**Remember only around 2 minuets to look at portfolio**

Needed when discussing project

**What**

* What is the work?

Show it through download, source code, video, images etc

**Why**

* What project is trying to demonstrate
* What intended learnt by doing it
* Challenges and how they were overcome

**How**

* How it was actually done, i.e the source code

Projects

**Island Survival**

Ue4 C++ Multiplayer Procedural Terrain Student Work(Group of 2)

Video + What I worked on

Created as a second-year project, where any AI or PCG system could be implemented, this tech demo

                                has players collect fuel for their boat and evade zombies on some randomly generated islands.

                                The terrain system was implemented with the idea of its future use within

                                many other projects so that their development times can be significantly reduced. Due to this all elements

                                have been designed to be highly customisable and easily removable based on the project’s needs.

                                The added challenge of

                                island generation was implemented to allow for a more natural border to the game world.

as terrain is a large part of many games, so being able to quickly generate it at a click of a button can greatly speed up production, especially in the prototyping phase.

**Raindrop**

Created as a second-year project, this sees the player, the sun, fall in love with a raindrop, gain its trust and guide it up through various stages from the lake, all the way up to space. Being made with the two themes *The life of a raindrop* and *trust and betrayal* at its core the aim was to make an enjoyable game which followed good design principles. As the main programmer I endeavoured to ensure that the elements developed were highly modular and easy to use across all the various levels of the game.

**Pacman**

Unity v2019.3.3 C# Gameplay AI Student Work

Video

This first year project, developed largely between **August 2020 -** **October 2020(10 Weeks)** was undertaken to learn the fundamentals of game development. Tasked with recreating the entirety of the game Pac-Man with a distinctive style as well as creating a unique design innovation. Here I challenged myself by implementing an entire procedural maze system, which works around the constraints of the Pac-Man level design.

**Showcased at the UTS 2020 Spring Showcase**

Through implementing the well-known game Pacman, the project was undertaken to learn the fundamentals of game development within the Unity engine. Created as a first year project between **August 2020 -** **October 2020(10 Weeks)** the first **6** **weeks** involved creating the

The last **4 weeks** involved the implementation of the various systems such as enemy AI, player movement etc

Throughout this all sprites, animations, particles and code were created by me.

Some features to highlight are:

Sheep AI

Four types of AI exist:

* 1st Sheep(blue): move away from player
* 2nd Sheep(orange): move towards player
* 3rd Sheep(pink): move in random direction
* 4th Sheep(red): follow map border in clockwise direction

One notable challenge here was the 4th Sheep as the waypoint system used had to be easily implementable within the maze generation system. This was solved by making each green tile act as a waypoint with the sheep mealy finding and navigating between each in turn, in a clockwise direction.

Accurate navigation was also a substantial challenge, especially on a large map so an efficient **A\* pathfinding** system was implemented to give accurate results.

This was achieved by making the green border tiles themselves act as the waypoint system, with them individually storing the neighbours tile location, within a clockwise direction.

Procedural Maze Generation

* Generates maze border
* Spawns in centre piece and teleports
* Uses a xxx backtracking method to generate the maze

This was a particularly challenging system to implement as all inside walls needed to be at least 2-thick requiring a number of extra conditions to be determined and checked against.

* Spawn in correct map tiles with appropriate orientation

Other important features

* Character / AI movement via tweening
* Score / Lives System
* Saving / Loading via XXXXXXX
* Minimap system

Two

The maze generation system was implemented challenge myself beyond the core essentials of the assignment by gaining an understanding of how maze generation works. A core challenge was implementing the appropriate wall tile sprite in the right location and the fact that all inside walls needed to be a minimum of 2 tiles thick.

Play in browser here

See source code on github

**Island Survival**

**Raindrop**

**FEIT Prototype Exhibition**

**- Spring 2021**

**FEIT Prototype Exhibition**

**- Spring 2021**