**s014088m C++ For Engines Technical Design Doc  
The Plan Part 1**

**The Idea**The idea of this game is going to be a platformer with a speedrun element, each level is going to have a set time that the player has to complete it within, medals will be awarded for time completion and trophies for score in a level.   
Gold – Fastest time - 3rd Trophy – Hidden Collectable  
Silver – Faster Time - 2nd Trophy – Deaths  
Bronze – Fast Time - 1st Trophy – Score Threshold

The player will use their movement mechanics to traverse the environment to press buttons, collect items and unlock the final door to progress to the next stage. Stopping the player from exiting the room will be various AI that charge, shoot, and chase the player to stop them from achieving their goal. This level-based progression was inspired by portal having levelled systems and the movement mechanics of Titanfall & Ghost runner for the mechanics I want to implement.

**New Movement mechanics**

The Dash  
- Dash in any direction  
- cooldown ability  
- Limited Usage  
- Can be chained with other dashes movement techniques.

The Slide  
- Slide under gaps that a running character cannot reach  
- Requires Speed  
- Can be used anytime  
- Once slowed if the crouch key is held it will remain crouched  
- If key let go whilst under slide it should remain crouched.

Wall Run  
- Must have certain speed to achieve wall grab  
- Its automatic, no key presses needed  
- Can only hold on for 5 seconds until the player needs to swap walls  
- Can only hold 3 walls consecutively without needing a floor to reset  
- If the player doesn’t jump as they get off a wall there will be no upward velocity as they leave the wall. A screenshot of a computer

Description automatically generated

**Enemies**

Patrol AI  
- Walks Around and Looks for the player  
- Specific route it follows  
- Spots player visually  
- Rotates towards the player and runs at them  
- If they are already at the player, it will attack them.

A diagram of a game

Description automatically generated

Run & Explode AI  
A diagram of a game

Description automatically generated

Automated Turret

Automated Follow Turret

**Weapons**

General - (Data Table Information)  
Using a Data Table will allow for me to create many guns with minimal requirement of creating classes, this allows me to spawn either a hit scan weapon or a projectile then have any data inserted via data table as to save on memory, this also centrally stores all the data stats for each gun which can help with balancing the weapons within excel or the editor.

- Audio  
- Particle Spawns on Shoot & Impact  
- Meshes  
- Display HUD stats  
- Damage  
- Reload Speed  
- Ammo Clip Size  
- Time Between Shots  
- Projectile Type  
- Range

Rockets – A new Projectile class  
 - Explosion on impact  
 - Distance from explosion impact scales damage down  
 - Applies forces at location for extra gameplay uses (Rocket jumping)

**Environment**

Doors

* Proximity – Opens when the player gets close to it.
* Key – Required item to open the unlock.
* Passcode – Input a pin number Using UI Widgets to unlock.
* Button / Lever – Opens the door when activated (Calls OnOpenDoor)

A diagram of a software flowchart

Description automatically generated

Jump Pad  
- Only Launches the Player  
- Infinite uses but a cooldown between uses.  
- Specific direction that it will launch the player (if its angled it will shoot them more horizontally than vertically)

Damage Volumes  
- Launches player to give visual feedback as well as the HUD flashing red  
- Damages any actor with the health component.

A diagram of a launchpad

Description automatically generated

**User Interface**

Interact Widgets – Pops up when the player has an interactable they can interact with

Ammo Counters – Updates every time the player fires, reloads and picks up ammo.

Score Counters – Updates every time an enemy is killed, or another scoring item has been hit.

Timer – Updates every second to allow the player to see the current time.

A diagram of a software

Description automatically generated