

Assignment →

Coversheet (Individual)



Please print clearly in block letters using a black or blue pen and complete all relevant sections on this form.

Academic honesty

As a JMC Academy student, you are responsible for ensuring any work you submit for assessment appropriately and accurately references the sources of all work you have used in your assignment that are not your own. Academic dishonesty is academic misconduct and can refer to act of plagiarism, collusion, cheating and contract cheating, falsification, fabrication and enabling academic misconduct. You are responsible for reading and understanding JMC Academy's policies and procedures in relation to academic honesty, and the process for investigating and determining act/s of academic misconduct, and what penalties will apply. Refer to your Student Handbook on PLATO and/or the JMC Academy website for more information.

First name: Last name:

Student number:

Course: Acting Animation Audio Engineering Design Certificate III
 Entertainment Film & TV Game Design Music Postgraduate Program in Creative Industries

Campus: Sydney Melbourne Brisbane

Assessment Details

Certificate III students Course code: Course name:

Higher Ed. students Unit code: Unit name:

Assessment number: Assessment name:

Lecturer name: Due date:

Assessment type

Essay Storyboard/Plans Analysis
 Report Folio Written assignment
 Presentation Research paper Schedule
 Recording Journal Rationale
 Logbook/Workbook Proposal Media
 Quiz Other (please specify):

Are you submitting your assessment in multiple parts?
If yes, please specify part number:

Yes parts No

Please tick appropriate

First submission Approved extension Approved submission

Please attach evidence if this is an approved extension or resubmission.

Student declaration

- I have read and understood the Academic Integrity Policy and Procedure
 - I understand the consequences of committing an act of academic dishonesty
 - I understand that JMC Academy will investigate any alleged act of academic dishonesty, and I confirm I will participate fully in that process
- I understand a copy of my assignment may be used by JMC Academy as part of the cycle of moderation
- I declare that this assignment is all my group's own work, and that we have appropriately and accurately referenced sources of all material that is not our own. I declare that I have not plagiarised, colluded, cheated, contract cheated, falsified, fabricated any of the work presented in this assignment

Date:

Office use only

Date received:

Received by:

Rapid Prototyping A1 Report

Marceline Lentz

Overview

I chose to follow Unit 5 (UI) and created a *Fruit Ninja* style clicker game. The player can select between three difficulty levels, and must click on spawning fruits before they fall out of frame. There are also bombs, which will deduct (-10) points if clicked, but will be rewarded points (+20) for fruits. The game will end if a fruit falls out of frame without being clicked.

The additional feature I chose to add was a pop-up score that appears when a fruit (or bomb) is clicked to make the adding and subtracting of score clearer to the player, as I felt this was not quite clear enough in the original prototype. Additionally, it creates a more fun visual experience.

