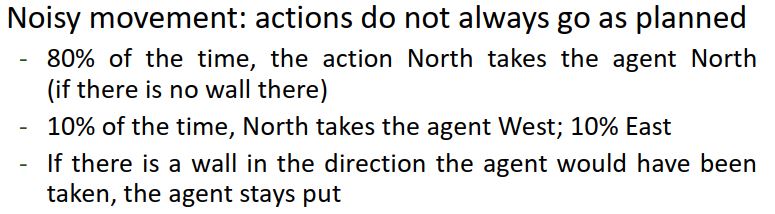
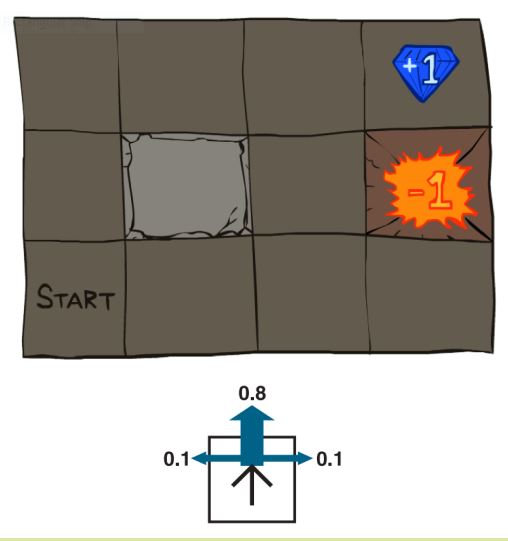
به نام خدا

