Ruby’s Adventure简略版复习笔记

# 初识Unity

1.Unity五大基本视图面板

Project Window(项目面板)：项目面板会显示当前项目中的所有文件和目录，包含了项目里所有的资源文件。

Hierarchy Window(层次面板)：层次面板会显示当前场景中的所有游戏游戏对象，并显示父子级关系。

Scene view(场景演示面板)：场景演示面板是当前加载的场景和层次视图中所有游戏对象的实时预览窗口。

Game View（游戏演示面板）：游戏演示面板是测试游戏时将显示内容的窗口面板。

Inspector window(检视面板)：在工程面板或者层次面板中选中一个对象时，检视面板会显示它所有的数据，并且

可以修改。

2.其他面板

Console Window(输出面板)：输出面板会显示当前游戏开发中生成的警告错误。

3.工具栏

Play(运行按钮)：运行按钮用于测试当前在层次面板中放置的游戏物体加载后的游戏运行情况。

Pause(暂停按钮)：暂停在游戏面板里的游戏测试，帮助发现游戏问题。

Step(逐帧运行按钮)：逐帧运行按钮用于逐帧遍历暂停的游戏场景。

Hand Tool(视野查看工具)：可以拖拽移动当前视野，快捷键Q。选中此工具时，按下左键，鼠标滚轮（中间）或

Alt+左键可以拖拽移动视野。按下Alt+右键或滚动鼠标滚轮可以放大或者缩小视野。

Move Tool(移动工具)：可以将选中的物体进行移动，快捷键W。

Rotate Tool（旋转工具）：可以将选中的物体进行旋转，快捷键E。

Scale Tool（缩放工具）：可以将选中的物体进行大小缩放，快捷键R。

Rect Transform Tool(矩形工具)：可以查看和编辑2D游戏物体的rectTransform组件，可以移动缩放旋转

2D和UI等等游戏物体，快捷键T。按下shift键可以等比缩放游戏对象。

Rotate Move Or Scale Tool(多功能工具)：可以对选中的游戏物体进行移动，旋转，缩放，快捷键Y。

其他一些快捷键

聚焦某一个游戏物体：选中游戏物体，按下F。

视野移动：按下鼠标右键或鼠标中键。

调整视野大小：滚动鼠标中键。

3D游戏视野：按下鼠标右键+键盘上的WSADQE键进行视野移动。

# 主角Ruby的创建

1.坐标

场景中的所有物体都有3个坐标。分别是Horizontal水平轴向（x）,Vertical垂直轴向（y），depth深度（z）。

如果游戏对象无父对象，则坐标是GameObject距场景中心的距离。

如果游戏对象有父对象，则坐标是GameObject距其父对象的距离。

2.脚本包含命名空间，类（组件），函数（功能），其中两个重要的函数。

Start():当游戏开始时，Unity只在Start中执行一次代码,且在第一帧更新之前调用。

帧：为了给人留下运动的印象，游戏（就像电影）是高速播放的静止图像。通常在游戏中，30或60个图像显示在

一秒钟内，这些图像都称为帧。

Update():在创建该图像（帧）之前，Unity会执行在所有游戏对象的Update函数中编写的代码。1秒钟大约调用60次。在此函数中，可以编写任何希望在游戏中连续发生的事情（例如，读取玩家的输入、移动游戏对象或计算累计时间）。

