

سوال ۱ :

در قسمت اول این پروژه فاصله منتهی تمامی غذا ها نسبت به موقعیت فعلی پکمن را به دست می آوریم و نزدیک ترین آنرا در min و دورترین آنرا در max ذخیره میکنیم همچنین این مقدار با توجه به تغییر موقعیت پکمن, آپدیت میشود. سپس ما با توجه به موقعیت پکمن و اندازه گیری فاصله منتهی آن نزدیک ترین روح را شناسایی کرده و موقعیت آنرا در minGhostDist ذخیره میکنیم همچنین این مقدار با توجه به تغییر موقعیت پکمن, آپدیت می شود.

Question q1

=====

```
Pacman emerges victorious! Score: 1239
Pacman emerges victorious! Score: 1056
Pacman emerges victorious! Score: 1236
Pacman emerges victorious! Score: 1062
Pacman emerges victorious! Score: 1247
Pacman emerges victorious! Score: 1247
Pacman emerges victorious! Score: 1242
Pacman emerges victorious! Score: 674
Pacman emerges victorious! Score: 545
Pacman emerges victorious! Score: 716
Average Score: 1026.4
Scores:      1239.0, 1056.0, 1236.0, 1062.0, 1247.0, 1247.0, 1242.0, 674.0, 545.0, 716.0
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
```

همانطور که مشاهده می کنیم پکمن ما با این الگوریتم در تمامی حالات موفق ظاهر شده و نمره میانگین ۱۰۲۶.۹ را در ۱۰ تست خود به دست آورده است.

در این سوال در ابتدا چک میکنیم که آیا یکی از شروط پایان بازی برقرار هست یا نه سپس اگر index برای agent ما ۰ بود یعنی پکمن است پس باید تابع max روی آن صدا زده شود و در غیر این صورت index ما برای ghost است و باید تابع min روی آن صدا زده شود. همچنین برای پیدا کردن مقدار value فرزندان به صورت بازگشتی عمل میکنیم و پیش میرویم.

```
### Question q2: 5/5 ###
```

```
Finished at 22:31:51
```

```
Provisional grades
```

```
=====
```

```
Question q2: 5/5
```

```
-----
```

```
Total: 5/5
```

در این قسمت ما مطابق با قسمت قبل توابع را تعریف میکنیم و روش کلی حل ما همانند قبل است با این تفاوت که ۲ مقدار آلفا و بتا داریم که یکی از آنها را با مقدار منفی بزرگ و دیگری را با مقدار مثبت بزرگ مقاردهی میکنیم. این کار به ما کمک میکنید تا در هنگام بررسی و هرس کردن هزینه کمتری نسبت به قبل صرف کنیم.

```
### Question q3: 5/5 ###
```

```
Finished at 9:53:58
```

```
Provisional grades
```

```
=====
```

```
Question q3: 5/5
```

```
-----
```

```
Total: 5/5
```

```
C:\Users\Desktop\farhan\multiagents>python pacman.py -p AlphaBetaAgent -a depth=3 -l smallClassic
Pacman emerges victorious! Score: 998
Average Score: 998.0
Scores:          998.0
Win Rate:        1/1 (1.00)
Record:          Win
```

در نهایت مشاهده میکنیم که این پیاده سازی به خوبی عمل کرده و ما نمره قبولی کسب کرده ایم.

سوال ۴ :

در این بخش ما الگوریتم expectimax را پیاده سازی میکنیم به این شکل که برای همه روح ها یک متود پیاده سازی میکنیم که هربار نوبت یکی از روح ها بود خروجی آن یکی از حالت های ممکن برای روح ها را برمی گرداند.

```
C:\Users\Desktop\farhan\multiagents>python pacman.py -p AlphaBetaAgent -l trappedClassic -a depth=3 -q -n 10
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Average Score: -501.0
Scores:          -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0
Win Rate:         0/10 (0.00)
Record:          Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss
```

```
C:\Users\Desktop\farhan\multiagents>python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -q -n 10
Pacman died! Score: -502
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Average Score: 15.0
Scores:          -502.0, -502.0, 532.0, 532.0, -502.0, 532.0, -502.0, 532.0, -502.0, 532.0
Win Rate:         5/10 (0.50)
Record:          Loss, Loss, Win, Win, Loss, Win, Loss, Win, Loss, Win
```

همانطور که مشاهده میکنیم با اجرای AlphaBetaAgent میبایزم اما با اجرای Expectimax ما طبق چیزی که انتظار داشتیم win rate, پنجاه درصد داریم.

در این قسمت ما برای عملکرد بهتر تابع تعداد غذا های باقی مانده و همچنین فاصله از کپسول ها را نیز حساب میکنیم
همچنین با بررسی حالت های مختلف متوجه میشویم که قرار دادن ضریب از جهت اینکه باعث می شود که پکمن گمراه
نشود برای ما مفید است

```
Question q5
=====
Pacman emerges victorious! Score: 1167
Pacman emerges victorious! Score: 883
Pacman emerges victorious! Score: 1173
Pacman emerges victorious! Score: 1345
Pacman emerges victorious! Score: 874
Pacman emerges victorious! Score: 1081
Pacman emerges victorious! Score: 980
Pacman emerges victorious! Score: 1335
Pacman emerges victorious! Score: 960
Pacman emerges victorious! Score: 1353
Average Score: 1115.1
Scores:      1167.0, 883.0, 1173.0, 1345.0, 874.0, 1081.0, 980.0, 1335.0, 960.0, 1353.0
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases\q5\grade-agent.test (6 of 6 points)
***      1115.1 average score (2 of 2 points)
***      Grading scheme:
***      < 500:  0 points
***      >= 500:  1 points
***      >= 1000: 2 points
***      10 games not timed out (1 of 1 points)
***      Grading scheme:
***      < 0:  fail
***      >= 0:  0 points
***      >= 10:  1 points
***      10 wins (3 of 3 points)
***      Grading scheme:
***      < 1:  fail
***      >= 1:  1 points
***      >= 5:  2 points
***      >= 10: 3 points

### Question q5: 6/6 ###

Finished at 10:36:56

Provisional grades
=====
Question q5: 6/6
-----
Total: 6/6
```

همانطور که مشاهده میکنیم این الگوریتم و قرار دادن ضریب برای ما نمرات مطلوبی را بدست آورد.