暑培 Unity

by ysq

1. Unity介绍

什么是游戏引擎?

游戏引擎（Game Engine）是指一些已编写好的可编辑电脑游戏系统，或者一些交互式实时图像应用程序的核心组件。这些系统为游戏设计者提供各种编写游戏所需的各种工具，其目的在于让游戏设计者能容易和快速地做出游戏程式而不用由零开始。

什么是Unity?

Unity是实时3D互动内容创作和运营平台。包括游戏开发、美术、建筑、汽车设计、影视在内的所有创作者，借助Unity将创意变成现实。Unity平台提供一整套完善的软件解决方案，可用于创作、运营和变现任何实时互动的2D和3D内容，支持平台包括手机、平板电脑、PC、游戏主机、增强现实和虚拟现实设备。

为什么选择Unity?

- 优秀作品
- 学习成本低，教程多，文档完善
- 好就业

2. Unity安装与配置

版本

- Unity Editor：(LTS)
- Unity Hub
- 正式开发前，团队成员要统一开发版本，并一直用到结束。

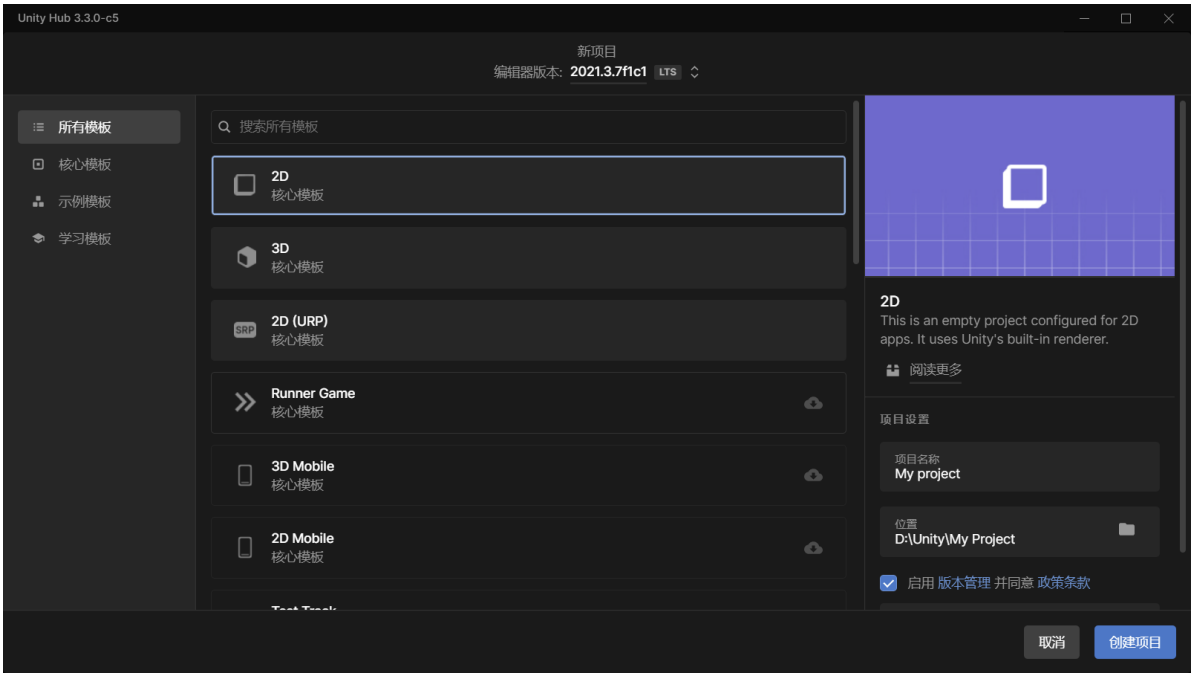
开发工具

- Visual Studio：代码编辑工具。用于编写游戏脚本。
- Unity Hub：Unity官网下载。用于管理现有Unity项目，Unity Editor等，今后项目创建、打开现有项目等均在Unity Hub中进行。
- Unity Editor：在Unity Hub->安装->安装编辑器中下载。用于进行UI设计。
- 其他工具