3.变量（Variable）:是我们在计算机内存中命名的位置（这个由自己命名的），可以使用这个变量获取到在那里存储的值。

4.变量类型：告诉计算机要存储哪种数据，以便可以在内存中获得适当的空间量。

Vector2变量类型：一种存储两个数字的数据类型，存贮在Inspector面板里关于transform组件中positon的x,y值。

# Ruby的移动控制

1.输入设置 Edit > Project Settings > Input Manager

2.（静态）方法的调用：类名.方法名（参数）

3.如果想要游戏物体以每秒多少的速度移动可以乘上Time.deltaTime

4.代码的注释使用“//”，多行注释快捷键ctrl+k+c,解除注释ctrl+k+u

# 使用TileMap创建世界地形

1.在移动游戏物体时同时按下ctrl键不松，可以让游戏物体移动固定长度，固定长度值需要在Edit>Grid and Snap

Settings>Move里设置

2.TileMap创建：Hirerachy>右键>2D Object>TileMap

网格(Grid)：可以用来将游戏对象均匀地放置在其单元格中。

瓦片地图（Tilemap）：是网格的子游戏对象。由Tiles（瓦片）组成的。

精灵（Sprite）:纹理的容器。大型纹理图集可以转换为精灵图集(Sprite Sheet)。

瓦片（Tile）：一种特殊的精灵。使用瓦片就像在画布上画画一样，画画时可以设置一些颜色和属性。

调色板(Palette)：保存瓦片，将它们绘制到网格上。

笔刷(Brush)：用于将画好的东西绘制到画布上。使用Tilemap时，可以在多个笔刷中任意选择，

绘制出线条、方块等各种形状。

3.每单位像素数：选中精灵资源在Inspetor面板里会看到，表示在一个Unity单位中一行或一列有几个像素点

4.切割图集：选中一张精灵图片资源，将SpriteMode从single改为mutiple，可制作成图集，点击SpriteEditor可进行

切割。切割有三种形式：自动，按像素切割，按行列切割。

5.调色板的工具栏与快捷键：

选择工具（Select Tool）：选择想要使用的瓦片，可以点击选择或拖拽多选选择，快捷键S。

移动工具（Move Tool）：可以选择并移动瓦片位置，快捷键M。

画笔工具（PaintBruch Tool）：选择瓦片并在场景面板中绘制，快捷键B。

方形区域填充工具（Fill Box Tool）：选择一个或一片区域大小的瓦片，绘制区域小于选中区域时会部分绘制选中内容

的纹理，等大区域时会全部显示，大于则会平铺显示,快捷键U。

取色工具（Picker Tool）：取样选中瓦片并绘制,快捷键I。

橡皮擦工具（Eraser Tool）：擦去在场景中已绘制瓦片,快捷键D。

填充工具（Fill Tool）：大范围绘制选中的瓦片,快捷键G。

6.Edit按钮激活下是对调色板中瓦片的操作，未激活是对Scene面板中瓦片的操作。

7.层级(Layer)：可以在2D游戏物体中的SpriteRenderer和地形TilemapRenderer组件中的Order in layer属性中去设置层级的大小，值越大，越后渲染，值越小，越先渲染，值大的游戏物体会覆盖值小的游戏物体。

# 第五章 丰富游戏世界

1.根据游戏对象的某个轴向去绘制游戏对象：Edit > Project Settings > Graphics > Camera Settings > Transparency Sort Mode，调整想要决定绘制的轴向，值的正负决定是大或小的先绘制，例如：Y:1，则Y坐标越小的游戏对象越后绘制（渲染），我们可以看到，不会被遮挡。Y：-1，则相反。

2.轴心点（Pivot）：是可以自定义的特殊点，充当精灵的“锚点”。精灵以此为支点进行旋转，坐标点位置则是指轴心点的位置。

3.修改精灵渲染顺序点：将游戏物体中的SpriteRenderer组件中的SpriteSortPoint从Center设置为Pivot。

4.修改精灵资源轴心点的两种方法：单张图片的精灵资源可选中Pivot属性进行设置。单张和图集都可使用SpriteEditor去自定义轴心点位置。

5.预制体(Prefab)是Unity中的一种特殊资源。预制体就是一个或者一系列组件的集合体，可以使用预制体实例化克隆体，后续可对克隆体属性进行统一修改。

6.进入预制体编辑模式的方法：1.双击 2.选中预制体资源，点击openPrefab选项 3.点击蓝色克隆体名称右边的小箭头 4.点击选中克隆体，点击检视面板中的Open按钮

