## 开发文档

一．Project概述

本次project共包含17个类，其中逻辑类4个，传入类6个，界面类7个，除了基础的下棋功能，还增加了音效，重载以及一些简单动画，接下来将介绍类的功能及所包含的方法

传入类：

1 .TileType 读取外部文本文件，并将其转化为int数组，并提供getNumber（）方法获取数组

2 .Animal 继承TileType类，获得动物地图的阵营及数字信息，提供getNumberOfAnimal（）和getTheTeam（）获取动物的数字数组和阵营数组。

3 .Map继承TileType类，获得地形地图的阵营及数字信息，提供getNumberOfMap（）获取地形的数字数组

4 .AnimalImageView 读取png文件并提供相应的get方法获取文件信息

5 .AnimalImage 构造函数AnimalImage(int theTeam[][], int numberOfAnimal[][])通过传入动物的数字和阵营信息，将相应的png图片加入数组的相应位置，提供getAnimalImage（）获取ImageView数组

6 .SoundControl 读入MP3文件，并且判断当前应放的音效是哪一个。提供play（int index）方法来进行音效的播放

逻辑类：

1 .GameRule 主要用于判断某动物在某方向是否可以移动，动物对象被分为鼠··、狮虎和普通动物三种，方向判断分为上下左右，例如canMouseUp（int[][] theTeam, int[][] numberOfAnimal, int[][] numberOfMap, int i, int j）就用于判断该鼠可否向上移动（其它方法类似）

2 .Control 主要用于动物的具体移动，由于动物移动分为普通移动和狮虎跳河，因此将移动除了上下左右的区分，还对这俩种情况进行区分，例如turnUp（int[][] theTeam, int[][] numberOfAnimal, ImageView[][] animalImage, int i, int j）就是普通的向上移动，而jumpUp(int[][] theTeam, int[][] numberOfAnimal, ImageView[][] animalImage, int i, int j)就是狮虎向上跳河

3. GameHistory 接口Serializable类，提供方法将当前步骤的阵营数组与动物数字数组保存于三维数组中，例如saveTeamHistory(int[][]theTeam, int k)保存当前动物的阵营数组；另外提changeMap(int[][]teamHistory,int[][]theTeam,int[][]animalHistory,int[][]numberOfAnimal, ImageView[][]animalImage)方法重新调用历史地图来替换当前地图，通过get方法可获得保存的历史阵营数组与历史动物数字数组。

4. WhetherWin 用于判断某方是否胜利，例如leftDiedOut(int[][] theTeam )用于判断左方动物死光leftCannotMove(**int**[][] theTeam, **int**[][]numberOfanimal, **int**[][]numberOfMap)用于判断作方是否无子可动，rightEnter(int[][] theTeam)用于判断右方是否进入敌人洞穴。左方胜利同理

界面类：

1 .HelpPane 帮助界面，继承Pane，用label将斗兽棋的游戏规则与说明加入该Pane中

2 .WelcomePane 欢迎界面，继承Pane，主要是提供按钮进行界面的转换，例如“开始”按钮就是进入游戏界面

3 .LoadPane 重载界面，继承Pane，主要是用于载入之前保存的游戏信息

4 .FunctionPane 进行游戏时包含特殊功能的界面，例如悔棋等，一般情况下该类不可见，当点击“Function”按钮时界面显现，当鼠标离开界面时界面不可见

5 .MapPane（主要界面）三个内部类

AnimalHandlerThree 接口EventHandler<MouseEvent>，通过调用GaneRule的方法的判断，Control的方法的操作和GameHistory的保存方法对player，lastStep，currentStep，动物阵营、数字和图片进行操作，并将相应内容保存入历史数组

AnimalHandlerTwo 接口EventHandler<MouseEvent>，通过调用GaneRule的方法的判断，当符合条件时显现相应位置用于提示的rectangle

AnimalHandler 接口EventHandler<MouseEvent>，通过调用GaneRule的方法的判断，当鼠标离开该动物时，清除相应的标志

6. WinPane 胜利提示界面，继承Pane，用于胜利提示并禁止玩家进行下棋操作

7. MainStage 运行类，为界面提供切换和运行的平台

（注：除提及的MapPane的方法是private，其余方法都是public）

二．优缺点

优点：1.该project将逻辑和界面比较好的分离，各个类的功能比较明确。将主要的步骤都进行了方法化，同时初步尝试了继承的使用。

2．用css简化了代码内容

缺点：1.部分的方法需要传入5个参数，再加上有些参数是通过对动物或地形通过get（）方法获得的，使得在使用这些方法时看起来比较复杂，缺乏可读性

2.代码只注释了一些主要内容，对于一些重要细节的注释上仍存在不足

3．在MapPane中还掺有一些逻辑代码，尤其是对于鼠标的操作中仍有较多判断

三．编程经历

所遇困难：1.对于javafx不是很熟悉，一些界面设置的方法并不了解

2.一开始对界面的切换比较迷惘，尝试在一个Pane上叠加导致自己很乱

3.一开始不知道组件的叠加会使在底层的组件无法显现或响应鼠标操作，结果出bug，自己很着急

4. 由于类比较多，出bug时比较难debug

5.加音频时用MediaPlayer时音频路径找不到

6. 对存读档的文件输出比较迷惑

解决方法：1.初步阅读了教材，没有掌握的请教同学

2.在同学建议下将多个Pane放置于Stage中，有调控可见来进行界面切换

3. 同学帮我找到问题，自己之后按照操作要求对组件安排叠加顺序

4. 同学建议将音乐包设置成资源文件，并用SoundControl.class.getResource("/" + soundName[i]).toString()获取路径名

5.. 在同学教导下完成了

四．图形化界面

固定stage的大小，同时将图片设置成相同大小，将图片存入ImageView中，并将ImageView加入相应的Pane中，通过改变Pane的可见来进行界面切换。在Pane上加入Button，用来执行相应功能。HelpPane中在Label上写入文字，Label加入HelpPane上。

LoadPane中将存档文件名加入File中，再将File加入LoadPane中。MapPane中，加入Rectangle，用这些Rectangle是否可见来提示当前棋手，当前点击的动物以及智能提示，动物图片也保存于ImageView中后加入MapPane中。一开始将FunctionPane与WinPane设置不可见加入MapPane中，在一定条件下可见。WinPane的文字部分也是加入Label中的

五. 进阶功能

智能提示 当鼠标经过某动物，显示该动物可走的方向

音效 加入了点击动物时的动物音效以及背景音乐

存档读档 可以多次存档读档，但是存档后文件需要保存在saves文件夹中