

3/19/2016

UNITYCODER.COM

MOBILE PAINT V1.96



# Mobile Paint

---

## Table of Contents

INTRODUCTION .....	2
RELEASE NOTES .....	2
FEATURES .....	2
LIMITATIONS.....	2
REQUIREMENTS.....	3
FILES IN THE PACKAGE.....	3
USAGE INSTRUCTIONS.....	4
PUBLIC METHODS/FUNCTIONS .....	10
CUSTOMIZATION .....	10
KNOWN ISSUES.....	11
SUPPORT & FEEDBACK .....	12

# Mobile Paint

---

## INTRODUCTION

Simple painting program, optimized for mobiles, supports multi-touch.  
(Can be used with mouse also)

## RELEASE NOTES

See MobilePaint/ Documentation/release\_notes.txt

## FEATURES

- Optimized texture painting (much faster than SetPixel/SetPixels or SetPixels32)
- Multi-touch support (paint simultaneously with many fingers)
- Works with Unity Indie & Pro
- Simple painting demos included (see also .apk demo download in the blog)
- Overlay texture on canvas
- Flood fill
- Palette image color picker
- Custom brushes
- Additive color brush ("soft brush")
- Alpha canvas (can see background behind it)
- 1 Step undo
- Locking mask painting (only paint to target area)
- Custom mesh painting
- Using new UI
- Mask Only Locking paint area (v1.5)
- Threshold option for paint & fill
- Paint with texture pattern
- Eraser
- Area fill calculation
- Many example scenes to play with
- And more.. (see website or github site for latest info)

## LIMITATIONS

- Custom brush image is tilted (if you want it to be straight, make it as tilted image originally)
- You will need to modify existing C# codes if want to add custom features!

## REQUIREMENTS

- V1.96 is uploaded with Unity 5.1, so the scenes wont open in 4.x (the scripts should still work, but you will need to open the project in 5.1, see the example scene settings, and use the same in 4.x)

## FILES IN THE PACKAGE

Files inside the “**MobilePaint/**” folder:

FOLDER / FILE	INFO
/Documentation/ release_notes.txt	Short notes about what is new/changed
/Documentation/ MobilePaint.pdf	(This pdf)
/Materials/Brushes/*	Custom brush samples
/Materials/DemoMaterials/*	Materials used in demo scenes, not required
/Materials/Mats/CanvasDefault	Material for the drawing plane (default)
/Materials/Mats/CanvasLockMask	Material for the locking mask painting sample
/Materials/Mats/CanvasWithAlpha	Canvas with alpha, can see objects behind it
/Materials/Mats/CanvasWithMask	Canvas with mask image layer
/Materials/Mats/Shaders/canvas	Simple texture shader
/Materials/Mats/Shaders/canvasTexture	Simple texture shader, with mask image
/Materials/Palettes/color_palette_sample1	Example color palette *Note: must have [x] read/write enabled
/Materials/Shaders/*	Shaders for the canvas materials
/Materials/UI/*	Example GUITexture images & buttons
/Sample Images/*	Sample images for testing
/Scenes/NewUI/	Example scenes
/Scripts/Brush/CloseCustomBrushDialog	
/Scripts/ Brush /CustomBrushPicker	Selection grid for custom brush picking
/Scripts/Common/GUIScaler	Example script for scaling GUITextures based on device resolution
/Scripts/Common/ObjectRotator	Example script for rotating object (in lockmask demo scene)
/Scripts/ Common /PaintTools	To-be collection of some helper methods
/Scripts/NewUI/BrushSizeUI	Sample scripts for setting brush size
/Scripts/NewUI/ColorUIManager	Sample scripts for color pencils
/Scripts/NewUI/CustomBrushesUI	Sample scripts for custom brush list
/Scripts/NewUI/ToggleBrushModeUI	Sample scripts for setting brush mode
/Scripts/NewUI/ToggleCustomShapeModeUI	Sample scripts for setting custom brush mode
/Scripts/NewUI/ToggleFloodFillModeUI	Sample scripts for setting flood fill mode
/Scripts/UserInterface/AdjustBrushSize	Sample script for brush size adjust
/Scripts/UserInterface/ClearButton	Sample script for calling ClearImage()
/Scripts/UserInterface/CloseColorDialog	Closes the color dialog
/Scripts/UserInterface/ColorPicker	New color picker from palette guitexture
/Scripts/UserInterface/CustomBrushDialog	Opens modal brush “window”
/Scripts/UserInterface/GetStartColor	Gets initial paintcolor for color preview guitextures
/Scripts/UserInterface/PaletteDialog	Opens modal palette “window”
/Scripts/UserInterface/ToggleMode	Swaps between draw & flood fill modes
/Scripts/MobilePaint	The Main script for drawing!

