The University of Melbourne
School of Computing and Information Systems
COMP30019 Graphics and Interaction

# Project 2, 2021

Gameplay Video (1-2 minute video): 4pm, 17 October
Final Electronic Submission (project): 4pm, 1 November

Marks: This project counts towards 60% of the marks for this subject

**Participants:** This project should be completed in groups of 3-4 people.

Marking will be the same regardless of the number of participants.

## Aim

The purpose of the project is to expose you to three-dimensional graphics programming and user interfaces. You will develop and evaluate a simple game using Unity. You are free to choose any type of game that your group would like to develop, and we are happy to discuss these possible projects with you before you begin implementation.

## Task

Your task will involve questions of:

1. how to facilitate user interaction,
2. how to create and render objects/entities,
3. how to effectively manage the graphics pipeline so that it runs without substantial lag,
4. how to evaluate your game with participants, and make improvements based on the collected information.

## Git/Github

As with project 1, an important part of this project is not just the submission itself, but the collaborative process. You will use the version control software Git (via Github) to track changes to the project and allow each group member to easily make remote contributions. **It is important that all group members commit their changes to Github from early on in the process.** Furthermore, ensure your repository is kept as small as possible by only committing what is necessary to build and run the project, and taking care with binary assets.

# Specifications and marking criteria

As stated above, we have provided you with a great deal of freedom in what you make. In particular, you are not necessarily required to implement all of the entities and functions present in your intended game if you choose not to. For example, you may choose to only utilise a core subset of functionality. However, the game must still be usable/playable, as well as polished.

- Gameplay (*14 marks*):
    - Instructions are well specified within the game and controls respond as expected
    - The game has a clearly defined objective, with the player being able to progress towards and achieve that objective
    - Gameplay is well executed, bug-free, and operates at a reasonable frame rate (sufficient to play the game)
    - The gameplay and control scheme are polished, easy to use, enjoyable and suit the design of the game
    - A video demo effectively and creatively captures the key gameplay elements (see below for details).

- Graphics (*12 marks*):
    - Objects and entities are clearly visible, and clearly distinguishable.
    - Objects, entities, textures, lighting and user interfaces should suit the style of the game, and there should be consistent aesthetics employed throughout the game
    - Camera orientation, positioning and motion should be comfortable and well-polished.
    - At least one object or game environment should be procedurally generated in a manner that enhances the game's visuals. Give a high level overview of the technique and/or algorithm used in your report.

- Shaders & Special Effects (*16 marks*):
    - At least two clearly distinct custom Cg/HLSL vertex or pixel shaders should be present which appropriately enhance the game's visuals. At least one shader should produce a non-trivial effect **not** explored in the labs. For example, Cel shading, water effects, fog, or other artistic effects as desired. Descriptions of how the shaders work must be clearly detailed in the report, along with the paths to the source files. If you write more than two shaders, specify exactly which two should be marked. It should also be made clear how the use of a shader provides a benefit over an equivalent CPU based approach, if applicable.
    - A particle system should be used to create suitable effects within your game. Clearly specify in your report which particle system should be marked (if there is more than one), and how to locate it in your Unity project.

- Evaluation, Report, Project Organisation (*18 marks*):
    - Use at least one querying technique to evaluate and then improve your game, with a minimum of 5 participants.
    - Use at least one observational method to evaluate and then improve your game, with a minimum of 5 different participants.
    - Overall quality of the delivered report (see below for details). The report should be present in the root directory of the project as a README.md.
    - Github must be correctly used to track changes to the assignment code. The project should be easily downloadable via the given commit ID (see below for details).
    - Each group member should make at least one commit representing a personal contribution to the project.

## Gameplay Video

In order for everyone to be able to check your project in action, you will need to submit a short gameplay video to YouTube demonstrating the key features of your game (1 to 2 minutes long). You can set the video to unlisted if you wish to prevent it from being viewed publicly, but please ensure that we can access it from the provided link without requiring any special permissions.

In particular, ensure that you do not toggle the "Made for kids" option, as this will prevent us from being able to save the videos into a playlist to show in the final two lectures. A video showing how to ensure that your demo is not set as "Made for kids" can be found below:

https://www.youtube.com/watch?v=0AzyGGpWE00

## Report

You must include a report (*README.md*) that describes your application, specifically what it does, how to use it, and how you evaluated and improved it. Several paragraphs of text under each of the following headings should be sufficient:

- Brief explanation of the game,
- How to use it (especially the user interface aspects)
- How you designed objects and entities,
- How you handled the graphics pipeline and camera motion,
- The procedural generation technique and/or algorithm used, including a high level description of the implementation details,
- Descriptions of how the custom shaders work (and which two should be marked),
- A description of the particle system you wish to be marked and how to locate it in your Unity project.

- Description of the querying and observational methods used, including a description of the participants (how many, demographics), description of the methodology (which techniques did you use, what did you have participants do, how did you record the data), and feedback gathered.
- Document the changes made to your game based on the information collected during the evaluation.
- References and external resources that you used
- A description of the contributions made by each member of the group.

## Electronic submission

Your submission *must* open and run in Unity 2021.1.13f1 without modification via a direct download from GitHub. It is highly recommended that you test this regularly while working on the project to ensure that your work is being committed correctly. **One member** of your group must submit a text file to the LMS (Canvas) by the due date which includes the *commit ID* of your final submission. **All group members** should commit changes to Github from the very start of the project. You will not be marked on the content of the commits before the final submission commit, but these may assist in investigating disputes over unequal contributions to the project.

## Plagiarism and attribution

Use of the Unity asset store (and external resources generally) must be strictly limited to artistic assets. In simple terms - images, models, animations, sounds and music tracks are fine, but **not** code, game logic, or non-artistic prefabs. Note that it is possible to get a great mark without using flashy external resources! You are assessed based on what *you* create, and how well you address the criteria in this specification. In other words, producing a fun and well polished game with consistent visuals is much more important than sourcing “AAA game” assets.

Remember that copying code from external sources without attribution is considered academic misconduct. Additionally, we expect that a majority of your submission is your own regardless. You may copy code from the labs, but full credit will not be awarded where there is an over-reliance on external code, even if it is attributed. We will be checking for similarity between submissions and with code available over the Internet.

## Late submissions

Make sure you deliver your work on time. Overdue delivery will result in a reduction of 10% of the marks (6 marks) for each day of delay.