individually, you will advance your code project idea to complete an original code prototype. You will work both inside and outside class to develop your project. You will submit a project prototype that demonstrates your code literacy. Your code project submission will include a 3-minute explainer video, a 1000-word written personal reflection, and a Git repository with the completed prototype. In the voice-narrated explainer video, you will first provide an overview of the project, and then focus on core aspects of the prototype and explain the operation of your code. In the written reflection, you will reflect on your personal learning experience throughout the project, referencing relevant key concepts and technical terms introduced during the semester to demonstrate how the code project has developed your code and data literacy. In the Git repository you will include all required source code and project dependencies (e.g. code libraries, media files etc.) and include sufficient documentation to navigate and run the code prototype.

Specifications

Length:	3-minute explainer video, 1000-word personal critical reflection, Git repository.
Due Date:	Final Assessment Period.
Submission Format:	Video presentation, document, repository link
Subject Objectives:	a, b, c, d and e
CILOS:	1.1, 2.1, 2.2 & 6.2

Criteria

Criteria	Weighting
Quality of project overview and depth of understanding of demonstrated project code	35%
Overall quality and clarity of the explainer video	10%
Depth and quality of written reflection and degree of synthesis with the literature and project journal	35%
Evidence of peer collaboration, clarity of expression and quality of APA referencing	10%
Degree of original code within the prototype, quality of code comments, completeness and quality of Git repository	10%

Brief

You are to now advance your code project idea into an original working code prototype. The prototype must include a **substantial original code contribution plus code comments** to clearly demonstrate important code elements (e.g. variables, lists, functions, classes) and their operation (more details on the requirements for code originality are listed under 'Notes on Your Original Code Contribution' below). Also, be mindful, elements not directly generated by code (e.g. sprites, game sounds), are not regarded as you "building though code" and similarly do not contribute to your code literacy. As a result, what may appear to be a complex multiplayer game (but with minimal original code) may score significantly lower that a completely original text-based game running on an CLI (Command Line Interface) where a significant amount of original code was included in the prototype. In sum, the complexity of output is not necessarily an indicator of code engagement nor understanding or code literacy. Rather, it is **your exploration, engagement with code and developing code literacy that its important, the prototype is the means (the vehicle if you will) that takes you on that journey.**

In a 3-minute voice narrated explainer video, you will *briefly* introduce your project to provide context and then deeply explain your code and how it operates to implement the core features/functionality within your prototype (see 'Explainer Video' section for more).

You will also submit a 1000-word written reflection that demonstrates your code literacy and draws from your 'Project Journal', plus integrates the concepts and literature introduced during coursework — lectures, readings, viewings, listenings, playings, code conundrums, tutorials and so on — in addition to any supplemental research you have conducted during the development of your project). More is available in the 'Written Reflection' section.

You will also submit your project via a publicly available GitHub repository that will include all source code, libraries, assets (e.g. graphics, video, audio etc.) together with a README.md document. (Visit 'Git Repository' for more).

Notes on Your Original Code Contribution.

The inclusion of work from other code authors (e.g. from tutorials, guides, code libraries) is allowed, however as per traditional written publications (e.g. websites, books, research papers), these must be clearly acknowledged within the code comments in addition to the written personal reflection whenever they are referenced. In sum, your original code contribution, along with that of other authors/contributors must be clear within your code. Failure to clearly identify and acknowledge the source of the code within your prototype will be regarded as plagiarism[†].

[†] University of Technology Sydney. (2023, February 7). *What is plagiarism and cheating?* University of Technology Sydney. https://www.uts.edu.au/current-students/support/academic-support/academic-integrity/what-plagiarism-and-cheating

In addition, while Ethical Use of Generative AI[‡] is allowed in this subject, it comes with clear requirements, including *providing both the prompt and resulting output* as well as the use of APA referencing. Failure to adhere to these Gen AI rules and considerations may have implications regarding academic integrity. It is recommended that you read and understand the rules for both the proper acknowledgement of the work(s) of others and the appropriate use of Gen AI. It is also recommended that you include relevant details in your project journal for ease of recording and later retrieval when it comes to writing your reflective piece.

We also recommend saving early and often so that you have multiple versions (including the last working version of your prototype).

