تمرین دوم یادگیری ماشین

داریوش حسن پور

۹۳۰۸۱۶۴

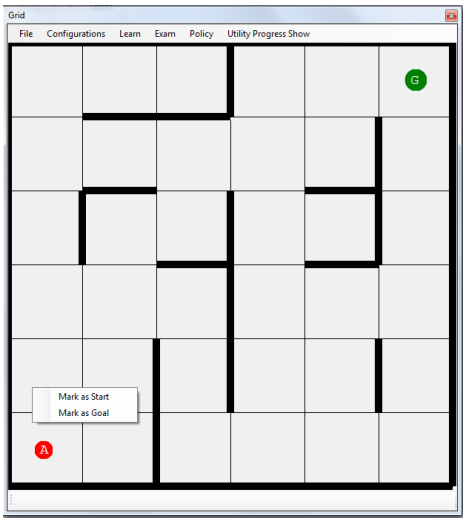
پاییز ۱۳۹۳

1 مقدمه

این تمرین در مورد پیاده سازی الگوریتم های $Q(\lambda)$[1] و $SARSA(\lambda)$[2] ولی به علت عدم توجه بنده به عبارت λ در هنگام خواندن تعریف تمرین در ابتدا بنده الگوریتم های $Q(s,a)$[3] و $SARSA(s,a)$[4] علاوه بر الگوریتم های $Q(\lambda)$ و $SARSA(\lambda)$ را نیز پیاده سازی کرده ام.

2 معرفی برنامه

برنامه به زبان C# تحت قالب کاری .Net 4.0 تحت محیط Visual Studio 2010 نوشته شده است. برنامه دارای ظاهری بسیار پویا است که قابلیت تغییر محیط و جابجایی موقعیت های عامل و هدف و همچنین پارامتر های یادگیری را به صورت گرافیکی دارا میباشد.



عکس ۱: یک نمایشی از محیط برنامه که با انتخاب رو خطوط میتوان آنها را به بلوک تبدیل کرد و برعکس؛ و همچنین با راست کلیک کردن بروی خانه ها امکان انتخاب موقعیت های عامل و هدف را وجود دارد.

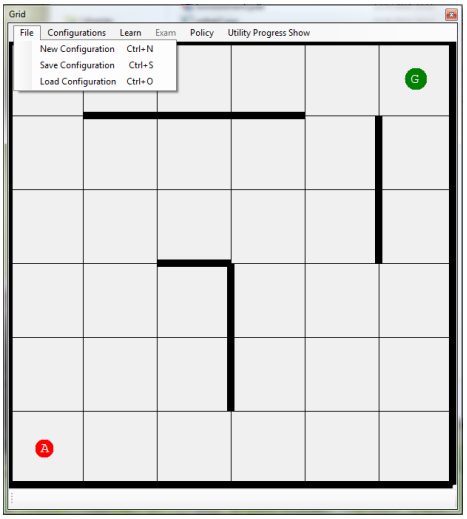
1.2 معرفی منوهای برنامه

در این قسمت به معرفی منوهای برنامه میپردازم.

File 1.1.2

در منوی فایل امکان ایجاد و ذخیره کردن و بازنشانی ساختار خام محیط گذاشته شده است.

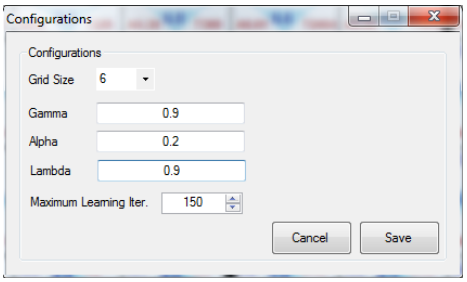
[1] http://webdocs.cs.ualberta.ca/ sutton/book/ebook/node78.html
[2] http://webdocs.cs.ualberta.ca/ sutton/book/ebook/node77.html
[3] http://artint.info/html/ArtInt_265.html
[4] http://artint.info/html/ArtInt_268.html



عکس ۲: منوی فایل

Configurations 2.1.2

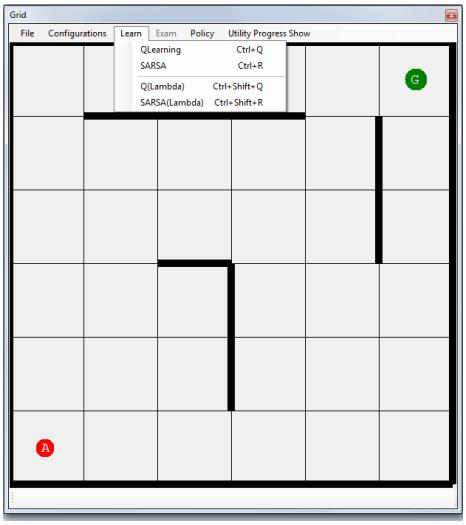
با انتخاب این منو به تنظیمات برنامه که مربوط به پارامتر های یادگیری و محیط مربوط میشود میرویم.



