00-吃豆人-tutorial

潘林朝

实验目的（仅供参考）

1. 熟悉博弈树及博弈搜索；
2. 掌握minmax搜索算法和alpha-beta剪枝算法；
3. 掌握评估函数设计方法，运用博弈搜索解决吃豆人游戏问题；

实验内容（仅供参考）

1. 改进Reflex智能体；
2. 设计Minimax智能体；
3. 设计Alpha-Beta智能体；

p.s. 几乎所有需要修改的内容都在**multiAgents.py**文件里面

程序流程(ReflexAgent)

测试命令：

```shell
# 键盘控制
python pacman.py
```

```
#使用ReflexAgent
python pacman.py -p ReflexAgent

#在testClassic地图上使用ReflexAgent
python pacman.py -p ReflexAgent -l testClassic
```

以Reflex智能体运行吃豆人

测试使用的命令：

```
python pacman.py -p ReflexAgent -l openClassic -n 10
```
SHELL

评分规则（仅供参考）：

在openClassic布局上运行**10**次。

1. 如果你的智能体运行超时，或者从未取胜就会得到0分。
2. 如果你的智能体最少**5**次胜利就能得到1分，胜利**10**次可得到2分。
3. 如果游戏平均得分超过**500**分就能额外得到1分，平均得分超过**1000**分就能额外得2分。

终端运行结果示例

```
Pacman emerges victorious! Score: 901
Pacman died! Score: -571
Pacman died! Score: -90
Pacman emerges victorious! Score: 580
Pacman died! Score: -157
Pacman died! Score: -273
Pacman died! Score: -179
Pacman died! Score: -341
Pacman died! Score: -403
Pacman died! Score: -344
Average Score: -87.7
Scores:        901.0, -571.0, -90.0, 580.0, -157.0, -273.0, -179.0, -341.0, -403.0, -344.0
Win Rate:      2/10 (0.20)
Record:        Win, Loss, Loss, Win, Loss, Loss, Loss, Loss, Loss, Loss
```

修改Reflex智能体代码

运行流程

multiAgents.py的**ReflexAgent**类，有`getAction`和`evaluationFunction`方法。其中，`getAction`在使用`gameState.getLegalActions`获取合法动作后，调用`evaluationFunction`方法评估每一个动作的得分，然后获取最大得分对应的动作下标`bestIndices`，最后随机选取一个动作返回。

