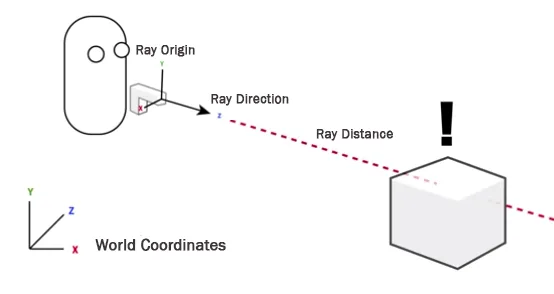
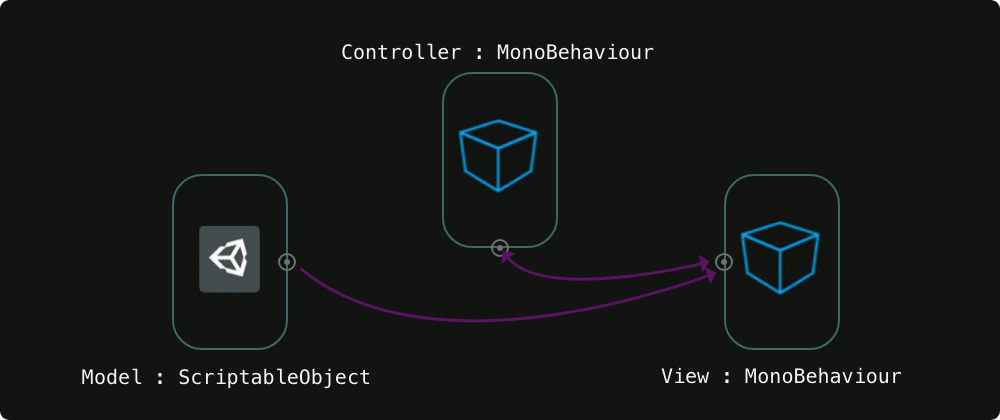
# **Raycast**

* Casts a ray, from point origin, in direction direction, of length maxDistance, against all colliders in the Scene.
* The Raycast function is extremely useful for creating connections between objects in Unity.
* Raycast in Unity is a Physics function that projects a Ray into the scene, returning a boolean value if a target was successfully hit. When this happens, information about the hit, such as the distance, position or a reference to the object’s Transform, can be stored in a Raycast Hit variable for further use.
* You may optionally provide a LayerMask, to filter out any Colliders you aren't interested in generating collisions with.
* While Raycast can be very straightforward to use, it can be a little unintuitive at first.

# **ScriptableObject**

* A ScriptableObject is a data container that you can use to save large amounts of data, independent of class instances.
* One of the main use cases for ScriptableObjects is to reduce your Project’s memory usage by avoiding copies of values.
* Using ScriptableObjects makes it easier to manage changes and debugging.
* Every time you instantiate that Prefab, it will get its own copy of that data. Instead of using the method, and storing duplicated data, you can use a ScriptableObject to store the data and then access it by reference from all of the Prefabs. This means that there is one copy of the data in memory.

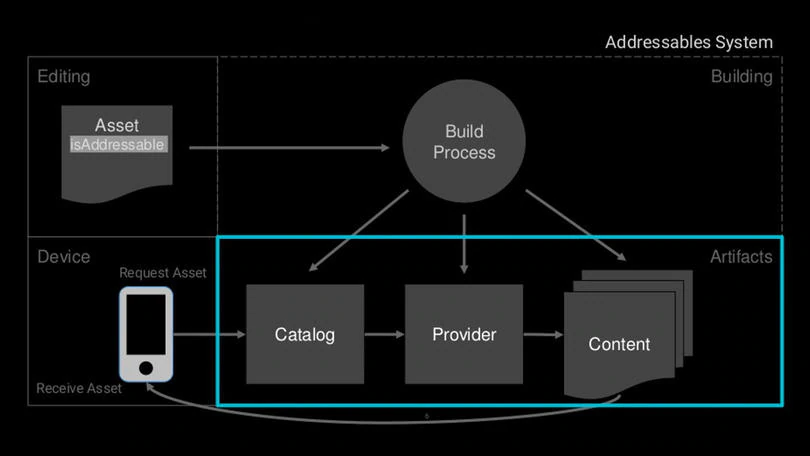


* Data that you save from Editor Tools to ScriptableObjects as an asset is written to disk and is therefore persistent between sessions.
* The main use cases for ScriptableObjects are:

