Vibrant Grass Shader Documentation

More information here : https://assetstore.unity.com/packages/slug/239134
If you like the asset, please **Write a Review** on the page, it helps a lot ♥ !

-Known Issues and Bug Fixes :

-UI Buttons not appearing in Scene View :

-The UI buttons in the Scene View should appear as soon as you spawn a VibrantGrassShader. If not, please click "FixUINotAppearingInSceneView" in MainControls-Secondary.

-The grass looks strange, each stalks are one-sided :

-Open the "VibrantGrassShader" shader found in "Data-Shaders", click on "Graph inspector" and make sure "Two Sided" is selected for the pipeline you are using. If you're using the Gamma rendering color space, then do the same in the VibrantGrassShaderGamma found in the same folder.

-Transparency not working :

-Transparency won't work in unity 2019 and 2020.
-To fix the Transparency not working : Change the Surface Type of the CreatedMaterial to Transparent, enable "Depth Write", then disable and re-enable the VibrantGrassShader.

-BugFixes Specific to Unity 2019 :

-If rendering in Gamma ColorSpace, the Example Scenes may not work properly.

-Gamma Rendering ColorSpace:

-The asset was made for Linear ColorSpace, so the visuals will be different.
-This is a good start to make the grass look better :
-WindPushStrength and SquashStrength : 0.5
-Reduce the brightness of ColorNoiseColor and WindAddedColor by 2
-TipPositionNoiseMultiplier.y = 0.15
-MinimumHeight : 0.15
-The material presets won't work
-The transparency won't work

-Invert Brush Key not working :

-Simply assign a different key to "InvertBrushKey" in "MainControls-PaintingKeyboardShortcuts".

-DirectoryNotFoundException Error :

-To fix this error, assign your own folder path to "CreatedFilesPath" (Create a folder - right click - Copy Path - paste in CreatedFilesPath). Also, the path can sometimes be too long, which can only be fixed by moving the Unity Project so that the Asset folder path is much shorter.

-Creation arrows missing/hidden behind other shaders :

-Open Assets/VibrantGrassShader/Data/Shaders/VibrantGrassShaderUI. In GraphInspector - Surface, select "Opaque" instead of "Transparent", then click Save Asset.

-Other :

-When painting, clicking once on the grass then pressing "Undo" doesn't work properly.
-In Editor in Play-Mode, stuttering and freezes can happen when rotating the camera, but it won't happen in build.
-If the wind makes the grass go through the ground no matter what values you assign to the WindHillCollision properties on the CreatedMaterial, try to make sure that your ground mesh has proper normals, then click "WrapMeshes" in MainControls in "Secondary".
-Errors related to Shaders and Serializable dictionaries can be thrown. They can be ignored as they won't change the result.

-Known minor Issues :

-PaintShadows Transparency Debug will not work if EnableTransparency isn't enabled.
-While creating a build, you can sometimes get these errors : "InvalidOperationException: EnsureRunningOnMainThread can only be called from the main thread" and "Unable to find player assembly".

--> Just ignore them, they won't affect the build.

-Specific to Built-In :

-While creating a build with the Built-In pipeline, you can get this error : floating point division by zero. Just ignore it.

-Specific to Unity 2022.3.2f1LTS and above :

-When importing, you can get errors about the VibrantGrassShader2019, just ignore them, they won't affect anything.

-Main Controls :

-Double click the Material in "Visuals" to modify its properties (don't modify any other material).
-To use the material presets, copy their properties and paste them onto your created material. You can find the presets in the "_4Presets" Folder.
-More details on the material properties in the "Material" category of this page.