修改evaluationFunction

successorGameState是**currentGameState**做出**action**的运行状态。

> 可以尝试去**pcaman.py**查看获取的游戏状态包含的信息有哪些。e.g.与幽灵的距离，是否吃到食物，是否吃到胶囊等。

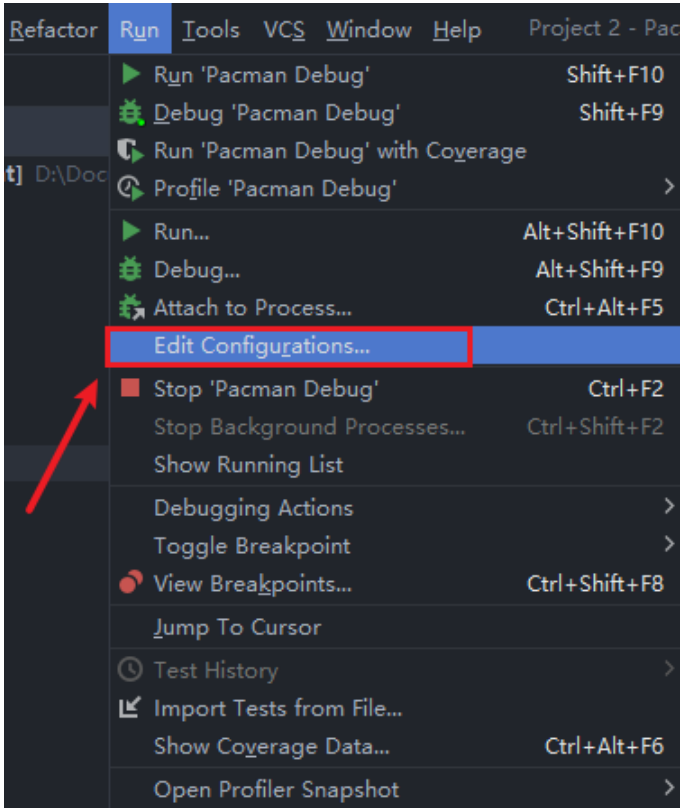
最后在evaluationFunction返回一个数值即可。

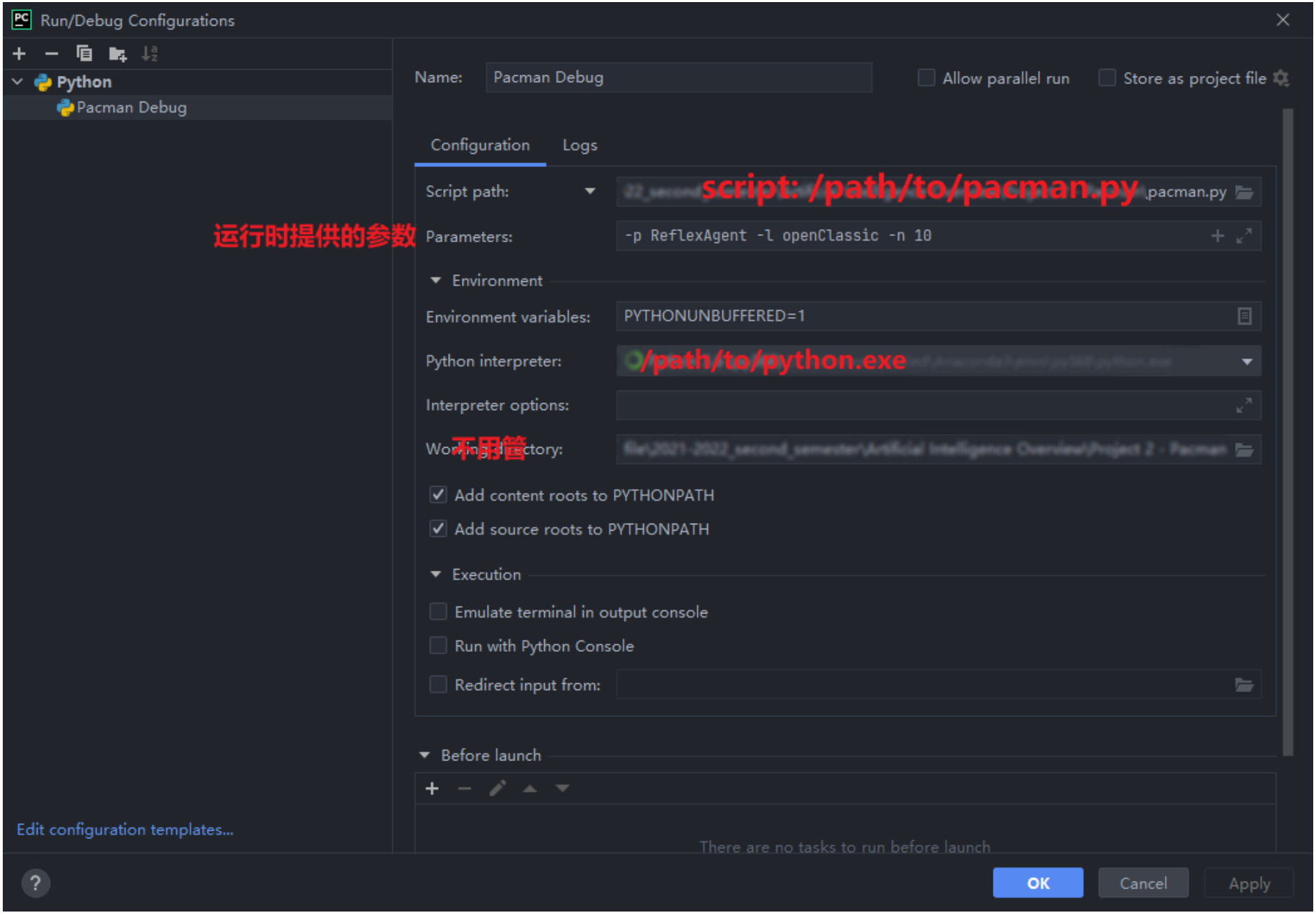
修改后的运行结果

```
Pacman emerges victorious! Score: 1239
Pacman emerges victorious! Score: 1221
Pacman emerges victorious! Score: 1231
Pacman emerges victorious! Score: 1234
Pacman emerges victorious! Score: 1220
Pacman emerges victorious! Score: 1244
Pacman emerges victorious! Score: 1243
Pacman emerges victorious! Score: 1218
Pacman emerges victorious! Score: 1242
Pacman emerges victorious! Score: 1228
Average Score: 1232.0
Scores:        1239.0, 1221.0, 1231.0, 1234.0, 1220.0, 1244.0, 1243.0, 1218.0, 1242.0, 1228.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
```

