

游戏 Boat 关键技术列表

1191230303 温柔 1191230306 郭子璇

1191230314 张妤帆

摘要：我们开发了一款 Unity 游戏《Boat》，并在其中应用到了两种关键技术作为游戏的基础玩法：**基于近似凸分解与 Voronoi 分区的实时刚体破碎技术**和**基于 Shader 实现的水波纹物理模拟系统**。将技术应用到复杂游戏场景中并且优化性能实现了较好的游戏视觉效果和趣味性。

•基于近似凸分解与 Voronoi 分区的实时刚体破碎技术

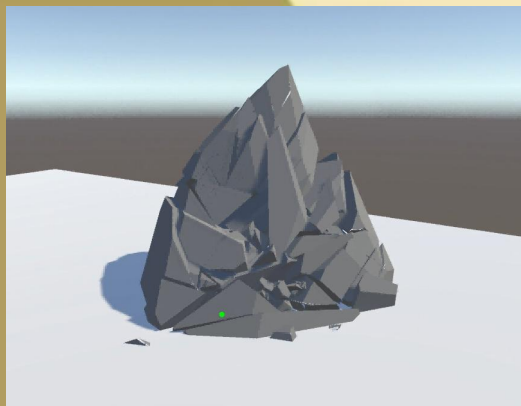


图 1 实时破碎效果

Voronoi 近似凸分解 (VACD): 基于 Voronoi 图的空间划分思想，通过迭代平面切割，将物体分解为多个凸多面体碎片。

分层抖动采样 (Stratified Jitter Sampling): 摒弃纯随机采样，根据物体形状（包围盒长宽比）自适应划分三维网格，并在每个网格单元内进行**受限的随机抖动**来生成种子点。该方法在保证碎片分布随机、自然的同时，强制控制了碎片的最小尺寸和均匀度，避免产生无效的细长薄片。

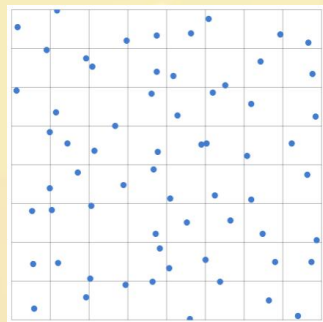
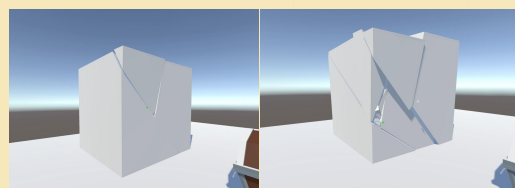


图 2 分层抖动生成的随机种子

迭代裁剪 (Iterative Clipping): 将三维物体视为一个“核心”，每次使用一个由种子点定义的平面，从核心上“切”下一块作为独立碎片，剩余部分作为新的核心继续切割，直至所有种子点用完。确保所有碎片都天然适配物理引擎，是实现物理稳定破碎的数学基础。



• 基于 Shader 实现的水波纹物理模拟系统

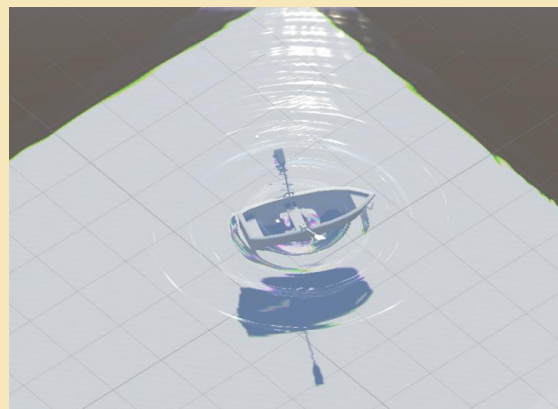


图 3 水波纹模拟效果

GPU 驱动的波动方程模拟：将二维波动方程在 GPU 上（Shader）进行**离散化和并行化**求解。每个像素点的高度变化由自身上一帧的高度和周围四个邻居像素的当前高度共同决定，在像素着色器中并行计算，实现极高的运算效率。

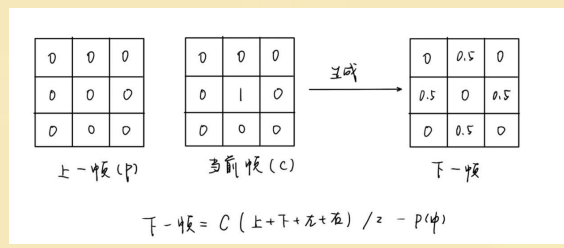


图 4 离散化波动方程原理

双缓冲 RenderTexture 帧缓冲技术：两张 RenderTexture（CurrentRT 和 PrevRT）交替读写实现帧间数据传递。