

Glossary

Aliasing – Jagged edges that appear in the rendering of polygon edges.

Alpha blending – Blending of transparent to non-transparent areas in materials.

Alpha value – The visibility of a renderer, often used to fade out a part of the game.

Alpha version – A build you may distribute amongst developers or early trusted partners.

Animation Wrap Mode – The mode in which the animation is played back, for example, looping, once, ping pong (back and forth).

Anisotropic filtering – A method that improves the quality rendering of textures shown at oblique angles.

Anti-aliasing – The smoothening of jagged edges that appear in the rendering of polygon edges.

Array – A way of storing multiple entries of similar data in a single variable.

Assets – Models, scripts, images, sounds, and other items, stored in your project or 'assets' folder.

Bake (noun) – A finished lightmap, often used to refer to the finished texture or look of the environment.

Bake (verb) – The process of saving the lighting of a scene to a texture to optimize performance, aka lightmapping.

Batching – The act of grouping computing tasks together.

Behaviors – Scripts that you write to define how an object responds to certain situations.

Beta version – A build that is more feature complete, that you are likely to release to the public. See also *Alpha version*.

Boo – A scripting language used by Unity, but much less popular than C# and Javascript, due to its lack of mobile support. See also **C#** and **Javascript**. Also, an exclamation used by ghosts!

Boolean variables – True or false variables, often referred to as a 'flag', these are often used simply to check if something has already happened.

Bottlenecks – Points in a piece of software or hardware that slow down performance by being slower than other elements in the system.

Bug – A problem in your script or game's design that needs fixing!

Build – A finished or test version of the game exported by Unity.

Build settings – The panel in Unity used to choose the platform to build to, and some additional settings for building.

C# – A scripting language used by Unity. See also **JavaScript** and **Boo**.

Caffeine – A substance useful around deadlines in your development!

Camera – Your viewpoint on the world. More than one camera can be used in a scene for rendering additional details.

Cartesian coordinate system – Coordinates expressed in X, Y, and Z format, for example Vector3 (5,10,5).

Child – An object nested beneath a parent object in a hierarchy.

Class – A way of grouping data within a script; variables, functions, and co-routines can also be part of a class and called as such.

Clip planes – The two planes of view that the 3D world is rendered within, by a camera.

Clones – Duplicates of a game object or a prefab.

Code embedding – Taking code, such as HTML, and placing it inside another page, such as you might need to do when embedding a YouTube video in another website.

Collider – Colliders give mesh objects in Unity a physical presence, so they can interact with other objects.

Collider scaling – The act of redefining the boundary of a collider, in order to alter where collisions with it occur.

Collision detection – The act of detecting when two object's colliders bump into one another, known as a collision.

Collision event – The occurrence of a collision, or often used as shorthand for the variable storing collision data.

Comments – Code that has been disabled from being executed within a script.

Component – Components add behavior to your objects; they may be part of Unity or scripts.

Concatenation – Adding together the values of variables to create a larger string of data.

Console – The error reporting window in Unity – the most important window there is! See **Window | Console** or use *Ctrl + Shift + C* (PC)/*Command + Shift + C* (Mac) to open it.

Co-routine – A function that may run in parallel to current functions to perform other instructions outside of the current execution order.

Crosshair – A texture used to render 'sights' onscreen, as a way of helping the player aim in shooter games.

CSS – Cascading Style Sheets; these are used to style web pages in web design.

Cubemap – A material made up of six textures sometimes used to create effects such as skyboxes.

Cut-scene – Part of a game that interrupts play to present a piece of narrative or view of a part of the environment (generally something that a character cannot currently see), to help the game progress.

Damping – A blanket term used to describe the phenomenon of the value getting attenuated to a lower value.

Dashboard – A separate screen of mini-applications built in to Mac OS X.

Debugging – The process of hunting for errors in a script.

Deployment – The act of building and providing your game to users.