## USAGE INSTRUCTIONS

### Getting started

- Try building the included “scene\_MobilePaint\_NewUI\_Default” in your mobile device.
- Note: Remember to enable: [x] enableTouch (and disable [ ] enableMouse) when building to mobile devices
- Check performance (you can use this FPS counter:  
<http://wiki.unity3d.com/index.php/FramesPerSecond> )
- If you need better fps try adjusting:
  - o Disable: [ ] Realtime texture update (and adjust TextureUpdateSpeed, higher value adds more delay, but uses less cpu)
  - o Disable: [ ] Use hiQualityBrush
  - o Adjust: ResolutionScaler value (0.5 is half resolution)

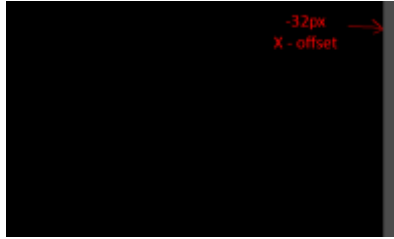
**PAINTING/CANVAS SETTINGS**

**\*\* NOTE: For latest info, see settings documentation online :**

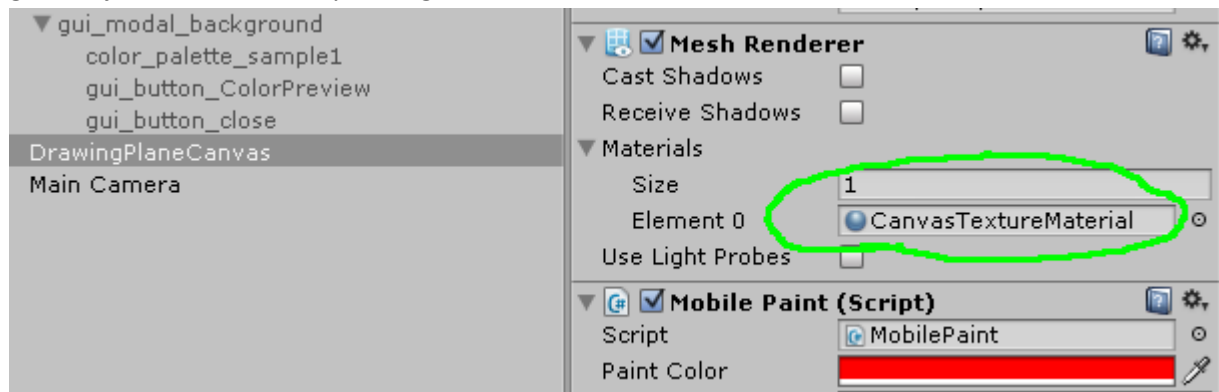
<https://github.com/unitycoder/UnityMobilePaint/wiki/Settings>

