Web3D技术探索之路

刘茜

奇安信科技集团

InfoQ

精彩继续！
更多一线大厂前沿技术案例

北京站

PCon
全球产品创新大会

时间：2021年8月20-21日
地点：北京 · 国际会议中心

扫码查看大会详情>>

深圳站

ArchSummit
全球架构师峰会

时间：2021年9月3-4日
地点：深圳 · 大中华喜来登酒店

扫码查看大会详情>>

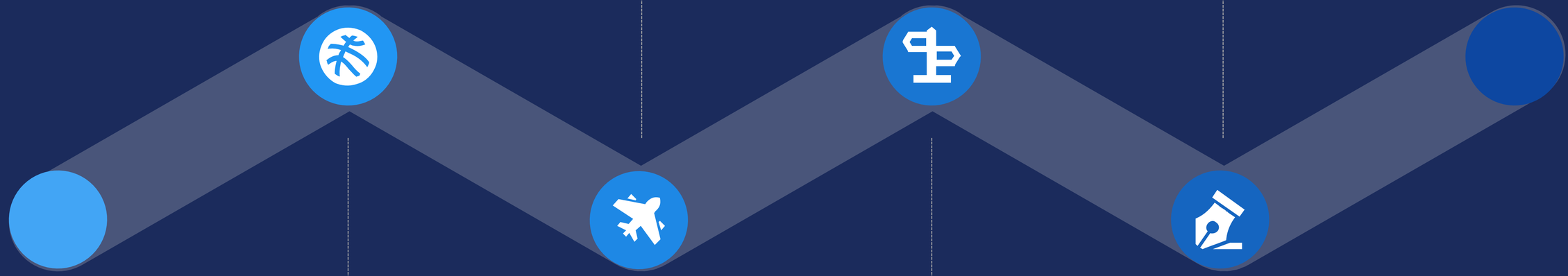
北京站

AiCon
全球人工智能与机器学习技术大会

时间：2021年9月17-18日
地点：北京· 国际会议中心

扫码查看大会详情>>

个人简介



大屏主视觉开发
2017-2018
QAX

Web3D开发
2020-今
QAX

流体、三维重构
2013-2016
USTB

3D地图引擎开发
2019-2020
QAX

雷尔可视化平台部
奇安信 VIS

简单，炫酷
用简单易用的接入方式，让客户获得炫酷的体验

技术领域

Web

图表　拓扑　流程图　图可视化分析
地图　可视化大屏编辑器

Design

UI规范　界面设计　用户体验　视觉创意
设计模式

U3D

地图　信息图表　MR/VR　特效动画
3D建模

学术互动

ChinaVis挑战赛2015–2020

主办ChinaVis可视化会议中的挑战赛环节，负责比赛题目的设置，数据的构造，评分以及颁奖

ChinaVis可视化会议中发表演讲

2015年 多维可视化分析以及协同分析
2016年 用可视化分析打击伪基站
2017年 WebGL和GIS在网络安全可视化中的应用
2018年 产学研论坛汇报及讨论
2019年 炫酷可视化在安全场景的应用
2020年 安全业务里的多端可视化技术探索与实践

IEEE VAST挑战赛

2015年 获得提名奖项“Honorable Mention for Good Support for Flexible and Collaborative Analysis”

2016年 获得最高级别的奖项“Award for Outstanding Comprehensive Solution”

2017年 获得“Multi–Challenge Award for Combining Automated and Visual Analytics”

2018年 获得"Award: Insights Generated Through the Use of a Custom Tool"

2019年 获得"Honorable Mention for Effective Summarization and Communication of Results"