1. Saving and storing data during an Editor session.
2. Saving data as an Asset in your Project to use at run time.

# **Addressables**

* The Addressable Asset System allows the developer to ask for an asset via its address.
* Once an asset (e.g. a prefab) is marked "addressable", it generates an address which can be called from anywhere.
* Use ‘Window->Asset Management->Addressables’ to begin working with the system.
* Addressables use asynchronous loading to support loading from any location with any collection of dependencies.
* By packing asset bundles more efficiently and reducing iteration time, Addressables provides a simple way to make your game more dynamic.

****

# 

# **AssetBundles**

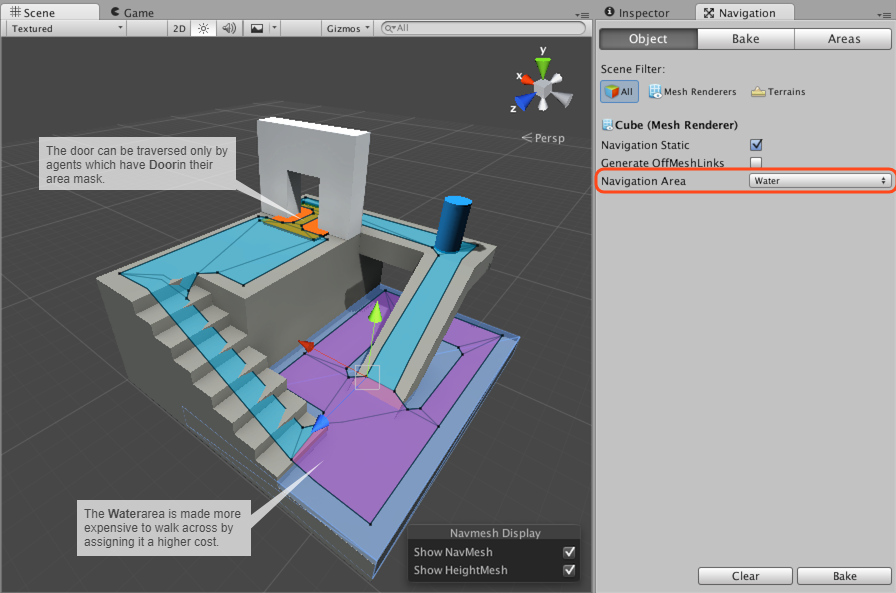
* An **AssetBundle** is an archive file that contains platform-specific non-code Assets (such as Models, Textures, Prefabs, Audio clips, and even entire Scenes) that Unity can load at run time.
* AssetBundles can express dependencies between each other; for example, a Material in one AssetBundle can reference a Texture in another AssetBundle.
* AssetBundles can be useful for downloadable content (DLC), reducing initial install size, loading assets optimised for the end-user’s platform, and reduce runtime memory pressure.

**Spine animation**

* spine animation refers to using the Spine 2D skeletal animation tool to create and animate characters or objects.
* Spine is a popular third-party tool that allows you to create smooth and dynamic animations by animating skeletal meshes.
* It's widely used in 2D game development to bring characters to life.

# **NavMesh**

* A NavMesh is a navigational mesh used for pathfinding in Unity.
* It allows characters and objects in your scene to find their way around obstacles and reach their destinations efficiently.
* A NavMesh is a designated mesh in your Unity scene, which specifies navigable areas in your environment, including areas where characters can walk, as well as obstacles.
* This is useful for scenarios which incorporate pathfinding and AI-controlled navigation. The NavMesh Agent component helps characters to avoid each other, move around the scene toward a common goal, or any other type of scenario involving spatial reasoning or pathfinding.



# **Shader Graph**

* Shader Graph is a tool that enables you to build shaders visually. Instead of writing code, you create and connect nodes in a graph framework. **Shader** Graph gives instant feedback that reflects your changes, and it’s simple enough for users who are new to shader creation.
* Shader Graph is available through the Package Manager window in supported versions of the Unity Editor.