عکس ۳: منوی تنظیمات برنامه؛ توجه شود که اندازه شبکه همیشه مربعی در نظر گرفته شده است.

Learn 3.1.2

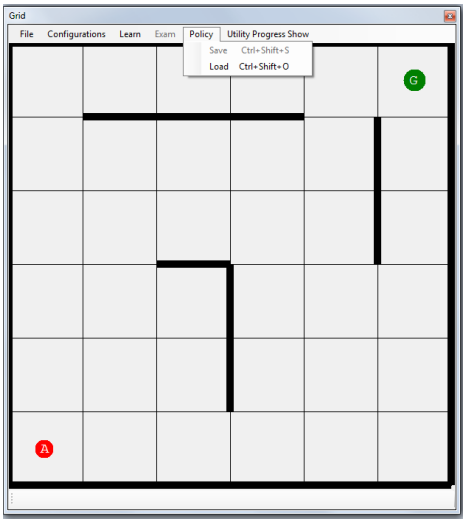
در این منو میتوانیم یکی از ۴ الگوریتم نوشته شده را برای شبکه طراحی شده آموزش داد و نتایج بطور گرافیکی نمایش خواهند یافت که در قسمت های بعدی در مورد چگونگی تفسیر این نتایج بحث خواهد شد.



عکس ۴: با انتخاب یکی از چهار الگوریتم؛ الگوریتم انتخاب شده شروع به یادگیری شبکه خواهد کرد.

Policy 4.1.2

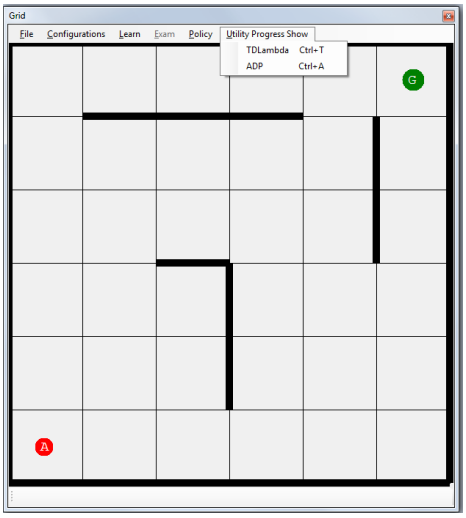
این منو برای ذخیره سازی و بازنشانی سیاست یادگرفته شده و سایر اطلاعاتی که پس از یادگیری شبکه بدست میاید از قبیل مقادیر سودمندی موقعیت ها و دیگر اطلاعات مرتبط مورد استفاده است که برای راحتی کار در منویی جدا قرار داده شده اند.



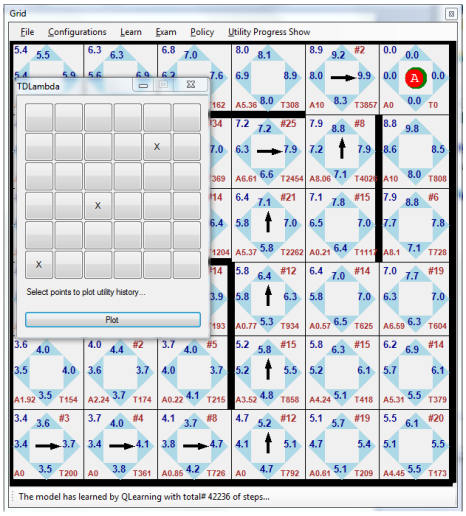
عکس ۵: بعد از یادگیری سیاسیت و میزان سودمندی موقعیت ها میتوان آنها را ذخیره و در دفعات بعد بازنشانی کرد.

Show Progress Utilities 5.1.2

برنامه در هنگام یادگیری میزان سودمندی موقعیت ها تاریخچه ای از نحوه تغییر مقادیر سودمندی تمامی موقعیت ها ذخیره میکند. که توسط این منو میتوان نحوه ی رشد یا عدم رشد سودمندی هر یک از خانه ها را به صورت جدا یا باهم(برای مقایسه) مشاهده کرد.



