

How it works

The idea is pretty simple. We put paintcraft canvas to the scene somewhere outside of the level so player won't see it. Then we take render texture used by paintcraft from back layer and just use it with plane anywhere in the scene.

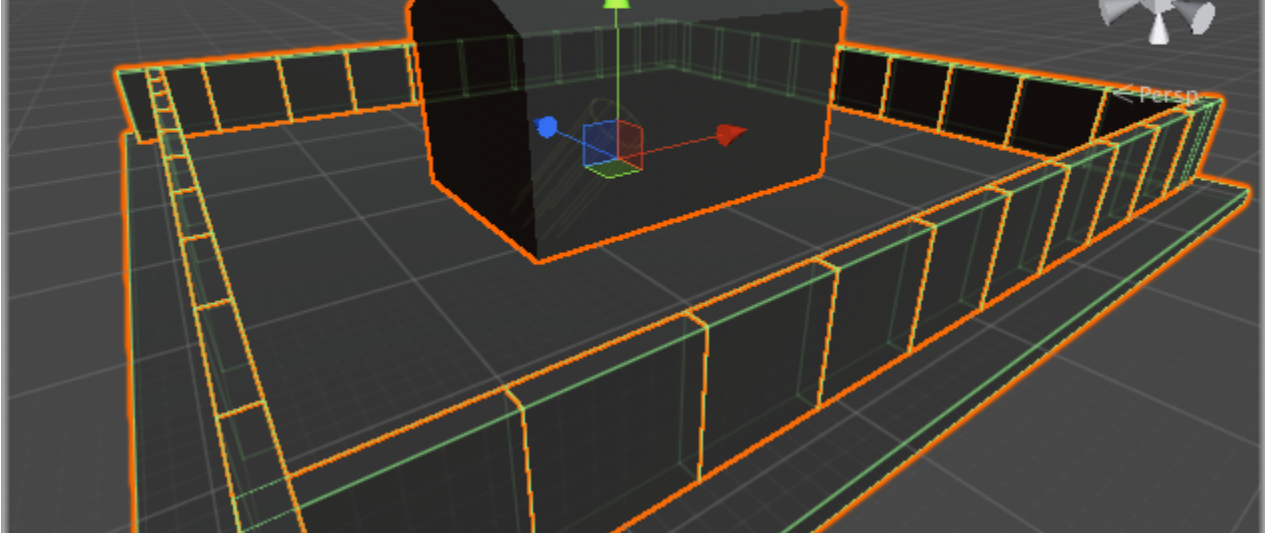
All related scripts and objects located at **PaintCraft/Pro/3D** folder

Final part is to transfer drawing events to paintcraft. You can find the script below which use raycast from the center of the screen determine position and just forward this event to paintcraft.

Let's create 3D scene step by step.

Environment

I used 3d models from Prototyping standard assets. So just drag and drop **PaintCraft/Pro/3D/Prefabs/Environment** prefab in to your scene to get a simple level with a house on your scene.



CharacterController

Next let's add Character which will be able to run on our scene. You can find standard **FPSController** at **PaintCraft/Pro/3D/Standard Assets/Characters/FirstPersonCharacter/Prefabs** folder

Drag and drop **FPSController** to the scene somewhere inside our level.

Now if you start the scene you should be able to run on the level using [WSAD] keys and mouse

Paintcraft Canvas setup

Now we need to place PaintCraft canvas somewhere on the scene where player wont see it. I prefer position where x,y,z = 10000. Important rule: **Don't change scale or rotation of this object**

If you wish to add several canvases their positions must not overlap each other. Otherwise PaintCraft camera will see changes from one canvas on another.

