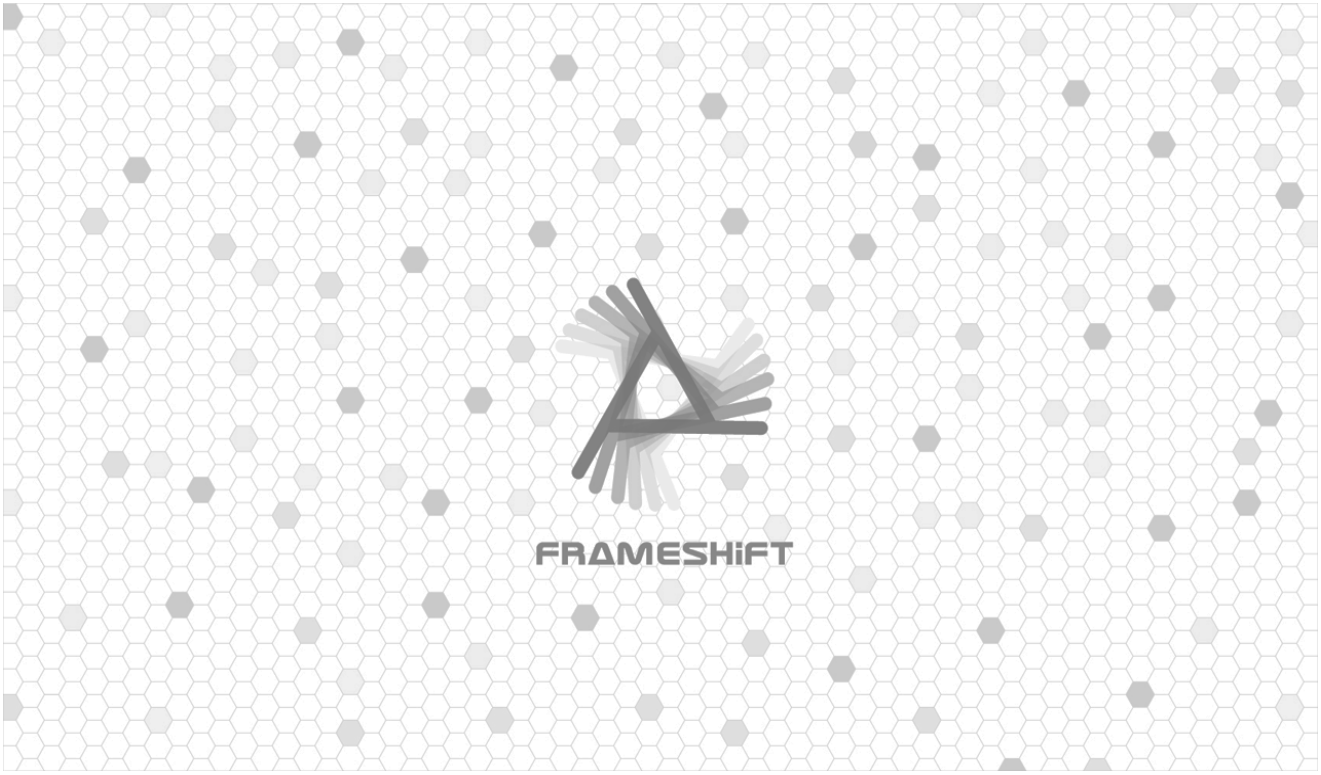
Decal Framework 1.0
User Manual

1. Welcome

Frameshift Decal Framework

for Unity 2.6.1 Game Engine

User Manual



version 1.0

2. Overview

Overview

Usually decals used to create bullet holes, blood splatters or burning of the explosions, but in some cases, the use goes beyond such things. In addition, decals can be created not only in the dynamics, but also during level development. With decals walls of buildings can be uniquely decorate torn posters, cracks or patches of paint.You can use several types of blending to achieve a suitable result. To improve the visualization of decals you can use standard techniques of lighting, such as the bump mapping, parallax mapping and more. Parallax especially looks good as gives much more volume and allows such things as bullet holes go deeper into geometry. Our framework allow you to create a decals in editor under development level design and in runtime. Make a design of your project more interesting and believable.

Features

Platforms :

- Windows
- Mac OS X

Unity:

- Versions : 2.6.1
- Free (Basic)
- Pro

Decal Mesh Generation :

- Based on source surface mesh
- Full Tangent Space Generation (or preserve for non-projective decals)

Decal UV Generation :

- Projective
- Preserve
- Normalized

Additive :

- Mesh offset adjustment
- Mesh generation normal threshold

Static Decals. The system allows you to create static decals in editore during scene development. Create a decal just like prefab and then use it repeatedly. Convenient positioning system allows for one click to put a decal on specified point. Using standard tools of Unity, you can quickly and easily configure other options: decal size, rotation, etc. After creating a set of decals system allows to combine all the decals of a given type at a specified object in single mesh, this feature allows you to optimize the draw calls for maximum performance. Combining can also perform automatically before starting the level, you need only call one function in a library.

Static Skinned Decals. You can create not only simple but also skinned decals. You can quickly and simply give each of your characters necessary uniqueness simply by adding to it a few decals such as stains, blood, patches on clothing and scratches on the body and much more.

Dynamic Decals. Dynamic decals are decals that are created in runtime, such as bullet holes, blood splatters, footprints. Our system allows you to create all these elements without which does not exist any serious project. For each dynamic decal is only necessary to call two functions from the library. The first creates a mesh decals, the second directly creates an object decals. The system monitors all created dynamic decals and controls their behaviour. Combining, randomization, and work with the texture atlas, destroing , fading. Combining the decals in runtime will reduce the draw calls to a minimum and to obtain high performance.

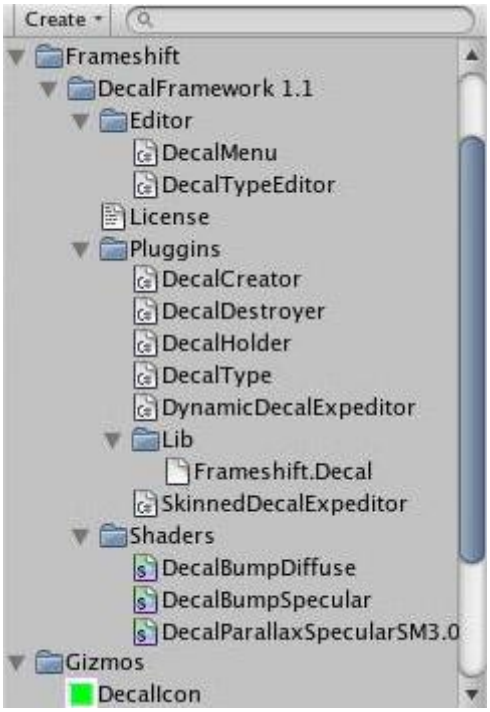
Dynamic Skinned Decals. Dynamic decals also can be skinned. You can easily create blood splatters or damage on your characters in runtime. The system can control the maximum number of decals on skinned object and quality of the skinning, 1,2,4 bones.

Advanced. Often we need to create a decal on the surface that has bump mapping shader. If Decal has bump mapping shader by itself than we need for blending between the surface normal map and normal map of decals. Our system allows you to do this while preserving the original UV coords and tangential space of the source surface in decal mesh. To achieve the effect we provide several shaders which is a modification of Unity bump shader. All you need is either to use these shaders or write you own using our shader as an example , this is not so difficult.

3. Installation

Installation

1. Open your project.
2. Select menu item **Assets -> Import Package .**
3. Select file **Decal Framework 1.xx.unitypackage .**
4. After extracting the files in the проект, hierarchy of imported files will look like



5. Done. You can begin use this tool now.

NOTE! Before importing, make sure that none of the imported files do not overwrite the existing file in the project. Otherwise you may lose information .

4. Activation

Activation

Restrictions

Unregistered version has some restrictions

- You can not have more than 3 decals per scene.
- If you try to create more than 3 static decals then one will be deleted.
- Activation window will be apear periodically.