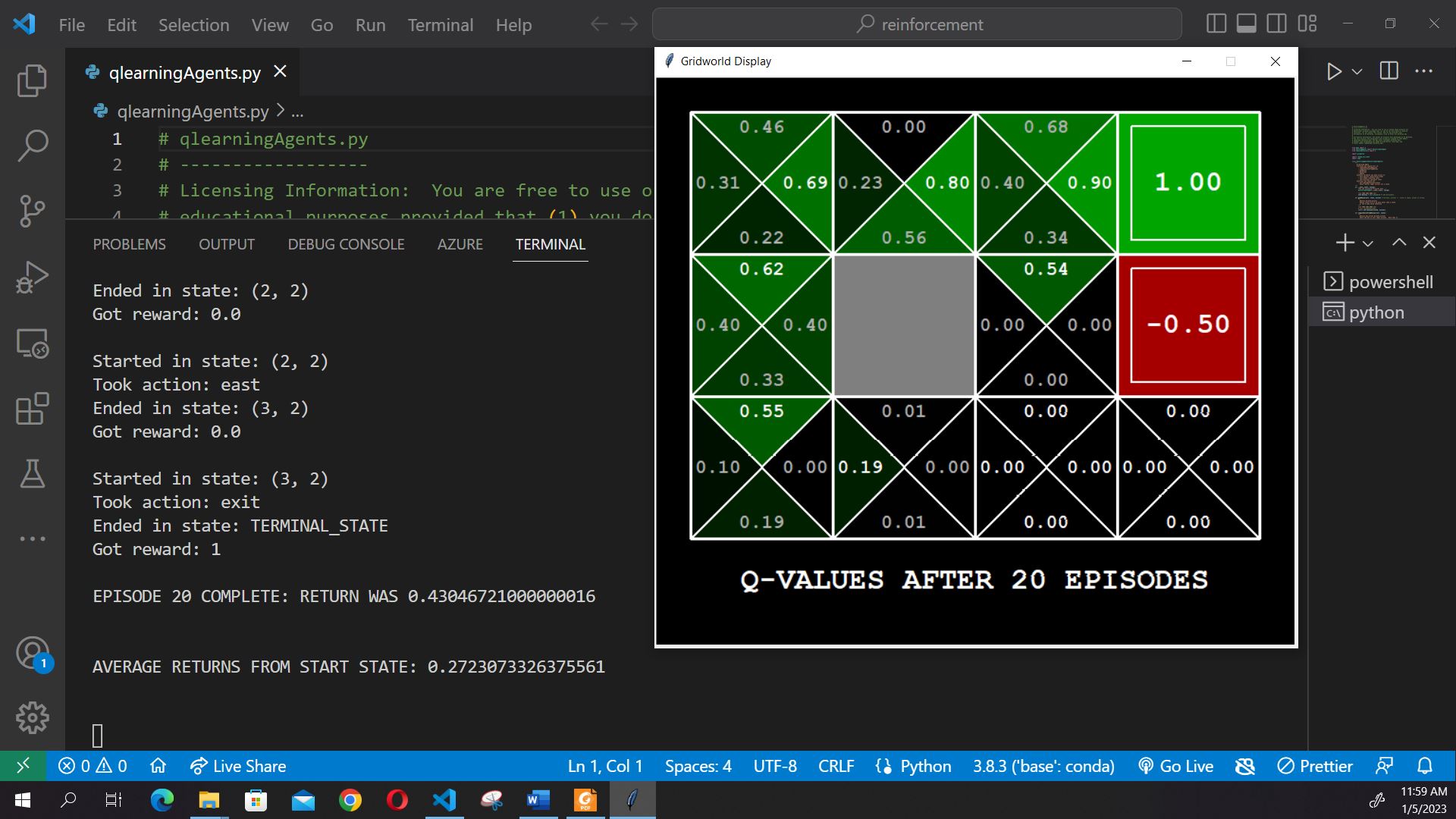
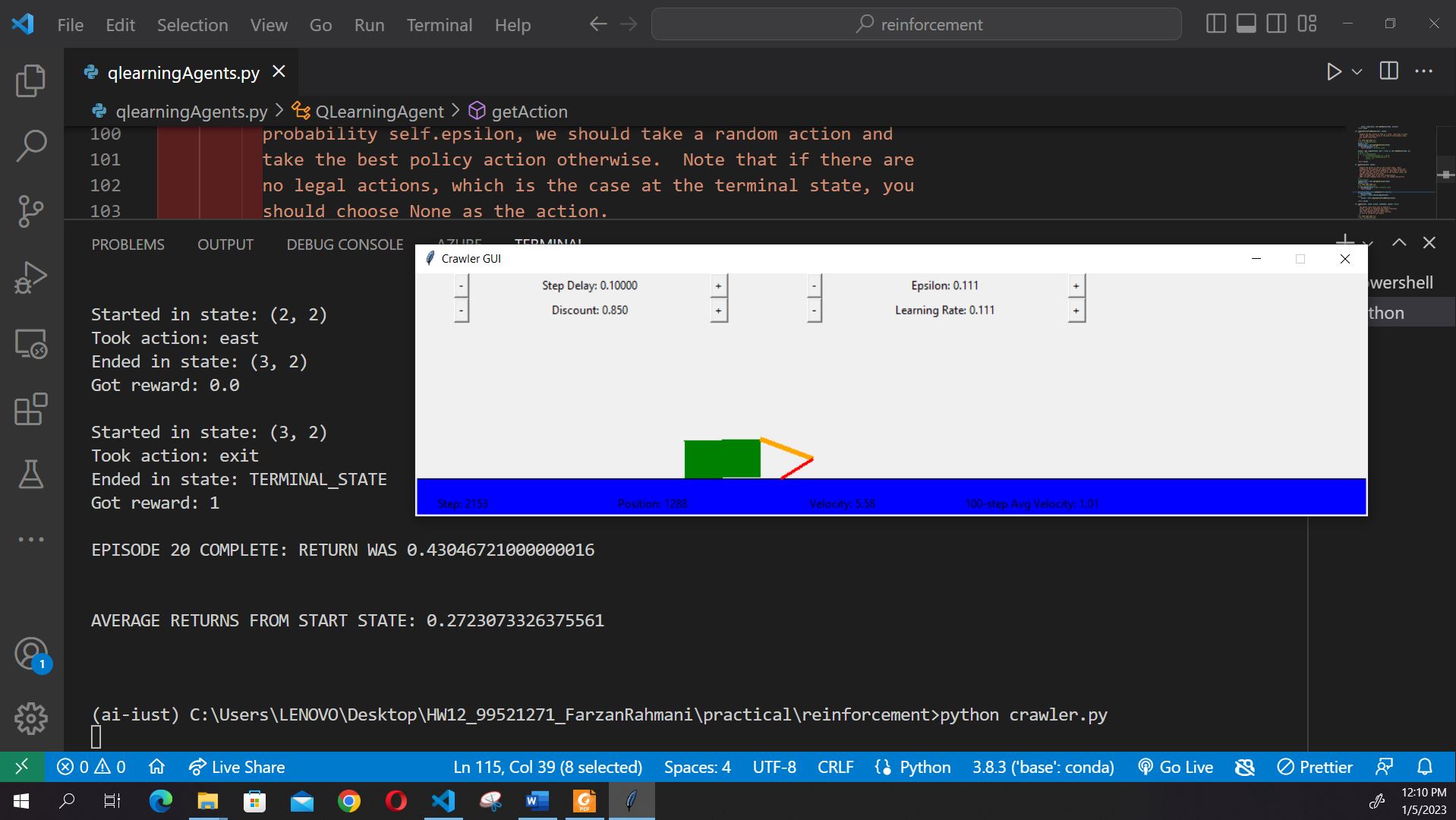
# فرزان رحمانی 99521271

