

## GLOSSARY

**2D Object:** A 2D GameObject such as a tilemap or sprite.

**3D Object:** A 3D GameObject such as a cube, terrain or ragdoll.

**Animation:** Animation is the process of creating a sequence of static or dynamic images that give the illusion of movement. These images can either be two-dimensional (2D), or three-dimensional (3D). Animation has been around for centuries as an art form, and while it used to be created with hand-drawn images, modern technology allows animators to create using computer software.

**Animation Blend Shape:** Having a mesh smoothly change its form to another mesh.

**Animation Blend Tree:** Used for continuous blending between similar Animation Clips based on float Animation Parameters.

**Animation Clip:** This represents a single animation asset in Unity. It contains keyframe data for animating specific GameObjects or characters.

**Animation Key:** An animation key, also known as a keyframe, is a specific point in time inside an animation where a parameter (such as position, rotation, or scale) is set to a particular value.

**Animation Layer:** The process of animating visual elements or components arranged in layers within a user interface (UI) or a 2D/3D scene.

**Array:** A type of collection, which is a set of data that's stored and accessed using a single variable instead of multiple variables.

**Aspect ratio:** An aspect ratio is a proportional relationship between an image's width and height.

**Asset:** An asset is a representation of any item that can be used in your game or project. An asset may come from a file created outside of Unity, such as a 3D model, an audio file, an image, or any of the other types of files that Unity supports.

**Asset Package:** Asset packages are collections of files and data from Unity projects, or elements of projects, which Unity compresses and stores in one file with the .unitypackage extension.

**Asset Store:** The Unity Asset Store contains a library of free and commercial assets that Unity Technologies and members of the community create.

**Boolean:** In computing, the term Boolean means a result that can only have one of two possible values: true or false.

**Box Collider:** The Box collider is a built-in cube-shaped collider. It is useful for in-application items such as boxes and crates, or as a simple collider shape that you can stretch and flatten to make ledges, steps, and any other cuboid objects.

**Build:** A build in software development, refers to compiling and packaging the source code and assets into an executable or deployable package for testing or distribution.

**Camera:** A camera is a device through which the player views the world.

**Canvas:** The Canvas is the area that all UI elements should be inside. The Canvas is a Game Object with a Canvas component on it, and all UI elements must be children of such a Canvas.

**Capsule Collider:** The Capsule collider is a built-in 3D capsule-shaped collider made of two half-spheres joined together by a cylinder.

**Character Controller:** A CharacterController allows you to easily do movement constrained by collisions without having to deal with a rigidbody.

**Cinemachine:** Cinemachine is a modular suite of camera tools for Unity which give AAA game quality controls for every camera in your project.

**Collider:** A collider is a Unity component that defines the shape of a GameObject for the purposes of physical collisions.

**Component:** Components are the functional pieces of every GameObject. Components contain properties which you can edit to define the behavior of a GameObject. For more information on the relationship between components and GameObjects, see GameObjects. To view a list of the components attached to a GameObject in the Inspector window, select a GameObject in either the Hierarchy window or the Scene view.

**Console Window:** The Console Window shows errors, warnings and other messages generated by Unity. You can also show your own messages in the Console using the Debug class.

**Content:** The content is the term used to describe the objects in your scene that are being rendered. Their appearance is a result of the lighting context acting on the materials that have been applied to the objects.

**Culling Mask:** In Unity, the "Culling Mask" refers to a property of cameras used to control which layers of objects are rendered in a scene. This property determines which layers are visible to the camera and which are not.

**Event System:** The Event System is a way of sending events to objects in the application based on input, be it keyboard, mouse, touch, or custom input.

**Extrude Edges:** The Extrude Edges tool pushes a new edge out from each selected edge, connected by a new face for each edge

**FirstPersonShooter:** A type of video game whose gameplay involves shooting enemies and other targets and in which a player views the action as though through the eyes of the character they are controlling.

**Float:** Float is a shortened term for "floating point." By definition, it's a fundamental data type built into the compiler that's used to define numeric values with floating decimal points.

