

学号	姓名	论文规范性 (10)	问题分析与调研 (30)	方案创新性 (20)	实验结果分析与讨论 (40)	结课论文总成绩 (100)
21301172	许一涵	7	25	17	35	84

屏幕水幕效果调研与实现

作者：许一涵

学号：21301172

摘要：目前，水幕效果被较多的应用于游戏、影视等片段中，模拟了玻璃上有水珠的效果，能够较好的模拟雨天的情况。目前，水幕效果的实现有较多的方法，本人使用较为简单的基于屏幕空间的水幕效果实现了显示器上有水珠的效果，并对实验结果进行了简单的分析。

关键词：水幕、屏幕空间、水珠效果

目录

屏幕水幕效果调研与实现	1
引言	3
相关工作介绍	3
方法描述	4
实验结果与分析	6
结论	6
参考文献	7

引言

水在游戏中的实现一直是一个难题。水因为其具有反射、折射、流动性等性质，因此对图形学的算法提出了巨大的挑战。

而在本人的电脑中安装了壁纸软件 wallpaper engine，并且使用了一张毛玻璃下雨的视频作为壁纸。该壁纸的效果如图 1 所示。该壁纸通过玻璃上有水珠的效果以及毛玻璃效果实现了真实的下雨场景。

这种水幕效果在游戏开发中也有广泛的应用。在赛博朋克 2077 等 3A 游戏，开发者大量使用了这种效果来营造真实的雨天效果，用来增加画面的真实感与玩家的沉浸感。

因此，本次课程论文中本人打算深入调研该效果的实现方式并且自己实现一个简易的基于屏幕空间的水幕效果，实现一个下雨的视觉效果。

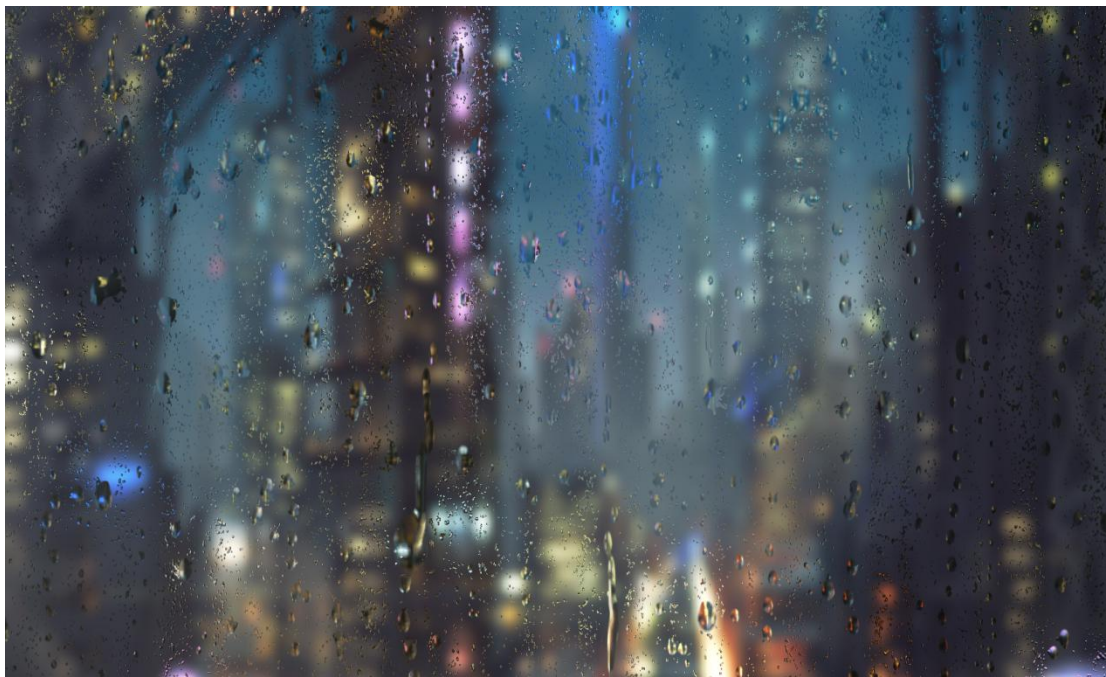


图 1：水幕效果壁纸

相关工作介绍

当前，实现水幕的效果已经较为成熟，在学习完毕了 LearnOpenGL[1]相关课程后，我从网站 <https://www.shadertoy.com/> 中找到了较多能够较好的视线水幕效果的链接，[Tokyo \(shadertoy.com\)](https://www.shadertoy.com/view/4tWf8t)[2]实现了一个在雨中漫步的效果，展示了行走在东京雨中的场景（虽然作者说他没有去过东京），而 [Heartfelt \(shadertoy.com\)](https://www.shadertoy.com/view/4tWf8t)[3]和 [The Drive Home \(shadertoy.com\)](https://www.shadertoy.com/view/4tWf8t)[4]都较好的实现了如图 1 所示的效果，这些算法都使用了背景模糊加上屏幕水珠效果实现了落雨的特效。

