1. 程序基本逻辑

场景运行时程序的基本逻辑顺序：运行之前的场景包括8个货架和两个无人机，场景运行之后，货架会根据脚本CreatMatter创建出物料，并且将物料的对应属性写到物料的property脚本里，包括物料种类和位置，并进行一些初始化设定。镜头切换到主相机视角，并且呈现主菜单界面，上面有各类按钮，可以调整视角、启动飞机等等，按下起飞键，对应的飞机起飞，按照预设路线一路行进，此时物料上的MatterCatched脚本会判断物料是否进入无人机视野，如果进入，则会发送相应信息到物料统计界面。统计界面则根据这些信息生成图表以供使用者查看。并且与此同时，已经被扫描到的物料在主视角下也可以通过点击直接打开物料的详细信息界面。

1. 脚本功能和实现方式介绍
   1. 相机相关脚本

**Monitor：**无人机第三人称视角相机的初始化，以及其旋转功能。挂载于两个无人机的第三人称视角相机上。

**MoveCamera：**相机移动脚本，当该相机激活时可以通过对应操作进行移动和旋转视角，挂载于主相机上。

**CameraInformation：**相机提示框，当对应相机激活时，该脚本会控制UI界面的CameraInform对象显示出相应信息。挂载于主相机上。

**FlyCameraInformation：**功能和CameraInformation一样，但是该脚本是挂载于除了主相机之外的四个无人机相机上。

**DestroyWindowForCam：**按钮脚本。隐藏相机信息提示框，挂载于UI界面的CameraInform对象上的退出按钮上。

**CameraChangeToMain：**按钮脚本。通过改变对应相机enable的方式来切换五个相机视角，挂载于主UI界面左上角的五个相机切换按钮上。其中有五个Click函数，分别对应了五个相机。

**CameraReset：**按钮脚本。将主相机视角复位。挂载于UI界面主相机按钮下的Reset按钮上。

* 1. 无人机相关脚本

**FlyCatch：**无人机轨迹脚本。该脚本是以时间为判断条件，使无人机进行运动的，每个无人机需要扫描四个货架，每个货架有两个面，每个面的扫描动作依次为：向前平移——向下平移——向前平移——向上平移。每组平移之间停顿0.5s。挂载于两个无人机上。

**FlyInformationUpdate：**该脚本挂载于相机提示框下的文本Text\_FlyInformation上，这是当第一人称视角相机激活时，控制相机提示框同时显示飞机信息的脚本。其中飞机的坐标直接调取对应无人机的transform。而位置则调取FlyCatch中的时间变量t，通过时间变量t来判断无人机此时所处的位置。

**Rotate：**螺旋桨旋转效果脚本，挂载于无人机的四个螺旋桨上，仅作为动画效果展示，对总体运行不产生影响。

**FlyInformLeft：**左侧的常驻无人机信息框的实时更新脚本。方法同FlyInformationUpdate。

**InformButton：**按钮脚本。左侧的常驻无人机信息框的进入按钮。当关闭了该信息框之后会显示该按钮，点击可再进入。

* 1. 货架相关脚本

**InitializeShelf：**定义一个int型变量shelf，需要在Unity中自行输入，代表的是货架的编号。目前的场景一共8个货架，故货架编号从一至八。挂载于场景中所有的Shelf对象上。

**CreateMatter：**挂载在货架的隔板上，自动生成物料，生成方式采用的是对Prefab进行instantiate重载，然后将物料对应的信息写入物料的property脚本中，可以通过修改两个公有变量Type和Number的值来生成不同类型的物料。

**Counting：**货架柱状图统计脚本。该脚本的作用是在统计界面生成对应货架的柱状图统计。采用的基础方法是重载，在统计图界面预先创建一个image方块作为柱状图的一个单元，当物料被扫描到之后，物料上挂载的MatterCatched脚本会修改Counting脚本中的type值，然后Counting脚本在每一帧都会根据它的type值来重载一个对应的image单元，这样就做出了柱状图的效果。但缺陷是如果在同一帧里同一货架上有多个物料被扫描到，就会出现漏统计（type只会保留最后一个物料的type值），因此在MatterCatched脚本中设定了每一帧只统计一个物料。该脚本挂载在统计界面的Shelf1-8上。

**ShelfSelector：**统计界面下拉菜单脚本。控制不同编号的货架的统计信息出现或隐藏。

* 1. 物料相关脚本

**property：**物料的属性脚本。该脚本定义了物料的各类属性，包括类别type，所处货架shelf，所处隔板board，在隔板上的位置location，扫描飞机fly。其中，前四者是货架在创建物料时写入的，fly是根据shelf的值直接确定的。

