Drawing Board VR Unity Package Documentation

Table of Contents

1. Introduction

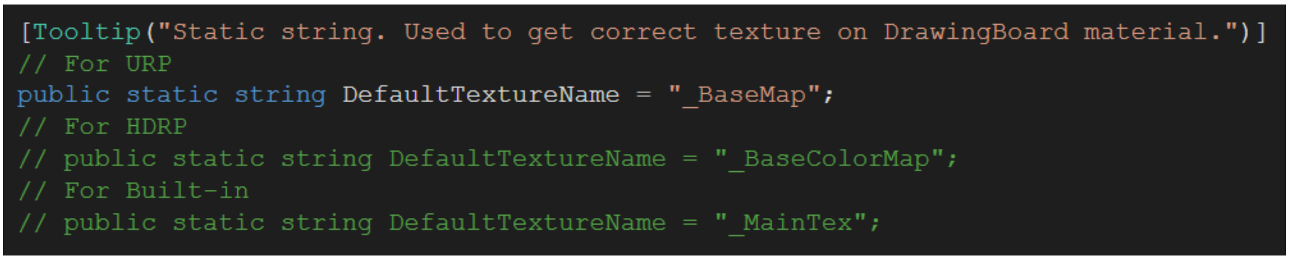
Welcome to the documentation for the "Drawing Board VR" Unity package. This package enables users to draw on virtual boards in a virtual reality (VR) environment, mimicking the experience of using markers on a physical board. The drawing boards provide a fun and immersive drawing experience.

2. Package Setup

To set up the Drawing Board VR package in your Unity project, please follow these steps:

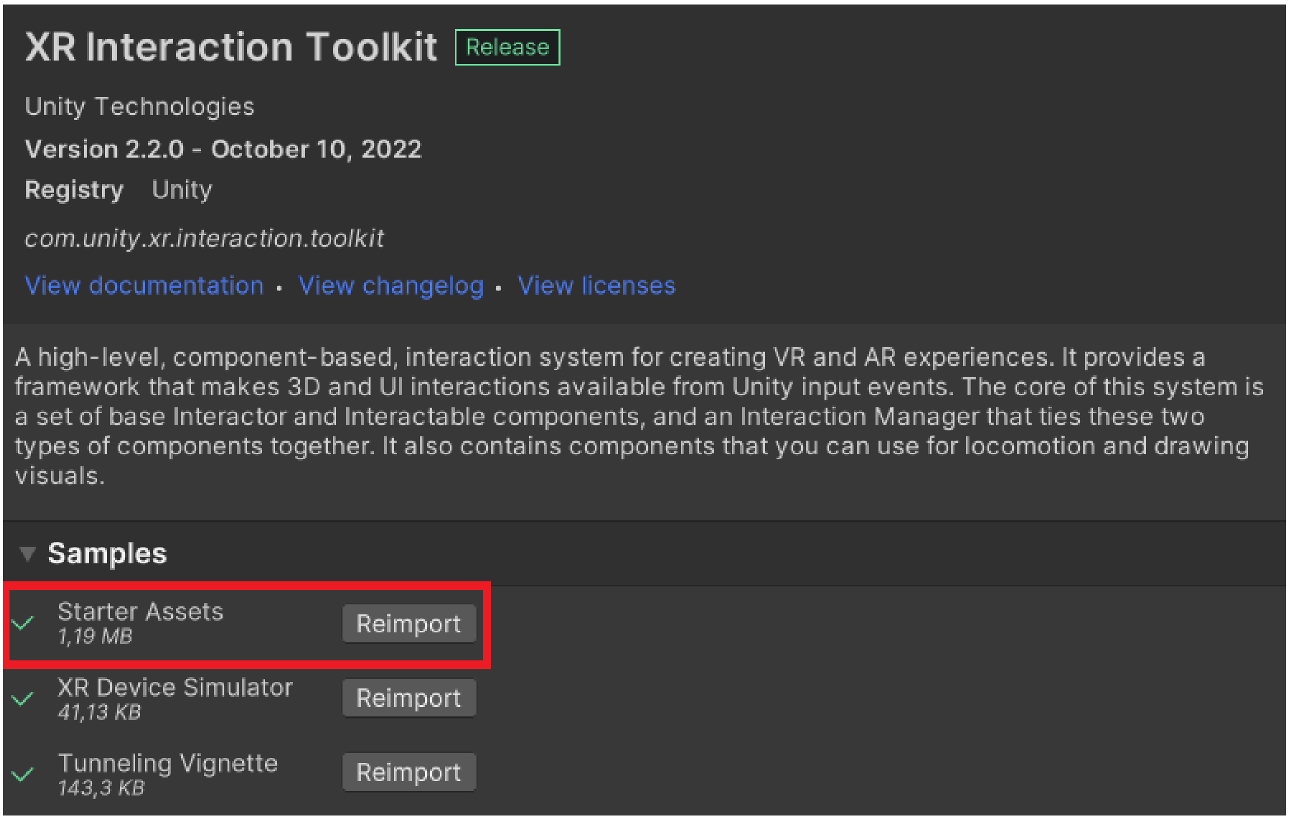
1. Ensure that you have the Universal Render Pipeline (URP) installed in your project. The Drawing Board VR package was developed using URP and requires it to function properly.
2. If your project is using a different render pipeline, you may encounter pink materials in the demo scenes and on the prefabs. To resolve this, you need to update the shaders on the affected materials to match your render pipeline. Refer to the documentation or resources provided by your chosen render pipeline for instructions on how to update the shaders. You also have to uncomment the correct line in the DrawingBoardSettings script depending