عکس ۶: با انتخاب هر یک از الگوریتم های یادگیرنده ی میزان سودمندی خانه ها میتوانیم شاهد مشاهده ی نحوه ی تغییر سودمندی موقعیت باشیم.



عکس ۷: با انتخاب خانه های مد نظر(با هم یا به بطور تکی) می توانیم به مقایسه نحوه ی تغییر میزان سودمندی خانه ها بپردازیم.

3 نحوه ی نمایش داده ها

در نحوه ی حرکت عامل در برنامه تعریف شده است که عامل میتواند در هر موقیت تصمیم به انتخاب یکی از ۵ حرکت { شمال؛ شرق؛ جنوب؛ غرب و حفظ موقعیت } نماید. که در حفظ موقعیت عامل بی حرکت می ماند.

پس از یادگیری سیاست بهینه و سودمندی حالات داده های مرتبط با هر موقیت به شرح زیر در همان موقعیت نمایش داده میشوند:

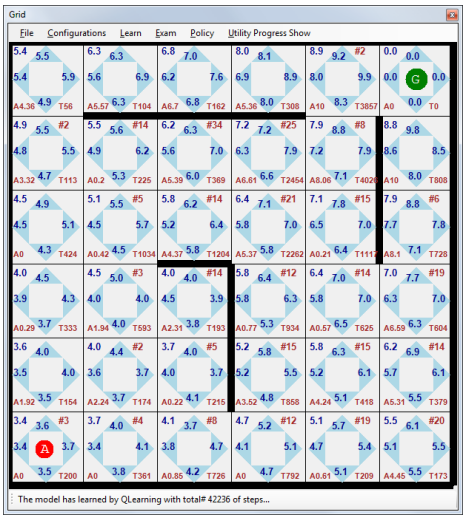
در هر موقعیت مقدار سیاست بهینه یادگرفته شده برای چهار جهت اصلی در فلش های آبی رنگ مرتبط با همان جهت نمایش

داده شده است.
گوشه بالایی سمت چپ هر موقعیت مقدار سیاست بهینه مربوط حرکت «حفظ موقعیت» میباشد.
گوشه بالایی سمت راست هر موقعیت این حقیقت را مشخص میکند که الگوریتم در آخرین مرحله یادگیری چند بار به خانه ی مورد نظر در رجوع کرده است.
مقدار گوشه پایینی چپ هر خانه مقدار سودمندی محاسبه شده با استفاده از برنامه نویسی پویای وفقی برای آن موقعیت را نمایش میدهد و متعاقبا گوشه پایینی راست هر خانه مقدار سومندی آن خانه که توسط روش $TD(\lambda)$ حساب شده است را نمایش میدهد. [5]

4 اجرای برنامه

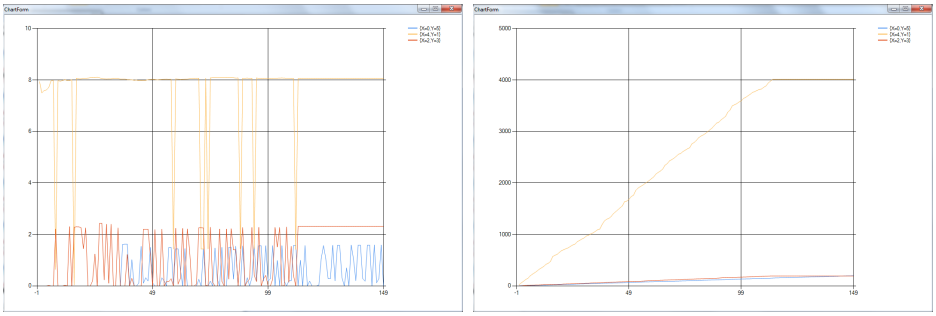
شبکه درخواست شده در متن تکلیف توسط ۴ الگوریتم معرفی شده در مقدمه این متن مورد یادگیری قرار گرفته است. که شرح نتایج آنها به صورت زیر است.