大纲

- 业务背景
- 场景拆解和搭建
- 技术架构
- 优化方案
- 未来规划和展望

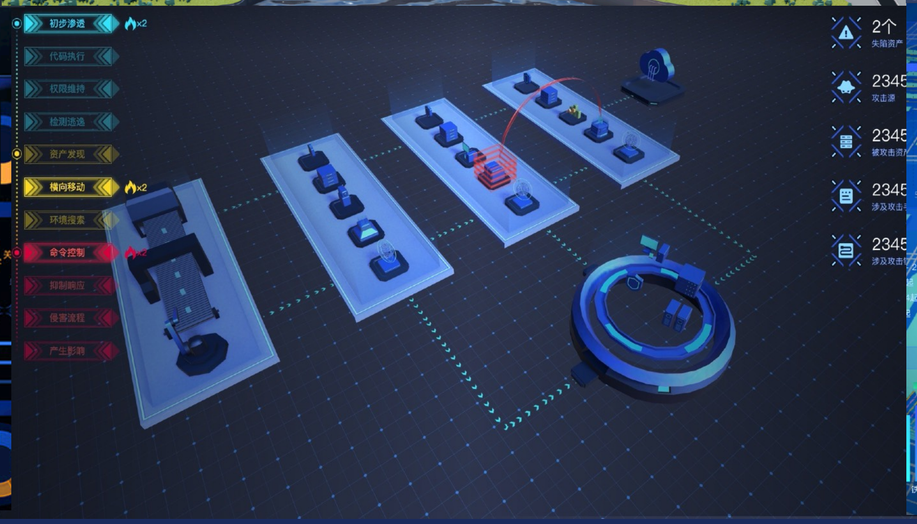
大纲

- 业务背景
- 场景拆解和搭建
- 技术架构
- 优化方案
- 未来规划和展望

大屏主视觉业务


风险信息
通报记录
合肥市政府
XXXXXX单位
XXXXXX单


黑帽SEOer
基础设施
攻击能力
目标信息


初步渗透
代码执行
权限维持
检测逃逸
资产发现
横向移动
环境搜索
命令控制
抑制响应
侵害流程
产生影响


应急指挥态势
最近30天
行业
区域
合肥市政务局
单位1
单位3
合肥市中级法院
单位2
合肥市政府

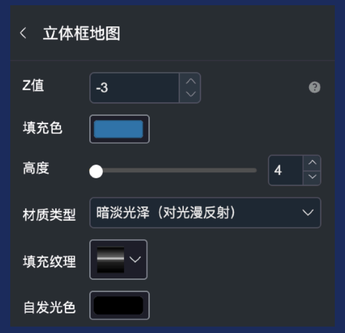
2021年之前

业务场景需求

3D开发能力

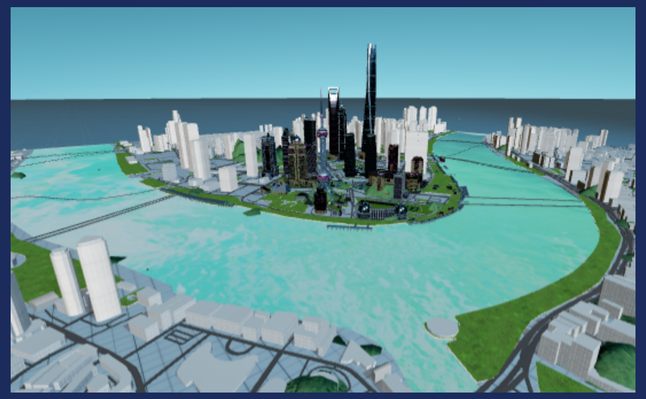
平面设计能力

3D贴图工程师

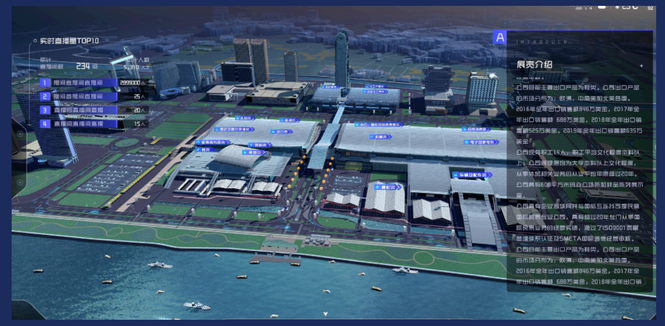


竞品技术方案对比

DataV



RayData



51World

2021年之后

业务场景需求

3D开发能力

3D建模能力

3D搭建工程师

奇安信3D演化道路

2019-2020

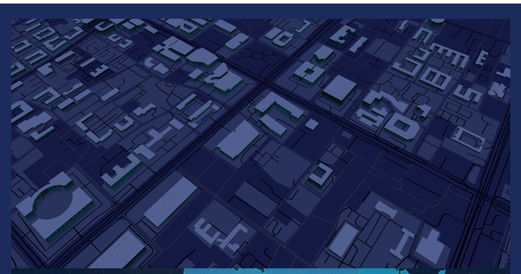
2020-2021

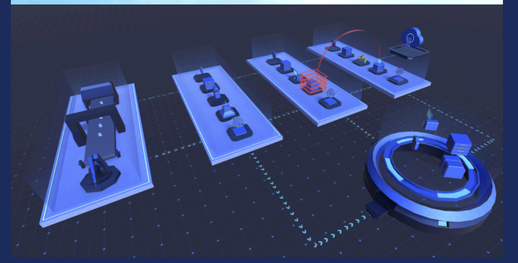
City

District

Earth

Topo



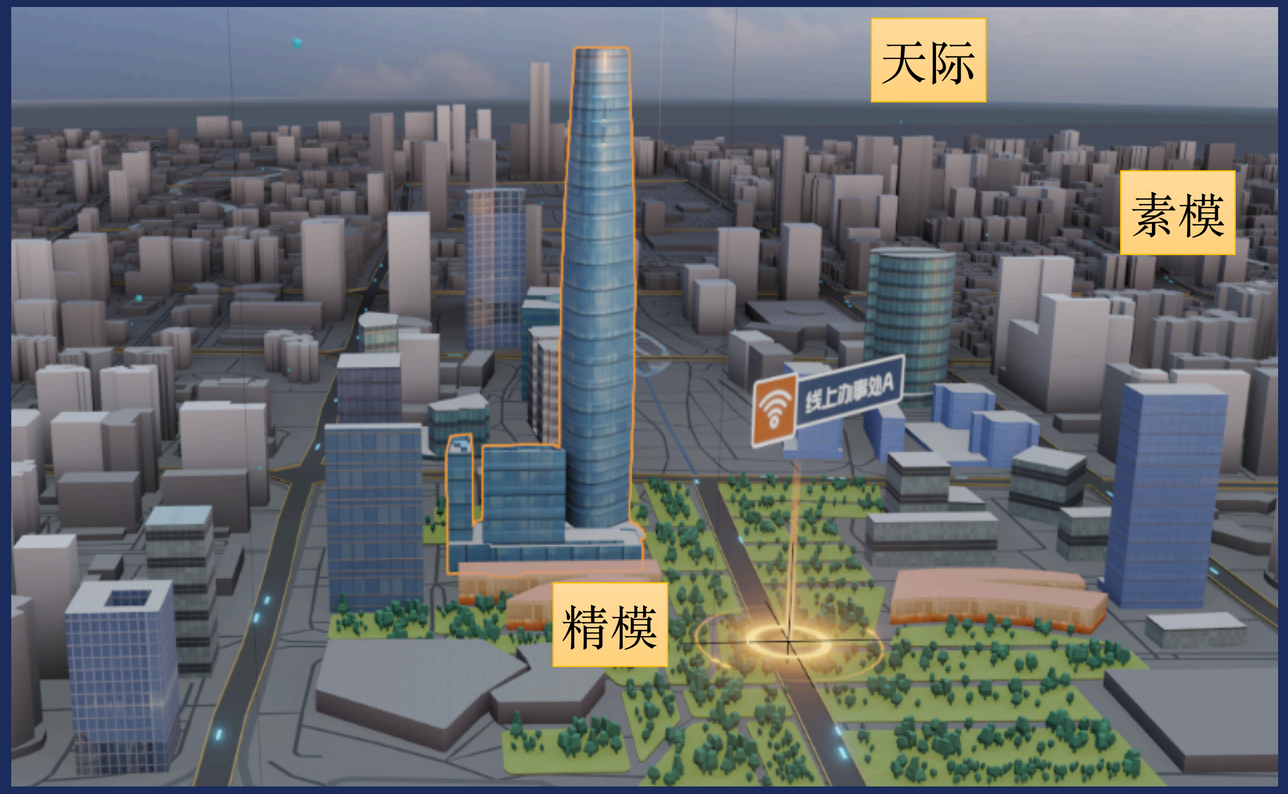


大纲

- 业务背景
- **场景拆解和搭建**
- 技术架构
- 优化方案
- 未来规划和展望

City场景拆解

水平近远　　精模　　素模　　天际线



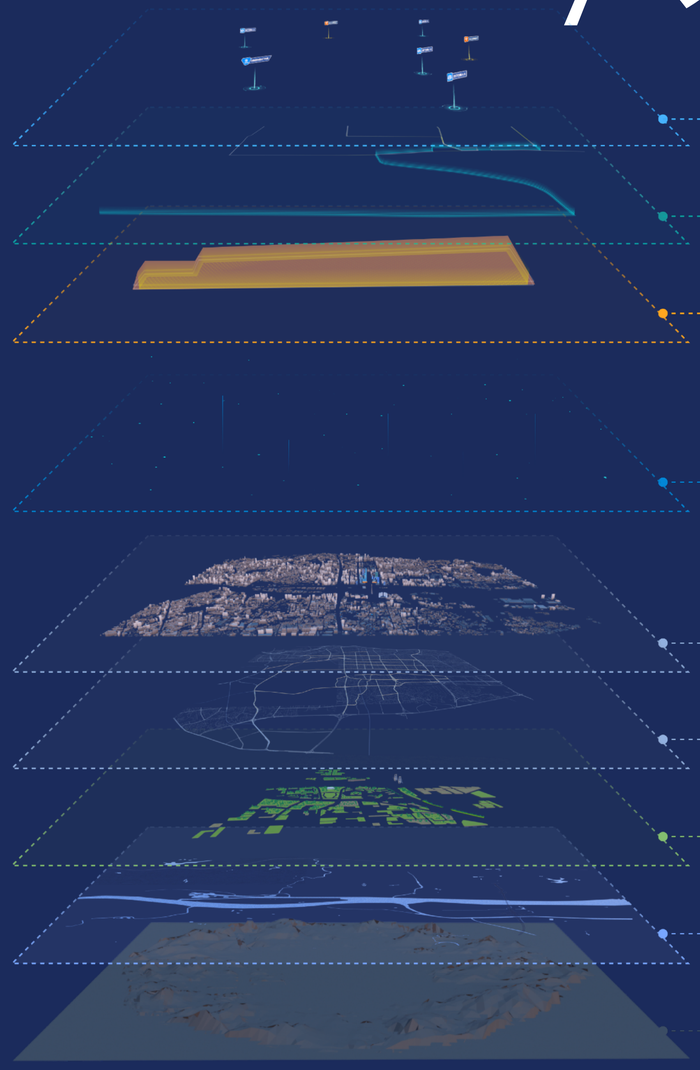
City场景拆解

垂直高度

数据

氛围

城市基础



点

线

面

数据形态

氛围

动效

建筑

道路

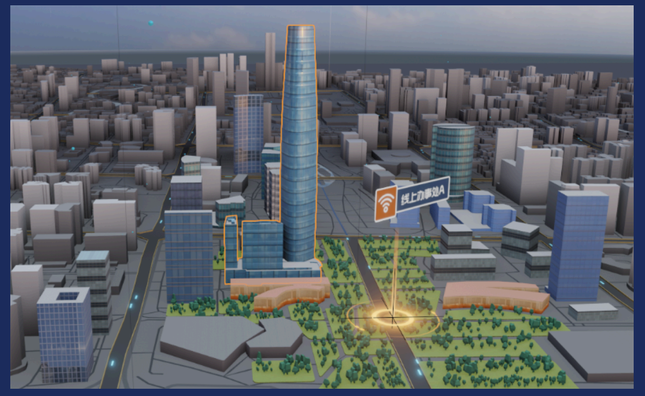
植被

水

地形

城市基础

三维数字行政区视觉规范



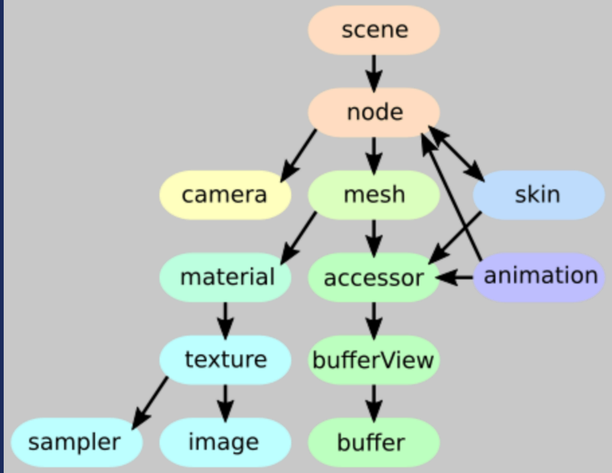
模型选择

- fbx
  - FilmBox（MotionBuilder），3D 角色动画软件，Autodesk 家族
  - 专注于动作制作，在modeling和rendering上有赖于其他软件的配合
- obj、mtl
  - AliaslWavefront（Alias），加拿大图像业巨头
  - 两种格式需要配合使用，很受欢迎
- collada
  - Sony，基于 XML
  - 为娱乐而生、Google Earth
- glTF
  - 工业界（Microsoft、Cesium、Unity etc.）共同提出、维护
  - 高效、可扩展

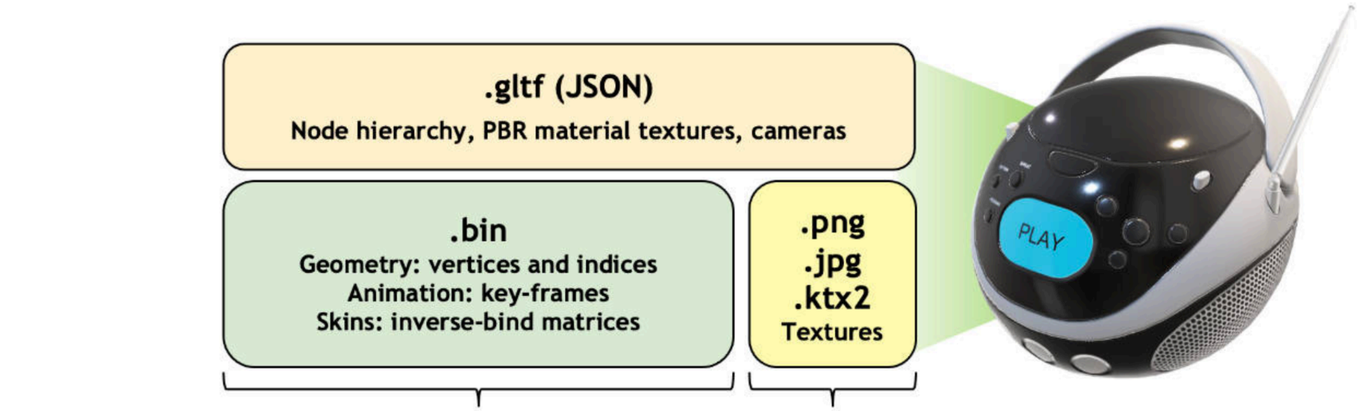
glTF2.0

Concepts

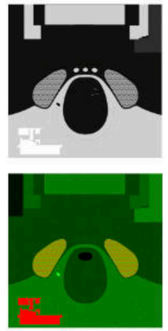
The conceptual relationships between the top-level elements of a glTF asset are shown here:



glTF 2.0 Scene Description Structure



Geometry



Texture based
PBR materials

PBR stands for
"Physically-Based Rendering"

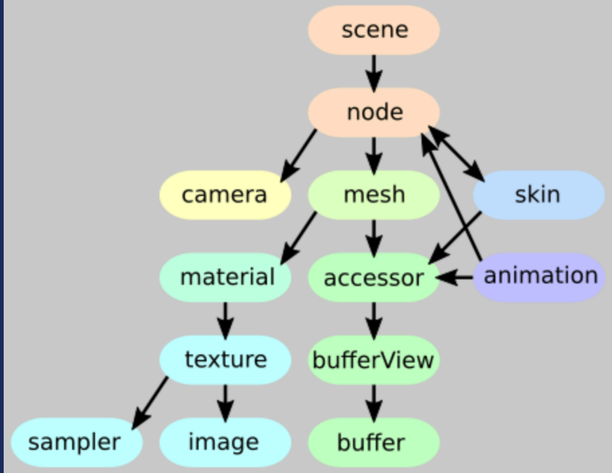
Mandatory Metallic-Roughness Materials
Base Color (Albedo) | Metalness | Roughness
Emission | Normal Map | Baked Ambient Occlusion

Optional Specular-Glossiness Materials
Diffuse | Specular | Glossiness

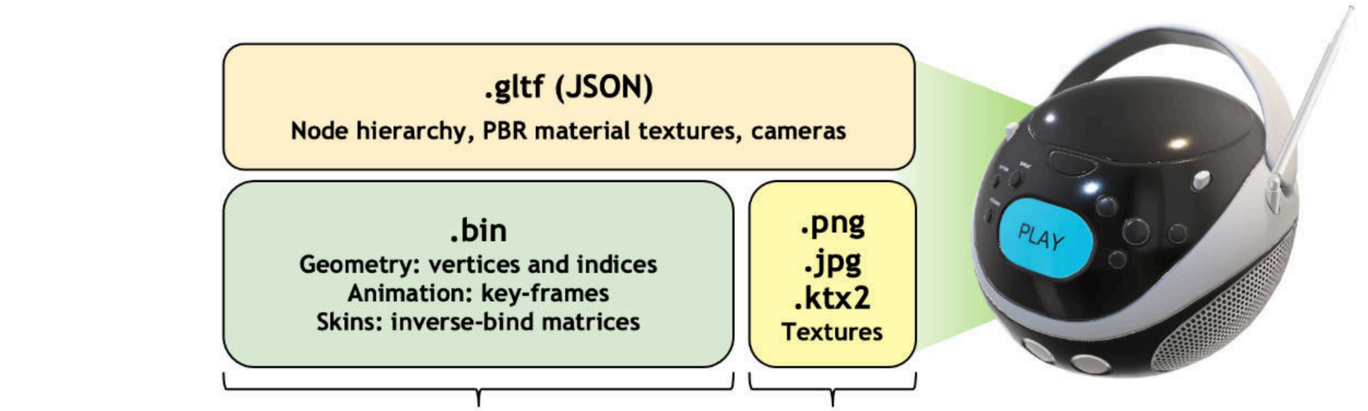
glTF2.0

Concepts

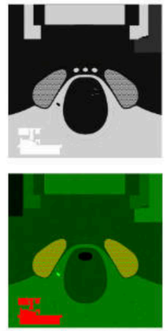
The conceptual relationships between the top-level elements of a glTF asset are shown here:



glTF 2.0 Scene Description Structure



Geometry



Texture based
PBR materials

PBR stands for
"Physically-Based Rendering"

Mandatory Metallic-Roughness Materials
Base Color (Albedo) | Metalness | Roughness
Emission | Normal Map | Baked Ambient Occlusion

Optional Specular-Glossiness Materials
Diffuse | Specular | Glossiness

建模工具选择

Blender优势

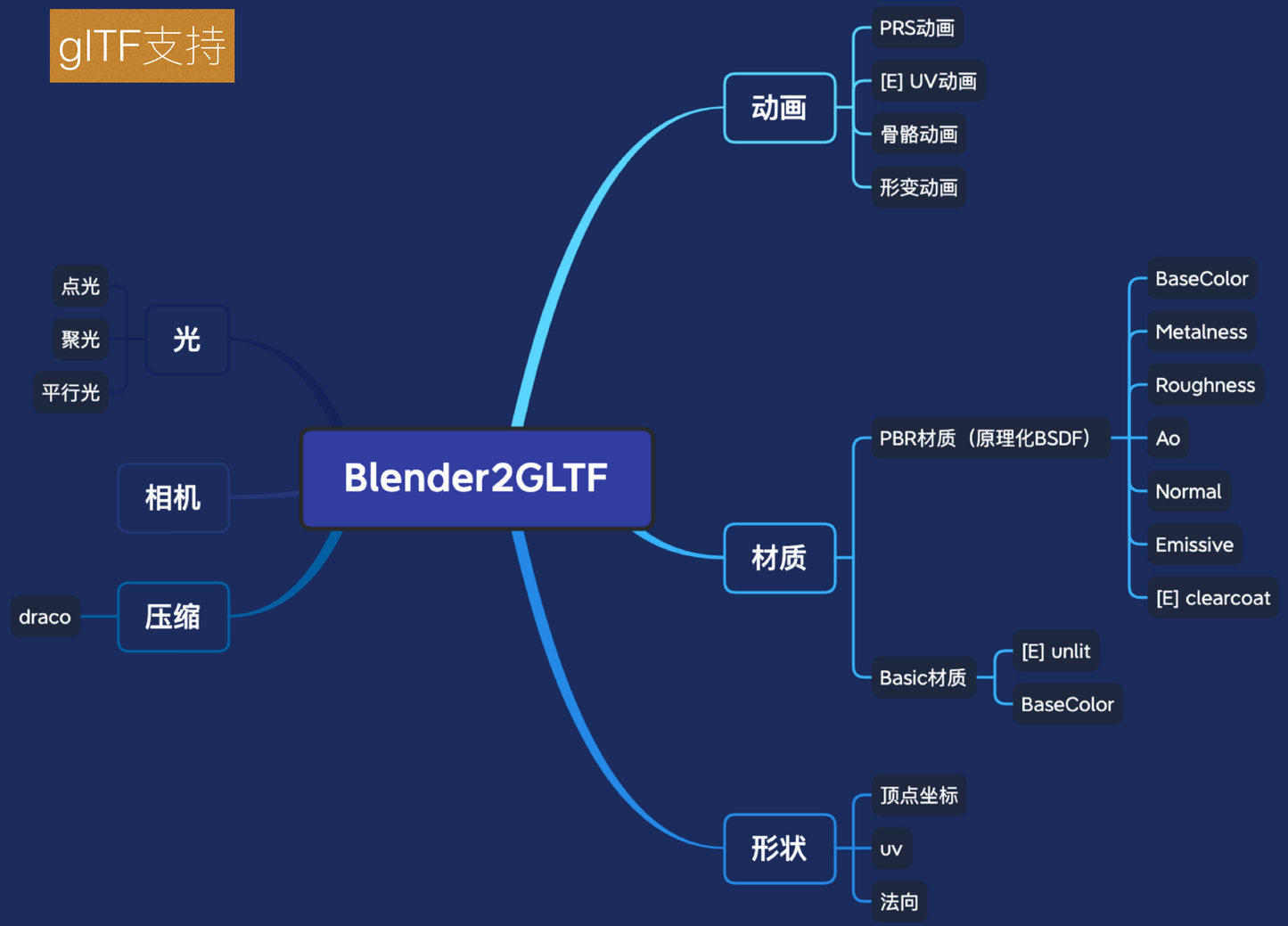
- 实时渲染
- 开源免费，扩展性强，插件丰富
- 全平台（Windows、MacOS、Linux）
- 基于工作流的交互方式
- 操作效率高

