

天谕手游-鱼群

白欣

网易雷火 - 资深技术美术工程师

天谕手游鱼群效果



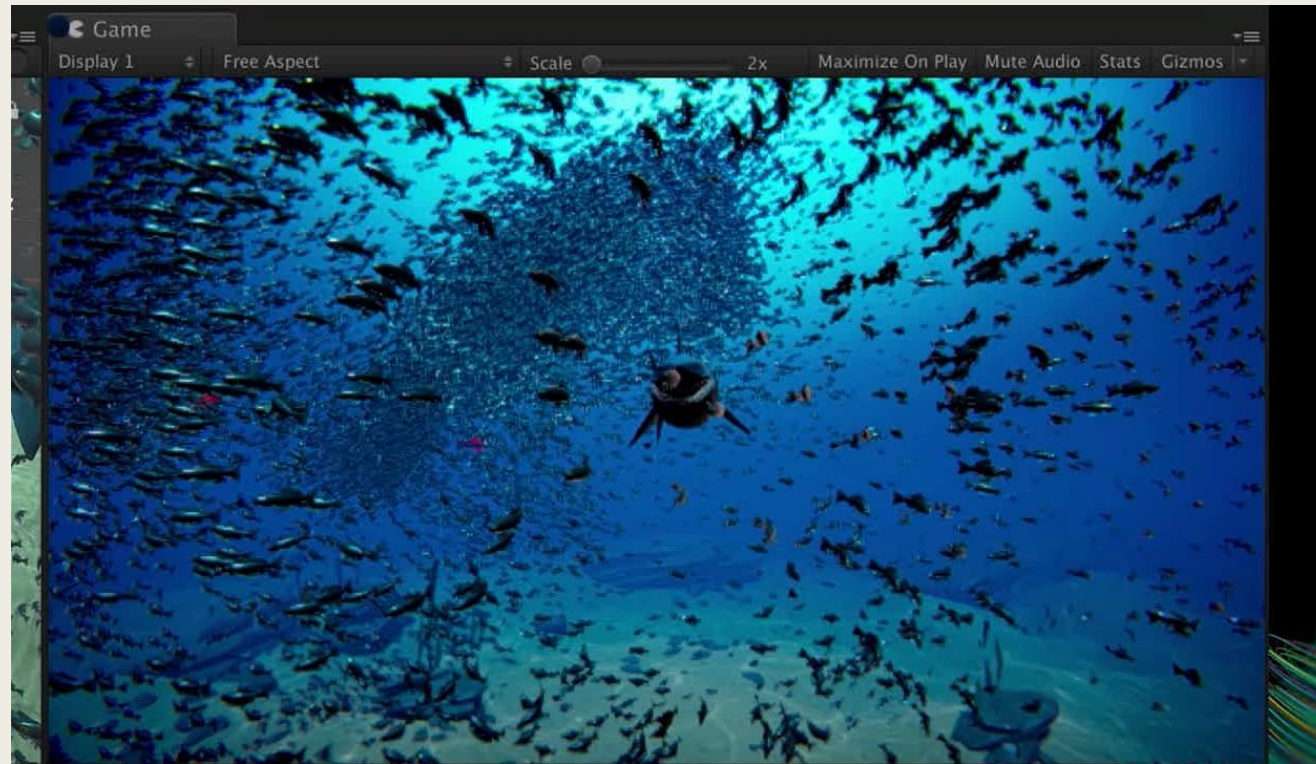
主要内容

- 鱼群基础算法
- 鱼群行为优化
- 鱼群性能优化
- 其他Tips

Unity基于DOTS的鱼群分享

<https://www.youtube.com/watch?v=p65Yt20pw0g>

<https://github.com/Unity-Technologies/EntityComponentSystemSamples>



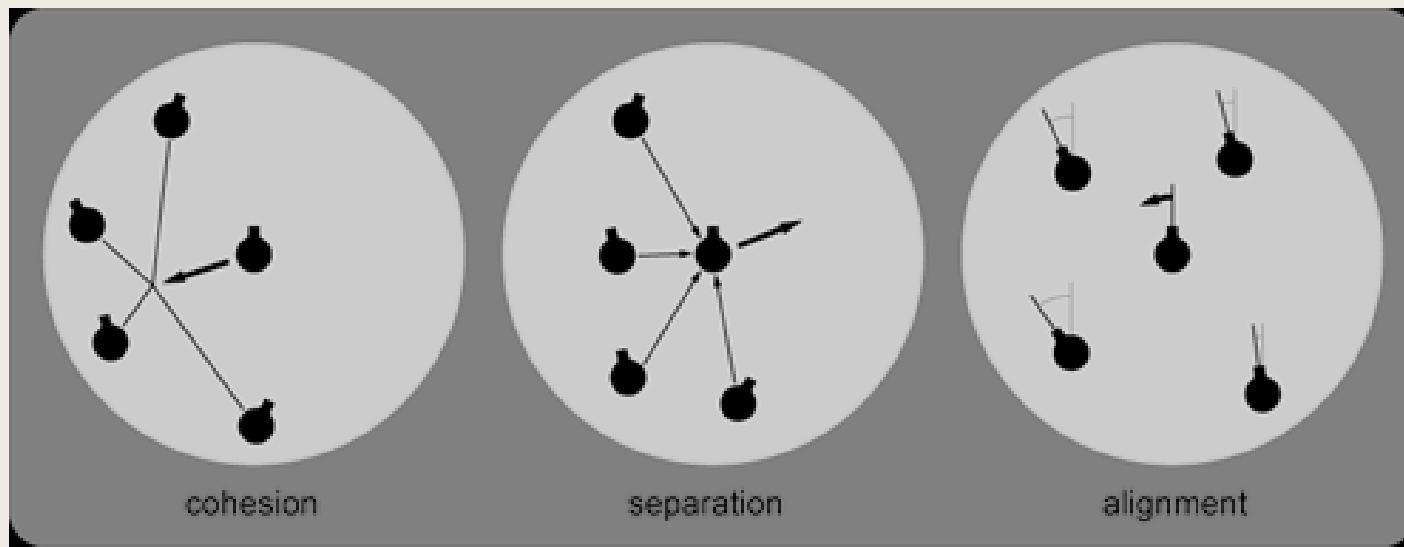
DOTS数据驱动技术栈

- JobSystem
 - 多线程并行计算，充分利用多核优势
- EntityComponentSystem (ECS)
 - Structure Of Arrays内存布局，提高缓存命中率
- Burst Compiler
 - 针对多线程和ECS内存布局开发的编译器，生成高性能运行代码

Boids基本算法

<https://www.red3d.com/cwr/boids/>

- Cohesion内聚方向：靠近周围个体的方向
- Separation排斥方向：远离周围个体的方向
- Alignment对齐方向：周围个体的平均方向



CellHash优化

- 获取周围鱼的数据
 - $O(n^2)$ 复杂度过高
 - 简化为CellHash
- NativeMultiHashMap
 - 一个Key对应多个Value
 - 多线程写安全

16 cell grid-size, 10 moving entities

12	13	14	15
		D	
8	F	E	C
4		6	7
0	A	1	3
I	G		

Pivots

Cell	U, I, F
0	3, 0, 3
1	0, #, #
2	0, #, #
3	0, #, #
4	0, #, #
5	1, 3, 4
6	0, #, #
7	2, 4, 6
8	1, 6, 7
9	2, 7, 9
10	0, #, #
11	0, #, #
12	0, #, #
13	0, #, #
14	1, 9, 10
15	0, #, #

Hash Table

A
G
I
B
H
J
F
C
E
D

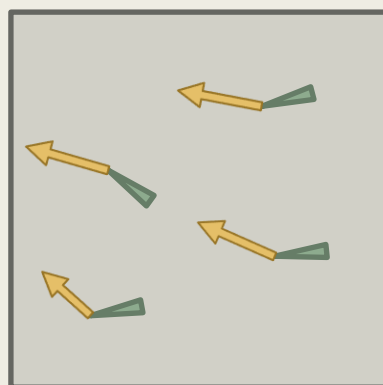
Object Index

Entity	Cell
A	0
B	5
C	9
D	14
E	9
F	8
G	0
H	7
I	0
J	7

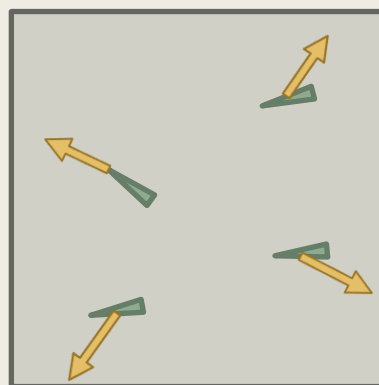
简化算法

- Target目标方向：朝向指定目标点游动方向
- Separation排斥方向：远离Cell内鱼群平均位置的方向
- Alignment对齐方向：向Cell内鱼群平均方向靠拢