on the render pipeline you use.



```
[Tooltip("Static string. Used to get correct texture on DrawingBoard material.")]
// For URP
public static string DefaultTextureName = "_BaseMap";
// For HDRP
// public static string DefaultTextureName = "_BaseColorMap";
// For Built-in
// public static string DefaultTextureName = "_MainTex";
```

3. In order for the Drawing Board VR package to work, you will need to have a VR setup in your project. This includes having OpenXR installed and configured.
4. Additionally, the package relies on the XR Interaction Toolkit, which is a Unity package for building VR and AR interactions. To use the Drawing Board VR package, you will need to install the XR Interaction Toolkit and include the sample "Starter Assets" provided with it.



XR Interaction Toolkit

Unity Technologies

Version 2.2.0 - October 10, 2022

Registry Unity

com.unity.xr.interaction.toolkit

View documentation • View changelog • View licenses

A high-level, component-based, interaction system for creating VR and AR experiences. It provides a framework that makes 3D and UI interactions available from Unity input events. The core of this system is a set of base Interactor and Interactable components, and an Interaction Manager that ties these two types of components together. It also contains components that you can use for locomotion and drawing visuals.

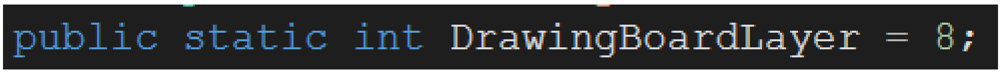
Samples

✓ Starter Assets *1,19 MB* Reimport

✓ XR Device Simulator *41,13 KB* Reimport

✓ Tunneling Vignette *143,3 KB* Reimport

5. In the “DrawingBordSettings” script set the correct “DrawingBoardLayer”.



```
public static int DrawingBoardLayer = 8;
```

3. Features

The Drawing Board VR package offers a range of features to enhance your virtual drawing experience:

- Drawing Board: Create virtual drawing boards that mimic the look and feel of physical boards.
- Marker Interaction: Use VR controllers to draw on the virtual board. Markers support both sphere or mesh colliders to detect contact area between board and marker and give as much precision as possible.
- Eraser Interaction: Fine-tune your drawings by using the eraser feature. The package includes erasers with box or mesh colliders, allowing you to erase specific parts of your drawing, providing greater control and precision.
- Set Image on Board: Load and display images on the virtual drawing board. This functionality allows you to import images onto the board surface, providing a versatile tool for creative expression.
- Save Drawn Image: Capture and save the drawings created on the virtual board as image files. This feature enables users to preserve their artwork or share it with others outside the VR environment.
- Ultra Performance: Achieve high performance in your VR application with the Drawing Board VR package. By leveraging the power of compute shaders, the package maximizes performance efficiency, enabling smooth and responsive drawing experiences even on resource-constrained systems.

4. Usage

To use the Drawing Board VR package in your Unity project, follow these steps:

1. Drag and drop the "DrawingBoard" prefab into your scene.
2. Set resolutionX (width) of board (unity supports RenderTextures up to 16384x16384), resolutionY (height) is calculated automatically so pixels are even size.
3. Drag and drop the "Markers" and “Erasers” prefabs into your scene.
4. Set up your VR camera rig if you haven't already. (you can use one provided with XR Interaction Toolkit sample assets)

5. API Reference

The Drawing Board VR package provides the following API methods for interacting with the virtual drawing board:

DrawingBoardTexture Class

- **bool SaveImage()** - Saves the image on the drawing board as a PNG file using the specified on DrawingBoardTexture component file path and name. Returns true if the image is saved successfully.

