

Notes on Semantics

Overview

Semantics basically associates certain data from one stage of the graphics pipeline to another. For example, Unity associates different parts of vertex data from the engine to the Vertex Stage via predefined semantics. From the vertex stage to the fragment stage, however, you can assign whatever value you wish to different semantics, granted that you assign the proper data type to the corresponding semantic.

Semantics Between Stages of the Pipeline

Note that you can use the same semantic names at different stages of the pipeline, but they are not equivalent or associated with one another.

For example:

Engine -> Vertex TEXCOORD0 and Vertex -> Fragment TEXCOORD0

are NOT the same.

Unity Documentation

Semantics Overview

<https://docs.unity3d.com/Manual/SL-ShaderSemantics.html>

Vertex Program Input Semantics

<https://docs.unity3d.com/Manual/SL-VertexProgramInputs.html>

Semantics Tables

Vertex Input Semantics

Semantic	Datatype	Description
POSITION	float3 / float4	Vertex position in object space.
NORMAL	float3	Normal vector at vertex.
TEXCOORD0	float2 / float3 / float4	1st UV coordinate set.
TEXCOORD1	float2 / float3 / float4	2nd UV coordinate set.
TEXCOORD2	float2 / float3 / float4	3rd UV coordinate set.
TEXCOORD3	float2 / float3 / float4	4th UV coordinate set.
TANGENT	float4	Tangent vector.
COLOR	float4	Vertex color.
SV_VertexID	uint	Vertex ID.

Vertex Output/Fragment Input Semantics (Interpolator Variables)

Semantic	Datatype	Description
SV_POSITION	float4	<p>Vertex position in clip space. Note that the vertex position in the Vertex Stage of the pipeline has to convert the input vertex position (object space) to clip space with the method:</p> <pre>UnityObjectToClipPos(<float3 or float4 inputVertexPosition>);</pre> <p>When converted, assign the return value to the data structure holding the clip space vertex data. See the Basic Shader for an example.</p>
TEXCOORDN	float2 / float3 / float4	Some arbitrary data. N indicates an uint starting from 0.
COLOR0	float2 / float3 / float4	<p>Some arbitrary data. These are low precision values between 0 - 1.</p> <p>Usually used for simple colors, since they are float4 with values between 0 - 1.</p>
COLOR1	float2 / float3 / float4	<p>Some arbitrary data. These are low precision values between 0 - 1.</p> <p>Usually used for simple colors, since they are float4 with values between 0 - 1.</p>

***NOTE:** There are limits to how many interpolator variables you can use here. For best performance minimize number of semantics used here. See Unity documentation link under Semantics Overview.

Interpolator Variable Limits

Platform	Interpolator Variable Limit
OpenGL ES 2.0 (Android) DirectX 9 - Shader Model 2.0	8
DirectX 9 - Shader Model 3.0 (#pragma target 3.0)	10
OpenGL ES 3.0 (Android) Metal (iOS)	16
DirectX 10/11 - Shader Model 4.0 (#pragma target 4.0)	32

Fragment Output Semantics

Semantic	Datatype	Description
SV_Target	fixed4	The final pixel color.
SV_TargetN	fixed4	For writing multiple colors, where N is some uint. SV_Target0 = SV_Target.
SV_Depth	float	For overriding the depth buffer. Not commonly used.