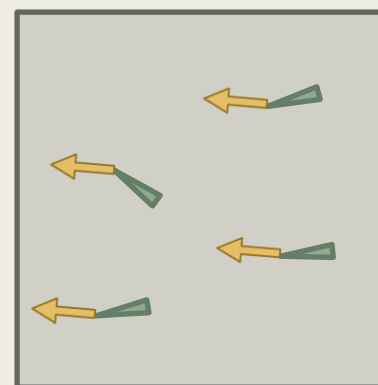
●
Target



Target方向

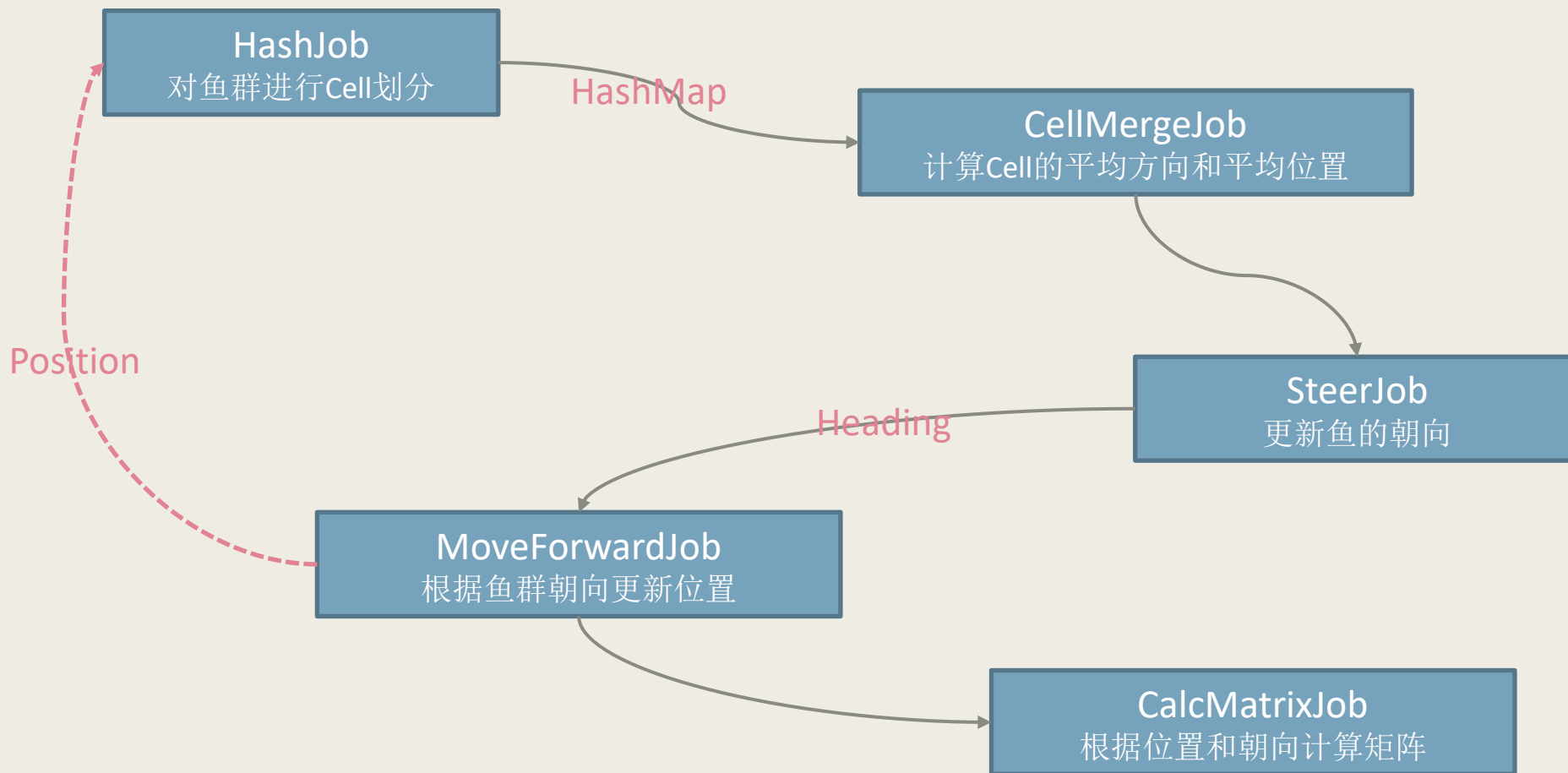


分散方向



对齐方向

鱼群基本逻辑

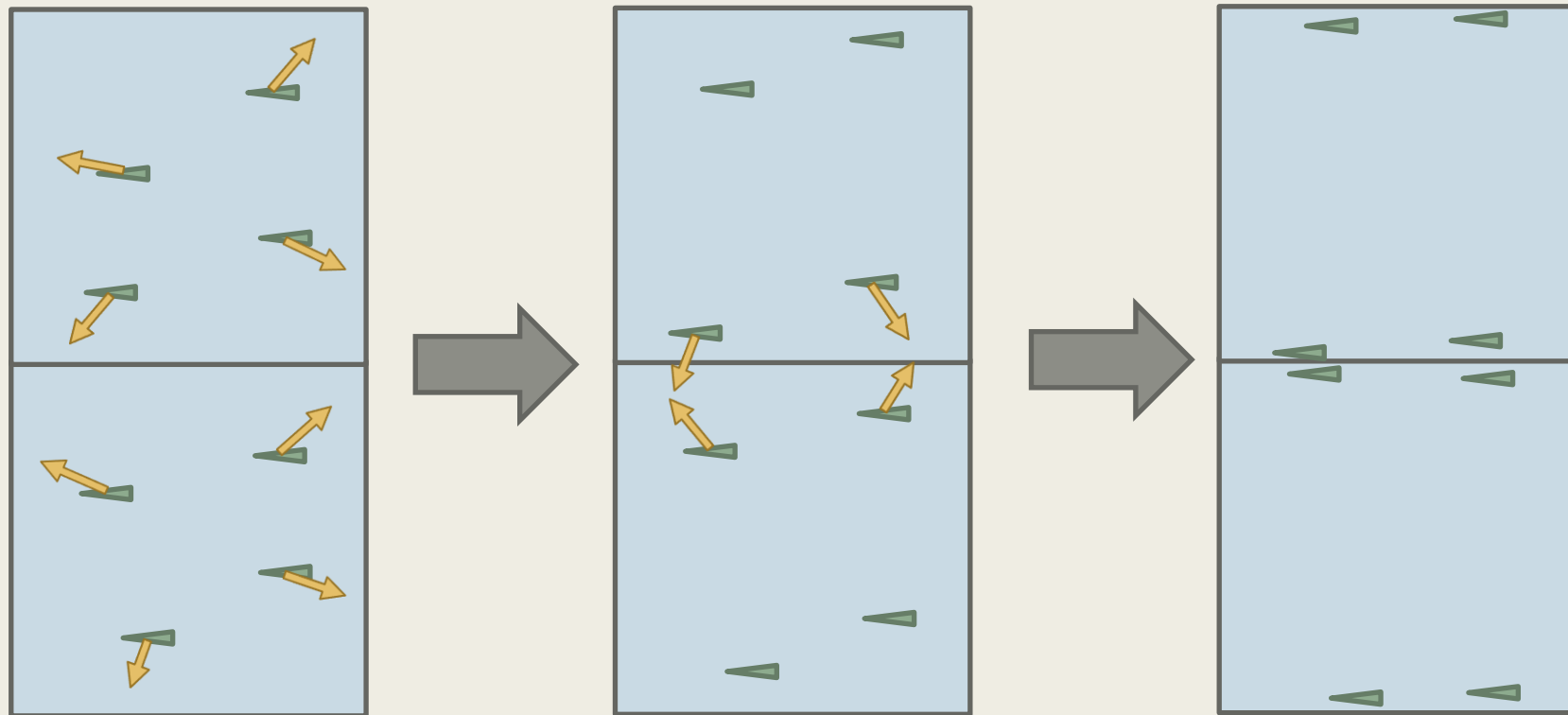


Cell对齐现象



Cell对齐现象

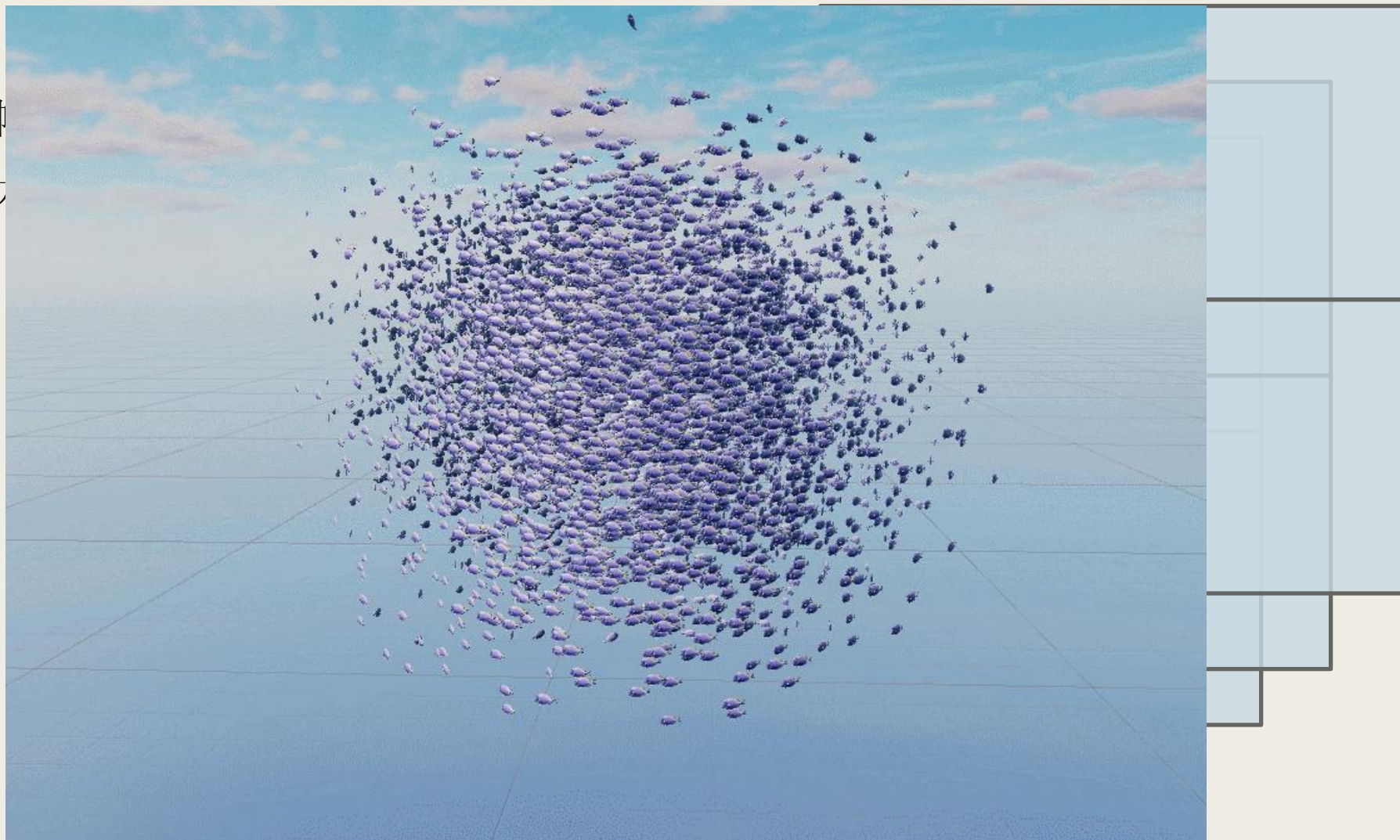
排斥方向总会向Cell接缝靠拢



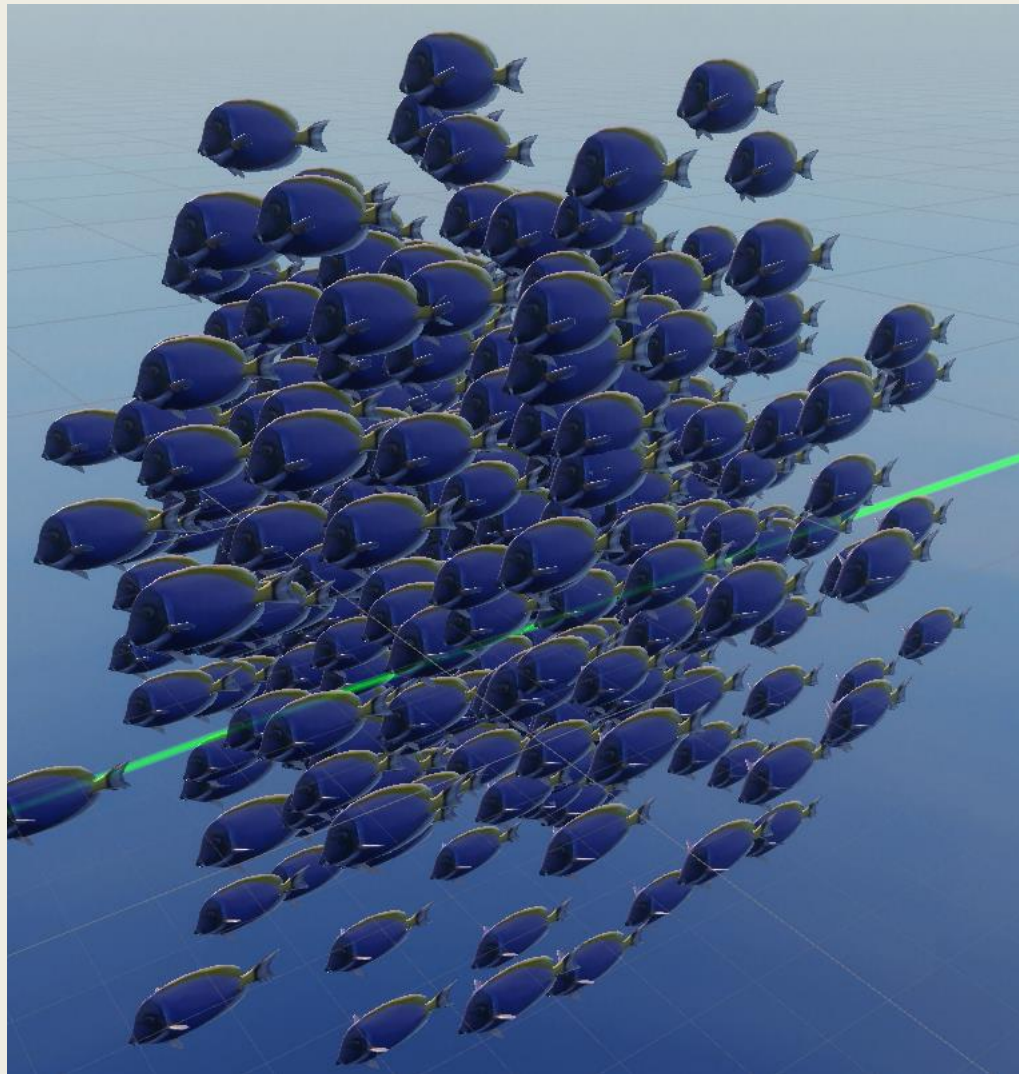
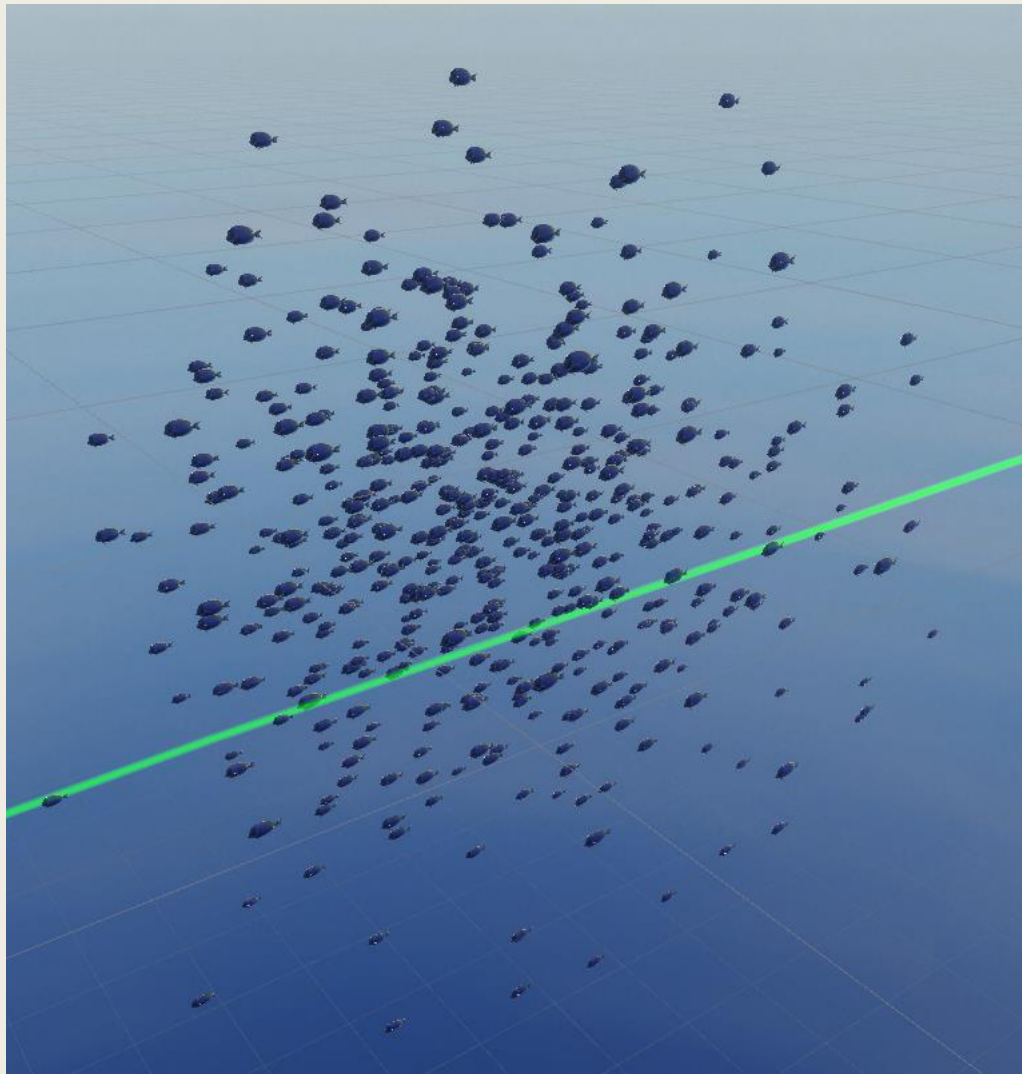
Cell对齐现象

