SARSA와 Q-learning의 Decaying 입실론

Episode가 충분히 많이 실행되어 큐함수가 많이 업데이트 되고 agent가 목표를 잘 찾아가게 되고, 탐험이 많이 이루어져 더 이상 가보지 않은 곳이 없게 되면, 더 이상 e-greedy에 의해 Random한 Action을 하는 것이 비효율적이게 된다. 따라서 iteration이 계속될수록 e 값을 계속해서 작게 만들어보았다.

```
class QLearningAgent:
    def __init__(self, actions):
        self.actions = actions
        self.step_size = 0.01
        self.discount_factor = 0.9
        self.epsilon = 0.1
        self.iteration = 1
        self.q_table = defaultdict(lambda: [0.0, 0.0, 0.0, 0.0])
```

```
class SARSAgent:
    def __init__(self, actions):
        self.actions = actions
        self.step_size = 0.01
        self.discount_factor = 0.9
        self.epsilon = 0.1
        self.iteration = 1 #시간이 지남을 알려주는 변수 생성
        # 0을 초기값으로 가지는 큐함수 테이블 생성
        self.q_table = defaultdict(lambda: [0.0, 0.0, 0.0, 0.0])
```

SARSA와 Qlearning 모두에 iteration이라는 value를 만들어준다

```
# 입실론 탐욕 정책에 따라서 행동을 반환
def get_action(self, state):
    new_epsilon = self.epsilon/self.iteration
    if np.random.rand() < new_epsilon:
        # 무작위 행동 반환
        action = np.random.choice(self.actions)
    else:
        # 아니면 큐함수에 따른 행동 반환
        state = str(state)
        q_list = self.q_table[state]
        action = arg_max(q_list)
    self.iteration+=1
    return action
```

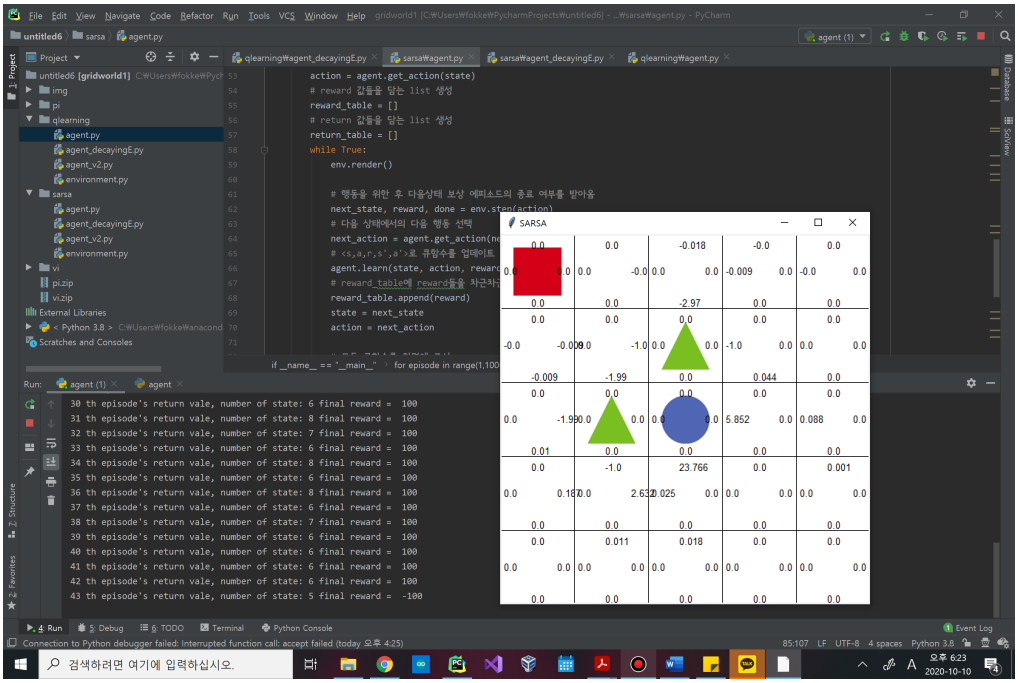
두가지 모두 get_action함수는 동일하고 iteration값으로 epsilon값을 나눠준다

매번 iteration값에 1씩 더해주어 점점 epsilon값이 작게 만들어준다.

