ASSIGNMENT 2: 'FIX' Projectile Direction - C++ (1/3)

- **Problem:** Projectiles don't hit where the player expects. (under 'crosshair')
    - 1) Projectile is spawned in player hand which can be left / right based on orientation.
    - 2) Camera Perspective. Aiming at something really close will miss target due to camera alignment with character.

- **Solution:** Line-trace from Camera to World and find desired 'impact' location.
    - Re-calculate new projectile spawn Rotation by using:
        - Spawn Location (Hand position)
        - Impact Location (Line Trace result)
        // Reference 'Find Look At Rotation' function for help.
    - (If nothing was hit, use 'trace end' vector as desired target)
    // Line trace against multiple object types: WorldDynamic, WorldStatic

ASSIGNMENT 2: Blackhole Projectile - Blueprints (2/3)

- "Blackhole" Ability
    - Projectile class spawned via key input (same as *MagicProjectile*)
    - *RadialForceComponent*, but using continuous 'Force' (Not Impulse) to pull-in Actors.
        - Ignore 'Pawn' collision object type. (Player must remain unaffected of pull)
    - Extra *SphereComponent* to 'Destroy' actors on overlap (black hole)
        - Must only destroy 'simulating' actors.
    - Particle Component for the black hole VFX.
    - Destroy itself after ~5 seconds (match the particle system duration)
    - Made in **Blueprints**!
        // Binding of input and spawning of projectile still done in C++.

ASSIGNMENT 2: "Dash" Ability - C++ (3/3)

- Dash/Teleport Projectile Ability
    - Projectile class spawned via input key (same as MagicProjectile)
    - ParticleComponent to show projectile
    - 'Explodes' after 0.2 seconds. (Timer)
        - Play particle effect at point of detonation.
    - Waits 0.2 seconds again (Timer) before Teleporting PlayerCharacter (aka the 'Instigator' of the projectile)
        // Let the detonation effect play before we teleport
        // Make sure you 'stop' the projectile movement while you wait(!)
        // Look back at 'Instigator' we used for ignoring collisions, and re-apply here for teleportation.
    - On hit with world: immediately stop movement and execute same behavior (explode + teleport)
    - Made in **C++**!