**MatterCatched：**挂载于物料上。主要功能为判断物料是否进入无人机视野，如果进入，则进行相应操作。该脚本中首先划分了左右两排物料，分别由无人机二号和一号负责。然后在Update中使用WorldToViewportPoint函数来进行判断。如果物料进入对应无人机视野，则会调取property中的信息传递给统计页面。需要注意的是该脚本中涉及的几个状态量——send，mark，isEmpty。send是为了防止物料被重复统计，所以当完成一次统计之后就会将其改为Off。mark是为了防止物料被“透视”，因为是否被扫描到是采取的视野判断方法，而货架上有两排物料，因此如果不做区别，那么无人机在扫描正面物料时反面物料也会在视野中，就会出现反面物料也被扫描到的“透视”现象，因此需要加入mark量，这个变量是由物料创建脚本CreatMatter来初始化的，在创建物料时，将反面物料的mark都定义为0，正面为1，当无人机完成正面扫描之后，会在FlyCatch脚本中也创建一个mark量等于1，这时MatterCatched会修改自身的mark量以激活反面物料。IsEmpty量是位于无人机脚本FlyCatch上的变量，它的主要功能是确保每一帧只有一个物料进入统计，防止统计界面的柱状图漏统，当任意物料进入视野后会修改对应无人机的isEmpty量，这样就确保了这一帧里后执行的其它物料的Update函数中的判断不会通过。然后在LateUpdate中再将这一变量修改为true。

**MatterInformationScript：**物料信息脚本，挂载于物料上。当物料被扫描到之后，点击物料，会弹出相应的物料信息。采用的方法是对预设对象MatterInformation的重载并重新编辑。判定点击的方法是OnMouseDown和OnGUI，因此和UI界面按钮不会互相覆盖。

**Drag：**挂载于MatterInformation上。使得物料信息窗口可以被拖动，并且点击相应的物料信息窗口会使它浮于UI界面的最上层。

**DestroyWindow：**挂载于MatterInformation上的退出按钮上。销毁对应的弹窗。

**InitializeWindow：**挂载于MatterInformation上。使弹窗初始模板MatterInformation上的Drag脚本处于非激活状态。这一设计是为了解决UI界面按钮相互覆盖的问题。否则Drag脚本会使初始模板浮现于UI界面最上层而导致对退出按钮的遮挡。

**MatterNumber：**挂载于ChartCanvas界面下的Matter1-6对象的文本中，作用是对文本进行实时更新。

**Matter1Detail：**按钮脚本，挂载于ChartCanvas界面下的Matter1-6对象按钮上，作用是界面切换，进入到对应的物料类别的详细统计界面。

**DetailedData：**挂载于DetailedCanvas上，会根据Matter1Detail发送过来的相应type值来创建对应类别物料的详细统计表。基本方法是对预设模板DetailedMatter1的重载并编辑，这个模板包含一个按钮和一个文本，文本显示物料的编号（编号规则见下文），按钮可以进入该物料的详细信息栏。这个脚本同时会接收MatterCatched脚本送来的物料的property信息，然后会给每个物料根据它们的property信息来进行重编号并显示在图表中。编号规则可以参照以下例子：

No.2F-T1S1B3L1

意义为一号无人机（F，二号是FF）扫描到的第二个物料（No.2），物料类型1（T1），位于一号货架（S1）第三层（B3）正面第一个（L1）。

**Detailed：**按钮脚本。挂载于DetailedCanvas下的DetailedMatter1上。作用是创建相应物料的详细信息框。依然是对Canvas界面下的MatterInformation对象进行重载并重新编辑，此处的重新编辑是直接对物料编号进行了“解码”，即根据编码规则从编码重新推导出该物料的对应信息。

其余直接位于Script文件夹下的脚本则是UI界面上的各个按钮的相应脚本。

**Continue：**飞机起飞按钮。

**ESC：**退出游戏按钮。

**ESCUI：**隐藏主UI界面按钮。

**GetIntoUI：**进入主UI界面按钮。

**GoToChart：**进入统计界面按钮。

**ReturnToMain：**从统计界面退回主UI界面按钮。

**ReturnToChart：**从详细的物料表统计界面退回统计界面按钮。

**Initialize：**UI界面的三个Canvas初始化。