# 第六章 Unity中的物理系统

1.Unity中的物体系统：模拟游戏世界中物体的运动受力与碰撞。

2.刚体组件（Rigidbody）：使游戏物体能获得重力，接受外界的受力和扭力功能的组件，可以通过脚本或是物理引擎为游戏对象添加刚体组件。

3.碰撞器组件（BoxCollider）:使游戏物体具有跟挂载刚体组件的游戏物体发生碰撞能力的组件。

4.发生碰撞检测的必要条件：双方必须都有碰撞器，且一方有刚体，最好是运动的一方。因为Unity为了性能的优化，刚体长时间不发生碰撞检测会休眠。

5.2D游戏物体因为碰撞受力会发生Z轴上的角度旋转，可以通过Rigidbody2D>Constraints>FreezeRotation去冻结Z轴上

的角度旋转。

6.2D游戏物体之所以碰撞会发生抖动，是因为物理系统控制的位移与开发者编写代码产生的位移起了冲突，需要使用

物理系统里的位移方式来进行位移，也就是借助刚体组件的移动方法来移动。

7.静态方法的调用：类名.方法名（参数），全局可访问，不需要实例。普通方法的调用：对象名.方法名（参数）。

8.给Tilemap游戏地形上添加碰撞器：在Tilemap游戏物体上添加Tilemap Collider 2D组件，将不需要添加碰撞器的瓦片资源选中，将Collider Type从Sprite改为None。

9.合并瓦片碰撞器：在Tilemap游戏物体上添加Composite Collider 2D组件，在Tilemap Collider 2D组件中启用Used by Composite属性，之后将Rigidbody 2D组件中的Rigidbody Body Type属性设置为Static。