N+2帧

Cell做小帧
Cell变化大



鱼群间距

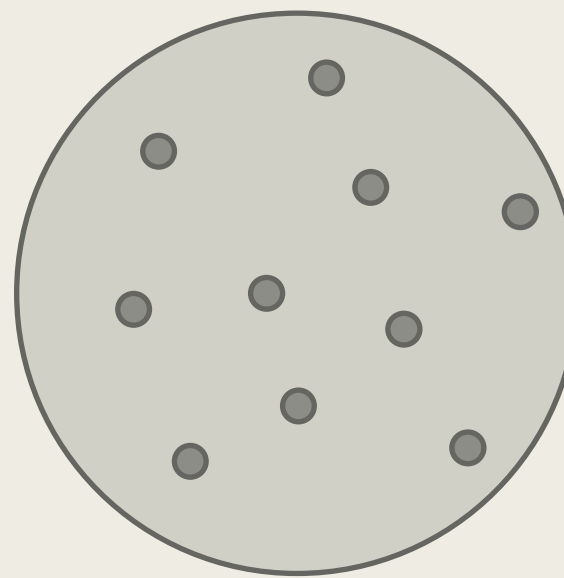
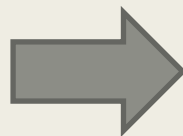


鱼群间距

将一个Target变为球内随机的目标点Targets

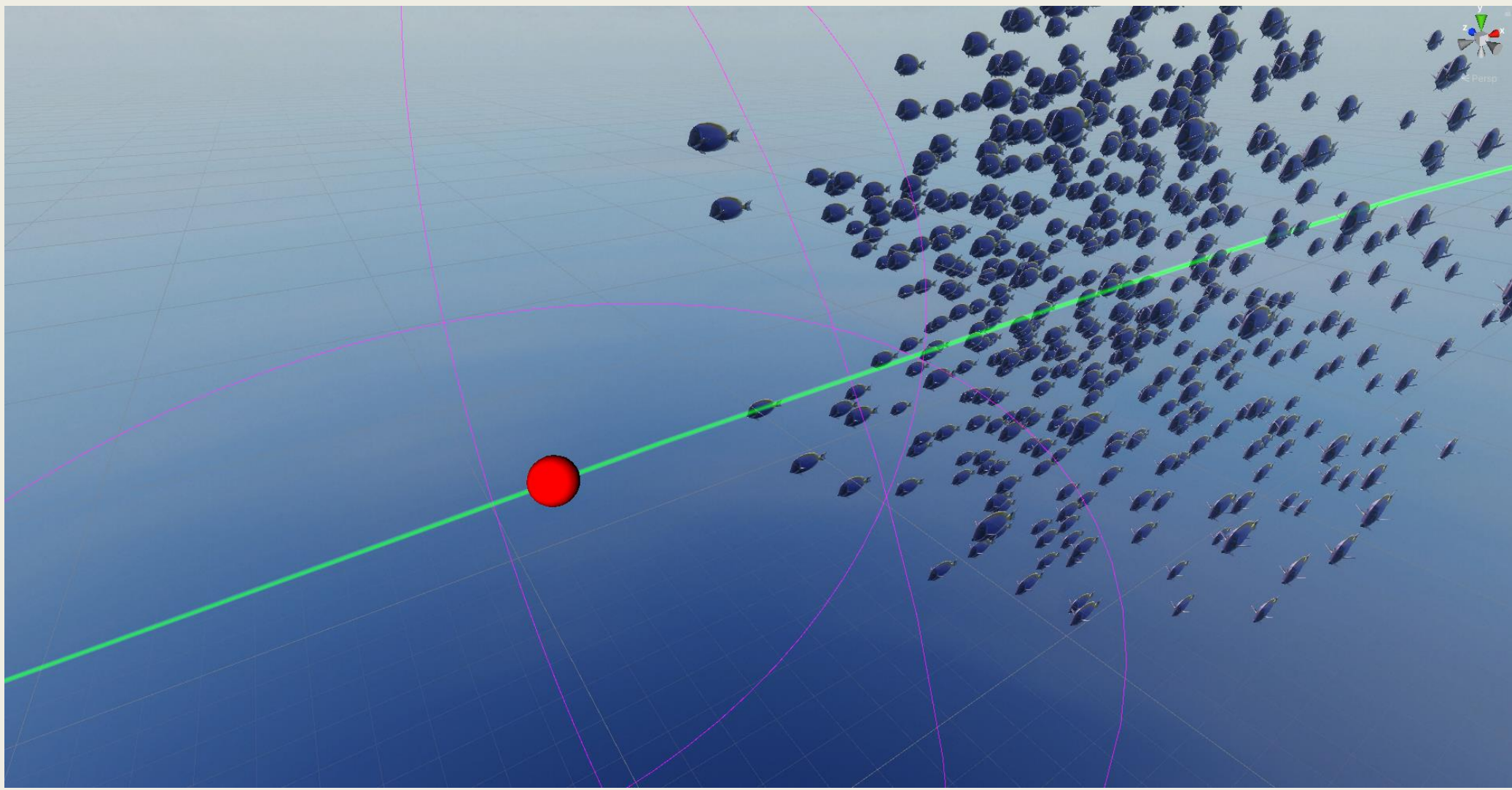


Target

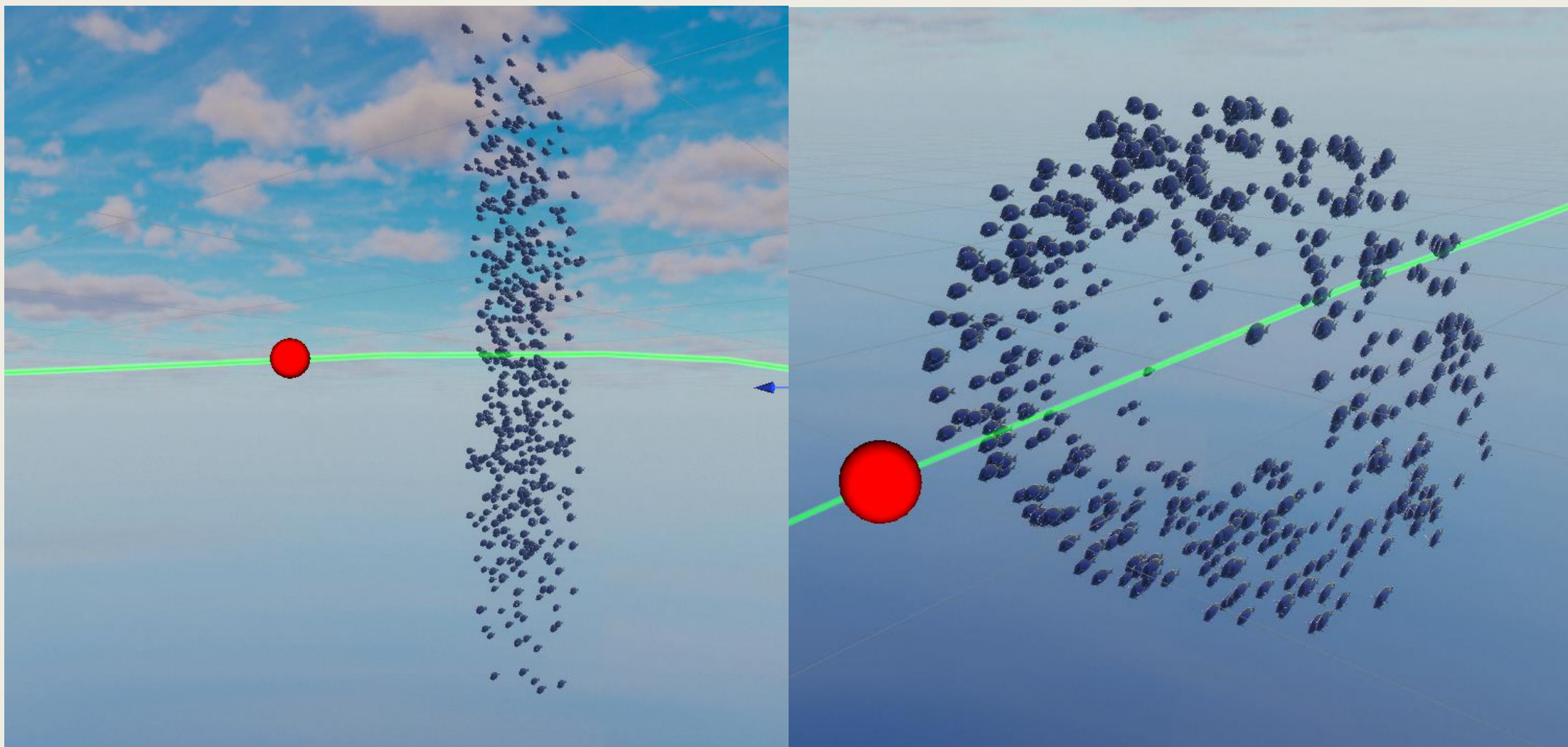


Targets

鱼群间距



鱼群形状



鱼群形状

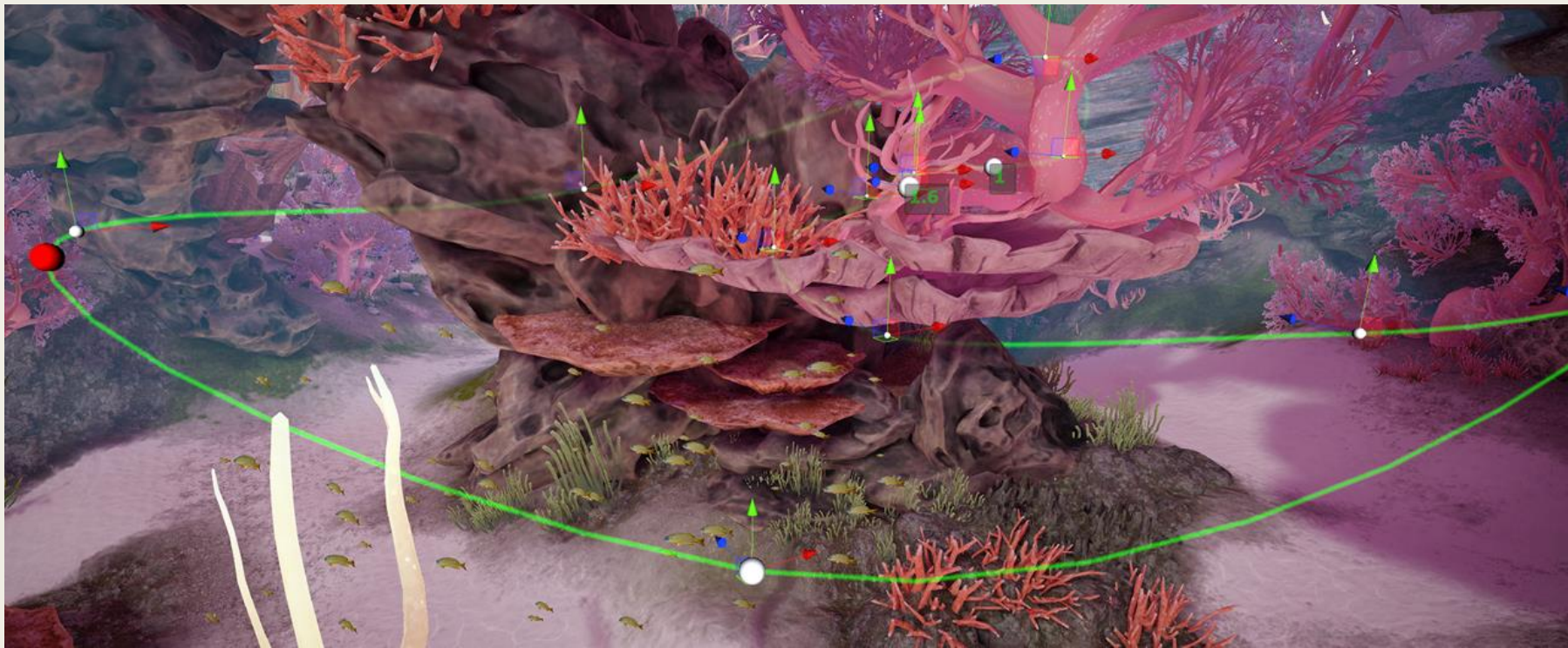
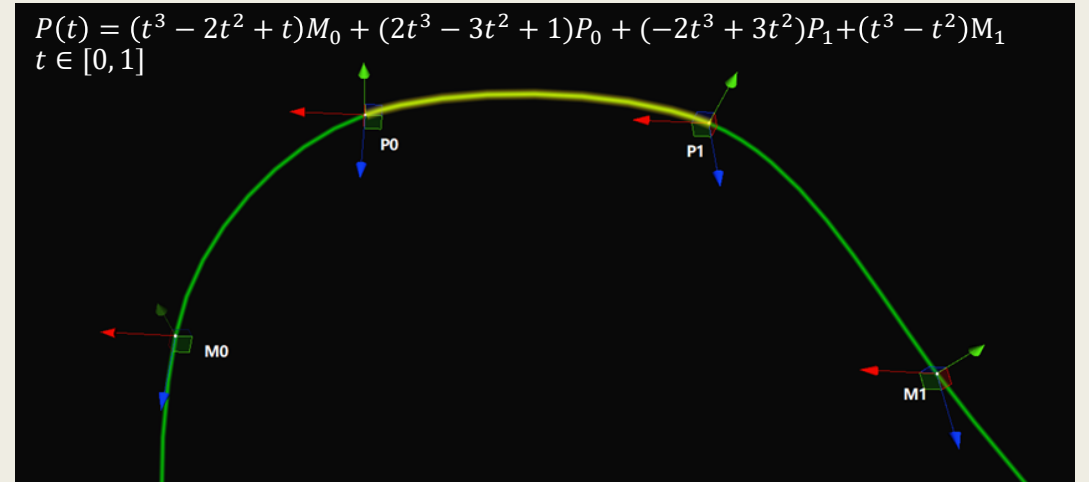
美术希望能实现沿着轨迹游动的鱼群



