Third Person Grapple Hook Asset Documentation

1. *Animation Events Script*

   Includes public unity events variables and the method for every event variable. This Events used in

   Animations of Grappling for “Start the Rope”, “Start the Movement of the Player”, “Stop the Rope”

2. *Camera Handler Script*

   Just a simple Third-Person Camera which you recommend to replace with your own. That’s one is used only for example.

3. *Grapple Point Script*

   Script with references to MeshRenderer and Animator and variables for distance and “can be hooked flag”. In Awake state initialize the MeshRendered and Animator components. In Update method the scripts calculate the distance from the gameObject(Grapple Point) to a Player. If distance > “distance to activate” variable the meshRendered.enabled = true. If distance < “distance to activate” variable then the color of gameObject changes with Color.Lerp method. If distance < “distance to hook” then variable for “can be hooked flag” = true.

4. *Spring Script*

   Rewrited Script of original Unity’s Spring Component for customizable and animated rope.

5. *Third Person Movement Script*

   Includes a simple Third-Person Movement with Running and Jumping which you recommend to replace with your own. That’s one is used only for example.

In new update Third Person Movement Script was separated to script for movement(“Third Person Movement Script”) and script for grappling(“HookToGrapplePoints”). Also added script for new grappling method to player’s mouse position(“HookToMousePoint”). You can choose any of methods but don’t forget to check the “IsHookedToGrapplePoints” Boolean in Unity Inspector on Third Person Movement Script.

6. Hook To Grapple Points script

   Includes a code in Update method for finding Grapple Points with OverlapSphere and “Grapple Point” tag and adding them to “Grapples” List. From list code finding the closest grapple point to the player to show up only one point. Also includes a method “Hook to Grapple Point”. If closest grapple point to the player has a “can be hooked flag” = true. Code saves it’s transform.position to prevent the changing of closest grapple point while player moves. After that finds the Vector to grapple point and move the player. Last one is “Draw the Rope” method which uses the Spring Script and closest grapple point references to find the Vector and set position of LineRenderer in LateUpdate method.

7. Hook To Mouse Point

   In Update method includes a code to find a center of player’s view via Camera and uses that center to cast a ray. Point where ray collides saves for grappling. If that point is in range of “maxDistanceToMousePosition” code saves its position to prevent from changing and after that

moves player to that point. Player moves while he's not in the range of minimum distance variable that disable "IsHooked" boolean. If player landed already but "isHooked" Boolean is still "True" then code doubles the minimum distance variable while it fits to the if statement. After player hooks again minimum distance return to the first number via saved variable in Start method.

All Scripts has //documentation lines which can help you understand the code. But if you still have trouble, you can text me on the mail: mukiraus@gmail.com