Development cycle – A phase of a game's development. This could be a Prototyping Cycle in which mechanics are tested, or it could be a Final Cycle in which finishing graphical polish is added.

Draw Calls – A rendering request made to the graphics card; the more of these that are made, the harder your hardware must work, and the slower your project will perform.

Draw distance – The distance within which a camera renders an object.

Dynamic batching – The grouping together of similar material and scaled objects for improved performance in rendering.

Editor (Unity) – Usually used to refer to Unity, rather than your script editor.

Editor scripting – Scripts written to modify how the editor works.

Environment – What the player of your game inhabits, this could be a terrain, an indoor model, or outer space!

Expensive – Something that requires a lot of CPU and GPU power to run in a game.

Extrapolation – A mathematical technique used to predict values based on known measurements.

Fading – Decreasing visibility over time .

Favicon – The icon seen to the left of the website title, at the top of a browser, or, in some browsers, to the left of the address bar.

FBX – The file format used to store and import 3D models in Unity.

Fixed Update – A function that runs its commands, every fixed physics frame, independent of frame rate.

Floating point or 'float' – A numerical data type for variables that contain a decimal placed number.

Fog density – Similar to the alpha value of fog, this defines how thick the fog is, how well objects can be seen through it.

Fog/Distance fog – An effect to simulate fog that can be added in Unity's Render settings, in order to assist hiding disappearing objects at the limit of a camera's draw distance.

Force – A way of propelling a rigidbody object in 3D; also, the element that surrounds us, binds us, and holds the universe together.

FPS (frames per second) – The number of frames of the game that are rendered each second, a general benchmark of performance in games.

Function – A set of instructions carried out in a script.

Game Engine – A game engine (for example, Unity!) is a piece of software that handles rendering, sound, and other plugged-in code libraries, such as physics, to create games.

Game Object – An object in your currently open scene is referred to as a game object.

Geometry – A general term for the vertices, triangles, and other data that make up a 3D model.

GUI (Graphical User Interface) – Menus, HUDs, and so on, are all examples of GUI elements; often 2D, but sometimes 3D. These elements allow the player to interact with the game, sometimes in addition to what their avatar allows.

GUI Skin – An asset created by Unity and stored in the project that allows you to style GUI scripted elements.

GUI Skin assets – The assets that Unity creates to allow you to style GUI scripts.

GUI Text – A simple component allowing you to render text on the screen in 2D.

GUI Texture – A component allowing you to render 2D textures on screen.

Heightmaps – Image files containing light and dark areas to define height geometry.

HTML – Hypertext Markup Language, the language web pages are written in.

HUD (Heads Up Display) – The non-environmental graphical display that gives the player information about the game. In a character-based game, this could be a health meter; in a racing game, it could be the speedometer of the car, or position amongst the other racers.

Instantiate – The act of creating an object within the world during gameplay.

Instantiation – The act of creating instances of a game object or prefab during runtime.

Integer or 'int' – A numerical data type of a variable that is a whole number.

Invalid arguments – Data sent to a function's arguments that does not match its data type, for example, sending a GameObject when a function is expecting a floating point number.

Inventory – Items the player character may have collected during the game.

IRC – Internet Relay Chat, one of the oldest internet chat systems – simply install mIRC (PC) or Colloquy (Mac) to get started.

JavaScript – A scripting language used by Unity.

Keyframes – Specific points of an animation timeline that you set values for and use Tweening to interpolate between.

Layer Collision Matrix – Found in the Physics manager, this allows you to uncheck layers in order to stop objects set on those layers from colliding.

Layers – A way of separating objects in your scene for the purpose of differentiating such things as collisions and lighting.

Lerp – Short for linear interpolation, a transition between two values in an equation.

Level of Detail (LOD) – the amount of detail rendered on screen, which may decrease over distance from the camera.

Light – A light in Unity can be added as a component of an object.