What you can use freely

You can freely use Dynamic and Dynamic Skinned decals.For this you just need store decals as prefabs in project and use those in scripting.

Activation

For activation select menu item**Decal Framework -> Activate** .Enter the email provided during registration and the serial number you received after purchasing .After activation you will get message about successful activation your copy of Decal Framework.You can register your copy only on one machine.

NOTE! If, after making a purchase you did not receive an activation key or are having difficulty with activation please contact us immediately.

5. Reactivation

Reactivation

Reactivation is the same as in Unity. If you need to install your copy on another machine, you must send an email to support@unity3dstore.com asking for an opportunity to activate your license on another machine .

In message must specify your serial number. After obtaining consent, you can activate your copy Decal Framework on another machine following the instructions in paragraph Activation .

6. Class Overview

Class Overview

DecalCreator

This is the main class for creating Dynamic and Dynamic Skinned decals. By creating scripts using the decals you always will be use his static methods. There are four main methods.

1. CreateDecalMesh(...) - This function creates a mesh (not the real object) of decals with all texture coordinates and tangent. After creating all the components of the mesh are in world space, thus easier for us to bring the mesh in any other space.
2. CreateDynamicDecal(...) - This function creates a real object decal . Any decal and Expeditors are always the child objects of the object from which they were born.
3. CreateDynamicSkinnedDecal(...) - This function the same as previous but creates decals on skinned objects.Because creating skinned decals not the same as creating usual decals.
4. CreateCombinedStaticDecalInGame() - This is helper function allow combine all uncombined Static decals before scene start.

DecalType

Main decal class. Here contains all setups for static and dynamic decals. In editor used for creating Static and Static Skinned decals.

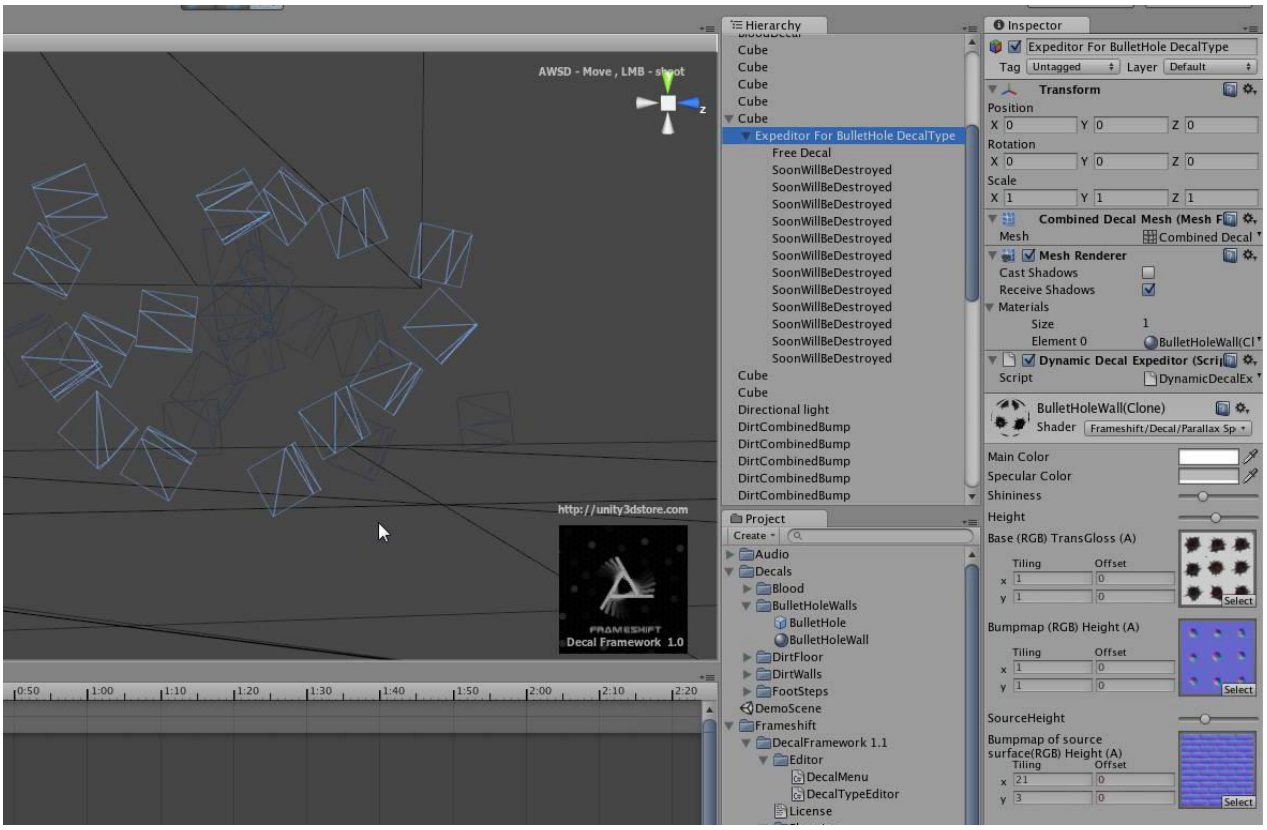
DecalHolder

Helper class for dynamic decals. The component is added to each object from which decal created. Contains dictionary DecalType->DecalExpeditor. And access methods to Expeditors.

1. GetExpeditor(...) - Get decal Expeditor certain DecalType.
2. GetAllExpeditors() - Get all Expeditors for all DecalTypes.

DynamicDecalExpeditor

Component controls the behavior of a certain type of decal on the object. It difines creation, combinning, deleting decals under settings DecalType , and memory clearing. In most of cases component carry out all you work. Mesh of this object always consist from all combinned decal meshes. All uncombined or sended to destroy decals will be child of this object.



SkinnedDecalExpeditor

Component similar to the previous but manages skinned decals .

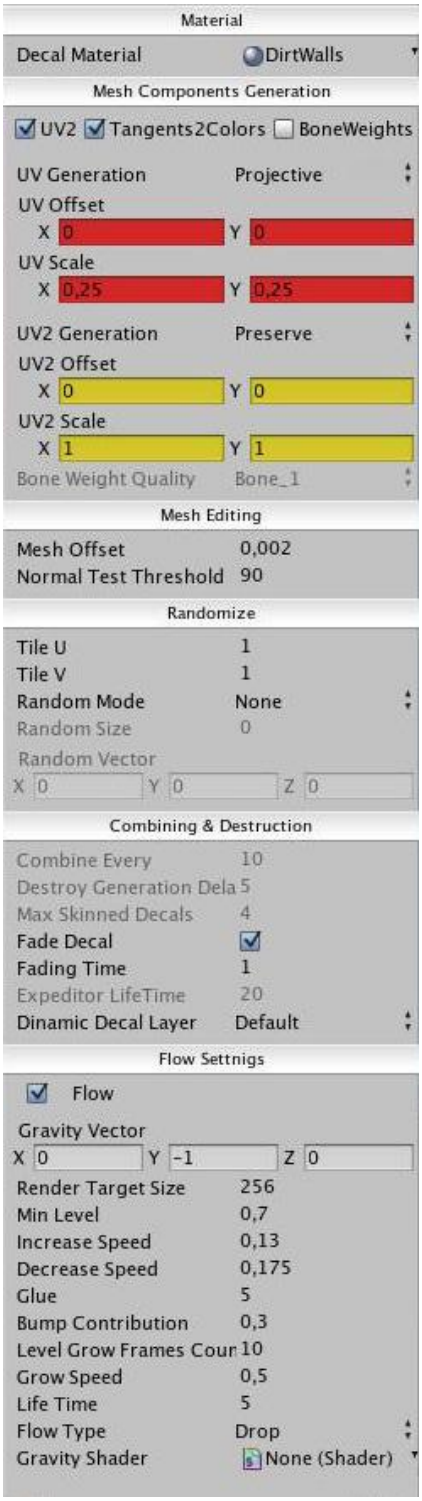
DecalDestroyer

Component manages decal wich will be destroyed soon.

