

TooShader v1.0 – Doc v0.4

English version

DeformVertex :

- (DeformVertexes Wave <Wave Func> <Base> <Amplitude> <Phase> <Frequency>)
- (DeformVertexes Normal <Wave Func> <Base> <Amplitude~0.1~0.5> <Phase> <Frequency ~1.0~4.0>)
- DeformVertexes Scale <Wave Func> <Base> <Amplitude> <Phase> <Frequency>
- DeformVertexes Move <Wave Func> <Base> <Amplitude> <Phase> <Frequency> <Dx> <Dy> <Dz>
- DeformVertexes Rotate <Angle per second in degrees> <Axe X> <Axe Y> <Axe Z>
- DeformVertexes
 - Orientates the mesh in front of camera, like a sprite
- DeformVertexes None
 - No deformation (default value)
- DeformVertexes Prev
 - Use deformation from previous stage

Coeff = WaveFunc(Phase + Time * Frequency) * Amplitude

<Wave Func> can be:

- Sin
- Triangle
- Square
- Sawtooth
- InverseSawtooth

BlendFunc :

- BlendFunc <Src> <Dst>
 - Buffer = Texture * Src + Buffer * Dst ;
- BlendFunc Add
 - => BlendFunc One One
- BlendFunc Filter
 - => BlendFunc DestColor Zero, ou BlendFunc Zero SrcColor
- BlendFunc Blend
 - => BlendFunc SrcAlpha InvSrcAlpha

BlendFunc is used for transparency, to indicate how to combine the stage with the current pixels in the video buffer (the screen).

We can use several layers of blending to create more evolved effects.

The Add, Filter and Blend function used as 2nd (or further) layer on a opaque polygon will be automatically optimized (by using ColorOp functions).

<Src> and <Dst> can be:

"Zero": 0

"One": 1

"SrcColor": it's the color from the current stage texture (modified by RgbFunc, etc...)

"InvSrcColor",

"SrcAlpha",

"InvSrcAlpha",

"DestAlpha",

"InvDestAlpha",

"DestColor": it's the current color in the video buffer

"InvDestColor",

"SrcAlphaSat",

ColorOp :

- ColorOp <Func> [InvArg]

ColorOp allows doing Blending between the several textures layers of a polygon, before to be put in the video buffer (with or without Blending).

"Disable": the stage is off

"SelectArg1": by default, Arg1 = Texture from this stage

"SelectArg2": by default, Arg2 = Texture from previous stage

"Modulate"

"Modulate2X",

"Modulate4X",

"Add",

"AddSigned",

"AddSigned2X",

"Subtract",

"AddSmooth",

"BlendDiffuseAlpha",

"BlendTextureAlpha",

"BlendFactorAlpha",

"BlendTextureAlphaPM",

"BlendCurrentAlpha",

"Premodulate",

"ModulateAlpha_AddColor",

"ModulateColor_AddAlpha",

"ModulateInvAlpha_AddColor",

"ModulateInvColor_AddAlpha",

"BumpEnvMap",

"BumpEnvMapLuminance",

"DotProduct3",

"MultiplyAdd",

"Lerp",

InvArg : optional, inverts Arg1 and Arg2 ; so Arg1 = Texture from previous stage, and Arg2 = Texture from this stage (it's useful for some functions which works just in one definite way)

TcMod :

- TcMod Rotate <Angle per second in degrees> [base Angle in degrees]
- TcMod Scale <Scale S> <Scale T>
- TcMod Scroll <DeltaS per second> <DeltaT per second>
- TcMod Stretch <Wave Func> <Base> <Amplitude> <Phase> <Frequency>
- TcMod Transform <m11> <m12> <m21> <m22> <ds> <dt>
- TcMod Turb <Base> <Amplitude> <Phase> <Frequency>
- TcMod None
 - No transformation (default value)

RgbGen :

- RgbGen Identity
 - No modification on texture (color = white)
- RgbGen LightingDiffuse
 - Uses color from diffuse lighting
- RgbGen LightingSpecular
 - Use color from specular lighting
- RgbGen Wave <Wave Func> <Base> <Amplitude> <Phase> <Frequency>
 - Uses color created from Wave function

- RgbGen Entity
 - Uses color from object (specified in the code, by the programmer)
- RgbGen OneMinusEntity
 - Uses the invert of the color from object
- RgbGen Prev
 - Uses the same RgbGen than the previous stage
- RgbGen Fixed
 - Specify a fixed color in 24 bits code (ie: 0xRRGGBB) for RgbGen, and a 8 bits (from 0 to 255) code for other color Gen functions

RedGen, GreenGen, BlueGen can be used when RgbGen is not specified, or when RgbGen is used with Identity, Wave, Entity, or OneMinusEntity. Each one modifies only one component of the final color. The supported functions are : Identity, Wave, Entity, and OneMinusEntity.