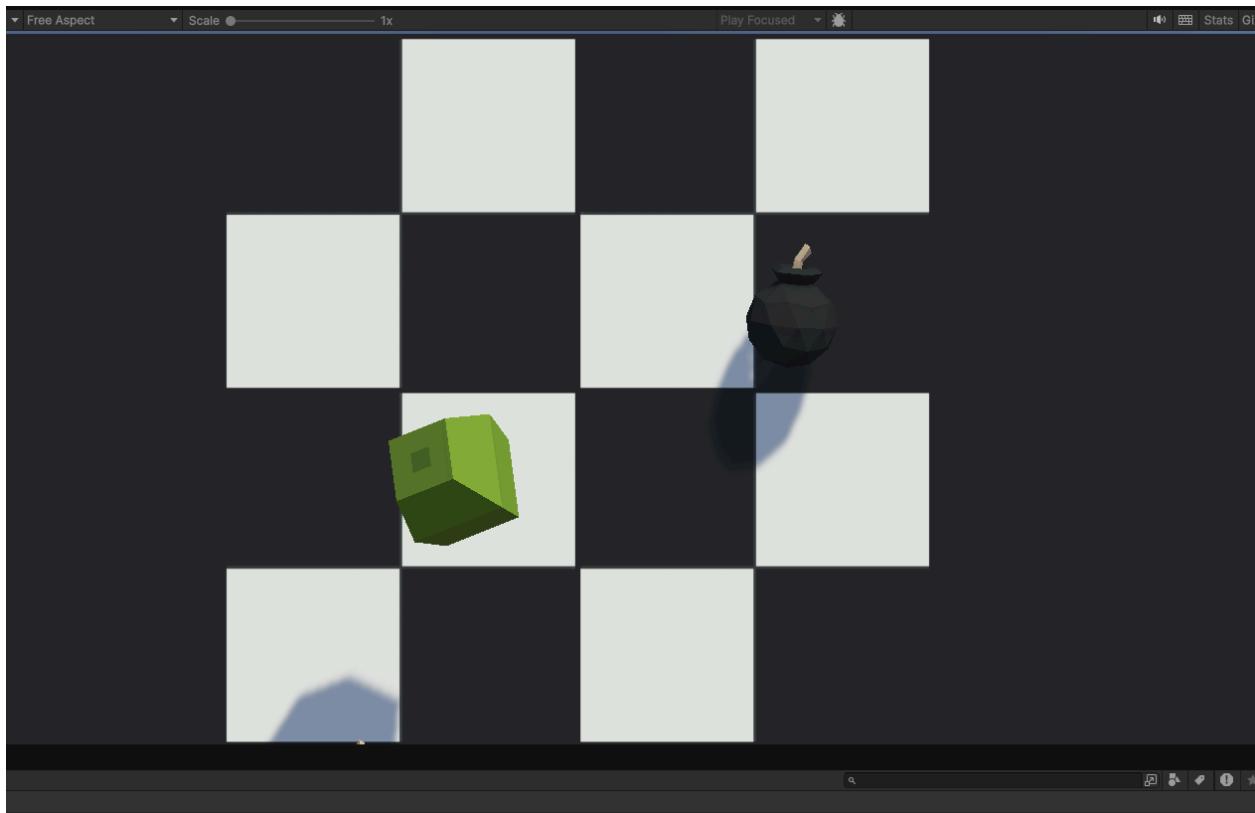
Progress Reports

30/09/2025

Adjustments:

- Adjusted torque rotation values from tutorial as they had crazy rotation in the scene
- Troubleshooting the ver of unity being incompatible with some of the code in the tutorial (OnMouseDown fun c not working), had to adjust unity settings to allow older version functions

Progress: spawns & launches randomised items into the air which are destroyed on mouse click & also destroyed when they hit a collider below the screen

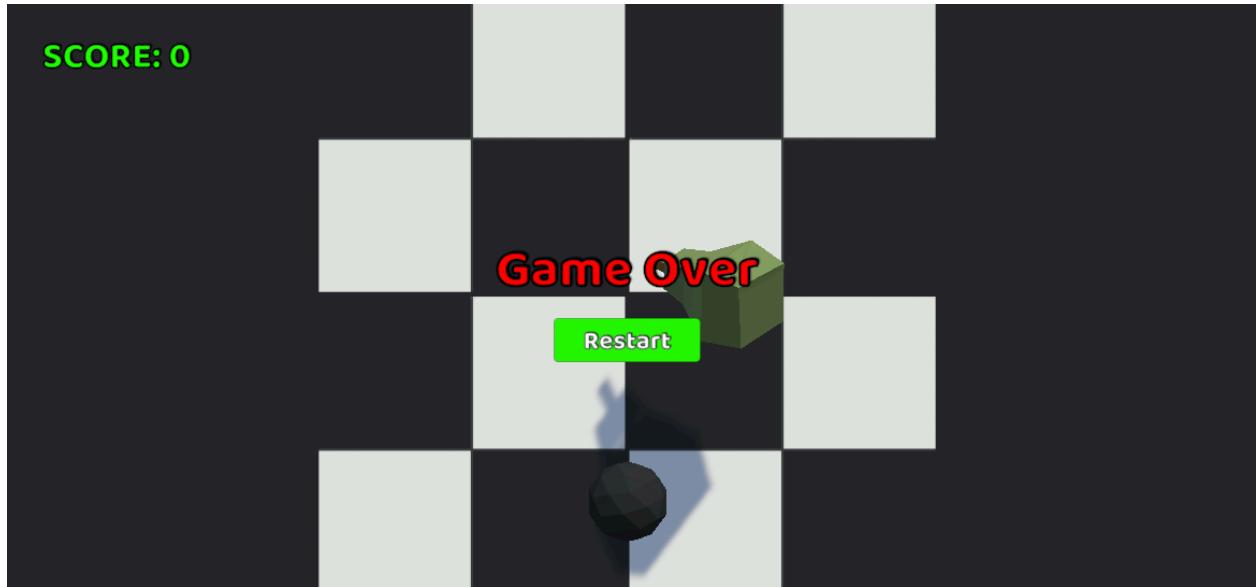


07/10/2025

Adjustments:

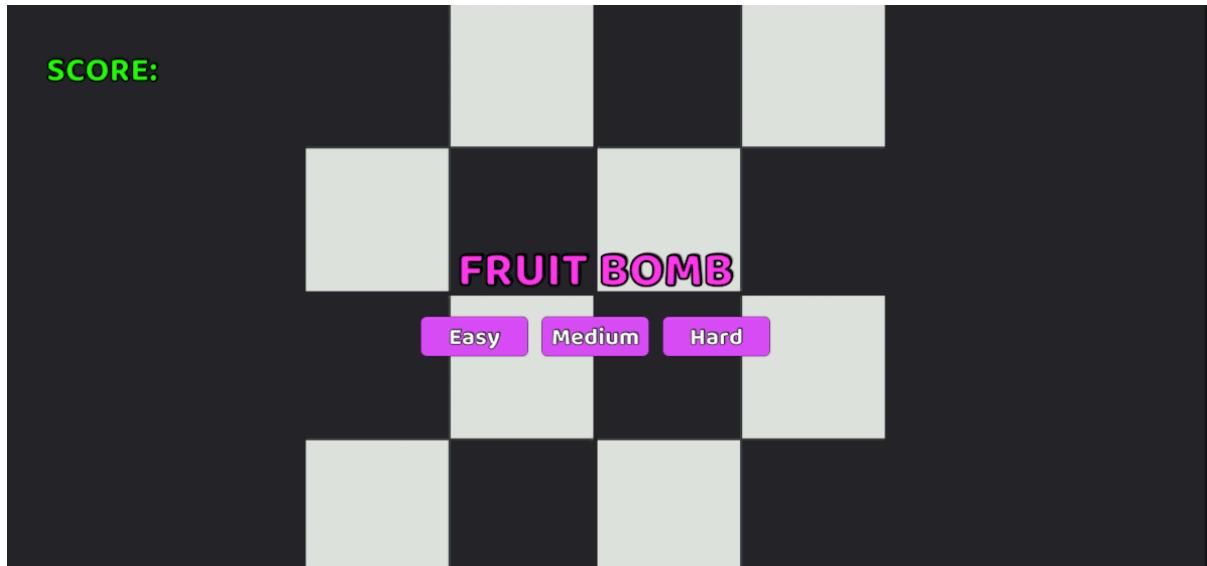
- Adjusted score values
- Changed the particle colours to match the items (appears more cohesive)

Progress: Added score, adds score for “good” times and takes away for “bad” items (bombs), added particles animation on destroy, added game over screen & restart button, items can no longer be destroyed after game over



11/10/2025

Progress: added title screen & difficulty buttons, added difficulty measurements on button press

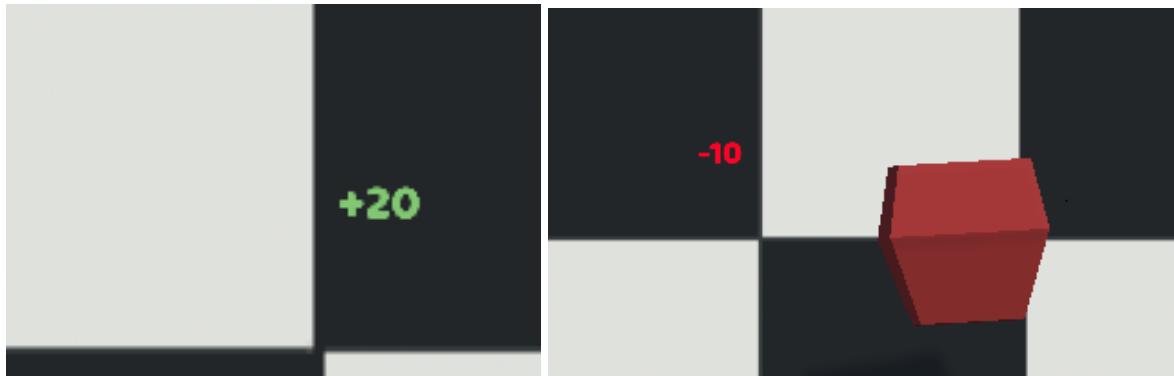


12/10/2025

Adjustments:

- Adjusted speed & ySpawn values to work better – i felt that a number of the fruits would spawn so low that they could not be clicked before falling out of frame so I made the y spawn a little higher to have this be less of an issue

Progress: added score pop up feature (explained below) & completed the tutorial challenge



struggled to get a good clear picture as the text disappeared faster than the screen capture would load

Added Feature

The feature I chose to add was score pop-ups that appeared when the fruit was destroyed, as I felt just having the score off to the side wasn't clear enough, in particular, I didn't think it was clear enough that the bombs were "bad" or that the score was being taken away at all.

I created a new text prefab and created an Instantiate method inside the target script on the fruits (and bomb) prefabs that would spawn the text when the object was pressed. To make this cleaner, I added a script to the text that would make it disappear after a short period of time, as well as a simple scale animation that would make it scale down over time too.

I had to do a bit of troubleshooting during this process as I struggled to get the text to spawn correctly at the position of the target and not just randomly off to the side, but I solved this by making the text just a regular TextMesh instead of TextMeshPro, which fixed the issue.

After I had created the prefab with all the code and animations, I then duplicated it and changed the text to say -10 instead of +20 and the colour from green to red and changed the spawner on the bomb object to this new prefab. Now, when the bombs were pressed it would show that score was being taken away, and when the fruits were pressed it would show the score being added, making it super clear to the player which were good and bad items, and how much was being added and taken away for each. Doing this *after* adding all the animations and code meant that I didn't have to do everything twice, and kept everything consistent between the two prefabs, only editing the TextMesh features.