7. Interface

7.1 Inspector

Inspector



DecalMaterial

Material of decal.

UV2

If **true** , will be generated second UVs.

Tangents2Colors

If **true** , source tangents will be stored (packed) in decal Mesh.colors array. This function used for combining bump of decal with bump of source surface.

BoneWeights

If **true**, will be generated BoneWeigths . This must be always true whan you make skinned decals.

UVGeneration

Type of UV generation for decal.

- Projective. Usual planar projection coords.
- Preserve. Preserve tex coords from source surface.
- Normalized . Preserve tex coords from source surface and normalize it.

UV2Generation

Type of second UV generation for decal.

UV Offset

Offset for tex coords. Similar to offset in material setups.

UV Scale

Scale for tex coords. Similar to tiling in material setups.

Bone Weight Quality

Skinning quality. Similar to Quality in SkinnedMeshRenderer component.

Mesh Offset

How much every vertex will be offset along normal direction. You can use this option to avoid artifacts of depth testing with Offset -1 -1 in your shader .

Normal Test Threshold

Very critical for the performance parameter. If angle between normal vertices and direction of decals will be more than this, then this vertex will not be taken into account when calculating the mesh decals. Thus the final stage of mesh generation, we will handle the lesser number of vertices and triangles. For example:

- For flatten decals as bullet holes it must be 10 degrees
- For blood splatters it must be much more that 10 degrees

Tile U / Tile V (Only for Dynamic , Dynamic Skinned Decals)

If you use a texture atlas for decals, you can use this option to install a random shift in the generation of texture coordinates. Each parameter specifies the texture number of samples in each direction, just as in the component particle renderer.

Random Mode (Only for Dynamic , Dynamic Skinned Decals)

The type of randomize for decal size.

- None - no randomize
- Evenly - randomize will be evenly in each direction X Y Z
- PerComponent - randomize will be per component in each direction X Y Z

Random Size (Only for Dynamic , Dynamic Skinned Decals)

Using with RandomMode.Evenly. Amount of randomize size +/-.

Random Vector (Only for Dynamic , Dynamic Skinned Decals)

Using with RandomMode.PerComponent . Amount of randomize size per component+/-.

Combine Every (Only for Dynamic Decals)

To improve performance decal Expeditor periodically combines all uncombined decals on the object. This parameter define how fast decals will be combined. I.e. after every decal creation if (AllDecalCount%CombineEvery= = 0) all uncombined decals will be combined in one mesh

NOTE! Combining of skinned decals happens after every decal creation.

Destroy Generation Delay (Only for Dynamic Decals)

This parameter indicates when decals will start destroy. I.e. If Combine Every = 10, and Destroy Generation Delay = 3 after the third combination of the first 10 decals will be sent to destroy .

If you want to combining perform every decal creation set CombineEvery = 1 and Destroy Generation Delay = 100 for example.

Max Skinned Decals (Only for Dynamic Skinned Decals)

Max count of skinned decals. Adding skinned decals performs as "Last In First Out". I.e for example if Max Skinned Decals = 4 than after 5 decals creation first decal will be destroyed, after 6 decals creation second decal will be destroyed etc...

Fade Decal (Only for Dynamic Decals)

If **true** , before destroy decal will be smooth faded. This works only with alpha blended shaders.

NOTE! Skinned decals can not be fade.

Fading Time (Only for Dynamic Decals)

Time for fading decal in sec.

Expeditor LifeTime (Only for Dynamic , Dynamic Skinned Decals)

Parameter shows how long Expeditor and all decals will be alive after last decal added. For example if Expeditor LifeTime = 60 than after 60 sec since last decal added Expeditor and all decals will be destroyed. After ecery decal creation timer is reset.

Dynamic Decal Layer **(Only for Dynamic , Dynamic Skinned Decals)**

Layed for dynamic decals and expeditors.

Flow **(Only for Dynamic)**

Set fluid settings enable.

Gravity Vector **(Only for Dynamic)**

Vector of gravity in world space.

Render Target Size **(Only for Dynamic)**

Size of render target wich used for rendering gravity map of fluid decal.

Min Level **(Only for Dynamic)**

Minimal level of fluid.

Increase Speed **(Only for Dynamic)**

How fast fluid increased due gravitation in particular pixel.

Decrease Speed (Only for Dynamic)

How fast fluid decreased due gravitation in particular pixel.

Glue (Only for Dynamic)

Fluid viscosity.

Bump contribution (Only for Dynamic)

How hard source bump will be affect on fluid flow.

Level grow frames count (Only for Dynamic)

How long level will be grow when decals created.

Grow speed (Only for Dynamic)

Grow increment per frame.

Life time (Only for Dynamic)

Fluid decal life time .

Flow Type (Only for Dynamic)

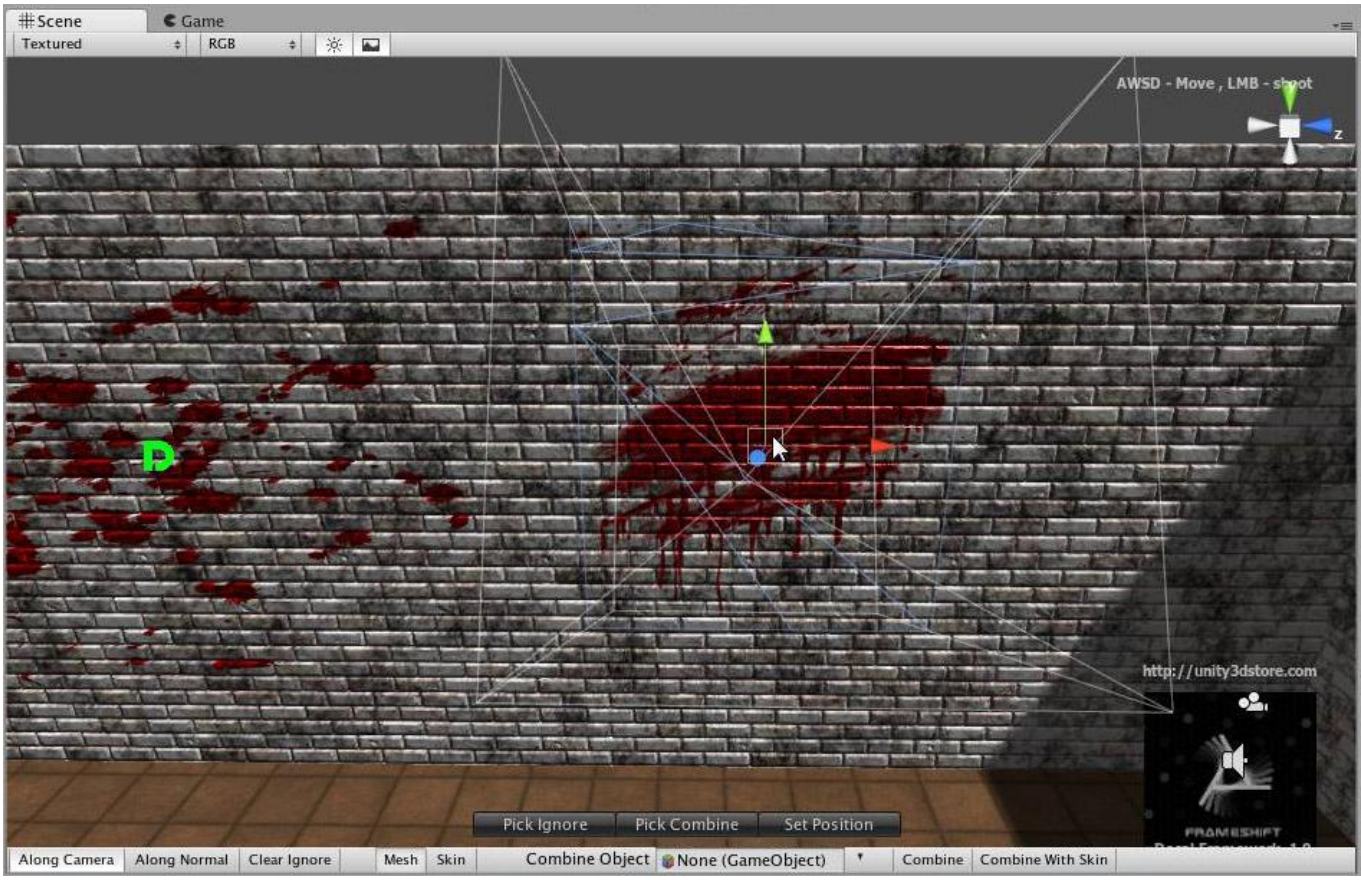
- Drop
- Spread

Gravity Shader(Only for Dynamic)

Shader for gravity map rendering .

