

TPCWC Documentation :

I recommend sticking to the YouTube videos for clear understanding or you can also see all of these definitions inside the Inspector as well.

YouTube Link :

<https://www.youtube.com/watch?v=P2lv76AuN6M&list=PLmsjo7zubHe6hFqELbr0Fva8EF4uJQ6YP>

All scripts fall into 3 major categories :

1. Controller Scripts [Location : Assets/TPCWC/Scripts/Controller/]

- Camera Controller - Responsible for all the camera movement
- Character Motor - Manages all the animations of Player and Companion.
- Companion Input - Input based on Artificial Intelligence. It uses Unity's Nav Mesh Agent for calculations of speed, velocity, etc.
- Game Controller - Controls the follow / stop and Player Switch Mechanics. It also restarts the level when both Player and Companion dies.
- Input Manager - Main Class which takes all the inputs from the user and feed them to Character Motor.

2. Enemy Script [Location : Assets/TPCWC/Scripts/ Enemy /]

- Enemy AI - Brain of the Enemy NPC.

3. Utilities Scripts [Location : Assets/TPCWC/Scripts/ Utilities/]

- Door - Attach this to any object which you want to Animate via Push Button or a Switch.
- Push Button - Attach this to any object which will then act as a pushable button which will trigger the door open event.
- Push Button IK - This script is attached at RUNTIME as soon as player starts interaction with the Push Button. It is destroyed as soon as interaction ends. This script is responsible for the IK Matching of the right hand of the player with the push button.

- Switch - Trigger the door open event as soon as it collides with the Player.
 - Health - Attach this to any player or NPC you want to receive damage on.
 - Weapon - Attach this to any object to make it behave as Melee Weapon or make a weapon using Shift+Alt+W.
 - Weapon Standalone - This is a standalone script which you can attach to any NPC and if it is having a weapon, it will start attacking the Object.
 - State Machine Behaviours Scripts -
 - Enable Weapon - Define the Weapon's trigger's enable and disable time.
 - Toggle Parameter - It will enable / disable the animator parameter.
-

Detailed Script definitions with their public properties :

1. Controller Scripts [Location : Assets/TPCWC/Scripts/Controller/]

- Camera Controller -
 1. **Follow Speed** : How fast the Camera Follows the Player
 2. **Mouse Speed** : How fast Camera Rotates with mouse
 3. **Controller Speed** : Camera rotation speed with X-Box Controller!
 4. **Min Rotation** : Minimum Allowed Rotation
 5. **Max Rotation** : Maximum Allowed Rotation
 6. **PlayerTarget** : Will be autoassigned to current player
- Character Motor -
 1. **Model Mesh** : The Humanoid Model of this character which holds the animator component.
 2. **Move Speed** : Define how fast you want player to move.
 3. **Run Speed** : Define how fast you want player to run.
 4. **Rotate Speed** : Define how fast you want player to rotate.
 5. **Attack Time** : How long is your Attack animation.

- Companion Input -

1. **Tag To Attack** : Define which tag you want your companion to attack. *(By Default it is set to 'Enemy')*.
2. **Follow Target** : The target you want this companion to follow. Set it to the player or you can change this via script to make him follow something else.
3. **Attack Target** : It will be autoassigned to the nearby 'Tag To Attack' when it comes in the attack distance.
4. **Companion Sensor** : The trigger collider of this companion which is responsible for all the senses. It will be disabled if this Companion becomes Player.
5. **Awareness Radius** : The radius of the Companion Sensor Trigger.
6. **Follow Speed** : Define how fast the companion follows the 'Follow Target'.
7. **Stopping Distance** : Define how far the companion stops from the 'Follow Target'.
8. **Attack Distance** : Define from how far you want the Companion to attack at 'Tag To Attack'. *(This should always be less than 'Awareness Radius')*.
9. **Attack Delay** : Define how much this Companion waits between his Attacks.

Can Follow, Chase and Attack are the AI states. These are here for debug purposes so you can see at RUNTIME in which state the Companion is currently in!

- Game Controller -

1. **P1** : Assign here the first Player.
2. **P2** : Assign here the second Player.
3. **Inactive P** : Auto Assigned at RUNTIME with the player which is behaving as the Companion driven by AI Input.
4. **Active P** : Auto Assigned at RUNTIME with the player which is behaving as the Player driven by User Input.
5. **Restart Delay** : Define the delay you want in restarting scene when both player-companion dies! *(You must Replace this logic if you want something else to happen when both dies like an end screen!)*
6. **Follow Button** : Type the Button name that you have defined in Input for toggling the Companion Follow!
7. **Switch Player Button** : Type the Button name that you have defined in Input for toggling between Player and Companion!

- Input Manager - Define all the Input button names in the Inspector.

2. Enemy Script [Location : Assets/TPCWC/Scripts/ Enemy /]

- Enemy AI -
 1. **Tag To Attack** : Define the tag you want this Enemy to attack.(By default it is set to 'Player')
 2. **Waypoints** : Add the Empty GameObjects which will be working as Waypoints.
 3. **Patrol Speed** : Define how fast this Enemy will Patrol between above define Waypoints.
 4. **Patrol Delay** : Define how long you want Enemy to stay at a single Waypoint before moving to the Next.
 5. **Chase Speed** : Define how fast this Enemy will chase 'Tag To Attack'.
 6. **Awareness Radius** : Define the radius of the child Sphere Sensor Collider of this Enemy.
 7. **Attack Distance** : Define from how far this Enemy will attack the 'Tag To Attack'.
 8. **Attack Delay** : Define how long this Enemy wait before attacking again.
 9. **Target** : It will be auto assigned once the 'Tag To Attack' reaches inside 'Awareness Radius'.

All the AI States i.e Patrol, Chase, Attack, Reset Path are for debug purpose, you can see them at RUNTIME changing with the surroundings!

3. Utilities Scripts [Location : Assets/TPCWC/Scripts/ Utilities/]

- Door -
 1. **Door Open Animation** : Name of the animation which opens the door
 2. **Door Close Animation** : Name of the animation which Closes the door
 3. **Need Key** : If true, door opens automatically.
- Push Button -
 1. **IK Target** : The position and rotation which you want player to align while interacting with this Push Button.
 2. **Event Time** : Time in seconds when you want the Open Door Event to occur. Coordinate this time with your animation, for default animation it is at 1.1s.
 3. **Door Target** : Door you want to open with this Push Button.
 4. **Input Button** : Button name that we have defined in Input that will Trigger the Push Button Event! (*Right now it's set to 'P'*)

NOTE : *This Script Attaches the 'PushButtonIK' script at RUNTIME to the Player's Animator Object as soon as it starts interaction.*

'PushButtonIK' script is responsible for the IK Matching of the Right Hand of this Player.

- Switch -
 1. **Door Target** : Assign the 'Door' object you want to open with this Script.
 2. **Close Door On Exit** : If true, as soon as the Player exits this switch trigger, door will close again!
- Health -
 1. **Max Health** : The maximum amount of health you want this object to have.
 2. **Current Health** : The current health you will see at RUNTIME of this object.
 3. **Animator** : It will be auto assigned to this object's animator.
- Weapon - All the fields will be overwritten by the parent 'Weapon Standalone' Script. If you want to change something please change it inside that parent script.
- Weapon Standalone -
 1. **Tag To Attack** : The Tag you want this weapon to attack.
 2. **Weapon** : Drag the 'Weapon' you have added inside this Object.
 3. **Damage** : Define how much Damage you want this weapon to cause to 'Tag To Attack'.
 4. **Anim** : The Animator will be autoassigned at runtime!