游戏《超级饼里奥》开题报告

本游戏为冒险闯关类游戏，玩家需要控制游戏主人公饼里奥在地图中进行运动，避开尖刺进行存档或者击败怪物，推箱子到指定位置,探索地牢最终通关。

1. 选题动机

本学期的《软件构造基础》课程学习了c#语言的语法和一些用法。为了加深对c#语言的理解以及在实践中锻炼编写c#代码的能力，我们小组选择将c#与Unity结合设计一款马里奥游戏。这是一款经典、大受玩家喜爱的游戏，基于从前玩这款游戏的体验和了解，以及加上小组成员的新的想法，我们计划设计出一款命名为《超级饼里奥》的游戏。

二、功能设计

《超级饼里奥》游戏总共设计了多类文件夹，分别是：玩家类Player、敌人部分Enemy、小怪物类Monster、大怪物类Boss、、技能模块AttackSkill、UI界面模块和视图类Camera，音效管理SoundManager模块

同时还另外设计两个界面scene，即开始游戏界面和游戏结束界面。

1. Player

玩家左右移动，跳跃的基础功能分别由Move、Reverse、Jump函数实现，检测角色是否触及地面，以控制玩家在地面行走并且跳跃后能够回到地面。ontrigger函数检测角色是否攻击敌人，若成功攻击敌人，触发monster类或Boss类的减血。

用鼠标来实现射击或攻击，此时实现射击功能的Shot函数可以被玩家触发。

用碰撞体和触发器检测玩家是否被怪物，障碍物碰撞或被boss的技能击中，如 果被击中则调用Die函数触发角色死亡事件或者减少血量。

1. Enemy

敌人的基类Enemy中实现了怪物的普通平移、左右转向、检测障碍物、自动改变行进方向的基础功能，分别由函数Move、Reverse、CheckBlock、ChangeMoveDirection函数实现。

1. Monster

Monster是普通小怪，在地图内随机产生。除了从基类继承的基础移动功能，Monster具有被踩踏和被射击死亡的效果，分别通过GetHit和GetShot函数触发Die函数。

1. Boss

Boss是游戏的特殊怪物，在地图特点位置唯一生成。通过函数Fly和CheckIsGround实现飞行并且在一定时间内回到地面的功能。Boss通过Attack函数对角色发动攻击。GetHit函数监控Boss被击中的事件，并且减低boss血量参数blood，当boss血量参数下降为0则触发死亡效果die。

1. AttackSkill
2. 技能分为子弹或剑导弹等武器
3. 每种武器具有一定量的伤害
4. 触碰到怪物或方块可以触发对应的功能

5、UI

1）UI界面显示菜单并包含多个按键，开始游戏可以正常开始游戏，加载游戏可以加载玩家的存档，退出游戏可以直接结束游戏。在游戏中按下tab键可以显示自己的游戏时长和十万次数

