# UTS Animal Logic Academy

Technical Direction for 3D Animation and Graphics Projects (41801)
Assessment Brief

---

## Assessment Task 2: Developing Part of a 3D Animation / VFX Pipeline

Type: Project / Group Work /Group and Individually assessed
Weighting: 60%
Length: Approximately 50-70 hours of work to produce a portfolio of code and associated documentation.
Due: Part 1 (Work in progress presentation) Week 7 / Friday September 22, 2023 (Weight 20%). Part 2 (Final submission) Week 12 / Friday October 27, 2023 (Weight 80%)
Note: Students will receive feedback every two weeks on their progress in class

## Assessment Description

In this assessment, students gain hands-on experience in creating and maintaining part of a 3D animation / VFX production pipeline.

Students are required to work in small groups with individual tasks to create tools for a section of a 3D animation / VFX pipeline. Based on a specific brief, they plan and execute the development of these tools using industry standard workflows. Deliverables include, functional computer code, documentation and evidence of Agile methodologies and usage of Git.

## Assessment Task Summary

Students are required to work in small groups to create sections of a 3D animation / VFX pipeline.

Each group will create the following:

- An asset publishing system
- A tool that performs an integrity check on each asset
- A tool that builds a lighting scene containing all the elements required to create a rendered image
    - Please note: Assets for departments such as Modelling, Animation and Surfacing will be supplied to create this scene.

Groups will present a file structure system for their project and outline why they have chosen said structure.

## Assessment Task - Part 1: Work in progress presentation

Groups will give a presentation to the class which outlines their project plan.

Each member of the group will present at least one section of the presentation which will contain the following:

- Production schedule and plan (group assessed) which contains:
    - A schedule for the work, with milestones, and the student who is assigned to work on each tool.
    - Planned folder structure with Work in Progress (WIP) and Publish areas.
    - Data types to be passed between departments.
- Pseudocode for each tool to be developed (individually assessed)
- A Git repository, which academic staff will need to been given access to

## Submission - Part 1

Students will present their production schedule, pseudocode and Git repository in class on the due date.

Students will also upload the following to the Canvas ‘Assignments’ page by the due date:

- Their production schedule clearly highlighting which tool each individual will develop and their deliverables (PDF)
- Pseudocode for their tool (Python script)
- A Git repository, which which academic staff will need to been given access to

**Due date: Week 7 / Friday September 22, 2023**
**Weight: 20%**

## Marking Criteria - Part 1

Student submissions will be assessed using the following criteria:

- Production schedule (group assessed) 80%
- Pseudocode (individually assessed) 20%

Students will be marked against the criteria identified in the rubric on Canvas. Please check the rubric carefully and ensure that your submission addresses the criteria appropriately before submitting.

Students will receive written feedback and a letter grade for their work.

Written feedback on Part 1 will be provided two weeks after submission via Canvas, and will indicate performance on each of the criteria. The intention is that this feedback will help identify any areas that need improvement in order to successfully develop their pipeline tool.

## Assessment Task - Part 2: Final Submission

Groups will incorporate the feedback given on Part 1 to their project. They will then complete each tool in the pipeline so that they can present a working version of each tool in class on the due date.

Teaching staff will check on progress every two weeks during development and provide feedback.

Each individual will present the tool they have developed to show that it works in conjunction with the other tools developed by their group.

Groups will also present the following:

- Functional computer code that works with both Unix and Windows file paths
- User documentation for each tool
- Usage of a Git repository showing the frequency and quality of commits for each individual student in the group

## Submission - Part 2

Students will present their code, documentation and Git repository use in class on the due date.

Students will also upload the following to the Canvas ‘Assignments’ page by the due date:
.

- Final code for each tool that works on both Mac and Windows (Python files and project folders)
- User documentation for their tool (PDF)
- Link to the group’s Git repository so that frequency and quality of commits for each student can be checked

**Due date: Week 12 / Friday October 27, 2023**
**Weight: 80%**

## Marking Criteria - Part 2

Student submissions will be assessed using the following criteria:

- Tool functionality 40%
- User documentation 40%
- Git repository use 20%

Students will be marked against the criteria identified in the rubric on Canvas. Please check the rubric carefully and ensure that your submission addresses the criteria appropriately before submitting.

Students will receive written feedback and a letter grade for their work.

