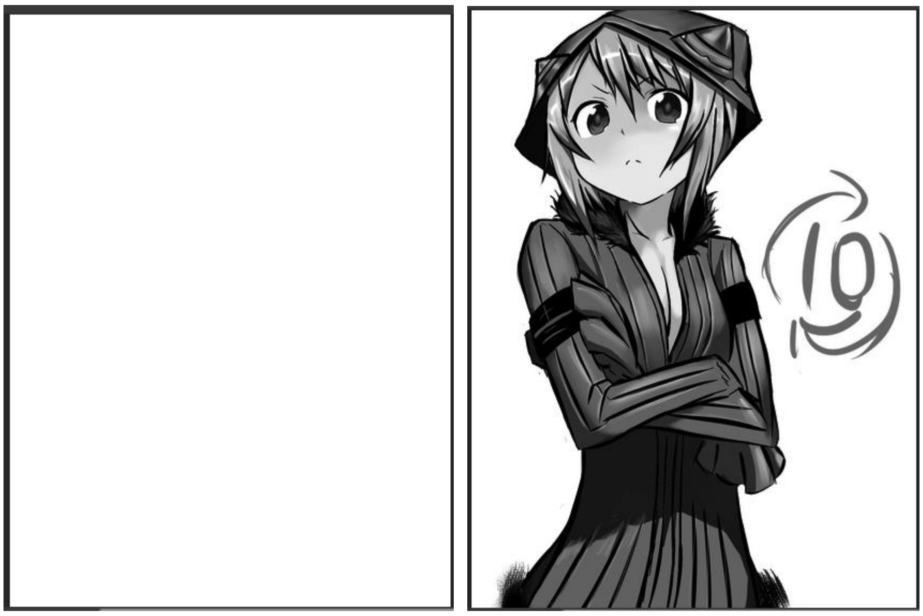
Hidden image



A single image can show two different pictures by changing the background color(black/white). The left one with white background shows nothing and the right one with a black background has a character. This kind of image is occationaly seen at mobile applications which display the image with white background and when the user click on and open the image it will be shown at fullscreen with black background.

After reading some articls about this kind of image, it seems that alpha blending is the key to realize this technic. For alpha blending, to caculate the color:

$$\mathrm{Color} = \mathrm{Color}_{front} * \mathrm{Alpha} + \mathrm{Color}_{back} * (1 - \mathrm{Alpha})$$

For image 1, we have: $\mathrm{Color}_1 = (\mathrm{r}_1, \mathrm{b}_1, \mathrm{g}_1, 1)$

For image 1, we have: $\mathrm{Color}_2 = (\mathrm{r}_2, \mathrm{b}_2, \mathrm{g}_2, 1)$

For two images mixed: $\mathrm{Color}_{mix} = (\mathrm{r}_{mix}, \mathrm{b}_{mix}, \mathrm{g}_{mix}, 1)$

Then we get the equation:

$$\begin{cases} r_1 = r_{mix} * a_{mix} + (1 - a_{mix}) \\ g_1 = g_{mix} * a_{mix} + (1 - a_{mix}) \\ b_1 = b_{mix} * a_{mix} + (1 - a_{mix}) \\ \\ r_2 = r_{mix} * a_{mix} \\ g_2 = g_{mix} * a_{mix} \\ b_2 = b_{mix} * a_{mix} \end{cases}$$

The solution should be:

$$\begin{cases} a_{mix} = 1 - r_1 + r_2 \\ a_{mix} = 1 - g_1 + g_2 \\ a_{mix} = 1 - b_1 + b_2 \\ \\ r_{mix} = r_2/(1 - r_1 + r_2) \\ g_{mix} = g_2/(1 - g_1 + g_2) \\ b_{mix} = b_2/(1 - b_1 + b_2) \end{cases}$$

Obviously, the solution would be simple if it's a grayscale image because $\mathrm{r} = \mathrm{g} = \mathrm{b}$

Thus, the solution for the mix image is:

$$\begin{cases} a_{mix} = 1 - r_1 + r_2 \\ r_{mix} = r_2/a_{mix} \end{cases}$$

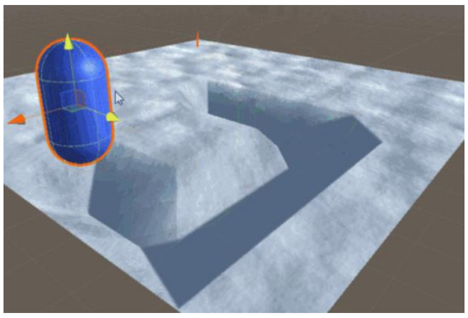
Moreover, in consider of the range of alpha ([0,1]) we must fix the value of r_1 and r_2. The appropriate way is to scale down the value of r_2.

Optional:

Since the topic above might be away from the shader we talked in class, I got another optional project idea about the Snow Deformation.

That is common in a video game when the character works through the snow field and remains a trail behand.

The simple idea of this effect is to record the position of the character and use the vertex shader to change the vertex position at the snow mesh. Here is a sample found on internet:



Much easier, I think it can also be done by using the height map and change the value of height map to make the deformation.Additionally, there is much to do with the Snow Deformation. I found some introduction at GPU pro 7 which talks about how to calculate the snow trail more realistic.

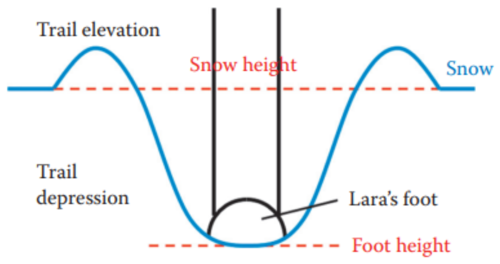


Figure 1.2. The various components of a snow trail.

Part2: This work might be done individually, for both project seems enough work for one person.