GAMES 10 作业框架

Utopia 简介

庄涛
中国科学技术大学

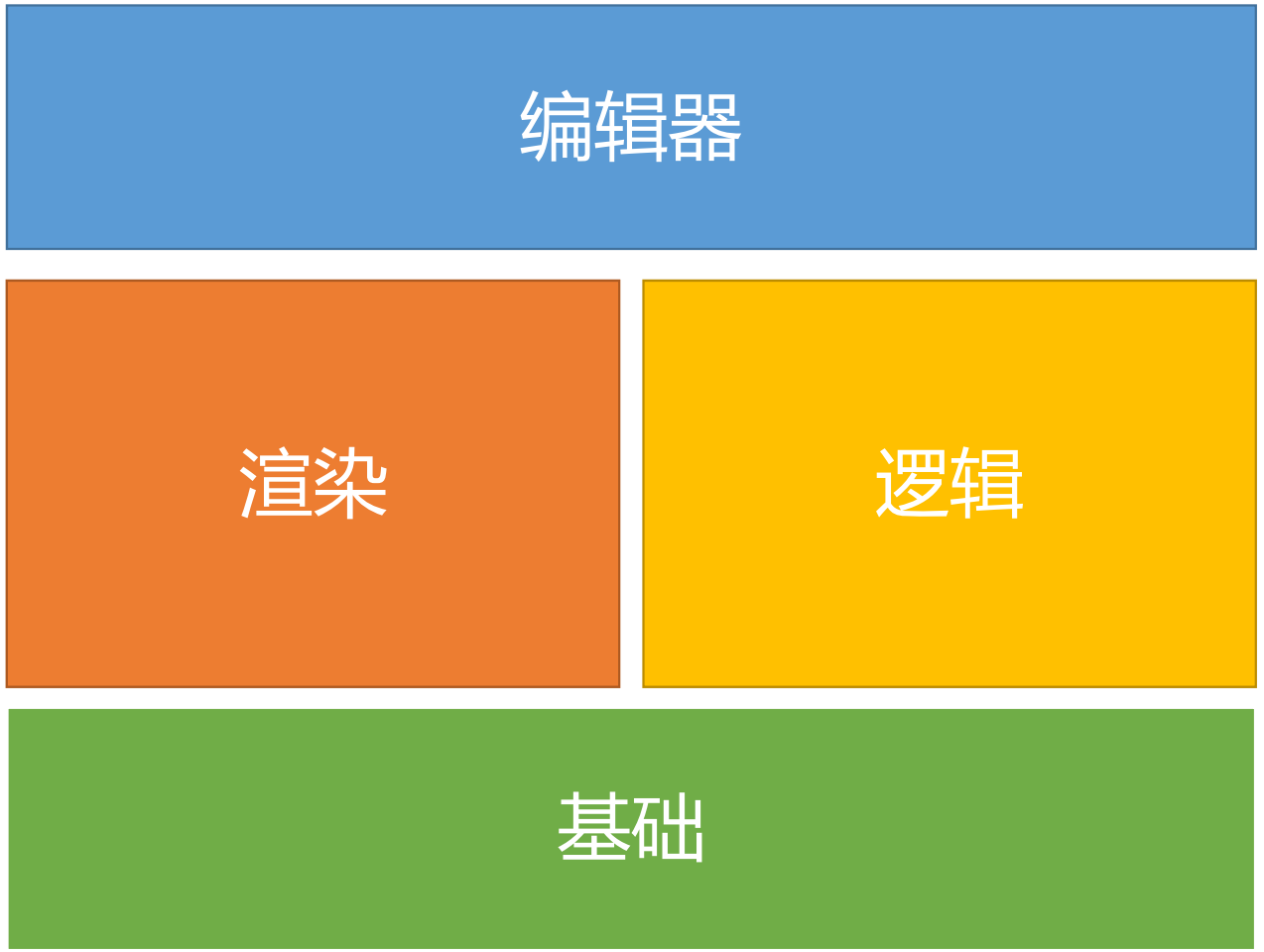
目录

架构

使用

架构

架构



基础

反射

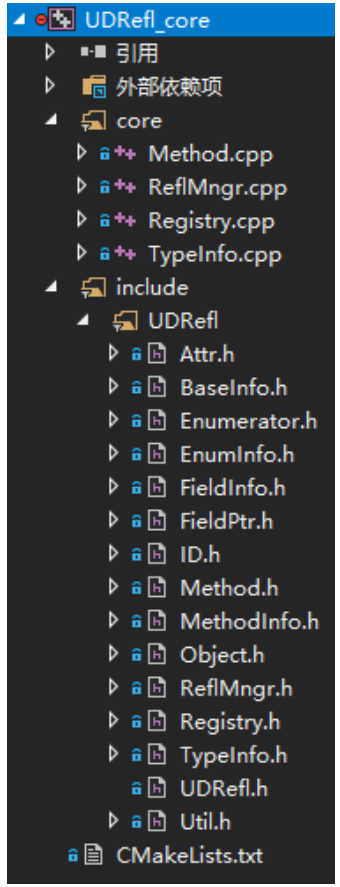
图形数学库

STL

CMake

CMake

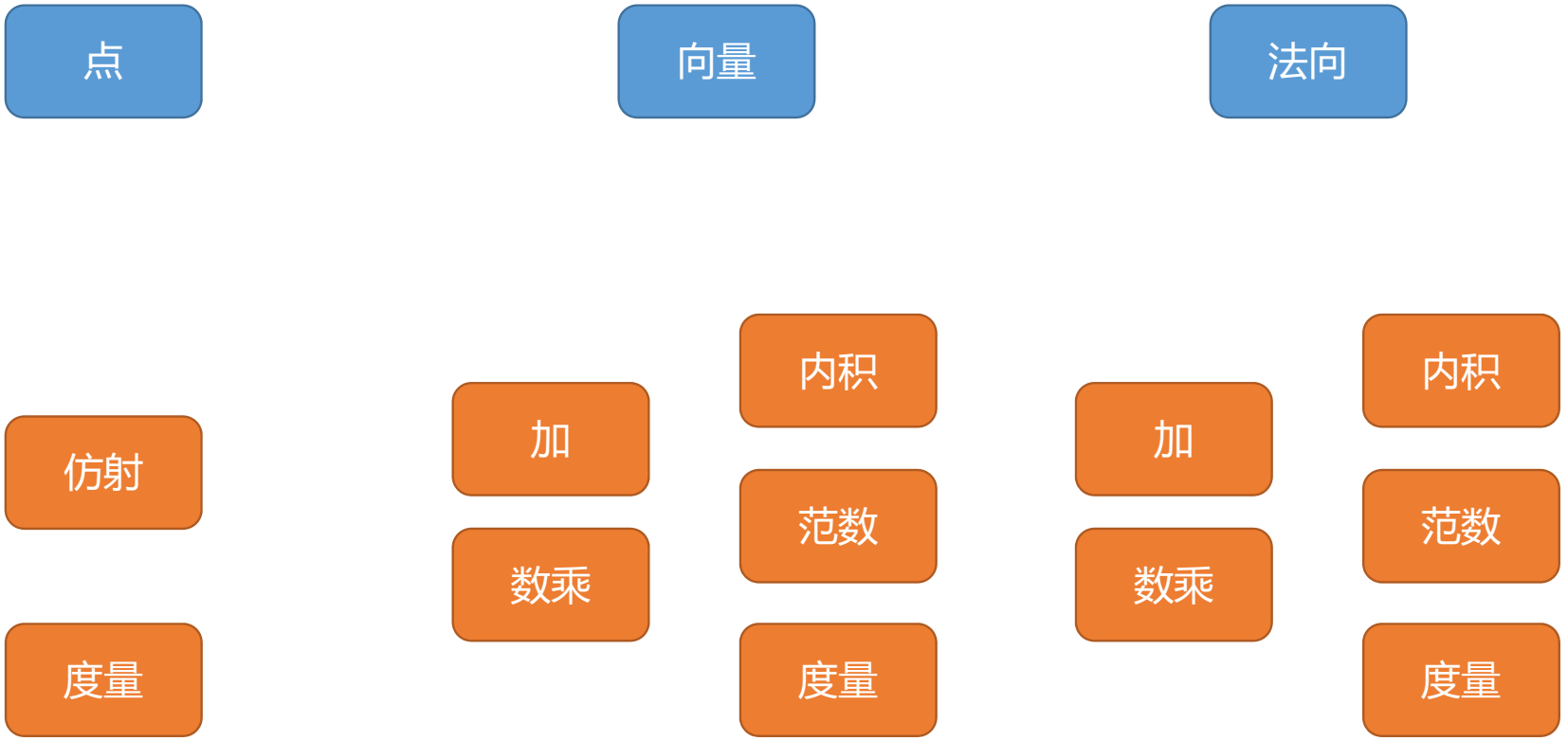
```
Ubpa_AddTarget(
  MODE STATIC
  SOURCE
    "${PROJECT_SOURCE_DIR}/include/UDRefl"
  INC
    "${PROJECT_SOURCE_DIR}/include"
  LIB
    Ubpa::UContainer_core
)
```



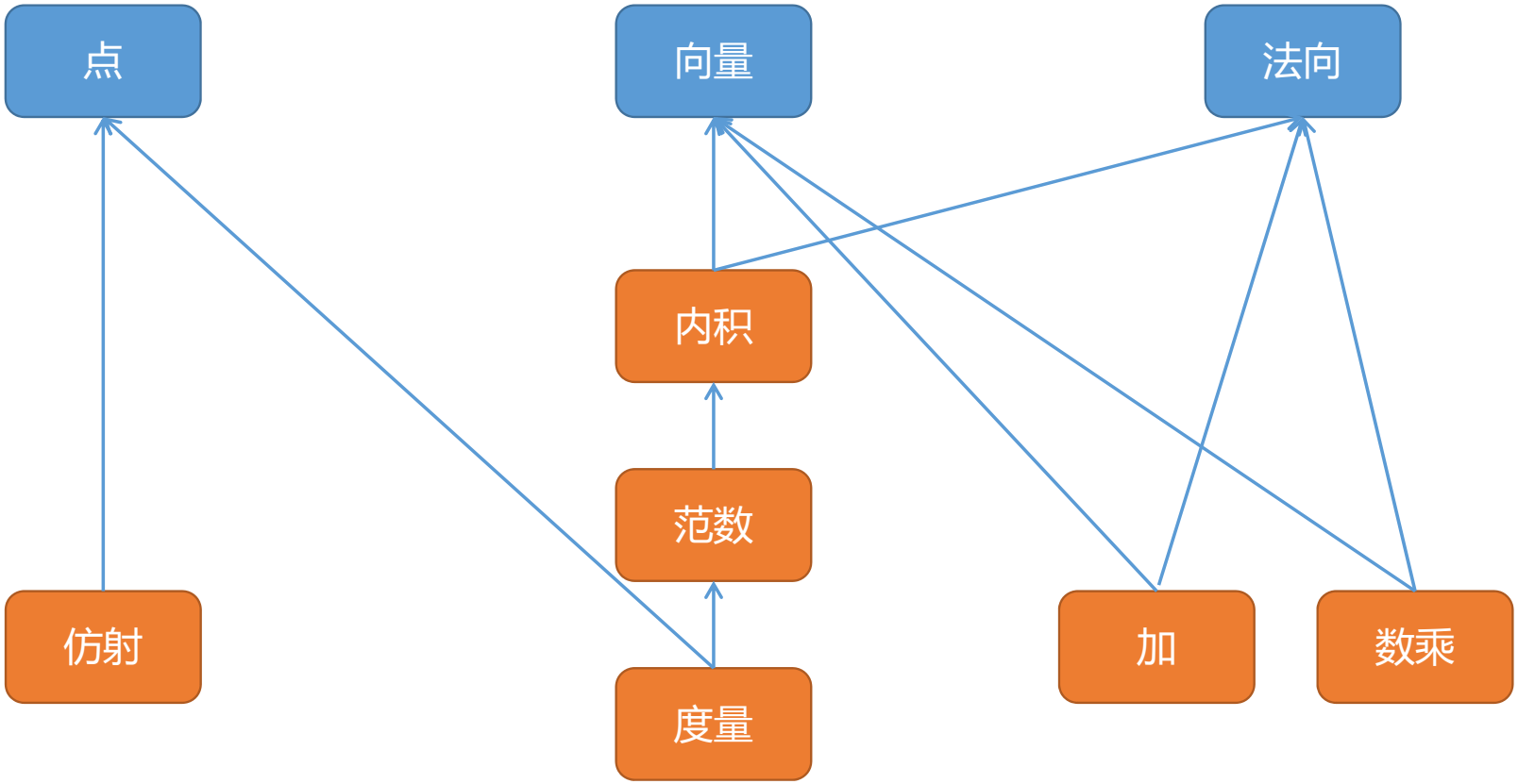
图形数学库

```
transformf tsfm{
  vecf3{1,1,1},                          // T
  quatf{vecf3{1,0,0}, to_radian(90.f), // R
  scalef3{2.f}}                          // S
}; // T * R * S

pointf3 p{ 1,2,3 };
vecf3   v{ 1,1,1 };
normalf n{ 0,1,0 };
bboxf3  b{ p, p + v }; // min: 1 2 3, max: 2 3 4
rayf3   r{ p, v }; // point: 1 2 3, dir: 1 1 1, tmin: EPSILON, tmax: FLT_MAX

cout << tsfm * p << endl; // 3 -5 5
cout << tsfm * v << endl; // 2 -2 2
cout << tsfm * n << endl; // 0 0 0.5
cout << tsfm * b << endl; // 3 -7 5, 5 -5 7
cout << tsfm * r << endl; // 3 -5 5, 2 -2 2, EPSILON, FLT_MAX
```

