首先这次作业设计到的知识点：

1. 常量，变量
2. Input, print
3. 字符串，元组，列表，字典
4. If，elif，whille，break
5. 定义函数以及他的返回值。用 return 结束函数的运行
6. 读任务书以及逻辑思维能力

Part 1 – PDF/CONCEPTS 解读

Pdf 中 getting Start 大家都懂吧，主要介绍文件夹中的文件

Level: 下一段 Concepts 中的 levels 是一个字符串，也就是打印出来的地图，涉及到 load_level(在 a1_support.py 中)函数。

load_level(filename)： 输入一个文件，比如 level1 之后会返回一个字符串，打印这个字符串就可以打印整个地图。

Position： 是一个二维坐标（x，y）。我们将 level 这个地图想象成二维的坐标系，position 就是坐标系的点。

但是怎么将坐标系上的点对应的符号，转化成打印出来字符串上对应的符号呢？

1. 字符串使用 string[index],可以找到对应 index 下的字符；
2. 这时候用 a1_support 中 position_to_index(position, size)，将 position 转化，可以得到 index；
3. 这个函数中的 size 是下面 level_size(level)，输入 level 得到的，使用 level_size 函数，他的返回值就是 size。

Tile： 就是每个字符代表的意思。A1_suport 里面常量 (# Level constants 下面), 所有的常量你要用他们的大写的名称代替，这样你会符合代码要求~~主要的要是改变常量，只要改他对应名称赋值就行了，所以当你要写 x== “ ”,写成 x==AIR。

Part2 - 下面我们开始写在主函数准备用到的 function:

1. **get_position_in_direction(position, direction)**

   这个看起来在 pdf 写的好几句，但是在实际上我们看 a1_support 的 DIRECTIONS, 在# Movement constants 下面，这个就是坐标变化。

   大家的准备知识是要知道字典是什么、怎么用 key 找 value。 下一个是解包，（解包的意思是 x, y=position，position 是一个元组，如下图所示）

   x 加上 x 变化值，y 加上 y 变化值，最后返回新的坐标。

```
In[2]: position = (0, 1)
In[3]: x, y = position
In[4]: xi, yi =(0, 1)
In[5]: position = (x+xi, y+yi)
In[6]: print(position)
(0, 2)
```

2. **get_tile_at_position(level, position)**

   level 是一个字符串，所以用 level_size(level) + position_to_index(position, size) + string[index]

就可以了

3. **get_tile_in_direction(level, position, direction)**

你有了 get_position_in_direction(position, direction)返回的新 position，然后 get_tile_at_position(level, position)就可以。

4. **remove_from_level(level, position)：把给定位置变成 air**

实现如何把字符串中给定位置字符改变，怎么说呢，这个是个比较难的问题，所以先 google 一下；

方法有多种 ，replace 是一种

第二种先变成 str->list，再变回字符串，我个人很推荐第二种（如下图所示）。

```
In[2]: str1='qwert'
In[3]: list1=list(str1)
In[4]: list1[0]='a'
In[5]: str1=''.join(list1)
In[6]: print(str1)
awert
```

5. **move(level, position, direction)**

难点。大家读这段逻辑很复杂，简单起见，我给出我的一种通过的解决方案，不让大家发愁 test 过不去。

i. 首先走一步，如果下一步是 WALL 或者 AIR 才会判断，如果其他符号不会判断只返回位置。

ii. 当下一步是墙或者空气，为了防止出现多个的情况，所以用 while True 循环：在循环内，如果不是空气，UP；如果 UP 后是空气了，那就直接返回 UP 后空气的位置，结束循环，结束 function。

iii. 如果不是空气，进行下一次循环。再循环内如果是空气，那就向 DOWN 再继续循环，直到不是空气。

这样避免了 test 中 test falling onto symbol 的情况，这种情况下 player 要求站到 symbol 上（怪兽之类）

这种只是思路之一，我觉得大家容易理解，写起来还算简单，但我不知道是不是最简单的。

6. **attack(level, position)**

get_tile_in_direction(level, position, direction)左边是不是 monster：

是：remove_from_level(level, position)并且打印 Attacking the monster on your left!：

get_tile_in_direction(level, position, direction)右边是不是 monster：

是：remove_from_level(level, position)并且打印 Attacking the monster on your RIGHT!：

其余情况：No monsters to attack!

7. tile_status(level, position)

这个东西你第一眼看很难理解，这是用来干啥的？

但是我们现在不需要理解，只要知道要求写出什么东西就行

所以看 pdf 理解一下英语，啧啧，但是像我这种英语差的，读不明白，那咋办？我一般用有道


return the level with the monster tile removed.
If neither the left side nor the right side of the player contains a ... rn the level unchanged.

tile_status(level, position)
If the tile at the position is the goal tile then print Congratulati
If the tile at the position is a monster tile then print Hit a mons
If the tile at the position is either a coin tile or checkpoint tile, rer
Finally, return a tuple containing the tile character and the level.

main

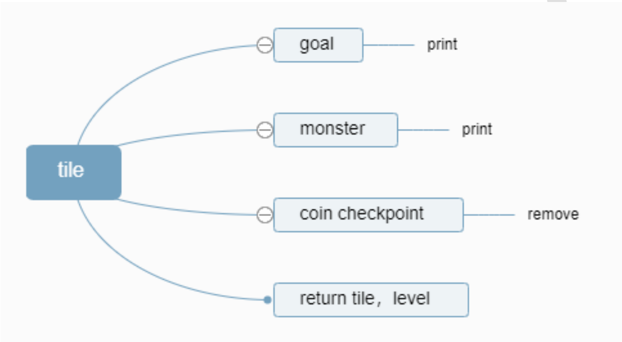
Handles the main interaction with the user.