渲染底层选择

编辑器选择

设计–研发约定

gITF支持

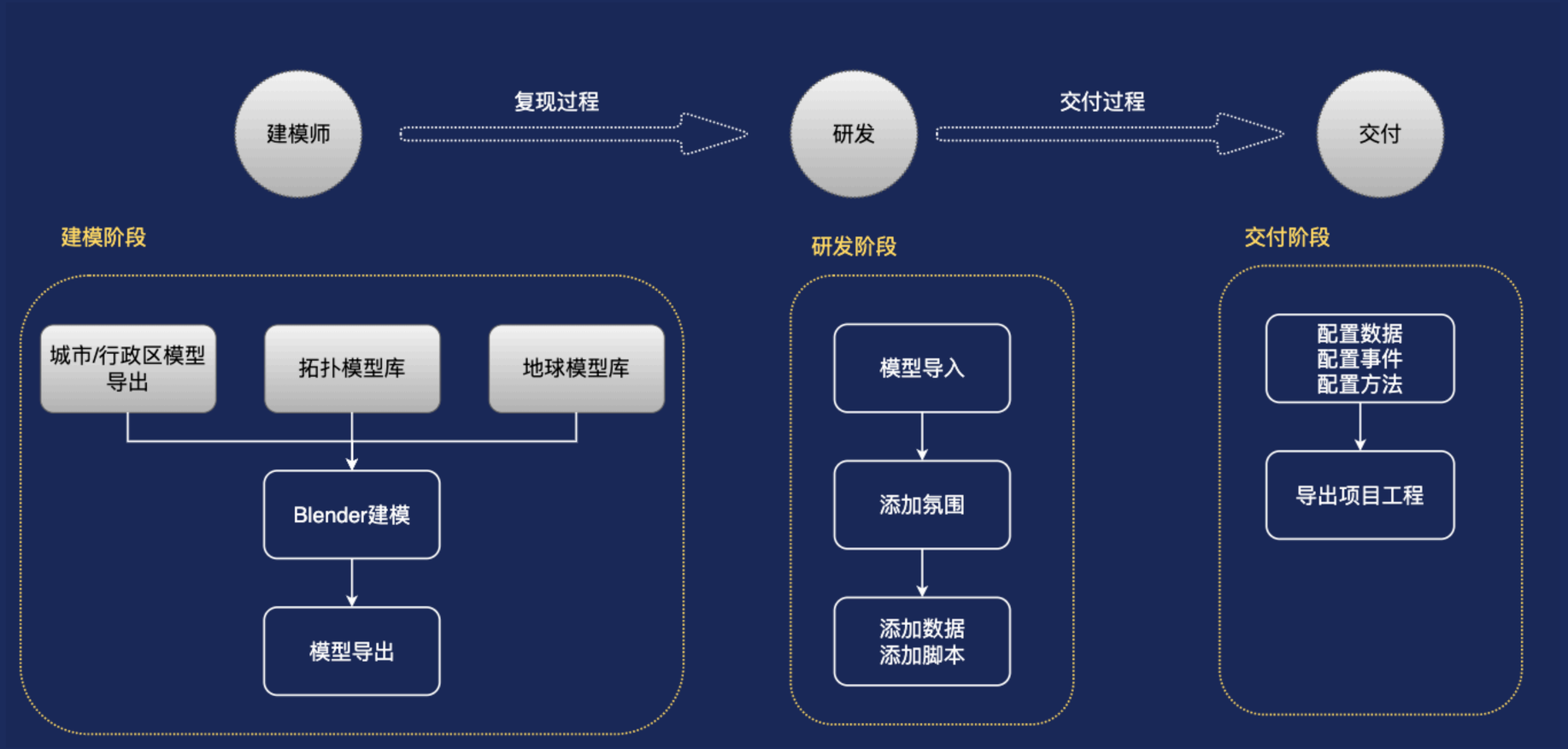


gITF不支持

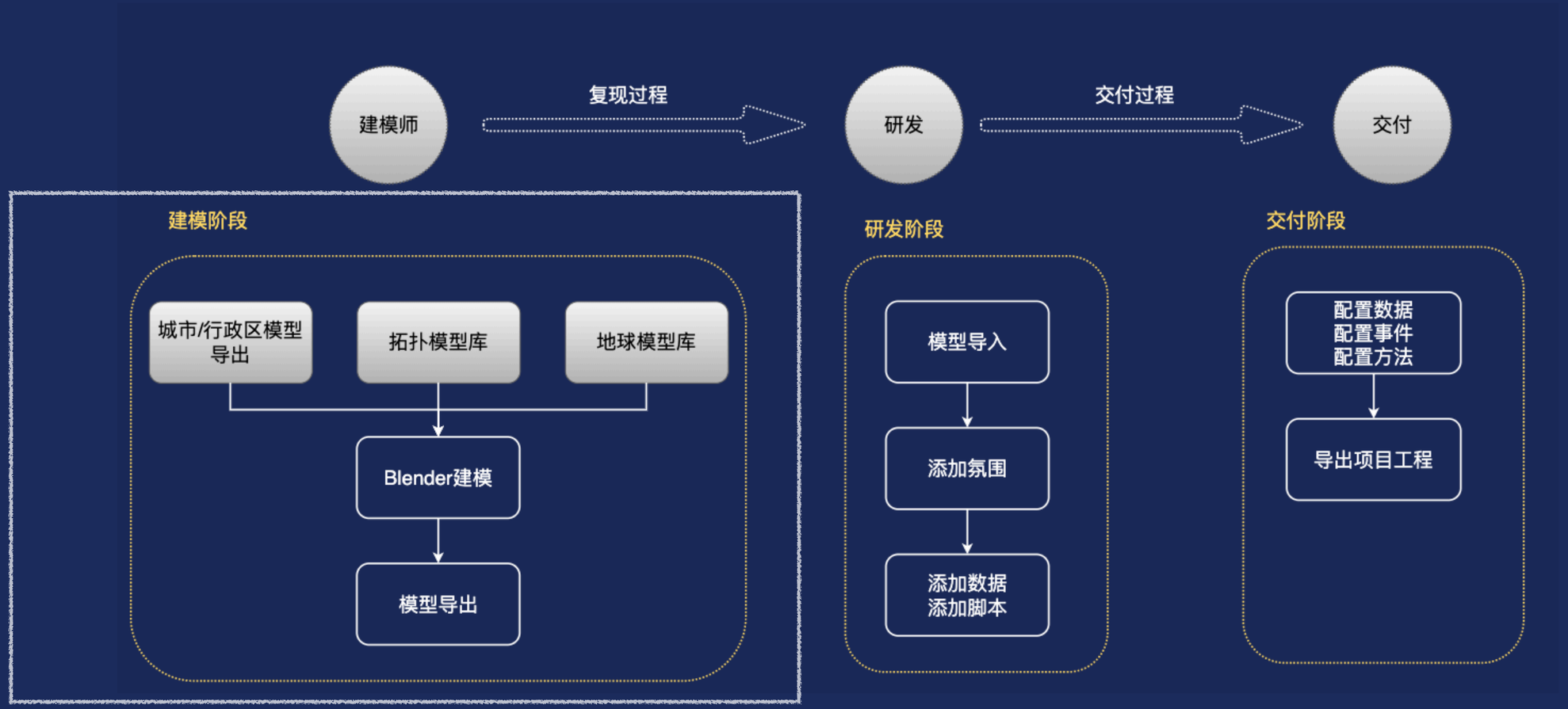
- Bloom
- SSAO
- HDR
- 粒子动画
- 涟漪动画
- 菲涅尔
- 脚本逻辑
- 。。。

基于业务的编辑器

基于glTF2.0设计研发工作流



建模阶段


建模师
复现过程
研发
交付过程
交付
建模阶段
城市/行政区模型导出
拓扑模型库
地球模型库
Blender建模
模型导出
研发阶段
模型导入
添加氛围
添加数据
添加脚本
交付阶段
配置数据
配置事件
配置方法
导出项目工程