6、Camera

1）Camera设计左边界、右边界和界面比例，跟随玩家移动而转换视角，显示当前玩家所在地图位置附近的界面。

三、技术难题

（一）Unity原理

（1）Unity开发的本质是在Unity引擎的基础上利用反射和引擎提供的各种功能进行的拓展开发。

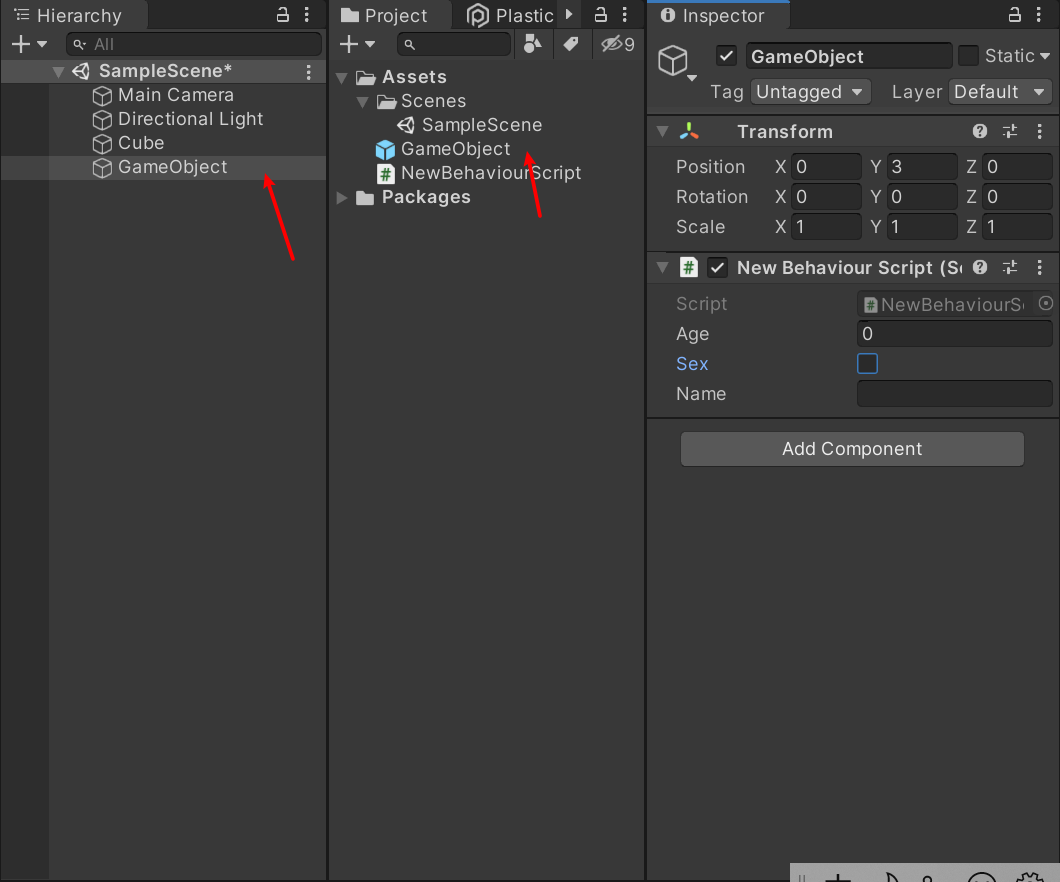
Unity相当于一个成熟游戏开发平台，做好了很多底层工作，我们只需遵循一些规则和学会调用一些接口来实现游戏界面的开发。

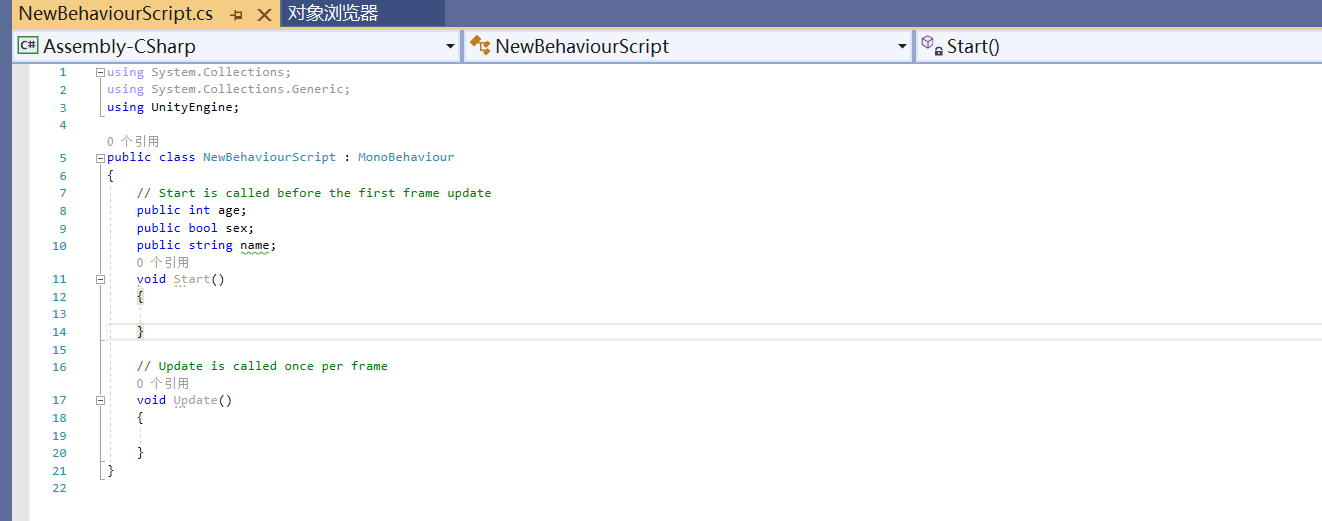
程序在运行过程中，可以查看其它程序集或者自身的元数据（类，函数，变量，对象等），一个运行的程序查看本身或者其它程序的元数据的行为称为反射。在程序运行时，通过反射可以得到其它程序集或者自己程序集中代码的各种信息，比如类、函数、变量、对象等等，可以实例化它们，执行它们，操作它们。