3. Unity界面介绍

Unity Hub界面

创建新项目

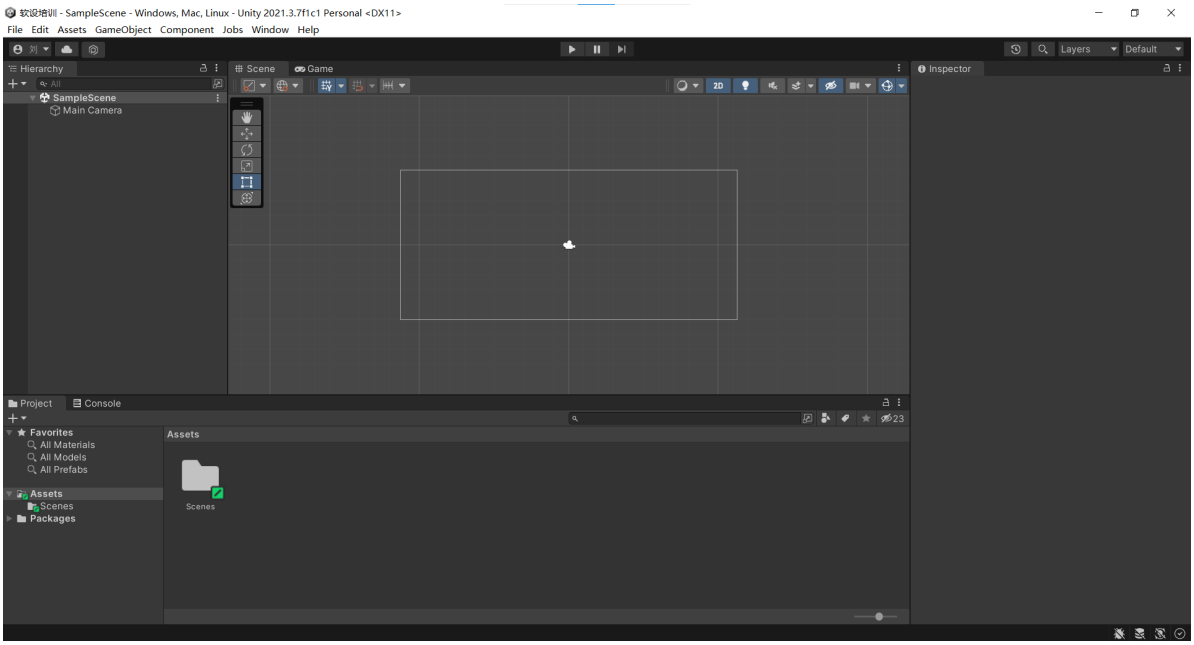


2D开发和3D开发区别不大，3D只是在2D基础上增加一个维度。下面以2D为例进行介绍，3D开发过程完全类似。

新建Unity项目所需时间较长，学习和练习过程可以使用一个固定的项目完成。

Unity Editor界面

首次创建项目后可自动进入Unity Editor



Unity Editor界面主要包括：

- 导航栏

  File Edit Assets GameObject Component Jobs Window Help