1.4 $Q(s,a)$



عکس ۸: سیاست و میزان سودمندی های یادگرفته شده توسط الگوریتم $Q(s,a)$

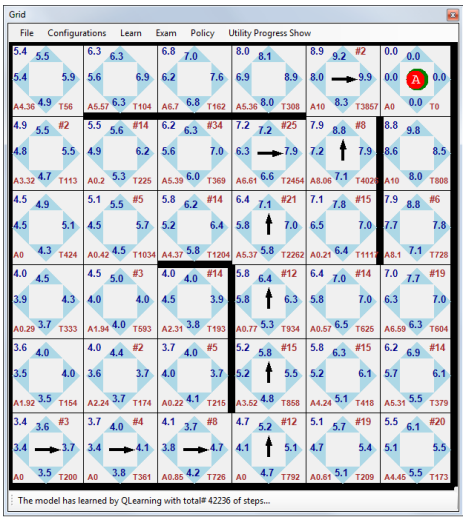
[5] توجه شود که در برنامه نویسی این برنامه مبدا مختصات موقعیت های شبکه را گوشه بالایی سمت چپ در نظر گرفته شده است ولی با توجه به متن تمرین داده شده میتوان این را نتیجه گرفت که مبدا مختصات گوشه پایینی سمت چپ است؛ بنابراین ایندکس های خانه های چاپ شده با توجه به مبدا مختصات برنامه یعنی گوشه بالایی سمت چپ در نظر گرفته شده است.



الف: روند تغییر مقادیر سودمندی یادگرفته شده توسط روش $TD(\lambda)$

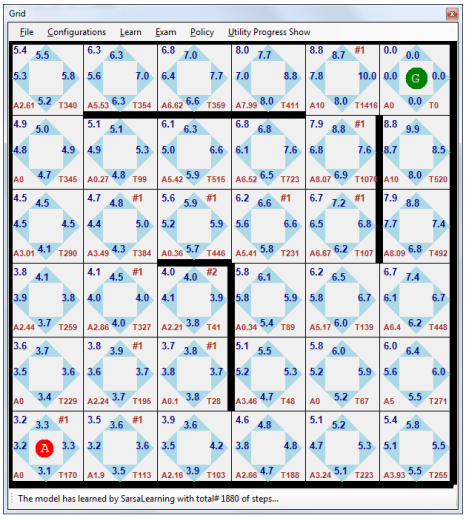
ب: روند تغییر مقارایر سودمندی یادگرفته شده توسط روش ADP

عکس ۹: روند تغییر مقدریر سودمندی یادگرفته شده توسط روش های $TD(\lambda)$ و ADP برای خانه های $(1,1)$, $(3,3)$ و $(5,5)$

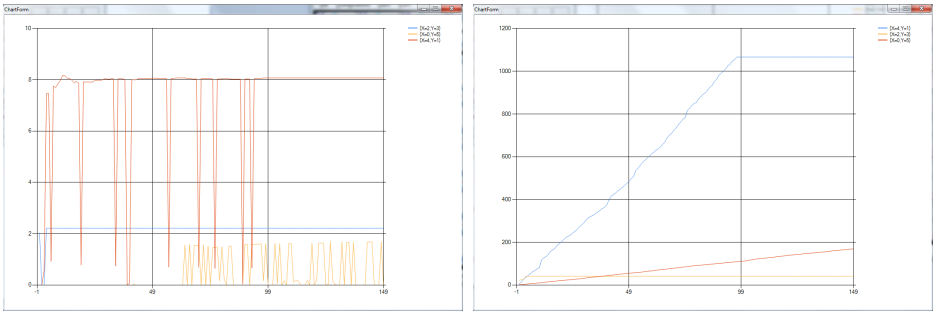


عکس ۱۰: مسیر طی شده توسط عامل با توجه به سیاست یادگرفته شده توسط الگوریتم $Q(s,a)$

4.2 $SARSA(s, a)$



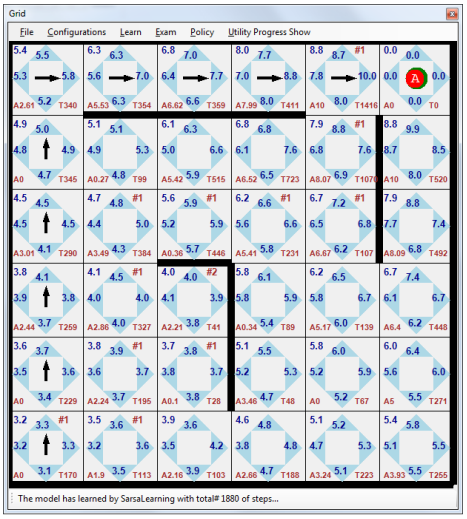
عکس ۱۱: سیاست و میزان سودمندی های یادگرفته شده توسط الگوریتم $SARSA(s, a)$