7.2 Scene Editor

Scene Editor



Pick Ignore

Enter in ignore object selection mode. Those objects that will not be taken into acount during decal calculation.

Pick Combine

Enter in combine object selection mode.

Set Position

Enter in one click position mode. One click position system works via raycasting, thus you can set position for decals only onto objects that has collider attached.

Yuc can use Along Camera / Along Normal for more detail setup decal orientation.

Along Camera

If selected decal will be oriented along direction between cam position and position of decal.

Along Normal

If selected decal will be oriented along normal in point of raycast.

Clear Ignore

Clears ignore list.

Mesh

If selected decal will be projected onto usual meshes.

Skin

If selected decal will be projected onto skins.

Combine Object

Current object for combining.

Combine

Combine all selected decals on Combine Object. Decals will be combined by material. I.e. if you have 10 static decals and they use two materials, then you will get two combined objects for every materials separate. You can also set Combine Object = null , in this case decals will be combined in World Space.

Combine With Skin

Combine allselected decals on SkinnedMeshRenderer.All Decals will be added in Skinned Mesh. Combine Object can not be null when you use Combine With Skin.

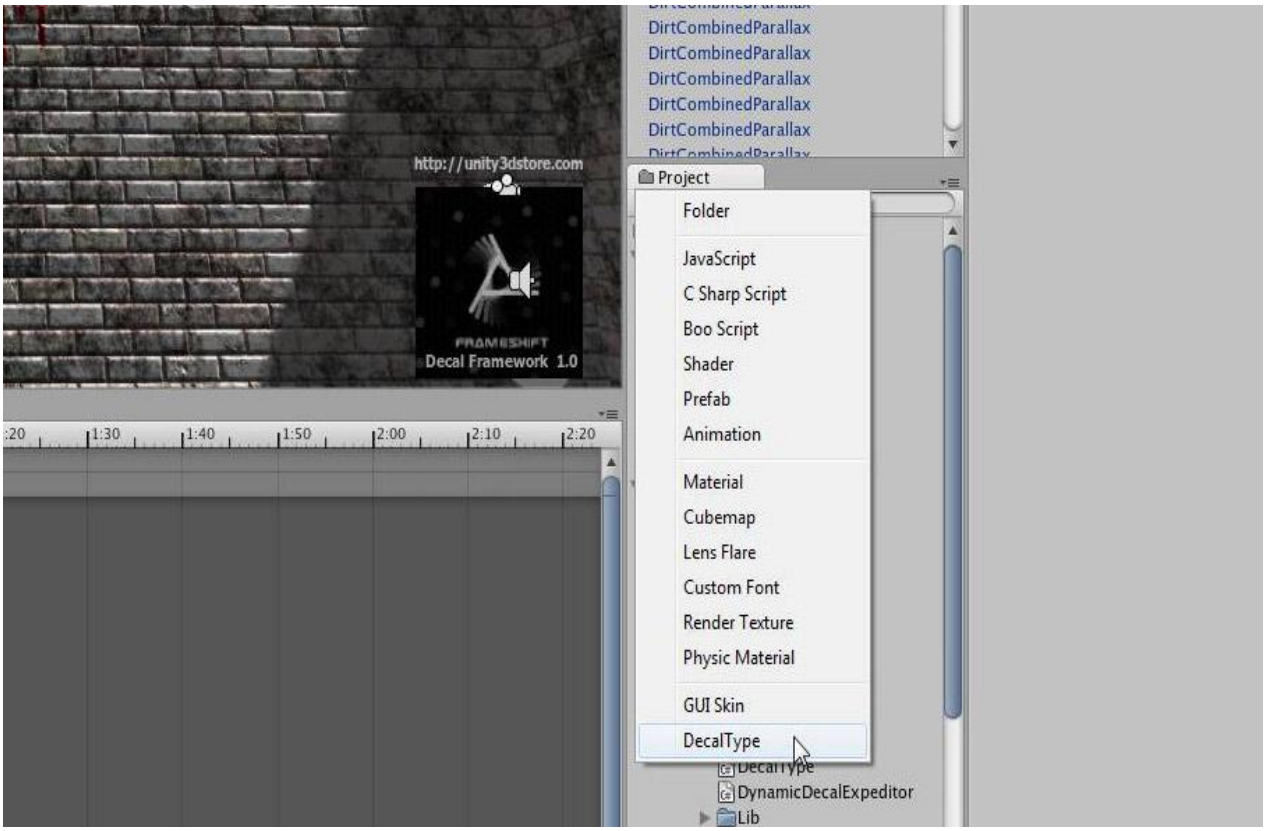
8. Step by step instructions

8.1 Creating Static Decals

Creating Static Decals

1. Open your project. If you don't imported yet Decal Framework in you project do it using page Installation.

2. Create decal as prefab using menu item **Project -> Create -> DecalType** .

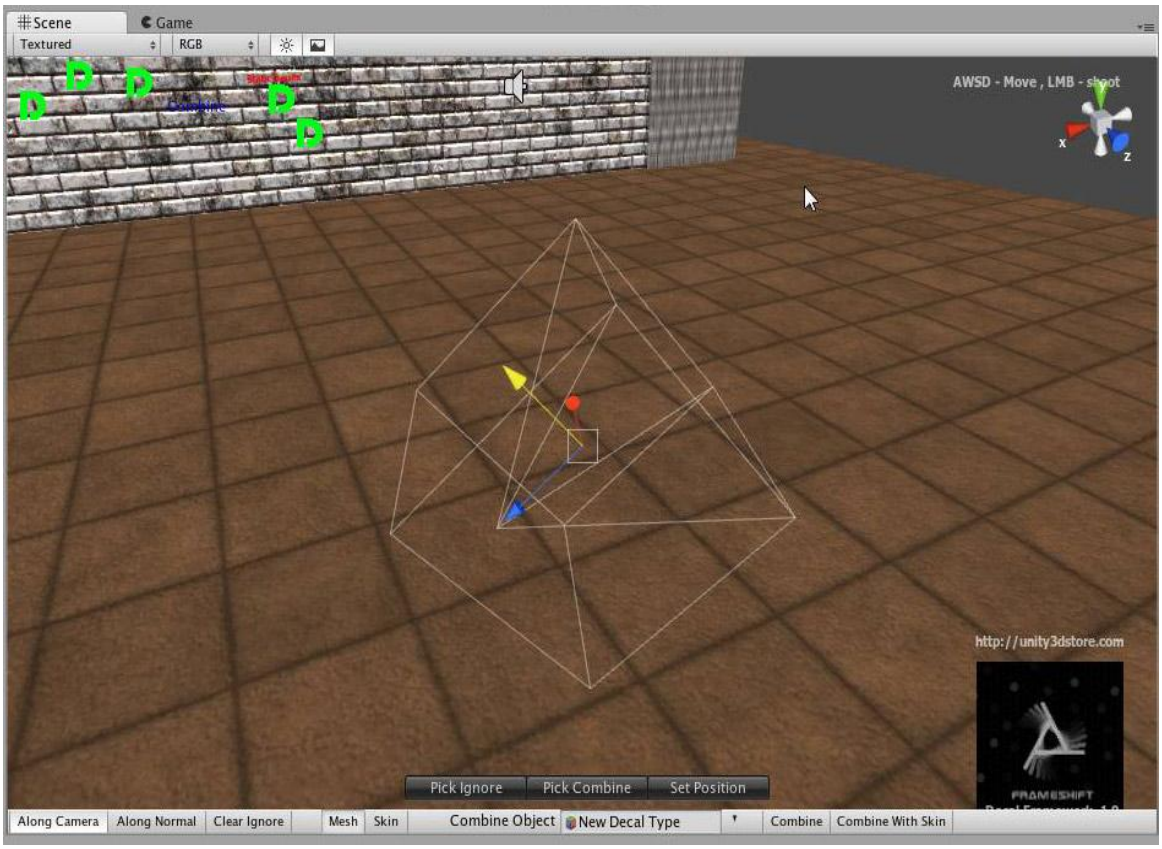


3. Select material for decal.

4. In top of Inspector will be shown textures that belong to decal material. This works only with _MainTex и _BumpMap textures.