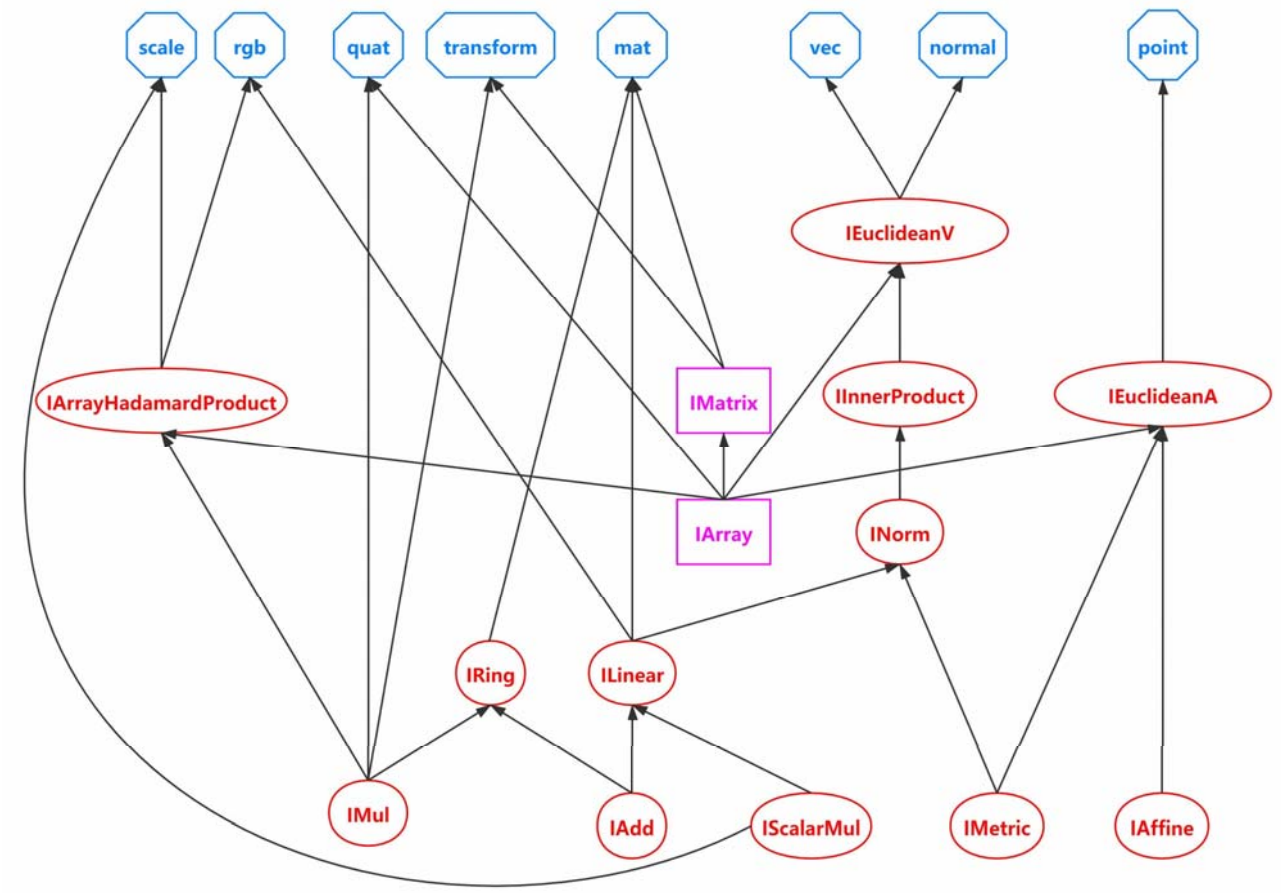
图形数学库



图形数学库



图形数学库


scale
rgb
quat
transform
mat
vec
normal
point
IEuclideanV
IArrayHadamardProduct
IMatrix
IInnerProduct
IEuclideanA
IArray
INorm
IRing
ILinear
IMul
IAdd
IScalarMul
IMetric
IAffine

图形数学库

```
template<typename Base, typename Impl>
struct IInnerProduct : Base {
    using F = ImplTraits_F<Impl>;

    F dot(const Impl& y) const;

    F norm2() const;

    F distance2(const Impl& y) const;

    F cos_theta(const Impl& y) const;

    F cot_theta(const Impl& y) const;

    Impl project(const Impl& n) const;
    Impl perpendicular(const Impl& n) const;
};
```