[‡] University of Technology Sydney — Code Subject. (2024). *Ethical use of Generative AI in this Subject*. uts.edu.au. https://canvas.uts.edu.au/courses/33108/pages/ethical-use-of-generative-ai-in-this-subject?module_item_id=1822463

Explainer Video

Your 3*-minute voice-narrated explainer video is where you:

- **provide a brief overview** of your project prototype;
  where you provide the viewer with some context of your project by way of an introduction, including video of it in operation (estimated 20-30s)
- **demonstrate core features and/or functionality** of your prototype;
  where you demonstrate and highlight features and/or functions that you consider the most important, offer the most innovation, novelty or creativity you've built through your code
- then **explain your underlying code** you have written to implement those features and/or functionality; this is where you show and walkthrough the code that you have written and explain, in detail, its operation with respect to your chosen features and/or functionality (i.e. how the code gives rise to those features/functions).

*±10% (i.e. minimum of 180s – 18s ≈ 160s = 2min 40s, maximum of 180s + 18s ≈ 200s = 3min 20s)

Note that, **as an explainer video it is crucial that you explain how the code you have written works and implements the core functionality,** and this helps demonstrate your code literacy. Be mindful that this is the core objective of this aspect of the task so significant time will be spent explaining the working of your code behind core features/functionality. Demonstrating the operation of your prototype and its core features and/or functionality, without explaining the underlying code offers little insight into your code literacy and will deeply affect your criteria score.

In sum your video covers:

- **briefly what you have built:**
  the overview is there only to offer **context of what it is that you have built**, then…
- **what it does:**
  you demonstrate select core features/functions **simply to illustrate what your project does**,
- **explain in depth how these functions/features work via your code:**
  and **your code is explained in depth showing how the functionality emerges**

The video is also judged on its overall quality and clarity as is typically of video presentations and this is also detailed ion the A2 Rubric.

Project Journal

As part of your project, you are expected to implement and regularly contribute to a project journal. **It is important to first note that the 'Project Journal' itself is not assessed. However, it will become a critical source of information when writing your 'Written Reflection' which is assessed.** More on the project journal follows below.

You are required to keep a project journal during the progress of your A2 project prototype. There you are to keep notes such as personal reflections and insights, feelings (surprises and frustrations), code snippets, screenshots, ideas, feedback such as from peers, links to resources, and so on. Code, visual information (such as screenshots and so on) of prototypes is strongly encouraged as these can offer rich examples that can demonstrate deep understanding of code concepts and insights that emerged during prototype development.

We recommend creating a file in Miro (http://miro.com) that will allow you to include a wide variety of sources. The journal will become the main repository for keeping track of your evolving project and evolving code literacy. The project journal will become an important source of information to draw from when coming to write your written reflection.

We also strongly recommend that you regularly add to your journal and, as a bare minimum, that you will contribute at least once per week.

Again, as mentioned previously, the project journal itself is not assessed, however you are expected to draw examples from the journal in your reflective piece.

Visit the 'Project Journal' page on Canvas for more details and tips on your project journal.

Written Reflection

Your submission will include a 1000*-word personal written reflection that **demonstrates how the code prototype had developed your code literacy**. This reflective piece draws on your personal experience, knowledge and insights that have captured during your project journaling, and seamlessly incorporates and synthesises key concepts and literature introduced during coursework (lectures⊕, tutorials, readings and so on) together with any sources independently found during your own research when building your prototype

Usual limits apply 1000±10%; minimum of 900 words, maximum of 1100 words

⊕Note that video lectures typically include a list of references.

A framework — "What? So What? Now What?" — which provides prompts to assist with developing your skills in critical reflection and has been made available on Canvas.

Written reflections are judged on the level of criticality, demonstrated insight and integration of key concepts, relevant technical terms, the literature and personal experience Additionally, a rubric has been included to both demonstrate and assist you with understanding the quality and depth of any written given reflection. You are expected to reference all relevant sources (including any code libraries, tutorials etc. used during the creation of the project, alongside academic and industry sources) using APA Style.

Visit 'Reflective Writing' on Canvas for more details and tips on reflective writing, the recommended "What? So What? Now What?" framework and includes a rubric that details the evaluation of the depth and quality of reflective writing.

