DecalFramework Reference

Frameshift.Decal Namespace 3
DecalCreator Class 3
CreateCombinedStaticDecalInGame Method 3
CreateDecalMesh Method (Overloaded) 3
CreateDynamicDecal Method (Overloaded) 9
CreateDynamicSkinnedDecal Method (Overloaded) 11
CreateFluidDecal Method (Overloaded) 12
DecalHolder Class 15
GetAllExpeditors Method 15
GetExpeditor Method 15

Frameshift.Decal Namespace

Classes

DecalCreator, DecalHolder

DecalCreator Class

Main Decal creator class

```
[C#]
public class DecalCreator : MonoBehaviour
```

Requirements

Namespace: Frameshift.Decal
Platforms: Unity 2.6.1
Assembly: Frameshift.Decal (in Frameshift.Decal.dll)

Methods

CreateCombinedStaticDecalInGame, CreateDecalMesh, CreateDynamicDecal, CreateDynamicSkinnedDecal, CreateFluidDecal

DecalCreator.CreateCombinedStaticDecalInGame Method

Create combined meshes and GameObjects for all uncombined Static Decals

```
[C#]
public static void CreateCombinedStaticDecalInGame();
```

Example

```
private void Start()
    {
        //Combine all uncombined Decals
        DecalCreator.CreateCombinedStaticDecalInGame();
    }
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDecalMesh (Frameshift.Decal.DecalType,

UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.Collider()) Method

Create Decal mesh from colliders array.

```
[C#]
public static Mesh CreateDecalMesh(
    DecalType decalType,
    Vector3 point,
    Vector3 forward,
    Collider[] colliders
);
```

Parameters

decalType
Type of Decal that will be created.
point
Point in world space where Decal will be calculated.
forward
Direction of decal. Usually -hit.normal.
colliders
Array colliders for which are trying to create decal.

Returns

Decal mesh in world space.

Example

```
//Find colliders near raycast hit point
Collider[] colliders=Physics.OverlapSphere(hit.position, 0.3F);
//Burn decal mesh
Mesh decalMesh = DecalCreator.CreateDecalMesh(i_decalType,hit.position,hit.direction,
colliders);
//Create decal object
DecalCreator.CreateDynamicDecal(decalMesh, hit.collider, i_decalType);
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.Collider(), UnityEngine.Vector3) Method

Create Decal mesh from colliders array. Set directly orientation.

```
[C#]
public static Mesh CreateDecalMesh(
    DecalType decalType,
    Vector3 point,
    Vector3 forward,
    Collider[] colliders,
    Vector3 decalWoldUpVector
);
```

Parameters

decalType
Type of Decal that will be created.
point
Point in world space where Decal will be calculated.
forward
Direction of decal. Usually -hit.normal.
colliders
Array colliders for which are trying to create decal.
decalWoldUpVector
Decal world up vector, i.e. where top of decal mesh will be look.

Returns

Decal mesh in world space.

Example

```
//Find colliders near raycast hit point
Collider[] colliders=Physics.OverlapSphere(hit.position, 0.3F);
//Burn decal mesh with vertical orientation
Mesh decalMesh = DecalCreator.CreateDecalMesh(i_decalType,hit.position,hit.direction,
colliders, Vector3.up);
//Create decal object
DecalCreator.CreateDynamicDecal(decalMesh, hit.collider, i_decalType);
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject) Method

Create Decal mesh from GameObject.

```
[C#]
public static Mesh CreateDecalMesh(
    DecalType decalType,
    Vector3 point,
    Vector3 forward,
    GameObject obj
);
```

Parameters

decalType
Type of Decal that will be created.
point
Point in world space where Decal will be calculated.
forward
Direction of decal. Usually -hit.normal.
obj
GameObject on which Decal will be created.

Returns

Decal mesh in world space.

Example

```
//Burn decal
Mesh decalMesh = DecalCreator.CreateDecalMesh(i_decalType, hit.point,
-hit.normal, hit.collider.gameObject);
//Create Decal Object
DecalCreator.CreateDynamicDecal(decalMesh, hit.collider.gameObject,
i_decalType);
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3,

UnityEngine.GameObject, UnityEngine.Vector3) Method

Create Decal mesh from GameObject. Set directly orientation.

```
[C#]
public static Mesh CreateDecalMesh(
    DecalType decalType,
    Vector3 point,
    Vector3 forward,
    GameObject obj,
    Vector3 decalWoldUpVector
);
```

Parameters

decalType
Type of Decal that will be created.
point
Point in world space where Decal will be calculated.
forward
Direction of decal. Usually -hit.normal.
obj
GameObject on which Decal will be created.
decalWoldUpVector
Decal world up vector, i.e. where top of decal mesh will be look.