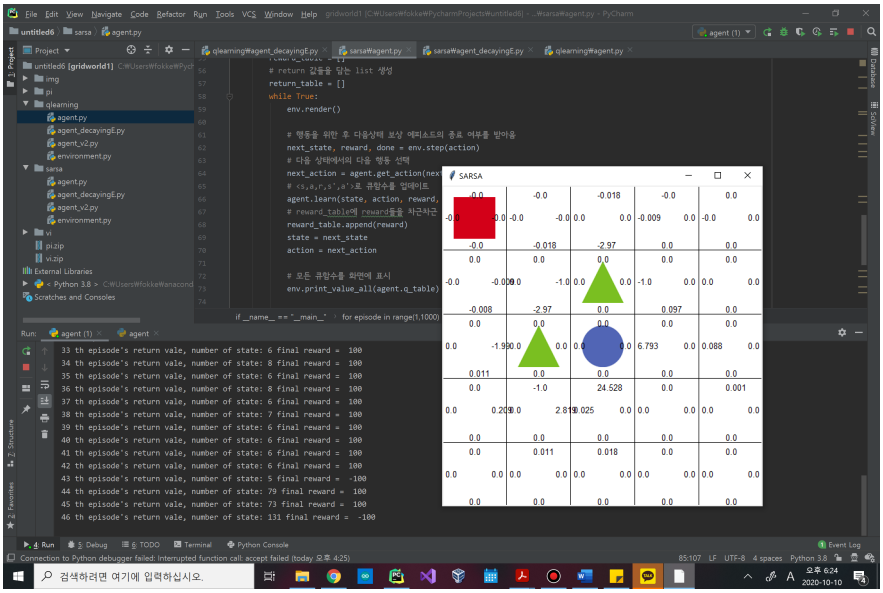
또한 episode마다 거쳐온 state의 개수와 terminal state가 원이였는지 세모였는지를 구분해주기위해 final reward도 출력해보았다.

이제 실행결과를 비교해보자.