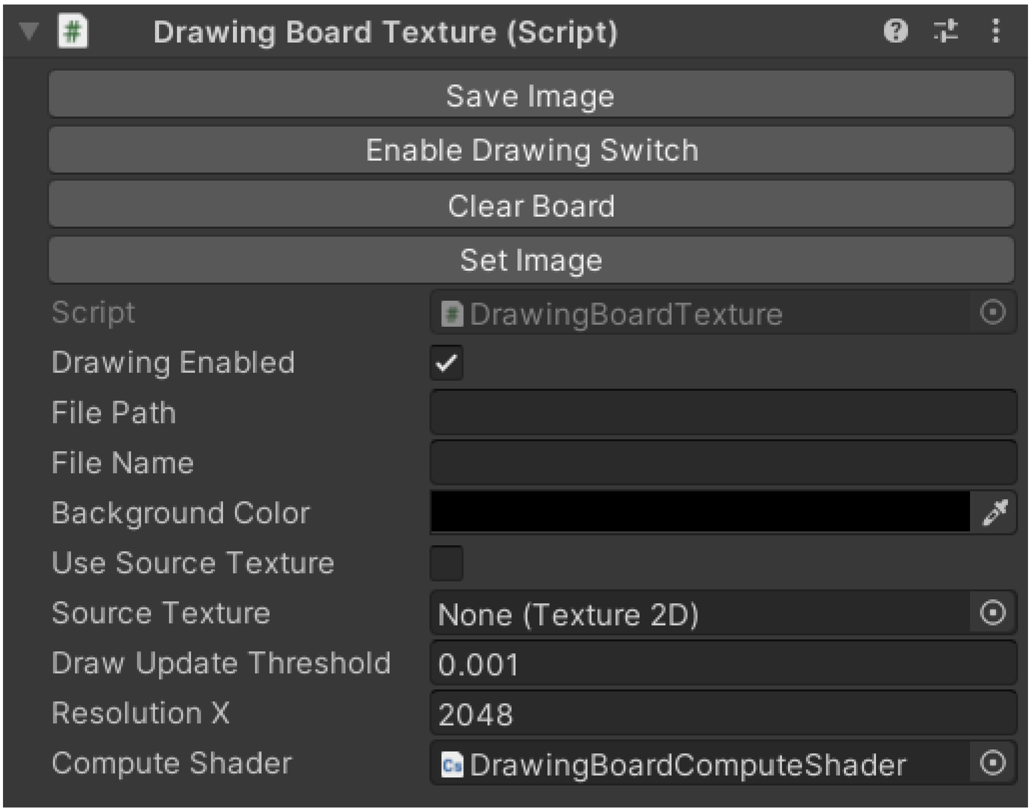
- **bool SaveImage(string path, string name)** - Saves the image on the drawing board as a PNG file at the specified path and with the given name. Returns true if the image is saved successfully.
- **void ClearBoard()** - Clears the drawing board by setting all pixels to the background color.
- **Color GetBackgroundColor()** - Returns the background color of the drawing board.
- **void SetBackgroundColor(Color col)** - Sets the background color of the drawing board.
- **void SetImage(Texture2D tex, bool resize = false)** - Sets the given texture as the image on the drawing board. Optionally, you can resize the board to match the texture's resolution.
- **void EnableDrawing(bool value = true)** - Enables or disables drawing on the board.
- **Texture2D GetTexture()** - Returns the image on the drawing board as a texture.
- **bool IsDrawingEnabled()** - Returns true if drawing on the board is enabled.

Marker Class

- **void SetColor(Color col)** - Sets the color of the marker.