鱼群形状

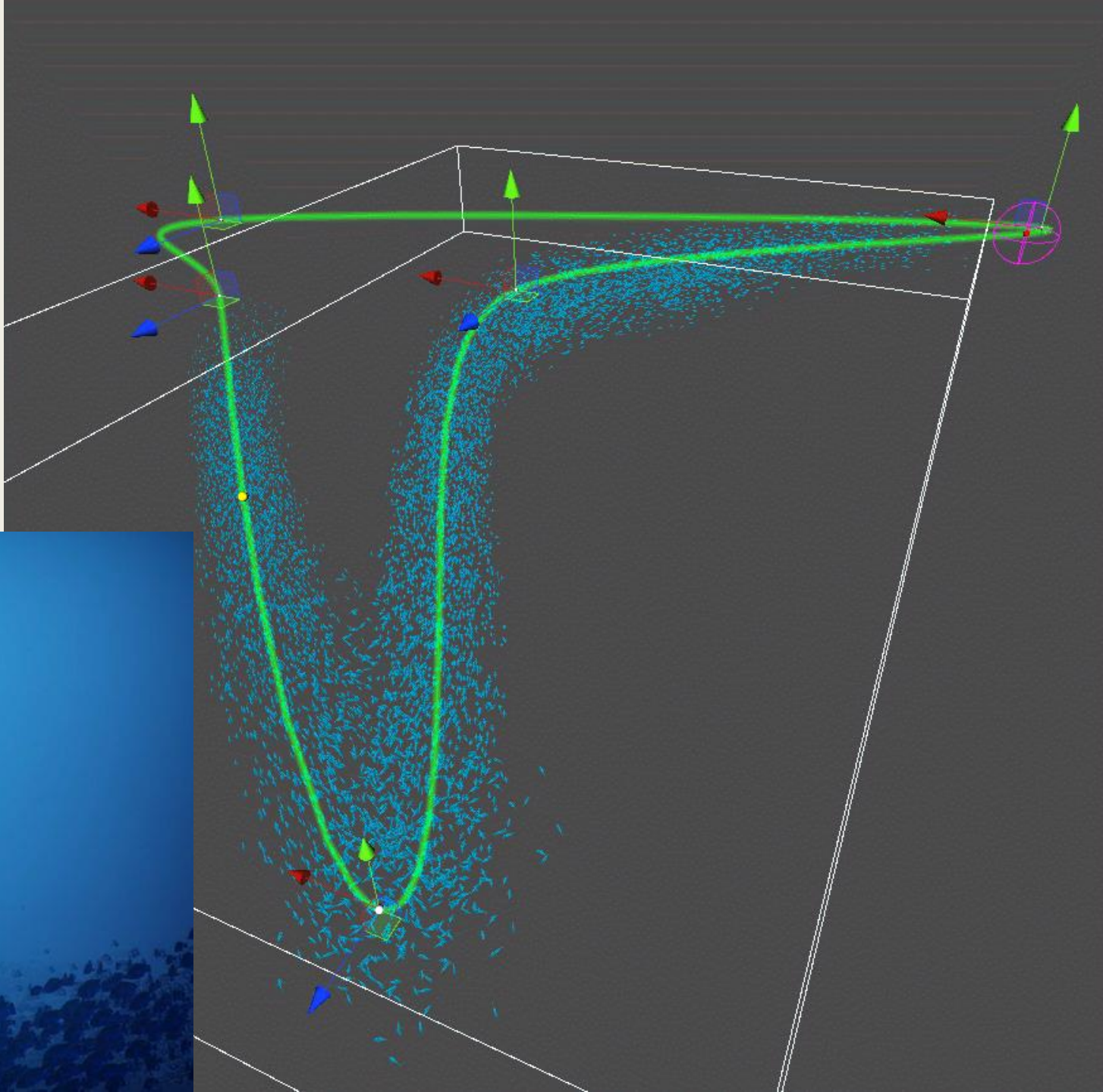
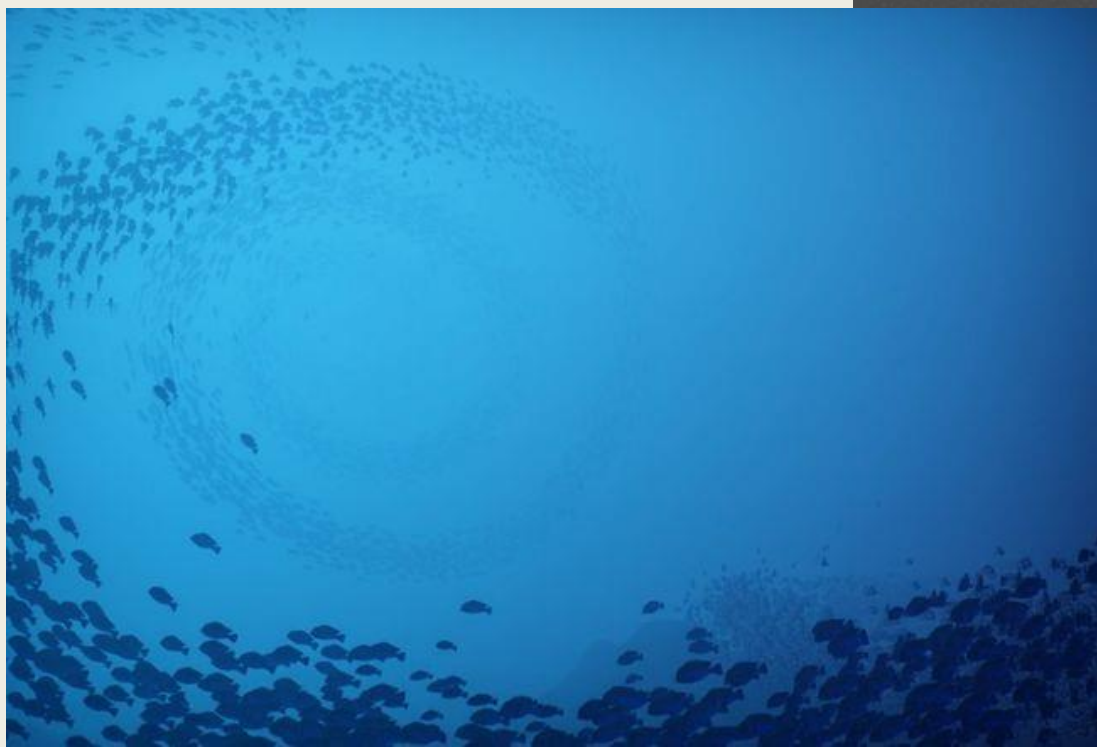
鱼群轨迹编辑器

- 利用Catmull-Rom样条线实现轨迹编辑



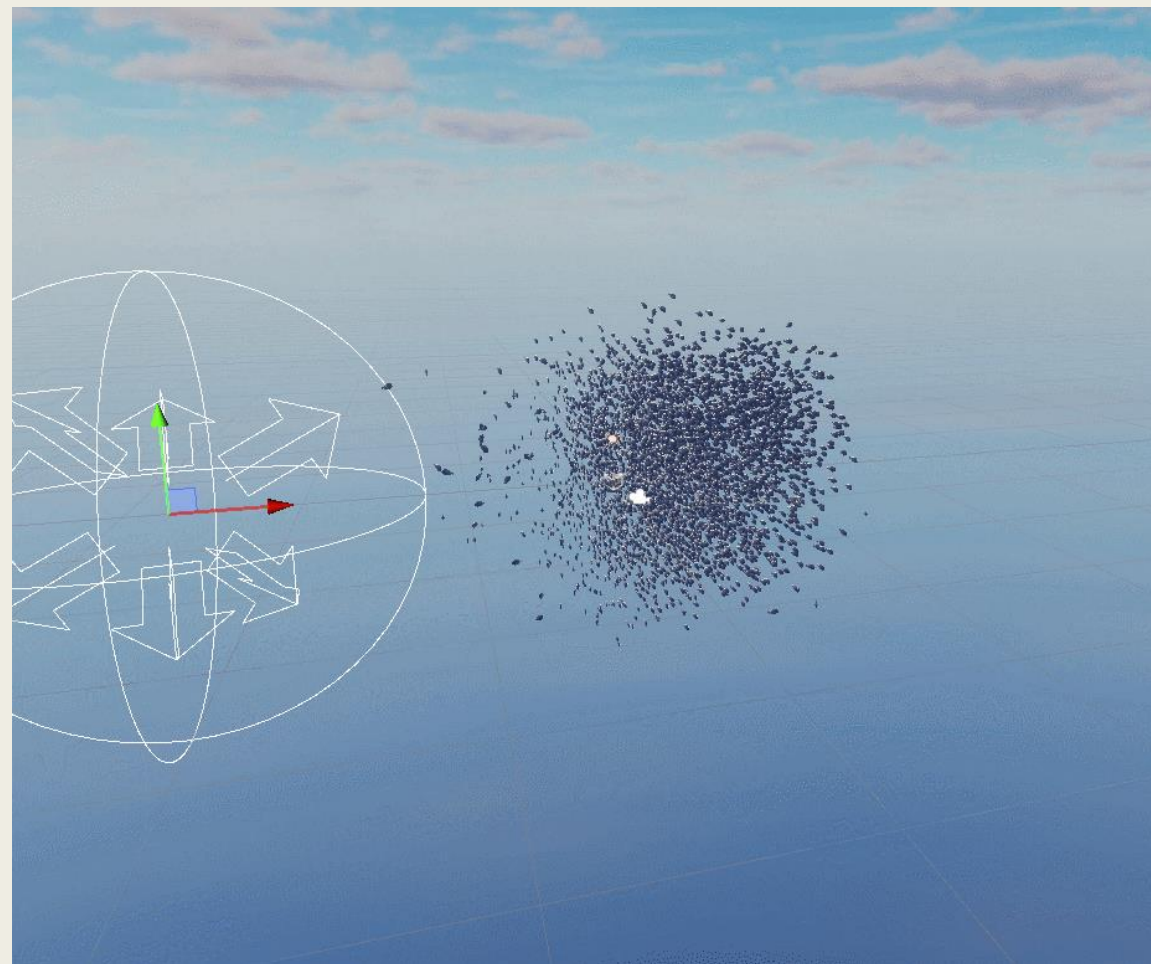
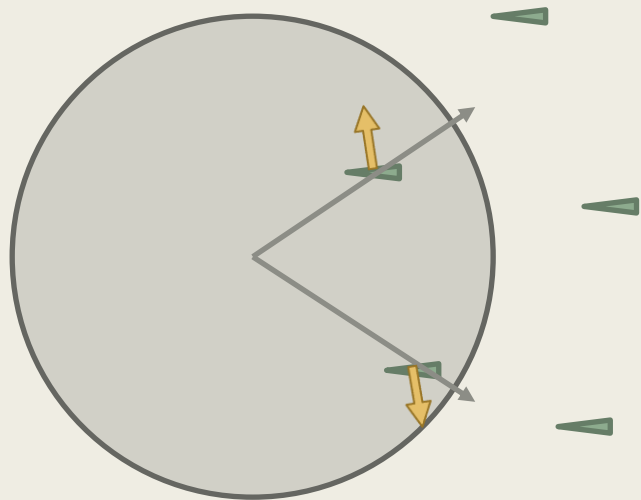
轨迹鱼群

