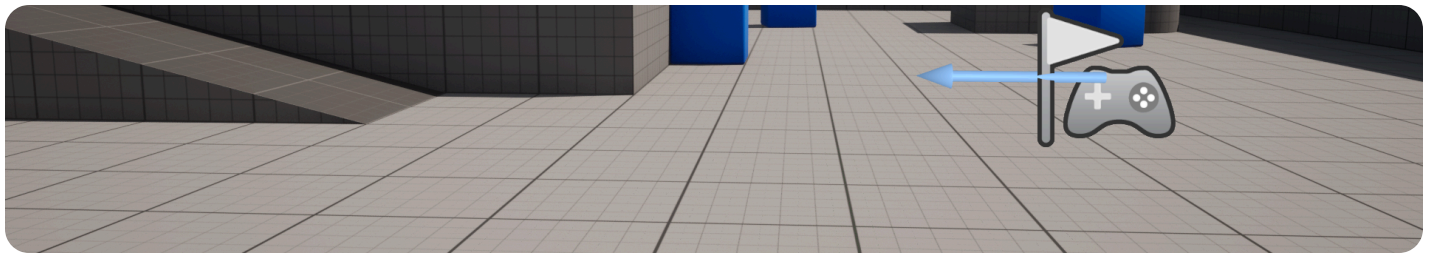# Player Start Actor

Use Player Start Actors to set up starting locations for players.

For any Unreal Engine project that uses playable characters, you can use the **Player Start** Actor to control where, in the game world, each player character will spawn at runtime.

The Player Start Actor is visually represented as a game controller icon with a flag and an arrow that indicates which direction the player will be facing when they spawn.

When you select a Player Start Actor, you will see two additional components: a Capsule that is a visual representation of the physical space the Actor occupies in the game world, and a Transform widget.

A Player Start Actor in the Level. Left: Unselected; Right: Selected. Click the image for full size.

If your Level doesn't have a Player Start Actor, the player will always start at the 0,0,0 coordinates in the world.

Placing a Player Start Actor

Player Start Actors are placed into the Level via drag-and-drop, same as other Actors.

> 💡 You can have multiple Player Start Actors into the Level and control which Player Start is used as the player spawn location using Blueprint or C++.

To place a Player Start Actor, follow these steps:

Click the image for full size.

1. Click the **Quick Add** button on the **Main Toolbar**.
2. Select **Basic > Player Start**.

3. Click and drag the **Player Start** Actor into the Level Viewport.

When using a Player Start Actor to spawn a player, the position and rotation of the player character will match the Player Start's position and rotation. If you want to have a player look at something specific when they spawn, make sure that the **arrow** component of the Player Start you use to spawn them is pointing in the direction that you want the player to look.

If you have more than one Player Start Actor in your Level, the player will start from a random Player Start each time, prioritizing Player Start Actors that are unobstructed.

Player Start BADsize Warning

While placing or moving a Player Start Actor, you may see a white label with red borders and the text **BADsize**.

Click the image for full size.

This warning appears when the Player Start Actor overlaps or intersects with something else in the Level, such as props or terrain. To resolve this, move the Player Start until its **Capsule** component no longer overlaps or intersects with anything else.

Using Player Start Actors in Blueprints

Unreal Engine has several Blueprint nodes that you can use to work with Player Start Actors. These are useful for a number of different scenarios, including respawning players at a Player Start Actor's location or spawning a player at a different Player Start based on different conditions.

The following nodes for working Player Start Actors are available in the **Game Mode** Blueprint:

- [FindPlayerStart](): Finds a Player Start Actor in the Level based that has a specific **tag** (defined in the **IncomingName** property). You can set a Player Start Actor's tag from the Actor's **Details** panel.

- [Restart Player at Player Start](): Respawns a player using a specified Player Start Actor.

The screenshot below shows an example of how to use these nodes to restart a player at the location of a Player Start with the tag `TheStartIWant` when pressing the **E** key. If you want to try this yourself, recreate these nodes in your project's **Game Mode** Blueprint.

Click the image for full size.

Ignoring Player Start in PIE

When you preview your game in the Unreal Editor using [Play-in-Editor (PIE)](#), you may not want to spawn your character at the position of the Player Start actor. Instead, you might want to navigate to a specific location in your level first, and start your PIE session from that location.

There are two ways you can achieve this:

- In the Play Options menu, in the **Spawn Player At** section, you can choose either to spawn at the current camera location or at the default Player Start location.

  ]
- You can **right-click** your desired starting location in the Level Viewport and select **Play From Here** from the context menu.

Customizing Player Start Behavior

Most of the behavior of the Player Start Actor is implemented in the `FindPlayerStart()` and `ChoosePlayerStart()` functions of the Game Mode Base class (`AGameModeBase`). If you want a different behavior, you can override one or both of these functions in your project's Game Mode class.

For example, instead of having these functions choose a Player Start Actor at random as described above on this page, you might want them to choose one according to a different set of criteria, like:

- the Player Start Actor that is closest to the last checkpoint the player reached.
- the Player Start Actor that is closest to the current location of the player's teammates.
- other attributes associated with the player or their character, like their starting team assignment.

Both the `FindPlayerStart()` and `ChoosePlayerStart()` functions are passed the Controller for the player being spawned in the Player parameter. You can use this reference to access information about the player in your own implementations if necessary.

FindPlayerStart

The engine calls `FindPlayerStart()` automatically whenever it needs a new starting position for a player. If that player has previously been spawned with an Actor to mark its starting location, this function returns that Actor. Otherwise, it calls `ChoosePlayerStart()` to select an Actor.

This is a good choice to override if you may want to determine a different starting location when a player restarts than the one that was chosen when they initially joined the game.

ChoosePlayerStart

The default implementation of `ChoosePlayerStart()` chooses a Player Start Actor at random, preferring those that are not obstructed by other blocking geometry.

This is a good choice to override if you want to keep the default behavior of `FindPlayerStart()` in place, but you want to use your own custom logic to choose a starting location when a player first joins the game.

Example

The following implementation of `FindPlayerStart()` is from a Blueprint class that derives from Game Mode Base. It gets the current Game State, and looks for a reference position called Checkpoint Location that has been saved in that Game State. If it is able to get that reference position from the Game State, it returns the Player Start Actor that is closest to that position. Otherwise, it falls back on choosing a Player Start Actor at random by calling the original implementation in the base class.

Click the image for full size.

ⓘ For more information on the Game Mode and Game State, see [Game Mode and Game State](#).