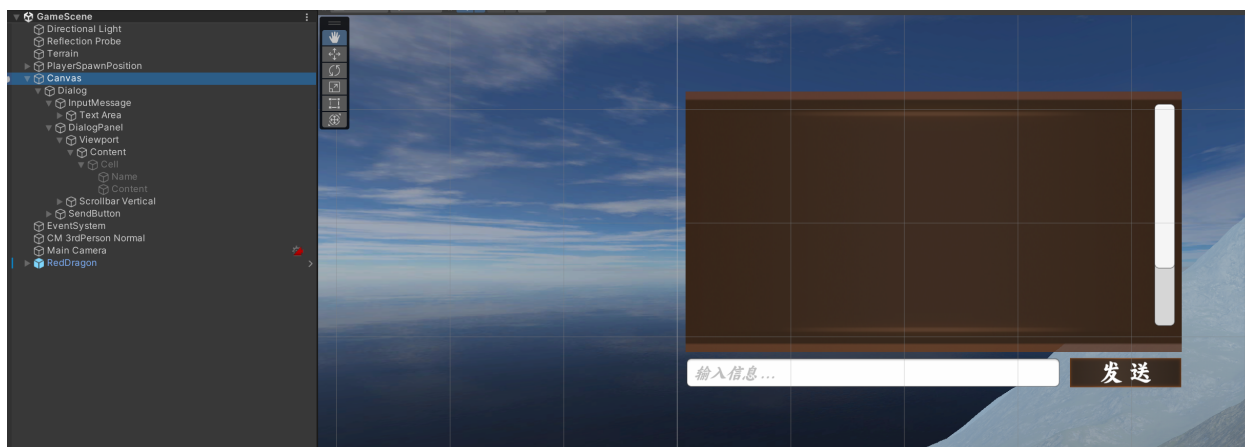




# 游戏场景

说明：游戏场景主要内容有对话框、地形包、玩家，需要注意多人联机模式当位置、动画改变时都需要同步数据



核心脚本：

1. GameController.cs：游戏场景控制器
2. DialogListCellController.cs：对话框Cell控制器
3. PlayerInit.cs：玩家相关组件设定脚本，例如设置摄像机
4. PlayerMove.cs：玩家移动控制脚本
5. PlayerSync.cs：玩家数据同步脚本
6. OwnerAnimator：动画持有者

## 游戏场景控制器 GameController.cs

说明：GameController挂载在GameScene场景中一个空物体上，设置该空物体的原因是将该物体位置坐标一个初始点，在此点周围生成玩家模型

重要资源：

```
public static GameController Instance;
[SerializeField] private CinemachineVirtualCamera cameraController; // 相机控制器
[SerializeField] private Transform canvers; // 对话框父物体
private TMP_InputField inputMessage; // 输入信息
private RectTransform DialogContent; // 对话框Content
private GameObject DialogContentCell; // 对话框中的单条消息
private Button sendButton; // 发送消息按钮
```

### OnNetworkSpawn

1. 从当前canvas中获得输入框、聊天框content、聊天框cell、消息发送按钮组件，并将<发送消息按钮点击事件>绑定到消息发送按钮组件上
2. 调用base.OnNetworkSpawn

### 发送消息按钮点击事件

1. 如果发送信息为空则返回
2. 根据LocalClientId得到当前机器上玩家id，从GameManager中拿到对应的玩家数据 PlayerInfoData
3. 根据当前玩家姓名和输入框内容在本地聊天框内<添加一个聊天框内容cell>并显示
4. 如果当前是服务端，则<服务端向所有客户端发送消息 ClientRpc>
5. 如果当前是客户端，则<客户端向服务端发送消息 ServerRpc>

### 服务端向所有客户端发送消息 ClientRpc

1. 查看当前是否为Server或者LocalClientId==PlayerInfoData.playerId，防止自己给自己发送消息
2. 根据服务端传来的玩家姓名和玩家发送内容在本地<添加一个聊天框内容cell>并显示

