# Modulizer

Modulizer is a game made by myself and two others for the Ludum Dare 45 game jam. The theme for this game jam was “Start with nothing”. This gave us the idea to create a roguelike style game as this would fit in with the theme precisely as the player would start with nothing. We also wanted to make the game somewhat unique so we decided to give the player modules that they could attach to their body, these modules could provide a weapon, increased damage or increased health. We also limited the players upgrade potential by only allowing the player to equip four upgrades, one for each side of the players body. This gave the game some balance however if the game was continued it would have been a good idea to add extension modules that other modules could attach to.

For this game I was mainly tasked with creating and implementing audio. As this was one of the first games that I worked on I was limited in the amount that I was able to do however, I believe I managed to learn a lot about game development from working on this game.

# Pacman

# Robots Must Die

This was a group game made for my universities “Games Development” module. This project was one of my first attempts at making a game as part of a team.

This game was made within Unity with my main role being the programming for the enemies. The enemies used simple finite state machines for the AI as they didn’t need any complex functionality. The main feature that the enemies had were some of their animations. A number of the enemies in the game used procedurally animated legs through the use of inverse kinematics. By utilising inverse kinematics, it was possible to create smoothly animated legs with very little effort, this gave the enemies a unique and creepy movement style which fit in with the theme of the game very nicely.

I also had a number of miscellaneous tasks that I had to work on, such as creating simple shaders, adding some functionality to weapons, creating an exp system and implementing a number of skills. Most of these tasks were very simple and didn’t require a lot of effort on my part, however certain skills required additional work. One of the skills that wasn’t very simple was making the energy gun shoot additional projectiles. The idea behind this skill was that the gun would shoot three projectiles in a row with two of the projectiles being at 30-degree angles. The difficulty with this was ensuring that the projectile would shoot at a 30-degree angle based on the players forward direction, however after utilising some simple math it became relatively straight forward.

Overall, this project worked very well and we were able to create, what I believe to be, a very good game in a short amount of time. The main downfall of the game is the balancing. The game becomes very difficult very quickly which can be a deterrent for players.

# Group RPG

This was a group game made for my universities “Technical Games Production” module. For this project we were tasked with designing and creating a game using methods of our choosing. We very quickly decided on creating an RPG with inspirations from TES:V Skyrim using Unity.

My main role on this project was creating enemies. This meant finding models, adding animations to the models and creating the AI to get the enemies to work. Finding and animating the models was very easy as we could use the Unity asset store and Mixamo to easily get models and animations. For the AI I decided to create a very simple behaviour tree style system. The system worked by creating tasks and using them in a script. These tasks accessed variables by using a blackboard, the blackboard stored a number of dictionaries with strings as the keys and the actual value as the value. This made it easy to get any necessary data as each task can have access to the base blackboard class and get values from the specific blackboard that it needs data from. This system wasn’t perfect, there were no visual controls to create and edit the behaviour trees which makes it somewhat difficult to use, however, for this project it worked very well and did exactly what we needed it to.

During the development of this project, we had some difficulties with the team, one of our team members left around halfway through the project which meant we had some incomplete systems that we weren’t completely familiar with. This wasn’t a big issue as the systems were easy to understand and made reasonably well. As well as this, a few members of the group were somewhat lazy when working on the project and didn’t manage to get a lot done. This affected the development of the project a decent amount as we didn’t get some of the systems that we needed working until relatively late in the project however, in the end, we managed to get the project to a good state with a good number of features working properly.

Overall, the project was a good success as it taught me a lot about working in teams and the difficulties that can come with it.

# Unity FPS

This FPS game was made for my universities “Game Engine Programming” module. This module involved using two different game engines, Unity and UE4, to create games and introduce you to the game engines. I decided to create an FPS game with some advanced movement mechanics as well as procedurally animated enemies.

The movement was also physics based as I wanted the movement to feel more fluid. As well as this, it made it easy to add other movement systems. The more advanced parts of the movement system were wall running and grappling. The process for wall running was fairly simple, I did a raycast to each side of the player to see if there was a wall in range, if there was a small force would be applied in the direction of the wall and gravity would be reduced. Grappling was also very simple, I just used a spring joint component to link both objects, this made sure that one object couldn’t move to far from the other, it also gave the connection spring like properties which made it feel like the player was bouncing if they were too far away. Finally, I used a line renderer to draw a line from the player to the grapple point.

To procedurally animate the enemies I created legs that used inverse kinematics. The leg would stay attached to the enemy on one side and the floor on the other. This was a very simple implementation so the overall look of the movement was very good however it worked well for this project. I also checked the height of the main body of the enemy and if it was too low I added a small force to move it up, this made it look as if the enemy was bouncing.

Overall, this project was a reasonable success and helped me learn a lot about making games in unity.

# UE4 FPS

This FPS game was made for my universities “Game Engine Programming” module. This module involved using two different game engines, Unity and UE4, to create games and introduce you to the game engines. My initial idea for the game was to create a simple idle game, I then took inspiration from games like Time Warpers and Aim Labs to create and Idle Aim trainer type game.

The game featured a variety of weapons, upgrades as well as an AI drone that would assist the player. The weapons could be bought in the world, most weapons had some unique properties. For example, the player starts with a simple pistol that isn’t very effective at the start, but they can then upgrade to a shotgun, automatic rifle or grenade launcher as they progress through the game. Each weapon used the same energy resource that the player had however each weapon would use a different amount.

There were a large number of upgrades. Each weapon had their own set of upgrades that could be purchased that could upgrade their damage, fire rate or other unique properties. You could also upgrade the player and enemies. Upgrading the player would increase health, energy or provide passive income while upgrading the enemies would increase the number and frequency that the enemies spawn. Finally, the player could upgrade drones, this would increase the number of the drones as well as the movement speed, damage and fire rate of the drones.

Drones are small flying spheres that will fly to an enemy and shoot it, once the enemy has died the drone will return to the player and follow them until they are able to shoot again.

Overall, this project was a good success and helped me learn a lot about making games in unreal engine.

# Shape Tower Defence