

Creating a Prefab in Unity for an agent

Prefabs are types of assets. They are used to create a reusable *GameObject*, which can be used in other scenes and to create instances at runtime. If you change properties in a prefab, it is automatically applied to all instances of that prefab.

Creating a prefab

First, create an instance of an agent, which you want to store into a prefab, in the scene. Add all components, like scripts, colliders, etc. and change the component parameters as you wish. Now create an empty prefab in the Project View and drag and drop the instance from the Scene View to the empty prefab in the Project View.

Components that must be attached on an agent prefab

- Derived AgentScript
 - Add the four cameras to the agent, two for each eye.
- CharacterController
- ControllerCollider
- HeadLookController

Parameters that must be filled

- Derived AgentScript
 - All four camera variables (*CameraLeft1*, *CameraLeft2*, *CameraRight1*, *CameraRight2*)
 - The invisible object to control the head animation (*StandardLookAtPoint*)

Instantiation of objects from prefab via Editor

Simply drag and drop the prefab into the scene.

Instantiation of objects from prefab at runtime

For instantiation at runtime, the prefab must be stored in “Assets/Resources”. Instantiation is done like in the following example. The code instantiates the prefab named “Lerpz”, at position (0,0,0) and with zero rotation.

C# CODE:

```
Instantiate(Resources.Load("Lerpz") as UnityEngine.Object,  
            new Vector3(0,0,0),  
            Quaternion.identity);
```