## تمرین دوازدهم هوش مصنوعی و سیستم های خبره قسمت عملی (سوال 2 – پیاده سازی Q-learning)

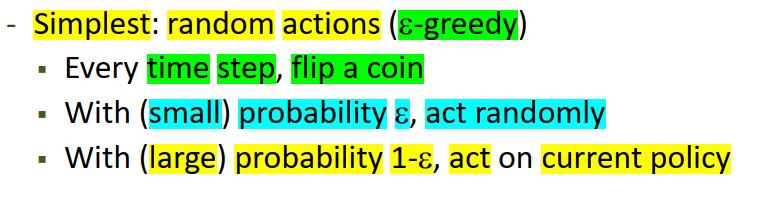
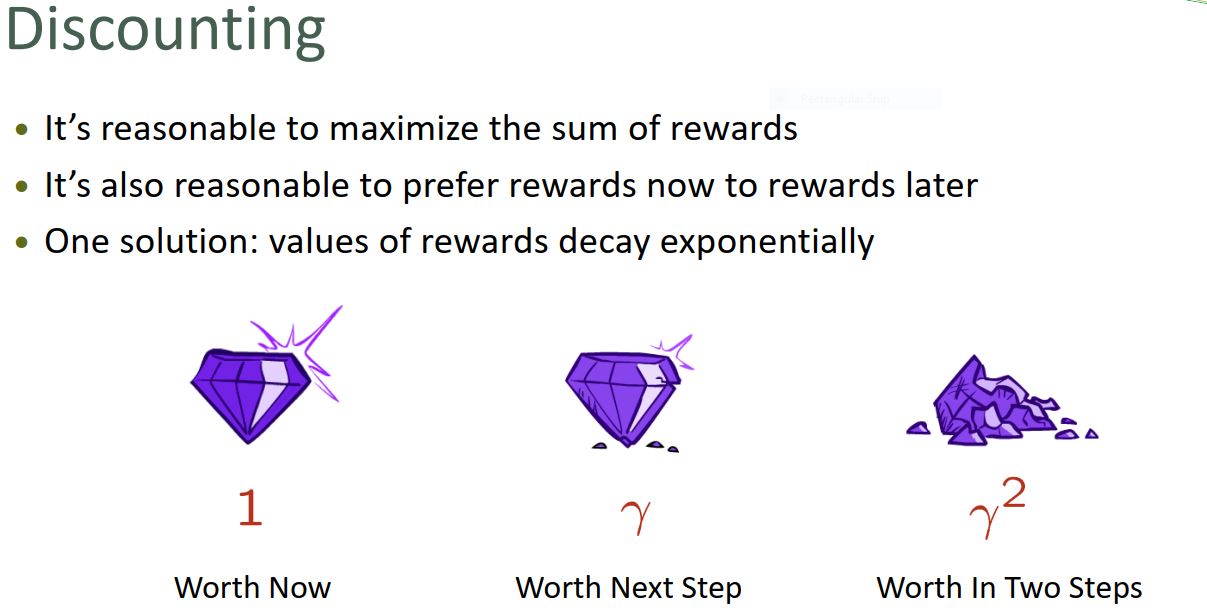
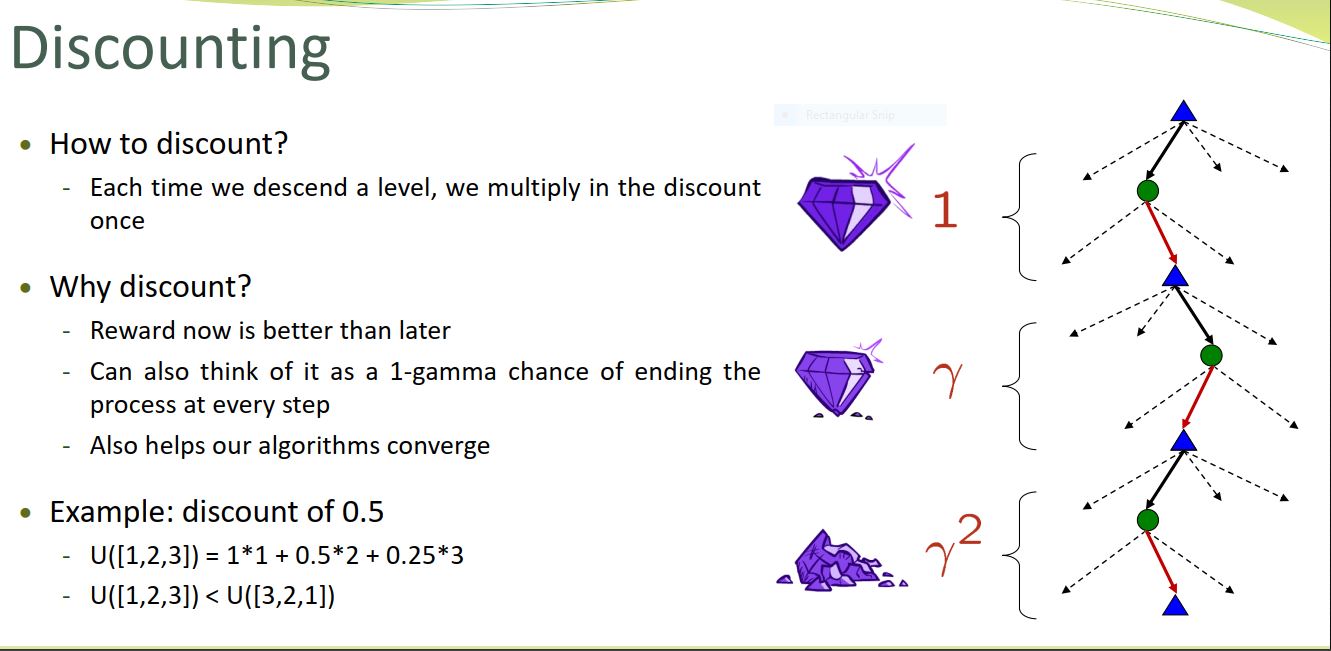
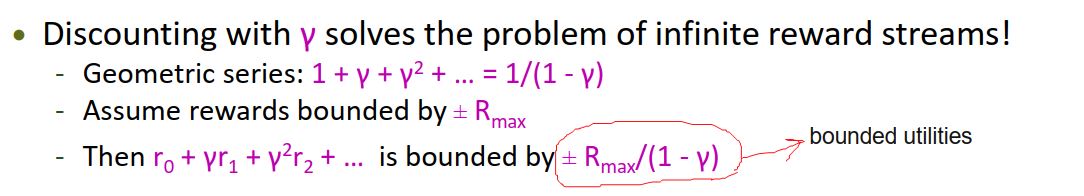
1 .2 ارزیابی پیاده سازی

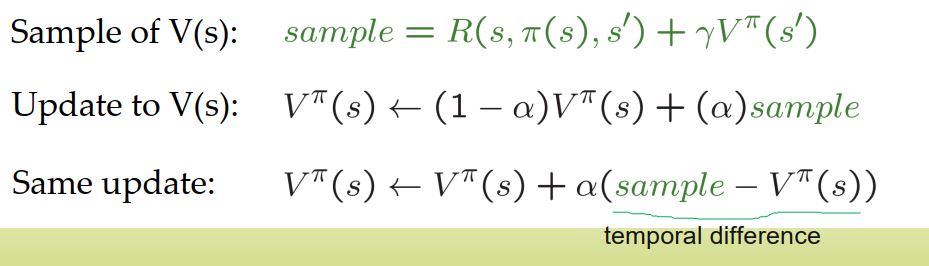
به چه علت است که همواره agent مطابق جهتی که شما میدهید حرکت نمیکند؟   
زیر محیط بازی GridWorld محیط non-deterministic(stochastic) هست و transition model آن احتمالی می باشد. به عبارتی دارای noise = 0.2 می باشد.  
  