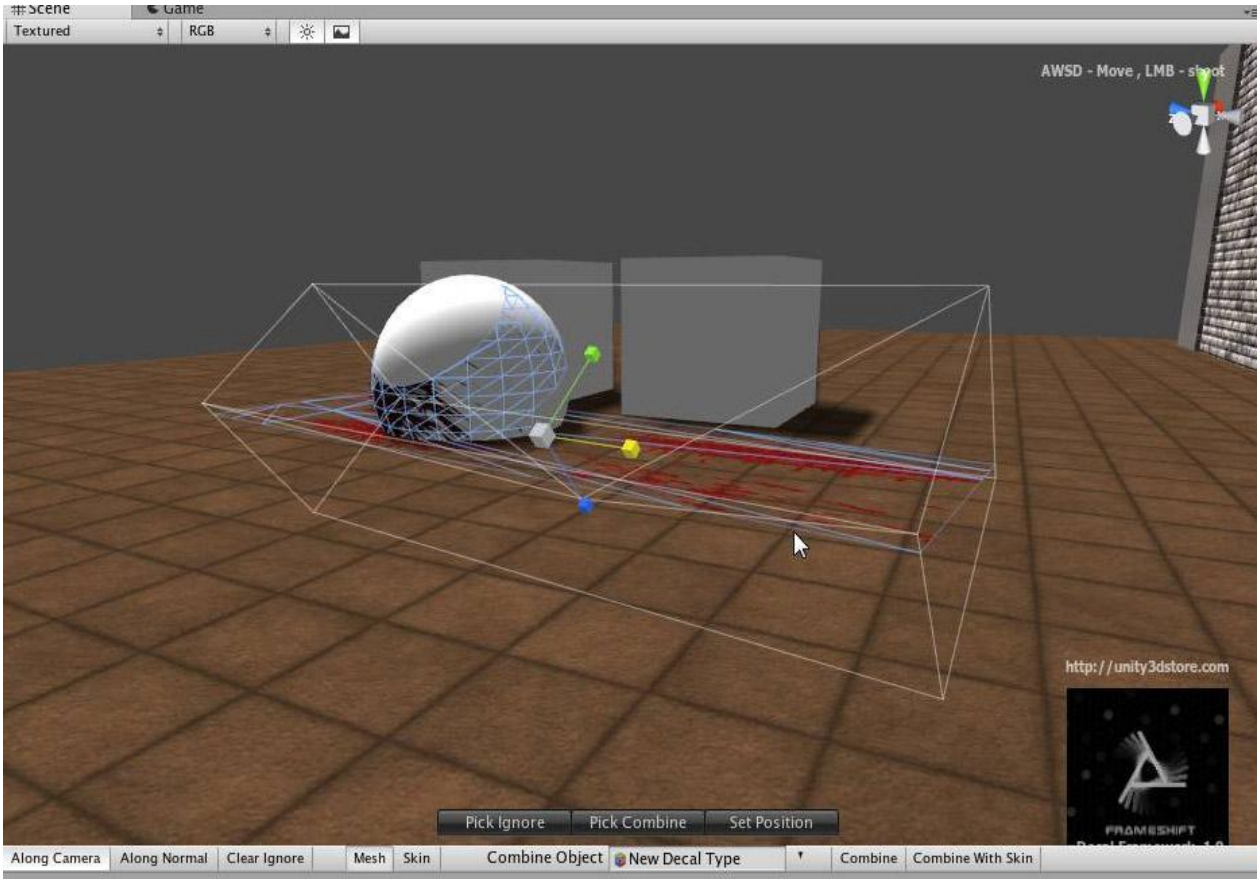
5. Decal parameters grouped it categories click on button to expand.

6. Instanse decal prefab into scene.

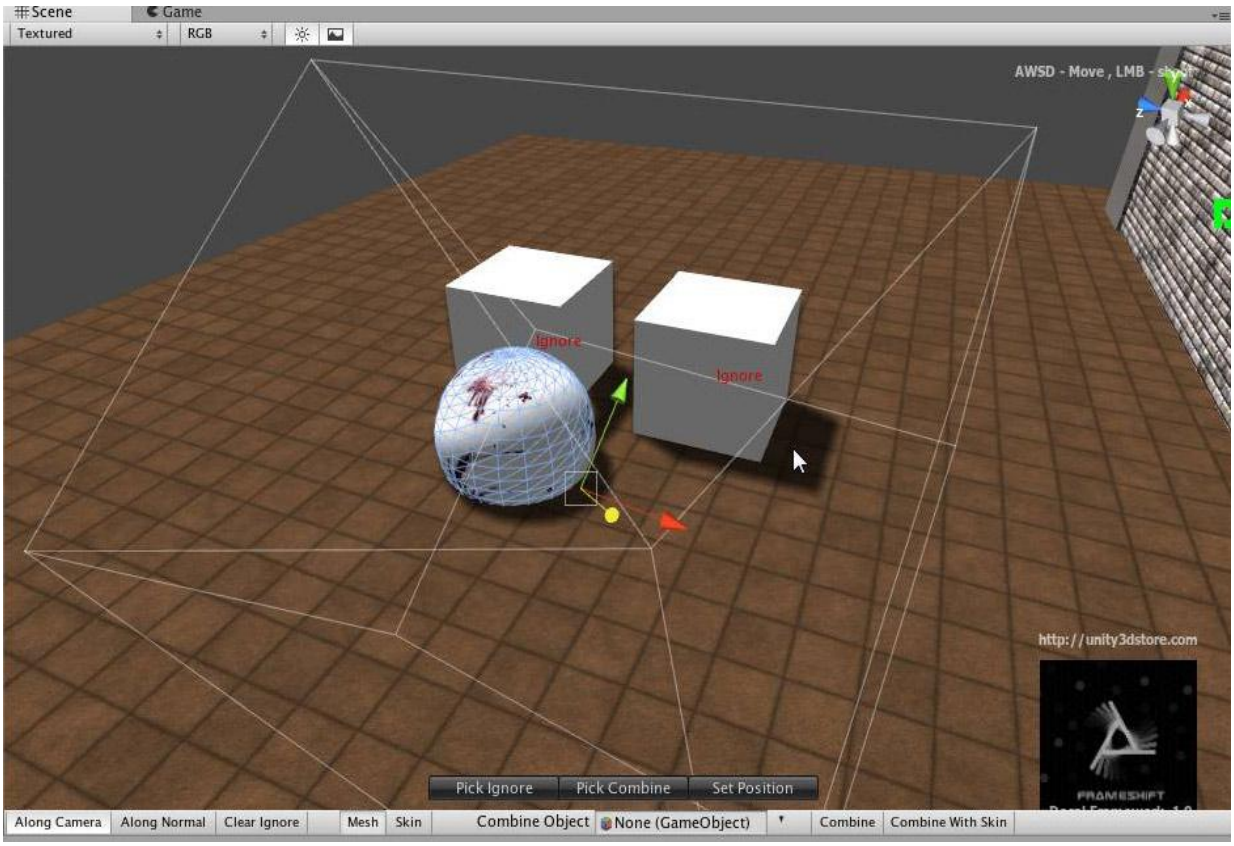


7. Set position and orientation of decal using one click position system or standart unity tools. For enter in one click position press Set Pisition. Then click onto collider. You can also use hotkeys. Hold down **"D"** , then click onto collider.

