第十课，检测--交互

1.检测

在我们之前的课程中已经讲过交互了。并且也讲了一下怎么检测我们脚下的方块。那么我们今天将要来学习一下，怎么检测敲击方块。

在这里我们一定可以想到我们会用到block文件。其实这是毋庸置疑的。因为我们要检测方块嘛。 然后我们今天还要用到event文件，字面意思，事件。没错。 今天我们要学的就是敲击事件。 通过event来判断我们是否敲击了方块。

pollBlockHits()这个方法就是监测方块是否被敲击，并且返回一个被敲击方块的坐标值组成的列表。当然，它监测的是整个minecraft世界的方块。所以，我们要对比一下。我们敲击的方块与我们想要敲击的方块是否在同一个位置。所以我们用这段代码，获取到了被敲击方块的坐标列表 events = mc.events.pollBlockHits()。然后与我们建造的方块进行对比。

2. 10\_1BlockHit.py

分析：既然我们学习了函数，就肯定要把敲击判断这段代码写在函数中，然后去调用，当然了。 我们也有很多不必要的代码不用写在函数中， 比如，创建与服务器连接的对象。在敲击之前所创建的一个钻石方块，我们都不用写在敲击的函数中， 那么我们一开始的代码就要来创建一个钻石块mc.setBlock(D\_pos.x,D\_pos.y,D\_pos.z,block.DIAMOND\_BLOCK.id)

，钻石块创建好了。我们开始写checkHit这个敲击所用到的函数

用events获取到敲击方块的坐标。然后我们通过for循环遍历。与之前我们钻石块坐标的对比，如果一样，那么就说明，我们敲击到了这个钻石块！

第十一课，寻宝游戏（编程序）

1，游戏流程讲解

今天的任务就是写代码，一节可能写完就非常不错了。如果写不完，下节课最多给20分钟写代码并且熟悉代码，那么在我们开始编程之前，我们来了解一下这个游戏的玩法和流程。首先我们运行程序之后，在我们周围会随机出现一个钻石块。在我们角色脚下会出现一个黄金块随着人物移动而增加。每增加一个黄金块会减1分直到我们敲击了钻石块位置，当我们敲击钻石块的时候，会+10分。并且消除之前出现的所有黄金块。所以想要得到最多分，就要选择最短的鹿城到达敲击钻石块的位置。

好，那么我们先来看看第一个程序

2，11-12\_1skyHunt.py

声明了三个坐标变量，声明了一个RANGE钻石方块随机位置的范围。声明了一个score分数的变量。

声明了四个函数。

placeTreasure（）用来随机钻石方块的设置

checkHit（）用来判断敲击的方块

homingBeacon（）计算角色与钻石块的距离。

buildBridge（）建造黄金桥梁，并计算减分情况

循环：判断什么时候调用placeTreasure（）创建钻石块

然后调用checkHit（） homingBeacon() buildBridge()

来完成这段程序

写完这程序之后，我们就已经完成我们本次游戏程序的框架了。接下来我们编写每个函数的具体内容就行了。接下来我们一起完成这本次的任务程序11-12\_2skyHunt.py

第十二课，寻宝程序（讲程序）

1.11-12\_2skyHunt.py

好了。 本次课程比较难，希望大家仔细听讲。同时，本次课程也充分的展现了函数的重要性。如果不用函数，那么我们将会很难写出这次项目的程序来。 在我们看函数之前我们讲一下之前没有见过的None这个关键字。我们看到我们给treasure\_x,treasure\_y,treasure\_z这三个变量赋值的是None，那么这是一个特殊的值。是一个没有空值，也就是说，它只是用来占个位置的。实际上是没有值的。

现在我们看一下，placeTreasure（）这个函数。首相将 treasure\_x,treasure\_y,treasure\_z这三个变量声明成了全局变量，什么事全局变量呢，如果某个变量在函数中需要改变值。那么我们必须将这个变量声明称全局变量，也就是在变量前加一个global关键字。下面我们将treasure\_x 赋值成 random.randint(pos.x,pos.x+RANGE)这样的一个随机数。范围是玩家x坐标和玩家坐标+5的这么一个范围。y坐标则是玩家坐标+2和玩家坐标+5的这么一个范围也就是说，这个钻石方块会出现在玩家周围5的范围。并且不会低于玩家（我们把玩家整体当做两个方块作比较）注释：这里老师可以给孩子在白板上或者在大屏幕上画图演示一下。然后我们用这个随机出来的三个坐标值，创建一个钻石块