- 将轨迹计算公式移植到Job中
- 鱼群目标位置延轨迹布置

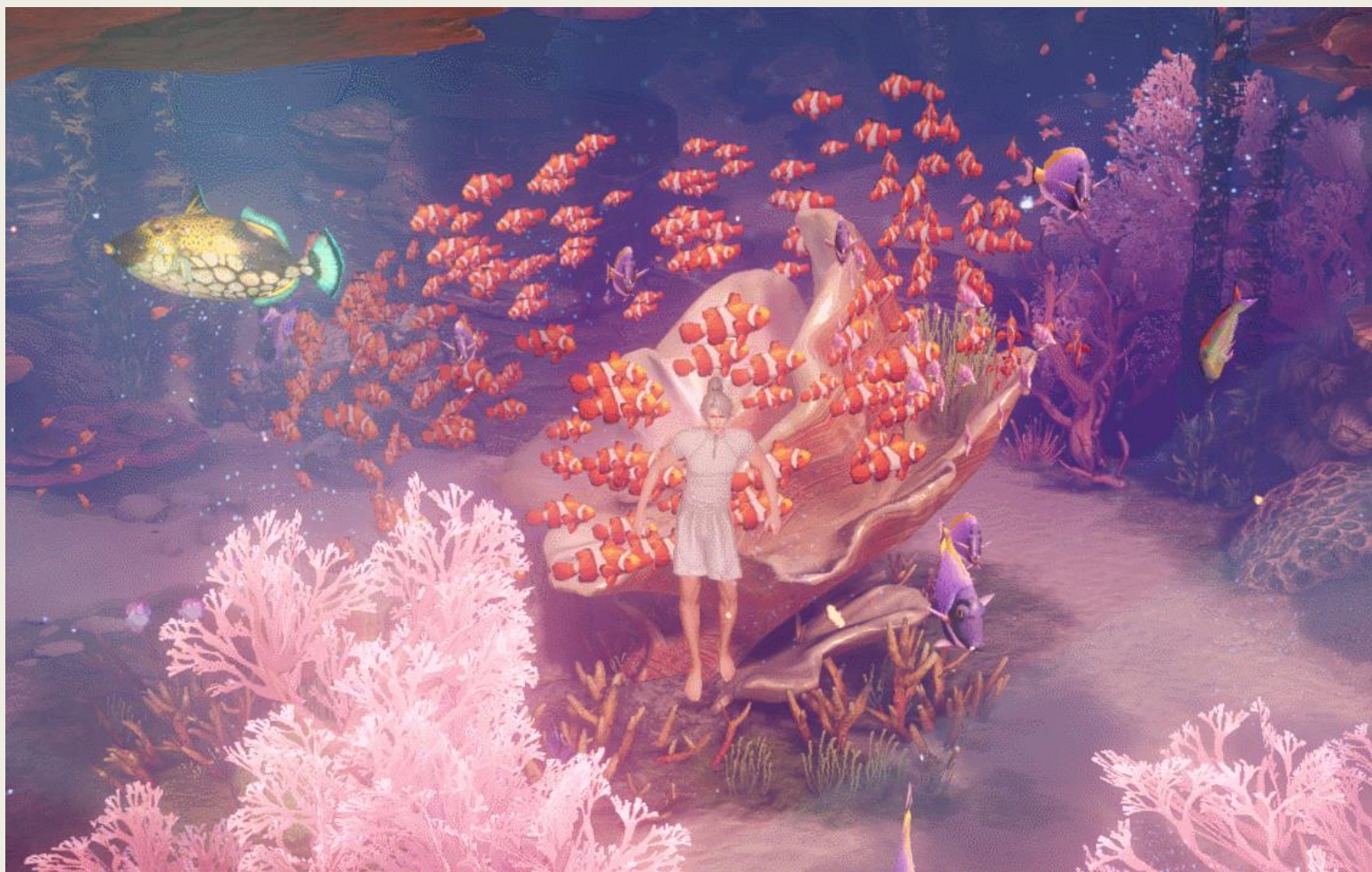


鱼群交互

球形障碍避让



鱼群交互



鱼群交互

烘焙有符号距离场（SignedDistanceField）

- 可以直接获取鱼到最近障碍的距离
- 可以通过少量几次采样确定鱼群的躲避方向

- 内存占用
 - 精度5m一个1byte记录到场景表面的距离
 - $100m^3$ 大小8KB，以Chunk流式加载，内存控制在1M以下

鱼群交互



鱼群交互

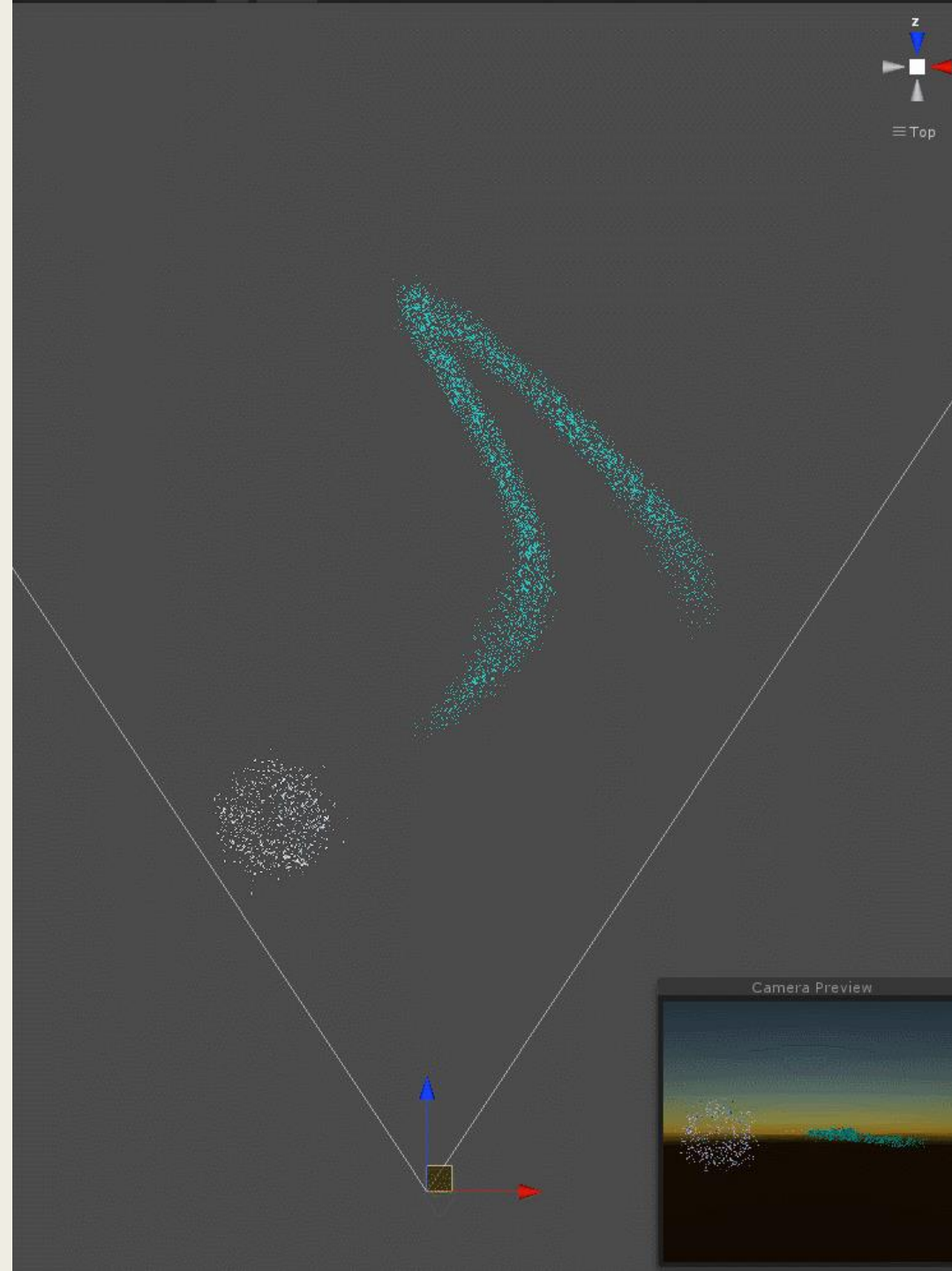
- 玩家
- 玩家

Target



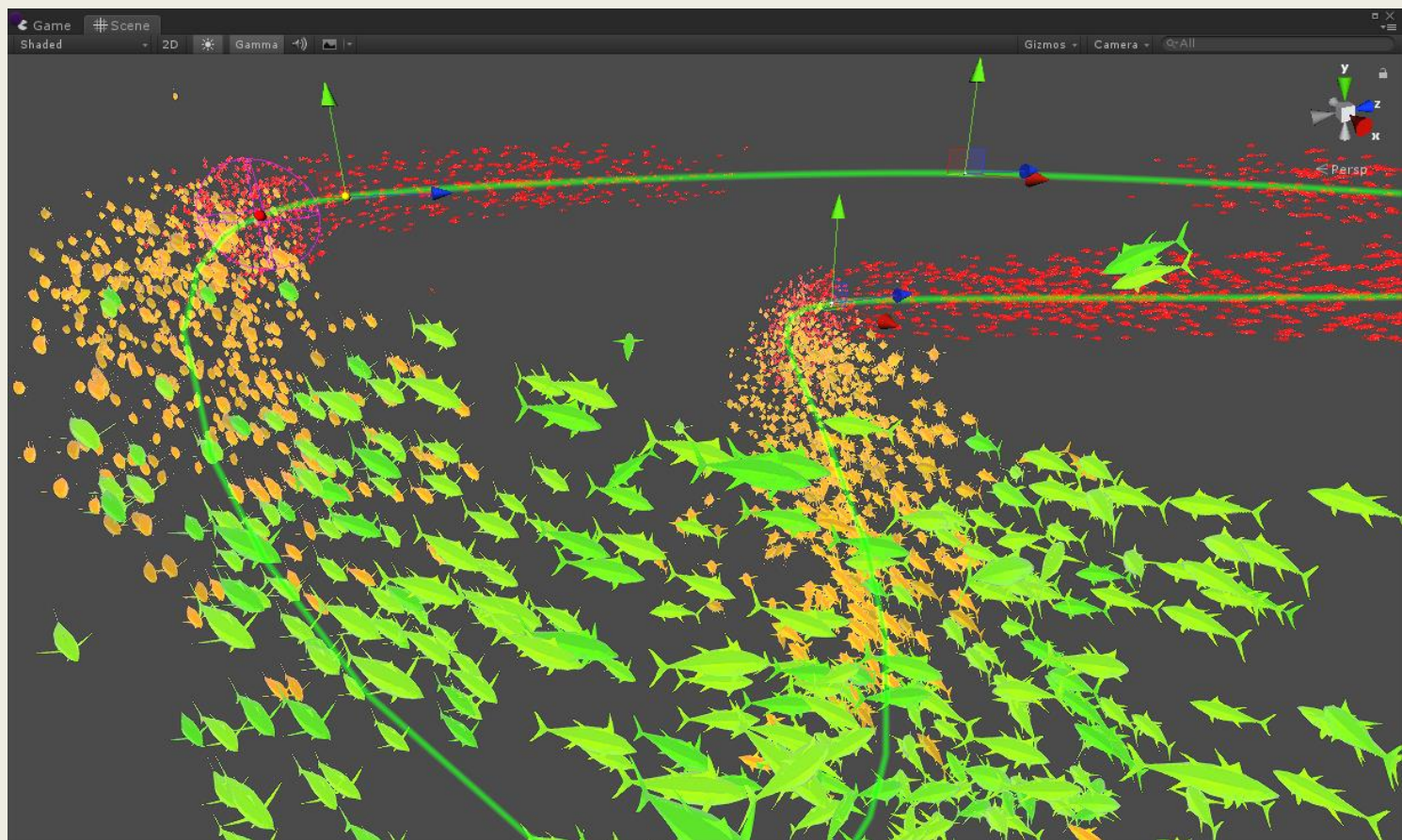
视锥剔除

将相机的裁剪面传入Job中计算视锥裁剪



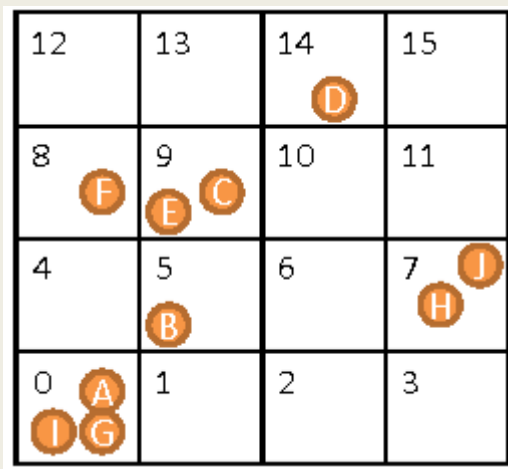
LOD

- 四级LOD
 - LOD1 面数50%
 - LOD2 面数20%
 - LOD3 消失
- 根据到相机距离划分级别

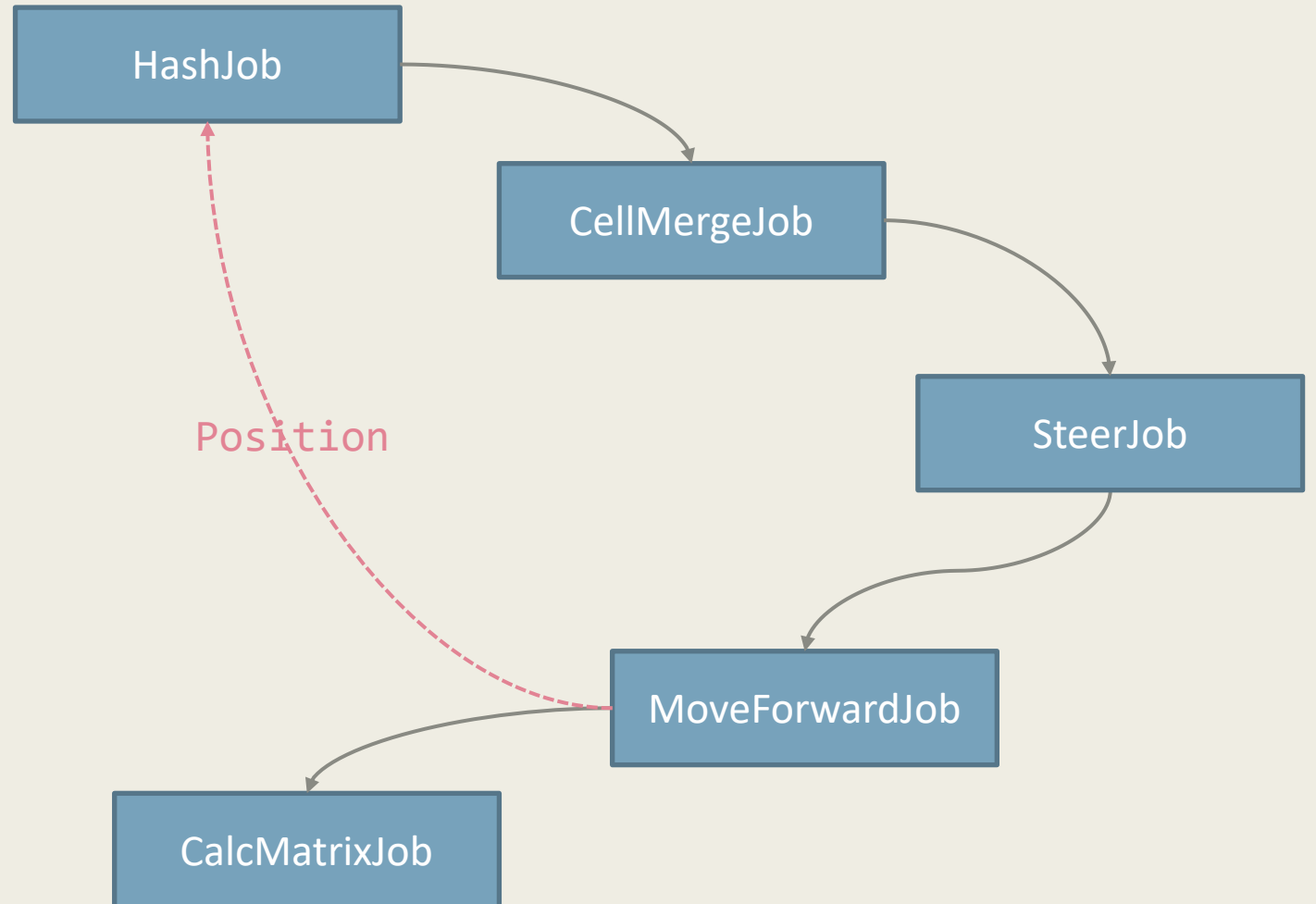


计算优化

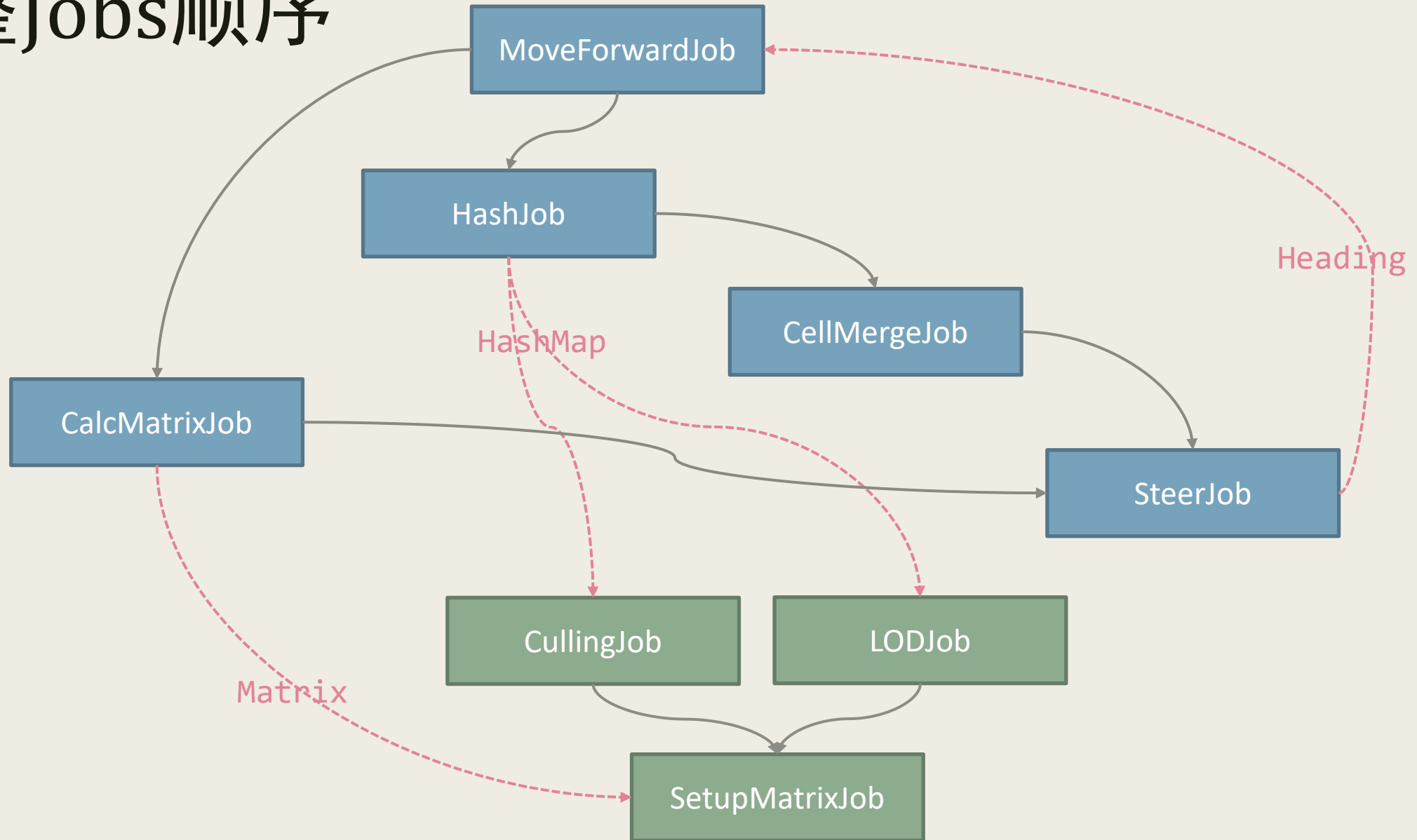
- 以HashMap的Cell为单位做视锥剔除和LOD计算
- 降低60%~80%的计算量



鱼群基本逻辑

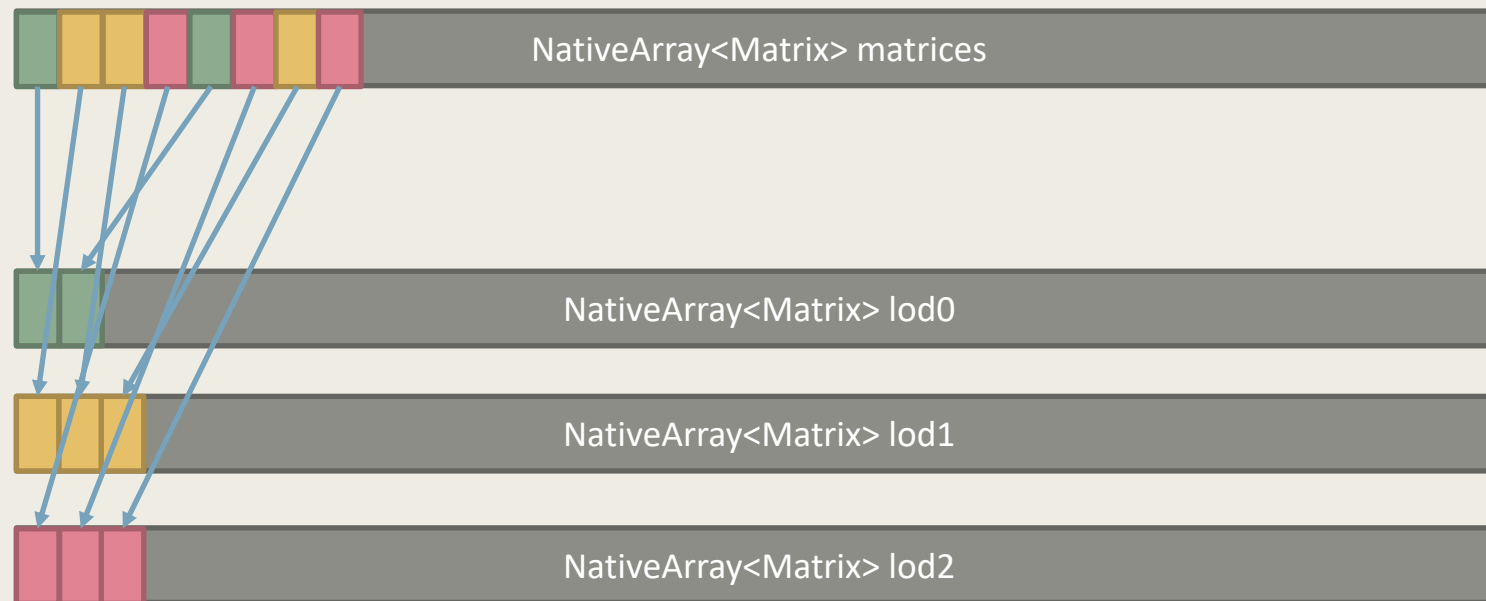


