**设计文档**

**程序设计结构与分析**

**1、首先用Scanner 的方法读入tile.txt ，animal.txt，然后我定义了两个二维数组，一个是map，一个是tile，tile只用来存储地形地图，map用来存储动物和地形合起来的状态。我定义了一个叫tileInput的方法，用switch语句可以根据文档中的数字将地形储存在tile和map中，在map中增加动物地图时，我用switch语句，当文档中为0时，map[i][j] = tile[i][j]，其他时候12345678分别对应八个动物。（我动物文档是分开读取的**

String theleft = theLine1.substring(0,4); String theright = theLine1.substring(4,9);

**这样是为了区分左右方动物）这样就得到了tile（地形）和map（整体）两个二维数组。**

**2、代码中没有多个类。**

**3、**

**String [][] map :记录当前步数时的整体地图情况**

**String [][] tile:**

**只用来记忆整体地形地图，在我的程序里可以帮助判断动物前方的地形，从而帮助确定移动方法，同时可以在动物移动到非空地地图，再离开时保证地形地图不被改变。**

**Int [][] site:**

**用来记录每一步时所有动物的坐标，跟随动物的移动而改变，从而帮助实现控制动物的移动。**

**String [][][] mapHistory:**

**用来记忆玩家所有步数时的地图情况，帮助实现悔棋功能**

**int [][][] siteHistory:**

**用来记忆玩家所有步数时动物位置的坐标，是为了解决悔棋后动物坐标错乱的问题。**

**boolean player:根据他的true 或者false 来判断和更换左右玩家移动**

**char number: 获取的玩家要移动动物所对应的数字，从而判断移动哪个动物**

**String direction:**

**获取的用户的输入，用来判断玩家的命令（如exit、restart、1d、2s等）来确定玩家具体的下一步行动。**

**int [] currentstep:记录本步是游戏的第几步，帮助实现悔棋**

**int [] laststep: 我用它来记录玩家能撤销悔棋的最大步数，帮助实现悔棋。**

**int nextstep:用于判断悔棋后下一步是总的第几步，帮助实现悔棋和撤销悔棋。**

**4、int** compare(**int** a, **int** b)：

**主要用于比较动物对应数字的大小来判断能否吃掉敌方。**

**void** printMap(String[][] map)

**用于打印地图**

String[][] copyArray(String[][] array)

**用于实现当前地图(map)和历史地图(mapHistory[i])的互相复制，从而帮助实现悔棋功能。**

**int**[][] copyArray(**int**[][] array)

**用于实现当前动物坐标（site）和历史动物坐标（siteHistory[i]）的互相复制，帮助实现悔棋功能**

**void** tileInput(Scanner scanner, String [][] map )

**用于将地形储存在二维数组map 和tile中**

**void** animalInput(Scanner scanner1, String[][] map, String[][] tile)

**用于将动物地图储存在二维数组map中**

**boolean** admove(**int**[] laststep, **int**[] currentstep, **boolean** player, String[][] map, String[][] tile, **char** number, **int**[] thisstep, **int**[][] site, **int** nextstep, **int** nextstep1)

**这是程序的主要移动方法之一控制ad （左右）移动**

**主要改变的就是根据移动指令改变map，实现移动，**

**改变site 实现坐标跟随动物移动，**

**改变当前所处步数currentstep[0]，实现历史棋盘的复制**

**改变player 实现玩家的判断和转换**

**boolean** wsmove(**int**[] laststep, **int**[] currentstep, **char** d1, **boolean** player, String[][] map, String[][] tile, **char** number, **int**[] thisstep, **int**[][] site, **int** nextstep, **int** nextstep1)

**这是程序的主要移动方法之一控制ws（上下）移动**

**主要改变的就是根据移动指令改变map，实现移动，**

**改变site 实现坐标跟随动物移动，**

**改变当前所处步数currentstep[0]，实现历史棋盘的复制**

**改变player 实现玩家的判断和转换**

**boolean** move(**int** i, String direction, **int**[] laststep, **int**[] currentstep, **boolean** player, String[][] map, String[][] tile, **char** number, **int**[][] site)