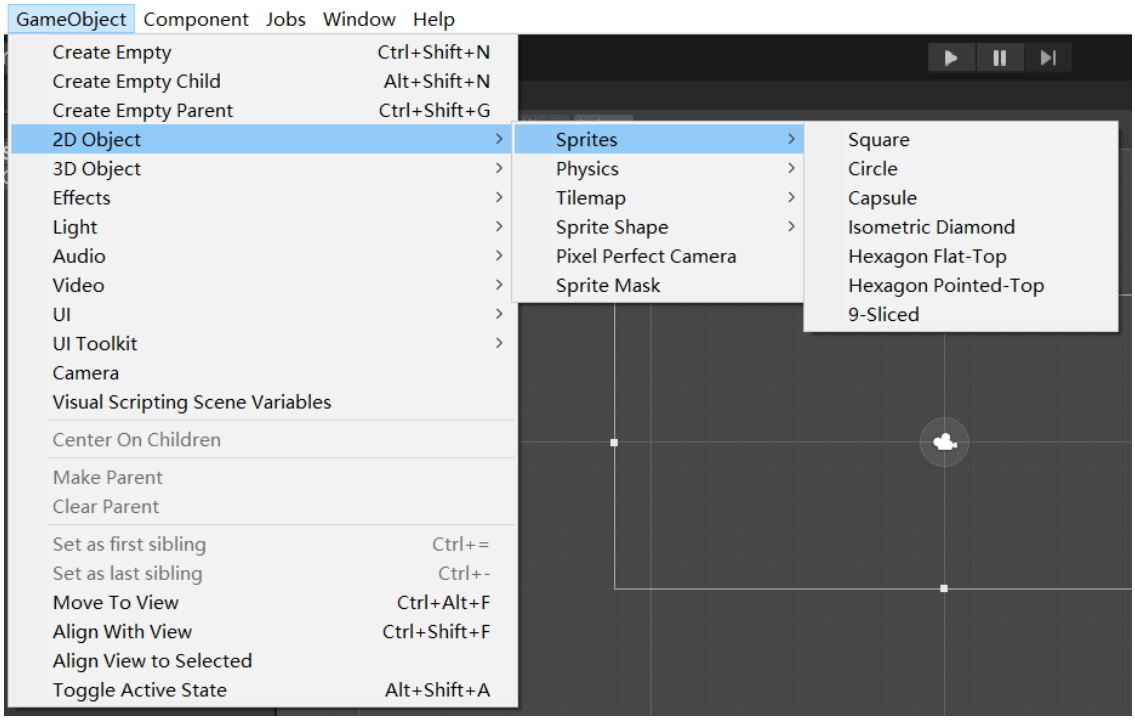
- Hierarchy面板：用于存放创建的游戏对象，并展示对象之间的父子关系、预制体等。
- 资源面板
  - 用于存放项目所需的一切图片、音频、脚本、预制体、材质（Materials）等。
  - 与实际资源的存储目录一致
  - 注意养成归类整理的习惯
- Inspector面板：用于管理游戏对象所挂载组件（Components）的属性

- Scene和Game界面
    - Scene是实际创建的游戏场景，Game界面显示最终呈现的游戏画面
    - 点击运行按钮开始游戏调试，再次点击时调试终止
    - 点击停止按钮暂停调试，再次点击时游戏继续