模型数据来源

Blender GIS 插件

坐标映射

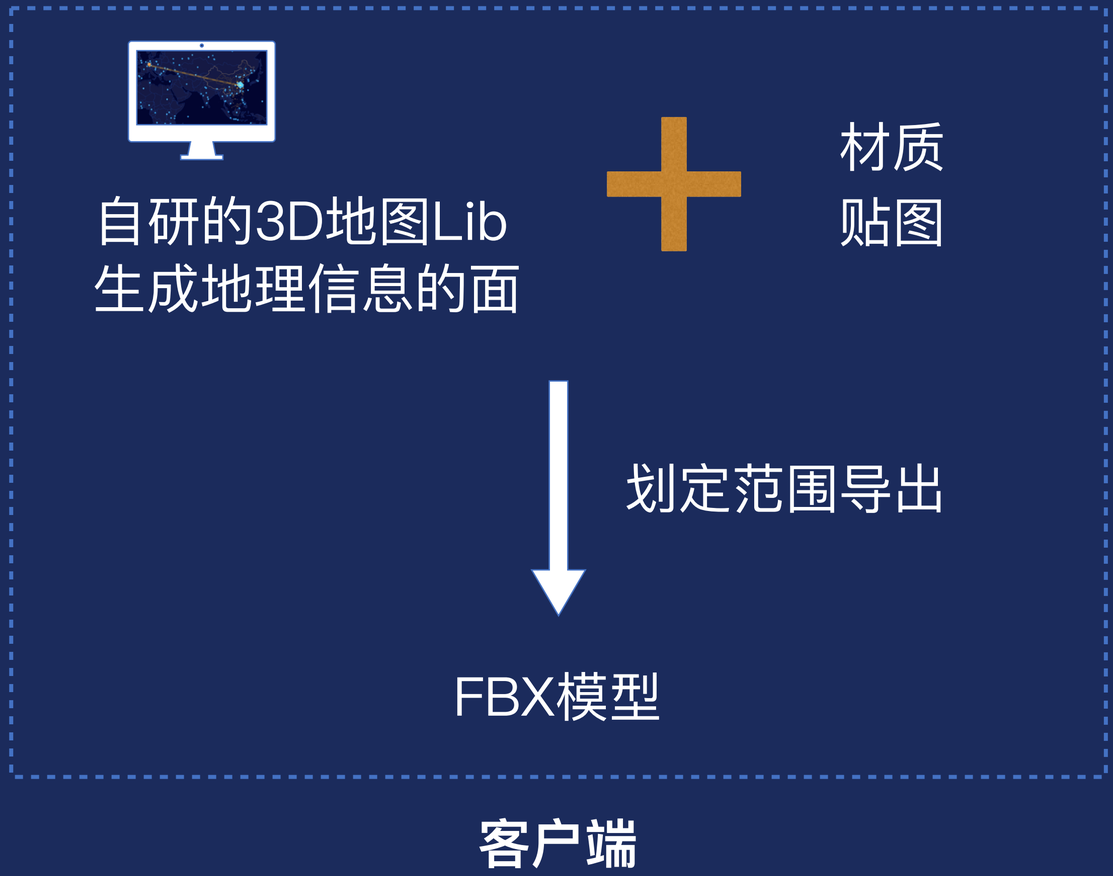
行政区划
GeoJSON

开放地理数据
ShapeFile

地图数据模型导出
FBX

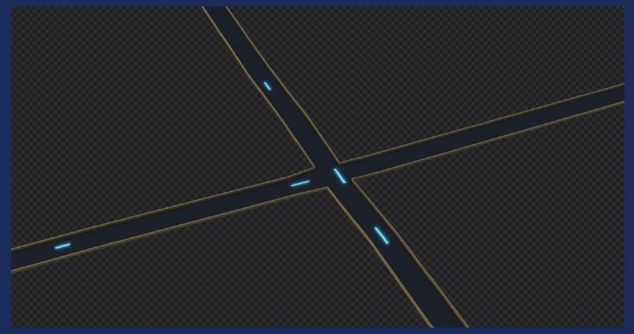
城市模型导出

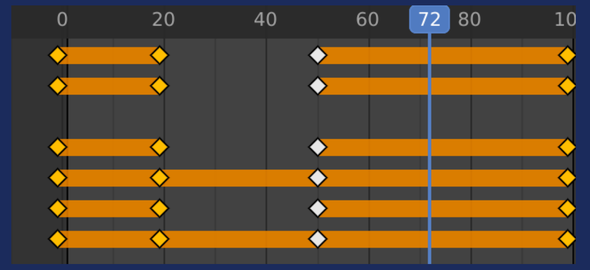
binary



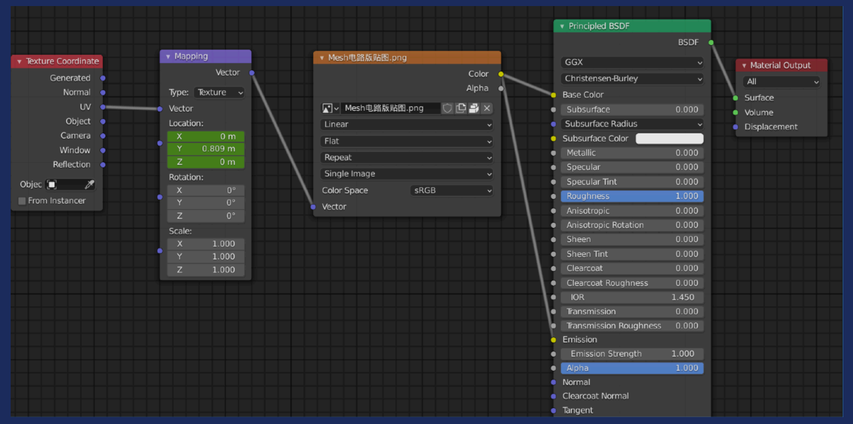
Blender插件

插件实现UV动画导出

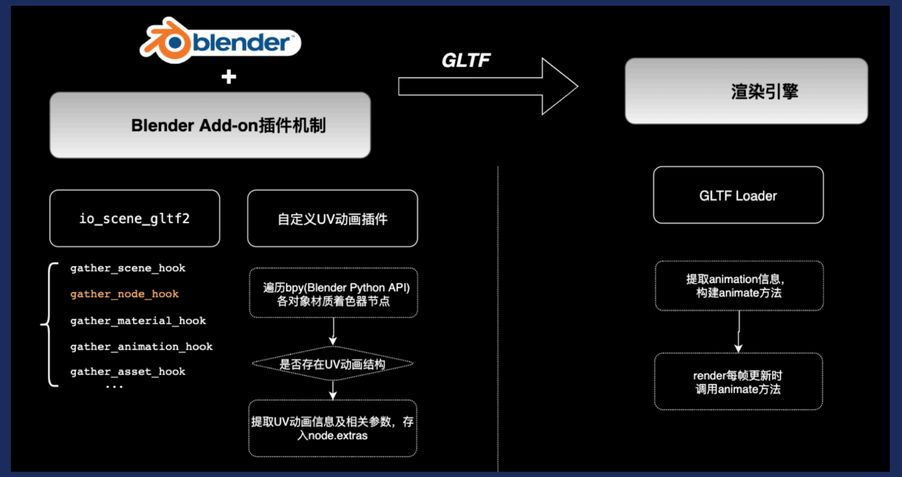




Mapping的k帧

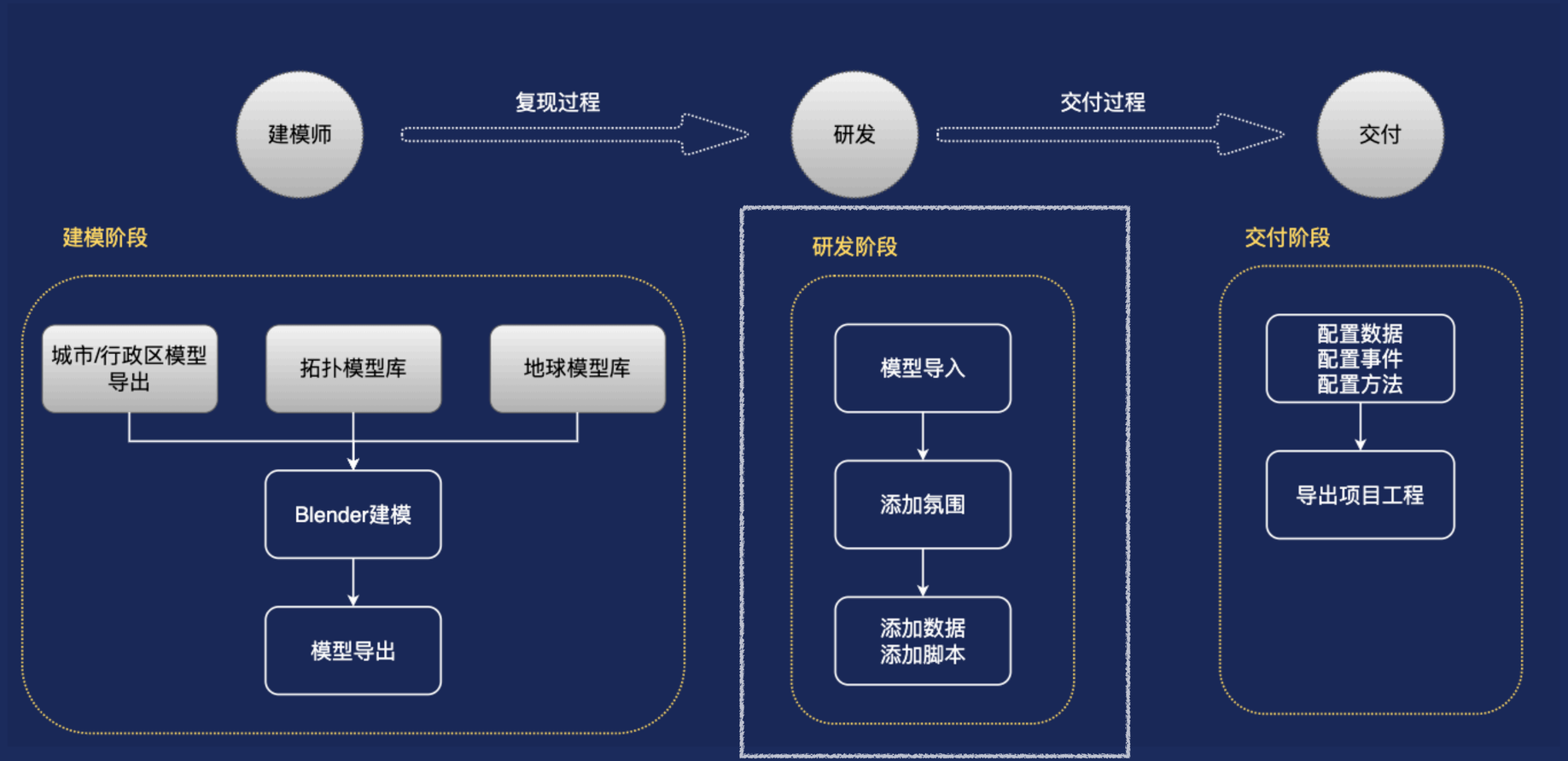


Blender材质编辑器



Blender插件导出机制及引擎解析

研发阶段


建模师
复现过程
研发
交付过程
交付
建模阶段
城市/行政区模型导出
拓扑模型库
地球模型库
Blender建模
模型导出
研发阶段
模型导入
添加氛围
添加数据
添加脚本
交付阶段
配置数据
配置事件
配置方法
导出项目工程