反射

<type_traits>

Type properties

Defined in header <type_traits>

Primary type categories

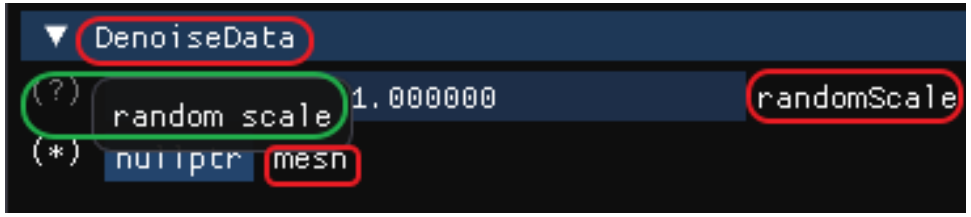
is_void	(C++11)
is_null_pointer	(C++14)
is_integral	(C++11)
is_floating_point	(C++11)
is_array	(C++11)
is_enum	(C++11)

Type relationships

is_same	(C++11)
is_base_of	(C++11)
is_convertible is_nothrow_convertible	(C++11) (C++20)
is_invocable is_invocable_r is_nothrow_invocable is_nothrow_invocable_r	(C++17)

静态反射

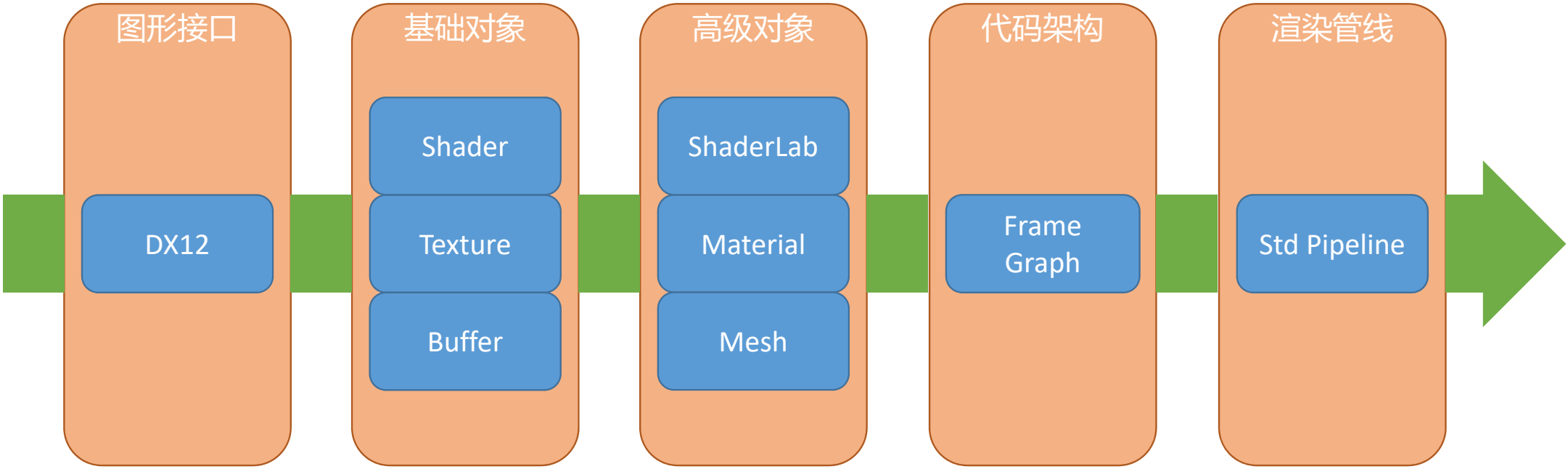
USRefl




```cpp
struct DenoiseData {
    [[UInspector::min_value(0.f)]]
    [[UInspector::tooltip("random scale")]]
    float randomScale = 1.f;

    std::shared_ptr<Ubpa::Utopia::Mesh> mesh;

    [[UInspector::hide]]
    std::shared_ptr<HEMeshX> heMesh{ std::make_shared<HEMeshX>() };

    [[UInspector::hide]]
    Ubpa::Utopia::Mesh copy;
};
```

```cpp
template<>
struct Ubpa::USRefl::TypeInfo<DenoiseData> :
    TypeInfoBase<DenoiseData>
{
    static constexpr char name[12] = "DenoiseData";

    static constexpr AttrList attrs = {};
    static constexpr FieldList fields = {
        Field {TSTR("randomScale"), &Type::randomScale, AttrList {
            Attr {TSTR(UInspector::min_value), 0.f},
            Attr {TSTR(UInspector::tooltip), "random scale"},
        }},
        Field {TSTR("mesh"), &Type::mesh},
        Field {TSTR("heMesh"), &Type::heMesh, AttrList {
            Attr {TSTR(UInspector::hide)},
        }},
        Field {TSTR("copy"), &Type::copy, AttrList {
            Attr {TSTR(UInspector::hide)},
        }},
    };
};
```

渲染


图形接口
DX12
基础对象
Shader
Texture
Buffer
高级对象
ShaderLab
Material
Mesh
代码架构
Frame Graph
渲染管线
Std Pipeline

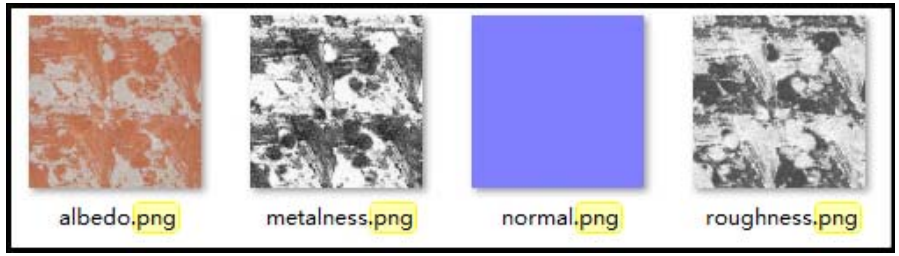
渲染

Shader

```
struct VertexOut
{
    float4 PosH : SV_POSITION;
};

VertexOut VS(VertexIn vin)
{
    VertexOut vout = (VertexOut)0.0f;

    // Transform to world space.
    float4 posW = mul(gWorld, float4(vin.PosL, 1.0f));

    // Transform to homogeneous clip space.
    vout.PosH = mul(gViewProj, posW);

    return vout;
}

struct PixelOut {
    float4 color : SV_Target0;
};

PixelOut PS(VertexOut pin)
{
    PixelOut pout;
    pout.color = float4(gColor, 1);

    return pout;
}
```