实验Tips

1. 使用调试功能查看程序的运行过程，了解吃豆人怎么移动
2. 在调试过程查看每个变量在运行过程中保存的值的类型，方便程序编写过程中的调用


Refactor
Run
Tools
VCS
Window
Help
Project 2 - Pac
Run 'Pacman Debug' Shift+F10
Debug 'Pacman Debug' Shift+F9
Run 'Pacman Debug' with Coverage
Profile 'Pacman Debug'
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Attach to Process... Ctrl+Alt+F5
Edit Configurations...
Stop 'Pacman Debug' Ctrl+F2
Stop Background Processes... Ctrl+Shift+F2
Show Running List
Debugging Actions
Toggle Breakpoint
View Breakpoints... Ctrl+Shift+F8
Jump To Cursor
Test History
Import Tests from File...
Show Coverage Data... Ctrl+Alt+F6
Open Profiler Snapshot


Run/Debug Configurations
Python
Pacman Debug
Name: Pacman Debug
Allow parallel run
Store as project file
Configuration
Logs
Script path:
script:/path/to/pacman.py
pacman.py
运行时提供的参数
Parameters:
-p ReflexAgent -l openClassic -n 10
Environment
Environment variables:
PYTHONUNBUFFERED=1
Python interpreter:
/path/to/python.exe
Interpreter options:
不用管
Add content roots to PYTHONPATH
Add source roots to PYTHONPATH
Execution
Emulate terminal in output console
Run with Python Console
Redirect input from:
Before launch
There are no tasks to run before launch
Edit configuration templates...
OK
Cancel
Apply


Project 2 - Pacman
multiAgents.py
Pacman Debug
multiAgents.py
pacman.py
getAction chooses among the best options according to the evaluation function.
getAction takes a GameState and returns some Directions.X for some X in the set {NORTH, SOUTH, WEST, EAST, STOP}
"""
# Collect legal moves and successor states
legalMoves = gameState.getLegalActions()
# Choose one of the best actions
scores = [self.evaluationFunction(gameState, action) for action in legalMoves]
bestScore = max(scores)
bestIndices = [index for index in range(len(scores)) if scores[index] == bestScore]
chosenIndex = random.choice(bestIndices) # Pick randomly among the best
"Add more of your code here if you want to"
return legalMoves[chosenIndex]
def evaluationFunction(self, currentGameState, action):
debug
添加断点，debug时候会在这里停止
ReflexAgent
getAction()
Debug:
Pacman Debug
Debugger
Console
Frames
MainThread
getAction, multiAgents.py:50
run, game.py:732
runGames, pacman.py:694
<module>, pacman.py:734
Variables
bestIndices = {list: 4} [0, 1, 2, 3]
bestScore = {float} -1.0
chosenIndex = {int} 1
gameState = {GameState}
legalMoves = {list: 4} ['West', 'Stop', 'East', 'South']
scores = {list: 4} [-1.0, -1.0, -1.0, -1.0]
self = {ReflexAgent} <multiAgents.ReflexAgent object at 0x000001BFE7FF64E0>
变量的具体信息