1-1. SARSA with no decaying epsilon

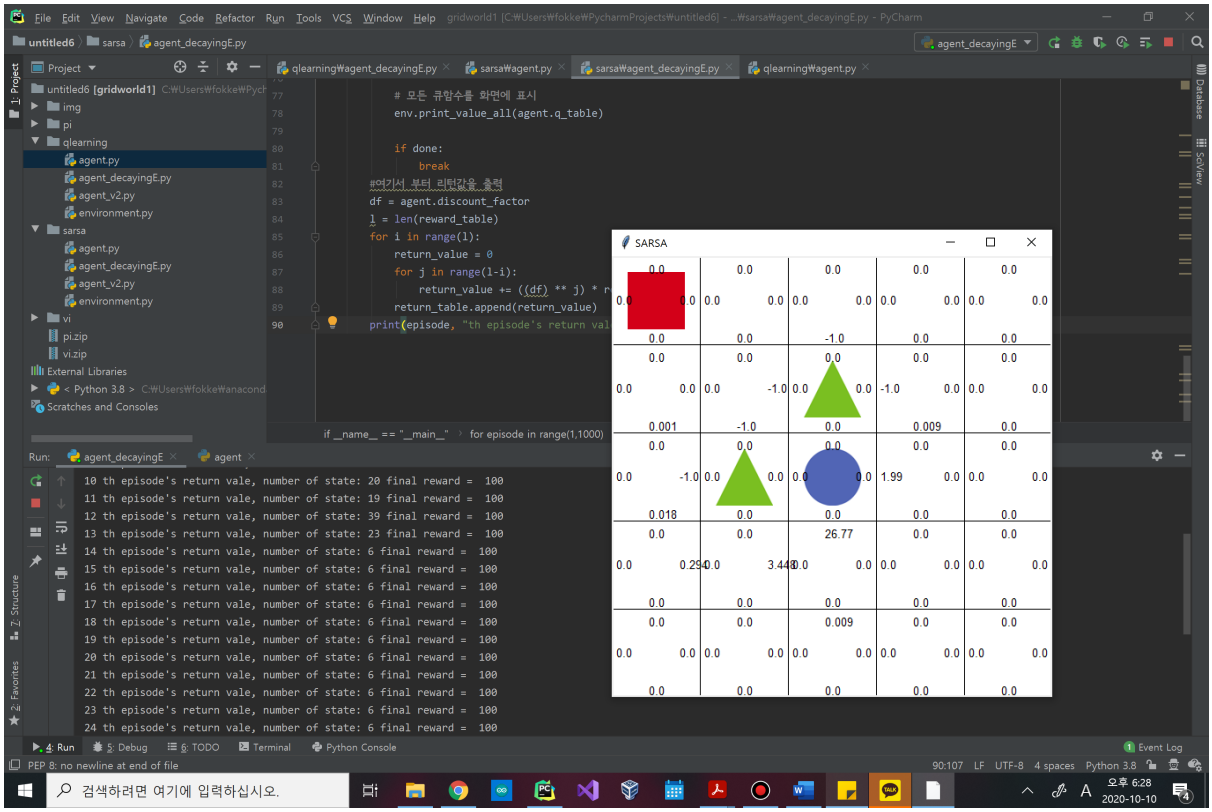


한 40번째쯤 episode에서부터 잘 찾아가는가 싶더니

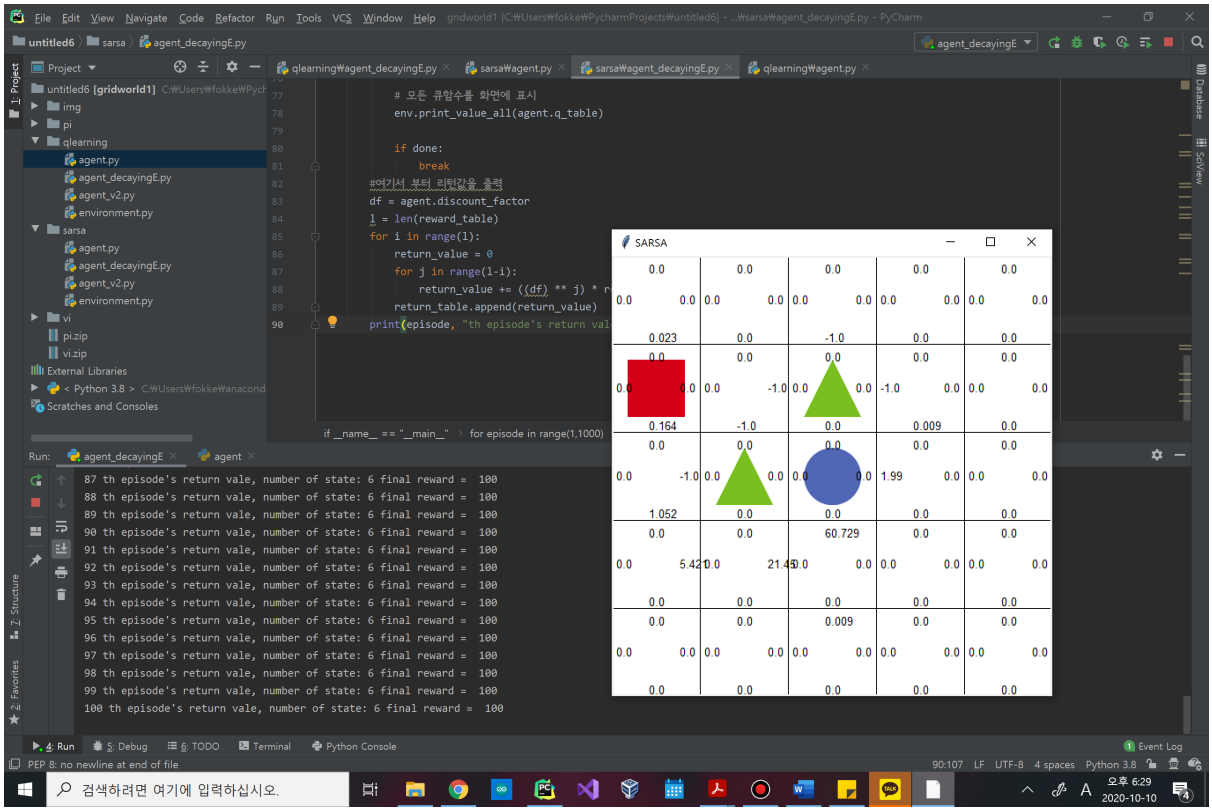


SARSA의 고질적인 문제점인 갇힘현상이 일어났다. 더 이상 움직이지 않는 지경에 이르렀다.