- Select "DrawingPlaneCanvas" gameobject from Hierarchy
- MobilePaint script has public variables visible in the inspector:
  - **[x] Enable Mouse** : Enable painting with Mouse
  - **[x] Enable Touch** : Enable painting with Touch
  - **PaintLayerMask** : Painting uses this layer for checking raycast collision on canvas
  - **[x] Create Canvas Mesh**: Enabled by default, creates full screen quad for drawing canvas, if disabled, uses existing mesh instead
  - **Reference Area** : Take canvas size from this UI object (instead of using full screen quad)
  - **[x] Connect Brush Strokes**: Enabled by default, if you draw & move fast, line is drawn between the 2 points. Should disable this when using custom mesh or otherwise the line wraps from other side
  - **Paint Color** : the initial color to paint with
  - **Resolution Scaler** : 1 = full resolution, 0.5 = half resolution ( faster) , 0.25 = quarter resolution (event more faster)
  - **Brush Size** : Initial brush size \*Note use SetBrushSize() to adjust size at runtime
  - **Brush Size Min** : Minimum brush size
  - **Brush Size Max** : Maximum brush size
  - **[ ] Use Additive Colors** : Basically soft brush effect, adjust brush color alpha less than 1 also
  - **Brush Alpha Strength** : Multiplier for alpha value while painting (0.1 = soft, +1=strong/fast)
  - **DrawMode** : Enum list: Default, CustomBrush, FloodFill, Pattern, Shape Lines
  - **[ ] Use Lock Area** Use the new locking mask paint feature (only paints into the target color, determined by "hidden" floodfill on click)
  - **[ ] UseMaskLayerOnly** : Will create locking area mask from MaskImage only
  - **[ ] Smoothen Mask Edges** : To avoid white pixels around black borders, Note: doesn't work with thin lines
  - **[ ] UseThreshold** : If enabled, painting & filling uses threshold, instead of exact match only (enabling this helps painting over line/border jaggies) \*Note: Simple R+G+B distance comparison is used for threshold, might not always work perfectly in all images.
  - **PaintThreshold** : Value between 0-255. 0 = Exact match only, 128 = Good value..
  - **[x] GetAreaSize**: If enabled, fires event after painting and returns target area size in pixels
  - **[x] Can Draw On Black** : If disabled, use cannot draw on black pixels (to avoid drawing on black border lines) \*Not used/working in all methods yet
  - **CanvasSizeAdjust** : \*Not really needed with the new UI\* Leaves space for UI (default value is x=-32, y=0, that means, right side of the screen has -32 pixels removed from painting area. \*This value is automatically scaled based on GUIScaler calculations, so in HD devices it would end up being more than -32px)

