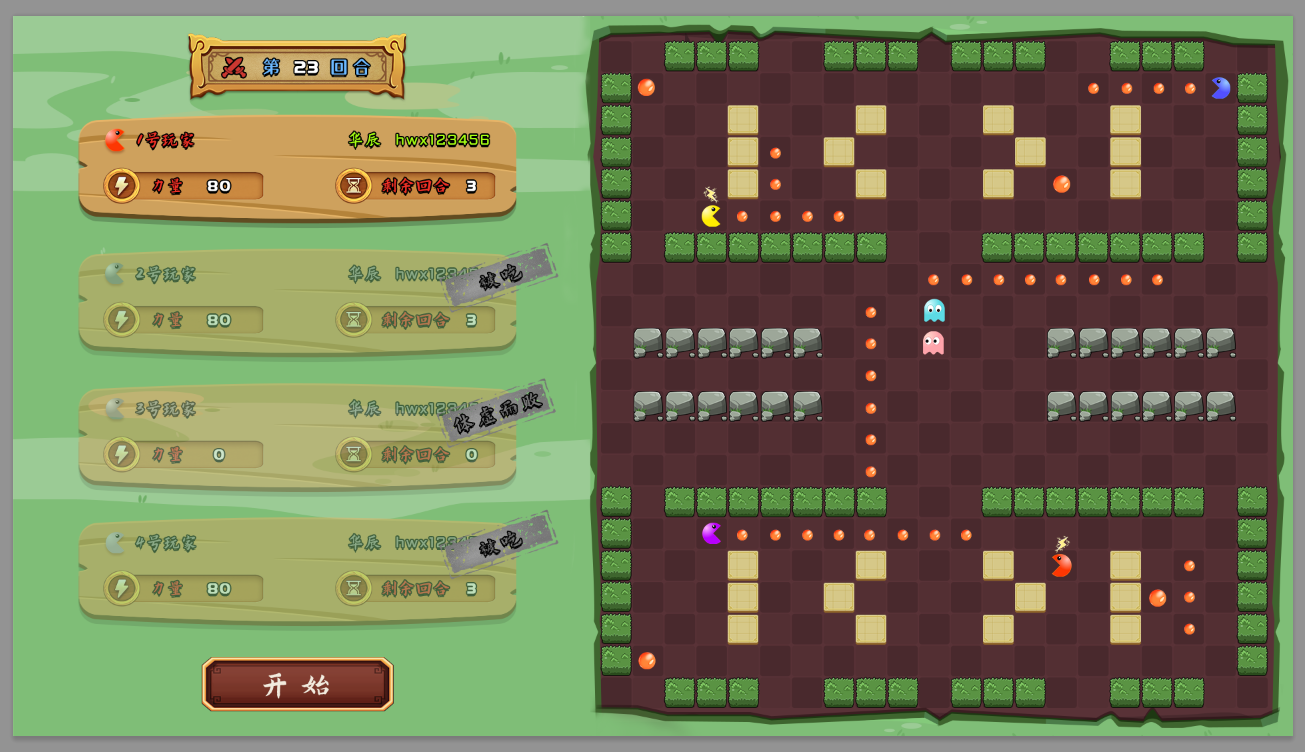
Pacman

1. 游戏介绍

趣味编程大赛 “吃豆人”：比赛随机性更高，游戏性更强，采用的是很多人都玩过的吃豆人小游戏，包含熟悉的元素，小豆子、功能豆子、幽灵、障碍等。

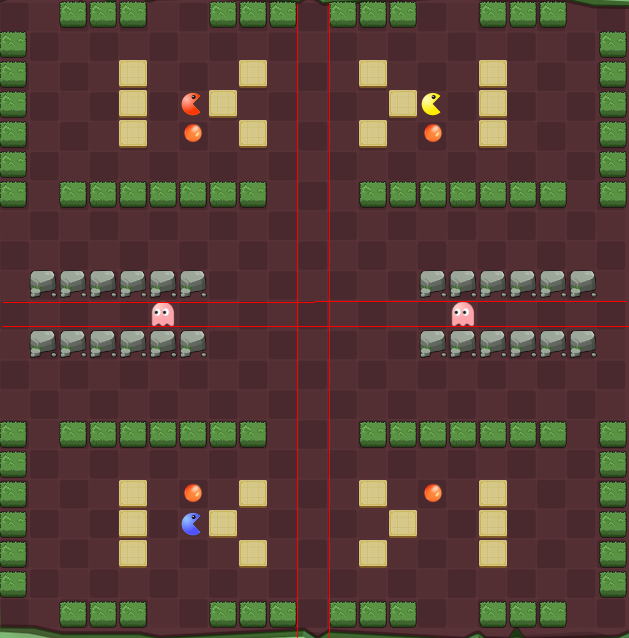
* 1. 游戏界面



* 1. 游戏规则

这是一个多人（可以2-4人）对抗小游戏，相比以往，随机性更高，加入多人对抗，以及ghost。

* + 1. 地图规则



* 21\*21的地图，基本单位是格子。
* 格子可能是通道，也可能是障碍。
* 边界是互通的，比如board[0][0]可以往上走到board[20][0].
* 随机地图，但场地是中心对称的，每个四分之一场地都是一模一样，这包括场地的初始道具、障碍、玩家初始位置等。
  + 1. 玩家角色

1. 力量：表示自己的实力，游戏结束时仅依据游戏结束瞬间的力量确定名次。
2. 无敌光环剩余回合：表示自己由于使用大豆子而获得的力量增益还剩几回合。
3. 是否死亡: 战斗失败、撞到障碍物、撞到ghost等都有可能死亡
4. 动作: 每回合，玩家需要决定自己人物的行动。行动可以是移动、或不动。移动，即选择上、右、下、左任意一个方向行进一格。