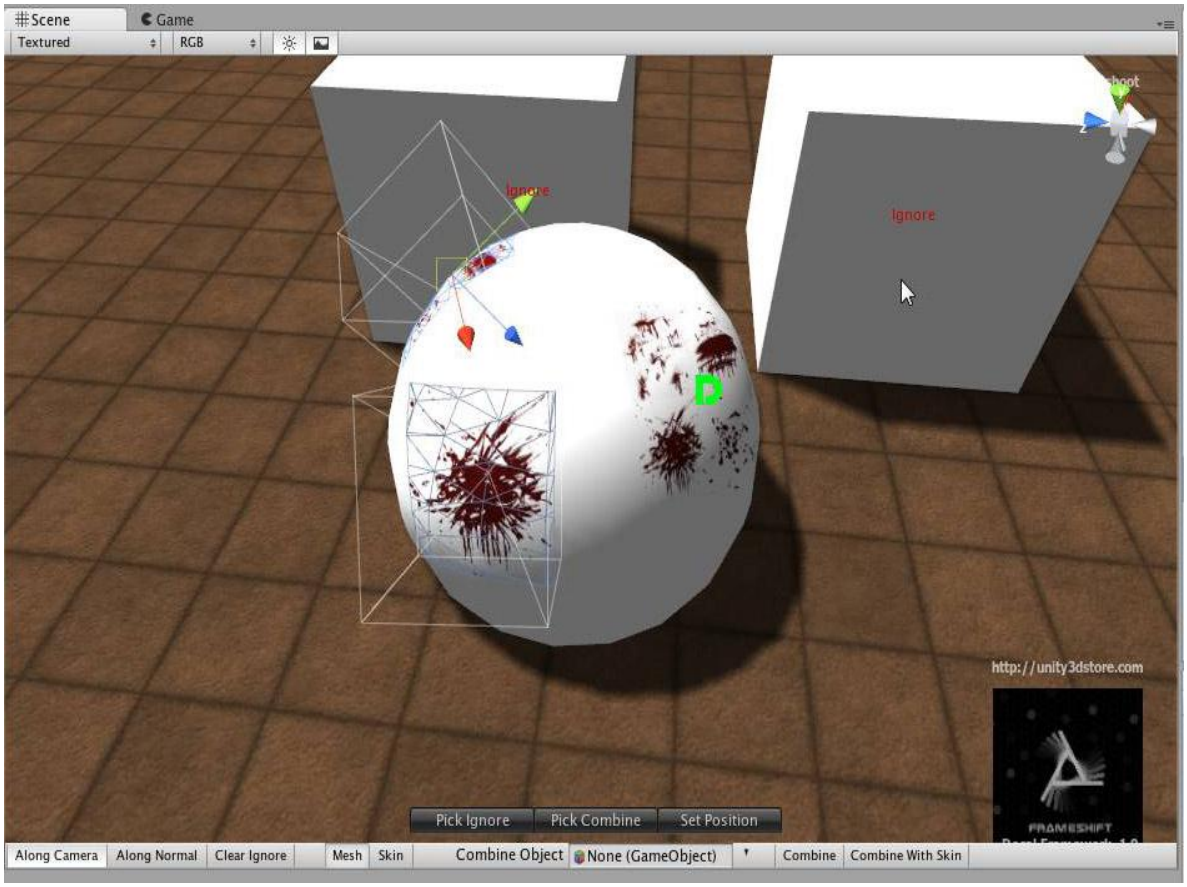
8. Using standart unity scale handle setup size of decal.



9. Select ignore objects. Enter to ignore mode press Pick Ignore. Select objects that should not be taken into acount. You can also use hotkeys. Hold down **"D+C"** , then select objects. Click once more in object if you want to remove it from ignore list. Ignored objects marked "Ignore".

