Game Feel Implementation

Decreased Accuracy

To implement random decreased accuracy for the projectiles of the game I plan to randomly affect the starting direction of the projectiles by a number between two values (ie -1 and 1) this means the projectiles with start with a random direction between to values based on where the projectile was intended to go and using variables the system can be changed based on other factors as well as changed to balance the game.

Permanence

Instead of destroying the enemies, chests, and other entities I can instead have a deactivated sprite parented to the main entity game object that, on death, is activated, while the main sprite and all controlling scripts are deactivated. One problem with this system might be optimization as instead of the entities being destroyed they will stay within the world. To fix this I could make a cleanup system that goes through and deletes each entity that is no longer accessible by the player, I could also set a limit on the amount of dead entities the game allows however I don't feel that we would need this as apart from the dungeon generation, which could be done during a loading screen when the player isn't playing, the game is not inherently resource intensive.

Linear Camera Interpolation

To implement this I could use the very easy method of making the camera move towards where the player is, but only allow it to move a certain distance every frame. The only times there would be a problem with this method is if the player moves a great distance in one frame where the camera would take a long time to catch up, however due to the nature of the game I don't think this will happen. If we wanted to have the camera center or show something else other than the player we could take the point of interest and find the midpoint between the player and the point of interest and move the camera there, making sure not to have the player leave the view.

Screen Shake

There is two ways to implement screen shake that I feel we could use, one would be to animate the camera shaking and play that back, this is a very easy way to create a screen shake however one problem would be that it is the same motion for every shake, this may be fine but we also may find that it looks too “artificial” this is where the second method shines. The second method would be to pick random points a set distance around the cameras current position and move the camera between them. This is a lot more complicated but allows for us to change the shake intensity and time that the screen shakes for dynamically. Which might lead to a more organic feeling shake.

Pause on Impact and Slow Motion

Pauses and slow motion can be easily implemented using unity's built in time control ([Time](https://docs.unity3d.com/ScriptReference/Time.html).timeScale), this variable allows you to control the speed in which time passes, this includes physics calculations and script execution but only if they are frame rate independent. This could be a problem however I already planned to have all my methods frame rate independent as this allows the game to run exactly the same on differently performing computers.