Git Repository

The Git Repository is where you include all the required source code, plus prototype assets and any other dependencies in a shared online location. Your repository must include a README.md with details of your project, as well as how to navigate your project files and run your code to execute your project. It is also expected that you will clearly comment your code.

Tutorial time has been allocated to provide you with the necessary instruction to create your GitHub account, create the required read.me and be able to upload your project. You will also find all the relevant details on the 'Git Repository' Canvas page.

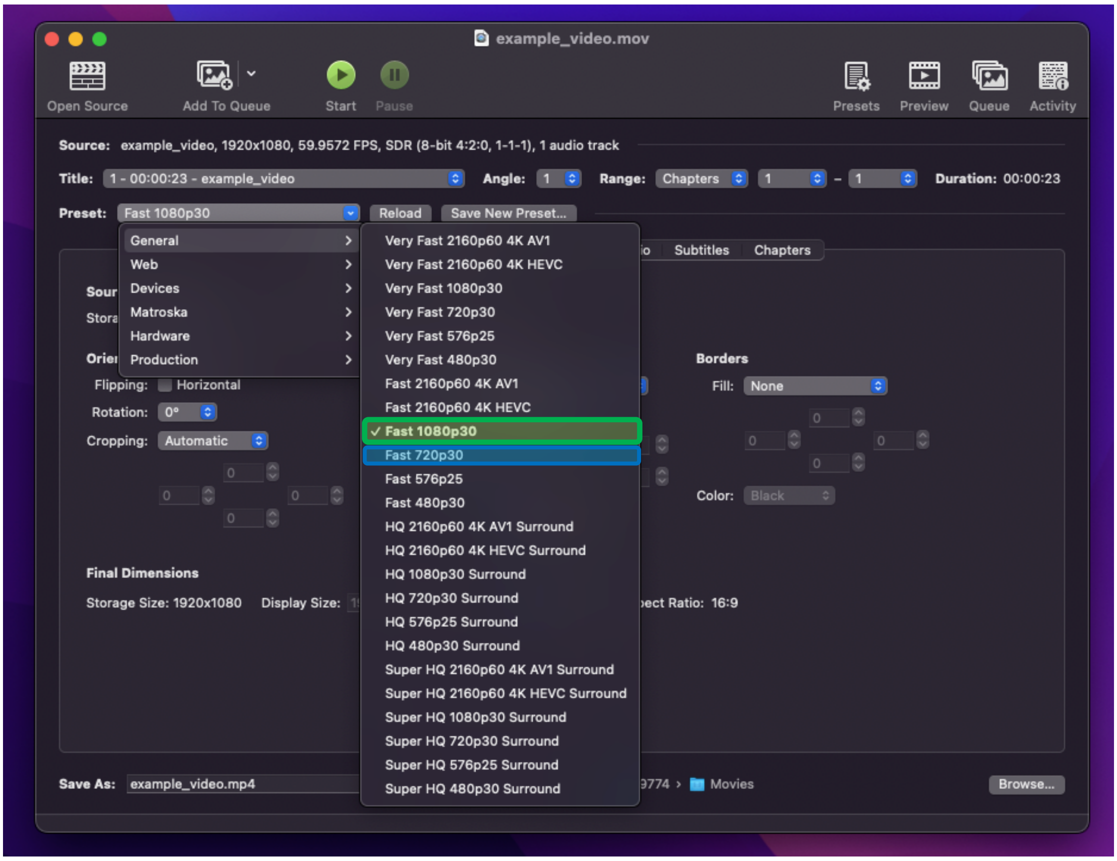
Submission Requirements

1. Explainer Video

You 3-minute explainer video must use narrated video demonstrations of your prototype in action, together with explaining your code. This must be submitted in a video format. You are strongly recommended to practice your presentation to ensure that you cover all key requirements for the video task. The presentation should be professionally delivered and demonstrate your skills in verbal, written and visual communication. If any links are included, ensure they follow the requirements for linked content as described in A Note when using Linked Content.