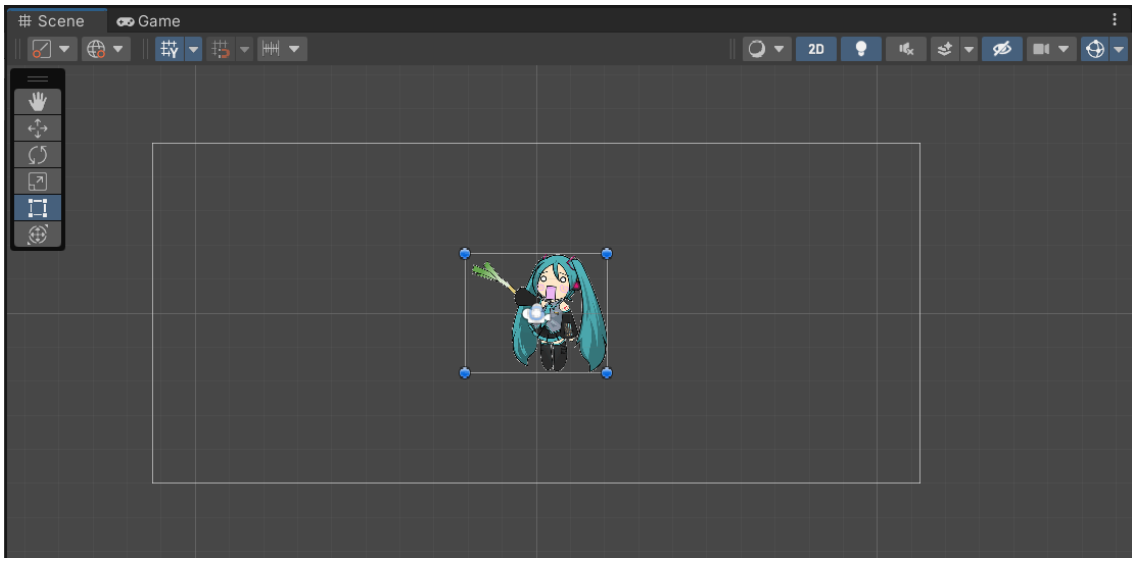
4. 对象的添加与调整

静止对象

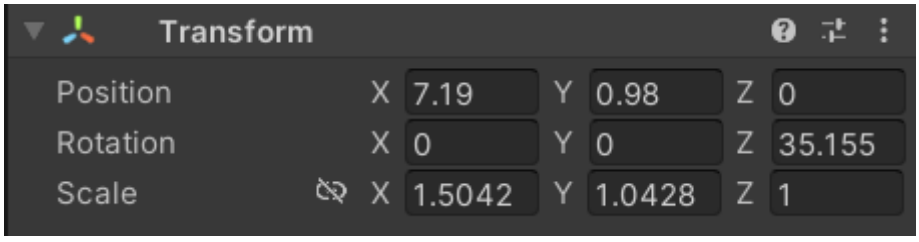
- GameObject中有简单的图形对象



- 将资源文件中的图片拖拽至Hierarchy/Scene下添加对象



尝试左侧移动、旋转、缩放等工具，看看游戏对象发生了怎样的变化，同时关注右侧transform组件下方的数据变化。



对象之间的父子关系

- 在Hierarchy中拖动对象创建父子关系
- 父对象变换（平移、伸缩、旋转）时，子对象跟随父对象变换
- 子对象可独立于父对象变换

对象的位置坐标

- 对象的坐标由一个三维向量给定
- 世界坐标：对象在空间中的绝对坐标
- 本地坐标：对象相对于父节点的坐标
- 2D中z分量不影响视觉位置，但可能影响对象之间的遮挡关系

Camera对象

- Camera是一个特殊的对象，用于对Scene进行拍摄（2D中是投影），最终的游戏界面是Camera拍摄的结果
- 通过Camera组件，可以修改最终游戏界面大小等
- 在Game界面中调整画面长宽比

5. 脚本

简介

- Unity脚本用于编写游戏的基本逻辑、控制对象的移动等操作。
- Unity脚本用C#语言编写。C#语言的语法与C、C++类似，特别是Unity脚本编写时，我们只需要填充函数体部分，因此比较好入门。
- 脚本需要挂载在对象下才能够运行，可以理解为该脚本控制某一特定对象的操作。

环境配置

- VS：工具->获取工具和功能->VS Installer->使用Unity的游戏开发
- Unity：Edit->Preferences->External Script Editor，选择VS

创建脚本

- Create->C# Script

脚本内容

- 每一个（挂载在对象上的）脚本定义一个类，该类继承自MonoBehaviour类，脚本文件名需与该类名完全相同
- 在游戏开始时调用一次`Start`函数
- 在游戏的每一帧中调用一次`Update`函数
- 脚本中还可以定义其他字段，这些字段既可以在脚本中设定，也可以在Editor中修改

实例一：控制台输出“Hello, World!”