Unity场景的本质是一个个特殊格式的配置文件.unity文件（保存了场景中所有事物的信息，格式Unity创建场景就是读取.unity里的信息来创建。

Unity场景中看到的事物的本质是GameObject对象，Transform表示GameObject对象的位置信息。可以为GameObject关联各种脚本（c#脚本）。

在Assets里创建脚本文件(c#脚本）并于场景中的GameObject关联：





在Inspector里会识别脚本文件并生成可编辑的相关属性。

（2）游戏的本质是一个死循环，每一次循环处理游戏逻辑就会更新一次画面（加载一帧），之所能看到画面在动是因为切换画面的速度达到一定时人眼就认为画面是流畅的，一帧就是执行一次循环。

Unity底层已经做好了死循环，我们只需要学习Unity的生命周期函数，利用它做好的规则来执行我们的游戏逻辑就行了。脚本中的生命周期函数就是依赖该脚本的GameObject对象从出生到消亡整个生命周期中通过反射自动调用的一些特殊函数。Unity帮助我们记录了一个GameObject对象依附了哪些脚本，会自动得到这些对象，通过反射去执行一些固定名字的函数。Unity加载每一帧时会自动调用脚本的生命周期函数。

生命周期函数：

（1）Awake()：当对象被创建时调用（创建即关联GameObject)，类似于构造函数

（2）OnEnable():依附的GameObject对象每次激活时调用

（3）Start():只调用一次，在Awake()之后执行，在第一次帧更新前执行

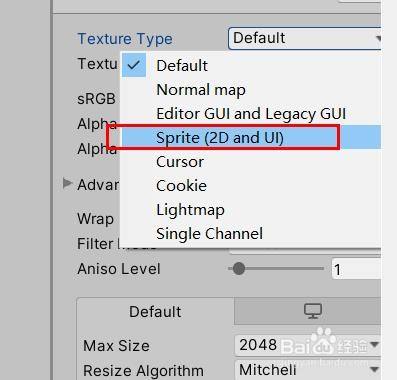
（4）FixeUpdate():主要用于物理更新，每一帧都执行，但两次执行的间隔时间可以设置