1-2. SARSA with decaying epsilon

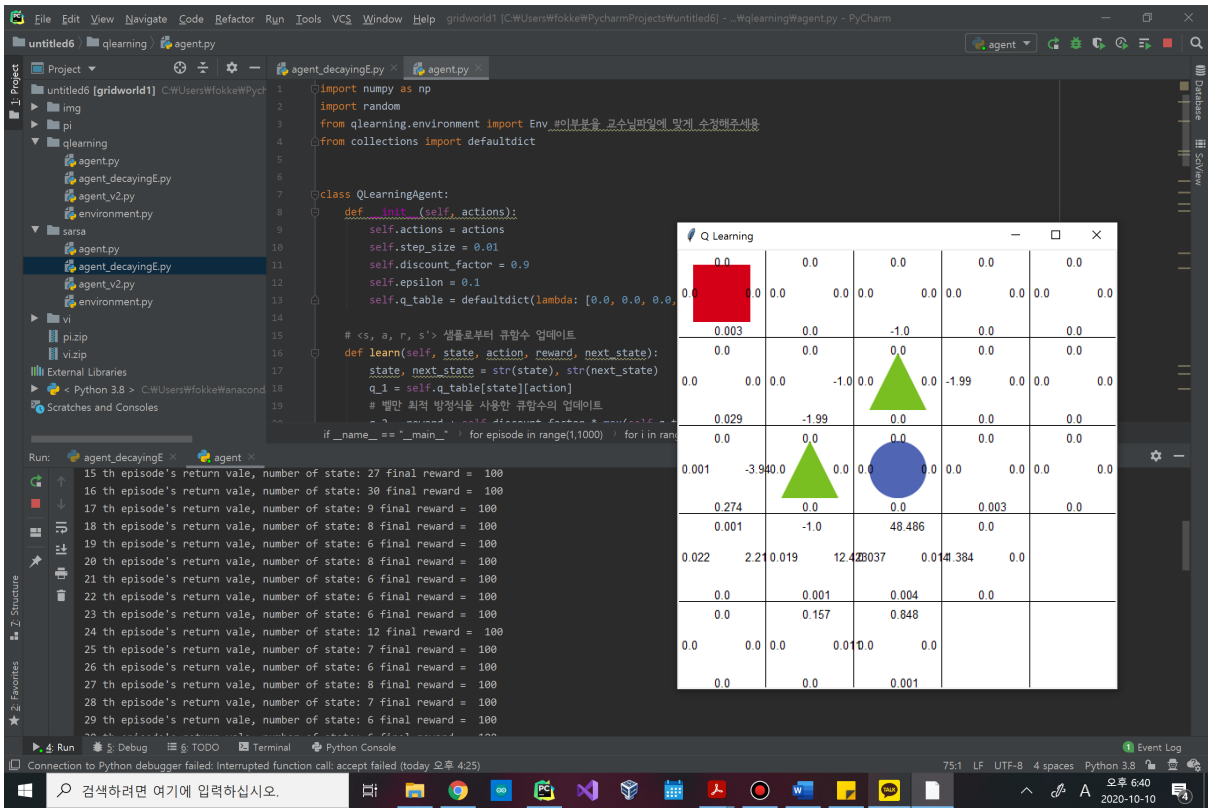


약 15번째 episode부터 최적의 경로로 원에 도착하는 것을 확인 할 수 있다.