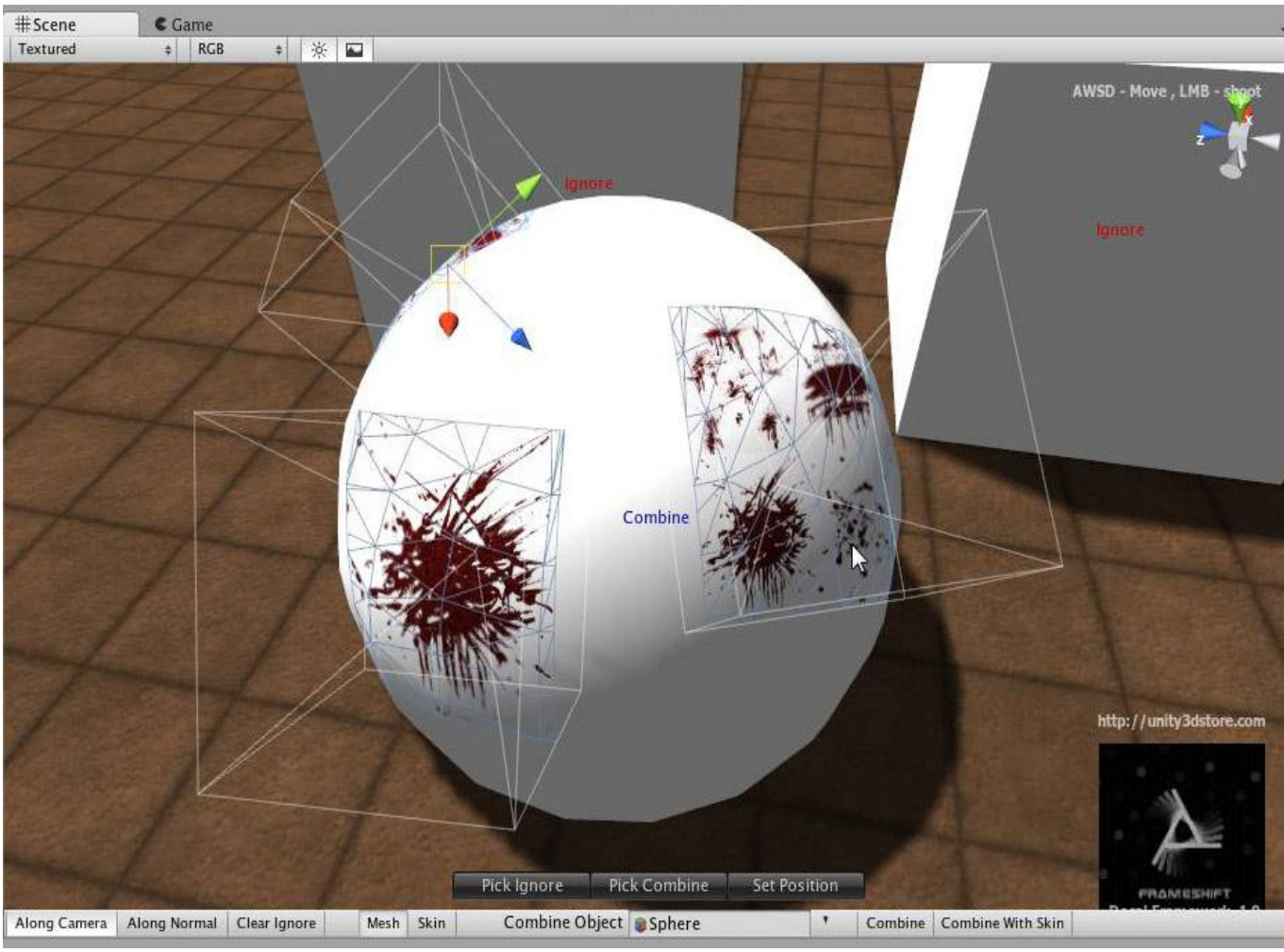


10. Create another decals.

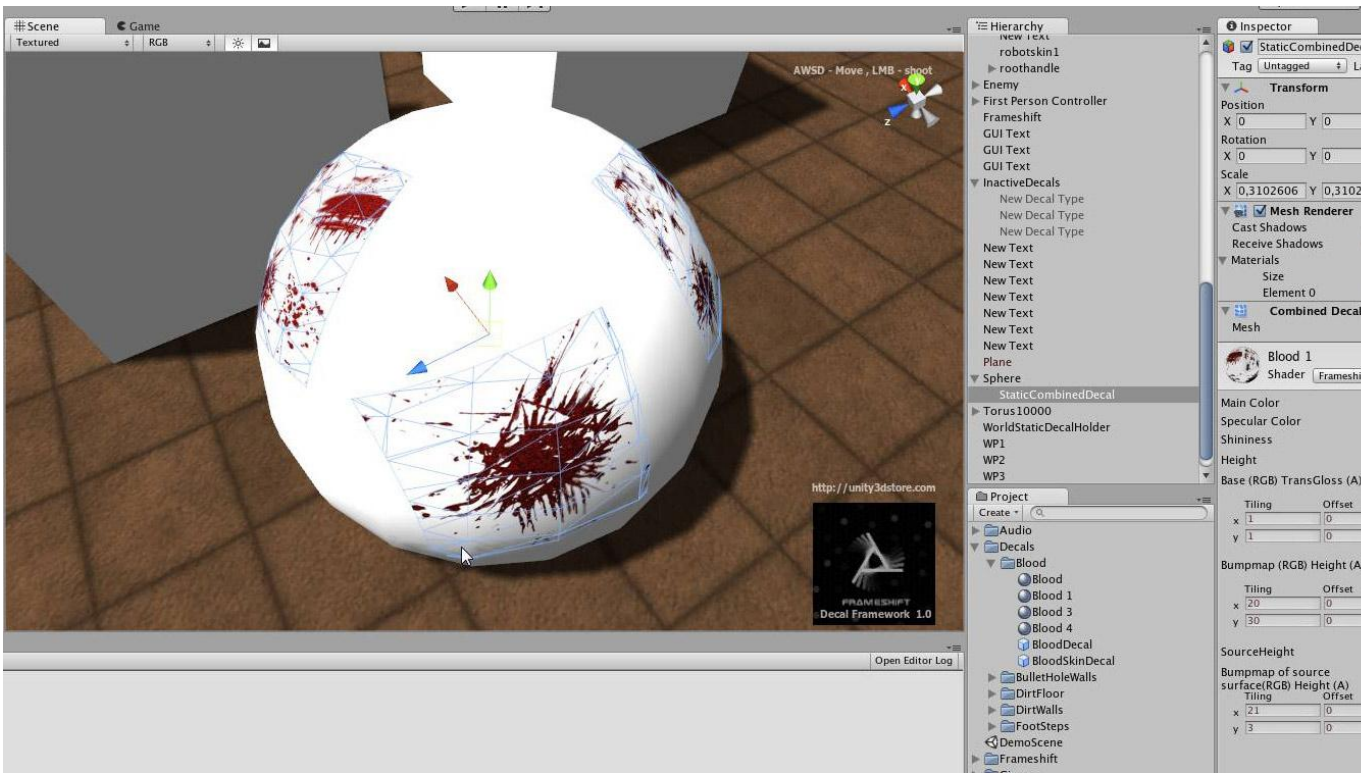


11. Now time to combine decals. Enter to select combine object mode press Pick Combine. Click onto object,the object will be marked "Combine" . Click once more if you

want to discard selection. You also can use hotkeys. Hold down **"D+S"** and selec object.



12. For combine decals press Combine. Chose will be decals destroyed or not. If you chose "NO" than decals will be enable = false and send to group InactiveDecals. After combining hierarhy of object will be as shown