Video recording and editing

You will submit your video recording as a video file (.mp4). You can record your video via QuickTime screen recording, Zoom recording, MS Teams recording, or other similar formats. You are welcome to edit your video (i.e., it does not need to be from a single take). It is recommended that you trim your presentation to ensure it begins and ends where you want it. You must also appropriately compress your video using transcoding software. It is strongly recommended that you use HandBrake (https://handbrake.fr) which is available for free on MacOS, Windows/PC and Linux platforms. Ensure that you use HandBrake and select Fast 720p30 (good) or Fast 1080p30 (best) to transcode your video file for upload.



Source: example_video, 1920x1080, 59.9572 FPS, SDR (8-bit 4:2:0, 1-1-1), 1 audio track

Title: 1 - 00:00:23 - example_video Angle: 1 Range: Chapters 1 – 1 Duration: 00:00:23

Preset: Fast 1080p30

Flipping: Horizontal

Rotation: 0°

Cropping: Automatic

Fill: None

Color: Black

Storage Size: 1920x1080 Display Size:

ect Ratio: 16:9

Save As: example_video.mp4

Naming Convention

Please ensure you save your video file using the provided naming convention.
LastName_FirstName_StudentNumber__TutorialNumber_ProjectName
(e.g.) Babbage_Charles_12345678_T09_DifferenceEngine.mp4

2. Written Reflection

Your 1000-word written reflection is to be submitted as a PDF document. It is expected to be written in first-person and presented within a cohesive narrative flow. (Note the “What? So What? Now What?” framework is to provide prompts only and is not intended as a formal structure for your work). We encourage the use of in-context relevant visual examples to support your written reflection and demonstrate your understanding. Additional materials may be included in an appendix at the end of the document. The documents must include references in APA format.

3. Git Repository

The Git repository is the online space where you must upload and make your entire finally project prototype accessible. This includes source code, libraries, assets (e.g. graphics, sounds etc.), together with the README.md document. You are also encouraged to include previous versions of the prototype (clearly labelled in separate folders) where the final version may encounter issues or if they are beneficial to include. If any links are included, ensure they follow the requirements for linked content as described in A Note when using Linked Content. If you decide to use linked projects, we would also encourage you to **both upload and link your project** to ensure it can be accessed and graded.

A Note when using Linked Content

While common sense, it included here for clarification as far too often we have received submissions that have included invalid/non-working links, or valid links that link to irrelevant/incorrect content, and/or the content is not accessible as the appropriate read access has not been provided (including passwords that were not supplied or incorrect). As a frame of reference, consider that you are providing your project to a client. As such it is your responsibility to ensure that the client has full access to all of your relevant content

As it is your responsibility to ensure that any linked content is fully accessible so it can be graded, we recommend that **you triple-check (i.e. test to ensure everything works)**:

- **URLs/links are valid and working**
- **URLs link to the appropriate content**
- **URLs provide the necessary public access** by providing read/view/public access to **anyone with the link**. **Do not password protect links**.

It is **your responsibility to provide valid, working links, to the necessary content that is accessible and able to be publicly viewed. Ultimately, if your work cannot be viewed, it cannot be assessed and consequently cannot be graded.**

Rubric

Criteria	High Distinction	Distinction	Credit	Pass	Fail
Quality of project overview and depth of understanding of demonstrated project code (35%)	The overview offers significant project context. The key features/functions of the prototype are expertly identified and demonstrated. The operation of the underlying code of the key features/functions is then expertly explained and demonstrated in significant depth, demonstrating an outstanding level of understanding.	The overview offers very good project context. The key features/functions of the prototype are solidly identified and demonstrated. The operation of the underlying code of the key features/functions is then explained and demonstrated in solid depth, demonstrating a very good level of understanding.	The overview offers sound project context. The key features/functions of the prototype are competently identified and demonstrated. The operation of the underlying code of the key features/functions is then explained and demonstrated in sound depth, demonstrating a sound level of understanding.	The overview offers acceptable project context. The key features/functions of the prototype are sufficiently identified and demonstrated. The operation of the underlying code of the key features/functions is then explained and demonstrated in adequate depth, demonstrating an acceptable level of understanding.	The overview offers little to no project context. The key features/functions of the prototype are insufficiently identified and/or demonstrated. The operation of the underlying code of the key features/functions fails to be explained and/or demonstrated in inadequate depth and does not demonstrate an acceptable level of understanding.
Overall quality and clarity of the explainer video (10%)	In terms of demonstrated content, audiovisual appearance, and duration, the quality and clarity of the explainer video achieves excellent professional standards.	In terms of demonstrated content, audiovisual appearance, and duration, the quality and clarity of the explainer video achieves high professional standards.	In terms of demonstrated content, audiovisual appearance, and duration, the quality and clarity of the explainer video achieves sound professional standards.	In terms of demonstrated content, audiovisual appearance, and duration, the quality and clarity of the explainer video achieves sufficient professional standards.	In terms of demonstrated content and/or audiovisual appearance and/or duration, the quality and/or clarity of the explainer video fails to achieve sufficient professional standards.