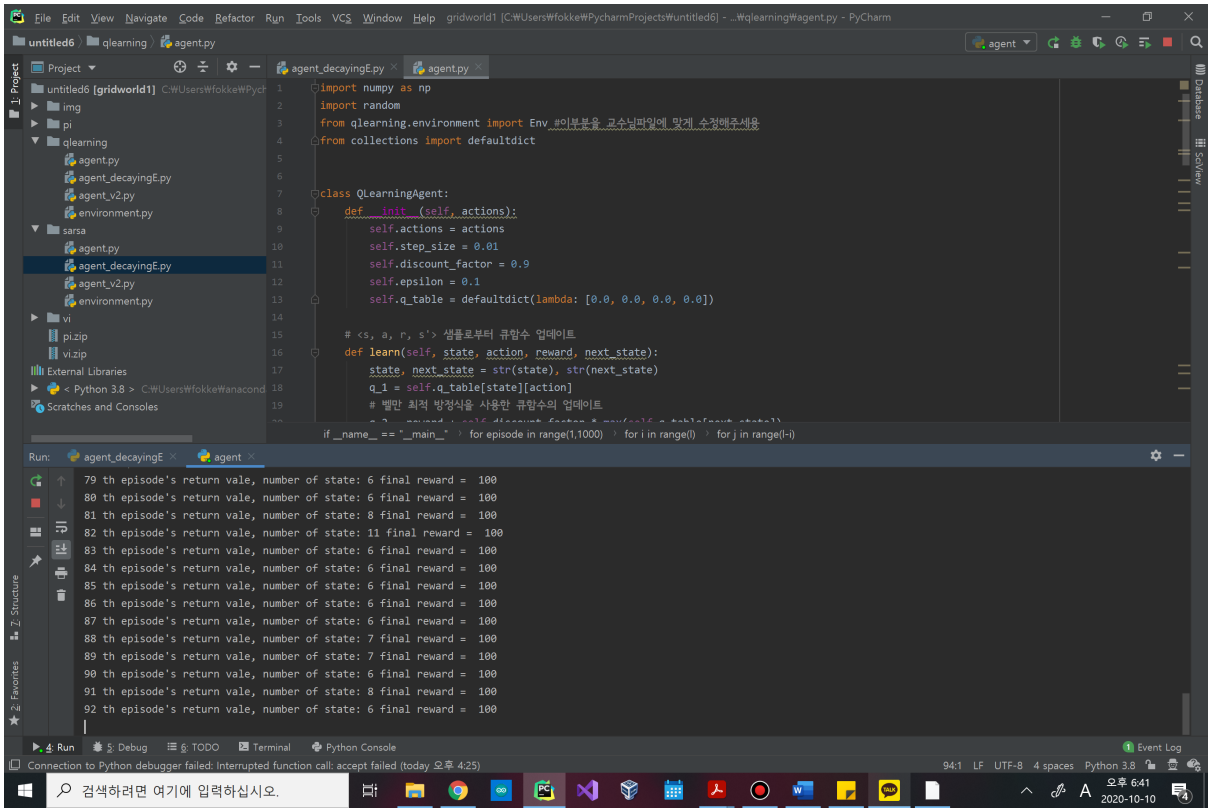


위에 e을 안 줄여줬을 때와 달리 episode가 계속 진행됨에도 갇힘 현상 또한 일어나지 않았다. 이유는 갇힘 현상이란 agent가 특정상황에서 random한 행동에 의해서 – reward를 가지게 되었을 경우 이전 state의 큐함수가 -값을 가지는 경우에 잘못 판단하게 되어 이런 현상이 일어나게 되는데 e값을 iteration 할때마다 줄여주는 경우 episode가 진행됨에 따라 agent가 random하게 행동하지 않고 점점 더 greedy하게 행동하게 됨으로 아주 운이 나쁜 경우를 제외하고 계속 최적의 경로로 도착하게 된다.