-Performance :

-EnableTransparency : Must be enabled for DistanceFadeTransparency to work. Can help or hurt the performance depending on the circumstances, best is to try it yourself.
-EnableDistanceFadeTransparency and Height : These will work only in play mode. Either make the grass fade in the distance using transparency, or using height (it'll hide underground).
-DistanceFadeOutMinMax : The X value is the range from camera at which the fade out starts, the Y is the range at which it stops. The Y value has to be higher than 75, otherwise it won't work properly.
-LightAndInteractMaxFPS : The maximum FPS that the light and interact system will draw at.
-MaxLightPerGrassField : Sets the maximum light source per grass field.
-MaxInteractPerGrassField : Sets the maximum interactions source per grass field.
-InteractionMaximumErasingTime : The amount of time during which the interaction keeps working after leaving a grass field. After this amount of time, the interaction trail will be erased. Keep it as low as possible to improve the performance.

-Audio :

-Main Audio Values (on MainControls in "Audio") :

-Show the audio sources by enabling "ShowAudioSources" (more details on how to use them below).
-WindAudioClip : You can replace this clip with your own. A softer Default clip is available in the "_3AudioClips" folder of the asset.
-WindVolumeDistanceCurve : The volume will increase and decrease while following this curve based on the distance between the camera and the audio sources.

-Audio Sources :

-Use the attached script (AudioSourceObject) to change the values associated with the position.
-You can duplicate the audio sources and delete them manually or use the booleans found on their respective script.
-"StickToGround" makes the object stick to the colliders that have one of the "GroundLayers" assigned during the setup (you can change them after setup).
-DistanceInterpolationMinMax : These 2 values will change the scale of the spheres. When the camera is inside the Opaque sphere, the volume is at its maximum value. When the camera is outside of the transparent sphere, the volume is at 0.0f. When in between, the volume will interpolate between 0.0f and the WindMaxVolume, while following the WindVolumeDistanceCurve (set in MainControls).

-Notes :

-It is recommended to position the AudioSources so that the inside opaque spheres envelop most of the tall grass, you don't have to be perfectly accurate.
-You can add many Audio Sources without worrying about the performance. The gameobjects will not be in the game, they are simply used as debugs.
-The direction of the sound is calculated using the average of the audio sources that are close to the camera.

-Other :

- CreateMeshesToHideFlattenedGrass : Click this to cut the meshes that have a Height too low to be seen (It doesn't do anything if you haven't painted any low height)
-ResetCutGrassFieldsWithMeshes : Click this to reset the meshes that have been cut by CreateMeshesToHideFlattenedGrass (so you can use them to modify the height again).
-CreatedFilesDirectory : This is here to show you where the created files are, please do not attempt to move the files manually and instead change the CreatedFilesPath in "Setup".
-FindLightAndInteractObjectsOnStart : Enable this to find all the gameobject that have the DynamicLightAndInteract component on them on Start. You should uncheck this boolean if you chose to add the objects manually or by script.
-More infos on Light and Interact Objects in the "Dynamic Lights and Interactions" category of this page.

-Secondary :

-These variables are secondary, you should rarely need them.
-MaxWrapVerticesPerFrame : This is the maximum vertices per grass field being worked on as the meshes wrap. If you are experiencing freezes while wrapping, you should try lowering this value.
-MaxWrapRayDistance : The maximum distance of the ray used to detect the ground when wrapping.
-WrapBasePosAddedHeight : The origin of the ray that is used for detecting the ground when wrapping will be set at this height from the grassfield.
-DebugWrapRay : Debugs the rays used when wrapping.
-WrapMeshes : This resets and wrap all the meshes (useful if you modified the ground), so it will take some time to load. It will not delete anything you have painted.
-InstanceMaterialsOnStart : When enabled, each grass field will instance its necessary materials on start. However, the materials will be automatically instanced when OnEnable() is called (which happens when a grass field enters the camera view), so if you don't want the materials to be instanced on start, it is best to have the VibrantGrassShader disabled before the game starts.

-DestroyInstMatsOnFieldDisable : When enabled, each grass field will have its instanced material destroyed anytime it is disabled (which happens as the grass field exits camera view). Can be useful for small areas to reduce memory usage.
- FixUINotAppearingInSceneView : The UI buttons in the Scene View should appear as soon as you spawn a VibrantGrassShader. If not, please enable this.
-PreventMeshSelection : When enabled, selecting a mesh will automatically select the GrassField instead.