我们再来看看checkHit（）这个函数，和我们之前写的十分像，只不过是多了把敲击的方块变成空气。其实这个很简单。当我们敲击方块的时候。用for遍历了一下返回来的被敲击方块的坐标然后我对比一下。确定了我们敲击的是我们要的钻石块。然后加分score = score+10那么现在score变成了10.然后我们，setBlock把原来的位置上的钻石变成空气就可以了。这时候，我们一定要将 treasure\_x 给重新赋值成None，因为我们在循环中通过确定 treasure\_x 的值是空的，来确定已经没有钻石块了，然后再创建新的钻石块。

在说homingBeacon（）函数之前， 我们都注意到了。TIMEOUT和timer这两个变量。这两个变量使用来计算次数的，让我们一起到homingBeacon（）函数中去了解的更多吧。首先在这个函数中有两个判断，一个大的判断，判断 treasure\_x 的值是否为None，也就是说如果没有钻石块。这个函数也就运行到了这里不会再运行了。

那么如果有钻石块。 我们将用timer来计数。每计算10次，就会执行一次函数主要内容。现在我们来看一下主要内容吧。diffx，diffy，diffz，这三个变量的值，分别是玩家坐标与钻石块坐标的差值，然后取绝对值赋值给他们。这样 这三个变量代表着。与钻石块的距离。因为三个值不好表示，所以我们声明了一个diff将diffx，diffy，diffz给相加到了一起。然后我们在聊天屏幕上打印一下score和treasure展示出我们当前的得分数以及与下一个方块的距离。

接下来让我们看一下最后一个函数buildBridge（）。

首先可以看到声明了一个全局变量score。因为分数在这里又会有改动。也就是我们的减分情况。每多一个金块就会减少一分。获取了玩家坐标，然后我们 用get方法获取到了玩家脚下的方块属性。这里我们会看到又要判断treasure\_x 的值是否为None，这里很好理解了，。 如果我们把钻石块敲碎了，那么我们就应该把路上的金块都消除。所以当钻石块消失的时候， 我们来判断一下bridge的长度， 如果大于0， 说明金块桥存在。那么我们就要用pop方法把bridge中的坐标都删除并且赋值给coordinate，然后我们把coordinate中的坐标位置设置成空气块，（金块变空气） 这样就实现了当我敲碎了钻石块，那么我所走过来的金块就都消失掉。

然后我们看到elif，这个我们曾经使用过的关键字。如果上边条件不成立，那么就判断这个条件。现在我们来看一下这段代码。

if treasure\_x == None:

elif b != block.GOLD\_BLOCK.id

如果treasure\_x是空的，说明钻石块被敲碎了，那么我们就消除金块，若果treasure\_x不为空，说明钻石块存在，那么我们要判断b != block.GOLD\_BLOCK.id脚下的方块是不是金块。那么我们就要set出金块来。然后我们把这个刚刚set出来的金块坐标放入coordinate中，然后把coordinate在append到bridge，这样每走一步，就会把这一步的坐标存入到bridge中。当我们敲碎钻石快的时候就会在通过bridge中的坐标一个一个的消除金块。

score = score-1.出现在这段代码中就是为了，当生成一个金块的时候，就让分数减一。

循环：因为这是一个一直玩的游戏，所以要用循环来一直调用，

先判断什么时候调用placeTreasure（）创建钻石块（当没有钻石块了，桥也没有了就要调用placeTreasure来创建了）

然后调用checkHit（） homingBeacon() buildBridge()

来完成这段程序

备注：这个程序逻辑性跟强，而且有些绕弯， 需要老师多带学生看几遍。争取每个学生都学会这个程序， 那么我们前边上的所有课程也就都差不多了。