### 客户端向服务端发送消息 ServerRpc

1. 在本地聊天框内<添加一个聊天框内容cell>并显示
2. <服务端向所有客户端发送消息 ClientRpc>

### 添加一个聊天框内容cell

1. 实例化除一个对话框Cell，设置Parent为DialogContent
2. 得到DialogListCellController组件并进行初始化，设置玩家姓名和聊天内容
3. 将对话框Cell显示出来

### 设置相机跟随目标

1. 设置CinemachineVirtualCamera的Fllow为传入的target，即玩家

### 随机生成一个玩家出生坐标

1. 获取当前游戏对象坐标，在前方和右方范围内随机生成一个坐标

---

## 玩家相关组件设定脚本 PlayerInit.cs

说明：该脚本继承自NetworkBehaviour，挂载到Player预制体上

### OnNetworkSpawn

1. 将OnStartGame方法添加到UnityEvent GameManager.OnStartGame中，在所有玩家加载完事件后调用该方法

### OnStartGame

1. 根据拥有当前游戏对象的clientId `OwnerClientId` 找到玩家数据，
2. 声明body变量控制男女性角色
  - a. 如果当前玩家性别为男性，则从当前游戏对象上获取Male游戏对象（男性角色），并获取组件PlayerSync后设置男性角色<切换男女玩家预制体>

- b. 如果当前玩家性别为女性则从当前游戏对象上获取Female游戏对象（女性角色），并获取组件PlayerSync后设置女性角色<切换男女玩家预制体>
- c. 关闭body的rigibody的 `isKinematic`
3. 打开当前PlayerSync组件，在此处打开的目的是防止该脚本被提前调用
4. 关闭未选择性别的预制体
5. 如果当前是本地玩家IsLocalPlayer，设置相机跟随当前玩家模型body，打开PlayerMove脚本
6. 最后<设置玩家随机出生地点>

## 玩家移动控制脚本 PlayerMove.cs

说明：该脚本从资源包中直接拉出使用，脚本中有部分代码不理解，因此暂不添加到文档中

## 玩家同步脚本 PlayerSync.cs

说明：该类继承自NetworkBehaviour，需要注意在网络中同步的变量要声明为NetworkVariable

重要资源：

```
private NetworkVariable<Vector3> syncPosition = new NetworkVariable<Vector3>(); // 玩家位置同步数据
private NetworkVariable<Quaternion> syncRotation = new NetworkVariable<Quaternion>(); // 玩家旋转同步数据
private Transform syncTransform; // 需要同步数据对象的transform
```

### 切换男女玩家预制体

1. 如果当前传入玩家性别为男性，设置syncTransform为当前Male游戏对象的transform
2. 如果当前传入玩家性别为女性，设置syncTransform为当前Female游戏对象的transform

### Update

1. 如果当前是本地玩家，则需要不断<上传玩家位置信息和旋转信息>

### FixedUpdate

1. 如果当前不是本地玩家则需要不断<同步玩家位置>

### 上传玩家位置信息和旋转信息

1. 如果当前是服务端/Host，则更新当前syncPosition、syncRotation
2. 如果当前是客户端，则需要调用<客户端要求服务端更新数据>方法

### 客户端要求服务端更新数据

1. 根据传入的玩家位置、旋转数据更新当前syncPosition、syncRotation，即将client玩家位置推送到服务器上

### 同步玩家位置

1. 将syncTransform的position、rotation设置为syncPosition和syncRotation的值

---

## 对话框Cell控制器 DialogListCellController.cs

重要资源：

```
private TMP_Text playerName;        // 对话框玩家姓名
private TMP_Text dialogContent;     // 对话框玩家说话内容
```

初始化方法

1. 根据传入的玩家姓名和玩家对话内容设置当前Cell中TMP\_Text中的.text变量

---

## 动画持有者 OwnerAnimator

说明：该脚本继承自 `NetworkAnimator`，需要重写 `OnIsServerAuthoritative` 方法后才能正常同步动画，原因未知