菜单Edit->Project Settings->Time，执行在LateUpdate()之后

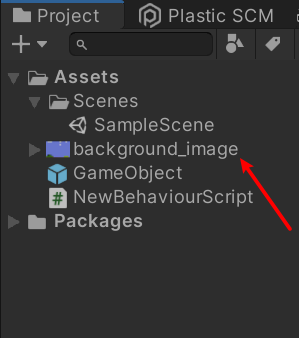
（二）、实验所需资源及利用方法

1.为游戏设置背景图片

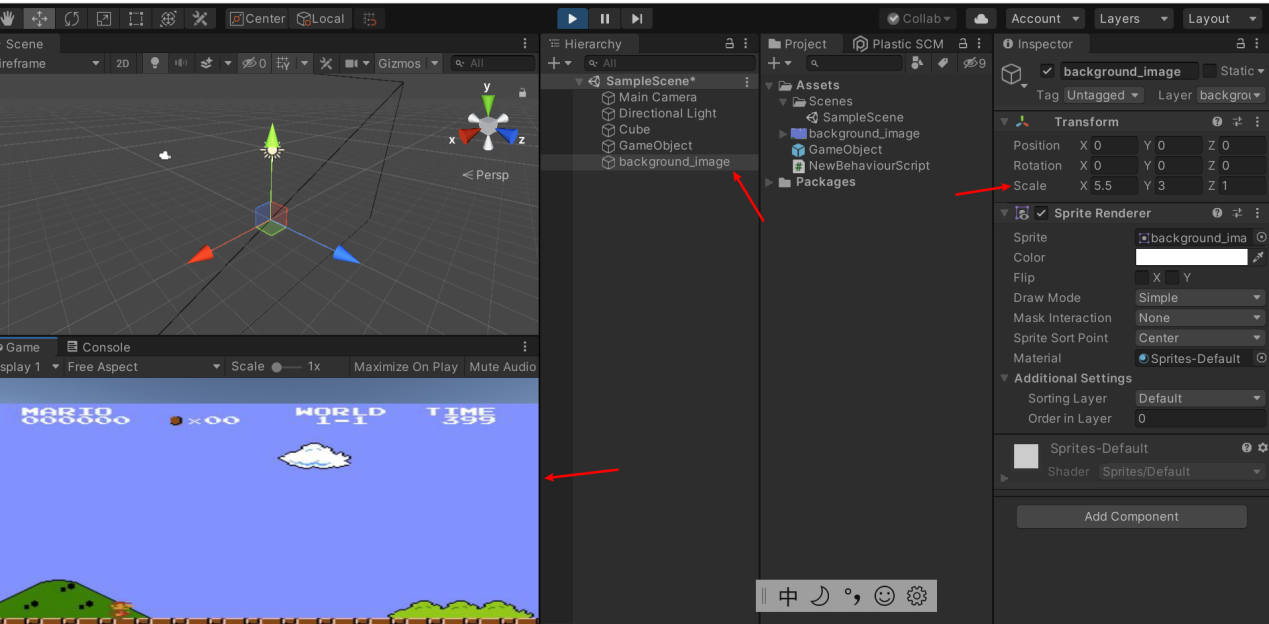
1. 首先找好需要用到的素材图片
2. 然后将其导入Project面板（或者直接进行拖拉）
3. 选中当前导入的图片，将其Texturn Type更改为 Sprit(2D and UI）



1. 然后点击右下角的Apply，之后我们可以发现图片前多了展开项，制作完成



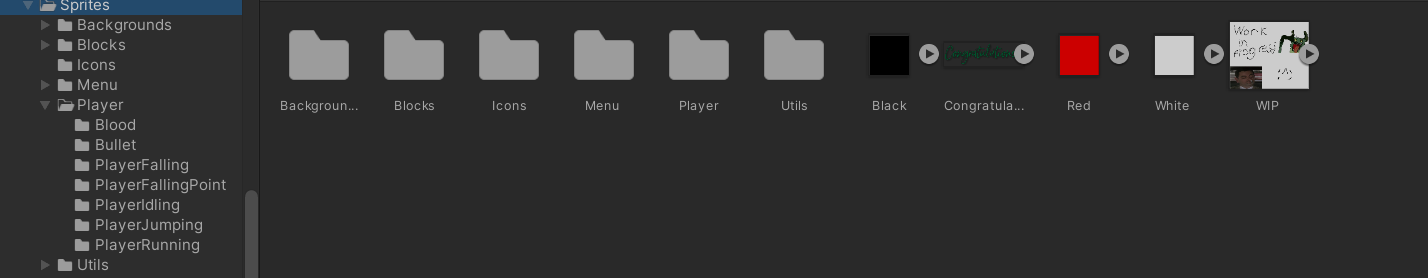
1. 将该图片拖拽到Hierarchy窗口生成GameObject对象，选中后即可在Inspector窗口里编辑它



1. 游戏所需物体

马里奥（玩家）、蘑菇怪、乌龟，地面（包括悬浮在空中的）、管道

为3d物体制作外观(利用已有资源）



1. 将游戏场景中的各个物体与c#脚本关联，实现与玩家之间的交互。根据游戏的逻辑实现动画界面。
2. 任务分配