调整Jobs顺序



构建绘制矩阵

- 把鱼群矩阵分拣到不同的LOD数组中供绘制
 - 并行执行
 - 减少内存



多线程原子操作

```
int Interlocked.Increment(ref int location)
```

```
int Interlocked.Add(ref int location, int value)
```

- 原子加

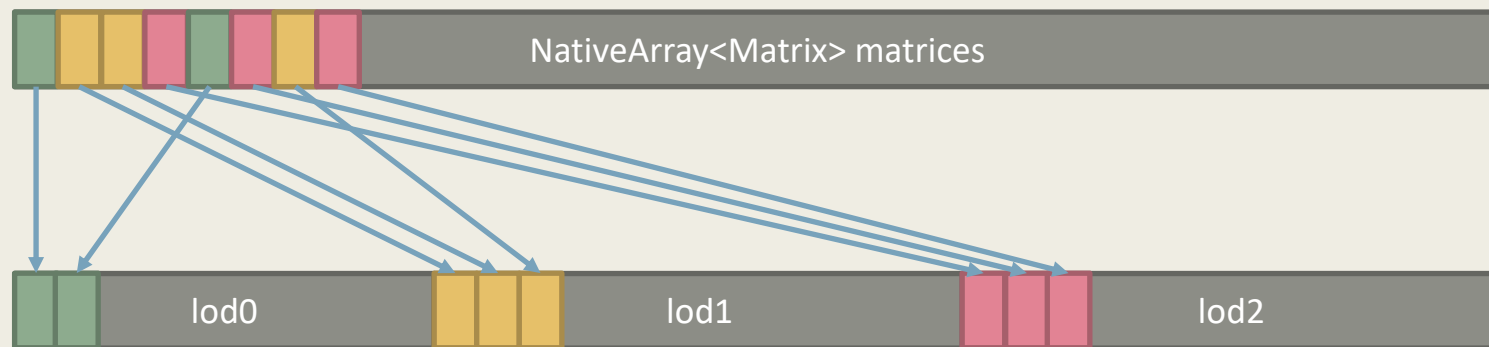
```
int* arrayPtr = (int*)array.GetUnsafePtr();  
Interlocked.Add(ref arrayPtr[index], val);
```

- 原子数组追加

```
int* countPtr = (int*)count.GetUnsafePtr();  
int index = Interlocked.Increment(ref countPtr[0]);  
array[index] = val;
```

构建绘制矩阵

- 在CullingJob和LODJob中使用原子加得到每一级LOD的鱼群数量
- 利用原子数组追加在特定偏移处添加元素



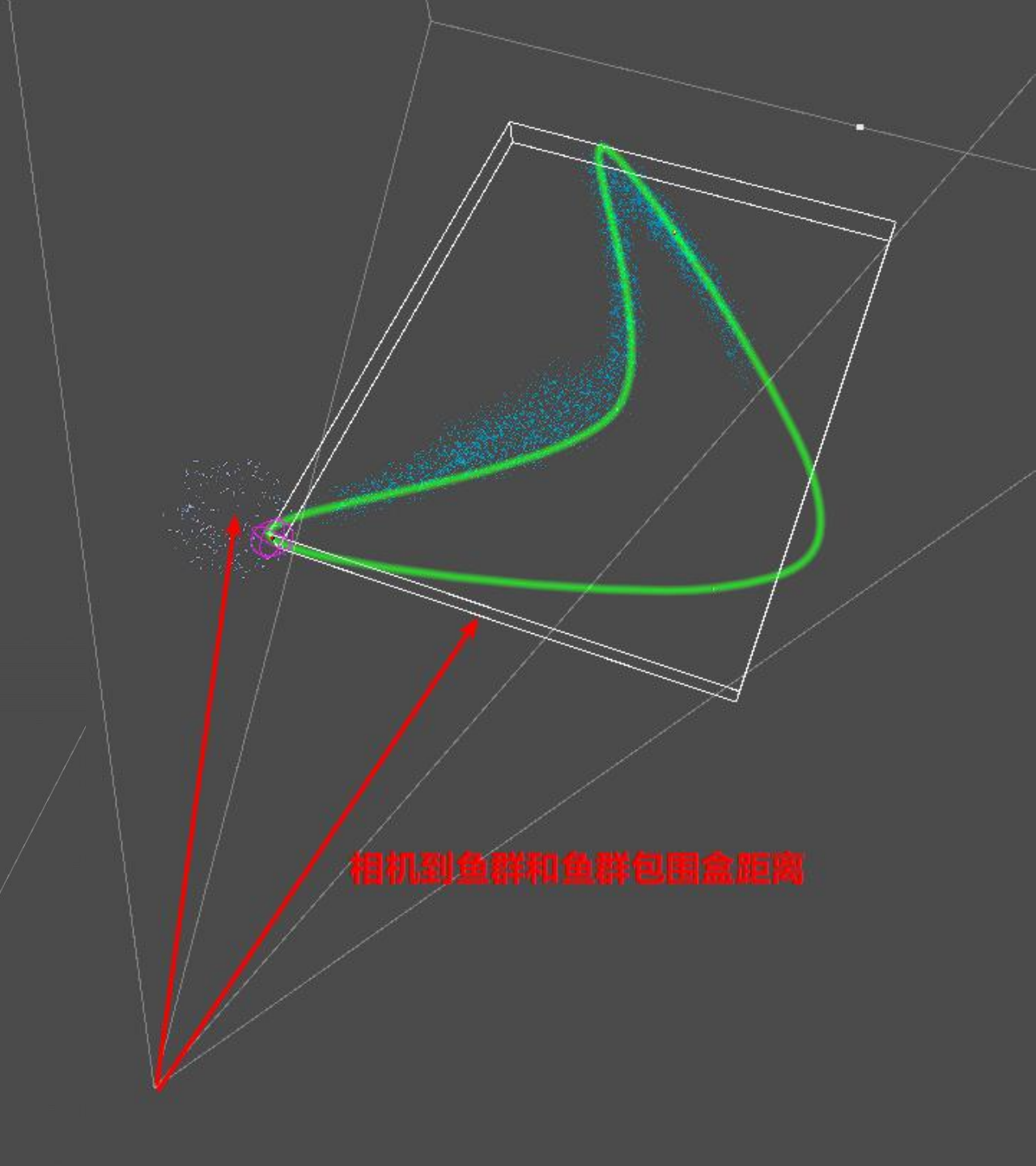
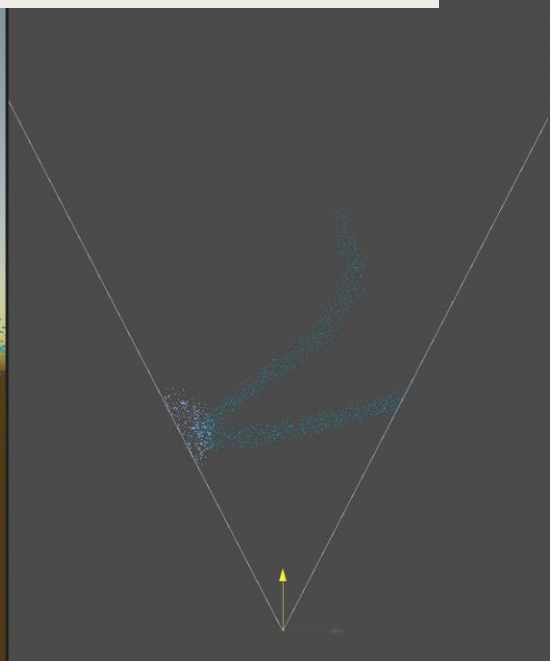
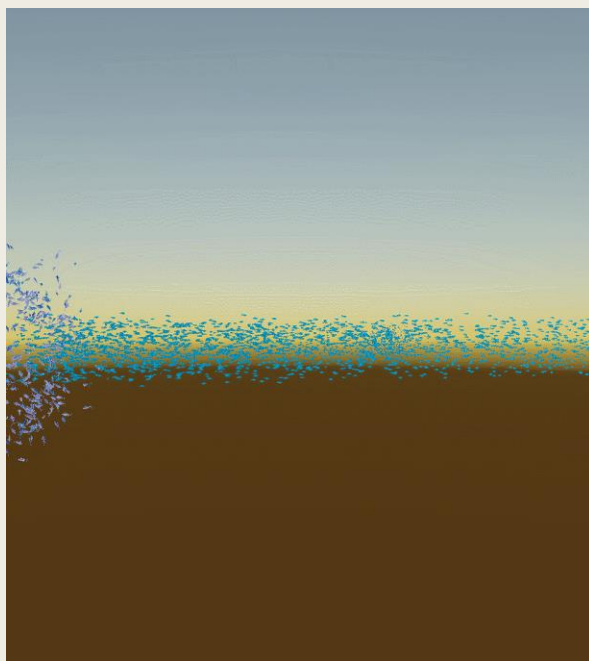
CPU开销