# 第七章 道具的添加与交互

1.访问级别：所有类型和类型成员都具有可访问级别，用来控制判断是否可以允许我们访问某些成员变量和方法。可使用访问修饰符来指定声明类型或成员的可访问性。

2.在C#语言中访问修饰符有很多种，这里我们只学习常用的三种：

public（公有的） ：任何代码均可以访问，应用于所有类或成员。

protected（受保护的） ：只能在当前类或其派生类中使用，应用于内嵌类或成员。

private（私有的） ：只能在当前类中使用，应用于内嵌类或成员。

3.整形（Int）：用来存取整数的变量类型。

4.当把一个变量设置成Public类型时，可在外部Inspector里去改变值。

5.变量赋值顺序：声明变量后直接赋值先于Inpector赋值先于Start方法里的赋值

6.触发器（Trigger）：触发某一个事件的机关。在BoxCollider组件中把isTrigger属性勾上可以把碰撞器做成触发器。

7.发生触发检测的必要条件：双方必须都有碰撞器，其中一方得是触发器，且一方有刚体，最好是运动的一方。

8.在脚本中gameObject指当前脚本挂载的游戏物体对象，transform指当前脚本挂载游戏物体对象身上的transform组件。

9.使用Destory()方法可以销毁游戏物体对象。

10.当某部分代码需要满足某些条件去执行的时候可以使用if条件语句，小括号内部是条件，大括号里是满足条件才会执行的代码。

11.关系运算符：判断两个变量是否满足某种关系的运算符。

12.为了使某个变量只能够在内部修改，外部只读，可以给它制作一个属性值。

# 第八章 敌人的开发

1.如果想要放大一张有很多个图形组成的精灵资源，直接放大会变形，则需要在这个游戏物体身上找到SpriteRenderer

组件，然后将DrawMode设置成Tiled（平铺），之后将TileMode改为Adaptive（自适应）。为了使图片填满整个矩形

区域，可以将图片资源中的MeshType修改为FullRect。

1.触发检测调用的方法：OnTriggerEnter（进入触发器触发）,OnTriggerStay（在触发器内每帧触发），

OnTriggerExit（出来触发器触发）。

2.布尔类型（Bool）：用作逻辑判断的变量类型。（类似于某个状态的开关）。有两个值：真(True)1,假(False)0。

3.return关键字：可以直接跳出当前方法，return之后的代码都将不再执行。

4.if条件判断语句：如果满足if条件判断语句中的条件，则会执行if代码块。如果不满足，则会执行if后续的代码，

直接跳过if代码块。如果有else语句，则执行else语句中的代码块。

5.碰撞检测调用的方法：OnCollisionEnter，OnCollisionStay，OnCollisionExit。

6.如果想要获取碰撞到的游戏物体可以使用Collision类型的对象.gameObject的形式获取，进而可以获取该游戏物体

身上挂载的某个组件。

# 第九章 Unity中的动画系统

1.动画制作组件（Animator）：为了完成游戏对象动态效果的控制和创建，Unity为我们提供的一个动画设计解决方案，同时也是一个组件。

2.动画控制器（Animator Controller）：根据当前游戏对象的状态控制播放对应的游戏动画。也可以叫做动画状态机。

3.动画（Animation）：Unity中的动画资源，可以由动画控制器，代码去进行播放，也是一种组件。

4.选中对应游戏物体打开动画面板，在动画面板中可以创建新的动画，第一次动画可直接创建，后续需要在动画名称上

点击，在下拉菜单中点击CreateNewClip创建。也可以在Project面板里右键进行创建。

5.创建动画后可以给动画添加该游戏物体上的所有组件所持有的属性改变的相应动画，点击Preview可以预览动画效果

，点击右边红色圆点按钮可以进行录制操作，在Inpector里属性的改变会直接将关键帧添加在时间轴上，也可以右键

在时间轴上添加关键帧。

6.将若干个图片拖拽进动画窗口可自动添加制作成一个关键帧动画。可全选待鼠标变为双向箭头进行等比拉伸，延长

动画时间，变慢动画速度。

7.如果没有对应动画的图片资源，可以借由改变SpriteRenderer中的Flip属性去翻转动画。

8.将动画拖拽进动画控制器窗口中可以添加动画状态。

9.Parameters列表里可以添加配置参数，参数可用于设置动画过渡的条件。

10.右键动画状态可设置默认动画以及添加过渡条件。

11.选中过渡条件连线可在Inpector面板中的Conditions去设置参数条件。

12.在代码中获取Animator组件之后可以使用set+变量类型的方法去设置状态机里的参数值。

13.如果想要让上一个动画瞬间过渡到下一个动画，需要把条件中的HasExitTime取消勾选，让动画没有退出时间，而是可以瞬间退出。且需要把TransitionDuration过渡时间间隔设置为0。

14.转换(Transition)用于在给定时间量内从一个动画状态 (Animation State) 平滑转换为另一个状态。转换指定为动画状态机 (Animation State Machine) 的一部分。如果转换迅速，则通常可从一个运动很好地转换为完全不同的运动。

15.混合树(Blend Tree)：用于允许通过按不同程度组合所有动画的各个部分来平滑混合多个动画。各个运动参与形成最终

效果的量使用混合参数进行控制，该参数只是与动画器控制器关联的数值动画参数之一。要使混合运动有意义，混合的

运动必须具有相似性质和时间。混合树是动画器控制器中的特殊状态类型。

16.Vector类型的变量既可以用来表示坐标，也可以用来表示向量。

17.Mathf.Approximately(a,b)可以用来判断两个浮点数a,b之间是否近似相等。

18.a.Normalize()方法可以用来使向量a单位化。a.magnitude是可以取到当前a的向量模长。

# 第十章 子弹的开发

1.通过物理系统给游戏对象施加力的效果，需要先获取到刚体组件，然后调用AddForce()方法。

2.监听玩家按下某个按键可以使用Input.GetKeyDown(KeyCode.\*)，\*代表某个按键。

3.实例化（Instantiate）：如果想要动态生成一个游戏物体，需要调用实例化方法来对预制体进行克隆。

第一个参数是想要克隆的预制体资源，第二个参数是需要生成到的位置，第三个参数是旋转角度。旋转角度跟检视面板里显示的是不一样的，是一个四元数类型的参数，不是Vector3类型的参数。

