Optimization 1: Cam shake update calls.

Instead of update in CamShake script, call a function when you want to do camera shake.

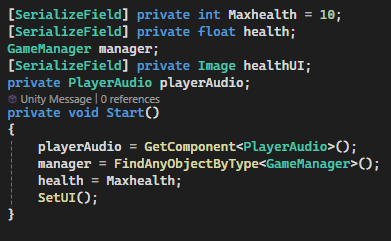
Additionally, remove the update in the PlayerMovement script for collision. Instead, use a collision function like OnCollisionEnter2D, which will only fire when a collision occurs. In here, you can check the layer, or you could also set the collider itself to only detect a certain layer, thereby eliminating any need to check what you collided with. Then, you can call the cam shake code. Also, rather than Find each time to play cam shake, do it on awake, or better yet, serialize the field and set it in the inspector.

A screen shot of a computer program

AI-generated content may be incorrect.

Optimization 2: Player health script find

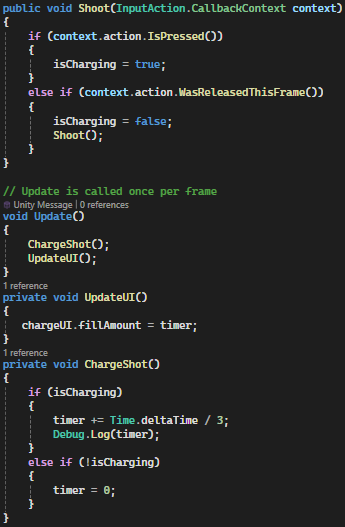
Can serialize field for PlayerAudio and GameManager, and then assign them in editor.



Optimization 3: Reduce update calls in PlayerShoot script

Instead of constantly checking charge shot, call the function in the Shoot function. When action is pressed, update the charge timer rather than constantly checking if it is being pressed.

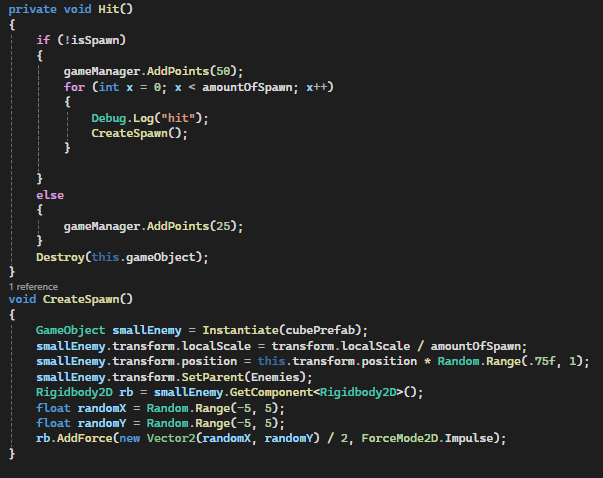
Additionally, can update the UI when charging rather than in the update function. For example, call UpdateUI in the ChargeShot func to update it only when things are happening with the charge shot.



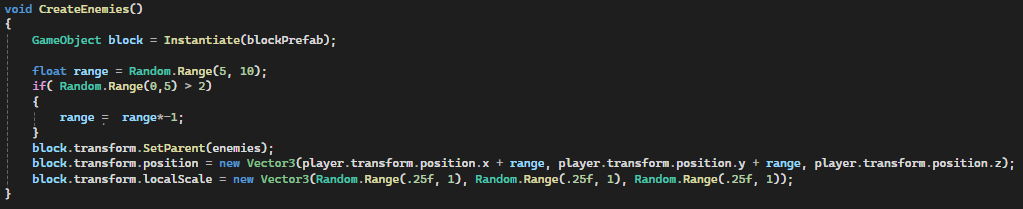
Optimization 4: Object pooling

Instead of destroying the object when done with it, and then making a new one when spawning one in, use object pooling. This can be applied to both the enemies and player bullets.

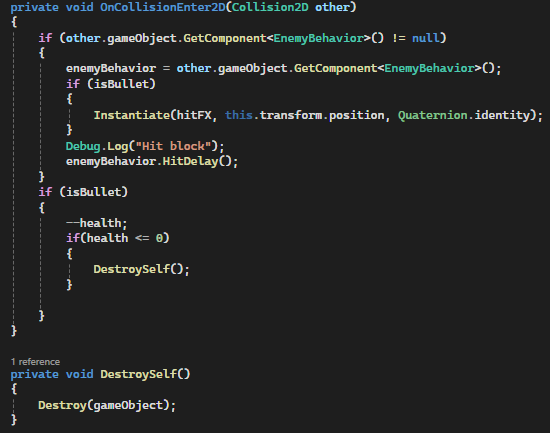
Enemy breaks apart and spawns smaller pieces



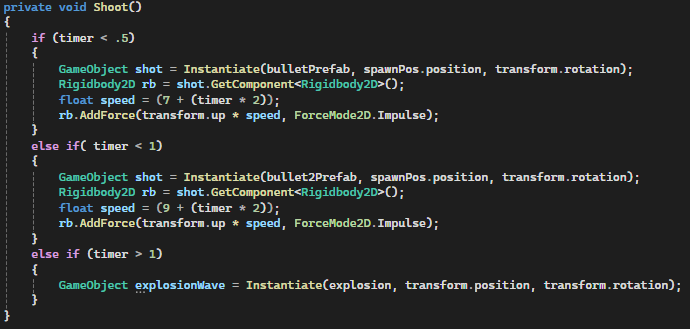
Initial big enemy spawning



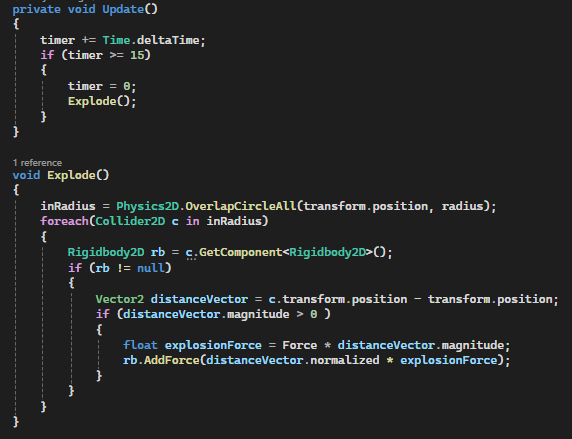
Bullet hits enemy

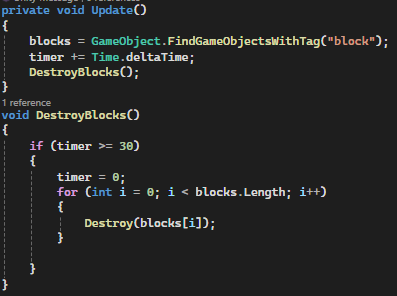


Bullet created



Optimization 5: Use invoke repeating rather than update to calculate a 15 second timer





Optimization 6: FindWithTag in update.

Optimization, but not preferred: when it’s time to destroy blocks, can find them all then rather than in update.

Enemies are created in game manager in both main menu and actual level. If logic is simple enough, could move it into main menu so that when each enemy is created, it can be added to an array rather than always looking for objects.

Alternatively, can check which level you are in when spawning enemies. If main menu, add it to the main menu array. Can assign a ref to main menu in the inspector. When determining which level, can use an int or something simple rather than comparing level name.