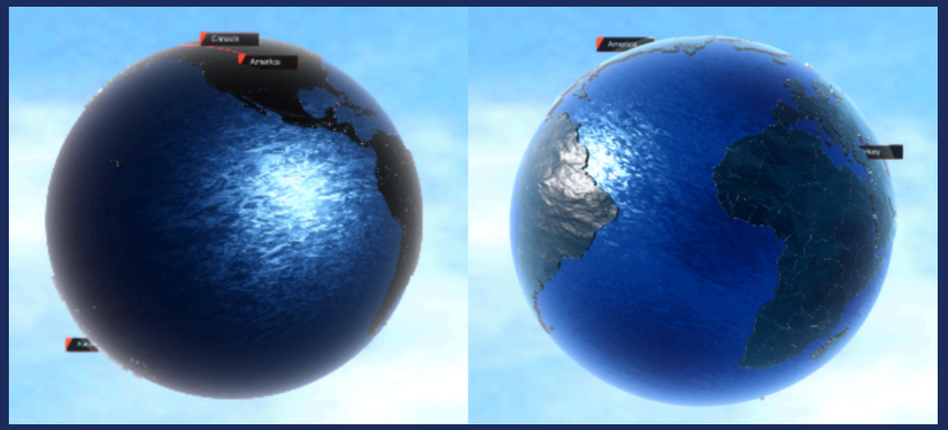
研发需要实现的能力

模型加载	代码生成 业务模型	光照	氛围
后期	数据	动画	脚本

IBL+HDR

等量矩形贴图

立方体贴图

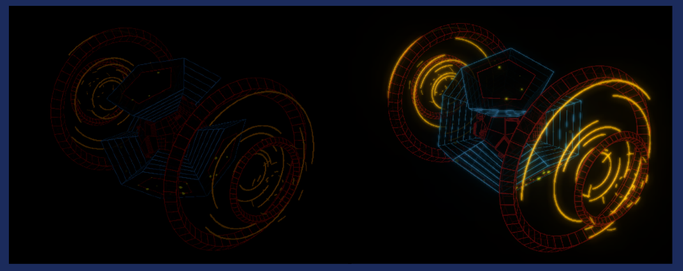


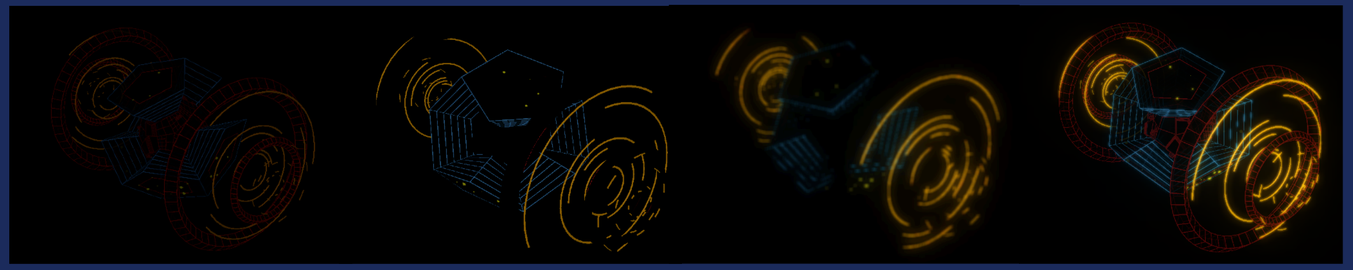
基于业务的编辑器

模型加载	代码生成 业务模型	光照	氛围
后期	数据	动画	脚本

后期

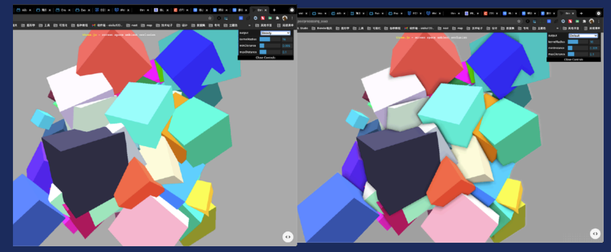
Bloom

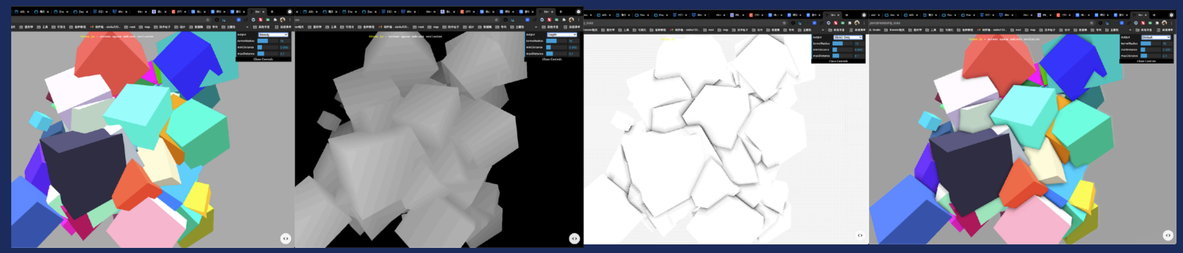




原图　亮度提取　高斯模糊　叠加原图

SSAO





原图　深度缓冲　AO　叠加原图

研发需要实现的能力

模型加载	代码生成 业务模型	光照	氛围
后期	数据	动画	脚本

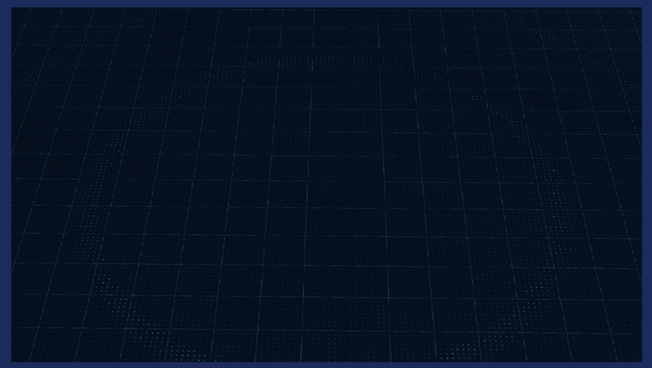
氛围

粒子动画

匀变速运动

$$x = v * t + 1/2 * a * t^2$$

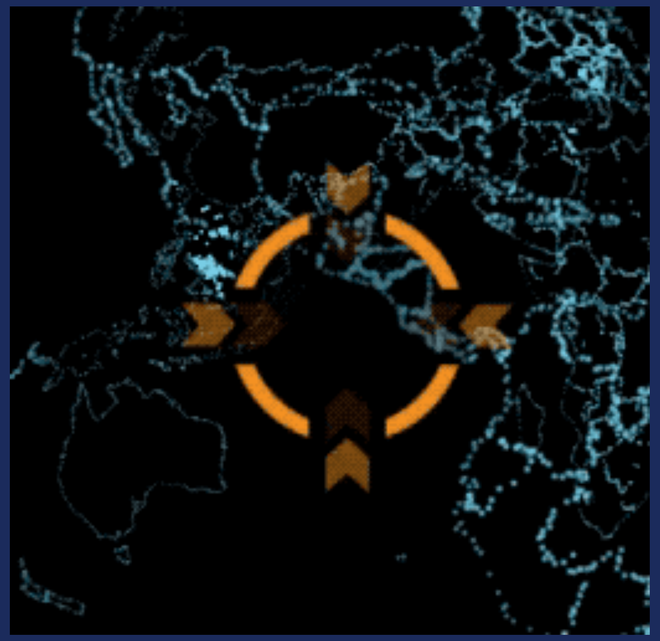
涟漪/扫光

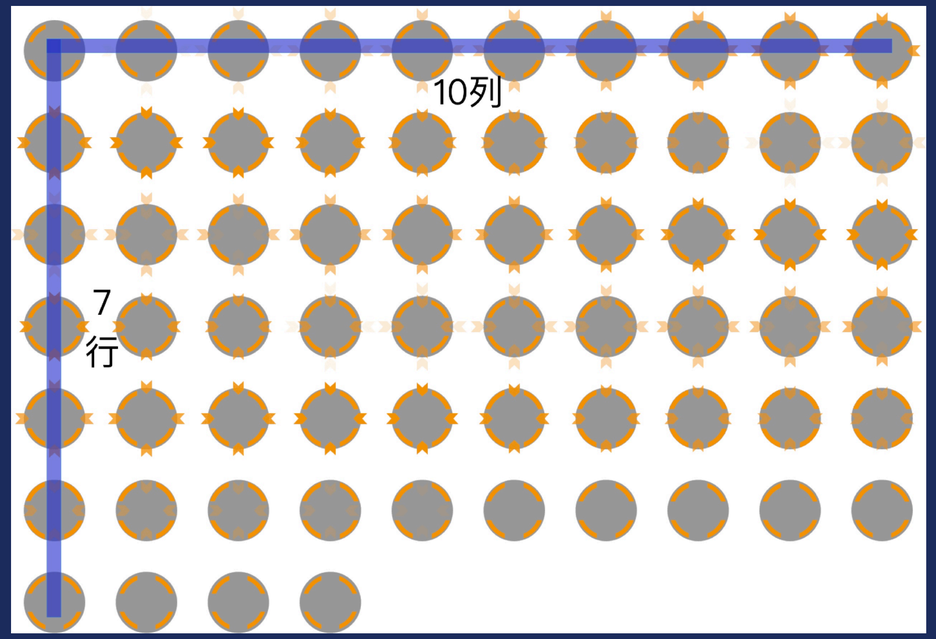


研发需要实现的能力

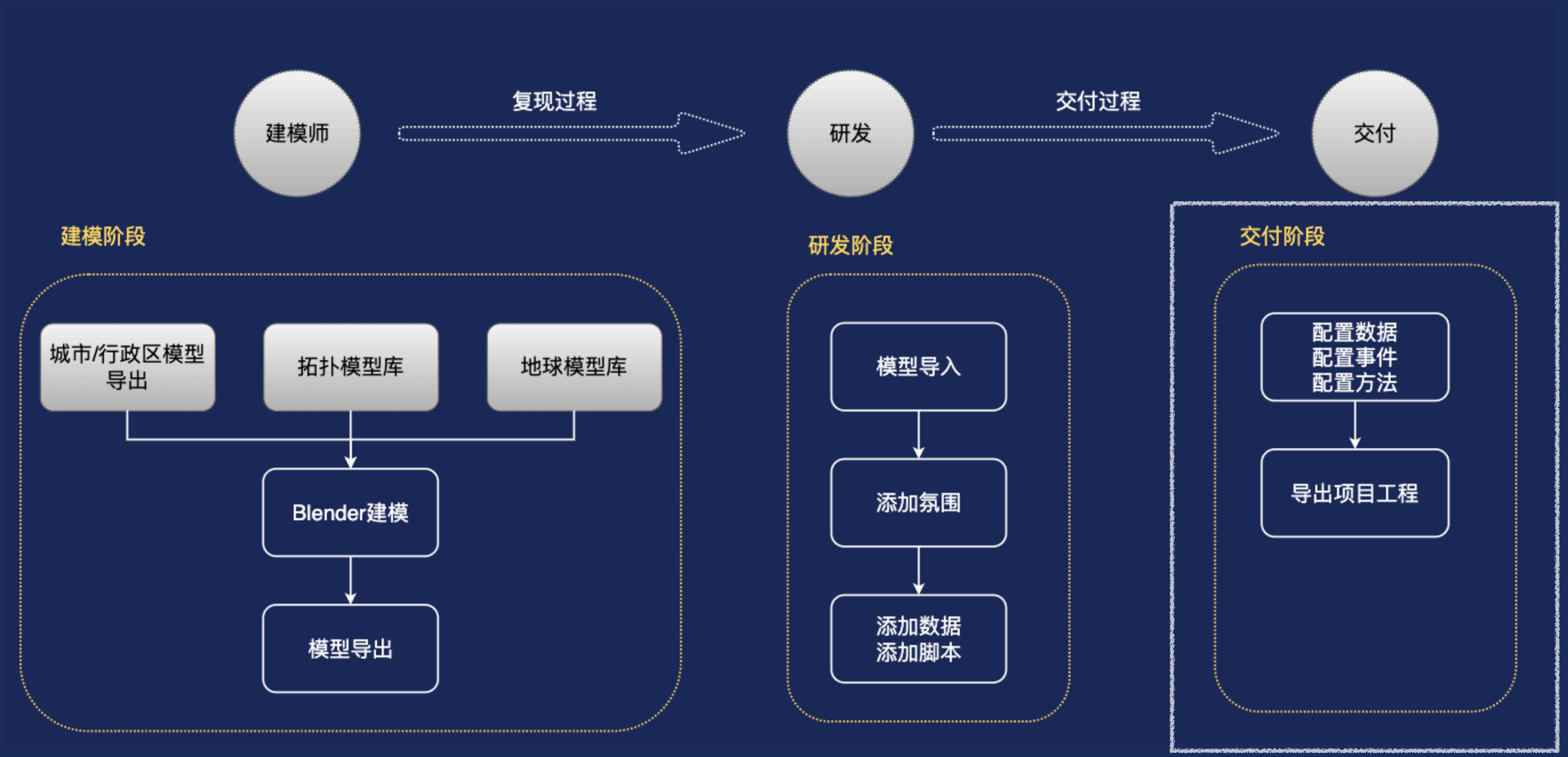
模型加载	代码生成 业务模型	光照	氛围
后期	数据	动画	脚本

帧动画





交付阶段

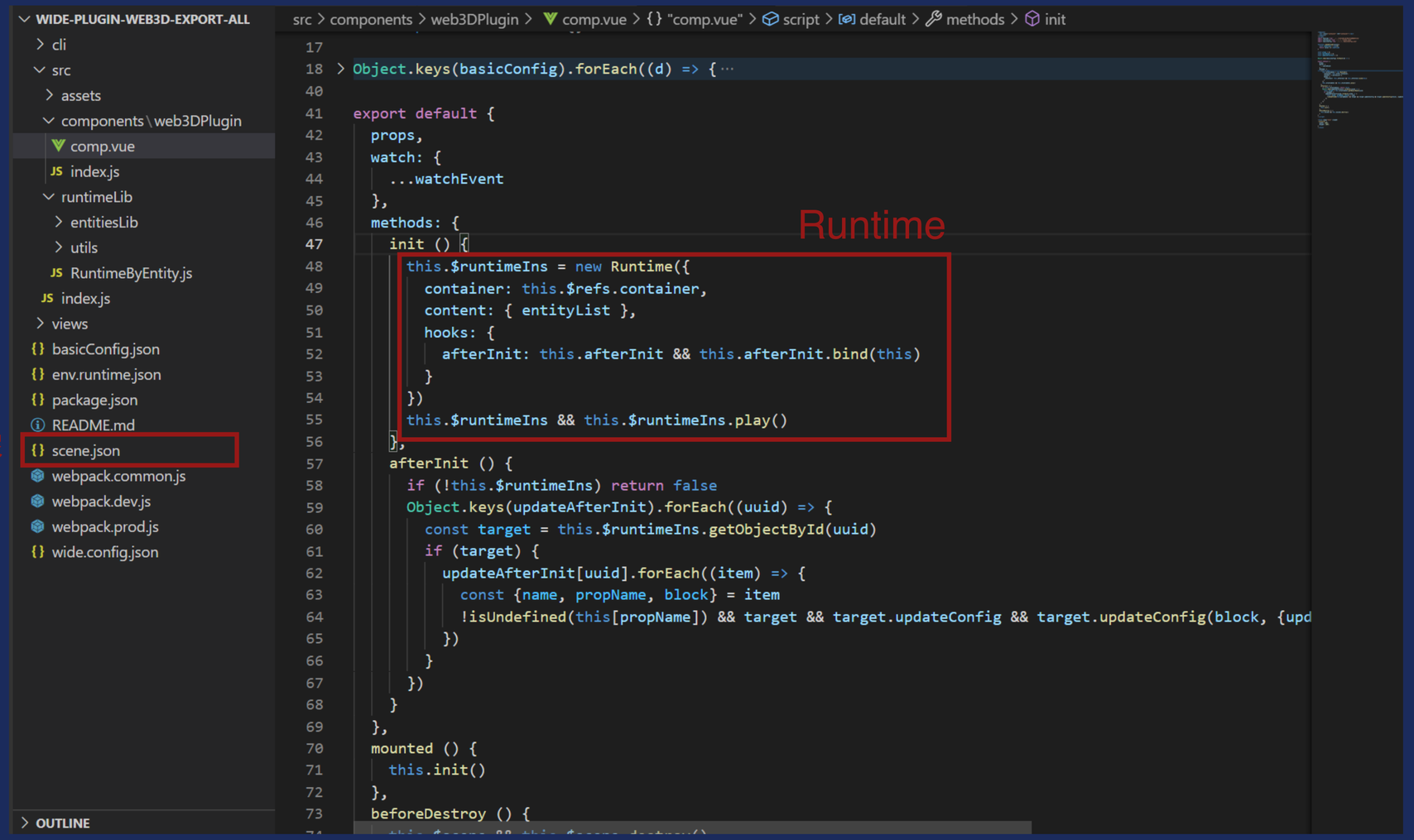

建模师
复现过程
研发
交付过程
交付
建模阶段
城市/行政区模型
导出
拓扑模型库
地球模型库
Blender建模
模型导出
研发阶段
模型导入
添加氛围
添加数据
添加脚本
交付阶段
配置数据
配置事件
配置方法
导出项目工程

