**s014088m  
C++ For Engines Technical Design Doc  
Game Principle**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 alongside different trap types. This level-based progression was inspired by portal having levelled systems that have dangers to the player and the movement mechanics of Titanfall & Ghost runner for the movement mechanics.

**Movement mechanics.**   
Planning DiagramA screenshot of a computer

Description automatically generated **Dash**  
- Dash in any direction  
- On a cooldown ability, once landed the cooldown should start  
- 2 Can be used in succession if the player has charged both cooldowns  
- Can be chained in succession with other dashes and movement mechanics.

A diagram of a person's car

Description automatically generated

A designer should be able to tweak these values and even be able to create varying dash speeds by changing the dash force/length and tweaking the number of dashes that a user has. This will allow for the user to move more freely within the level allowing for flowing gameplay and a “feel good” environment traversal tool.

**The Slide / Crouch**  
- Slide under gaps that a standing character cannot walk under  
- Requires Speed to slide under an object  
- Can be used anytime, if the player has enough speed it will be a slide and if the player doesn’t it will be a crouch  
- Once slowed if the crouch key is held it will remain crouched  
- If the crouch key let go whilst sliding under an object it should remain crouched.

A diagram of a mechanical scheme

Description automatically generated with medium confidence

A designer can tweak values to adjust speed clamps and crouch height which allows for more options, this traversal mechanic will have players increasing their speed on slopes and ducking under objects, this will have the player feeling that their movement is fluid and also gives more opportunities to decide their chosen path through the level.

**Wall Run**  
- Must have certain speed to achieve wall grab  
- Its automatic, only a jump key hold is required and once the player gets close to the wall it will automatically grab onto it.   
- The player will remain on the wall until the jump key is released or they have reached a non-grabbable wall  
- Can only hold 3 walls consecutively without needing a floor to reset  
- If the player doesn’t release the jump key as they get off a wall there will be no upward velocity as they leave the wall.

A diagram of a sun mechanic

Description automatically generated

Designers shall be able to create any piece of environment Wall Run-able by giving it the tag “Wall Run”, I have implemented a design pattern which has it as the blue walls that you can grab onto, it is expected for the designers to continue with this pattern. This mechanic will be an important one as it has large uses in the game, so it must feel good which is why a camera tilt and speed VFX will be implemented to give it some impact.**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 screenshot of a computer screen

Description automatically generated

Designers can increase the perceptiveness of AI to allow them to be adjusted for their environments on a per instance basis, this will allow for the player to have a more challenged gameplay experience if they so wished, it can also mean that designers can use these values to scale based on difficulty.

**Behaviour Tree Planning**A diagram of a game

Description automatically generated

Automated Turret  
- Gets player position(s) when in sight  
- Rotates towards the player position  
- Fires a projectile  
- Has a toggle to enable and disable movement  
- Projectile class is changeable with those of the same subclass.  
- Timer Loops the firing  
- When the player leaves zone it can remain firing or cancel dependant on a Boolean.

**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 game

Description automatically generated

[UML Diagram](https://mermaid.live/edit#pako:eNqtV21v4jgQ_itRPvVOUBXoC0WrlbggCqfluoLunXTHqjLJFHxy7MhxynIs__3GcQOJ46CutPmQl3nGM-NnxmNn74ciAn_gh4yk6YiStSTxki-5h9cweiU8hGgmXiEGrgIRJ4Ljy4fv7fbHBaMRHEXvGfEXYWye8R8aMyLpxhpg7miIg3wgMUx5qrSFD-12STrDedmaWqa1PjOyAxkIrqRgDGQxYVuO_idfRhVsQyQJFUgdXGPkTfbyGVmGnNY1WUDQWBM65fjEDyEtcgqArNgpOe329_eNeFz9C6Eq2a-YKdNv7nnVOCzsCwYOxYvRrCdtb2B9vTBBlPcbpOqJxnDxi408CUV0_dgo5coLBOXp86fO82eQ4TFUC-1W0cbYdJmU4noVNMI5hhJICtq5LHvP0RE4UMs-VpJt9EsSEQV5dDWTBtPlD03gBAhTmxq42IhtkZILZyLdAdrVWop2ypNMDUNFBfceJFmZVzeuQz6H533jnMKYyrN4MaVCx5p-gF1GdxodZ40bDeYBOBEduhM40ukCdbwNGTKpNeX7B2z1ZyAyTIB3LqFHffM9JzjPs-XRMMCZZdNX9tUFMiIxWcNJNoN086apX2scO6c8BwwjqomHSpGwTutIiqS0UmplWDS7WoPQ3XaRAEQ2gAvYkpsZeEEmJVZ90VGtQB55Y7UYCO06EGetaF6Mm4Ym4OjCpQkOpfxnqMFfv1aaajrlc8LLCTJa3gOoEaToNGqs0GEUlU1dmJH13MVI6zsUK6vBrVLpPwV4aNowXDyshGDeKiC8UKw146p1i-TGfXn_81Nvry7riFPyOJ5QNYc0Y-pYkKhcXYda8nsWJ2mDGNNEKKd8bXE1TU289pJ4Cwdn-ErVbhESBi4VbXsCdL1RTeiImu26vHpyz5ikJqKwODU0FnJLZPQn6GK5WM21Gy2vqWMatNyxueYdZCxF7MTneC5Q8LjF3ftJaFfpm8uGHFUPrrX-ksNoIaxRNaM8B53dJwdFAkO-rpPcbDOQIgs3Bfn2SlJEqtp2NTZUan7HTAg55S8sA0xOfe0nCdudfDs6dpSFkCvMsC5pwmjj8aVyED-Slp-uTDWbvagMzMi3qtBMWcsqXJzEAZZUJLb5Ea-KFl_5CWtCeMTAe3oujyk0iqepnUqXLiq2Kj1RXZg6YY_8E_oCLCUPm4mUkhZ_FQe_5WMouB4j_H3KCVn6aoMNZ-kP8DWCF4KcLv0l16okU2Kx46E_eCEshZafmQ3c_HEdpQnh_mDvf_MH3X7nsnPT7d12urdX_evbTsvf-YN297J7d3933-t2rjq9685N_-7Q8v8TAk1cXd7r67rfu7q-v-v1ezctHyKKpTJ7-8fTj9zH3_kAJTM4_A_OwXf5)

A diagram of a computer

Description automatically generated with medium confidence