الف: روند تغییر مقادیر سودمندی یادگرفته شده توسط روش $TD(\lambda)$

ب: روند تغییر مقاریر سودمندی یادگرفته شده توسط روش ADP

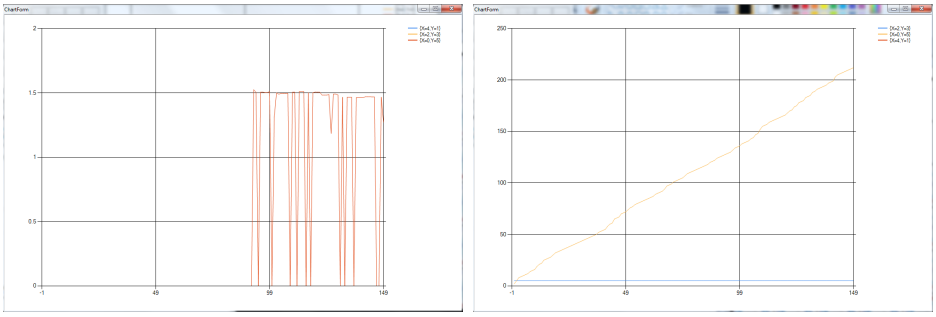
عکس ۱۲: روند تغییر مقدریر سودمندی یادگرفته شده توسط روش های $TD(\lambda)$ و ADP برای خانه های $(1, 1)$, $(3, 3)$ و $(5, 5)$



عکس ۱۳: مسیر طی شده توسط عامل با توجه به سیاست یادگرفته شده توسط الگوریتم $SARSA(s, a)$

4.3 $Q(\lambda)$

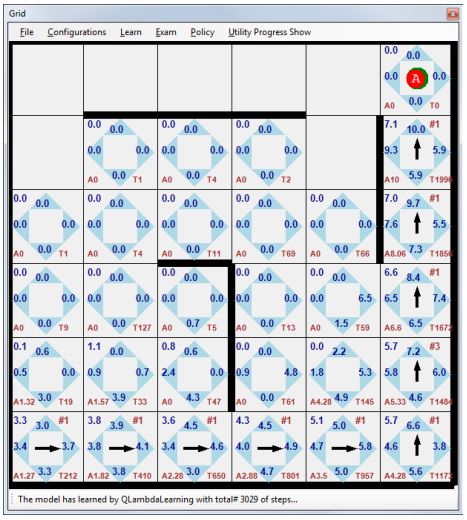
عکس ۱۴: سیاست و میزان سودمندی های یادگرفته شده توسط الگوریتم $Q(\lambda)$



الف: روند تغییر مقادیر سودمندی یادگرفته شده توسط روش $TD(\lambda)$

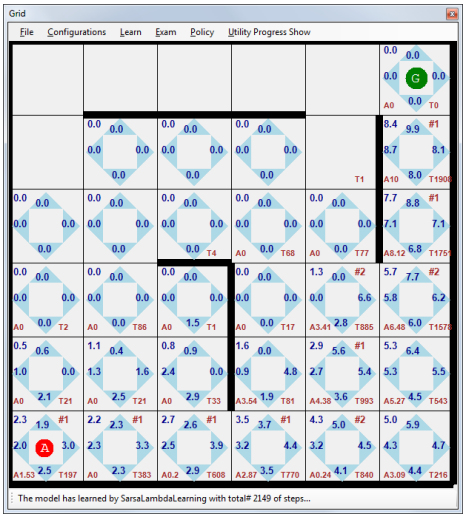
ب: روند تغییر مقارایر سودمندی یادگرفته شده توسط روش ADP

عکس ۱۵: روند تغییر مقدریر سودمندی یادگرفته شده توسط روش های $TD(\lambda)$ و ADP برای خانه های $(1,1)$, $(3,3)$ و $(5,5)$

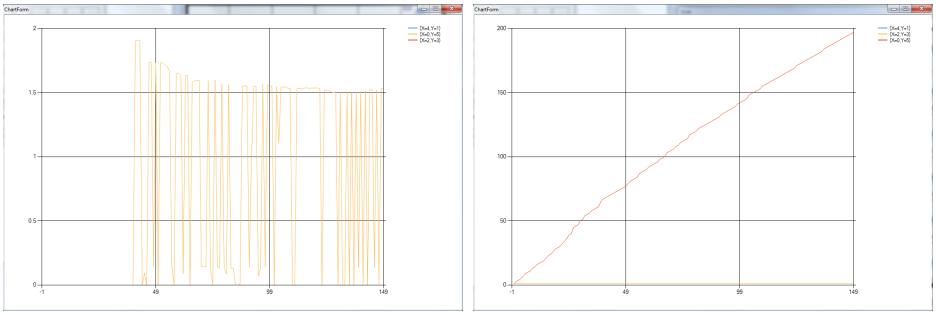


عکس ۱۶: مسیر طی شده توسط عامل با توجه به سیاست یادگرفته شده توسط الگوریتم $Q(\lambda)$