Light mapping – Saving the light of a scene to a texture file to optimize performance.

Lightmap – The texture file in which the light for a scene is stored.

Lightmapper – The built-in tool in Unity that handles the baking of lights to a texture file, in order to save performance and give better visuals.

Lightmapping – The act of baking lighting of a game to a texture file.

Linear Interpolation or **Lerp** – The act of transitioning between two values directly, using linear polynomials; that is assuming that they are connected with straight lines.

Local space – The coordinates of an object in relation to its parent.

Local variables – Variables that are declared within a function and are not accessible outside of it. See also **Private variables** and **Public variables**.

Loop – A way of repeating instructions in a script. Also, the repetition of an animation.

Mesh – A shape defined by joining dots, known as vertices, in 3D.

Mesh filter – A component that contains the shape of a 3D object.

Mesh renderer – A component that controls the visual appearance of a mesh object.

Monodevelop – The default script editor installed with Unity.

Movieclip— A data template in Adobe's Flash development environment, comparable to the Prefab concept in Unity.

Normalization— A mathematical approach to standardizing values. Often values are normalized into the range 0 to 1. When applied to vectors, Normalization means to make the length of the vector equal to unity.

Normals— A vector that points out perpendicularly at each vertex in a mesh.

Null— If a variable is left unassigned at any time in scripting, it is considered null— meaning 'not set'.

Object space— The coordinates of an object in relation to its parent.

Occlusion Culling— A method built into Unity that avoids rendering objects behind objects that the camera cannot see. In simple terms, it culls— removes— from rendering, objects that are not included— therefore occluded— from view.

OnTriggerEnter— A function used to detect collisions between triggers and other colliders.

Origin— The zero position of the 3D world, represented by (0,0,0) in Cartesian coordinates.

Orthographic view— A 3D view of your game scene.

Package— A way of storing content from a Unity project and sharing it with others.

Parent or Parent Object— The uppermost object of objects arranged in a hierarchy.

Particle— A single emission of a particle system (See *Chapter 8*).

Particle animator— The component in Unity that handles how particles behave over their lifetime.

Particle emitter— The component in Unity that handles the emission of particles in a particle system.

Particle renderer— The component in Unity that handles the visual appearance of particles in a particle system.

Particle system— A set of components used for visual effects, such as, smoke, dust, explosions, and so on (See *Chapter 8*).

Perspective— A 3D view of your game scene, where objects get smaller the further away they are.

PhysX— The nVidia physics engine that is built into Unity.

Pixel – A single dot that makes up a computer image on screen.

Plane – A flat, square, primitive shape that can be created inside Unity.

Playhead – The time indicator of the Animation window that shows the frame currently being shown or played back.

Points – Something scored in a game, but also another word for the Vertices in a mesh.

Polygon – A shape that makes up part of a mesh and is typically triangular.

Prefab – An object created in the scene and stored in the project; it can be instantiated at runtime.

Preset – A predefined group of settings.

Primitive – Basic 3D shapes, such as, cubes, spheres, planes, and capsules.

Private variables – Variables accessible only by the script itself. See also **Local variables** and **Public variables**.

Projected shadows – Shadows cast on the environment.

Prototype – A raw demonstration of gameplay, using basic shapes and simple effects to test if an idea works.

Prototyping – The act of creating a prototype.

Proximity-based sound – 3D sound that becomes louder as the player character or Audio Listener component (to be precise) approaches it.

Public variables – Variables accessible for modification in the Inspector. See also **Local variables** and **Private variables**.

Rapid prototyping – Quickly making a game mechanic for testing, without attempting to create final-quality visuals.

Raycasting – Creating a vector between points in the world, in order to detect intersection of other elements.

Rect – A variable data type containing four float values – X, Y, Width, and Height.

Registration point – The point from which a GUI element is rendered.

Remote camera – A camera not attached to the player specifically, but used to show another part of the environment as part of a cut-scene.

