API Docs

Ferr.SuperCube

- Ferr.SuperCube.Create
- Ferr.SuperCube.CreatePivot

Ferr.SuperPlane

- Ferr.SuperPlane.Create
- Ferr.SuperPlane.CreatePivot
- Ferr.SuperPlane.CreateRectXZ
- Ferr.SuperPlane.CreateRectXY

All important attributes and methods are exposed to you, and come complete with thorough comment docs your IDE can display through intellisense. Included in this document are only the static helper functions included for easy creation of Ferr SuperCube objects!

Ferr.SuperCube.Create

```
public static GameObject Create(
      Vector3   aAt,
      Vector3   aSize,
      Material  aMaterial,
      UVType    aWallUV         = UVType.WorldCoordinates,
      UVType    aTopBottomUV    = UVType.WorldCoordinates,
      float     aSliceDistance  = 0,
      PivotType aHideFaces      = PivotType.None,
      Material  aOverrideTop    = null,
      Material  aOverrideBottom = null,
      Material  aOverrideLeft   = null,
      Material  aOverrideRight  = null,
      Material  aOverrideFront  = null,
      Material  aOverrideBack   = null )
```

Description

Creates a SuperCube game object, and assigns the given data! This function will build the cube mesh right away.
Returns a GameObject named "SuperCube" with a fully built SuperCube, MeshFilter, Renderer, and BoxCollider component!

Parameters

aAt	The center of the SuperCube.
aSize	The width, height, and depth of the SuperCube.

aMaterial	The default material for the entire SuperCube.
[aWallUV]	The type of UV calculations to use for the +X, -X, +Z, -Z faces.
[aTopBottomUV]	The type of UV calculations to use for the +Y, -Y faces.
[aSliceDistance]	How far apart shall we try and place vertex slices on the faces? floor(size/sliceDistance)
[aHideFaces]	A bit mask describing which faces should be hidden.
[aOverrideTop]	Material override for the +Y face.
[aOverrideBottom]	Material override for the -Y face.
[aOverrideLeft]	Material override for the -X face.
[aOverrideRight]	Material override for the +X face.
[aOverrideFront]	Material override for the -Z face.
[aOverrideBack]	Material override for the +Z face.

Example

```
GameObject cube = Ferr.SuperCube.Create(
      Vector3.zero,
      Vector3.one,
      material);
cube.transform.parent = transform;
```

Ferr.SuperCube.CreatePivot

```
public static GameObject CreatePivot(
      Vector3   aPivotPt,
      PivotType aPivotType,
      Vector3   aSize,
      Material  aMaterial,
      UVType    aWallUV         = UVType.WorldCoordinates,
      UVType    aTopBottomUV    = UVType.WorldCoordinates,
      float     aSliceDistance  = 0,
      PivotType aHideFaces      = PivotType.None,
      Material  aOverrideTop    = null,
      Material  aOverrideBottom = null,
      Material  aOverrideLeft   = null,
      Material  aOverrideRight  = null,
      Material  aOverrideFront  = null,
```

```
      Material  aOverrideBack   = null )
```

Description

Creates a SuperCube game object using a pivot point to determine location, and assigns the given data! This function will build the cube mesh right away.
Returns a GameObject named "SuperCube" with a fully built SuperCube, MeshFilter, Renderer, and BoxCollider component!

Parameters

aPivotPt	Location in space for the pivot point to be placed.
aPivotType	A bit mask defining where on the cube the pivot is placed. Opposing sides indicate to center along that axis.
aSize	The width, height, and depth of the SuperCube.
aMaterial	The default material for the entire SuperCube.
[aWallUV]	The type of UV calculations to use for the +X, -X, +Z, -Z faces.
[aTopBottomUV]	The type of UV calculations to use for the +Y, -Y faces.
[aSliceDistance]	How far apart shall we try and place vertex slices on the faces? floor(size/sliceDistance)
[aHideFaces]	A bit mask describing which faces should be hidden.
[aOverrideTop]	Material override for the +Y face.
[aOverrideBottom]	Material override for the -Y face.
[aOverrideLeft]	Material override for the -X face.
[aOverrideRight]	Material override for the +X face.
[aOverrideFront]	Material override for the -Z face.
[aOverrideBack]	Material override for the +Z face.