导出项目工程

assets

lib

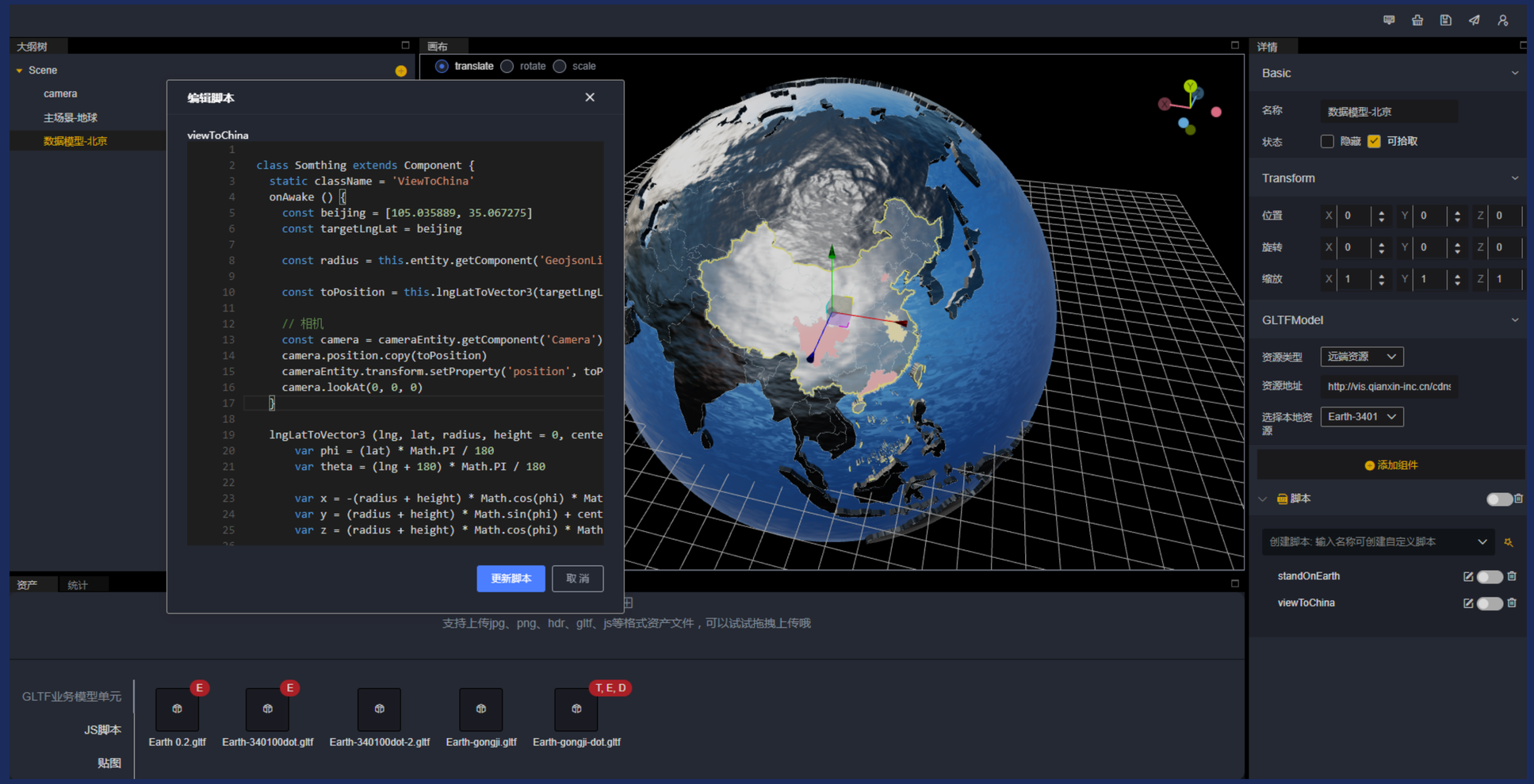
场景内容



大纲

- 业务背景
- 场景拆解和搭建
- **技术架构**
- 优化方案
- 未来规划和展望

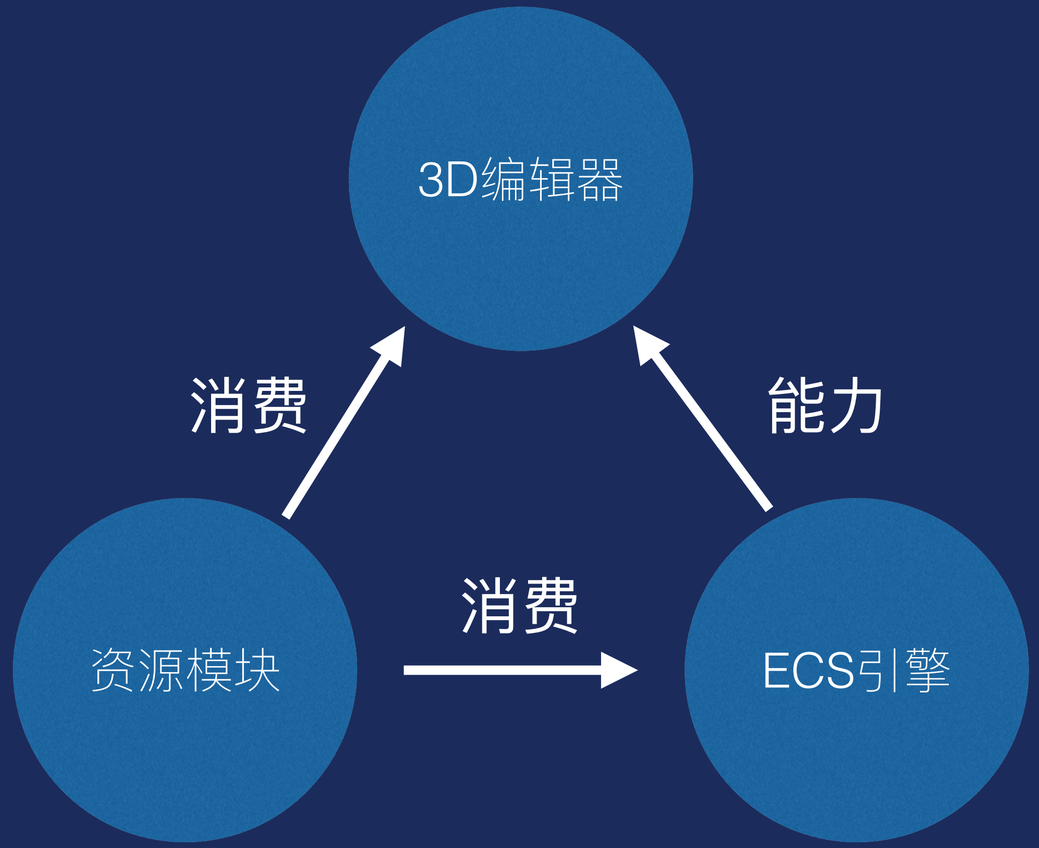
3D编辑器


大纲树
Scene
camera
主场景-地球
数据模型-北京
画布
translate
rotate
scale
编辑脚本
viewToChina
class Somthing extends Component {
static className = 'ViewToChina'
onAwake () {
const beijing = [105.035889, 35.067275]
const targetLngLat = beijing
const radius = this.entity.getComponent('GeojsonLi
const toPosition = this.lngLatToVector3(targetLngL
// 相机
const camera = cameraEntity.getComponent('Camera')
camera.position.copy(toPosition)
cameraEntity.transform.setProperty('position', toP
camera.lookAt(0, 0, 0)
}
lngLatToVector3 (lng, lat, radius, height = 0, cente
var phi = (lat) * Math.PI / 180
var theta = (lng + 180) * Math.PI / 180
var x = -(radius + height) * Math.cos(phi) * Mat
var y = (radius + height) * Math.sin(phi) + cent
var z = (radius + height) * Math.cos(phi) * Math
更新脚本
取消
详情
Basic
名称
数据模型-北京
状态
隐藏
可拾取
Transform
位置
旋转
缩放
GLTFModel
资源类型
远端资源
资源地址
http://vis.qianxin-inc.cn/cdn
选择本地资源
Earth-3401
添加组件
脚本
创建脚本: 输入名称可创建自定义脚本
standOnEarth
viewToChina
资产
统计
支持上传jpg、png、hdr、gltf、js等格式资产文件，可以试试拖拽上传哦
GLTF业务模型单元
JS脚本
贴图
Earth 0.2.gltf
Earth-340100dot.gltf
Earth-340100dot-2.gltf
Earth-gongji.gltf
Earth-gongji-dot.gltf

3D编辑器框架


应用层
WIDE Plugin
可部署项目Project
一体化: 资产中心
场景搭建
大纲树 Hierarchy
画布Canvas 渲染&&交互
资产管理 Resource
物料Entity
配置面板 Config Panel
脚本 Script
事件系统 Event
导出模块 Export
引擎层
ECS
FGL
THREE

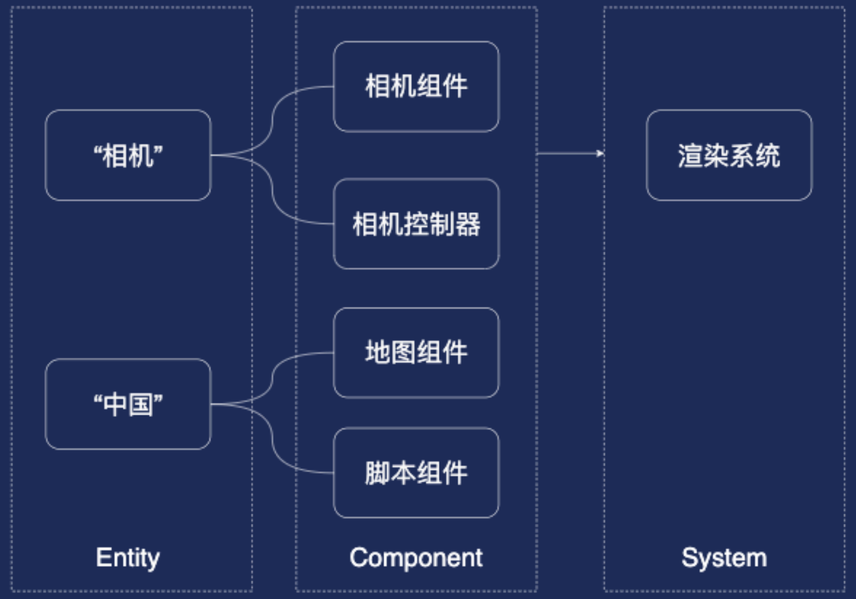
模块组织关系



资源模块


资源类型
GLTF业务模型单元
JavaScript脚本
Texture贴图
资源加载
资源校验
策略加载
预加载、闲时加载、按需加载
资源状态
资源管理模块
缓存机制
资源合并
场景标签
资源组织&&管理
Earth
City
District
Topo
自定义标签

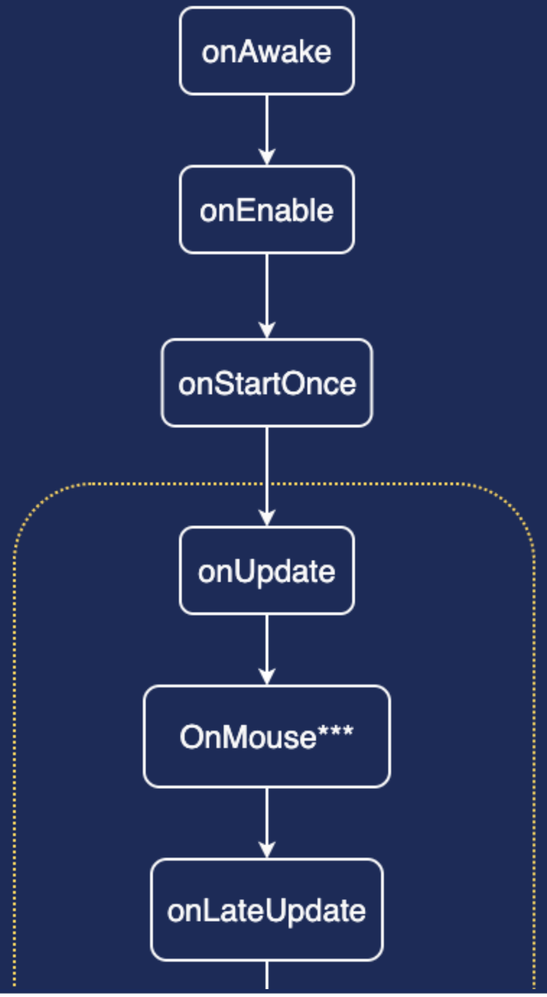
ECS设计



实体	场景中的实体节点，存储组件，具有GraphNode特性
组件	实体的能力，描述行为
系统	组件调度，控制行为

```
const engine = new Engine({
  autoStart: true,
  systems: [
    new RendererSystem({
      canvas: this.$refs.canvas,
    }),
    new PickerSystem()
  ],
})

const camera = new Entity('camera', {
  position: {
    z: 100
  },
})
engine.scene.addChild(camera)

camera.addComponent(
  new Camera({
    fov: 60,
    lookAt: { x: 0, y: -100, z: -100 }
  })
)
camera.addComponent(OrbitControl)

const china = new Entity('china')
engine.scene.addChild(china)

resourceManager.register({
  source: chinaUrl,
  autoLoad: false,
  name: 'chinaModel',
})
china.addComponent(
  GLTFModel,
  { name: 'chinaModel' }
)
china.addComponent(Zoom)
```

脚本生命周期



添加到实体中调用

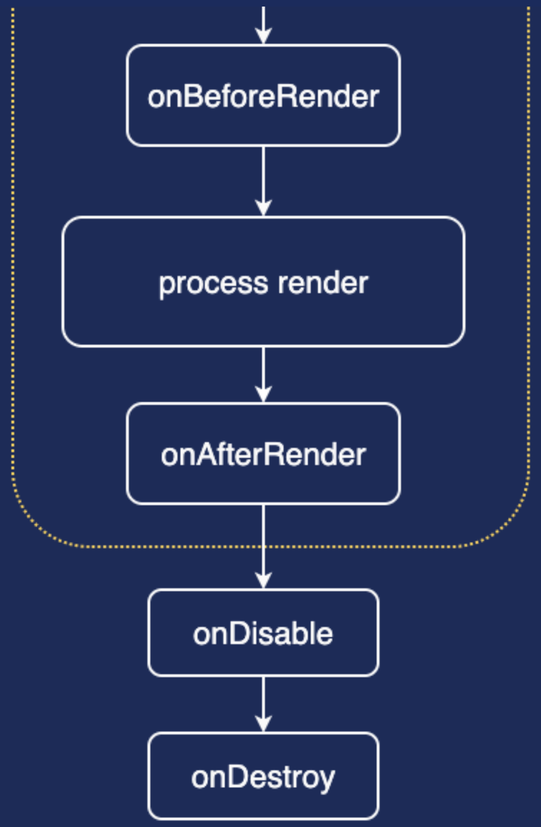
enabled属性为true时调用，初始化时如果enabled为true也会调用一次