Depth and quality of written reflection and degree of synthesis with the literature and project journal (35%)	The personal written reflection is of outstanding depth and quality, seamlessly integrating relevant concepts, viewpoints, sources from the literature, coursework and independent research, complemented and demonstrated by personal critical insights and reflections experienced and captured during the evolution of the prototype through the project journal.	The personal written reflection is of solid depth and quality in its attempt at integrating relevant concepts, viewpoints, sources from the literature, coursework and independent research, that is complemented and demonstrated by a very good attempt at offering personal critical insights and reflections experienced and captured during the evolution of the prototype through the project journal.	The personal written reflection is of sound depth and quality in its attempt at integrating relevant concepts, viewpoints, sources from the literature, coursework and independent research, that is complemented and demonstrated by a sound attempt at offering personal critical insights and reflections experienced and captured during the evolution of the prototype through the project journal.	The personal written reflection is of acceptable depth and quality in its attempt at integrating relevant concepts, viewpoints, sources from the literature, coursework and independent research, that is complemented and demonstrated by a sufficient attempt at offering personal critical insights and reflections experienced and captured during the evolution of the prototype through the project journal.	The personal written reflection fails to meet an acceptable depth and /or quality. Concepts, viewpoints, sources from the literature, coursework and independent research may be irrelevant or absent. This may or may not be complemented with an insufficient attempt at offering personal critical insights and reflections experienced that may or may not have been captured during the evolution of the prototype through the project journal.
Evidence of peer collaboration, clarity of expression and quality of APA referencing (10%)	There is exceptional evidence of peer collaboration, clarity of expression and quality of APA referencing within the written reflection.	There is solid evidence of peer collaboration, clarity of expression and quality of APA referencing within the written reflection.	There is sound evidence of peer collaboration, clarity of expression and quality of APA referencing within the written reflection.	There is satisfactory evidence of peer collaboration, clarity of expression and quality of APA referencing within the written reflection.	There is unsatisfactory evidence or absence of peer collaboration, and/or unacceptable clarity of expression and/or unacceptable quality or absence of APA referencing within the written reflection.

Degree of original code within the prototype, quality of code comments, completeness and quality of Git repository (10%)	The prototype code demonstrated an outstanding degree of originality. The volume and quality of in-situ code comments were of an excellent standard. The completeness of the Git repository — source code, project files, code libraries, external dependencies, assets and documentation within the readme.md file — was of an exceptional standard.	The prototype code demonstrated a very high degree of originality. The volume and quality of in-situ code comments were of a solid standard. The completeness of the Git repository — source code, project files, code libraries, external dependencies, assets and documentation within the readme.md file — was of a very good standard.	The prototype code demonstrated sound degree of originality. The volume and quality of in-situ code comments were of a sound standard. The completeness of the Git repository — source code, project files, code libraries, external dependencies, assets and documentation within the readme.md file — was of a sound standard.	The prototype code demonstrated satisfactory originality. The volume and quality of in-situ code comments were of an acceptable standard. The completeness of the Git repository — source code, project files, code libraries, external dependencies, assets and documentation within the readme.md file — was of an adequate standard.	The prototype code did not demonstrate satisfactory originality. The volume and quality of in-situ code comments were of not an acceptable standard. The completeness of the Git repository — source code, project files, code libraries, external dependencies, assets and documentation within the readme.md file — was not of an adequate standard.