AlphaGen :

- AlphaGen Identity
 - Uses alpha from texture, without modification
- (AlphaGen Vertex)
 - N/A
- AlphaGen Wave <Wave Func> <Base> <Amplitude> <Phase> <Frequency>
 - Create alpha using a Wave function
- AlphaGen Entity
 - Use alpha specified in objet (in the code, by the programmer)
- AlphaGen OneMinusEntity
 - Use the invert of the alpha specified in the object
- AlphaGen Prev
 - Use the same AlphaGen than previous stage
- AlphaGen Fixed
 - Specify a fixed alpha 8 bits code (from 0 to 255)

TcGen :

TcGen allows generating texture coordinates.

- TcGen Environment
 - Generate environment mapping coordinates
- TcGen CamSpacePosition <sx sy sz sd> <tx ty tz td>
 - Generate coordinates from the vertex position, in the camera space
 - $u = vx * sx + vy * sy + vz * sz + sd$
 - $v = vx * tx + vy * ty + vz * tz + td$
- TcGen CamSpaceNormal <sx sy sz sd> <tx ty tz td>
 - Generate coordinates from the normal direction, in the camera space
- TcGen WorldSpacePosition <sx sy sz sd> <tx ty tz td>
 - Generate coordinates from the vertex position, in the world space
- TcGen WorldSpaceNormal <sx sy sz sd> <tx ty tz td>
 - Generate coordinates from the normal direction, in the world space
- TcGen ObjectSpacePosition <sx sy sz sd> <tx ty tz td>
 - Generate coordinates from the vertex position, in the object space
- TcGen ObjectSpaceNormal <sx sy sz sd> <tx ty tz td>
 - Generate coordinates from the normal direction, in the object space

Texture Loading :

- Map <TextureName>
- BumpMap <TextureName> [Roughness]
- HeightMap <TextureName> [Roughness]
- NormalMap <TextureName> [Roughness]
- AnimMap <frequency> { TextureName1 [TextureName2 etc...] }
- NoMipMap

- LimitedMipMap

Lighting :

- NoSpecular
 - No specular light
- Specular p
 - Indicates the coefficient of specular light emitted
- Specular r g b
 - Indicates the coefficient of specular light emitted per component
- SpecPower
 - Indicates the power of Specular (default is 16)
- Diffuse p
 - Indicates the coefficient of diffuse light emitted
- Diffuse r g b
 - Indicates the coefficient of diffuse light emitted per component
- AutoLighted
 - The texture is full lighted (no shadowing modification)
- NoDynamicLight
 - Just the StarLight will apply on this shader

Miscellaneous :

- Cull {Back | Front | None | Disable}
 - Determines how the faces are culled
- AddressMode { Wrap | Mirror | Clamp | Border | MirrorOnce } [Wrap | Mirror | Clamp | Border | MirrorOnce] [BorderColor]
 - Indicates the repetition mode of the texture (outside of the [0; 1] range) for U and V (if there's no 2nd parameter, then the same mode will be applied to U and V)
- Mirror [BlendSrc BlendDst]
 - Indicates the surface is a mirror
- Water
 - Indicates the surface is water

Version Française

DeformVertex :

- (DeformVertexes Wave <Wave Func> <Base> <Amplitude> <Phase> <Fréquence>)
- (DeformVertexes Normal <Wave Func> <Base> <Amplitude~0.1~0.5> <Phase> <Fréquence ~1.0~4.0>)
- DeformVertexes Scale <Wave Func> <Base> <Amplitude> <Phase> <Fréquence>
- DeformVertexes Move <Wave Func> <Base> <Amplitude> <Phase> <Fréquence> <Dx> <Dy> <Dz>
- DeformVertexes Rotate <Angle par seconde en degrés> <Axe X> <Axe Y> <Axe Z>
- DeformVertexes Sprite2D
 - Oriente l'objet en direction de la caméra, comme un sprite
- DeformVertexes None
 - Pas de déformation (valeur par défaut)
- DeformVertexes Prev
 - Utilise la déformation du stage précédent

Coeff = WaveFunc(Phase + Temps * Fréquence) * Amplitude

<Wave Func> peut être :

- Sin
- Triangle
- Square
- Sawtooth
- InverseSawtooth

BlendFunc :

- BlendFunc <Src> <Dst>
 - Buffer = Texture * Src + Buffer * Dst ;
- BlendFunc Add
 - => BlendFunc One One
- BlendFunc Filter
 - => BlendFunc DestColor Zero, ou BlendFunc Zero SrcColor
- BlendFunc Blend
 - => BlendFunc SrcAlpha InvSrcAlpha

BlendFunc sert pour les transparences, pour indiquer comment est combiné le stage avec ce qu'il y a dans le buffer vidéo actuellement.