4.4 $SARSA(\lambda)$



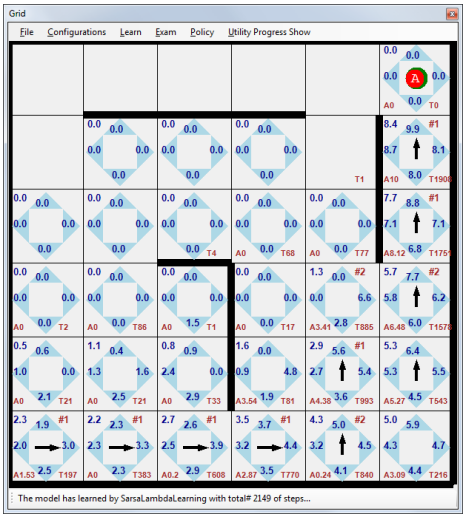
عکس ۱۷: سیاست و میزان سودمندی های یادگرفته شده توسط الگوریتم $SARSA(\lambda)$



الف: روند تغییر مقادیر سودمندی یادگرفته شده توسط روش $TD(\lambda)$

ب: روند تغییر مقادیر سودمندی یادگرفته شده توسط روش ADP

عکس ۱۸: روند تغییر مقدریر سودمندی یادگرفته شده توسط روش های $TD(\lambda)$ و ADP برای خانه های $(1,1)$, $(3,3)$ و $(5,5)$



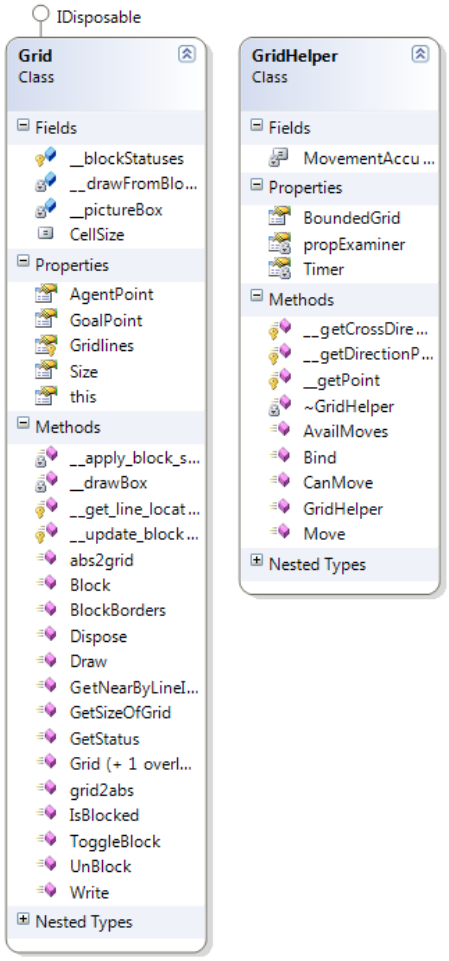
عکس ۱۹: مسیر طی شده توسط عامل با توجه به سیاست یادگرفته شده توسط الگوریتم $SARSA(\lambda)$

5 توضیحاتی در مورد برنامه نویسی شی گرا و اشیا برنامه

همان طور که میدانیم C# یک زبان مبتنی بر برنامه نویسی شی گرا میباشد. بنده برای پیاده سازی این برنامه در ابتدا به نوشتن ۲ عدد کتابخانه اقدام کردم یکی برای مدل کردن محیط و اعملیات های مرتبط با محیط و دیگری برای الگوریتم های یادگیری محیط میباشد. برنامه اصلی(اجرایی) از هر دوی این کتابخانه ها استفاده میکند و مدل انتزاعی محیط را به محیط گرافیکی تبدیل میکند. و با تحویل محیط انتزاعی به الگوریتم های یادگیری تقویتی نوشته شده در دیگر کتابخانه یک هماهنگی و ارتباط بین این دو ایجاد میکند.

