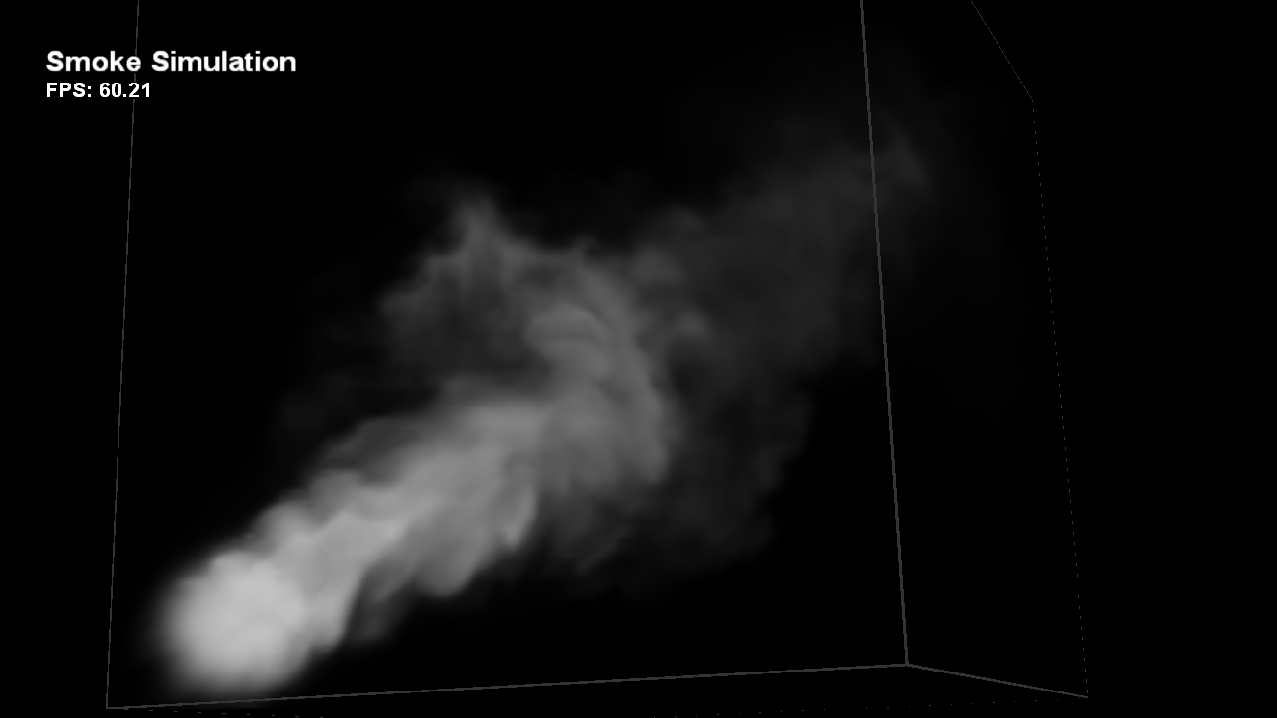


烟雾模拟

此示例与 Microsoft 游戏开发工具包预览版（2019 年 11 月）兼容

# 说明

本示例演示如何使用计算着色器 6.0 和 3D 纹理来实现基本的 3D 纳维-斯托克斯流模拟。本示例还演示如何使用简单的光线追踪算法呈现容积数据。

**

# 构建示例

如果使用 Xbox One 开发工具包，请将活动解决方案平台设置为 Gaming.Xbox.XboxOne.x64。

如果使用 Project Scarlett，请将活动解决方案平台设置为 Gaming.Xbox.Scarlett.x64。

有关详细信息，请参阅 GDK 文档中的“运行示例”。

# 使用示例

此示例使用以下控件.

|  |  |
| --- | --- |
| 操作 | 游戏手柄 |
| 退出示例。 | 选择 |
| 旋转相机 | 右摇杆 |
| 旋转发射器 | 左摇杆 |
| 移动发射器 | 左/右扳机键 |
| 重置发射器方向 | X 按钮 |
| 切换模拟暂停 | A 按钮 |
| 重置模拟 | B 按钮 |

# 实现说明

模拟技术

有关本示例中所用模拟技术的详细信息，请参阅《GPU 精粹》一书第 38 章“GPU 快速流体动力学模拟”。这本书介绍模拟技术并将其应用于 2D 流体，本示例将应用范围扩展到 3D 数据。在本示例中，模拟在 128x128x128 网格上运行，其状态存储在 3D 纹理中。

使用计算着色器 5.0 可以一次直接操作一个 3D 纹理的所有切片。相比采用几何着色器的备选方案，它的效率更高，可以用来指定要更新的 3D 纹理切片。

呈现技术

模拟会生成一个 3D 速度场，它表示某个模拟步骤中流体的状态。由于我们无法直接看到速度，所以我们需要有能够看到的、围绕在速度场边的示踪粒子。另外，除模拟本身所需的所有 3D 纹理之外，我们还提供了一种 3D 纹理来存储这些粒子。这种 3D 纹理是在呈现阶段可视化的纹理。

呈现时，将呈现立方体的正面。在像素着色器中，我们在立方体上从眼睛的位置向当前点投射出一条射线，沿射线以固定的时间间隔对前文中提到的 3D 纹理进行采样，同时积累颜色和不透明度。

本示例还演示了一种简单而高效的光线投射算法加速技巧。在模拟过程中，使用并行缩减生成 1/8 大小的 3D 纹理。在呈现时，我们不直接以固定的时间间隔对原始 3D 纹理进行采样（这可能会浪费大量的时间去采样空的空间），而是先对这个 1/8 大小的 3D 纹理进行采样，如果采样值小于阈值，则跳过空间中的区块。如果 3D 纹理中存储的数据是稀疏数据（例如本示例中的烟雾），则此技巧尤其有效。

# 更新历史记录

2019 年 4 月 - 从旧 Xbox 示例框架移植到新模板。

# 隐私声明

在编译和运行示例时，示例可执行文件的文件名将发送给 Microsoft，用于帮助跟踪示例使用情况。要选择退出此数据收集，你可以删除 Main.cpp 中标记为“示例使用遥测”的代码块。

有关 Microsoft 的一般隐私策略的详细信息，请参阅《[Microsoft 隐私声明](https://privacy.microsoft.com/en-us/privacystatement/)》。