*注意：地图上会有两只怪物，初始位置是中间行的固定位置，会随意走动(根据与人物距离选择猎物)，只有无敌光环才能杀死他，但怪物和人物出现在同一个格的时候，怪物力量仅次于无敌光环。*

* + 1. 道具介绍

游戏中每个格子上只能有一个道具（Ghost例外）。地图生成的时候会有初始道具，对称分布于场地上。游戏中的道具包括：

1）小豆子：玩家可以食用，永久增加1力量。

2）大豆子：可以在【LARGE\_FRUIT\_DURATION，默认10】回合内让玩家带有无敌光环，可以 累积。

3）障碍：无法通过，如果尝试翻阅，则判定死亡

4）豆子生成器（虚拟，在地图上并不存在），在每回合都会随机生成大小豆子，并随机放置 在空格上，每次生成4个小豆子，并按一定概率(10%)生成1个大豆子

5）ghost，幽灵，强大的怪兽，除非有无敌光环，否则无法杀死，采用最短路径搜索，追杀最近的敌人

* 1. 战斗判定
     1. 玩家相遇

如果一个格子上有多名玩家(包括怪物)，而且玩家的力量有不同，那么玩家会进行战斗判定。力量最大的(带无敌光环力量最大)所有玩家是战胜者，其他玩家是战败者。

战败者会失去一半的力量并死亡。（向上取整） 战胜者会获得另一半力量并平分。（向下取整）

* + 1. 遇到幽灵

在移动的过程中遇到ghost时，如果没有无敌光环，则被杀死，如果有无敌光环，则杀死ghost。

1. 参赛规则
   1. 参赛AI客户端程序开发

目前支持4种语言开发 java、python、c/c++（注意尽量使用64位）、js，参考附录中简单示例。

主要实现以下方法：

1. 游戏初始化

*/\*\*  
 \* 该接口在游戏中只会被调用一次,用于告知选手程序游戏的初始信息。  
 \*  
 \** ***@param*** *position 选手的位置  
 \** ***@param*** *timeLimit 每回合执行的时间约束，单位毫秒  
 \** ***@throws*** *Exception  
 \*/*void ready(int position, int timeLimit) throws Exception;

***注意1，游戏只有在开始节点也就是ready告知玩家的位置，后续回合需要玩家程序自行判断****。*

***。***

2. 玩家策略

*/\*\*  
 \* 该接口在游戏中每回合都会被调用一次，用于告知选手程序当前局面，选手程序自行决定行动。  
 \*  
 \** ***@param*** *board 局面的位置列表，假设a行b列(0<=a,b<21)，那么对应的值是 21\*a+b，当前位置为ghost为-1，为空表示-2  
 \* ，为-3表示大豆子,为-4表示小豆子,为-5、-6、-7表示不同的障碍，  
 \* 其余>=0 为该位置的pacman所代表的力量（包括一个自己）  
 \** ***@param*** *strength 当前的力量，由服务器传回来  
 \** ***@return*** *方向0代表不动，1,2,3,4 分别代表左、上、右、下，其他输入皆非法  
 \** ***@throws*** *Exception  
 \*/*int run(int[] board, int strength) throws Exception;

***注意：如果board数组中有值超过10000，代表它有无敌光环，strength也是。***

服务器运行时内存会加载参赛者对象，每回合调用对应的方法，所以可以自己定义变量，私有方法等，用来辅助这两个方法

* 1. 开发规范

1. 所有代码应该符合cleanCode标准
2. 游戏需要时间限制，基准是1000ms，参考往年的经验，C语言是\*2,java是 \*3，js是\*10，python是\*10，人工时\*100. 也就是分别为2s、3s、10s、10s
   1. 运行

参考项目中readme.md 。

至少带2个参数，分别为玩家程序(jar/js/dll/py)。上下左右依次分布

如：

java -jar pacman.jar Game1.jar Game2.jar human`

另外也可以加参数nogui 只在控制台运行：

java -jar pacman.jar -nogui MyPacman1.jar MyPacman1.jar human

如果玩家程序名称为human，这一方就会变成人工，需要手工点击棋盘（上下左右键代表方向）。

* 1. 胜负规则

1. 如果玩家执行异常，或超时，或撞到障碍物，该玩家判定为死亡

2. 如果玩家在没有无敌光环情况下，碰到ghost，判定为死亡

3. 如果玩家在战斗中牺牲，判定为死亡

4. 游戏比赛运行到只剩最后一个玩家，或者超过预定回合（500）结束比赛，根据力量值判定输赢

1. 附录
   1. 项目地址

如发现bug欢迎指正修改

项目地址：

<https://gitlab.huawei.com/NGCRM/Infrastructure/ProgrammingCompetition/pacman>

* 1. c/c++ 实现

**注意：尽量提供64位已编译通过的程序**

cplayer.cpp

//\_\_declspec (dllexport)  
void ready(int no, int timeLimit) {  
}  
  
//\_\_declspec (dllexport)  
int run(int board[] ,int strength) {  
 return 0;  
}

* 1. java实现

public class JavaPlayer {  
 public void ready(int no, int timeLimit) throws Exception {  
  
 }  
  
 public int run(int[] board,int strength) throws Exception {  
 return 0;  
 }  
}

* 1. python实现

def ready(no, timeLimit):  
 return  
  
  
def run(board,strength):  
 return *0*

* 1. js实现

function ready(no, timeLimit) {  
  
}  
  
function run(board,strength) {  
 return -1;  
}