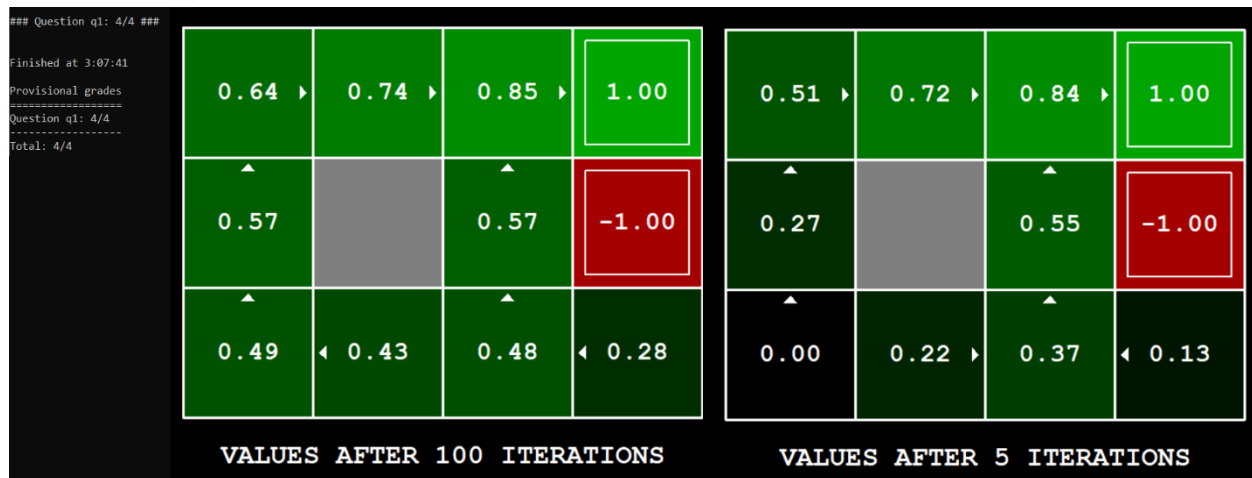


۱

ابتدا تمام state های ممکن را بررسی می کنیم و برای تمام action های state ها، value ها را محاسبه می کنیم و مقدار ماکزیمم را به state اختصاص می دهیم.

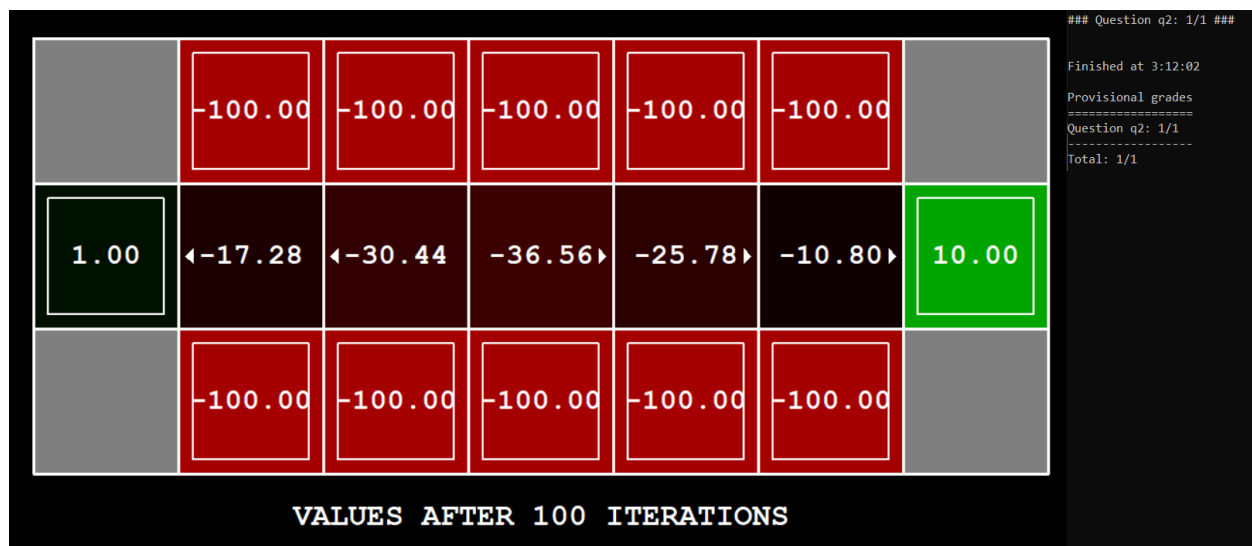
سپس QValue را از معادله ذکر شده (bellman ford) محاسبه کرده

و در آخر ماکزیمم QValue برای action مربوطه را از بین تمامی action های ممکن برمی گردانیم.