-Runtime Tools :

-DestroyRuntimeInstancedMats : Calls "DestroyInstancedMaterial_Event" at runtime, which destroys all instanced material.
-InstanceRuntimeMats : Calls "InstanceMaterials_Event" at runtime, which instances all material necessary (useful if DestroyInstancedMaterial_Event was called before).

-Data Changeable In Game :

-To access these variables, make sure to use the "VibrantGrassShader" namespace (add "using VibrantGrassShader" all the way up your script.
-LightAndInteractObjectList : Add your dynamic lights and interactions objects here to use them. You can add them by script as well, or enable "FindLightAndInteractObjectsOnStart" in "Other" to find them on start. Removing the objects when they are not being used is recommended for performance.
--> More info on this in the "Dynamic Lights and Interactions" category on this page.
-Notes : There are 2 more values that can be useful to access by script : "WindAudioVolumeApplied" and "WindAudioDirectionApplied", but they are not meant to be modified.

-Data (Don't Touch) :

-Ignore any of these variables, they are not meant to be modified.

-Dynamic Lights and Interactions :

-Add the DynamicLightAndInteract to a GameObject to make the grass use the object for light and interactions.
-The objects have to be added to LightAndInteractObjectsList (in MainControls under "Usable Data") to work. It'll be done automatically on Start if FindLightAndInteractObjectsOnStart is checked (in MainControls under "Other"). You can also add the gameobject manually in the editor (can't add manually in play mode).
-The dynamic lights and interactions will only work in play mode.

-Light :

-LightHeightInvLerpValuesAB : The X value is the distance from the ground at which the light starts to fade out. The Y value is the distance at which the light is completely disabled.

-Interaction :

-Please keep in mind that the interaction is not perfectly accurate.
-InteractObjectAddedHeight : This value will be added to the distance from the ground when comparing it to the height of the grass (to know how much to interact and when to ignore). It is useful when using a floating object. You can debug this using "DebugAddedHeight".
-InteractSizeMinMax and InteractStrengthMinMax : To explain how these work, let's take 2 values : "SpeedMultiplier" and "GrassHeightMultiplier". InteractObjectSpeedInvLerpValuesAB will change the SpeedMultiplier ; the X value is the object's speed at which the SpeedMultiplier is equal to 0.0f, the Y value is the object's speed at which the SpeedMultiplier is equal to 1.0f. The value interpolates in-

between. InteractGrassHeightAboveObjectInvLerpValuesAB will change the GrassHeightMultiplier ; the X value is the grass height above the object at which the GrassHeightMultiplier is equal to 0.0f, the Y value is the grass height above the object at which the GrassHeightMultiplier is equal to 1.0f. The value interpolates in-between.

--> SpeedMultiplier and GrassHeightMultiplier are then used together to determine what strength and size to apply, 0.0f being the minimum amount and 1.0f the maximum.

-Interaction Audio :

-GrassRustleSound : This is the clip being used, you can use the default clip "GoingThroughGrass" found in the "_3AudioClips" folder, or use your own.
-InteractAudioMaxDistance : The maximum distance from the object at which the sound will be heard.
-InteractAudioDistanceCurve : The volume will increase and decrease while following this curve.
-InteractAudioVolumeMinMax and the 2 InvLerp values : These values work the same way as the "Interaction" values.
-InteractAudioSmoothTimeUpAndDown : The X value is the SmoothTime used in a SmoothDamp while the volume is increasing, the Y value is the SmoothTime used while the volume is decreasing.

-Performance :

-LightDrawDistanceFromSource : If you are using a high size for the light, you should increase this value so the distance equate the distance of the light (use DebugLightMaxDrawDistance to see the distance in Play-mode).
-InteractDrawDistanceFromSource : Same as the above but for the interaction.
-InteractMaxHeight : Above this distance from the ground, the interaction will be disabled.