渲染

Texture



Buffer

```
v -1 -1 -1
v -1 1 -1
v 1 -1 -1
v 1 1 -1
v -1 -1 1
v 1 -1 1
v -1 1 1
v 1 1 1
v -1 -1 1
v -1 1 1
v -1 -1 -1
v -1 1 -1
v 1 1 1
v 1 -1 1
v 1 1 -1
v 1 -1 -1
v 1 -1 1
v -1 -1 1
v 1 -1 -1
v -1 -1 -1
v -1 1 1
v 1 1 1
v -1 1 -1
v 1 1 -1
```

```
vn 0 0 -1
vn 0 0 -1
vn 0 0 -1
vn 0 0 -1
vn 0 0 1
vn 0 0 1
vn 0 0 1
vn 0 0 1
vn -1 0 0
vn -1 0 0
vn -1 0 0
vn -1 0 0
vn 1 0 0
vn 1 0 0
vn 1 0 0
vn 1 0 0
vn 0 -1 0
vn 0 -1 0
vn 0 -1 0
vn 0 -1 0
vn 0 1 0
vn 0 1 0
vn 0 1 0
vn 0 1 0
```

```
vt 1 0
vt 1 1
vt 0 0
vt 0 1
vt 0 0
vt 1 0
vt 0 1
vt 1 1
vt 1 0
vt 1 1
vt 0 0
vt 0 1
vt 0 1
vt 0 0
vt 1 1
vt 1 0
vt 1 1
vt 0 1
vt 1 0
vt 0 0
vt 0 0
vt 1 0
vt 0 1
vt 1 1
```

```
f 1/1/1 2/2/2 3/3/3
f 4/4/4 3/3/3 2/2/2
f 5/5/5 6/6/6 7/7/7
f 8/8/8 7/7/7 6/6/6
f 9/9/9 10/10/10 11/11/11
f 12/12/12 11/11/11 10/10/10
f 13/13/13 14/14/14 15/15/15
f 16/16/16 15/15/15 14/14/14
f 17/17/17 18/18/18 19/19/19
f 20/20/20 19/19/19 18/18/18
f 21/21/21 22/22/22 23/23/23
f 24/24/24 23/23/23 22/22/22
```