۲

در اینجا باید noise را تا مقدار تقریباً صفر کاهش دهیم و تغییر دادن discount هم تفاوتی ایجاد نمی کند.



۳

برای اینکه خروجی نزدیک را ترجیح دهیم باید discount را افزایش و یا living reward را کاهش دهیم و نویز هم موجب می‌شود از خطر صخره اجتناب کند. حال با توجه به این موارد و با آزمون و خطا مقادیر را به دست آوردیم.

```
Question q3
=====
*** PASS: test_cases\q3\1-question-3.1.test
*** PASS: test_cases\q3\2-question-3.2.test
*** PASS: test_cases\q3\3-question-3.3.test
*** PASS: test_cases\q3\4-question-3.4.test
*** PASS: test_cases\q3\5-question-3.5.test

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

Finished at 3:13:56

Provisional grades
=====
Question q3: 5/5
-----
Total: 5/5
```

۴

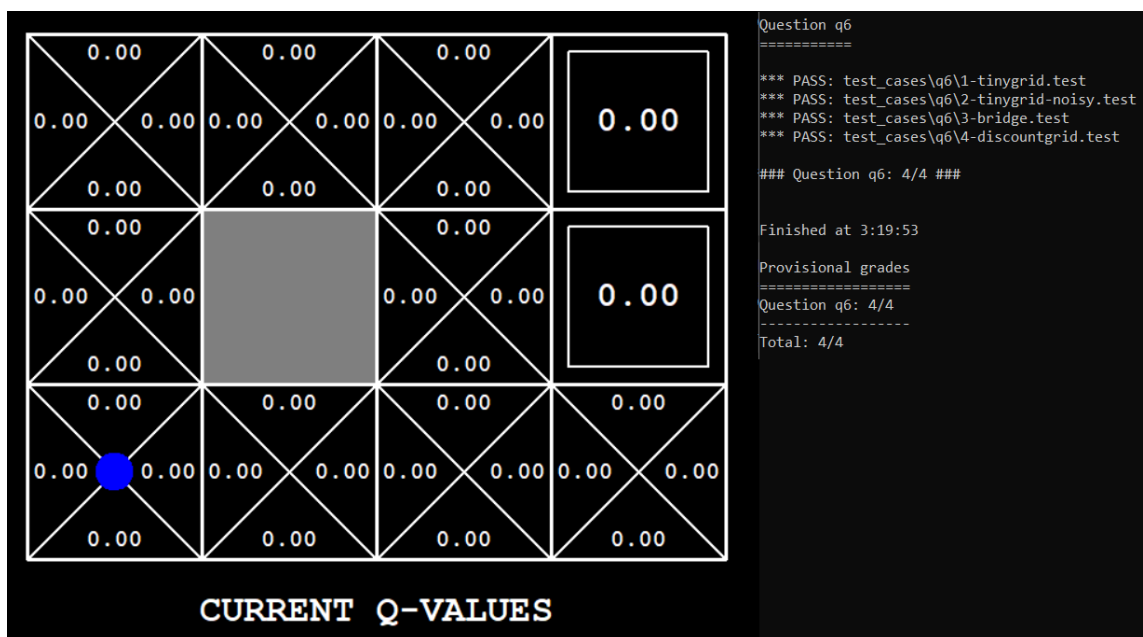
در هر iteration روی یک حالت کار می‌کنیم و valueهای action یک state بررسی و update می‌کنیم.