1.5 کتابخانه محیط

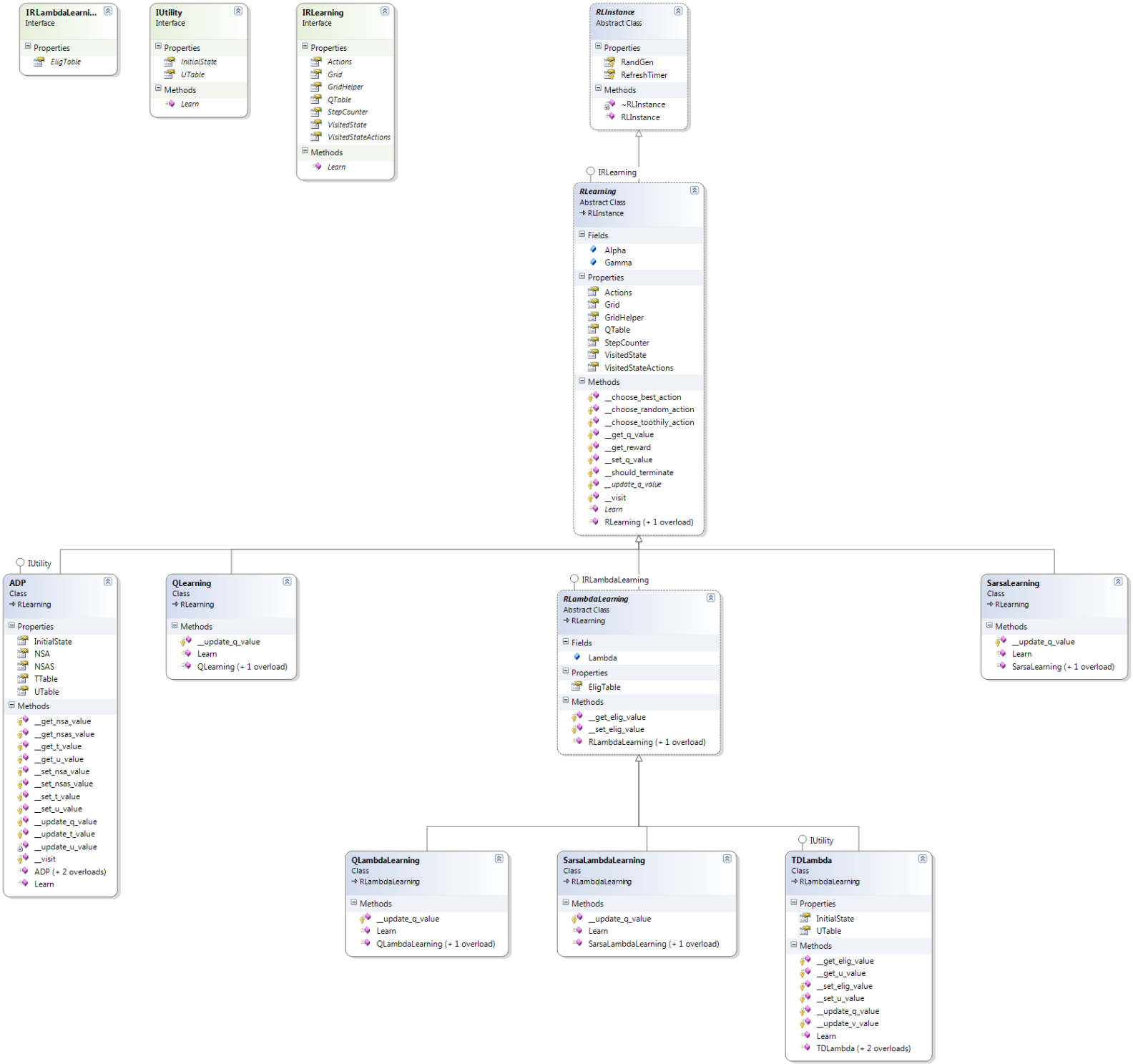
این کتابخانه امکانات کافی برای انتزاع محیط واقعی را در اختیارمان میگذارد. و همچنین قابلیت تبدیل مدل گرافیکی به مدل انتزاع وبرعکس را دارد. که در هنگام ذخیره و بازنشانی شبکه کاربرد دارد. نمودار کلاس های مرتبط با این کتابخانه در شکل شماره ۲۰ آمده است.