- 一次Job.Schedule()调用主线程耗时 $\approx 0.005\text{ms}$
- 100个鱼群 * 每个鱼群15个Job $\approx 7.5\text{ms}$

逻辑剔除

停止远处鱼群的计算

- 相机到鱼群/鱼群包围盒距离
- 根据曲线逐渐降低鱼群数量



优化结果



鱼群种类: 109
鱼群总数: 7,928
总面数: 321,476

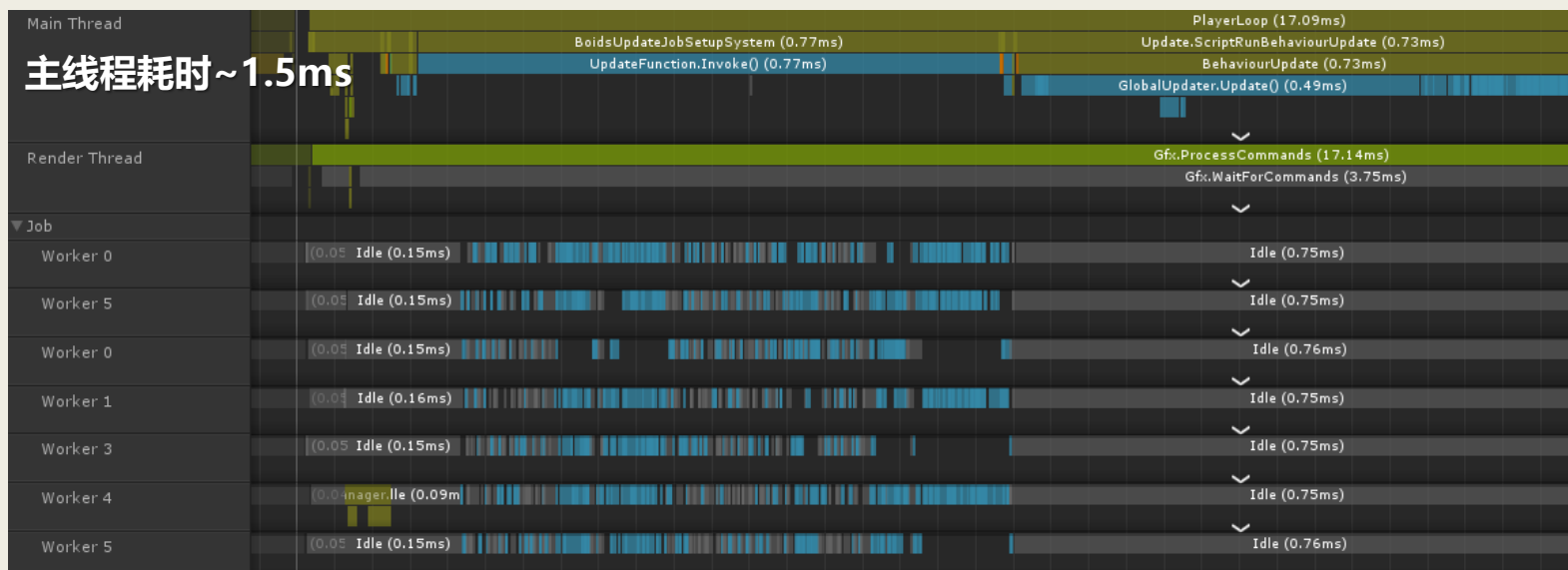
控制开关 鱼群信息 绘制信息 Lod信息											
<input type="checkbox"/> 不运行时LOD											
鱼群	总个数	更新个数	LOD0绘制个数	LOD0模型面数	LOD1消失距离	LOD1绘制个数	LOD1模型面数	LOD2消失距离	LOD2绘制个数	LOD2模型面数	绘制面数
总计	7,928	7,928									321,476
10_xysh_b_fish71	200	200	200	332	20	0	166	50	0	66	66,400
10_xysh_b_fish07	33	33	33	950	20	0	474	50	0	190	31,350
10_xysh_b_fish13	60	60	60	332	20	0	166	50	0	66	19,920
10_xysh_b_fish09	50	50	50	398	20	0	198	50	0	78	19,900

开启优化
更新种类: 21
更新总数: 1,550
渲染面数: 44,248

控制开关 鱼群信息 绘制信息 Lod信息											
<input checked="" type="checkbox"/> 不运行时LOD											
鱼群	总个数	更新个数	LOD0绘制个数	LOD0模型面数	LOD1消失距离	LOD1绘制个数	LOD1模型面数	LOD2消失距离	LOD2绘制个数	LOD2模型面数	绘制面数
总计	7,928	1,550									44,248
10_xysh_b_fish71	200	200	0	332	20	19	166	50	181	66	15,100
10_xysh_b_fish18	23	23	8	496	20	15	248	50	0	98	7,688
10_xysh_b_fish05	24	24	24	298	20	0	148	50	0	58	7,152
10_xysh_b_fish07	33	32	0	950	20	0	474	50	32	190	6,080
10_xysh_b_fish09	50	50	0	398	20	0	198	50	0	78	19,900

继续分析

- 20个鱼群 * 每个鱼群15个Job \approx 1.5ms
- 鱼群种类多，每种鱼群数量少（大部分在100条以下）



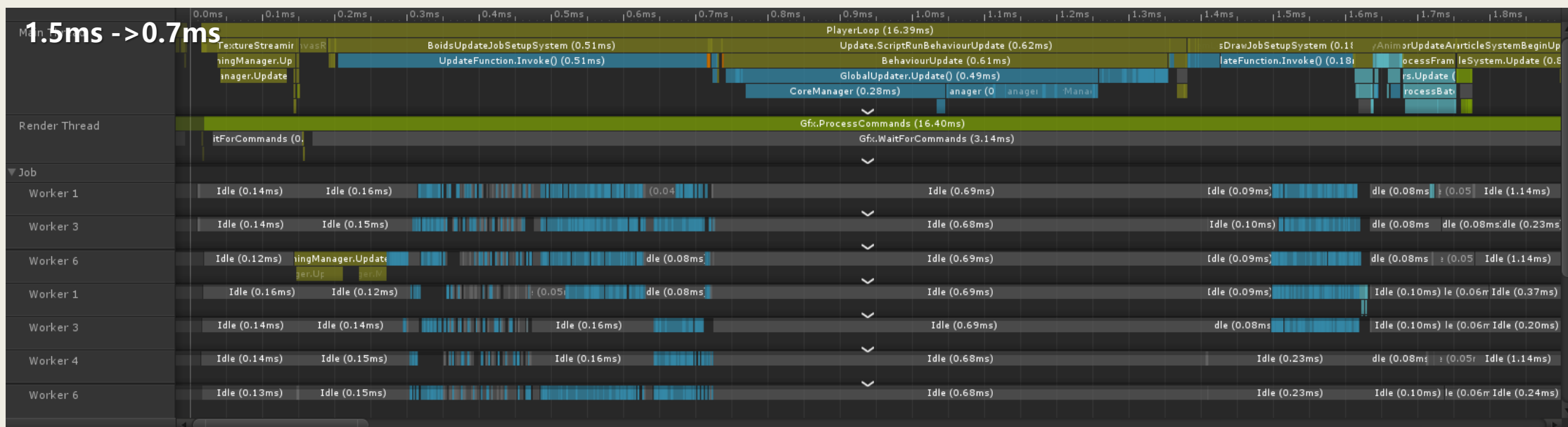
鱼群	总个数
总计	7,928
10_xysh_b_fish95	1200
10_xysh_b_fish75	722
10_xysh_b_fish58	400
10_xysh_b_fish88	393
10_xysh_b_fish86	333
10_xysh_b_fish74	322
10_xysh_b_fish73	322
10_xysh_b_fish94	273
10_xysh_b_fish87	266
10_xysh_b_fish55	220
10_xysh_b_fish42	200
10_xysh_b_fish41	200
10_xysh_b_fish71	200
10_xysh_b_fish40	170
10_xysh_b_fish44	160
10_xysh_b_fish43	160
10_xysh_b_fish20	120
10_xysh_b_fish51	80
10_xysh_b_fish13	60
10_xysh_b_fish60	60
10_xysh_b_fish47	60
10_xysh_b_fish52	60
10_xysh_b_fish37	57
10_xysh_b_fish08	56
10_xysh_b_fish46	55

精简Job

- 合并小Job
 - 合并TargetsJob和MoveForwardJob
 - 合并CullingJob和LODJob
- 功能Job开关
 - 关闭远处鱼群的交互计算Job
- 根据数量调整算法
 - 数量 <32 ，使用 $O(n^2)$ 算法，省去hash相关的Job
 - 数量 $==1$ ，主线程计算