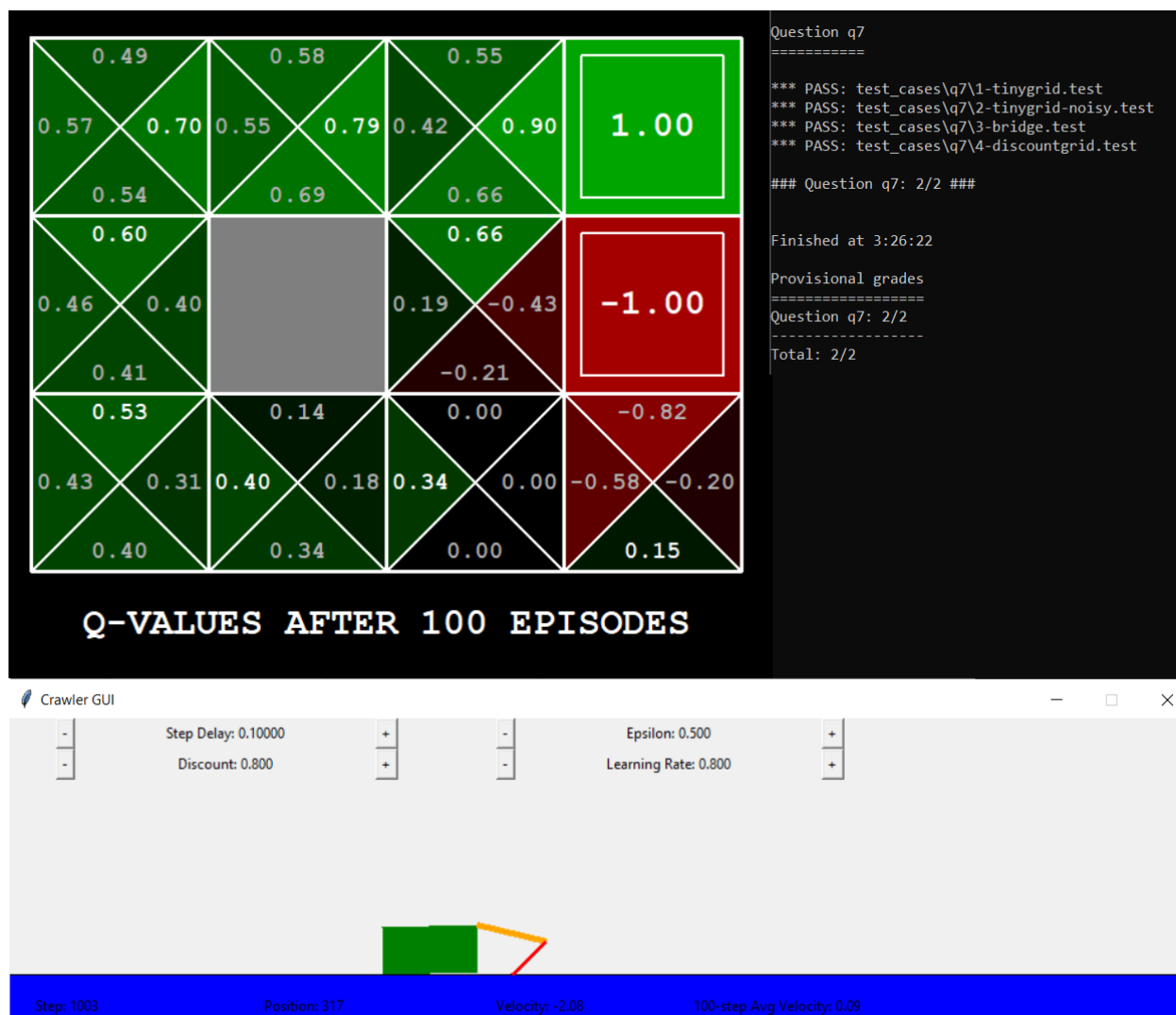
مطابق الگوریتم داخل صورت سؤال این قسمت را پیاده‌سازی می‌کنیم.



ابتدا مقدار value ها را در init تعریف می‌کنیم. سپس در getQValue با دادن state و action مقدار value را می‌گیریم. می‌توانستیم یک if هم تعریف کنیم که اگر نمی‌شد، صفر بدهد ولی بدون آن هم کار کرد. در computeValueFromQValue مقدار ماکزیمم را برمی‌گردانیم و در نتیجه بیشترین QValue انتخاب می‌شود. در computeActionFromQValue اگر action مجازی نداشتیم None برگردانده و بعد به صورت random از میان action های value برگزیده شده از میان QValue های state انتخاب می‌کنیم و برمی‌گردانیم. Update برای QValue با استفاده از فرمول است.



در اینجا با استفاده از `util.flipCoin` تابع `GetAction` را پیاده‌سازی می‌کنیم.



NOT POSSIBLE، زیرا در نصفی از انتخاب‌ها random هستند و به سیاست خاصی نمی‌رسیم.



```

Pacman emerges victorious! Score: 495
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 501
Pacman emerges victorious! Score: 501
Pacman emerges victorious! Score: 495
Pacman emerges victorious! Score: 501
Average Score: 500.0
Scores:      495.0, 503.0, 503.0, 499.0, 499.0, 503.0, 501.0, 501.0, 495.0, 501.0
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
  
```

```

### Question q9: 1/1 ###
Finished at 10:55:53
Provisional grades
=====
Question q9: 1/1
-----
Total: 1/1
  
```

در `getQValue` مثل کامنت عمل کرده و وزن را در `feature` ضرب کرده و در `update`، `features` را حساب کرده و `diff` را طبق معده گفته شده حساب می‌کنیم و در آخر با توجه به مقادیر، وزن را به روزرسانی می‌کنیم.

```

Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 495
Pacman emerges victorious! Score: 501
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Average Score: 500.8
Scores:      499.0, 503.0, 495.0, 501.0, 503.0, 503.0, 499.0, 503.0, 499.0, 503.0
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
  
```

```

### Question q10: 3/3 ###
Finished at 11:02:16
Provisional grades
=====
Question q10: 3/3
-----
Total: 3/3
  
```