4.Quaternion.identity表示预制体默认旋转角度。

5.空引用异常：我们使用到了某个只声明但并没有实例的对象。（只有容器没有东西）

6.输出面板里有错误详细信息，包括出问题的所有位置，哪个类，哪个方法，哪一行，双击可进入报错位置。

7.Awake方法先于Start方法调用，都是Unity内置封装的方法，跟Update方法一样，统称为生命周期函数。Awake方法

会在游戏物体实例化（生成）时就调用，Start方法不会。

8.层级（Layer）是Unity中场景物体的一种属性。可用于设置物理碰撞关系。

9.设置是否发生碰撞检测的层级Edit > Project Settings > Physics 2D，另外对应的游戏物体右上角要去设置对应的层级。

10.想要让某个游戏物体的刚体从当前物理系统中移除，不再发生碰撞检测，可以使用rigidbody2d.simulated=false

# 第十一章 Cinemachine工具包的使用

1.Unity提供了许多内置功能，大多数游戏和应用程序都需要这些功能。并非所有游戏都需要通过软件包提供更专业的功能。包只会增加使用它们的项目的大小，

并且不易于它们游戏的更新。

2.包是一组代码和资产，可以使用包管理器添加到项目中。它们将功能添加到项目中，这样开发者就不必自己编写代码，例如虚拟现实支持、后期处理效果等等。

3.可以在右上角Window > Package Manager里添加项目中需要使用到的工具包。

4.Cinemachine工具包可以快速创建复杂的相机设置，允许在多个相机之间移动，跟随和剪切。

5.摄像机的投影模式：

透视（Perspective）：所有离开相机的线都会聚到一个点上，当它们离相机越远时，使物体看起来越小。

正交（Orthographic）：所有平行线保持平行的地方。

6.规定摄像机边界，可以给地形做一个碰撞器边界，在CMvcanm里的cinemachineCOnfiner中的Boundingshape2D设置一个带有PolygonConllider2D或是CompositeCollider2D的碰撞器的空游戏对象。

# 第十二章 Unity中的粒子系统

1.粒子在交互应用中广泛用于效果。粒子系统会产生几十个甚至几百个粒子，这些粒子是具有方向、速度、寿命、颜色和许多其他特性的小图像。通过使用所有这些参数，粒子可以集体创建诸如烟雾、火花甚至火焰之类的效果。

2.在Hierachy窗口中右键点击Effect>ParticleSystem可创建粒子系统。

3.如果想要粒子系统使用精灵图片资源，则需要勾选SheetAnimation，且将模式改为Sprites。

4.如果想要在多个图片资源中随机产生则需要将StartFrame选项的Constant改为Random Between Two Constants。常量代表的是使用的资源索引。

5.如果不想要粒子在播放中更换图片资源，则需要将FrameOverTime的动画曲线设置为水平点，即删除线段末尾的关键帧点。

6.想要设置粒子系统的基础属性，则可以点击ParticleSystem打开属性栏。

7.启动寿命（Start Lifetime）：一个粒子的寿命是它在被粒子系统破坏之前在屏幕上的时间。如果在场景视图中缩小，将看到所有粒子在同一个位置周围消失。因为粒子以相同的速度和寿命开始，它们最终在相同的距离被摧毁。单击“开始生存期”右侧的小向下箭头，可以在两个常量之间选择随机产生数值。

8.开始大小（Start Size）：粒子创建时的大小。

9.开始速度（Start Speed）：粒子开始移动的速度。

10.Size Over Lifetime：随着生命周期粒子大小的改变。

11.Color Over Lifetime：随着生命周期粒子颜色的改变。

12.在粒子系统基础属性里，将Simulation Space的值从Local改为world，粒子就不会跟随游戏物体进行移动了。

13.想要在一段时间爆发产生大部分粒子可以使用Emission属性。

# 第十三章 Unity中的UGUI

1.玩家不会有一个控制台窗口来查看最终游戏中的日志。因此，为了给他们反馈，交互应用程序使用用户界面（简称UI），它将图像和文本覆盖在场景上以显示信息。这也被称为平视显示器（HUD）。

2.Unity中的UI必须使用画布（Canvas）组件来对其进行渲染，画布定义了每个UI元素在屏幕上的呈现方式，并负责呈现它的所有子UI组件。

3.（事件系统）EventSystem：创建画布的时候会同时创建，来处理事件和与UI的交互，比如鼠标点击。

4.Canvas组件中的RenderMode（渲染模式）：

屏幕空间-覆盖（Screen Space-Overlay）：这个默认模式使Unity一直在游戏顶部绘制用户界面。大多数应用程序使用此模式是因为它们希望UI始终位于最上面，以提供所有信息。（最常用）

屏幕空间-相机（Screen Space-Overlay）：这将在与相机对齐的平面上绘制用户界面。该平面的大小将始终填充屏幕，因此可以四处移动相机，并且该平面将与相机一起移动，以显示与覆盖相同的效果。但是，因为平面是在世界上绘制的，而不是在屏幕上，所以世界上的对象可以在用户界面上

绘制。（也就是说一些游戏物体可能会覆盖UI）

世界空间（World Space）：这将在世界任何地方绘制一个平面。可以将此平面用作游戏中计算机的屏幕、墙壁或角色的顶部。（比如制作血条）

5.画布缩放器组件（Canvas Scaler）：该组件定义UI如何根据不同屏幕大小进行缩放。

UI的缩放模式（UI Scale Mode）：

恒定大小（Constant Pixel Size or Constant Physical Size）：这使得UI保持相同恒定不变的大小，无论屏幕是什么大小或形状。这使得UI在任何屏幕上都保持可读性，但是较小的屏幕可能有很大的UI覆盖空间，如果屏幕太小，UI元素可能会重叠。