Returns

Decal mesh in world space.

Example

```
//Burn decal with vertical orientation
Mesh decalMesh = DecalCreator.CreateDecalMesh(i_decalType, hit.point,
-hit.normal, hit.collider.gameObject, Vector3.up);
//Create Decal Object
DecalCreator.CreateDynamicDecal(decalMesh, hit.collider.gameObject,
i_decalType);
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDynamicDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType, UnityEngine.Material) Method

Create Decal GameObject with material override.

```
[C#]
public static GameObject CreateDynamicDecal(
    Mesh decalMesh,
    GameObject obj,
    DecalType decalType,
    Material materialOverride
);
```

Parameters

decalMesh
Decal mesh in world space
obj
Parent(Holder) for Decal
decalType
Type of this Decal
materialOverride
Material override for Decal

Returns

DecalExpeditor for this decalType on this obj

Example

```
RaycastHit hit;
Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));
bool wasHit = Physics.Raycast(ray, out hit);
if (wasHit)
{
    Material m = null;
    if (hit.collider.gameObject.renderer)
    {
        //Get material instanse
        m = Instantiate(i_decalType.i_material) as Material;
        //Get bump from hited surface
        Texture2D bumpMap =
hit.collider.gameObject.renderer.sharedMaterial.GetTexture("_BumpMap") as
Texture2D;
        Vector2 bumpScale =
hit.collider.gameObject.renderer.sharedMaterial.GetTextureScale("_BumpMap");
        Vector2 bumpOffset =
hit.collider.gameObject.renderer.sharedMaterial.GetTextureOffset("_BumpMap");
        //Setup new bump
        m.SetTexture("_SourceBumpMap", bumpMap);
        m.SetTextureScale("_SourceBumpMap", bumpScale);
        m.SetTextureOffset("_SourceBumpMap", bumpOffset);

        //Burn decal
        Mesh decalMesh = DecalCreator.CreateDecalMesh(i_decalType, hit.point,
-hit.normal, hit.collider.gameObject);
        //Create Decal Object
```

```
        DecalCreator.CreateDynamicDecal(decalMesh, hit.collider.gameObject,
i_decalType, m);

    }
}
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDynamicDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType) Method

Create Decal GameObject.

```
[C#]
public static GameObject CreateDynamicDecal(
    Mesh decalMesh,
    GameObject obj,
    DecalType decalType
);
```

Parameters

decalMesh
Decal mesh in world space

obj
Parent(Holder) for Decal

decalType
Type of this Decal

Returns

DecalExpeditor for this decalType on this obj

Example

```
RaycastHit hit;
Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));
bool wasHit = Physics.Raycast(ray, out hit);
if (wasHit)
{
    if (hit.collider.gameObject.renderer)
    {
        //Burn decal
        Mesh decalMesh = DecalCreator.CreateDecalMesh(i_decalType, hit.point,
-hit.normal, hit.collider.gameObject);
        //Create Decal Object
        DecalCreator.CreateDynamicDecal(decalMesh, hit.collider.gameObject,
i_decalType);

    }
}
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDynamicSkinnedDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType, UnityEngine.Material) Method

Create dynamic skinned Decal GameObject with material override

```
[C#]
public static GameObject CreateDynamicSkinnedDecal(
    Mesh decalMesh,
    GameObject obj,
    DecalType decalType,
    Material materialOverride
);
```

Parameters

decalMesh

Decal mesh in world space

obj

GameObject with SkinnedMeshRenderer attached

decalType

Type of this Decal

materialOverride

Material override for Decal

Returns

DecalExpeditor for this decalType on this obj

Example