优化结果

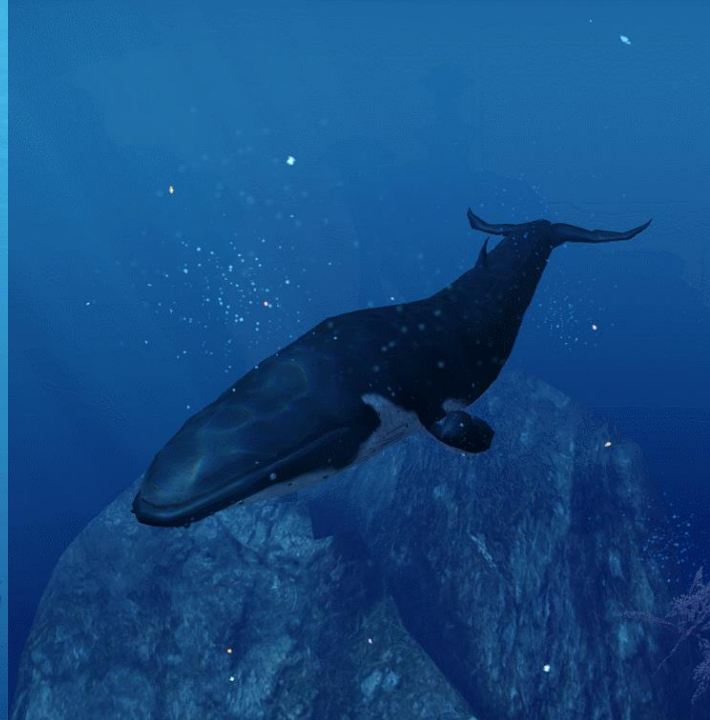
- 主线程1.5ms -> 0.7ms



鱼群动画

■ 顶点动画

- 纵尾，横向摆动（鱼类）
- 横尾，纵向摆动（哺乳类）
- 收缩（水母）
- 通过顶点色控制摆动幅度
- 分配随机值使动画时间偏移



鱼群动画

■ Gpu蒙皮动画

- 将idle动画骨骼信息烘焙到贴图
- 一个顶点只受一根骨骼影响
- 顶点位置+法线
- 支持Instancing绘制



鱼群光照

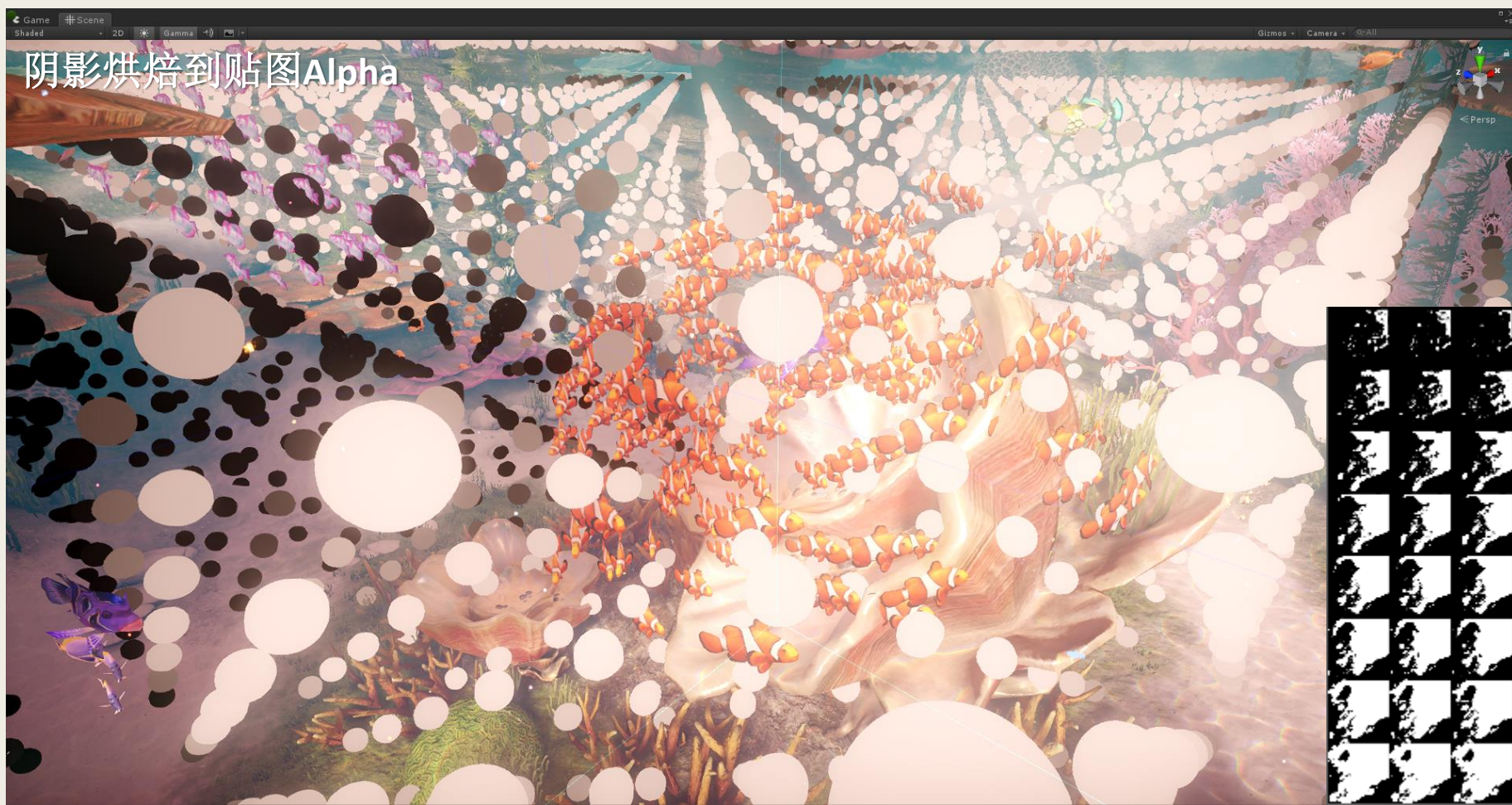
浅海中鱼群暗部太暗，深海鱼群过亮，鱼群不能融入场景



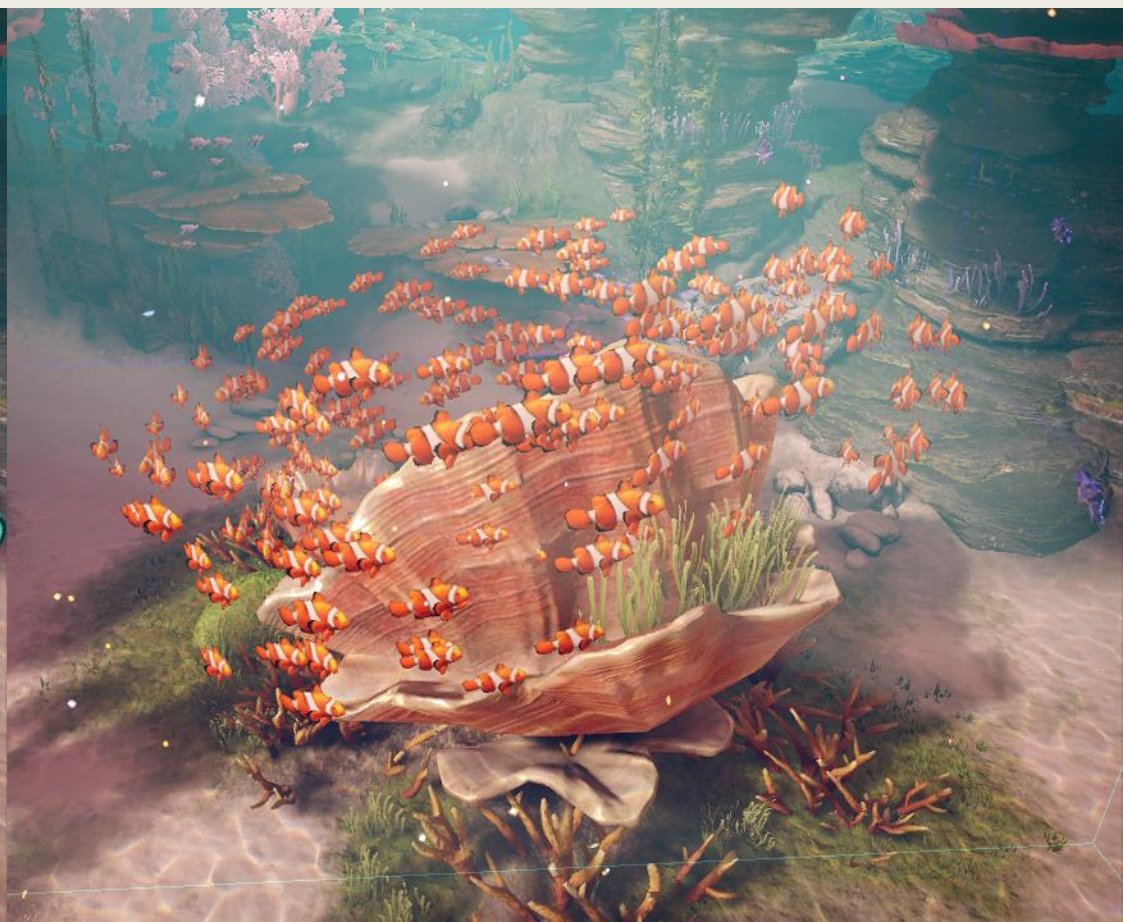
鱼群光照



鱼群光照

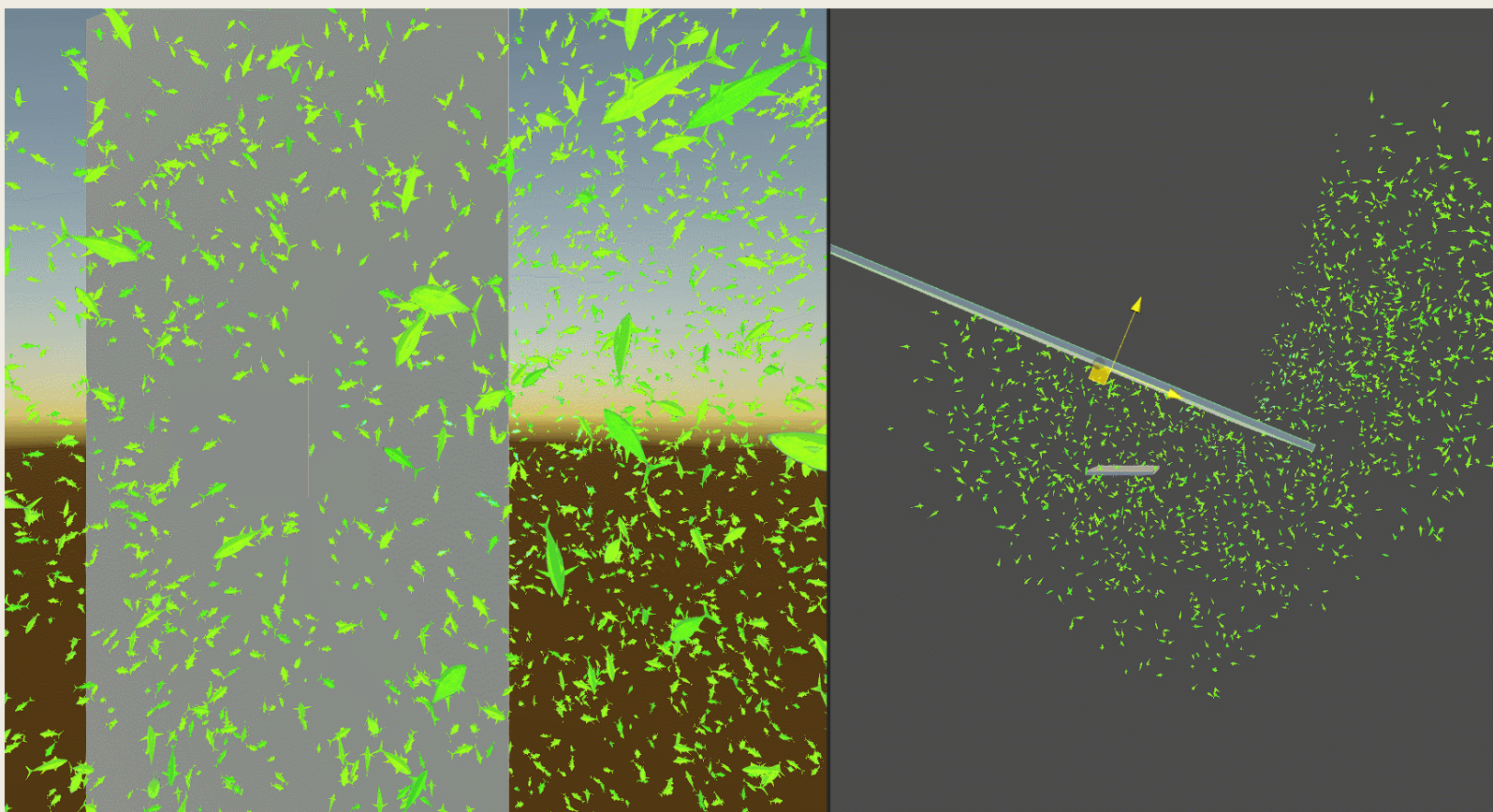


鱼群光照



遮挡剔除

利用Hi-Z Occlusion在GPU中做遮挡剔除



遮挡剔除

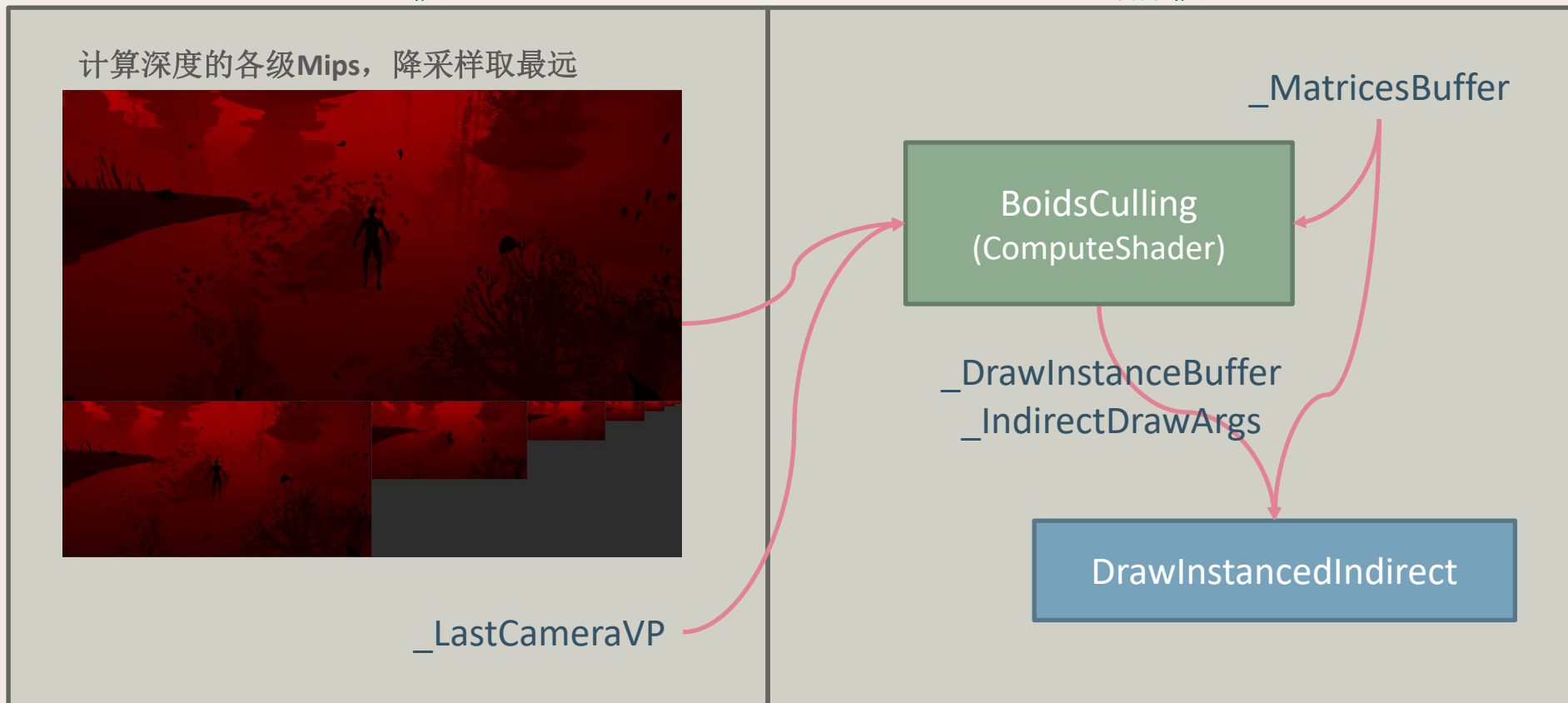
- 根据深度判断物体有没有被遮挡
- 借助深度Mips提高效率



遮挡剔除

上一帧

当前帧



遮挡剔除

- 有一帧延迟
- Mali的GLES设备不支持在VertexShader中使用ComputeBuffer
- 生成DepthMips开销较高

1000条鱼遮挡剔除	iPhoneXR	Oppo A5
DrawDepthMips	0.5ms	1.5ms
BoidsCulling	0.09ms	0.2ms

THANKS