方法描述

经过调研，基于屏幕空间的水珠算法的主要实现较为简单，主要原理是使用噪声图以及随机函数生成雨滴的位置与形状，并且实现从小到大的效果，然后通过折射公式实现水的折射效果从而达到屏幕落雨的效果。

首先需要获得一张噪声图片，在噪声图中进行采样，并且取其中 XY 两个维度的值作为偏移量，为雨水的位置添加随机性。

而后构建一个循环，循环 n 次，在每次循环中根据 n 的大小生成不同网格大小，从而控制网格中出现的水珠的概率与分布。网格大小的公式如下：

$$\text{grid_size} = \text{u_resolution} * i * c$$

其中 u_resolution 是屏幕空间大小， i 表示当前是第几次循环， c 是预先设定的常量。

得到 grid_size 之后，计算雨滴的位置，雨滴位置的计算公式为：

$$p = \sin(c_1 * \text{texcoord} * \text{grid_size} + (1 - c_2) * c_3);$$

通过这个公式，就可以得到一个相对比较自然的水滴位置，其中 $c_1 c_2 c_3$ 都是自己设置的常量。再从噪声纹理中采样 p 位置网格的数据作为雨滴的数据。为了保证雨滴落在网格里面，我们需要使用 `round` 函数对位置进行四舍五入的操作。

至此已经得到了雨点的信息，接下来就需要控制雨点的大小。算法基于时间、噪声等信息来得到一个雨点的大小 t ，这个计算公式可以人为调整，使用任意正弦函数即可。

最后，我们需要控制雨点绘制的比率，由于噪声随机采样，我们控制雨点参数的随机维度值比如 x 值小于一定数目来控制绘制的雨点的比率，防止雨点过多。我们控制大的雨点出现数目少，所以设置雨点参数满足 $d.x < (c_1 - i) * c_2$ 时出现，这样控制越大的雨滴数目越少。而后，我们控制雨点的最小值为 c_3 ，这样就形成了最基本的雨点效果。

此时场景中仅仅出现了黑色的块状物体，没有出现水滴的效果。经过调查发现是因为没有实现雨水的折射效果，只是相当在屏幕上覆盖了一层噪声。

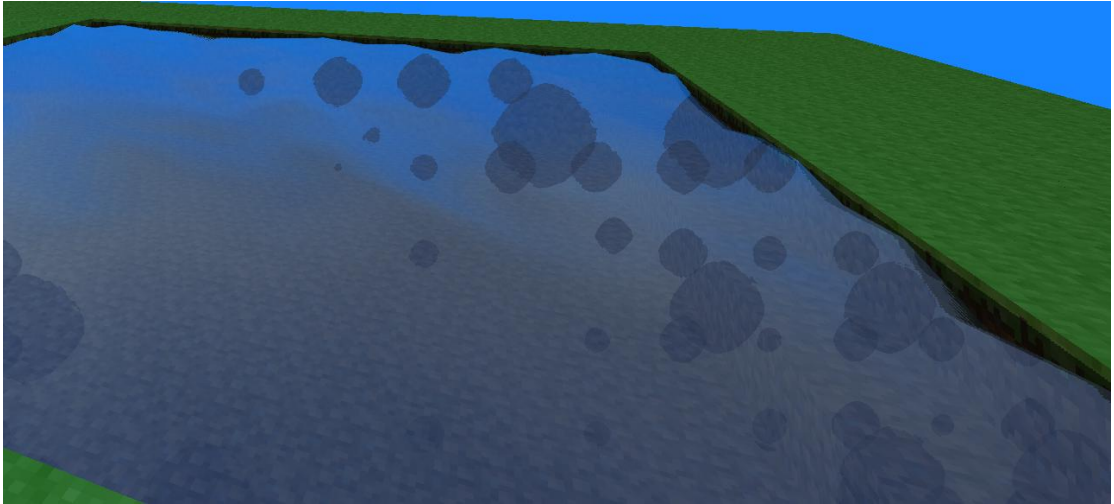


图 2: 无折射时的效果图

因此，水幕效果还需要实现简易的折射效果。我们假定固定了入射角，与折射率，因此只需要简单的计算就可以得到经过水珠后的效果。该折射效果的简单公式如下：

$$u_{\text{reflect}} = u_{\text{original}} - x * v_{\text{original}}$$

其中 u_{reflect} 是折射后的纹理， u_{original} 是原始纹理， x 是折射率， v_{original} 是水滴的法线方向，进过简单的运算就可实现折射效果。