```
RaycastHit hit;
Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));
bool wasHit = Physics.Raycast(ray, out hit);
if (wasHit)
{
     //If we hit character
     if (hit.collider.transform.root.name == "Enemy")
     {
          //Find SkinnedMeshRenderer
          SkinnedMeshRenderer smr =
hit.collider.transform.root.GetComponentInChildren<SkinnedMeshRenderer>();
          //Burn DecalMesh
          Mesh decalMesh=DecalCreator.CreateDecalMesh(i_blood, hit.point, -hit.normal,
smr.gameObject, Vector3.zero);
          //Create Skinned Decal
          DecalCreator.CreateDynamicSkinnedDecal(decalMesh, smr.gameObject, i_blood);
     }
}
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateDynamicSkinnedDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType) Method

Create dynamic skinned Decal GameObject

```
[C#]
public static GameObject CreateDynamicSkinnedDecal(
    Mesh decalMesh,
    GameObject obj,
    DecalType decalType
);
```

Parameters

decalMesh

Decal mesh in world space

obj

GameObject with SkinnedMeshRenderer attached

decalType

Type of this Decal

Returns

DecalExpeditor for this decalType on this obj

Example

```
RaycastHit hit;
Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));
bool wasHit = Physics.Raycast(ray, out hit);
if (wasHit)
{
     //If we hit character
     if (hit.collider.transform.root.name == "Enemy")
     {
         //Find SkinnedMeshRenderer
         SkinnedMeshRenderer smr =
hit.collider.transform.root.GetComponentInChildren<SkinnedMeshRenderer>();
         //Burn DecalMesh
         Mesh decalMesh=DecalCreator.CreateDecalMesh(i_blood, hit.point, -hit.normal,
smr.gameObject, Vector3.zero);
         //Create Skinned Decal
         DecalCreator.CreateDynamicSkinnedDecal(decalMesh, smr.gameObject, i_blood);
     }
}
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateFluidDecal (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject, UnityEngine.Material) Method

Creates fluid decal mesh and fluid game object with render sub-system

```
[C#]
public static GameObject CreateFluidDecal(
    DecalType decalType,
    Vector3 point,
    Vector3 forward,
    GameObject obj,
    Material materialOverride
);
```

Parameters

decalType
Type of Decal that will be created.

point
Point in world space where Decal will be calculated.

forward
Direction of decal. Usually -hit.normal.

obj
GameObject on which Decal will be created.

materialOverride
Material override for Decal

Returns

Decal Object and render sub-system

Example

```
RaycastHit hit;
Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));
bool wasHit = Physics.Raycast(ray, out hit);

 if (wasHit)
        {
            Material m = null;
            if (hit.collider.gameObject.renderer)
            {
                //Get material instanse
                m = Instantiate(i_flow.i_material) as Material;
```

```
                //Get bump from hited surface

if(hit.collider.gameObject.renderer.sharedMaterial.HasProperty("_BumpMap"))
                {
                    Texture2D bumpMap =
hit.collider.gameObject.renderer.sharedMaterial.GetTexture("_BumpMap") as
Texture2D;
                    Vector2 bumpScale =
hit.collider.gameObject.renderer.sharedMaterial.GetTextureScale("_BumpMap");
                    Vector2 bumpOffset =
hit.collider.gameObject.renderer.sharedMaterial.GetTextureOffset("_BumpMap");
                    //Setup new bump
                    m.SetTexture("_SourceBumpMap", bumpMap);
                    m.SetTextureScale("_SourceBumpMap", bumpScale);
                    m.SetTextureOffset("_SourceBumpMap", bumpOffset);
                  }
                else
                {
                     m.SetTexture("_SourceBumpMap", null);
                }
                //Flow decal
                DecalCreator.CreateFluidDecal(i_flow, hit.point, ray.direction,
hit.collider.gameObject, m);
            }
 }
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalCreator.CreateFluidDecal (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject, UnityEngine.Vector3, UnityEngine.Material) Method