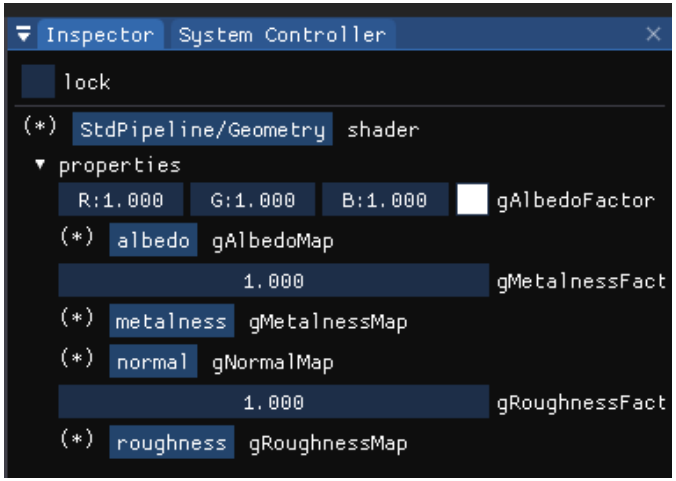
渲染

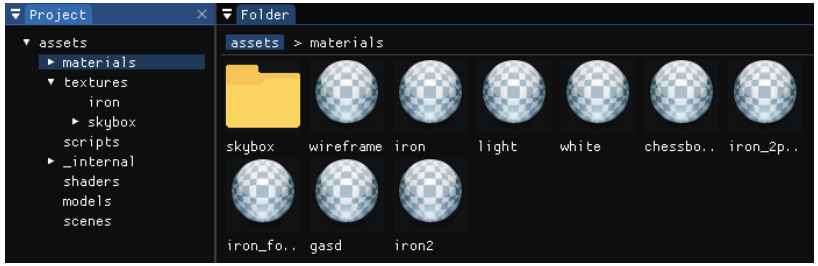
ShaderLab

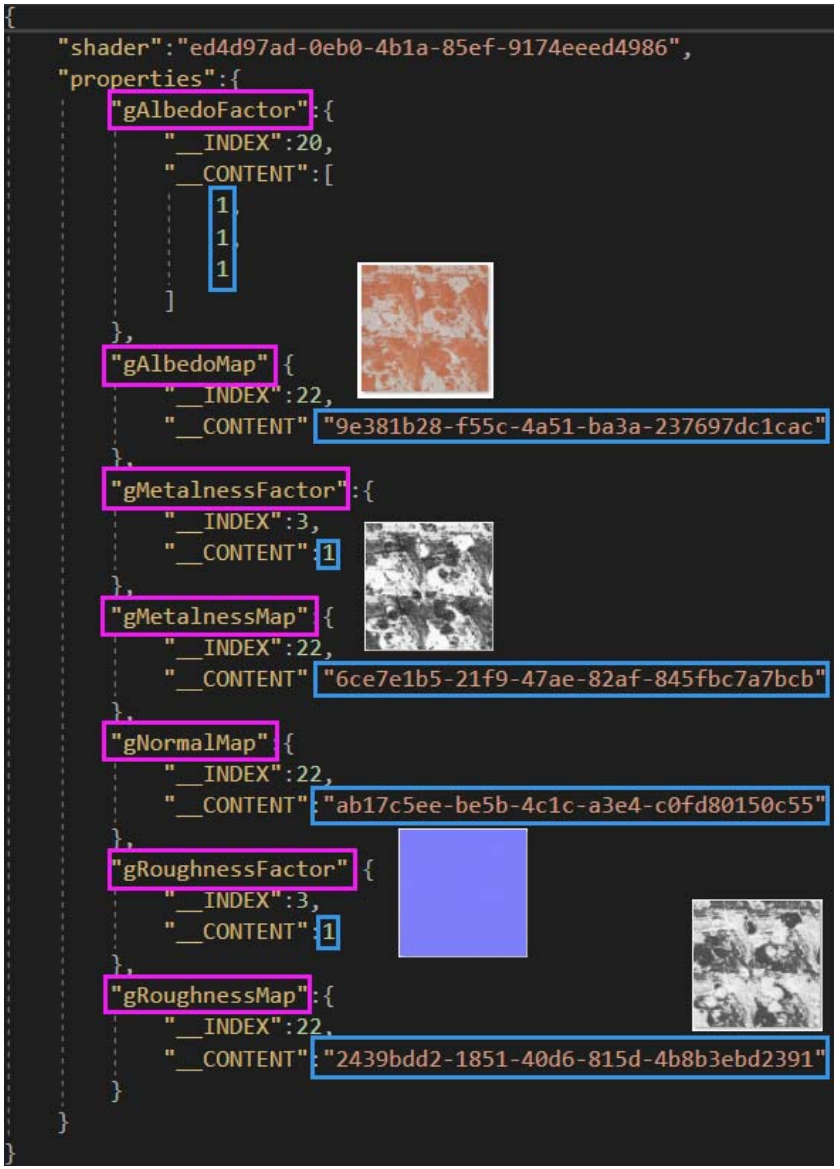
```
Shader "StdPipeline/Geometry" {
    HLSL : "d1ee38ec-7485-422e-93f3-c8886169a858"
    RootSignature {
        SRV[1] : 0
        SRV[1] : 1
        SRV[1] : 2
        SRV[1] : 3
        SRV[1] : 4
        CBV : 0
        CBV : 1
        CBV : 2
    }
    Properties {
        gAlbedoMap   ("albedo"   , 2D) : White
        gEmissionMap ("metalness", 2D) : Black
        gMetalnessMap("metalness", 2D) : White
        gRoughnessMap("roughness", 2D) : White
        gNormalMap   ("albedo"   , 2D) : Bump

        gAlbedoFactor   ("albedo factor"  , Color3) : (1, 1, 1)
        gEmissionFactor ("emission factor", Color3) : (1, 1, 1)
        gRoughnessFactor("roughness factor", float) : 1
        gMetalnessFactor("metalness factor", float) : 1
    }
    Pass (VS, PS) {
        Tags {
            "LightMode" : "Deferred"
        }
    }
}
```

渲染

Material

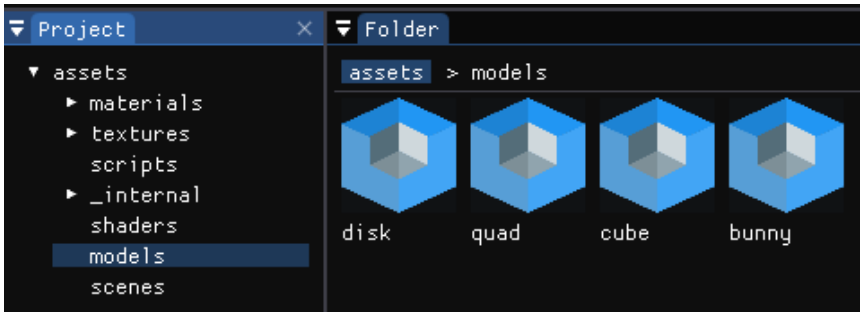

Inspector
System Controller
lock
(*) StdPipeline/Geometry shader
properties
R:1.000 G:1.000 B:1.000 gAlbedoFactor
(*) albedo gAlbedoMap
1.000 gMetalnessFact
(*) metalness gMetalnessMap
(*) normal gNormalMap
1.000 gRoughnessFact
(*) roughness gRoughnessMap

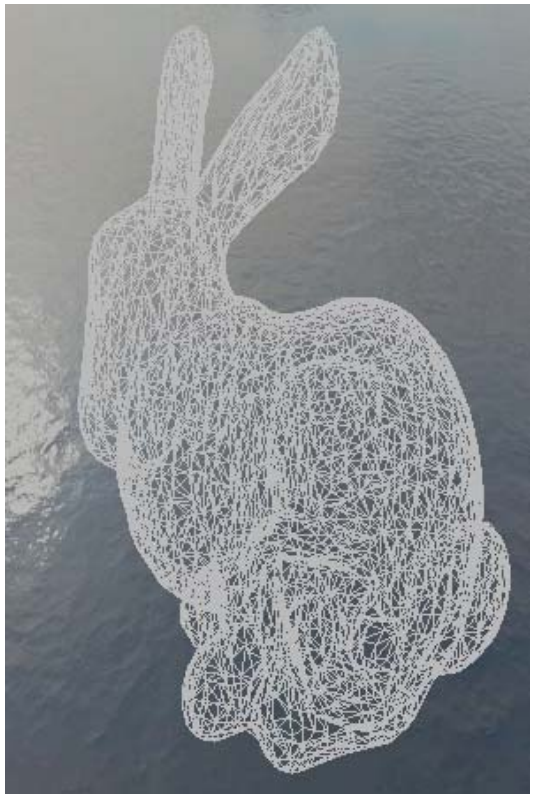

Project
Folder
assets
materials
textures
iron
skybox
scripts
_internal
shaders
models
scenes
assets > materials
skybox
wireframe
iron
light
white
chessbo..
iron_2p..
iron_fo..
gasd
iron2


{
"shader":"ed4d97ad-0eb0-4b1a-85ef-9174eeed4986",
"properties":{
"gAlbedoFactor":{
"__INDEX":20,
"__CONTENT":[
1,
1,
1
]
},
"gAlbedoMap":{
"__INDEX":22,
"__CONTENT":"9e381b28-f55c-4a51-ba3a-237697dc1cac"
},
"gMetalnessFactor":{
"__INDEX":3,
"__CONTENT":1
},
"gMetalnessMap":{
"__INDEX":22,
"__CONTENT":"6ce7e1b5-21f9-47ae-82af-845fbc7a7bcb"
},
"gNormalMap":{
"__INDEX":22,
"__CONTENT":"ab17c5ee-be5b-4c1c-a3e4-c0fd80150c55"
},
"gRoughnessFactor":{
"__INDEX":3,
"__CONTENT":1
},
"gRoughnessMap":{
"__INDEX":22,
"__CONTENT":"2439bdd2-1851-40d6-815d-4b8b3ebd2391"
}
}
}