Drag and drop **PaintCraft/Pro/3D/Prefabs/PaintcraftCanvas** prefab to the scene two times. then change following parameters

- PaintcraftCanvas 1
 - position = (-10000, -10000, -10000)
 - pageConfig = select anything what you like (e.g. Bear)
- PaintcraftCanvas 2
 - position = (-20000, -10000, -10000) (See x position here is different from position used in previous canvas)
 - pageConfig = select anything what you like (e.g. BlankPageConfig)

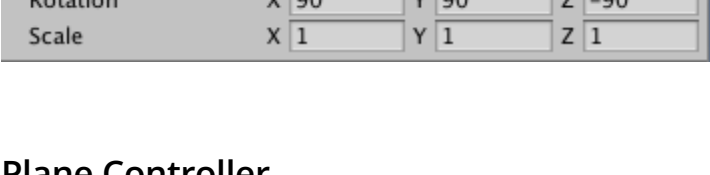
Add drawing to the scene

As I mention above. we just use render texture from paintcraft canvas. so what we need to do is just add standard **Primitive plane** to the scene and assign material which use paintcraft render texture.

Create Plane

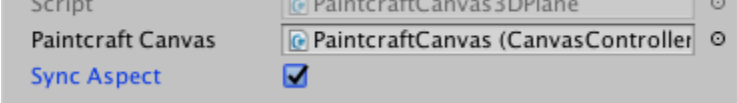
Use Unity top menu -> GameObject -> 3D Objects -> Plane

you can see that plane is an object which located at the floor. so we need to rotate it properly to do that you can make following rotation (90, 90, -90)



Plane Controller

Add **PaintCraft/Pro/3D/Scripts/PaintcraftCanvas3DPlane.cs** to the plane object. Then select **PaintcraftCanvas** parameter. and choose Paintcraft Canvas. You see that it has **SyncAspect** options. if you enable this parameter, this script will change horizontal scale and make aspect (width/height) of this plane the same as your paintcraft page.



You also need to change **material** for this plane. if you use coloring book with outline layer you should choose **PlaneMaterialWithOutlineLayer**. otherwise just **PlaneMaterial**.

Now start the scene and look at your plane. if you make everything right you should see the white plane with outline layer.

info: You are free to change the shader which will work with lightning. in this demo just unlit shader is used.

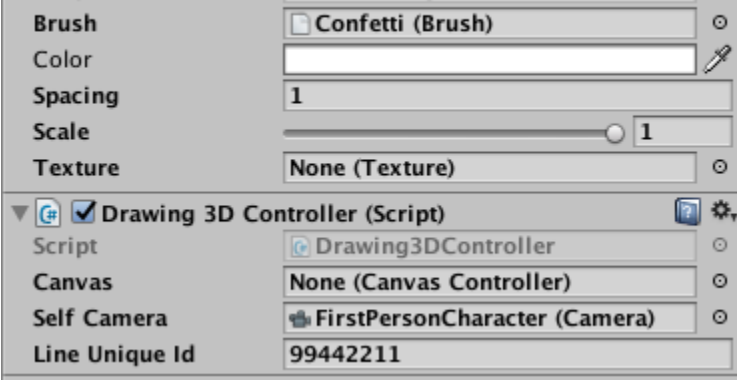
FPS drawing Controller

To make it works we need

1. Assign line config to the player. so player will use the same tool on any drawing canvas. Like select red color and draw a line on canvas1 then move to canvas2 and use the same red color
2. Use raycast from the screen center to determine which canvas we are drawing right now.