On peut mettre plusieurs couches de blending pour avoir des effets plus évolués.

Les fonctions Add, Filter et Blend utilisées comme 2nde couche sur un polygone opaque seront automatiquement optimisées (en étant converties en ColorOp).

<Src> et <Dst> peuvent être :

"Zero" : 0

"One" : 1

"SrcColor" : c'est la couleur issue de la texture du stage actuel (modifiée par la RgbFunc, etc...)

"InvSrcColor",

"SrcAlpha",

"InvSrcAlpha",

"DestAlpha",

"InvDestAlpha",

"DestColor" : c'est la couleur actuelle dans le buffer vidéo

"InvDestColor",

"SrcAlphaSat",

ColorOp :

- ColorOp <Func> [InvArg]

ColorOp sert à faire du Blending quand un polygone a plusieurs couches de textures, qui vont être combinées entre elles, avant d'être mises dans le buffer vidéo (avec ou sans Blending).

"Disable" : le stage est désactivé

"SelectArg1" : par défaut Arg1 = Texture de ce stage

"SelectArg2" : par défaut Arg2 = Texture du stage précédent

"Modulate"

"Modulate2X",

"Modulate4X",

"Add",

"AddSigned",

"AddSigned2X",

"Subtract",

"AddSmooth",

"BlendDiffuseAlpha",

"BlendTextureAlpha",

"BlendFactorAlpha",

"BlendTextureAlphaPM",

"BlendCurrentAlpha",

"Premodulate",

"ModulateAlpha_AddColor",
"ModulateColor_AddAlpha",
"ModulateInvAlpha_AddColor",
"ModulateInvColor_AddAlpha",
"BumpEnvMap",
"BumpEnvMapLuminance",
"DotProduct3",
"MultiplyAdd",
"Lerp",

InvArg : optionnel, inverse l'Arg1 et l'Arg2 ; ainsi Arg1 = Texture du stage précédent, et Arg2 = Texture de ce stage (c'est utile pour certaines fonctions qui ne marchent que dans un sens défini)

TcMod :

- TcMod Rotate <Angle par seconde en degrés> [angle de base en degrés]
- TcMod Scale <Scale S> <Scale T>
- TcMod Scroll <DeltaS par seconde> <DeltaT par seconde>
- TcMod Stretch <Wave Func> <Base> <Amplitude> <Phase> <Fréquence>
- TcMod Transform <m11> <m12> <m21> <m22> <ds> <dt>
- TcMod Turb <Base> <Amplitude> <Phase> <Fréquence>
- TcMod None
 - Pas de transformation (valeur par défaut)

RgbGen :

- RgbGen Identity
 - la texture n'est pas modifiée (couleur = blanc)
- RgbGen LightingDiffuse
 - utilise la couleur issue de l'éclairage diffus
- RgbGen LightingSpecular
 - utilise la couleur issue de l'éclairage spéculaire
- RgbGen Wave <Wave Func> <Base> <Amplitude> <Phase> <Frequency>
 - crée une couleur en utilisant une fonction
- RgbGen Entity
 - prend la couleur spécifiée dans l'objet, par le programmeur
- RgbGen OneMinusEntity
 - prend l'inverse de la couleur spécifiée dans l'objet
- RgbGen Prev
 - Utilise le même RgbGen que le stage précédent
- RgbGen Fixed
 - Spécifie une couleur codée en 24 bits (ex: 0xRRGGBB) pour RgbGen, et un code 8 bits (de 0 à 255) pour les autres fonctions Gen

RedGen, GreenGen, BlueGen peuvent être utilisées quand RgbGen n'est pas spécifié, ou quand RgbGen est utilisé avec Identity, Wave, Entity, ou OneMinusEntity. Chacun n'agit que sur un composant de la couleur finale. Les fonctions supportées sont : Identity, Wave, Entity, et OneMinusEntity.

