

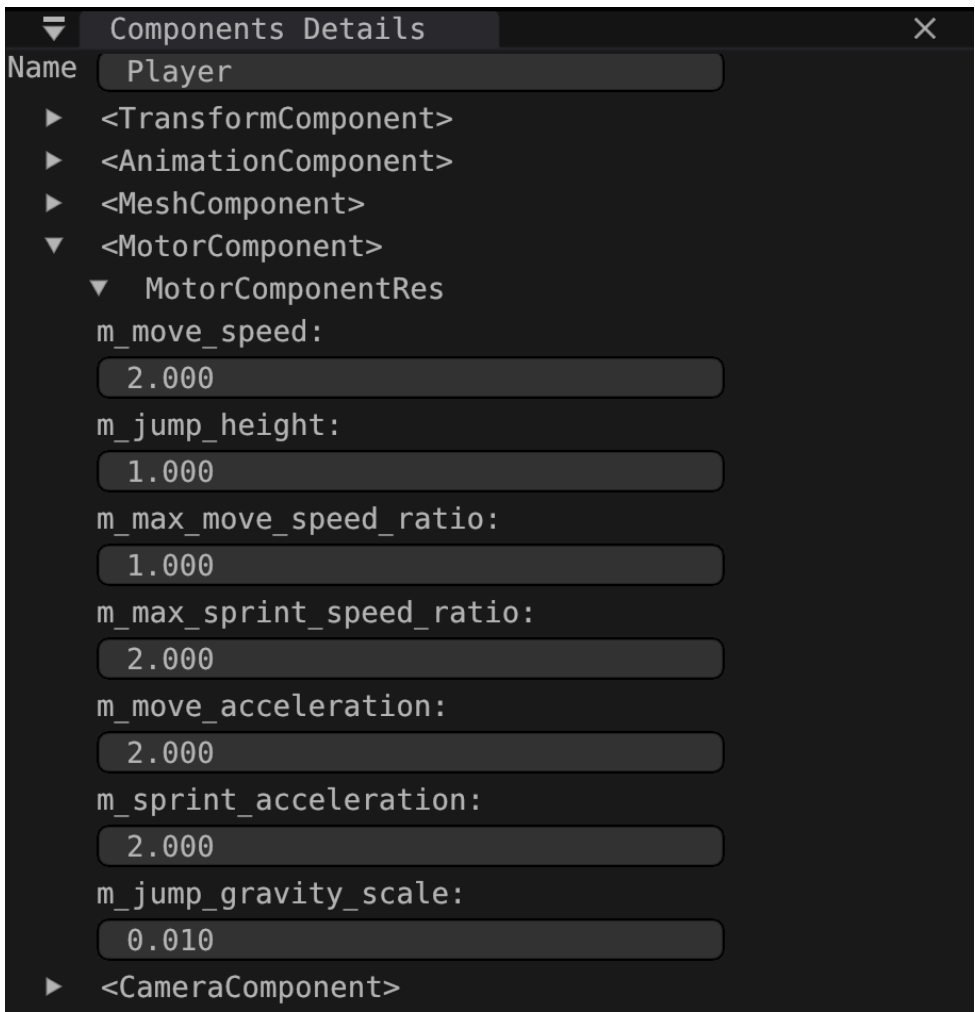
# Homework04 报告

## 属性描述

为运动驱动系统新增gravityScale属性，在组件MotorComponent->MotorComponentRes下可以找到 `m_jump_gravity_scale` 参数项。

该参数类似Unity引擎中的Rigidbody2D.gravityScale接口，用于描述对象受重力影响的程度。在Piccolo引擎物理系统中，重力受PhysicsScene的gravity全局设置影响，但您可以使用gravityScale属性来分别控制应用于每个对象的重力比例。例如，在实现角色飞行时，减少其重力影响通常要比模拟其浮在空中的力更为容易。

## 属性UI



在游戏人物角色的MotorComponent->MotorComponentRes下可以发现新增的 `m_jump_gravity_scale` 属性。

## 代码说明

```
REFLECTION_TYPE(MotorComponentRes)
CLASS(MotorComponentRes, Fields)
{
    REFLECTION_BODY(MotorComponentRes);
}
```

```

public:
    MotorComponentRes() = default;
    ~MotorComponentRes();

    float m_move_speed { 0.f};
    float m_jump_height {0.f};
    float m_max_move_speed_ratio { 0.f};
    float m_max_sprint_speed_ratio { 0.f};
    float m_move_acceleration {0.f};
    float m_sprint_acceleration { 0.f};
    /**新增：定义跳跃加速度受重力影响的程度
    float m_jump_gravity_scale { 1.f};

    Reflection::ReflectionPtr<ControllerConfig> m_controller_config;
};

```

在engine/source/runtime/resource/res\_type/components/motor.h 文件的 MotorComponentRes 类中添加了 `m_jump_gravity_scale` 浮点类型变量，用于定义重力影响因子。由于在Motor系统中已经搭好了反射框架，用的是Fields标签，表明类型中除开特殊标记的属性外均需反射，所以添加参数只需要一行代码不需要额外操作，非常简单。

```

void MotorComponent::calculatedDesiredVerticalMoveSpeed(unsigned int command, float
delta_time)
{
    std::shared_ptr<PhysicsScene> physics_scene =
        g_runtime_global_context.m_world_manager-
>getCurrentActivePhysicsScene().lock();
    ASSERT(physics_scene);

    if (m_motor_res.m_jump_height == 0.f)
        return;

    const float gravity = physics_scene->getGravity().length();
    //获取新增属性值
    float gravity_scale = m_motor_res.m_jump_gravity_scale;

    if (m_jump_state == JumpState::idle)
    {
        if ((unsigned int)GameCommand::jump & command)
        {
            m_jump_state = JumpState::rising;
            m_vertical_move_speed = Math::sqrt(m_motor_res.m_jump_height *
2 * gravity * gravity_scale);
            m_jump_horizontal_speed_ratio = m_move_speed_ratio;
        }
        else
        {
            m_vertical_move_speed = 0.f;

```

```

    }
}
else if (m_jump_state == JumpState::rising || m_jump_state ==
JumpState::falling)
{
    m_vertical_move_speed -= gravity * gravity_scale * delta_time;
    if (m_vertical_move_speed <= 0.f)
    {
        m_jump_state = JumpState::falling;
    }
}
}
}

```

在engine/source/runtime/function/framework/component/motor/ motor\_component.cpp 中的 MotorCompont::calculatedDesiredVerticalMoveSpeed 方法内含有计算角色跳跃过程中垂直移动的部分，仅需将获得的 `gravity_scale` 值倍乘于原始的位移计算式中即可实现重力影响因子对角色垂直移动加速度的改变。

## 小结

因为这次是开放式作业，简单起来还是蛮简单的。本来想针对相机系统增加视角移动时的阻尼特性，但是发现引擎中的相机组件UI怎么都出不来，后来在群里咨询后得知是指针暂不支持，就临时fallback到改motor系统了😂。