第一次update前调用，只调用一次

每帧更新

鼠标事件

每帧更新



渲染前调用

渲染后调用

销毁前会调用，enabled为false时也会调用

销毁时调用

研发需要实现的能力

模型加载	代码生成 业务模型	光照	氛围
后期	数据	动画	脚本

ECS脚本开发

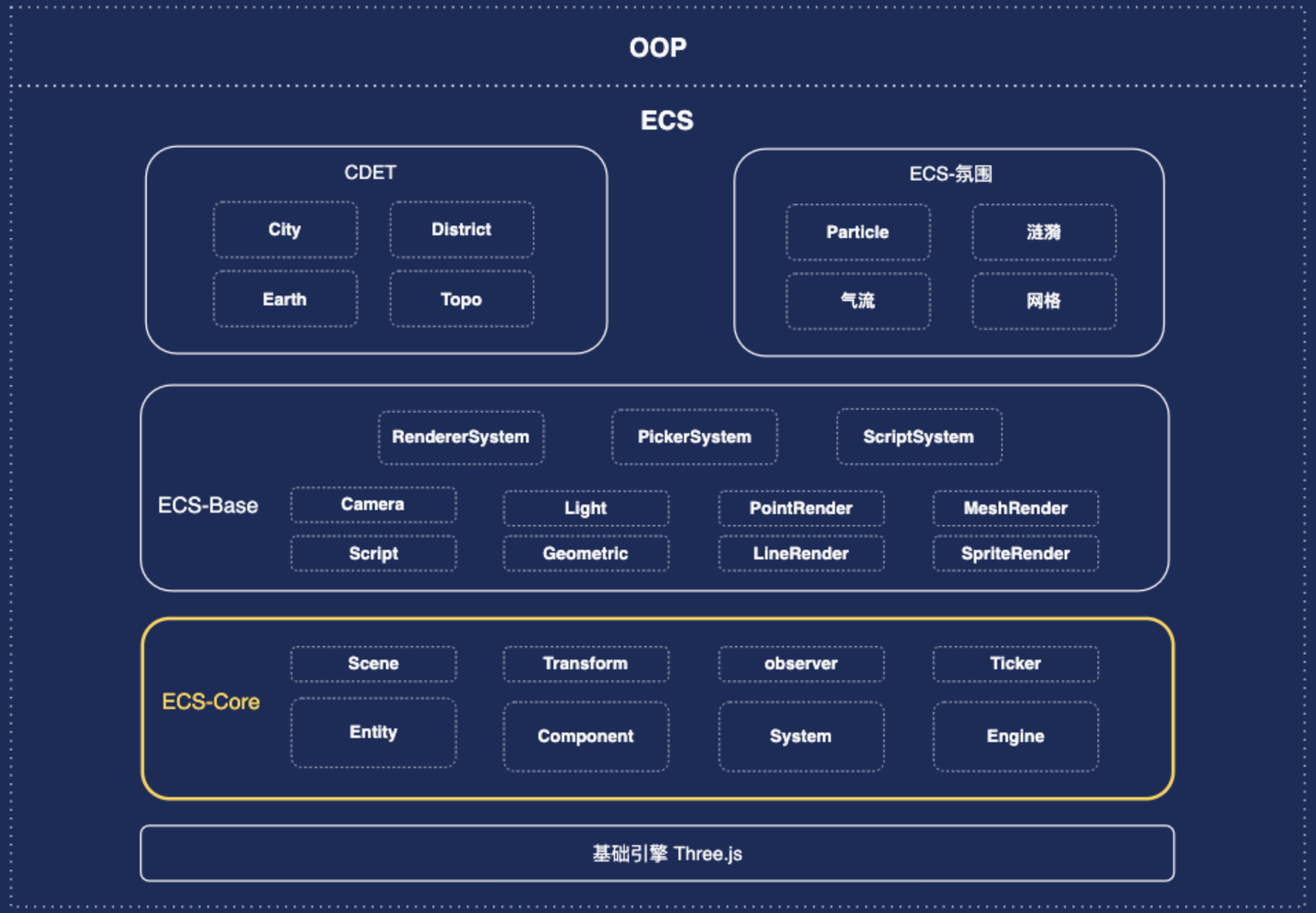
```
class Swing extends Component {
  static className = 'Swing'
  static defaultComponentParams = {
    speed: {
      x: 0,
      y: 0,
      z: 0,
    }
  }

  onAwake (params) {
    this.direction = 1
  }

  onUpdate () {
    const oldX = this.entity.transform.position.x
    const newX = oldX + this.speed.x * this.direction
    if (newX > 50) {
      this.direction = -1
    } else if (newX < -50) {
      this.direction = 1
    }
    this.entity.transform.position.x = newX
  }
```

自定义行为组件

```
const entity = new Entity('camera', {
  position: {
    z: 100,
  },
})

entity.addComponent(
  new Camera({
    fov: 60,
  })
)

entity.addComponent(
  Swing,
  { speed: { x: 1 } }
)
```

实体添加组件

代码分层



业务逻辑代码

业务组件

渲染基础

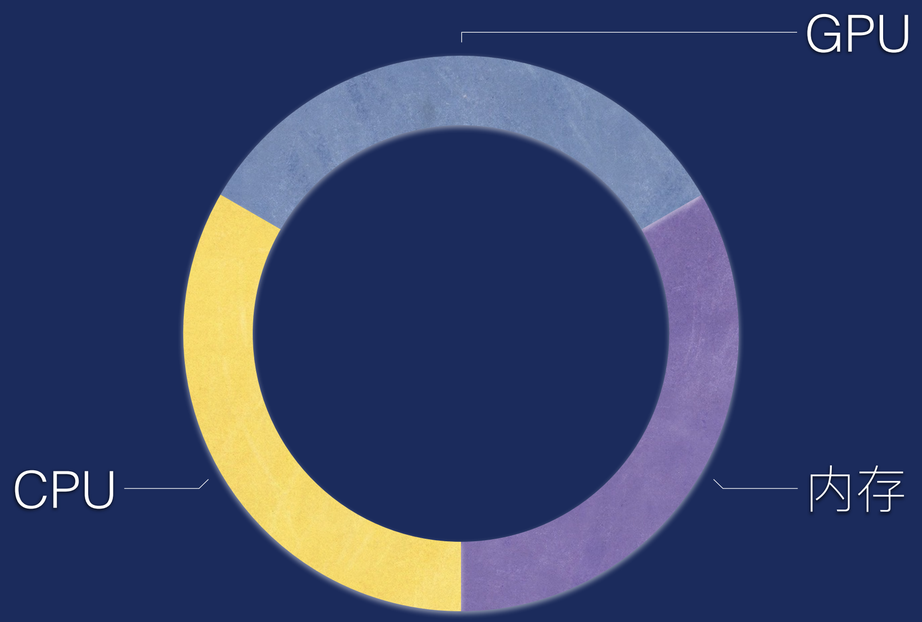
核心

依赖

大纲

- 业务背景
- City场景案例
- 技术架构
- 优化方案
- 未来规划和展望

优化方案



减少GPU上传
- 渲染优化：Mesh合并
- 纹理优化：纹理合并

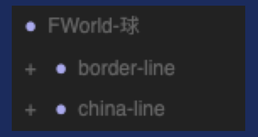
增加本地缓存，减少网络请求
多线程计算，减少主线程压力
- 计算优化：WebWorker
- 缓存优化：IndexedDB
- 交互优化：GPU拾取

减少GPU和CPU内存占用
- 面数优化：复用资源
- 文字优化：矢量字体

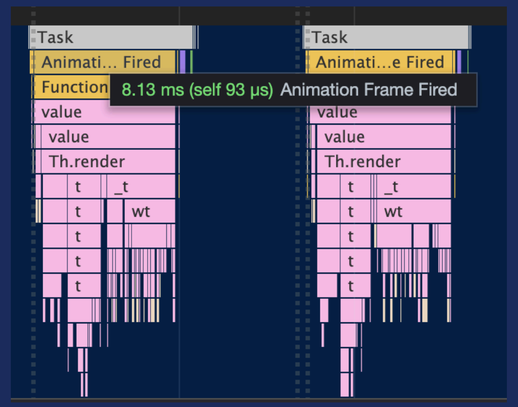
渲染优化

渲染的场景大，模型多，导致帧率低

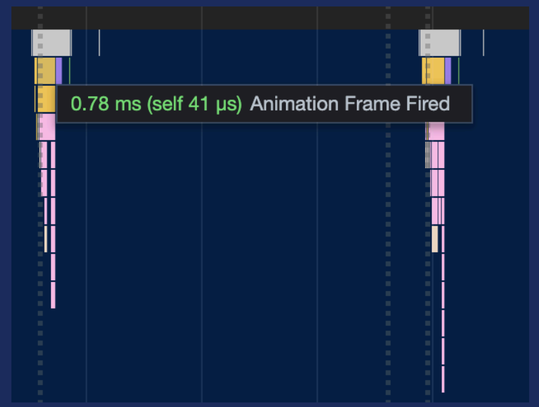
减少DrawCall 相同材质的物体合并Geometry



合并前



合并后



面数优化

大场景中，使用的模型面数多，文件大，复制的面数少，物体多

名称	大小
1x2000.gltf	1.1 MB
10x200.gltf	160 KB
40x50.gltf	134 KB
1000x2.gltf	2.6 MB
2000x1.gltf	5.7 MB

8ms 树

1ms

0.64ms 树林

0.5ms

0.5ms 森林

合并且复用物体

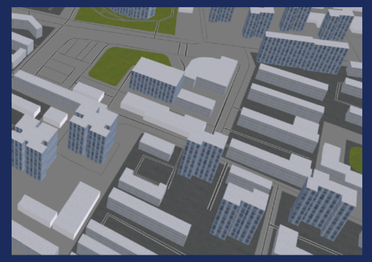
利用纹理和alpha使用简易面

纹理优化

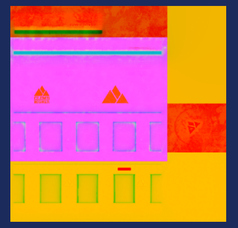
图片上传多、琐碎

纹理合并上传，Mesh调整UV使用

```
"atlasSize": [
  2000,
  500
],
"atlasMapping": {
  "ddos": { "x": 0, "y": 0, "width": 500, "height": 500 },
  "bug": { "x": 500, "y": 0, "width": 500, "height": 500 },
  "psd": { "x": 1000, "y": 0, "width": 500, "height": 500 },
  "scan": { "x": 1500, "y": 0, "width": 500, "height": 500 }
}
```



材质多个属性通道烘焙到一个纹理



Ambient Occlusion（R）

Roughness（G）

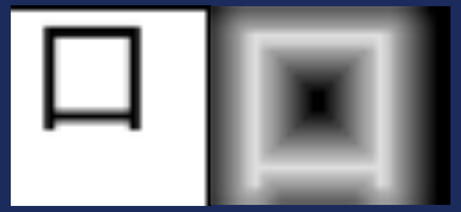
Metallic（B）

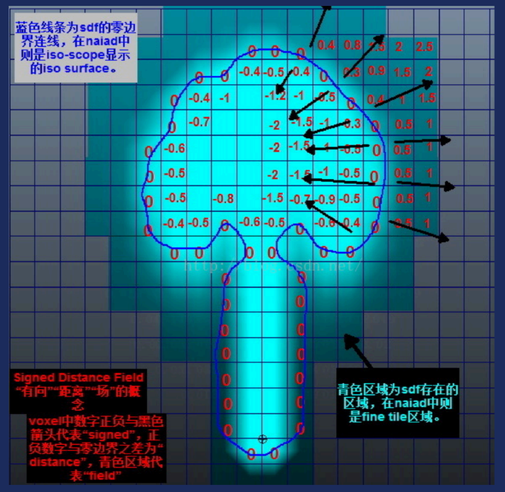
Base Color

NormalMap

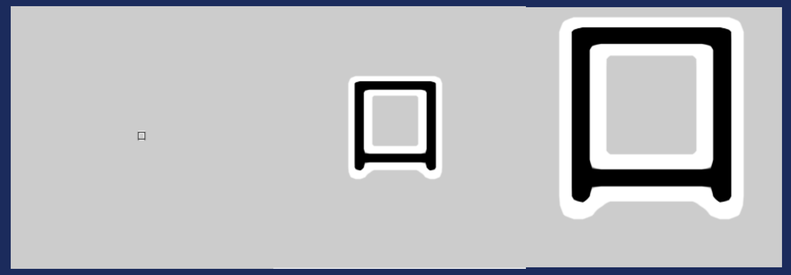
文字优化

超大文字渲染更优方案，SDF（Signed Distance Field）



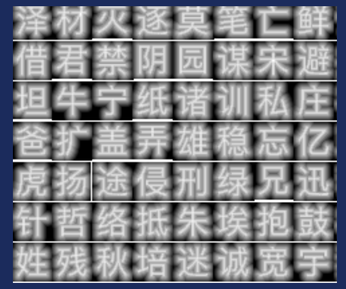


生成SDF文字



24　　256　　500

SDF字符集



渲染SDF文字代码

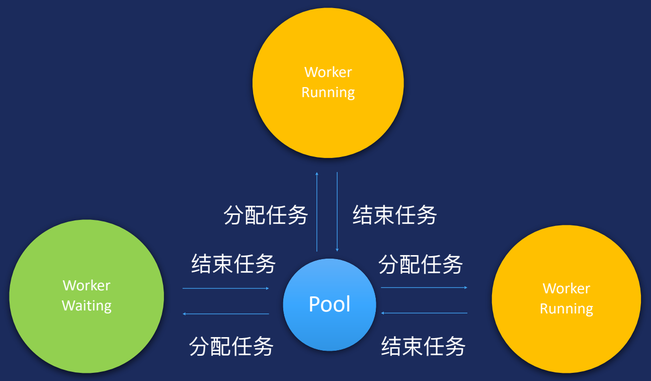
```
float dist = texture2D( texture, vec2(offset_x + scale_x, offset_y + scale_y) ).r;
float alpha = smoothstep(u_buffer - u_gamma, u_buffer + u_gamma, dist);
gl_FragColor = vec4(v_fillColor, alpha);
```

