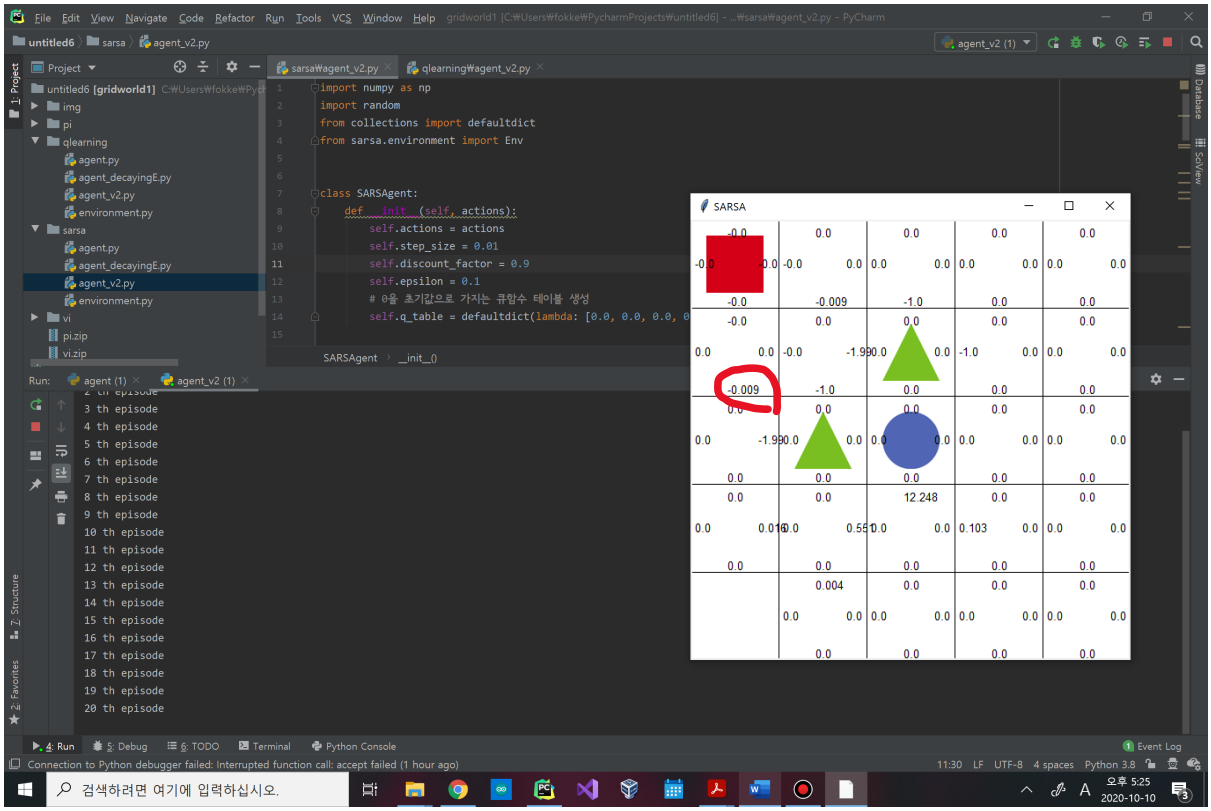
그리드월드의 살사에서는 충분한 탐험을 위해 E-greedy정책을 사용합니다. 또한 SARSA의 특징으로 앞으로의 2번의 행동을 정책에 반영한다는 특징이 있습니다. 이가 의미하는 바는 만약 탐욕정책에 따라 s'으로 오른쪽을 향하여 가는 action은 greedy하게 잘 선택했다고 가정해봅시다. 그다음 s'에서 다시 행동(a')을 하였는데 E-greedy에 의해 greedy한 행동이 아닌 random한 행동을 했고 그 행동이 하필 reward가 -인 나쁜 행동이였다고 가정해봅니다. 그렇다면 Q(s',a')은 낮은 값을 가지게 되고 이는 곧 Q(s,a)의 큐함수 또한 낮아지게 되는 결과를 가져오게 됩니다. 이로 인해 agent가 초기상태로 돌아갔을 때 random성으로인해 낮아지게 된 오른쪽으로 가는 행동을 꺼려하게 되고, 이는 곧 agent가 갇히게 되는 현상이 발생합니다.

실행결과를 캡쳐하여 알아보겠습니다.( 이전 실습문제의 Return값이 console에 뜨는 것이 너무 많아, 이를 지우고 몇번째 episode인지만 console에 출력했습니다.)