2-1. Q-learning with no decaying epsilon

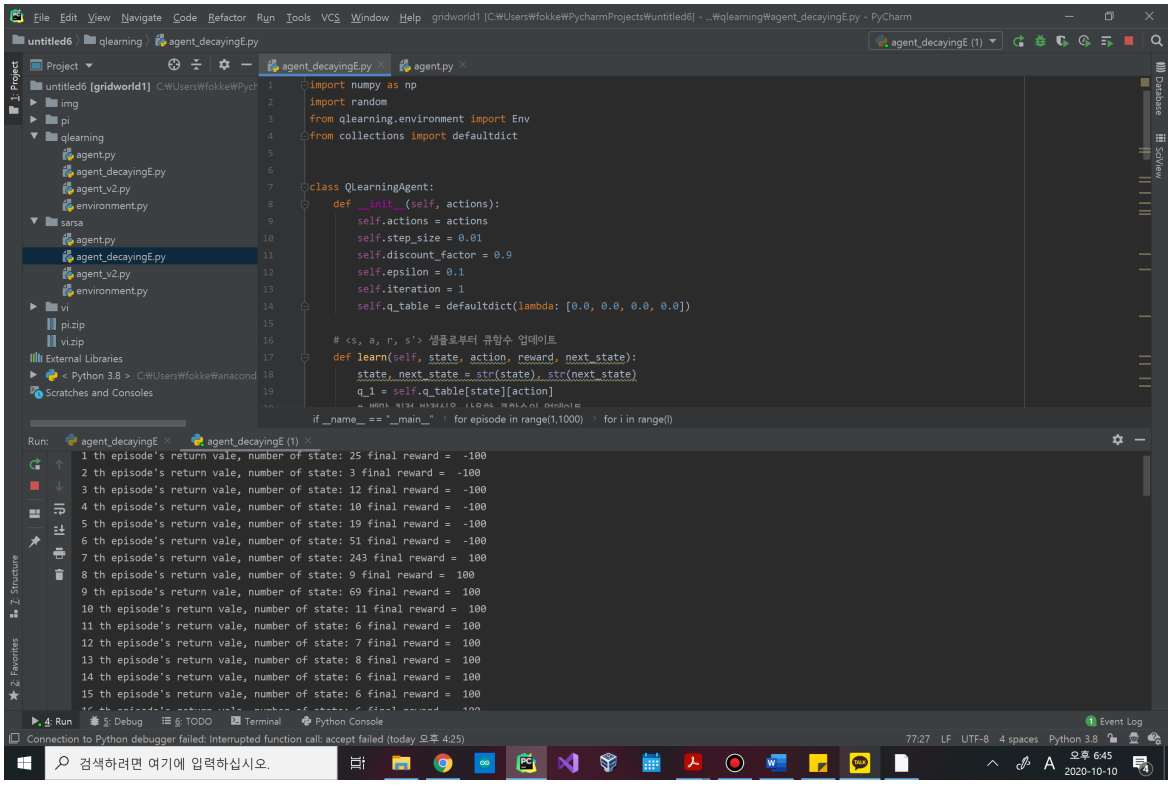


약 19번째 episode부터 최적의 경로를 슬슬 찾아가기 시작한다.

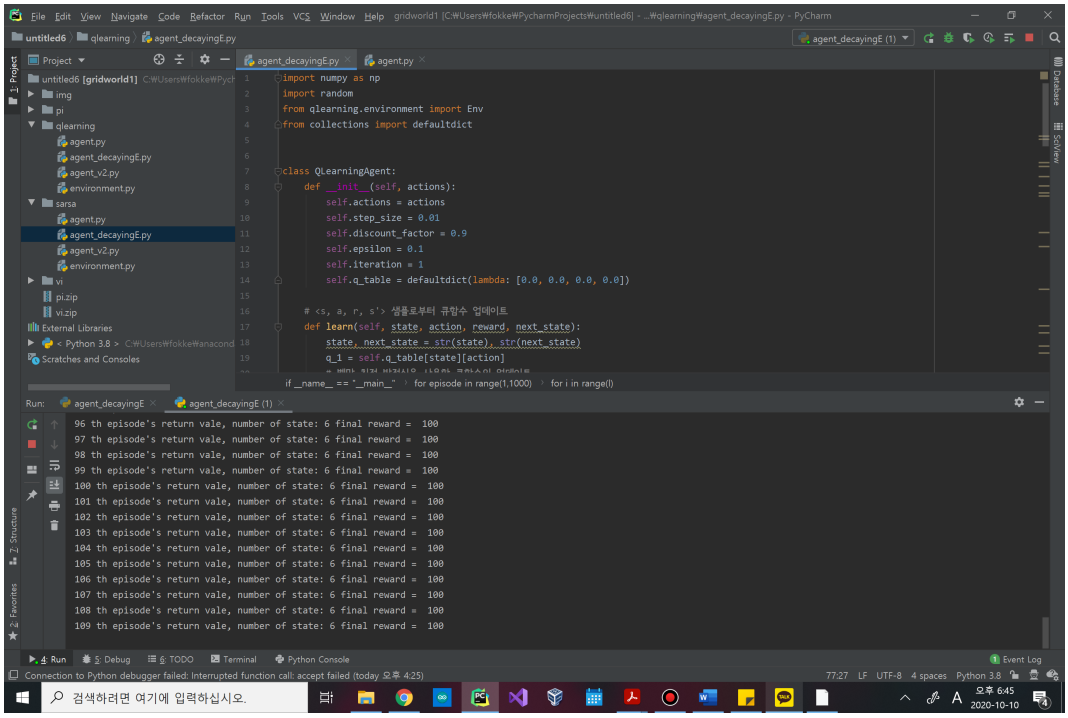


하지만 계속해서 높은 e값 때문에 탐험을 하여 가-끔씩 오래 걸리거나 – reward를 가지는 곳으로 가기도 하는 것을 관찰할 수 있다.

2-2. Q-learning with decaying epsilon



Decaying epsilon을 한 경우에도 최적의 경로를 약 15번째쯤부터 찾아가기 시작한다



하지만 위와 다르게 100번째의 iteration시에 완전히 e 값이 0에 가까운 값이 됨으로 계속해서 최적의 경로를 택해 episode를 수행하게 된다.

이와 같이 e을 계속해서 줄여줌으로서 많은 iteration이후에는 agent가 최적의 경로에 따라 행동한다는 것을 알 수 있다. 하지만 만약 e를 너무 빠르게 줄이게 된다면 explore를 너무 안하여 map전체를 탐색하지 않는 경우가 생기기 때문에 적절하게 줄여주는 것이 좋은 방향이라고 생각한다.