عکس ۲۰: نمودار کلاسی کتابخانه ی محیط

2.5 کتابخانه یادگیری تقویتی

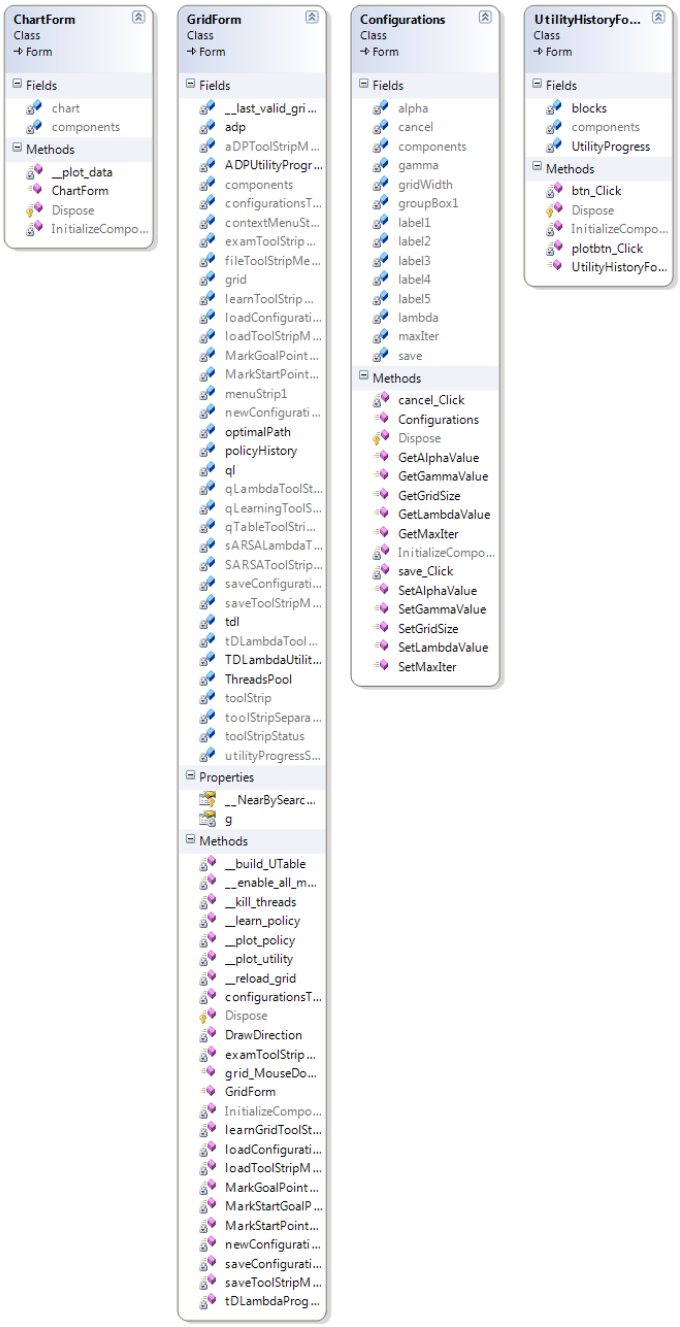
در این کتابخانه صرفا پیاده سازی الگوریتم های یادگیری تقویتی که در بخش مقدمه معرفی شده اند آمده اند. این کتابخانه از کتابخانه محیط برای اعمال امور مربوط به محیط از قبل شبیه سازی کردن جابجایی انتزاعی عامل در طی روند یادگیری و ... استفاده میکند. به علت در نظر گرفتن طراحی صحیح از نحوه ی اشتقاق کلاس های این کتابخانه؛ در پیاده سازی هر یک از الگوریتم ها فقط کافی است که متد های Learn() و ___update_q_value() مربوط به آن الگوریتم ها را پیاده سازی کنیم. مابقی مسایل بطور خودکار انجام خواهند شد. نمودار کلاسی کتابخانه ی یادگیری تقویتی در شکل شماره ۲۱ آمده است.(به نحوه ی اشتقاق گیری و توابع کلاس های پدر توجه شود.)



عکس ۲۱ : نمودار کلاسی کتابخانه ی یادگیری تقویتی

3.5 برنامه ی اجرایی

برنامه اجرایی در واقع پل رابطی بین دو کتابخانه محیط و یادگیری تقویتی میباشد. و همچنین برنامه ی اجرایی است که بستر مناسب برای ظاهری گرافیکی برنامه را فراهم میکند.



عکس ۲۲: نمودار کلاسی برنامه اجرایی