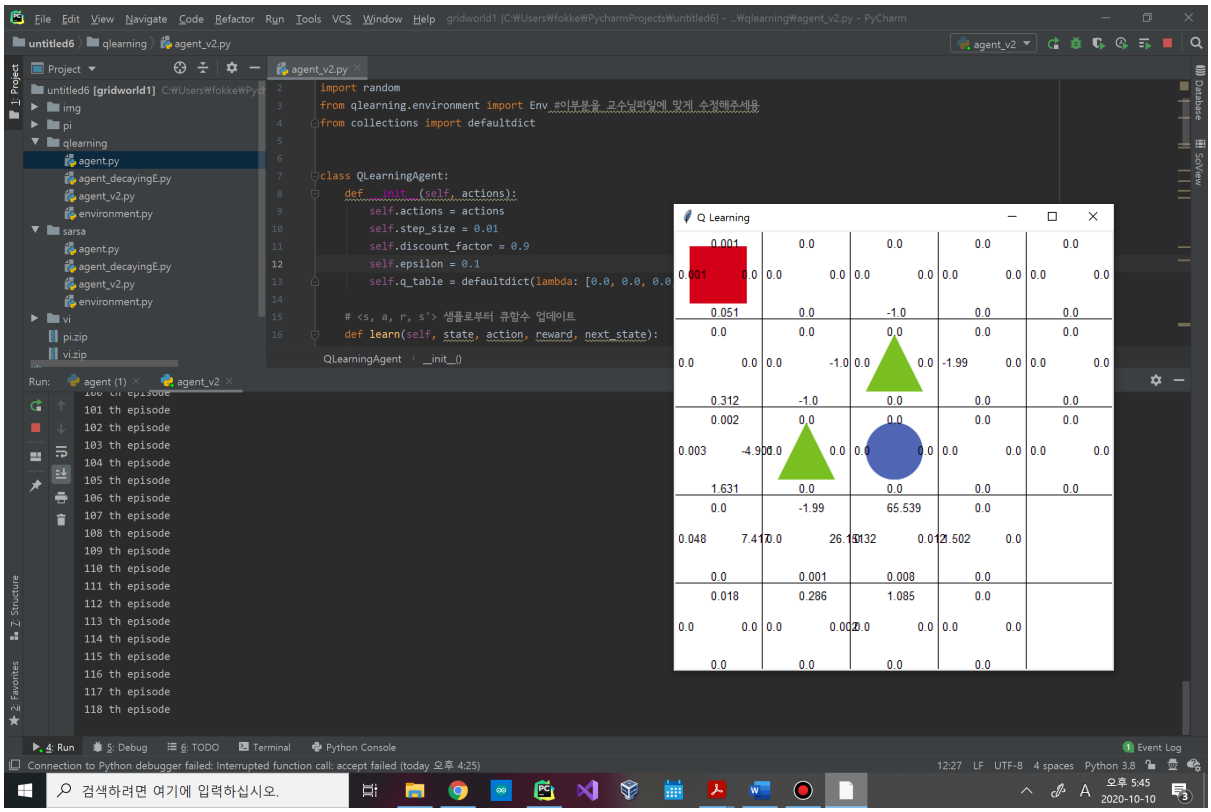



SARSA문제.mp4

SARSA의 실행 영상과 캡쳐화면입니다. 캡쳐화면을 보면 20번째 episode까지 가서 멈추는 것을 볼 수 있습니다. 또한 빨간 동그라미 부분을 보시면 큐함수값이 -0.0009로 바뀌어 있는 것을 볼 수 있습니다. 이로인해 agent는 계속 갇히게 되어 목표를 찾아갈 수 없게 되었습니다.

이 문제를 해결한 방법이 바로 Q-Learning입니다. 큐러닝은 같은 상황에서 살사와 달리 s'을 알게 되면 그 상태에서 가장 큰 큐함수를 현재 큐 함수의 업데이트에 사용합니다. 실제 다음 상태 s'에

서 다음 행동을 해보는 것이 아닌 s'에서 가장 큰 큐함수를 가지고 업데이트를 하여 이를 위해 필요한 샘플은 <S,A,R,S',A'>가 아닌 <S,A,R,S'>입니다. 살사에서는 큐함수를 업데이트 하기위해 벨만 기대방정식을 사용했지만 큐러닝에서는 큐함수를 업데이트 하기위해 벨만 최적방정식을 사용한다는 것 입니다.




보시는 것과 같이 100번의 episode가 넘어가도 위와 같은 현상없이 잘 찾아가는 것을 확인 할 수 있습니다.