渲染

Mesh

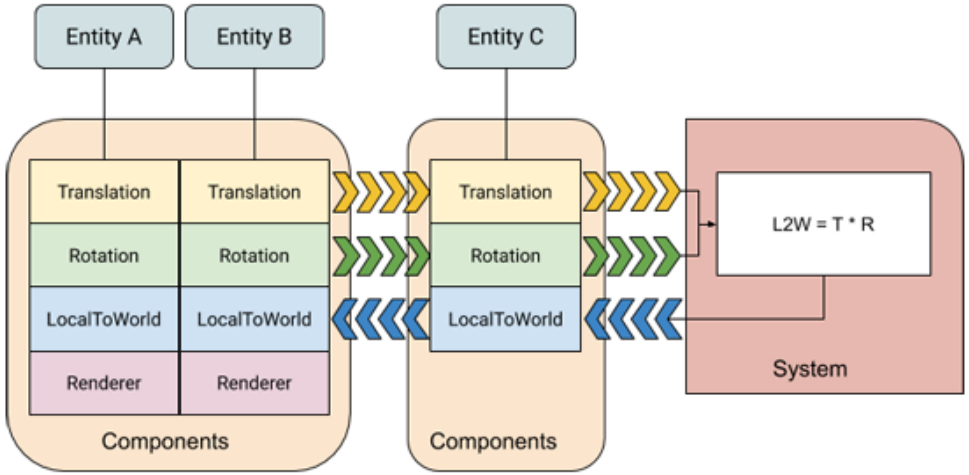

Project
assets
materials
textures
scripts
_internal
shaders
models
scenes
Folder
assets > models
disk
quad
cube
bunny



逻辑

ECS

Entity Component System



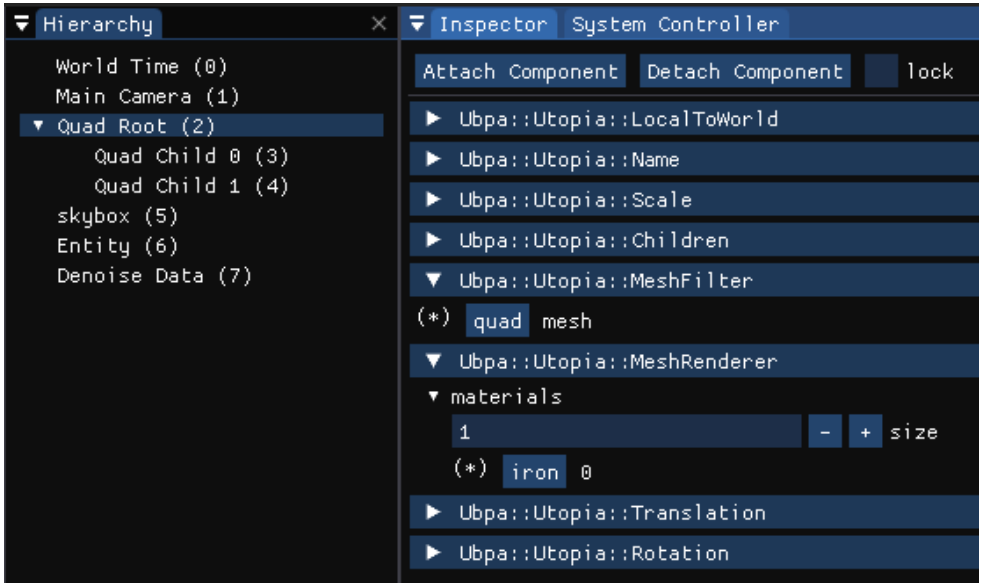
逻辑

ECS

```cpp
#include <UECS/World.h>

using namespace Ubpa::UECS;

struct Position { float val; };
struct Velocity { float val; };

struct MoverSystem {
    static void OnUpdate(Schedule& schedule) {
        schedule.RegisterEntityJob(
            [](const Velocity* v, Position* p) {
                p->val += v->val;
            },
            "Mover"
        );
    }
};

int main() {
    World w;
    w.systemMngr.RegisterAndActivate<MoverSystem>();
    w.entityMngr.Create<Position, Velocity>();
    w.Update();
}
```