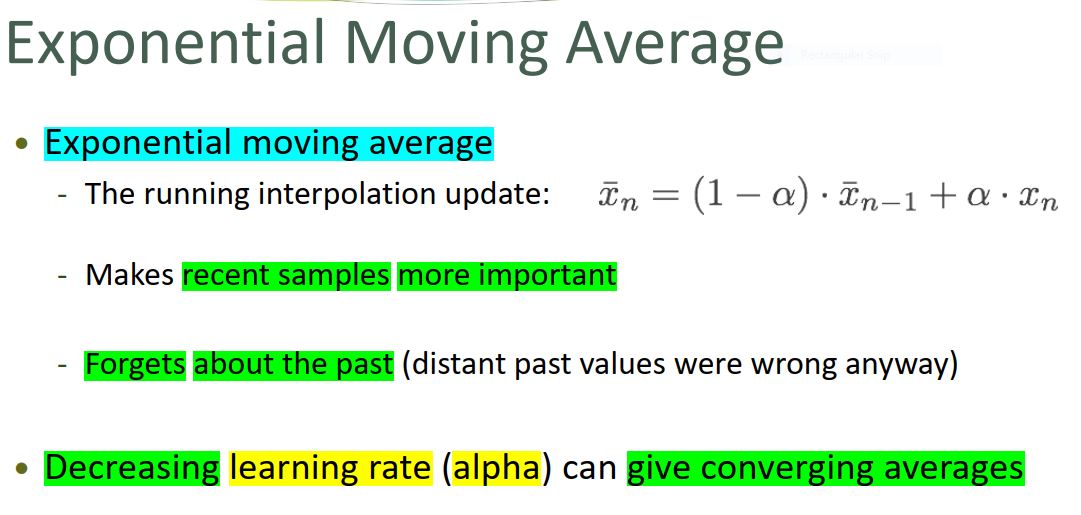


گزارش نتیجه بدست آمده از اجرا:  
  
  
  
با تغییر iteration ها از 5 به 20 agent بیشتر محیط را یاد میگیرد و Q های بهتری داریم. اما هنوز کامل و همگرا نشده است و با افزایش iteration ها بهتر محیط را یاد میگرد. همان طور که دیده می شود بیشتر اطراف حالت پایانی بهینه کشف شده است چرا که از computeActionFromQValues استفاده کرده ایم. البته به علت استفاده از epsilon-greedy و flipCoin به انتخاب های جدید نیز امکان انتخاب شدن داده ایم. در این شکل QValue های هر state را می بینیم و برای استخراج سیاست از بین آنها max را انتخاب می کنیم. با توجه به مقادیر موجود عامل ما ابتدا بالا می رود و بعد از دوبار بالا رفتن به راس می رود. چرا که وجود مانع باعث می شود اگر طبق اکشن پیش نرفتیم و وجود noise باعث نشود داخل آتش بیفتیم.  
  


Step Delay: با افزایش آن سرعت ربات کند تر می شود و با کاهش آن ربات سریع تر عمل می کند.

Epsilon(ε): احتمال حرکت تصادفی و نه با توجه به سیاست فعلی و QValue هایی که داریم و با افزایش آن Exploration رو محیط بیشتر می شود و اکشن های تصادفی بیشتری داریم.  
  
  
  
  
  
  
  
  
  
Discount(γ): باعث می شود تا به پاداش و استیت های فعلی نسبت به پاداش و استیت های آینده ارزش بیشتری بدهیم و از یه حدی به بعد صفر شوند.  
  
  
  
  


Learning Rate(α): مشخص می کند که برای میانگین QValue ها به نمونه های جدید چه ارزشی بدهد و هر چه بیشتر باشد نمونه های جدید ارزش بیشتری خواهند داشت و گذشته اش را فراموش می کند.  




پایان