Example

```
GameObject cube = Ferr.SuperCube.CreatePivot(
      Vector3.zero,
      Ferr.PivotType.Left |
      Ferr.PivotType.Bottom |
      Ferr.PivotType.Front,
      Vector3.one,
      material);
cube.transform.parent = transform;
```

Ferr.SuperPlane.Create

```
public static GameObject Create(
      Vector3  aAt,
      Vector2  aSize,
      Material aMaterial,
      UVType   aType          = UVType.Unit,
      float    aSliceDistance = 0)
```

Description

Creates a SuperPlane game object, and assigns the given data! This function will build the mesh right away.
Returns a ready-to-go SuperPlane GameObject named "SuperPlane" with a with a fully built SuperPlane, MeshFilter, Renderer, and BoxCollider component!

Parameters

aAt	Location in world space.
aSize	The width and height of the SuperPlane.
aMaterial	The material to assign to it, don't want that awful pink color!
[aType]	What type of UV coordinates do you want on the plane?
[aSliceDistance]	How far apart should extra verts be spaced out on the surface? 0 for none at all. floor(size/sliceDistance)

Example

```
GameObject plane = Ferr.SuperPlane.Create(
      Vector3.zero,
      Vector2.one,
      material);
plane.transform.parent = transform;
```

Ferr.SuperPlane.CreatePivot

```
public static GameObject CreatePivot(
      Vector3   aPivotPt,
      PivotType aPivotType,
      Vector2   aSize,
      Material  aMaterial,
      UVType    aType          = UVType.Unit,
      float     aSliceDistance = 0)
```

Description

Creates a SuperPlane game object, and assigns the given data! This function will build the mesh right away.
Returns a ready-to-go SuperPlane GameObject named "SuperPlane" with a with a fully built SuperPlane, MeshFilter, Renderer, and BoxCollider component!

Parameters

aPivotPt	Location to place the pivot point of the object. This is not the actual position of the object after the pivot is applied.
aPivotType	A bit flag that represents where the pivot is located. This only accepts Top, Bottom, Left, Right flag options, Top being +Z
aSize	The width and height of the SuperPlane.
aMaterial	The material to assign to it, don't want that awful pink color!
[aType]	What type of UV coordinates do you want on the plane?
[aSliceDistance]	How far apart should extra verts be spaced out on the surface? 0 for none at all. floor(size/sliceDistance)

Example

```
GameObject plane = Ferr.SuperPlane.CreatePivot(
      Vector3.zero,
      Ferr.PivotType.Top | Ferr.PivotType.Left,
      Vector2.one,
      material);
plane.transform.parent = transform;
```

Ferr.SuperPlane.CreateRectXZ

```
public static GameObject CreateRectXZ(
      Rect     aRect,
      Material aMaterial,
      UVType   aType          = UVType.Unit,
      float    aSliceDistance = 0)
```

Description

Creates a SuperPlane game object on the XZ plane from the given rectangle, Y=0, facing +Y. This function will build the mesh right away.
Returns a ready-to-go SuperPlane GameObject named "SuperPlane" with a with a fully built SuperPlane, MeshFilter, Renderer, and BoxCollider component!

Parameters

aRect	A rectangle describing the location and size of the plane on the XZ axis
aMaterial	The material to assign to it, don't want that awful pink color!
[aType]	What type of UV coordinates do you want on the plane?
[aSliceDistance]	How far apart should extra verts be spaced out on the surface? 0 for none at all. floor(size/sliceDistance)

Example

```
GameObject plane = Ferr.SuperPlane.CreateRectXZ(
      new Rect(0,0,1,1),
      material);
plane.transform.parent = transform;
```

Ferr.SuperPlane.CreateRectXY

```
public static GameObject CreateRectXY(
      Rect     aRect,
      Material aMaterial,
      UVType   aType          = UVType.Unit,
      float    aSliceDistance = 0)
```

Description

Creates a SuperPlane game object on the XY plane from the given rectangle, Z=0, facing -Z. This function will build the mesh right away.
Returns a ready-to-go SuperPlane GameObject named "SuperPlane" with a with a fully built SuperPlane, MeshFilter, Renderer, and BoxCollider component!

Parameters

aRect	A rectangle describing the location and size of the plane on the XY axis
aMaterial	The material to assign to it, don't want that awful pink color!
[aType]	What type of UV coordinates do you want on the plane?
[aSliceDistance]	How far apart should extra verts be spaced out on the surface? 0 for none at all. floor(size/sliceDistance)

Example

```
GameObject plane = Ferr.SuperPlane.CreateRectXY(
      new Rect(0,0,1,1),
      material);
```

```
plane.transform.parent = transform;
```