## Grading Specification

Based on the above listed criteria, it is possible to receive one of the following grades:

| **Grading Specification / Part 1** |
|---|

| Grade | Abr. | Grading Criteria<br>(Students must achieve a majority of appropriate rankings in all areas to achieve the designated grade) |
|---|---|---|
| **High Distinction** | **H** | • A highly detailed and realistic production schedule is presented. It includes folder structure, data types and schedule with time set aside for code review, tool milestones and weekly production meetings. All elements are presented in detail and the schedule is completely realistic<br>• The pseudocode is functional, efficient and is extremely easy to understand. Git repository extremely well set up |
| **Distinction** | **D** | • A detailed and achievable production schedule and technical plan is presented. It includes folder structure, data types and schedule with time set aside for code review, tool milestones and weekly production meetings. One element may be lacking in detail or be unrealistic to achieve<br>• The pseudocode is functional and easy to understand. Git repository well set up |
| **Credit** | **C** | • A basic and production schedule and technical plan is presented. It includes folder structure, data types and schedule with time set aside for code review, tool milestones and weekly production meetings. One or two elements may be lacking in detail or be unrealistic to achieve<br>• The pseudocode is functional, although one or two aspects may not be clearly expressed. Git repository correctly set up |
| **Pass** | **P** | • A basic production schedule and technical plan is presented but it may be lacking in detail, be unrealistic or not set enough time for key tasks<br>• The pseudocode is minimally functional and may contain some errors and/or is difficult to understand. Git repository set up, although may have some issues not affecting basic functionality |
| **Fail** | **Z** | • The production schedule was not presented or is incoherent<br>• Pseudocode is missing or is dysfunctional. Git repository missing or incorrectly set up |

| **Grading Specification / Part 2** | | |
|---|---|---|
| **Grade** | **Abr.** | **Grading Criteria**<br>(Students must achieve a majority of appropriate rankings in all areas to achieve the designated grade) |
| **High Distinction** | **H** | • The tool is fully functional in all respects. User is able to navigate their way around the tool extremely easily. The user interface is highly efficient and non-obtrusive and has advanced functionality built into it<br>• The user documentation is highly detailed and insightful. Both text and video documentation is clear, error-free and matches the actual tool<br>• The student committed code with high frequency to the repository. Their changes were always clearly and concisely explained in the comments |
| **Distinction** | **D** | • The tool is fully functional. User is able to navigate their way around the tool easily and the user interface is efficient and non-obtrusive<br>• The user documentation is detailed. Both text and video documentation is clear and matches the actual tool. However, it may contain one or two minor errors<br>• The student committed code with regular frequency to the repository. Their changes were almost always clearly and concisely explained in the comments |
| **Credit** | **C** | • The tool is fully functional although it may have some minor bugs. User is able to navigate their way around the tool without significant difficulty<br>• The user documentation is complete. However it may be slightly unclear in places, contain some minor errors or not fully match the actual tool<br>• The student committed code with irregular frequency to the repository |
| **Pass** | **P** | • The tool is functional although some aspects may be incomplete or have bugs. User is able to navigate their way around the tool, but may have some difficulties<br>• The user documentation is minimally complete but may be unclear in places, contain some errors or not fully match the actual tool<br>• The student only committed their final code to the repository |

| | | |
|---|---|---|
| **Fail** | **Z** | • The tool does not work or has major bugs or was not submitted<br>• The user documentation is missing, has major flaws or is incoherent or was not submitted<br>• The student did not contribute to the repository |

All assessment submissions must adhere the the following requirements:

- Submitted artefacts as per each assessment task must include sufficient evidence to demonstrate development and final achievement of each of the Subject Learning objectives (SLOs).
- All work must be of professional presentation, no or few errors such as spelling, grammar, formatting, and accessibility.
- Each submission must each be complete and comprehensive.
- References must use APA (American Psychological Association) referencing style: https://www.lib.uts.edu.au/referencing/apa
- Although the learning management platform Canvas uses rating points as a guide to generating grades, these are indicative only and do not constitute marks.

## Subject Learning Objectives

The criteria for this assessment task map to the following Subject Learning Objectives (note that some assessment criteria may map to more than one Subject Learning Objective).

| **Marking Criteria** | **Subject Learning Objectives (SLOs)** |
|---|---|
| • Production schedule<br>• User documentation | 1. Facilitate improved processes and pipelines for computer graphics projects. |
| • Production schedule<br>• User documentation | 2. Able to identify requirements and prioritise tasks with regard to the production process, team workloads, and social and ethical considerations. |
| • Pseudocode<br>• Tool functionality<br>• Git repository use | 3. Develop and configure animation and visual effects pipeline tools to improve production efficiency |
| • Production schedule<br>• Tool functionality<br>• Git repository use<br>• User documentation | 4. Contribute constructively to collaborative team dynamics and projects, enhancing the overall output through technical direction |