**Frame:** In Unity, a frame is considered a rendered image presented to the player's screen.

**Frames Per Second:** Frames per second (fps) is a measure of how many still images, or frames, are displayed in a single second of video or animation.

**Function:** A function is a self-contained program segment that carries out some specific, well-defined task.

**GameObject:** GameObjects are the fundamental objects in Unity that represent characters, props and scenery.

**Gizmo:** Gizmos are graphics associated with GameObjects in the Scene.

**HDR:** HDR stands for High Dynamic Range and refers to a technique that expresses details in content in both very bright and very dark scenes.

**IDE:** An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. The most commonly used IDE with Unity is Microsoft Visual Studio, which is often recommended and provided as a default option when installing Unity on Windows.

**Input Key:** An "Input Key" is an identifier used in software development, particularly in game development with platforms like Unity, to detect and respond to specific user inputs, such as keyboard presses, mouse clicks, or touchscreen gestures.

**Input Manager:** The Input Manager is where you define all the different input axes and game actions for your project.

**Inspector:** The Inspector displays detailed information about your currently selected GameObject, including all attached Components and their properties. Here, you modify the functionality of GameObjects in your scene.

**Instantiate:** Instantiating means bringing the object into existence.

**Integer:** A number without any decimal value, like 3 or 200.

**Interactable:** An object in a Scene that the user can interact with (for example, grab it, press it, or throw it).

**Interpolation:** Interpolation provides a way to manage the appearance of jitter in the movement of your Rigidbody GameObjects at run time.

**Invoke:** The Invoke functions allow you to schedule method calls to occur at a later time.

**Invoke Repeating:** The InvokeRepeating method in Unity is a built-in function that allows you to schedule a method to be called repeatedly at fixed time intervals.

**Keyframe:** Keyframe animation records the state of the object and then interpolates between the changes of each keyframe.

**Layer:** Layers are most commonly used by Cameras to render only a part of the scene, and by Lights to illuminate only parts of the scene. But they can also be used by raycasting to selectively ignore colliders or to create collisions.

**Layer Mask:** In Unity, a "layer mask" is a bitmask that allows you to selectively include or exclude specific layers from various operations, such as physics calculations, raycasting, and rendering.

**Material:** All rendered meshes use a material to define how that object is displayed. A material's attributes are based on what Shader is used by the material.

**Mesh:** All 3D objects in Unity are broken down into triangles or polygons which comprise of the mesh.

**Mesh Collider:** A collider component that takes the shape of the current mesh attached to it.

**Mesh Renderer:** A component used to control how a mesh appears in the scene

**Model:** Any 3D asset. Often created in a 3D modeling program.

**MonoBehavior:** Unity's base class that many Unity Scripts derive from. Offers many functions such as void Update, Start and Awake

**Near clipping plane:** The clipping plane that is closest to the camera.

**Normal:** A vector representing the direction perpendicular to a mesh. Unity uses Normals to apply shading and calculate object orientation.

**Orthographic:** A camera that doesn't measure depth. All objects regardless of how far away they are will look the same distance away from each other.

**Package:** A single file containing a collection of assets or features for importing to and exporting from Unity Projects.

**Parent:** An object that contains one or more child objects in the Hierarchy. Child objects move, scale and rotate as their parent does.

**Particle:** An individual object that is created and managed by a particle system. A particle can be a mesh, texture or even a collection of sprites.

**Particle System:** A means of creating various effects such as smoke, fluid and gas by using small, simple particles.

**Perspective:** A type of camera that measures Depth, similar to real-life cameras.

**Physics:** Physics in Games represents real-world attributes such as Gravity, Collisions, Time, and more. Physics can be changed through the settings and through scripting.

**Physics Engine:** Uses mathematics to simulate physics (such as gravity and collision) in a virtual environment

**Physics Material:** Used to simulate physics for 3D objects. A Physics Material 2D can be used for 2D objects.

**Pixel:** The smallest part of a computer image. Pixel size depends on screen resolution.

**Player Settings:** A window that contains settings for how Unity will build and display the final application

**Plug-in:** Code that can be added to create additional functionality in Unity.

**Position:** A value that represents where an object is positioned in space based on an X, Y and Z axis.

**Prefab:** A type of asset that is a stored version of a gameobject that includes all attached components. A prefab can be instantiated during runtime.