When the game starts, it should ask the user for a file to loa ... the level file (e.g. level1.txt):

有道 划词 英汉互译
If the tile at the position is the goal tile then
译文：
如果位置上的平铺是目标平铺，则打印
如果该位置的瓷砖是一个怪物瓷砖，然后打印
如果该位置的瓦片是硬币瓦片或检查点瓦片，则将瓦片从关卡上移除。
最后，返回一个包含平铺字符和leve的元组
进入翻译页面

看看乐呵就行了，翻译的一点不准

大概意思就是判断下面的 tile 是什么符号，然后打印，或者执行操作



Part3 - 为游戏运行而准备的程序做完了，所以现在我们要做的是主程序：

```
Please enter the name of the level file (e.g. level1.txt): level1.txt
Score: 0

                              $
            @               ####
          ###              ######
*       #######   $    $  ########   I
######################################
Please enter an action (enter '?' for help): r
Score: 0

                              $
            @               ####
          ###              ######
 *      #######   $    $  ########   I
######################################
Please enter an action (enter '?' for help): r
Score: 0

                              $
            @               ####
          ###              ######
  *     #######   $    $  ########   I
######################################
Please enter an action (enter '?' for help): r
Score: 0
```

很明显

score：

地图

"Please enter an action (enter '?' for help): "

这是一个循环，是游戏的主体。

那么在开始前要准备什么？

1. 输入得到文件名字：'Please enter the name of the level file (e.g. level1.txt):
   Load 变成字符串
2. Player：position，checkpoint，savescore，savemap，理由：要保存重启游戏的状态

开始游戏：

WHILE;

打印 score：

打印 level：

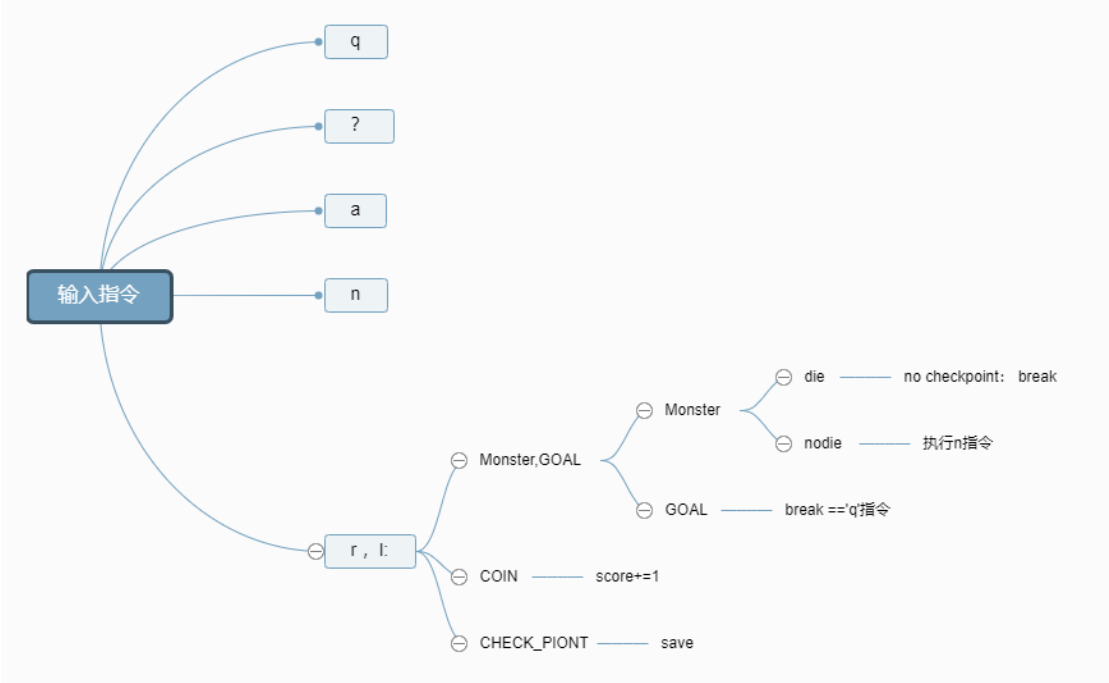
输入操作指令 r，l，？，a，n，q：if， elif 处理

最简单的 q：break 循环

其次 a：attack 函数返回新的 level 地图

？：打印 HELP

n：重开，之前的 Player position 变成 checkpoint，score 变成 savedscore，map 是 savedmap。



当输入是左右移动的时候，情况 4 种，回顾一下我们有什么东西没用到？

tile_status(level, position)

当使用这个函数，我们可以得到位置上对应的 tile 字符和新的地图；

i. 所以我们先用 move 得到下一步的位置，用 tile_status 得到该位置的字符和新的地图。

ii. 字符一定是以下情况 AIR，COIN， CHECKPOINT， MONSTER，GOAL（WALL 在 move 函数中已经解决了）
iii. 如果是 AIR，没有任何变化，只是走了一步。
iv. 如果是 COIN，score+=1.
v. 如果是 CHECK_POINT，需要将当前玩家的位置，地图，得分进行保存
vi. 如果是 MONSTER 需要检查保存的 CHECK_POINT。
vii. 如果开始游戏到遇到怪物被杀并没有经过 CHECK_POINT，是我们在初始位置设下的，我们就彻底狗带
viii. 如果经过保存点 CHECK_POINT, 我们将在上一次保存点重生。等同于输入指令：‘n’
ix. 如果是 GOAL，那就结束游戏 While 循环中 break