Drawing 3D Controller

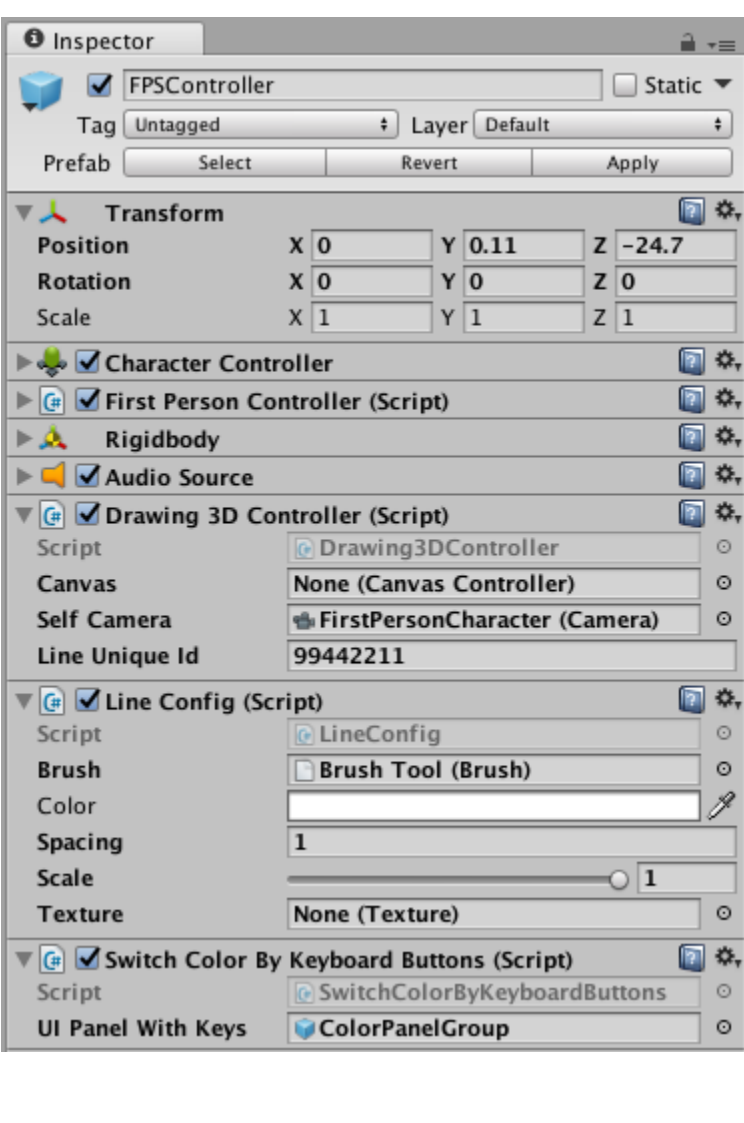
1. Add **PaintCraft/Pro/3D/Scripts/Drawing3DController.cs** component to the FPSController prefab instance
2. This component has dependency to the line config so you will get LineConfig on this object automatically
3. Select any Brush in the line config
4. We don't need to set canvas as it will change dynamically (remember that we can draw on many canvases)
5. Assign self camera. This camera is used as raycast source (for this prefab it's a **FirstPersonCharacter** object)
6. We need to set unique line id. E.g. if you have several players on scene which draw lines on the same canvas each LineConfig must have unique id.



Switch colors

Here you can learn how to change another parameter (Color) in the player LineConfig 1. Add **PaintCraft/Pro/3D/Prefabs/UI Canvas** to the scene. This prefabs contain UI element (color buttons and pointer). 2. Add **SwitchColorByKeyboardButtons.cs** script to the FPSController (this script locate at the same **PaintCraft/Pro/3D/Scripts/** folder) 3. Set **UI Pennel With Keys** to the **ColorPanelGroup** so now you will see on the screen which color is active and which color is assigned to the number keys

here is a final FPSController setup



Final part

That's it. if you play the scene you will be able to draw on the canvases. Now you can try to add more canvases and assign planes to them.

You can assign one paintcraft canvas to several planes so if you will draw on one another will show the live changes. You can setup several canvases on the scene, but keep in mind that undo history is stored in memory, so if you are doing WebGL build you need to make undo smaller otherwise you will see the memory exhausted error.

Feel free to open and modify any scripts from **PaintCraft/Pro/3D/Scripts** folder as they provided as reference. Probably if you want to make VR drawing your raycast source will be somewhere on the VR controller rather than camera. and color selection and brush change will be more complex in live project than in this example.

In this document

- Environment
- CharacterController
- Paintcraft Canvas setup
- Add drawing to the scene
- Create Plane
- Plane Controller
- FPS drawing Controller
- Drawing 3D Controller
- Switch colors
- Final part