计算优化

文字碰撞检测、模型解析占用主线程时间，希望可以不阻塞主线程

Web Worker

SharedArrayBuffer



Worker Pool

WorkerPool伪代码

```
const worker = new Proxy(new Worker(), {
  get (target, key) {
    if (key === 'postMessage') {
      target.parallel++
      return target.postMessage.bind(target)
    }
    if (typeof target[key] === 'function') {
      return target[key].bind(target)
    } else {
      return target[key]
    }
  },
  set (target, key, value) {
    if (key === 'onmessage') {
      target[key] = function (...args) {
        target.parallel--
        value(...args)
      }
    } else {
      target[key] = value
    }
    return true
  }
})
```

生成WorkerPool，根据task值判断结果类型

```
this._worker = new Worker(Parser, ({ data }) => {
  // task collision 代表文本碰撞
  if (data.task === taskTypes.COLLISION) {…
  }

  if ((data.task === taskTypes.TILE_PARSER) && (data
  }
})
```

创建任务，根据task值判断任务类型

```
this._worker.postMessage({
  task: taskTypes.COLLISION,
  timestamp: this._currentCollisionTimestamp,
  data: {…
  }
})
```

交互优化

物体多，物体合并，拾取不准确，射线检测慢

GPU拾取

给每个实体一种颜色，渲染一遍，有鼠标事件时读取像素颜色，反推实体ID

- 多Mesh合并后的精确拾取
- GPU修改顶点的物体拾取

```
/**
* id转颜色
* @param {Number} id
*/
function packID (id) {
    var r = id >> 16
    var g = (id - (r << 8)) >> 8
    var b = id - (r << 16) - (g << 8)
    return [r, g, b]
}
```

```
/**
* 颜色转id
* @param {Number} r
* @param {Number} g
* @param {Number} b
*/
function unpackID (r, g, b) {
    return (r << 16) + (g << 8) + b
}
```

八叉树拾取

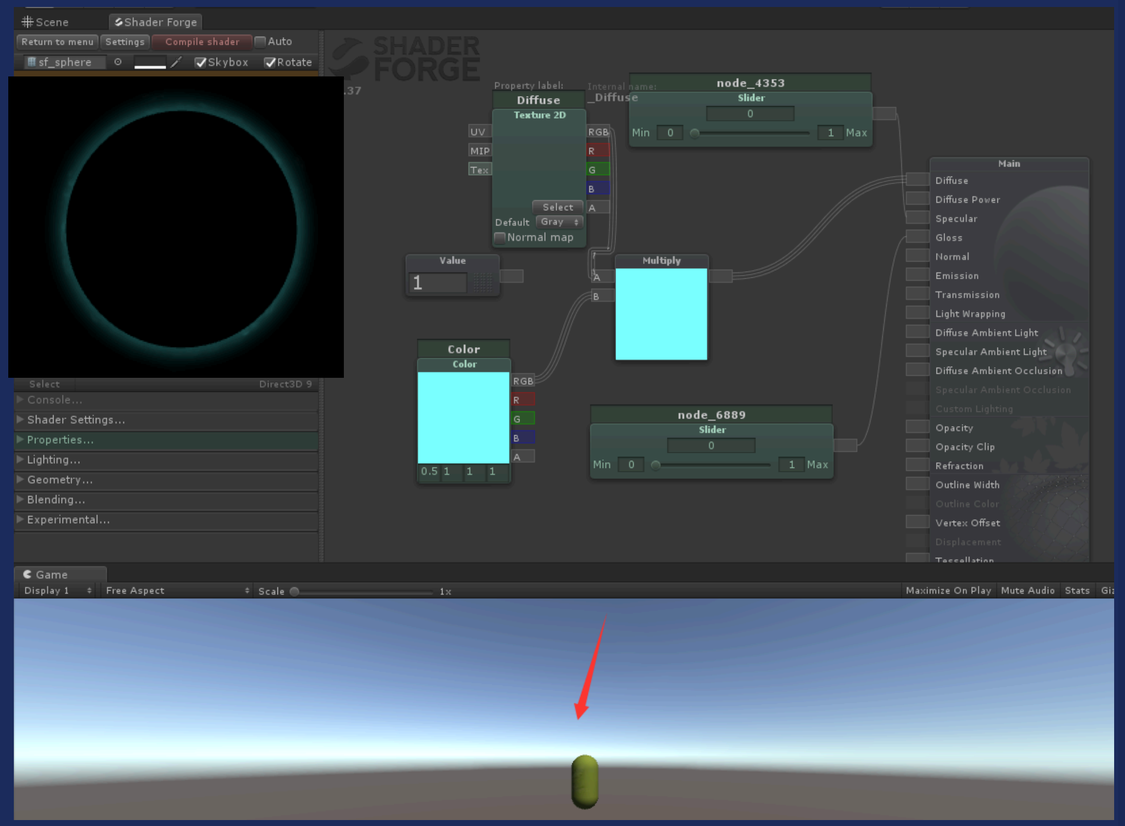
使用八叉树数据结构，将场景中的模型放入不同cell中，适合稳定布局场景

大纲

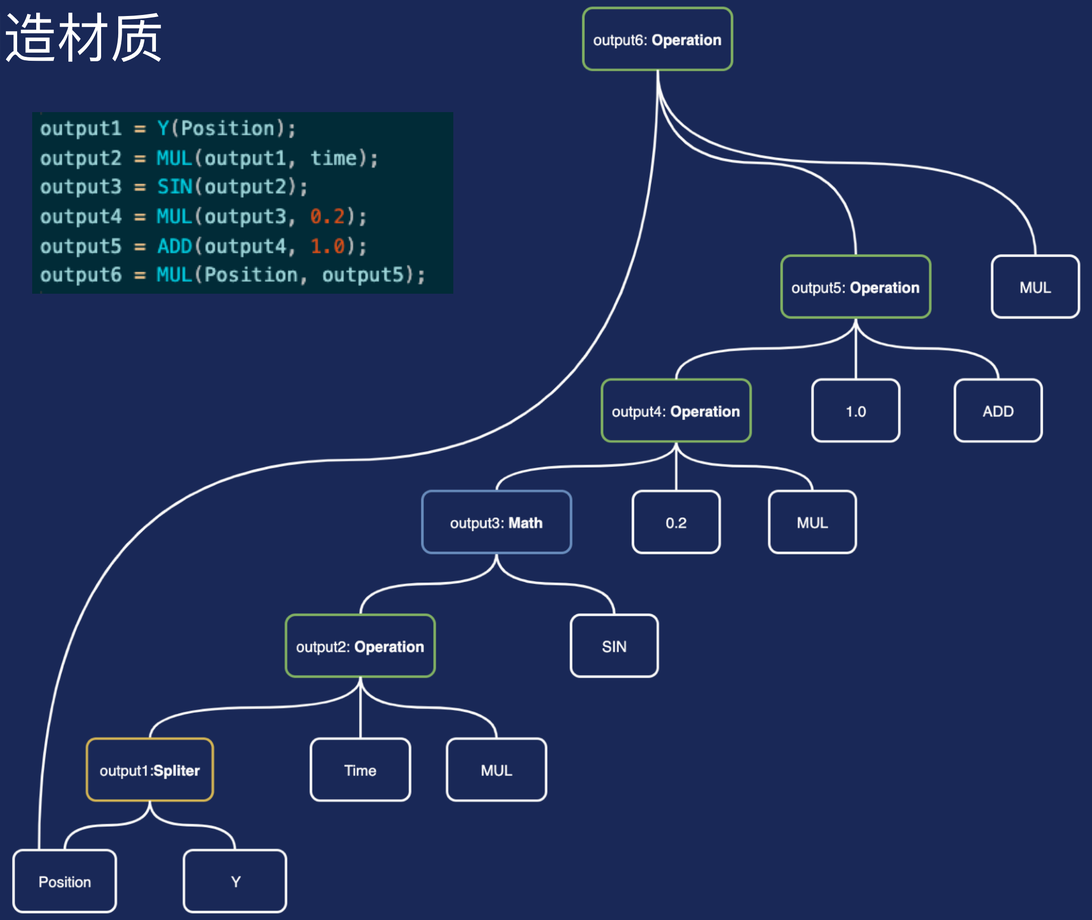
- 业务背景
- City场景案例
- 技术架构
- 优化方案
- **未来规划和展望**

材质编辑器

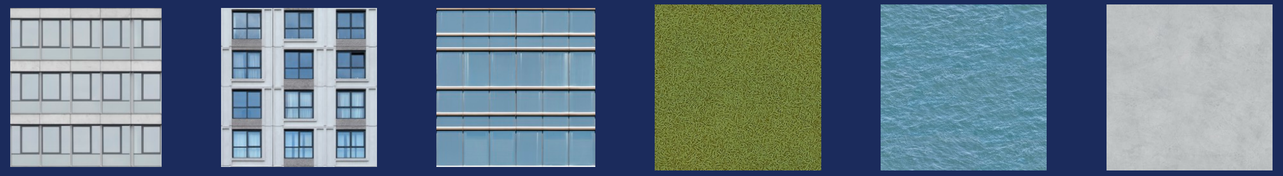
材质节点编辑，解决自定义材质问题，帮助设计师创造材质

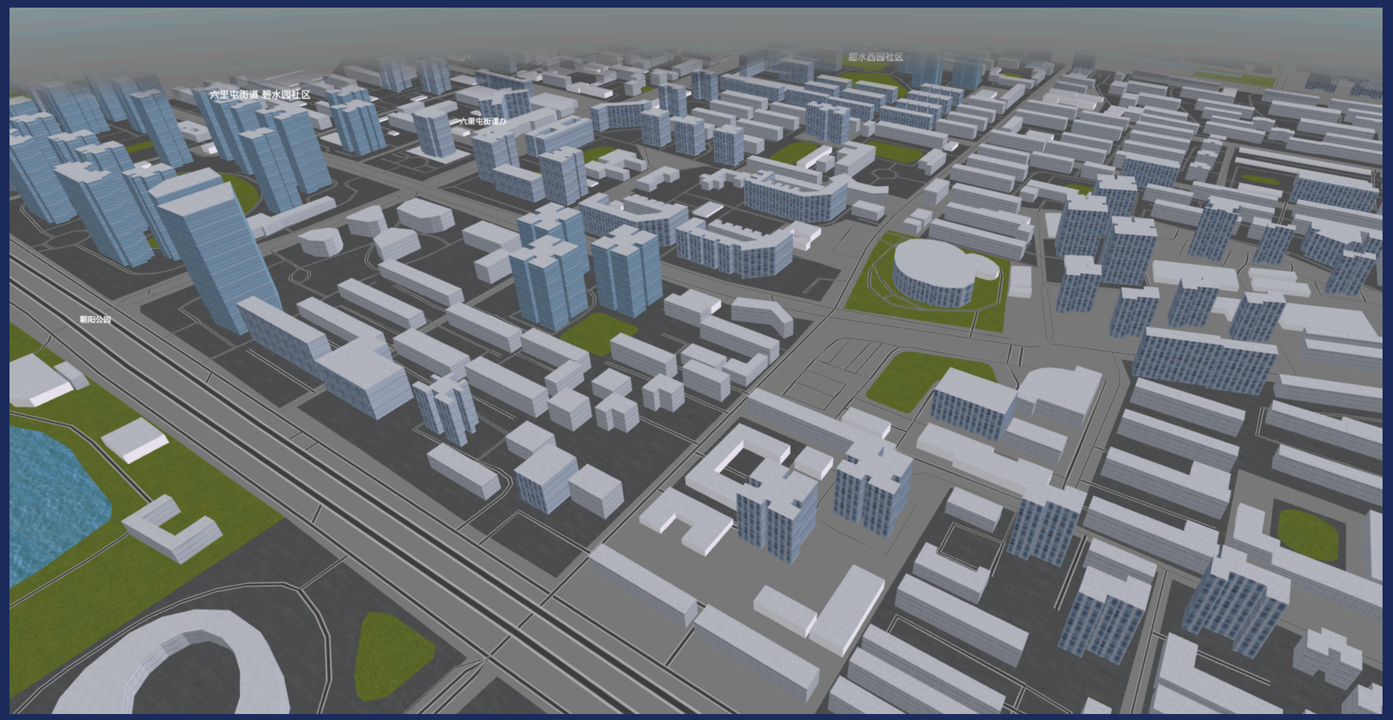


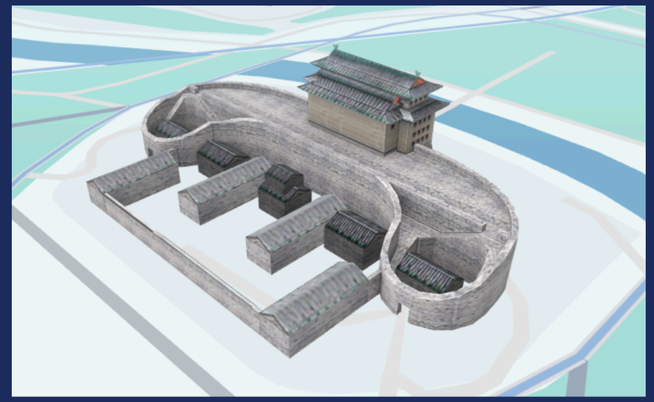

```
output1 = Y(Position);
output2 = MUL(output1, time);
output3 = SIN(output2);
output4 = MUL(output3, 0.2);
output5 = ADD(output4, 1.0);
output6 = MUL(Position, output5);
```



地图 vs 定制建筑







总结

- 3D模型化：释放压力
- 场景搭建和工作流：明确角色分工
- ECS引擎：组合灵活
- 优化方案：量体裁衣

GMTC
全球大前端技术大会

THANKS

InfoQ