**Unity Project**

**Free hand drawing application**

By

**Minh Dung Do**

1. **Requirements**

Develop a demo application in Unity, which will allow creating a screenshot from the actual scene.

**The demo application has to meet the following criteria:**

- The user will have available a toolbar with several buttons. In the demo application, only one button from this toolbar will work: **freehand drawing with changeable colour.**

- In this demo application, the user will be able to free-hand draw in the screenshot and save it to the filesystem,

- The drawing must work also with the mouse, but also on touchscreen/mobile/tablet.

1. **Application Outline**

The project is basically an application (Windows and Android) that focuses on letting the user draw and taking screenshots of the scene. The application needs to be able to run on both windows and mobile/tablet.

1. **Tool suite**

**Unity and C#:** The application was developed using Unity3D version 2019.3.12. Unity 3D is a cross platform game engine, such that I have support for both Windows and Android. For scripting I’ve used C#.

**Github:** I’ve used Github as a version control for the project.

**Figma:** I’ve used the Figma website (link below) to design the User Interface. Designing the UI before implementation leads to creating an intuitive and user friendly UI.

<https://www.figma.com/proto/SmYJbaxZPUDUyrtsj0RWPU/FreeHandDrawing?node-id=2%3A3&scaling=scale-down>

**Google drive:** To write and store documents. After that I exported them and uploaded them on Github.

1. **User Interface**

The UI consists of 2 main parts, “UiButtons” and “ColorPicker”.

The “UiButtons” contains functions to perform various actions of the application. For example:

* Change the color of the brush
* Erase all the lines that the user has drawn.
* Save the screenshot to the application path.

The “ColorPicker” represents the color palette which can be open by clicking on the color button. There are 4 variables of the ColorPicker:

* R
* G
* B
* BrushColor

The color of the brush (BrushColor) will be determined by the other 3 variables(R, G, B) according to the RGB color model.

1. **Implementation**

**Changing the brush color function:** To manage the brush color. I’ve created a colorPicker script for the ColorPicker GameObject. With a public Image for the BrushColor(image). There are 3 float variables in the scripts that are: r, g, b. The variables will be updated when the user edits the sliders. After that, from the 3 variables I will create a new color and set it to the BrushColor(image).

**Draw function:** To implement the draw actions. I have created a freeHandDraw script which has 1 public GameObject linePrefab(GameObject). The script will take the fingerPosition/mousePosition as a Vector list. From there it will instantiate the Prefab. Also in update() function I have checked if the user drew the first line or drew a new line. Because there are UiButtons on the screen, I also checked if the EventSystem point over GameObject(UiButton). If yes then it wouldn’t draw the line. I did this because I don’t want the application to draw lines under the UiButtons and it will affect when the user takes a screenshot of the scene.

**Erase function:** How the draw function works is that it will create clones of the original Prefab. So I sort all the GameObject with the same Tag = “Line” and then check if there is a word “Clone” in their name and delete those GameObject.

**Save screenshot function:** The script ScreenShotHandler is for saving the screenshot. The script requires inputting the width and height of the screenshot(I save it as public which are: w=971, h=402. The scale of the mainUI).

1. **Conclusion**

To create this application I have consulted/referred to some videos on youtube and some posts on stackoverflow:

<https://www.youtube.com/watch?v=pa_U64G7gkE&t=1s>

<https://www.youtube.com/watch?v=41HmgJA2cG0&t=207s>

The icons of the button are downloaded from: <https://freeicons.io>