AlphaGen :

- AlphaGen Identity
 - l'alpha est pris dans la texture, sans modification
- (AlphaGen Vertex)
 - utilise l'alpha obtenu par le moteur 3D (+ c'est loin, + c'est transparent)
- AlphaGen Wave <Wave Func> <Base> <Amplitude> <Phase> <Frequency>
 - crée l'alpha en utilisant une fonction
- AlphaGen Entity

- prend l'alpha spécifiée dans l'objet, par le programmeur
- AlphaGen OneMinusEntity
 - prend l'inverse de l'alpha spécifiée dans l'objet
- AlphaGen Prev
 - Utilise le même AlphaGen que le stage précédent
- AlphaGen Fixed
 - Spécifie un alpha codé sur 8 bits (de 0 à 255)

TcGen :

TcGen permet de générer des coordonnées de texture.

- TcGen Environment
 - génère des coordonnées pour une EnvMap
- TcGen CamSpacePosition <sx sy sz sd> <tx ty tz td>
 - génère les coordonnées en fonction de la position du vertex dans le repère de la caméra
 - $u = vx * sx + vy * sy + vz * sz + sd$
 - $v = vx * tx + vy * ty + vz * tz + td$
- TcGen CamSpaceNormal <sx sy sz sd> <tx ty tz td>
 - génère les coordonnées en fonction de la position de la normale dans le repère de la caméra
- TcGen WorldSpacePosition <sx sy sz sd> <tx ty tz td>
 - génère les coordonnées en fonction de la position du vertex dans le repère du monde
- TcGen WorldSpaceNormal <sx sy sz sd> <tx ty tz td>
 - génère les coordonnées en fonction de la position de la normale dans le repère du monde
- TcGen ObjectSpacePosition <sx sy sz sd> <tx ty tz td>
 - génère les coordonnées en fonction de la position du vertex dans le repère de l'objet
- TcGen ObjectSpaceNormal <sx sy sz sd> <tx ty tz td>
 - génère les coordonnées en fonction de la position de la normale dans le repère de l'objet

Chargement de texture :

- Map <TextureName>
- BumpMap <TextureName> [rugosité]
- HeightMap <TextureName> [rugosité]
- NormalMap <TextureName> [rugosité]
- AnimMap <frequency> { TextureName1 [TextureName2 etc...]
- NoMipMap
- LimitedMipMap

Eclairage :

- NoSpecular
 - Pas de lumière spéculaire
- Specular p
 - Indique le coefficient de lumière spéculaire renvoyée
- Specular r g b
 - Indique le coefficient de lumière spéculaire renvoyée par composante
- SpecPower
 - Indique la puissance de la lumière spéculaire (défaut = 16)
- Diffuse p
 - Indique le coefficient de lumière diffuse renvoyée
- Diffuse r g b
 - Indique le coefficient de lumière diffuse renvoyée par composante
- AutLighted
 - La texture est totalement éclairée (pas d'ombrage)
- NoDynamicLight
 - Seule la lumière du soleil affecte le shader

Divers :

- Cull {Back | Front | None | Disable}
 - contrôle comment les faces sont éliminées
- AddressMode { Wrap | Mirror | Clamp | Border | MirrorOnce } [Wrap | Mirror | Clamp | Border | MirrorOnce] [BorderColor]
 - Indique le mode de répétition de la texture (en dehors des coordonnées de texture [0 ; 1]) pour U et V (s'il n'y a pas de 2^e paramètre, le mode pour U sera appliqué sur V)
- Mirror [BlendSrc BlendDst]
 - Spécifie que la surface est un miroir
- Water
 - Spécifie que la surface est de l'eau

Exemples :

Ex 1 :

```
{  
//stage1  
BlendFunc ...  
}  
{  
//stage2  
ColorOp <Op>  
}
```

Les polygones seront envoyés en 1 seule passe ; le stage 1 et 2 seront combinés en utilisant l'opération <Op>, puis le résultat sera combiné au buffer vidéo en suivant les règles établies pour BlendFunc.

Ex 2 :

```
{  
//stage1  
BlendFunc ...  
}  
{  
//stage2  
BlendFunc ...  
}
```

Dans ce cas, les polygones seront envoyés en 2 passes ; le 1^{er} stage sera combiné avec le buffer vidéo et le résultat sera stocké de nouveau dans le buffer vidéo ; puis la 2^{nde} passe combinera le stage 2 avec les nouvelles données du buffer vidéo.