代码说明

1. 进入游戏
2. 进入选关

**MapScene 场景**

Mapcreater 对象挂的 DataLoader脚本

调用MapButtonDrawer（）

初始化地图上的所有关卡

Player.cs 中PlayerUtils Load()

根据存储在本地的key= KEY\_DATA = "DATA" False,0,0,0格式解析并生成Player()对象

对应：是否解锁，星星数量，最高分，背景id

string data = *PlayerPrefs*.*GetString*(KEY\_DATA, "");

dataSplit = data.*Split*(',');

for (int i = 0; i < 297; i++)

{

p = new Player();

p.Level = i + 1;

p.Name = (i + 1).ToString();

p.Locked = bool.*Parse*(dataSplit[i \* 4]);

p.Stars = int.*Parse*(dataSplit[i \* 4 + 1]);

p.HightScore = int.*Parse*(dataSplit[i \* 4 + 2]);

p.Background = int.*Parse*(dataSplit[i \* 4 + 3]);

list.*Add*(p);

}

调用 insmap(*Vector3* pos, int index) 初始化map 预设 到场景上面挂着Map.cs

1. 选择关卡

点击一个关卡后调用 CameraMovement.cs 中PopUpShow(Player \_map)方法，显示信息

点击Play ButtonActionController.cs 调用ArcadeScene(Player player)

赋值给PlayerInfo静态变量 MODE,MapPlayer 跳转到**PlayScene**

1. 开始游戏

GameController对象中GameController.cs *IEnumerator* Start()方法开始

GribManager.cs中 MapReader()

Assets/Resources/Maps/ 路径下的txt文件 生成宝石下面的地图

GribCreate(Map); 生成宝石地下的地图

BorderCreate(Map); 生成每块地图周围的边框

EffectCrash(Map); 生成地图特效

JewelSpawner.cs RJewelInstantiate()方法 实例化宝石

randomjewel（x,y） 随机宝石类型，当前类型与 x-1，x-2 不一样 并且与y-1 y-2不一样 每次随机100次

while (!Supporter.sp.isNoMoreMove()) 知道找不到可以相同的

Supporter.cs 支持代码 update中没5秒查找一遍

isNoMoreMove() 查找是否都三个消除 查找可以消除的并且播放suggestion提示动画

1. 开始游戏

游戏过程

GameController.cs void Update() 更新

JewelSelecter() 鼠标点击后显示红色框Selector

RuleChecker(Pointer, Selected); 检测交换两个宝石位置是否会消除

SwapJewelPosition 交换两个宝石

ListProcess(list1, obj2, obj1, obj1.*GetComponent*<JewelObj>().jewel.JewelType);

不同相同宝石 不同消除效果

dropjewel() 显示spawnController

SpawnController.cs Update()