**Project:** A collection of scenes, assets and settings that make up the entirety of what you create in Unity.

**Project Settings:** Settings that you can adjust to determine how physics, input and other parts of the project behave.

**Project View:** A panel in the Unity Editor that gives you access to all the assets in your project.

**Quad:** A 3D object that is similar to a plane but has a much more efficient mesh.

**Quaternion:** represents an object's rotation in 3D space

**Rendering:** The process through which Unity takes the objects in a scene and creates a graphic image.

**Rendering Mode:** A shader parameter that gives you transparency options.

**Rig:** A collection of joints in a mesh, much like a skeleton.

**Rigidbody:** A Component that is added so forces such as gravity affect the object.

**Rotation:** A value that represents how an object is rotated in space based on an X, Y and Z axis.

**Scale:** A value that represents how big an object is in the X, Y and Z axis.

**Scene:** A world that holds many GameObjects as needed by the developer. Scenes can be loaded and unloaded and are primarily used for multiple levels.

**Scene View:** The window in the Unity editor through which the current scene can be viewed and modified.

**Scripts:** Written code that directs the behavior of objects in a Unity Project. Scripts in Unity are written in C# (C Sharp)

**Shader:** A component that applies a material to a GameObject. Some shaders are more advanced than others can even manipulate how the texture moves.

**Skybox:** A material/texture used to represent the games sky.

**Sphere Colliders:** A collider component that is the shape of a primitive sphere.

**Sprite:** A graphic that has no depth. Can be anything from an image used as a texture to a 2D object.

**Sprite Mask:** A 2D texture used to hide parts of an underlying graphic from the renderer

**Sprite Renderer:** A component similar to a mesh renderer but specifically for 2D sprites.

**Sprite atlas:** A collection of 2D textures in a single image to save processing time.

**State Machine:** Used in the Animator Controller to keep track of and edit the various possible animation states and their transitions. It can also be a term that refers to an AI and how they act. For example, an AI may have a walk state or an attack state that code will transition between.

**String:** An object which can only have a value of text.

**Tag:** A reference that is applied to objects so they can be easily identified by a script.

**Terrain:** A term for a landscape in your scene, usually natural. A terrain GameObject can be edited through that object's inspector to quickly create a detailed environment.

**Terrain Collider:** A collider component that takes the shape of the terrain object it is attached to.

**Text:** Part of the UI that presents information to the user in a non-interactive way.

**Text Input Field:** A UI object that allows for the user to enter requested text.

**Text Mesh:** the mesh that is used to display a text string

**Text Mesh Pro:** A Unity Package that produces high-quality text, as well as other features involving UI elements.

**Texture:** An image that is applied to a GameObject, sprite or mesh.

**Third Person Shooter:** A shooter game where the camera is positioned above and behind the player.

**Toggle:** A UI checkbox that can be either on or off

**Toolbar:** A row of buttons and tabs to allow quick editing.

**Trail Renderer:** Applies trails behind a GameObject as it moves in the game.

**Transform:** A component that handles the object's position, rotation and scale.

**Trigger:** A type of collider that activates code but isn't physically touchable by Gameobjects,

**Variable:** A symbol that represents an object. For example, you may have a variable that represents a number, vector, object or an array of something.

**Vector:** A mathematical concept that allows you to describe a direction and a magnitude. In games, a vector is used to describe properties such as position, rotation or scale.

**Visual Studio:** An IDE created by Microsoft that is used to type in code.

**Void Awake:** A function that happens on the frame when a script is enabled, but before void Start.

**Void FixedUpdate:** Similar to Void Update, FixedUpdate is called every fixed frame-rate frame. This function is called at a fixed rate and is not dependent on the game's speed.

**Void Start:** A Unity Function that happens on the frame when a script is enabled, just before any of the void Update function is called.

**Void Update:** A Unity Function that happens once every frame. This function is called at a variable rate depending on the game's speed.