Technical Report: Ricochet Game

File Structure:

# All assets and resources are stored into folders coinciding with their file types. So all the scripts are all stored in Scripts, materials in Materials, etc.

Code Conventions:

# All variables were named with a high level of verbosity (thats a word apparently). This was to allow anyone viewing and reading the code to be able to quickly understand what each variable was used for with a single glance.

# The Capture Team Flag Gamemode

The modular coding / component based design:

Modular coding enables the part to function independently without changing and relying on any other script, thus better connecting with the tasks of other people in a group project. The CTF gamemode is considered a main component of the project. There are small components that forms this main component. Each component is responsible for a specific task.

For instance in the project:

Script “CarryFlagController” only manipulates the flag’s position according to its carrier’s position. This single-function script can be reimported and reused in future projects without a lot of modifications.0.

The use of events:

The use of unity events in the CTF gamemode supports the component based code structure by announcing events rather than referencing methods in other components. Methods or functions in other components can listen to the events announced and react.

For instance in the project:

An event (onFlagInRightTeamBase) is called every time a flag was successfully delivered to the right team base. Multiple methods/functions in the program listen to this event and act on call, including scoring method, UI announcer.

Using events to stage events in gameplay allowed for an easy way of viewing all the actions that will happen when said event is invoked in the inspector.

File Naming Convention

Adding prefix makes search and arrangement simple when looking for a certain type of asset.

Ex: a sprite for an apple would be named “spr\_apple”, a prefab of an enemy would be named “pre\_enemy”, a material for stone would be named “mat\_stone”.

The Application of Singletons:

The use of singletons in game managers:

An abstract singleton base class was used for all the managers in the project to ensure anything labeled as a manager would be able to be referenced exactly the same as each other. This helped a lot for referencing managers easily and cleanly in code.

Scriptable Objects: The weapons

Using scriptable objects to create various types of weapons allowed for these to be quickly created or edited/tweaked.