\*example: Original screen size 800x480, becomes: 768x480

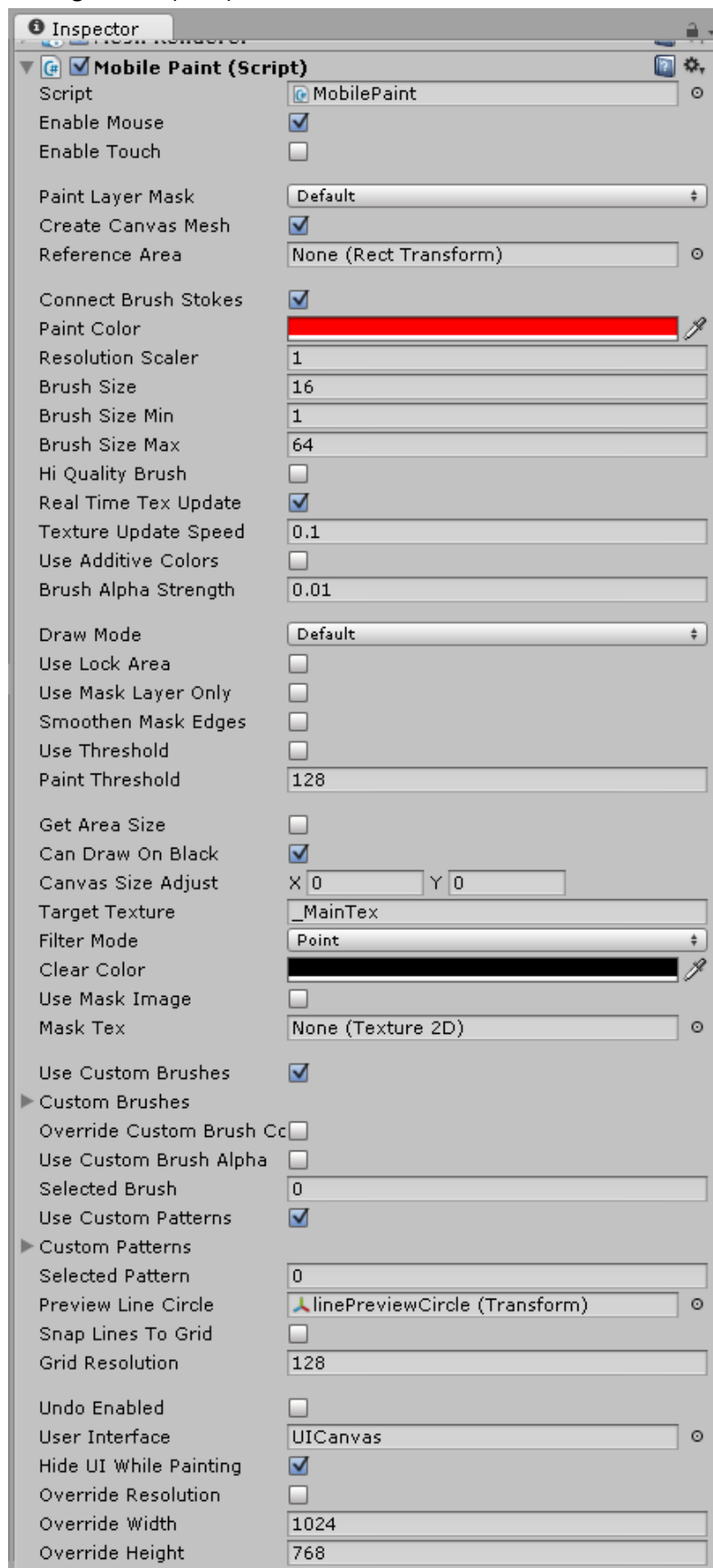


- **TargetTexture** : To which texture in the current material (shader) we should paint into. Usually its `_MainTex`.
- **Filter Mode** : Texture filter mode ("Point" is best, but the other 2 are smoother)
- **Clear color**: Color used for new image (clearing)
- **[ ] Use Mask Image**: Should we assign mask/layer image?
- **Mask Tex**: Assign your image here (example included "sample\_linedrawing1.png").  
\*Note: You must assign "CanvasTextureMaterial" material to "DrawingPlaneCanvas" gameobject to use mask/layer image.



- **[ ] UseCustomBrushes**: Is custom brush tool enabled
- **CustomBrushes**: Array of custom brush images  
\*Note: Read/write must be enabled for the texture (from importer settings)
- **[ ] Override Custom Brush Color** : Use paint color instead of brush texture colors
- **[ ] Use Custom Brush Alpha** : Should alpha value be taken from brush texture, if not enabled, then paint color alpha value is used instead
- **Selected Brush** : Index number of the selected brush
- **[ ] Use custom Patterns**: is custom pattern tool enabled
- **CustomPatterns**: Array of custom pattern brush images
- **Selected Pattern** : Index number of the selected pattern brush
- **PreviewLineCircle**: Circular sprite, used for linedrawing, to draw line start and end
- **[ ] Snap Lines To Grid**: Used for line drawing, not really useful yet
- **Grid Resolution**: Used for line drawing, snapping grid pixel size
- **[ ] Undo enabled** : 1 Step undo (restores changes 1 step back) **\*\*\*NOTE:** Currently there is no undo button for mobile, desktop version can press "U" key for undo. On mobile you have to call undo yourself (through some button for example) "public void DoUndo()" is located in MobilePaint.cs
- **User Interface**: Reference to new UI canvas root
- **[x] Hide UI While Painting**: For the new UI, hides UI while painting
- **[x] OverrideResolution**: If enabled, resolution below is used, instead of screen resolution  
\*Note: if texture or mask is assigned, this wont work (error/warning should appear)
- **OverrideWidth**: Custom drawing texture width

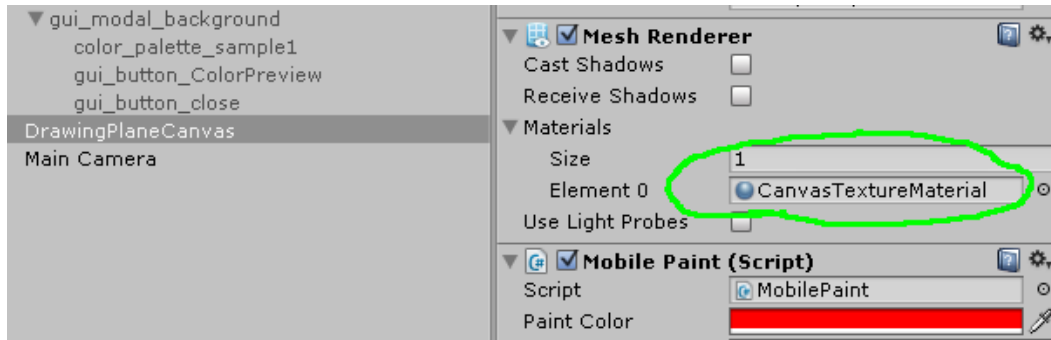
- **OverrideHeight:** Custom drawing texture height
- Settings view: (V1.9)