Creates fluid decal mesh and fluid game object with render sub-system, set directly orientation

```
[C#]
public static GameObject CreateFluidDecal(
    DecalType decalType,
    Vector3 point,
    Vector3 forward,
    GameObject obj,
    Vector3 decalWoldUpVector,
    Material materialOverride
);
```

Parameters

decalType
Type of Decal that will be created.
point
Point in world space where Decal will be calculated.
forward
Direction of decal. Usually -hit.normal.
obj
GameObject on which Decal will be created.
decalWoldUpVector
Decal world up vector, i.e. where top of decal mesh will be look.
materialOverride
Material override for Decal

Returns

Decal Object and render sub-system

Example

```
RaycastHit hit;
Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));
bool wasHit = Physics.Raycast(ray, out hit);

 if (wasHit)
        {
            Material m = null;
            if (hit.collider.gameObject.renderer)
            {
                //Get material instanse
                m = Instantiate(i_flow.i_material) as Material;

                //Get bump from hited surface

if(hit.collider.gameObject.renderer.sharedMaterial.HasProperty("_BumpMap"))
                {
                    Texture2D bumpMap =
hit.collider.gameObject.renderer.sharedMaterial.GetTexture("_BumpMap") as
Texture2D;
                    Vector2 bumpScale =
hit.collider.gameObject.renderer.sharedMaterial.GetTextureScale("_BumpMap");
                    Vector2 bumpOffset =
hit.collider.gameObject.renderer.sharedMaterial.GetTextureOffset("_BumpMap");
                    //Setup new bump
                    m.SetTexture("_SourceBumpMap", bumpMap);
                    m.SetTextureScale("_SourceBumpMap", bumpScale);
                    m.SetTextureOffset("_SourceBumpMap", bumpOffset);
                  }
                else
                {
                     m.SetTexture("_SourceBumpMap", null);
                }
                //Flow decal
                DecalCreator.CreateFluidDecal(i_flow, hit.point, ray.direction,
hit.collider.gameObject, m);
            }
 }
```

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalCreator

DecalHolder Class

Holder (parent) for all DecalExpeditors on certain GameObject

```
[C#]
public class DecalHolder : MonoBehaviour
```

Requirements

Namespace: Frameshift.Decal
Platforms: Unity 2.6.1
Assembly: Frameshift.Decal (in Frameshift.Decal.dll)

Methods

GetAllExpeditors, GetExpeditor

DecalHolder.GetAllExpeditors Method

Get all DecalExpeditors on this DecalHolder (parented to this GameObject)

```
[C#]
public GameObject[] GetAllExpeditors();
```

Returns

All DecalExpeditors (parents) for all DecalTypes on this GameObject (Holder).

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalHolder

DecalHolder.GetExpeditor Method

Get certain DecalExpeditor on this DecalHolder (parented to this GameObject)

```
[C#]
public GameObject GetExpeditor(
    DecalType decalType
);
```

Parameters

decalType
DecalType for search DecalExpeditor

Returns

DecalExpeditor (parent) for all Decals of type decalType on this GameObject (Holder).

Requirements

Platforms: Unity 2.6.1

See Also

Applies to: DecalHolder

Index

CreateCombinedStaticDecalInGame Method, 3
CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.Collider()) Method, 3
CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.Collider(), UnityEngine.Vector3) Method, 5
CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject) Method, 6
CreateDecalMesh (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject, UnityEngine.Vector3) Method, 6
CreateDynamicDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType) Method, 9
CreateDynamicDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType, UnityEngine.Material) Method, 8
CreateDynamicSkinnedDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType) Method, 11
CreateDynamicSkinnedDecal (UnityEngine.Mesh, UnityEngine.GameObject, Frameshift.Decal.DecalType, UnityEngine.Material) Method, 10
CreateFluidDecal (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject, UnityEngine.Material) Method, 12
CreateFluidDecal (Frameshift.Decal.DecalType, UnityEngine.Vector3, UnityEngine.Vector3, UnityEngine.GameObject, UnityEngine.Vector3, UnityEngine.Material) Method, 13
DecalCreator Class, 3
DecalHolder Class, 15
Frameshift.Decal Namespace, 3
GetAllExpeditors Method, 15
GetExpeditor Method, 15