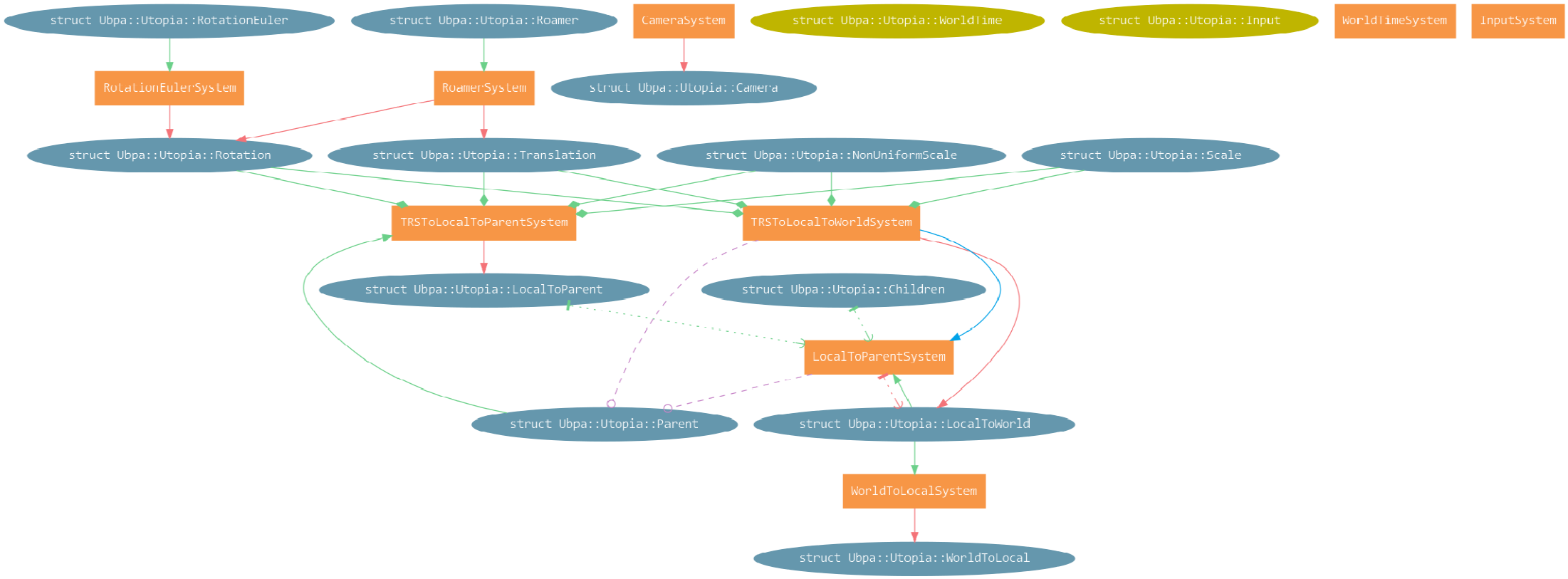
逻辑

ECS

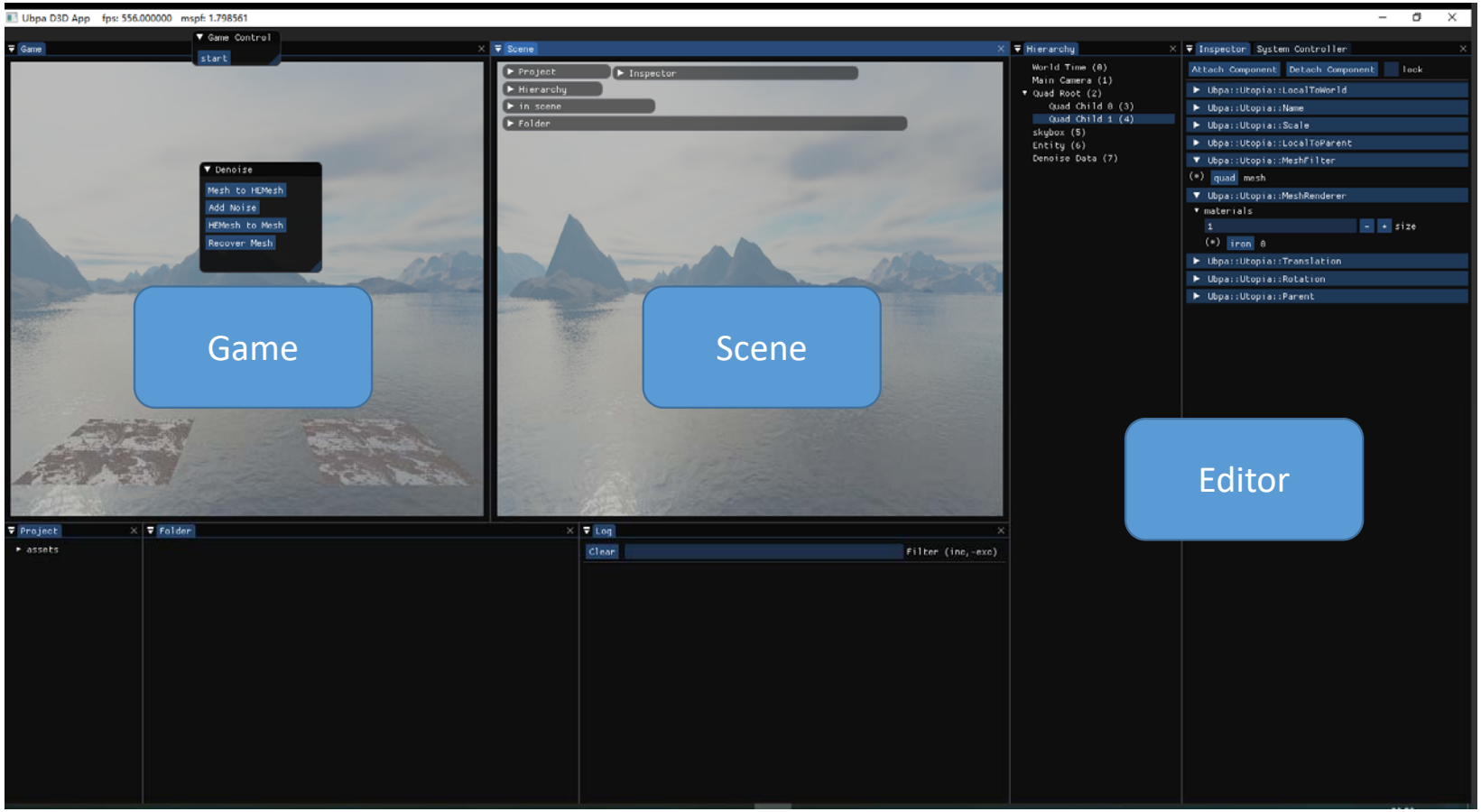

Hierarchy
World Time (0)
Main Camera (1)
Quad Root (2)
Quad Child 0 (3)
Quad Child 1 (4)
skybox (5)
Entity (6)
Denoise Data (7)
Inspector
System Controller
Attach Component
Detach Component
lock
Ubpa::Utopia::LocalToWorld
Ubpa::Utopia::Name
Ubpa::Utopia::Scale
Ubpa::Utopia::Children
Ubpa::Utopia::MeshFilter
(*) quad mesh
Ubpa::Utopia::MeshRenderer
materials
1
-
+
size
(*) iron 0
Ubpa::Utopia::Translation
Ubpa::Utopia::Rotation

逻辑

ECS


struct Ubpa::Utopia::RotationEuler
struct Ubpa::Utopia::Roamer
CameraSystem
struct Ubpa::Utopia::WorldTime
struct Ubpa::Utopia::Input
WorldTimeSystem
InputSystem
RotationEulerSystem
RoamerSystem
struct Ubpa::Utopia::Camera
struct Ubpa::Utopia::Rotation
struct Ubpa::Utopia::Translation
struct Ubpa::Utopia::NonUniformScale
struct Ubpa::Utopia::Scale
TRSToLocalToParentSystem
TRSToLocalToWorldSystem
struct Ubpa::Utopia::LocalToParent
struct Ubpa::Utopia::Children
LocalToParentSystem
struct Ubpa::Utopia::Parent
struct Ubpa::Utopia::LocalToWorld
WorldToLocalSystem
struct Ubpa::Utopia::WorldToLocal

编辑器



使用

（演示）

谢谢大家！