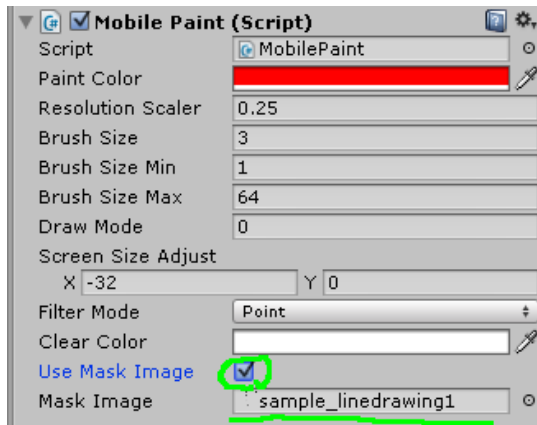


## Using mask texture

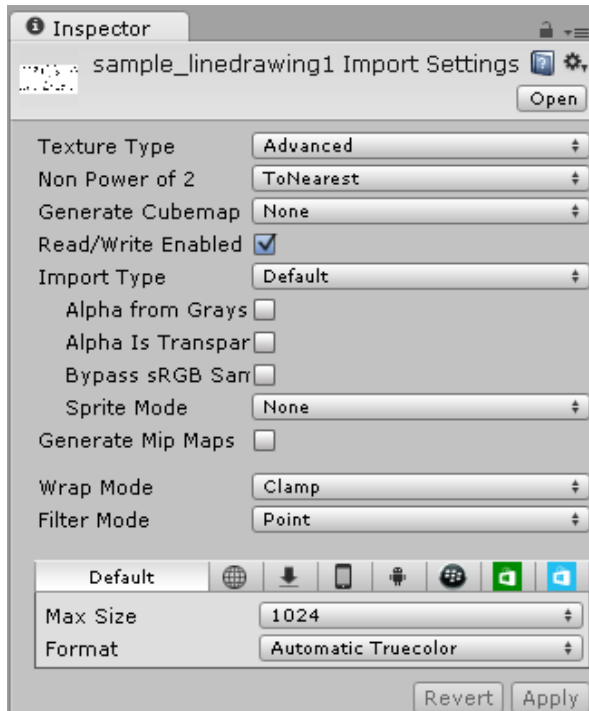
- Select "DrawingPlaneCanvas" gameobject from hierarchy
- Check that "CanvasTextureMaterial" is used on the mesh renderer:



- Then enable **[x] Use Mask Image** from "DrawingPlaneCanvas" gameobject and assign mask image:

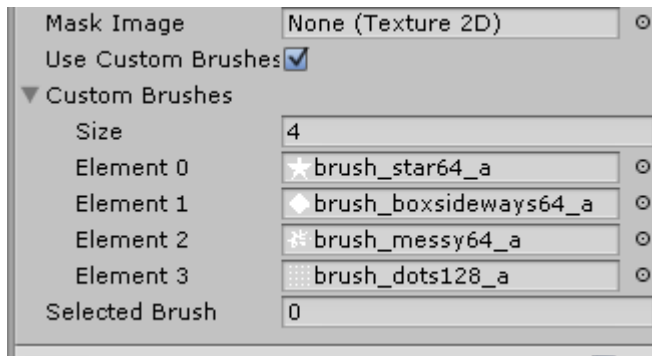


- \*Note: Mask texture needs to have **[x] Read/Write enabled** from inspector:

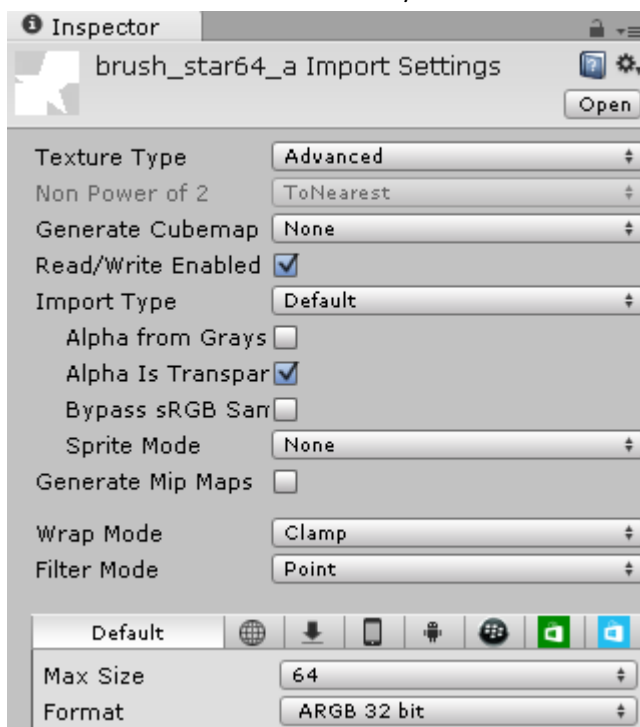


## Using custom brushes

- Enable [x] UseCustomBrushes from “DrawingPlaneCanvas” gameobject
- Assign textures into CustomBrushes array:



- Make use the textures have Read/Write enabled from import settings:



## PUBLIC METHODS/FUNCTIONS

### In MobilePaint.cs

- Public void InitializeEverything()
- public void GetCurrentBrush()
- public void DrawPoint(int x,int y) : Draws single pixel into give X, Y texture coordinate
- public void DrawPoint(int pixel) : Draws pixel using 1D pixel array index
- public void DrawCircle(int x, int y) : Draws circle into give X, Y texture coordinate.  
Uses current color, current brush width.
- public void DoUndo() : Executes undo (only 1 step back available)
- public void ClearImage()
- public void ClearImageWithImage()
- public void ReadClearingImage()
- public void ReadMaskImage()
- public void HideUI() \*Note these can be overridden, see example scene: "scene\_Custom\_Override"
- public void ShowUI()
- public void SetBrushSize()
- public void SetDrawModeBrush()
- public void SetDrawModeFill()
- public void SetDrawModeShapes()
- public ReadCurrentCustomPattern() : reads current selected pattern texture pixels
- public GetCanvasAsTexture() : returns current drawing layer (the pixels array) as Texture2D
- public GetScreenshot() : returns current scene screenshot (without UI) as Texture2D

*(this section fill be updated later, see source code for more info about those functions)*

## CUSTOMIZATION

### ~~Custom color palette (using old GUI)~~

- Check "ColorPicker.cs" for reference
- First you need reference to the canvas object/script: ~~public MobilePaint canvas;~~
- And then just send your own color there: ~~canvas.paintColor = yourCustomColor;~~
- Now that color is used as painting color in the "DrawingPlaneCanvas" gameobject
- For example, add different color pencils as buttons, each one sends the color to paintColor  
example image: <http://www.clipartbest.com/cliparts/9i4/erR/9i4erRdGT.png>

**Other notes**

- ~~If you add your own GUITexture elements which's are NOT meant to be clicked, set their layer as "Ignore raycast" (otherwise they would block the drawing raycast)~~

**KNOWN ISSUES**

- ~~Mesh paint example: Cannot paint with multi touch, for some unknown reason it paints in the middle of the touches instead.. (Fixed)~~
- Mesh Paint example scene (Rotating sphere) : If you don't move your finger, paint doesn't appear, because currently painting is only done when pointer is moved..
- Unity editor hangs if your texture format is set to "Automatic Compressed" with Android platform selected. (Hangs when GetPixel() is used to that texture. Just set texture format to RGBA32 for example)

## SUPPORT & FEEDBACK

NEW: You can also send requests, comments, bug reports directly to github issues page:

<https://github.com/unitycoder/UnityMobilePaint/wiki/Settings>

Send comments / feedback & requests to my blog:

<http://unitycoder.com/blog/2014/07/15/asset-store-mobile-paint/>

Unity forums: <http://forum.unity3d.com/threads/released-mobile-paint-with-multi-touch.262645/>

Email: [support@unitycoder.com](mailto:support@unitycoder.com)

- Add product name: "Mobile Paint" to the subject
- Include Asset Store invoice number in the email
- Include screenshot(s) if something strange is happening (helps solving the problem)

Or, Using Unity3D forums private message:

<http://forum.unity3d.com/members/mgear.22727/#info>

*\*\*\* also feel free to offer Unity jobs in .my or .sg or .uk \*\*\**