- 在`Start`函数中写入

```
Debug.Log("Hello, World!");  //控制台输出各种类型的数据
```

实例二：对象的移动

- 原理：每一帧更新对象的位置
- 在`Update`函数中写入

```
transform.Translate(0.01f, 0, 0, Space.Self);
```

- 可以发现，对象运动速度并不恒定，这是由于程序运行过程中每一帧的时长不等造成的，可以采取下面的办法使运动匀速

```
var velocity = new Vector3(3.0f, 0, 0);  //Vector3是已经定义好的类，包含三个浮点数
transform.Translate(Time.deltaTime * velocity, Space.Self);
```

- `Translate`函数的最后一个参数是参考系，包括`Space.Self`和`Space.World`两种。思考：何时二者产生不同的结果？

实例三：动态创建父子关系

- 对象之间的父子关系可以通过`SetParent`函数动态创建
- 在子对象的`Start`函数中写入

```
var square = GameObject.Find("Square");  //Find函数用于寻找特定名称的对象，若存在，返回GameObject类型
transform.SetParent(square.transform);  //将对象square设定为该对象的父节点
```

为父对象挂载控制移动的脚本，运行后父子对象应同时运动

6. 组件

什么是组件?

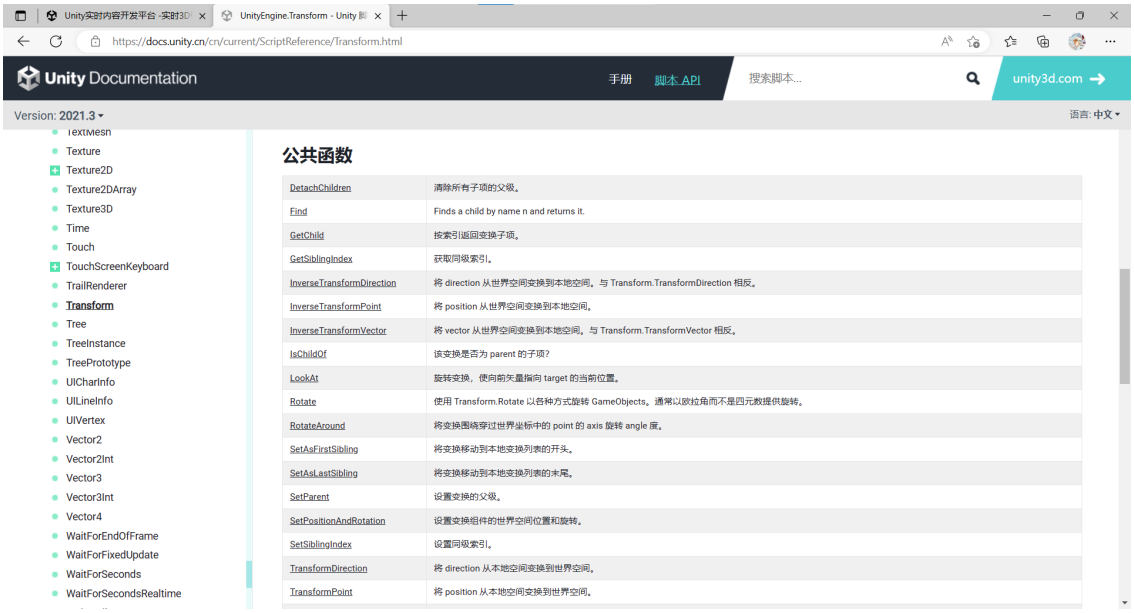
- 【金句来源于网络】脚本即组件，一切皆组件，方法是组成组件的零件，脚本相当于自定义的组件，通过方法的运用和编码组成各种功能的组件，组件是unity的基本单元。一切功能，一切显示皆由组件实现。不论是系统自带功能，还是自定义的脚本，都是以方法，字段属性，类等语言知识为基本元素组合而成，形成种种功能与组件，根据目的和需求使用基本元素和单元进行拼装组合，不论是系统内置还是编写脚本。组件的概念与方便之处，千变万化，有本无形。

常用组件

- transform：控制对象的位置、旋转、伸缩
- Sprite Renderer：渲染器，控制对象的显示
- Script：脚本，自定义组件，控制整个游戏的逻辑
- ……