法线的公式为 $normalize(-vec3(cos(p), mix(0.2, 2, t - 0.5)))$ ，该公式实现了水珠中间厚，四周薄的视觉效果，从而实现了雨滴的效果。

最终，我们得到了水幕的效果，该效果如下图所示

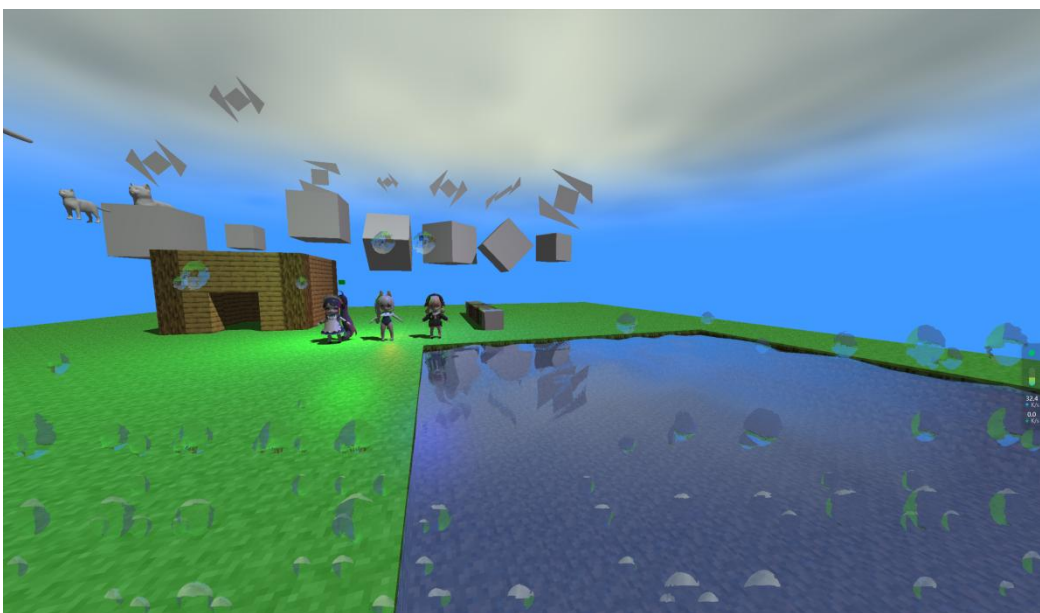


图 3: 最终效果展示

实验结果与分析



图 4：结果展示

通过本次实验，本人实现了基于屏幕空间上的水幕效果，能够模拟雨滴落在屏幕上的效果。由图四可见，水滴形态各异，并且大小不一，很好的模拟了水珠无规则的特点。在模拟水珠的特性上，通过计算水滴的法线方向，并且让法线方向从中心向四周的角度不断变大，从而让水滴四周的折射角度变大，模拟了水滴中间厚、四周薄的效果。并且水滴会随着时间由大到小变化，模拟了水滴蒸发或者流走的效果。通过联合第二次实验中实现的天气控制效果，可以对下雨天进行模拟，通过控制雨滴出现的比率，我们还能控制屏幕落雨的数目。

然而，目前实现的水滴效果仍有可以修改的地方，当前的水滴并不会流动，也不会与其他的水珠进行融合，但是由于实现难度较大，本文并未实现该效果。此外，目前实现的落雨效果水滴出现的周期性较强，通常会在一个位置反复出现，后续将考虑修改控制雨滴出现的函数并且加入噪声，让雨滴出现的更加自然无序。

结论

在本次实验中，本人实现了基于屏幕空间的水幕效果，能够较为真实的模拟出雨滴落在屏幕上的效果。模拟的步骤主要分为如下几步：

1. 通过随机函数与噪声图片采样，能够生成形态各异的雨滴形状与随机位置，并且通过正弦函数实现雨滴从大到小的状态控制。

2. 通过实现折射效果，模拟出了水滴的对光线的折射。并且我们控制水滴的折射角度从水滴中心向四周变大，从而实现出视觉上水珠中心厚四周薄的效果。

3. 通过循环生成网格，算法能够控制生成的雨点大小。

实验结果表明，我们的算法能够较好的生成大小不一，形态各异的水滴，可以有效的模拟出下雨天的情况。

参考文献

[1][基础光照 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)

[2][Tokyo \(shadertoy.com\)](https://shadertoy.com/view/4dWt8d)

[3][Heartfelt \(shadertoy.com\)](https://shadertoy.com/view/4dWt8d)

[4][The Drive Home \(shadertoy.com\)](https://shadertoy.com/view/4dWt8d)