

Assignment3 Report 设计报告

11611208 李一林

1. *Minimax Ghost 设计 Special evaluation function 设计*
2. 效果展示 “莽夫”Ghost
3. 评估和评价
4. *practice4 题目补充*

1. *Minimax Ghost 设计 Special evaluation function 设计*

Minimax 树这一部分没有什么特殊的设计，就是用的最大最小树，判断的时候根据 agent 是 Ghost 还是 pacman 进行区分，选择最大最小。但是我们注意到在已有策略中，不管 pacman 使用的是哪种 agent，其评价函数不是简单的分数，因此，我们需要对鬼设定出更加先进的评价函数才能击败 pacman。

正像刚才所说的，最大最小树就是一个结构，对现有和下一步状态进行评估和选择才是更重要的。在设计 Special evaluation function 时，我考虑了三个变量，第一个是**当前游戏分数**，第二个是***pacman 距离豆子的距离***，第三个是***两只鬼到 pacman 的距离和***（权重最大）。经过几个小时的反复测试，找出了各项变量，并确定了各项系数。具体参见代码。

最后一项的权重增加预期使得 Ghost 对 pacman 有角落夹击的作用，事实操作情况确实如此。但是也会造成部分时候被 pacman 溜着跑。而且我把吃大豆子造成 scare 状态的情况纳入评价函数后的 agent 表现不佳，所以索性剔除了这种考虑，使得 Ghost 变得更有攻击性。但是这种选择的代价也很明确——

Ghost 追得太紧，容易被打团灭。不过在实际测试中表现很好，虽然 pacman 有时候得分会很高，但是最后 Ghost 总能取得胜利。总结一下如果有什么不足和遗憾的话就是我一直想引入一个 Ghost 离太近一段时间（被溜着跑）自动散开的机制，但是没有成功。

还有一项优点是我的评价函数很简单，运行速度很快，可以说兼顾了胜利和效率。这也是被逼无奈，我的电脑跑这个太慢了。

2. 效果展示

来看效果展示！！这个阶段我使用我自己的 pacman agent 而且加上网上和同学的超过 5 份 multiagent 文件和超过 10 个 pacman agent 进行对战。所有战全胜！！而且是全胜！！没有平局！！最后全把 pacman 吃掉了！不过使用的地图全是 smallClassic，我用其中一份 pacman agent 的几种策略和我的 Ghost 在不同地图进行对战，最终 pacman Minimax agent 和我的 Ghost 在 MinimaxClassic 地图对战中，pacman 第一次取得了胜利，此外，Ghost 在 practice4 给出的 newlayout 中表现也不是很好。

以下是用 smallClassic 地图时的对战战绩，pacman agent 使用了 **十一个版本**

```
linideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p MinimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 79
Average Score: 79.0
Scores: 79.0
Win Rate: 0/1 (0.00)
Record: Loss
linideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p ExpectimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 732
Average Score: 732.0
Scores: 732.0
Win Rate: 0/1 (0.00)
Record: Loss
```

本。（特别感谢提供 pacman agent 版本的同学）

```
[linyideMacBook-Pro:~ liyilin$ cd /Users/liyilin/Downloads/prac4/prac4_Pacman_answer
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p ExpectimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 732
Average Score: 732.0
Scores:      732.0
Win Rate:    0/1 (0.00)
Record:      Loss
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p MinimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 88
Average Score: 88.0
Scores:      88.0
Win Rate:    0/1 (0.00)
Record:      Loss
linyideMacBook-Pro:prac4_Pacman_answer liyilin$
```

```
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p ExpectimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 732
Average Score: 732.0
Scores:      732.0
Win Rate:    0/1 (0.00)
Record:      Loss
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p MinimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 79
Average Score: 79.0
Scores:      79.0
Win Rate:    0/1 (0.00)
Record:      Loss
linyideMacBook-Pro:prac4_Pacman_answer liyilin$
```

```
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p MinimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 90
Average Score: 90.0
Scores:      90.0
Win Rate:    0/1 (0.00)
Record:      Loss
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p ExpectimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 60
Average Score: 60.0
Scores:      60.0
Win Rate:    0/1 (0.00)
Record:      Loss
```

```
[linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p MinimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 267
Average Score: 267.0
Scores:      267.0
Win Rate:    0/1 (0.00)
Record:      Loss
```

```
linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p MinimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 78
Average Score: 78.0
Scores: 78.0
Win Rate: 0/1 (0.00)
Record: Loss
linyideMacBook-Pro:prac4_Pacman_answer liyilin$ python2 pacman.py -p ExpectimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
Pacman died! Score: 666
Average Score: 666.0
Scores: 666.0
Win Rate: 0/1 (0.00)
Record: Loss
linyideMacBook-Pro:prac4_Pacman_answer liyilin$
```

3. 评估和评价

其实战绩比较好的原因主要在于评估函数的合理设计，对于两个 Ghost 和 pacman 距离和的权重反复试错，当初考虑距离和就是希望利用地图的角落对于 pacman 进行夹击，通过反复试错得到的这个权重比，但是也不能确定是最优的。另外还有一个有点之前我提到过的，就是这个算法的运行速度非常快，兼顾了胜率和效率，这也是我的电脑运行特别慢被逼无奈设计的。问题也是比较明显，当 pacman 在两个 Ghost 的同一侧并且两个 Ghost 距离很近的时候他俩没办法拆解，被溜了，我试图解决没成功，效果不好。此外，在较小的地图上（比如 prac 给出的 newlayout）也不是很智能。不过考虑这种“莽夫”策略虽然比较容易被打团灭，但是战绩还是不错的，也就没继续提高。如果以后还有提高的需要的话可以在这方面下功夫。

4. practice4 题目补充

- (1) 在 smallClassic 地图中的表现如上截图。但是在 newlayout 和自建小地图中的表现不如大地图。
- (2) Minimax Ghost 的战绩在上面截图中有呈现，普通 Ghost 的战绩：

```
Pacman emerges victorious! Score: 1362
Pacman emerges victorious! Score: 1374
Pacman emerges victorious! Score: 1161
Pacman emerges victorious! Score: 1374
Pacman emerges victorious! Score: 1367
Pacman emerges victorious! Score: 1331
Pacman emerges victorious! Score: 1375
Pacman emerges victorious! Score: 1170
Pacman emerges victorious! Score: 1171
Pacman emerges victorious! Score: 1165
Average Score: 1285.0
```

可以说 Minimax pacman agent 面对普通 Ghost 能够保持 100% 胜率，但是任何版本的 Minimax pacman 在同一地图都没发击败 Minimax Ghost.

- (3) **Minimax pacman & adversarial Ghost:** 在 smallClassic 地图里 Ghost 全部胜利。Ghost 前期容易被团灭。

Minimax pacman and Random Ghost: pacman 可以全部获胜。

Pacman 会根据 Random Ghost 的行动调节自己的选择。

Expectimax pacman and adversarial Ghost: 在 smallClassic 地图里 Ghost 全部胜利。追着 pacman 跑，也是 Ghost 前期容易被团灭。

Expectimax pacman and Random Ghost: 当 Ghost 靠近，pacman 会进行规避。pacman 可以全部获胜。

Minimax pacman & Random Ghost implements the correct assumption of the ghost behavior.

- (4) 首先最大最小树的使用保证了 Ghost 在移动评估的时候会考虑到 pacman 在当前情况下会做的最好的 action 并对 pacman 的 action 进行反应。另外 evaluation function 设置的时候考虑到夹击的可能性，对距离和进行加权。所以看起来有合作的感觉。