按屏幕大小缩放（Scale With Screen Size）：按屏幕大小缩放：这使得UI的缩放取决于开发者设置为参考分辨率的屏幕大小。

6.选中Image，点击组件中的Set Native Size按钮可以使图片保持原本大小。

7.选中矩形变换工具（快捷键T），按住shift键点击四角的蓝点可以等比缩放图片大小。

8.锚点可以保证当前UI的轴心点与锚点之间的距离保持不变，可以通过调整锚点的值来做UI的自适应。使用自定义锚点可以使图片基于父对象等比进行缩放。

9.不规则图形的填充可以使用mask组件，父对象UI需要挂载mask组件，子对象则为具体填充UI。

10.想要百分比填充UI，首先需要获取到mask组件的width属性（轴心点要设置为左上），设置血条填充百分比可以使用mask.rectTransform.SetSizeWithCurrentAnchors方法去设置。

11.想要全局访问某一个变量的方法或者成员变量可以使用单例模式。

# 第十四章 Unity中的射线检测

1.可以同时选中多张图片，直接拖动进Hierachy面板中，可以直接做出动画控制器和动画。

2.可以使用射线检测的方法去判断当前前方是否有某个游戏物体。

3.创建UI文本可以使用Text组件，outLine组件则可以描边。

# 第十五章 游戏音效与音乐的添加

1.音频剪辑资源（AudioClip）：音频剪辑是资源，如纹理或脚本。可以从音频文件（如mp3、ogg和wav文件）中导入音频片段。

2.音频监听器（AudioListener）:音频侦听器是定义“侦听器”在场景中的位置的组件。用来接收音频资源的播放，在使用空间化声音时非常有用。场景中有且仅需要一个。

3.音频源（AudioSouce）:音频源是一个组件，允许在组件所在的GameObject位置播放音频剪辑。

4.如果想要音乐循环播放需要勾上AudioSouce组件上的Loop属性。

5.如果希望游戏运行后播放音乐需要勾上PlayOnAwake选项。

6.volume控制音量大小。Spatial Blend控制音乐播放方式为2D还是3D。

7.音效的播放可以使用AudioSouce组件的PlayOneShot方法。

8.使用随机数可以使用Random.Range（a,b）方法，如果a,b为整数int，则随机数包含a,不包含b，如果是浮点数，则左右都包含。

9.如果想要播放持续音效，可以直接调用audioSouce.Play（），停止则可以使用audioSouce.Stop（），是否正在播放状态的判断可以使用audioSouce.isplaying

# 第十六章 游戏完善与打包

1.bool的默认值是flase，int和float是0。

2.Unity打包后的玩家可执行应用程序称为player，在 Edit > Project Settings > Player可设置player的一些配置参数。

3.公司名称（Company Name）：用于创建将要存储游戏中创建的文件的文件夹，或其他与系统相关的文件。

4.产品名称（Product Name）：游戏或应用程序名称。这将用于命名可执行文件/包（取决于平台），并在公司名称下创建

一个位置，以保存与该游戏相关的所有内容。

5.默认图标（Default Icon），即应用程序的图标，例如移动设备上的应用程序图标和桌面上的可执行图标。

6.默认鼠标指针样式（Default Cursor），可以将其设置为具有不同于系统默认箭头的光标。

7.不同平台可以配置不同平台的一些参数设置。

8.在Edit > Build Settings中选择添加当前场景与打包平台，单击Build即可打包。

9.Unity随后将构建游戏，在此期间，不能在Unity编辑器中执行任何其他操作，因为Unity正在压缩和打包的所有资源，

检测并忽略未使用的资源，将脚本编译为优化的表单和其他小任务，所有这些都是为了减少游戏占用的空间并使其访问数据更快。

10.游戏中如果没有设置退出，可以使用快捷键退出，在Windows/Linux上按ALT+F4或在Mac OSX上按Command+Q退出。

11.每次打包要换一个文件夹以及产品名称新设定的一些属性才可以在游戏包中起作用。