Rendering – The act of displaying graphics on a screen.

Require component – A way to make Unity automatically add a component to an object when applying a script to it.

Rigidbody – Unity's way of applying the physics engine to an object. Rigidbody is both a component and a class for scripting.

Routine – A part of a script that is currently being run by the game engine.

Runtime – The time when the game is running.

Scenes – Scenes are Unity's way of separating content; they can be used to create different levels or simply host content to be loaded additively.

Script Editor – A text editor designed specifically for working with coding languages, such as, C# and JavaScript.

Scripts – Written in C#, JavaScript, or Boo; scripts in Unity are used to provide behavior for objects or modify how the Unity editor itself works.

Self-illuminated shaders – Shaders that have a lit appearance without requiring light to be cast upon them.

SendMessage – A command used to call a function in a script on a separate game object to the one containing this command's script.

Spawn(ing) – see *Instantiation*.

Splash Image – A welcome image built into standalone builds of Unity Projects.

Sprites – Image files (textures) used as 2D graphics in a game.

Standalone – An exported desktop build of your Unity game.

Static objects – Objects that will not move in Unity and that, therefore, can be lightmapped and batch rendered, if they share a material.

Static variables – Public variables that can easily be accessed from other scripts simply by writing, for example, `className.variableName`.

Strafing – Stepping sideways when playing a game, usually whilst shooting a weapon.

Streaming – The downloading of data from the internet; in games, this may be the act of downloading additional content for the game, while it is being played.

Stylesheet – See CSS.

Tagging – Applying a single word to an object at the top of the inspector to help identify it individually or often as part of a group in Unity.

Tags – A unique identifier that can be applied to game objects in order to sort of find them in a scene

Tangent – A 3D vector that is parallel to the surface of a model.

Terrain – An environment geometry that can be created in Unity, usually used to model the ground and hills a player can walk over.

Terrain editor / toolkit – The toolset shown as a component on terrains in Unity.

Texture – An image file rendered as part of a 3D mesh or 2D GUI element.

Texture-Swapping – Exchanging one texture for another to create the effect of changing texture.

Transform – The name of the component in Unity that handles Position, Rotation, and Scale, and also a scripting class.

Translate – A command used to move objects in 3D space.

Translation – The act of moving an object within a scene.

Trigger Collision Detection – Detecting when one or more triggers set to trigger mode are intersecting; often used to detect colliders within a certain space.

Trigger(s) or Trigger mode – Triggers are colliders placed into trigger mode, meaning they do not have a physical presence, but other colliders can be detected when intersecting them.

Tweening – The automated feature of animation that transitions between values defined by Keyframes.

Unity – The game engine you're learning right now! When referred to, this usually means the editor itself.

Unity 3D – A name often mistakenly given to Unity, thanks largely to the software's website being `unity3d.com` – if you hear someone say Unity 3D, it's the same thing!

Unity script – A phrase often used to describe the JavaScript code in Unity.

Update – A function that runs its commands every frame.

UV channel – The part of a 3D model file containing data on how a texture is mapped onto the shape of a mesh.

Variables – A way of storing information in a script; these can be made adjustable in the inspector by making them public.

Vector – A line drawn in 3D space with a direction and a length.

Velocity – The physical quantity representing the speed and direction of a rigidbody object.

Vertex count – The number of vertices in a model's mesh.

Vertex – A point in 3D space that defines a corner of a triangle in a mesh model.

View Gizmo – The spokes and cube in the top right of the Scene view that allows you to switch between perspective and orthographic views.

Watermark – A semi-transparent 2D overlay, often of branding, on top of a game or video.

Web player – Unity's own web plugin that allows games to be played inside a web browser.

Widgets – Mac OS X widgets appear on the Dashboard feature of Mac Operating systems.

World zero – The zero (or origin) position of the 3D world.

Wrap mode – The way in which animation plays back, for example, 'once' or 'loop'.