6. Components

DrawingBoardTexture

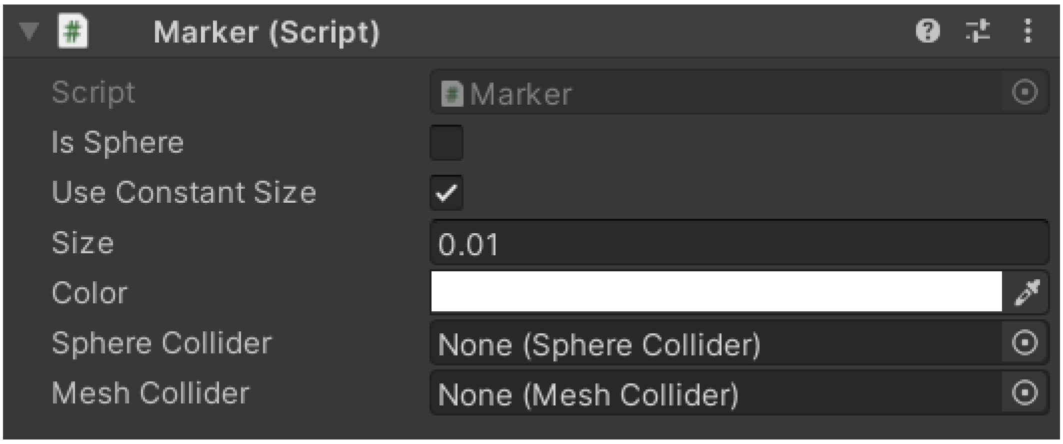


Script: DrawingBoardTexture
Drawing Enabled: ☑
File Path:
File Name:
Background Color: ■
Use Source Texture: □
Source Texture: None (Texture 2D)
Draw Update Threshold: 0.001
Resolution X: 2048
Compute Shader: DrawingBoardComputeShader

The DrawingBoardTexture component provides variables to control the behavior and appearance of the drawing board:

- **bool drawingEnabled**: A boolean indicating whether drawing on the board is enabled.
- **string filePath**: The file path where the drawn image will be saved.
- **string fileName**: The name of the drawn image file.
- **Color backgroundColor**: The color of the drawing board's background.
- **bool useSourceTexture**: A boolean indicating whether to set sourceImage image on board on awake.
- **Texture2D sourceTexture**: The source texture to set on the drawing board.
- **float drawUpdateThreshold**: A value that determines how far a marker has to move to update the board.
- **int resolutionX**: The X resolution (width) of the drawing board. Height is calculated automatically.

Marker



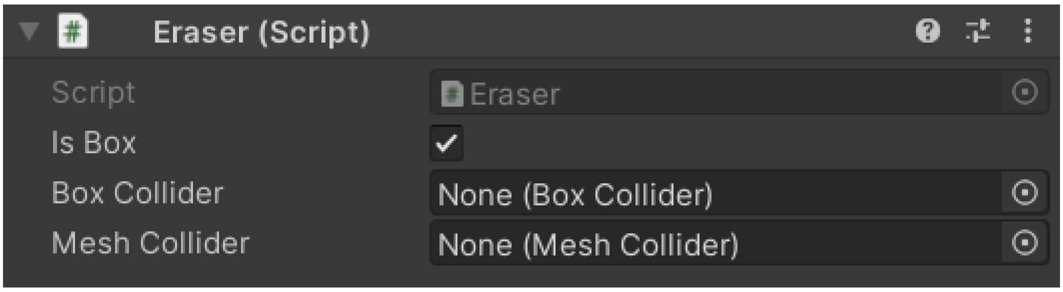
Marker (Script)

Script	Marker
Is Sphere	☐
Use Constant Size	☑
Size	0.01
Color	⬜
Sphere Collider	None (Sphere Collider)
Mesh Collider	None (Mesh Collider)

The Marker component provides variables to control the behavior and appearance of the marker used for drawing:

- **bool isSphere**: A boolean indicating whether the marker should use a sphere collider or mesh collider.
- **bool useConstantSize**: A boolean indicating whether the marker should have a constant size. If false size is the contact area of sphereCollider and the board.
- **float size**: The size of the marker if useConstantSize is true.
- **Color color**: The color of the marker.
- **SphereCollider sphereCollider**: The sphere collider used if isSphere is set to true.
- **MeshCollider meshCollider**: The mesh collider used if isSphere is set to false. Remember that “Convex” and “is Trigger” have to be set to true on MeshColliderComponent.

Eraser



Eraser (Script)

Script	Eraser
Is Box	☑
Box Collider	None (Box Collider)
Mesh Collider	None (Mesh Collider)

The Eraser component provides variables to control the behavior and appearance of the eraser used for removing drawings:

- **bool isBox**: A boolean indicating whether the eraser should use a box collider or mesh collider.
- **BoxCollider boxCollider**: The box collider component used if isBox is set to true.
- **MeshCollider meshCollider**: The mesh collider component used if isBox is set to false. Remember that “Convex” and “is Trigger” have to be set to true on MeshColliderComponent.

7. Troubleshooting

If you encounter any issues or have trouble using the Drawing Board VR package, try the following troubleshooting steps:

- Ensure that you have installed the package correctly by following the installation instructions.
- Verify that your VR controllers are properly configured and recognized by Unity.
- Check the Unity Console for any error messages related to the package.

8. Support and Feedback

For support or feedback regarding the Drawing Board VR Unity package, you can reach out to the package developer through the following channels:

- Email: beyondlimitsstudio.gaming@gmail.com