# Runtime API

All Paint In 3D features can be accessed from the C# API.

The main class is the ‘P3D_Painter’ class, which provides a simple painting API, and also allows you to lock a single mesh to allow for efficient 3D painting.

In your code, the first thing you should do is create a ‘P3D_Painter’ instance, for example:

```
var myPainter = new P3D_Painter();
```

NOTE: If you don’t want to keep making painter instances then you can just use the P3D_Helper.Painter static property, which will always return a valid painter instance which you can share with everything, just keep in mind that this will be slow if you plan to paint many different meshes all the time.

You then need to set the current mesh you want to paint, this can be done with SetMesh:

```
myPainter.SetMesh(myGameObjectThatHasAMeshFilter);
```

You then need to set the texture you want to paint:

```
myPainter.SetTexture(myGameObjectThatHasAMeshRenderer);
```

You then need to define a brush you want to paint with, for example:

```
var myBrush = new P3D_Brush();

myBrush.Color = Color.red;

myPainter.SetBrush(myBrush);
```

This will create a new alpha blending brush that’s red.

You can now begin painting, either in 2D or 3D coordinates:

```
// 3D paint first pixel between the start and end points
myPainter.PaintBetweenNearest(mytartPosition, myEndPosition);

// 3D paint the nearest pixel
myPainter.PaintNearest(myPosition, myMaxDistance);

// 2D painting
myPainter.Paint(uv);
```

Once finished, you need to apply changes to the current texture:

```
myPainter.ApplyPaint();
```

Keep in mind that this interface is designed to work with a single mesh. If you need multiple meshes then you might want to use raycasting to find which object you want to paint, then perform the steps above. If you do use raycasting then keep in mind that if you use non-convex MeshColliders then you can skip the myPainter.SetMesh part, and pass the UV coordinates directly to the myPainter.Paint method.

Also keep in mind that the P3D_Paintable component is an even higher level abstraction that makes all of these steps easier. Take a look at the **‘Mouse Raycast Painting’** demo scene to see how to use it.