**这个是综合了同时也利用了ad和ws的移动的方法而做出来的总的一个移动方法，也就是只要获取了玩家输入的number，就可以通过此方法实现移动**

**主要改变的就是根据移动指令改变map，实现移动，**

**改变site 实现坐标跟随动物移动，**

**改变当前所处步数currentstep[0]，实现历史棋盘的复制**

**改变player 实现玩家的判断和转换**

**5、**

**（1）我觉得用二维数组坐标记录每个动物的位置来控制动物的移动是我最初解决移动问题的一个比较好的方法，简单易懂。**

**（2）**

System.***out***.println(((player) ? **'左'** : **'右'**) + **"方玩家行动："**);

**这个之前助教用的简单的选择格式我觉得挺好的，不用一直自己判断该哪方移动了**

**（3）我新定义了一个二维数组tile只储存地形对我之后判断前方有水，有陷阱，和保持地形不被动物的移动而改变有很大帮助。**

**（4）我将移动方式分了两个方法ad移动和ws移动，这样使写代码的思路稍微清晰一点，但也增加了代码的长度，有较多的重复代码。**

**（）写的不好的地方就是移动判断时用的if嵌套太多，而且是一个一个分情况写的，显得特别繁琐，没有找到较好的统一方法。**

**（6）还有就是一个方法内需要输入的变量太多，容易乱，出错**

**6、**

**遇到的问题及解决方法（解决方法写在了（）里）**

1. **刚开始只是学了读取文件打印单个地图，不知道怎么将动物地图和地形地图合在一起**

**（刚开始是自己在学了二维数组后自己定义的地图，到了后来又通过几次lab后对数组理解的深化，通过数组间个别元素的复制实现了此功能）**

1. **获取用户输入时经常报错，有时按回车后不进行下一步还可以继续输入**

**（多次检查代码，一步步输出每个变量的值运行，看看自己是哪儿出错了，最后大概是有的地方\*\*\*.next()用的太多）**

1. **不知道怎么根据用户的输入来控制动物的移动**

**（刚开始有想过搜索什么的但自己不会，问同学，他们说遍历一下，然而那时我并不懂，后来我就想到用数组先定义一下他们的坐标，每走一步跟着变，于是wasd我每一个都写了一个移动方法（大不了以后再改），然后实现了动物的自由移动。。。。真的很自由。。。。。）**

1. **老鼠从水中一经过水就没了。。。。动物出来陷阱，陷阱也没了。。。。**

**（一开始我是崩溃的，不过后来想了几天，想到了单独定义一个地形地图，这样不管动物怎么动都改变不了地形了）**

1. **做完动物互吃后发现被吃的动物还能走出来。。。。。（好可怕）**

**（后来检查发现还是我定义位置坐标控制动物出了问题。。。。想到的解决方案是查看那个位置标记该动物大小的数字是否还在，如果变成了” ”，就是死了，而且这样也可以判断棋子全死了对方就胜利的胜利条件。）**

1. **狮虎跳河时判断是否有老鼠阻挡，费了很大功夫。**

**（刚开始是完全没有思路，到后来阴差阳错，我发现我正好分了ad移动和ws移动，这样我分别在移动方法里加了上下跳河和左右跳河的限制条件，以动物的位置坐标为基准判断敌方老鼠是不是和它在同一行或同一列，是否阻挡了它的路，基本完成了）**

1. **悔棋刚开始是只知道使用三维数组，但不知道怎么用**

**（在lab课上助教让我们做井字棋的悔棋时给了我大致的思路，基本上和斗兽棋的悔棋差不多，于是悔棋就有了进展）**

1. **我是用坐标定义的动物，但坐标只随着动物的wasd移动而改变，一悔棋动物的坐标就和他的真实位置分离了。。。导致动物不受控制**

**（我仿照地图历史的方法，又定义了一个三维数组定义坐标历史，和地图历史一样变化，实现了同步）**

1. **写设计文档**

**（看着自己以前的一次次历史代码，想着自己当时怎么想的，极力回忆把比较大的阻挡写pj速度的几点写的差不多了）**

**意见与建议**

1. **lab课上要做的lab能不能提前上传一下，这样在课下思考了之后，就不会在课上别人都写完了，自己还没思路那么紧张。**
2. **是不是可以每周给我们看一下其他和我们pj不太相关的小游戏的代码，让我们写代码的思路扩充一下。**