Problems Faced and Solutions cw1 4008

Throughout the development of *Escape Velocity*, I encountered a fair share of challenges, each of which pushed me to grow as a developer. Here's a breakdown of the most notable hurdles I faced and the solutions I implemented to overcome them:

1. **Hitting Enemies:**
   Early on, I noticed that projectiles weren't consistently connecting with enemies, which made gameplay frustrating. The issue was precision, enemy hitboxes were too small compared to their visuals. I fixed this by slightly enlarging the hitboxes beyond the enemy surfaces. This simple adjustment greatly improved accuracy and made combat feel more rewarding.

2. **Falling Off the Edge:**
   My player character occasionally fell through gaps in places it shouldn't. To solve this, I created an empty game object with a collider at the base of these areas, effectively preventing unintended falls. This was a simple yet effective fix that kept gameplay smooth.

3. **Finding and Using Assets:**
   Discovering free assets was a game-changer. Not only did it save time, but it also opened up new avenues for creativity. Some high-quality textures, models, and materials are ready made and seeing some of them even sparked fresh ideas. Plus, let's be honest, "free" is always a win!

4. **Prefab Edits Affecting All Instances:**
   One problem with prefabs was wanting to change a single instance without affecting the others. Initially, I didn't realise I had to override the prefab after making edits. Once I understood this, I re-prefabbed specific elements and then executed overrides as needed. A small learning curve, but worth it.

5. **Background Misalignment:**
   The game's background didn't sit right at first, which made the visuals feel awkward. I solved this by flattening background effects and images onto a plane mesh and projecting them as matte textures. This allowed the backgrounds to look 3D but remain visually cohesive in 2D space.

6. **Enemies Following the Player:**
   I struggled to get enemies to properly follow the player without moving in unpredictable ways. I ended up using both a Navmesh for pathfinding and a follow-object script for tracking. It worked well enough for now, though I'm sure there's a better way I'll discover as my skills grow.(revisited, the follow player broke near the end of the development cycle, noteworthy)

7. **Textures and Materials:**
   Getting good textures and materials was initially daunting. I leaned on free assets for this, as creating shaders is still a bit advanced for me. I decided to focus on nailing the basics before diving into the more intricate aspects of game visuals.

8. **Monster Chase Mechanic:**
   Making the Revenant chase the player presented a unique challenge. I experimented with various solutions like navmeshes and follow scripts but found the simplest and most effective fix was attaching the monster to the camera. With the camera in constant pan, the monster appears to be chasing the player while remaining locked in place on screen. To achieve this as best I could, I created a game object camera name cameraman, this helped me visualise where a camera can be vs where its looking.

9. **Seamlessly Connecting Blocks:**
   Aligning separate block objects to create seamless level paths was surprisingly easy once I discovered the *Ctrl + V* trick. It's a small detail, but it made building levels so much faster and more polished.

10. **Bullet Collisions:**
    Correcting bullet sizes and their interactions with the player was tricky at first. I had to adjust collider sizes, mass, drag, physics properties, and firing positions. I also found that starting the collider halfway into the bullet object improved accuracy.

11. **Scripts for Core Mechanics:**
    Writing scripts for damage control, object destruction, and instantiation feels like the bread-and-butter game development. Initially, I tried a small loop to optimise functionality, but as I'm working between Python and C# (Py#?), some things didn't translate as smoothly. Still, I persevered and grew more confident with each script I wrote.

12. **Polishing Gameplay:**
    Now that the basic mechanics are in place, I've been focused on refining the gameplay as much as my current skill level allows. I'm also working on adding menus for the start and end screens, along with intro and outro cutscenes to tie the game together narratively.

Reflection:
Each challenge has been an opportunity to learn and improve. What's exciting is how this process has fuelled my creativity—every fix and feature unlocks new possibilities.

With every step forward, I gain a deeper understanding of game development, and I feel increasingly proud of what we have created. Developing games is like crafting living, interactive art. It’s almost… godlike.