组件的使用

- 组件可以通过Editor添加和编辑。
- 组件可以在脚本中通过`GetComponent<>`函数获取
- 组件可以通过脚本调用：每一个组件对应一个类，每一个组件类都有丰富的方法和字段，可以在脚本中使用。
- 更多详细信息可以在Unity官方文档的“脚本API”部分查阅



7. 物理系统例：碰撞与触发

物理系统

【Unity手册】Unity 可帮助您在项目中模拟物理系统，以确保对象正确加速并对碰撞、重力和各种其他力做出响应。Unity 提供了以下不同的物理引擎实现方案，您可以根据自己的项目需求选用：3D、2D、面向对象或面向数据。

碰撞

- 制造一个刚体
    1. 为Circle对象添加RigidBody组件，Body Type选择为Dynamic
        - Dynamic：普通刚体，有质量，有速度
        - Static：静态刚体，质量无穷大，无速度（如地面）
        - Kinematic：运动学刚体，无质量，用于碰撞检测
    2. 为Circle对象设计材质
        - Create->2D->Physics Material 2D，创建新材质
        - 设定bounciness为0.9
- 制造一个碰撞体（collider）：为小球添加Circle Collider 2D组件
- 类似地，将地面设置为碰撞体，Body Type选择为Static，此时运行，小球落地后反弹
- 碰撞事件函数：包括`OnCollisionEnter2D`（碰撞开始时执行）、`OnCollisionExit2D`（碰撞结束时执行）、`OnCollisionStay`（碰撞过程每帧执行）

```
private void OnCollisionEnter2D(Collision2D collision)  //Collision2D类型包含碰撞信息
{
    Debug.Log(name + " collided with " + collision.collider.name);
}
```

触发

- 制造一个触发器（Trigger）：在碰撞体基础上，勾选Is Trigger选项
- 当两个碰撞体中有一个以上的触发器时，对象之间不会发生碰撞，而会互相穿过
- 触发事件函数：包括`OnTriggerEnter2D`（接触开始时执行）、`OnTriggerExit2D`（接触结束时执行）、`OnTriggerStay`（接触过程每帧执行）

```
private void OnTriggerStay2D(Collider2D collision)
{
    var rb = GetComponent<Rigidbody2D>();
    rb.AddForce(new Vector3(20.0f, 0, 0));
}
```

碰撞与触发规则

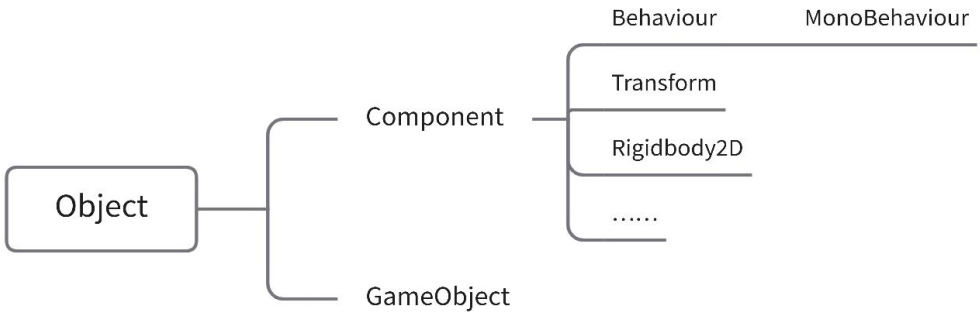
参考http://t.csdn.cn/sJ123

8. 写在最后

作为一个最简单的入门教程，本篇基本只是让大家认识unity，下载安装unity并且了解unity的基本界面和用法。作为一个工具型软件，最好的学习方法永远是实战，推荐大家直接教程开发一个小项目，体会到游戏开发的乐趣。

下面是一些实用的文档和教程，希望对大家有所帮助

- Unity官方文档
- Unity旧版教程（即将下线）哔哩哔哩bilibili
- Unity2022新手入门教程超细节*100*集课程哔哩哔哩_bilibili
- Unity脚本部分类的继承关系



致谢：THUEE_PLUTOScy- THUEE-Unity暑培教程