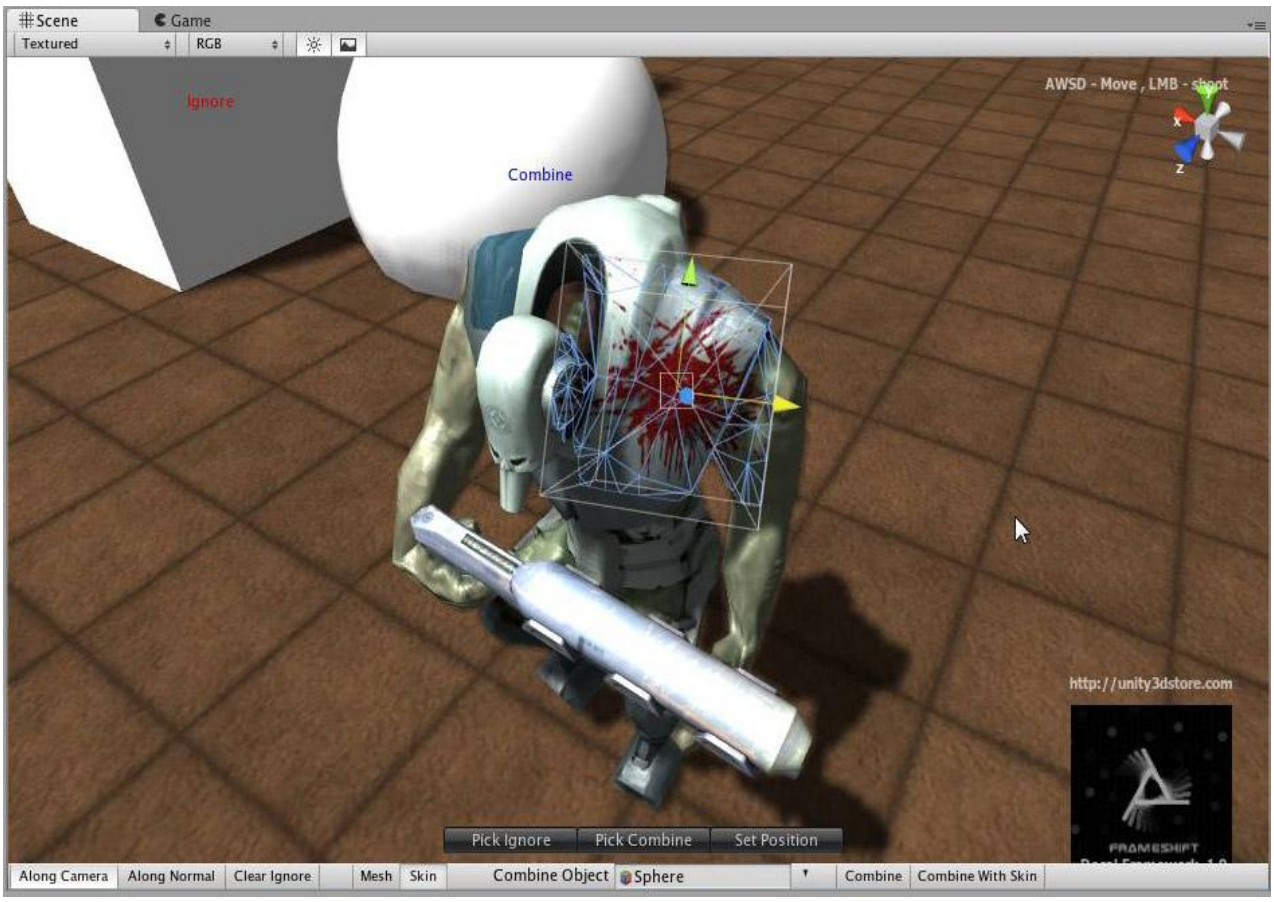
8.2 Creation Static Skinned Decals

Creating Static Skinned Decals

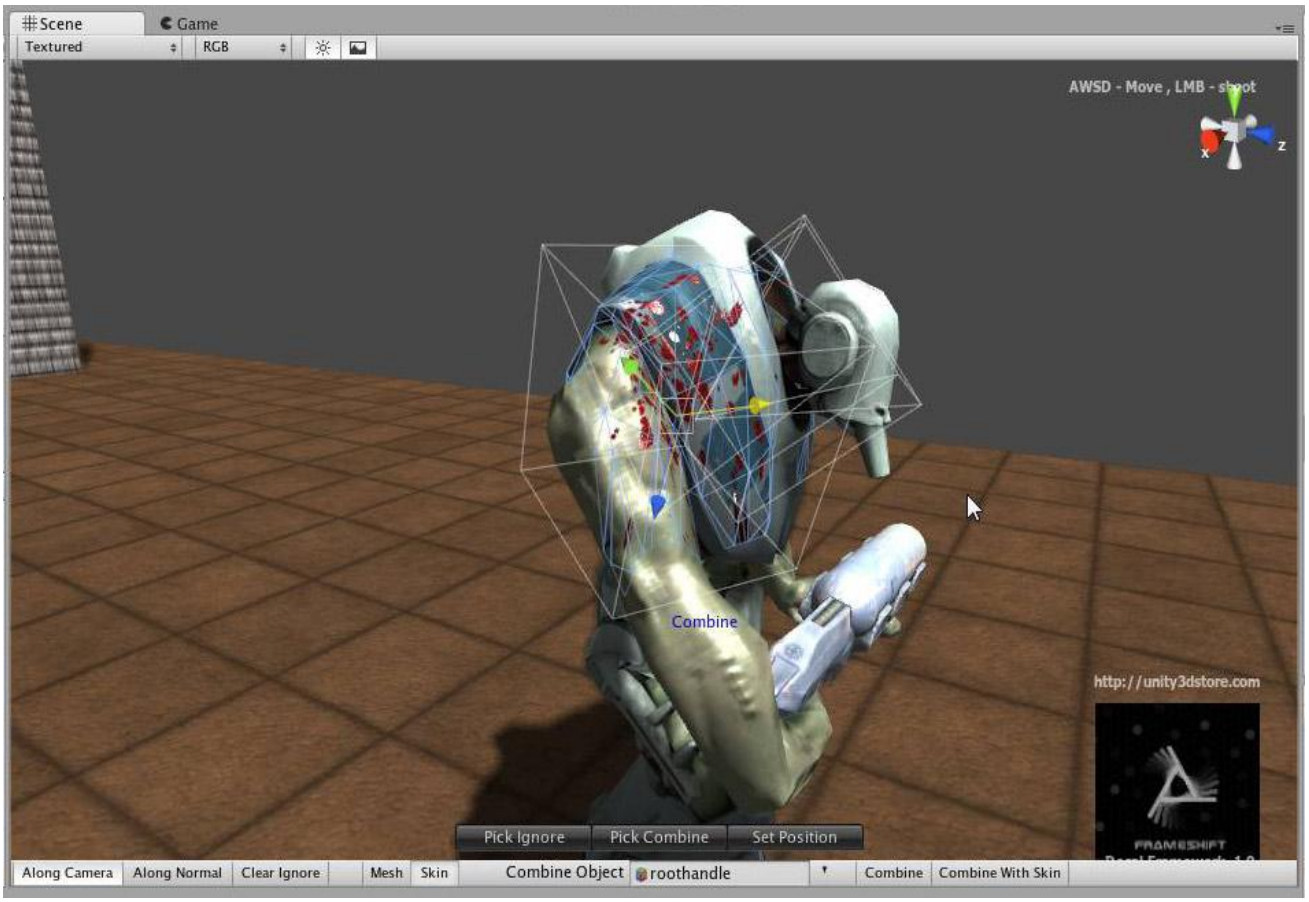
1. Follow stape 1-7 on page Create Static Decals .

2. Make sure that Skin toggle is Selected.

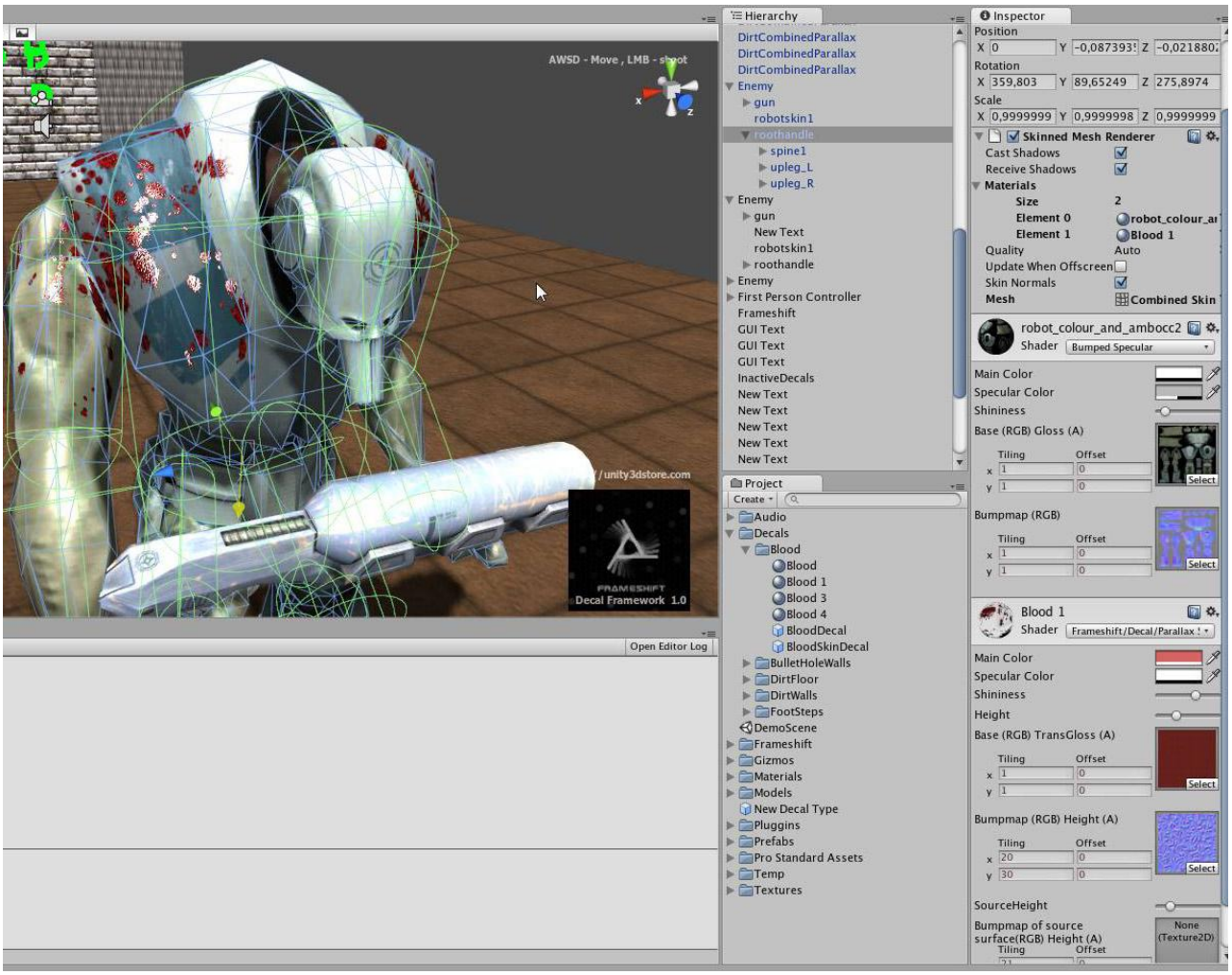
3. Set decal position.



4. Create another decals. Select Combine object. Combine object must be SkinnedMeshRenderer.



5. Press **Combine With Skin** . After combining hierarhy of object will be as shown



9. Gizmos

Gizmos

Unselected Gizmo

Unselected Gizmo it just "D" helps separate decals from another objects. You can use you own image, just rewrite file **DecalIcon** in folder **Assets/Gizmos** .

Selected Gizmo

Selected Gizmo it white wire cube bounds decal volume and white wire arrow. You can override this function :

```
public static void DrawDecalToolGizmo(DecalType _decalType, GizmoType gizmoType)
```

in script **DecalTypeEditor** , for make your own appropriate gizmo.