-Texture Painting :

-Click on the UI buttons to start painting.

-Shadows :

-Transparency Debug : This is a debug to help you draw, it won't be in the game.

-Height :

-Painting Floor : This prevents the brush from decreasing the height of the grass lower than the specified value.
-Painting Ceiling : This prevents the brush from increasing the height of the grass higher than the specified value.

-Color :

-The ColorMask system enables you to paint by color, so you can change your colors later without having to paint everything from scratch.
-The Color field above the "ColorMasks" title let's you pick the color of the selected Color Mask. The boxes below let you select which color mask you want to paint with.
-While painting using the Light Brush Type, inverting the brush will erase the light without erasing the color.

-Material :

-In the MainControls script in "Visuals", double-click "CreatedMaterial" to start customizing it.
-Do not modify any Surface Options.
-To use the material presets, copy their properties and paste them on the Created Material. You can find them in the "_4Presets" folder.

-Main :

Colors :

-StalkColorGradientDistance : the distance of the color gradient from the root to the tip of each grass stalk
-StalkColorTip : the color of the tip of the grass stalks
-StalkColorRoot : the color of the root of the grass stalks
-ColorNoiseColor : the color of the color noise
-ColorNoiseScale : the scale of the color noise
-ColorNoiseContrast : the contrast of the color noise

Shadows :

-*The Shadow properties below affect the shadowed areas created while painting shadows.*
-ShadowColorTip : the color of the tip of the grass stalks
-ShadowColorRoot : the color of the root of the grass stalks
-ShadowNoiseColor : the color of the color noise
-ShadowNoiseContrast : the contrast of the color noise
-PaintedColorsShadowMultiplier : how much the shadows affect the painted colors

Wind :

-WindSpeed : the speed of the wind
-WindScale : the scale of the wind noise
-WindPushStrength : how much the wind pushes the grass in the wind direction
-WindSquashStrength : how much the wind squashes the grass
-WindAddedColor : the color that the wind adds to the grass

-Secondary :

-WindHeightInvLerpValueA and B : the wind strength interpolates between the value A and the value B using the height of the grass, the higher the grass is, the stronger the wind is. The minimum strength is 0.0f unless WindMinStrength is above 0.0f.
-WindMinStrength : the minimum strength of the wind (will be used for WindHeightInvLerpValueA and B)
The MessinessNoise is used to make the wind more messy, it multiplies its strength.
-WindMessinessNoiseScale : the scale of the MessinessNoise
-WindMessinessNoiseStrength : the strength of the MessinessNoise
-WindMessinessNoiseSpeed : the speed of the MessinessNoise
The WindRandomSideNoise pushes the grass perpendicularly from the wind direction using a noise.
-WindRandomSideStrength : the strength of the WindRandomSideNoise
-WindRandomSideNoiseScale : the scale of the WindRandomSideNoise
-WindRandomSideNoiseSpeed : the speed of the WindRandomSideNoise
The normals of the ground are used to prevent the wind from pushing and squashing the grass through the ground. The wind reduces its push and squash as the grass gets close to the ground.
-WindHillCollisionSlowDownHeightStart : the distance from the ground at which the grass stalks start slowing down
-WindHillCollisionSlowDownStrength : the strength of the slow down
-WindHillCollisionStopHeight : the distance from the ground at which the grass stalks completely stop being pushed and squashed
EnableIntersectionsTransparency (on MainControls in "Performance") has to be checked for the Intersection Transparency to work. It uses Depth so it won't look exactly the same depending on the camera angle.
-IntersectionTransparencyMaxDistance : the maximum distance from the intersection at which the transparency starts
-IntersectionTransparencyStartPosition : changes the start position of the intersection transparency (MaxDistance is the limit)
-IntersectionTransparencyEndPosition : changes the end position of the intersection transparency

-IntersectionHeightMultiplier : how much the intersection transparency depends on the height (the higher the grass is, the higher the distance of the intersection transparency is)
-IntersectionHeightInvLerpValueA and B : the intersection transparency distance interpolates between the value A and B using the height of the grass
-IntersectionHeightMinDistance : the minimum distance of the intersection transparency (used in the value above)
-MinimumHeight : the minimum height of the grass
-HeightCutThreshold : below this threshold, the grass is disabled (it actually goes through the ground, but you can use "CreateMeshesToHideFlattenedGrass" in the MainControls script to cut it)
The TipPositionNoise is used to randomize the positions of the tips
-TipPositionNoiseScale : the scale of the TipPositionNoise
-TipPositionNoiseMultiplier : the direction that is multiplied with the TipPositionNoise
-TipMessinessNoiseMultiplier : *this noise is used to make the grass messy.* This property is the direction multiplied with it
-FlattenedGrassHeightRemoved : The height removed from grass that is completely flattened. Flattened grass is pushed underground to hide it (unless a cut mesh was created). This value can be overwritten on each grass field as well (in GrassFieldMaster-Other)

-Data :

-*The properties below are not meant to be modified*
-LightSplatTexture : the texture used for the dynamic lights
-InteractionSplatTexture : the texture used for the interactions
-ShadowTexture : the texture used for shadows
-HeightTexture : the texture used for heights
-ColorTexture : the texture used for colors
-GroundNormalTexture : the texture used for the WindHillCollision properties (it represents the normals of the ground)
-WindDirection : the direction of the wind
-WindDirectionDegrees : the direction of the wind in degrees
-OverallAlpha : the overall alpha
-ColorTextureAssignedInt : used to tell the shader whether the color texture is assigned or not
-DistanceFadeAlphaActivated : whether the transparency distance fade is activated or not
-DistanceFadePositionActivated : whether the height distance fade is activated or not
-DistanceFadeStart : the distance from camera at which the distance fade starts
-DistanceFadeEnd : the distance from camera at which the distance fade ends
-CameraPosition : the position of the camera
-IsTransparentMaterial : used to tell the shader whether the material is using transparency or not
-HeightTextureAssigned : used to tell the shader whether the height texture is assigned or not

-Performance :

-Large Areas - Recommendation :

-Even though there is technically no limit to the size of the grass areas, it is recommended to keep the total amount of grass fields loaded at the same time under 400 (300x300 meters), to avoid excessive memory usage.
-For performance and painting precision, it is recommended to keep each VibrantGrassShader instance under 400 grass fields (300x300meters), but you can stack them next to each other.

-Memory Usage :

-When completely done with a VibrantGrassShader at runtime, you should use the event "MainControls.DestroyInstancedMaterial_Event" before disabling/deleting the VibrantGrassShader. You can always use "MainControls.InstanceMaterials_Event" reset.

-MainControls-Visuals-RootsHeightsPrecision :

-You can overwrite this value on each GrassField.
-Low Value : Recommended for tall grass. Takes less disk space and less RAM but the roots won't precisely fit the ground, which is noticeable only when the grass is very short.
-High Value : Recommended for short grass, especially if the camera will come close to it to see the details.

-Other :

-Example Scene :

-Find the ExampleScenes in the "_2ExampleScenes" folder. Load the correct scene according to your project's pipeline.

-ExamplePresetsManager :

-ActivatePresets : Uncheck this if you want to modify the material of the grass, otherwise your changes will be erased
-Color Preset : Select a preset to apply it, it also modifies some other objects to fit the preset

-LightAndInteractObjects :

-It works only in play mode

-StaticLightsExamples :

-The grass below these objects has been painted manually using the PaintColor tool

-Other Tips :

-GrassFieldMaster :

-You can find this script on each seperate grass field.
-You can find on it the created assets for the specific grass field.
-You can reset and wrap one grass field at a time by clicking "ResetMeshAndWrap" in "Controls"

More information here : https://assetstore.unity.com/packages/slug/239134