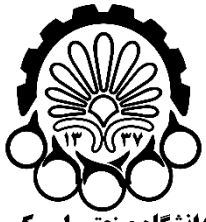
به نام خداوند بخشنده و مهربان



درس هوش مصنوعی

گزارش پروژه

گزارش بازی تخاصمانه

استاد درس:

دکتر قطعی

نام دانشجو:

امیررضا رادجو

۹۷۱۳۰۱۸

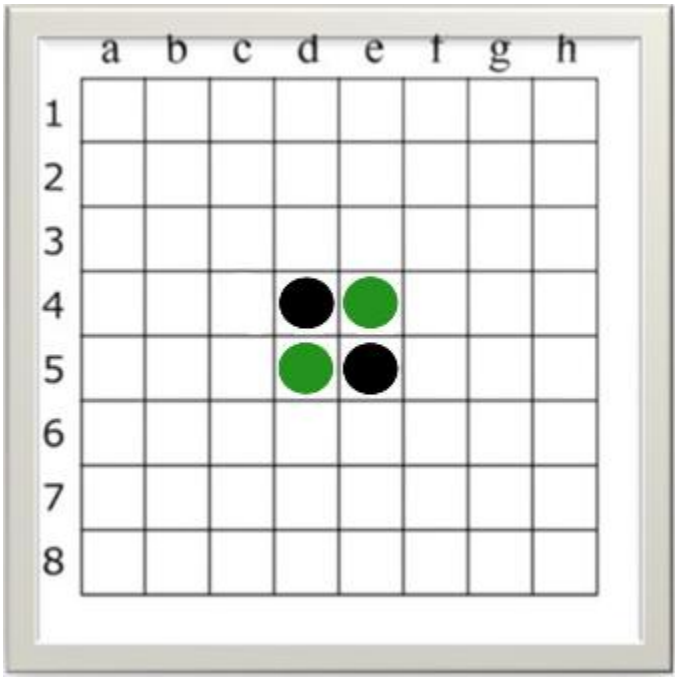
بهار ۱۴۰۰

آموزش بازی اتللو:

بازی اتللو در رده اول بازی های فکری دنیا است که از تعدادی مهره با دو وجه رنگی و یک صفحه بازی تشکیل شده است.

این بازی به صورت دو نفره انجام می شود.

در بازی فکری ۲ نفره بازیکنان با محاصره و تصاحب مهره های یکدیگر وارد رقابتی جدی و آموزنده می شوند و زمان خود را به نحوی خوب سپری می کنند . هم اکنون هزاران نفر در سراسر دنیا ضمن درگیر شدن در جذابتهای این بازی مشهور ، قدرت برنامه ریزی و آینده نگری خود را پرورش می دهند و استراتژی های رقابت کردن و برنده ماندن را طراحی و در عمل اجرا می کنند . شما هم با دوستان و خانواده ی خود سرگرم این بازی شوید و همزمان با اینکه هیجان ناشی از بازی اتللو را می چشید ، توان طراحی و برنامه ریزی ذهنی و عملی ساختن آن صحنه رقابت را بیاموزید .



روش بازی :

برای شروع، چهار مهره مطابق شکل در وسط صفحه به صورت ضربدری، قرار می گیرند. مهره تیره بازی را آغاز می کند. هر یک از دو بازیکن به نوبت یک حرکت انجام می‌دهند. مهره را جایی قرار دهید که یک یا چند مهره حریف را محاصره کند. انجام حرکت به معنی گذاشتن یک مهره (از طرف رنگ خود) در صفحه و محصور کردن یک یا چند مهره حریف در یک یا چند راستا است. در نتیجه مهره‌های محاصره شده را برگردانده و به رنگ مهره خود درآورید. این به معنی محاصره و تصاحب است. البته مهره هایی که در جریان بازی در بین مهره های شما قرار میگیرن به رنگ مهره های شما تبدیل نمیشن و فقط مهره هایی که در راستا های 8گانه مهره هایی که شما

گذاشتین قرار میگیرن به رنگ مهره های شما در میان. هدف داشتن بیشترین مهره رنگ خود روی صفحه در پایان بازی است.

پایان بازی:

وقتی تمام صفحه پر شود و یا هیچ کدام از دو طرف حرکتی نداشته باشند، بازی به پایان می‌رسد و بازیکنی که تعداد مهره‌های بیشتری روی صفحه بازی داشته باشد برنده می‌باشد.

قوانین:

- مهره را جایی قرار دهید که یک یا چند مهره حریف را محاصره کند. یعنی مهره قرار داده شده تعدادی از مهره های حریف را بین مهره جدید و مهره هم رنگ شما در صفحه محاصره کند سپس مهره های محاصره شده را چرخانده و به رنگ مهره خود درآورید.این به معنی محاصره و تصاحب است.
- قرار دادن مهره در صفحه به صورت نوبتی انجام می گیرد.
- مهره جدید را فقط در محلی می توان قرار داد که مهره ای از حریف محاصره شود. به عبارتی مهره ای به صورت آزاد در صفحه نمی تواند وجود داشته باشد و همه مهره ها حتما در مجاورت مهره های دیگر هستند.
- خط محاصره می تواند افقی یا عمودی یا مورب یا هر سه باشد
- فقط مهره هایی را می توان تصاحب کرد که بین مهره جدید و مهره قبلی موجود در صفحه محاصره شده باشند
- در صورتی یکی از بازیکنان در نوبت خود، مکانی برای محاصره حریف نداشته باشد و نتواند حتی یک مهره او را محاصره کند، در این صورت حرکت به حریف واگذار می شود. تا زمانی که امکان محاصره برایش ایجاد شود.
- در پایان بازیکنی که تعداد مهره های بیشتری روی صفحه بازی داشته باشد، برنده است.

تابع های استفاده شده در کد:

:move_Make

عملکرد این تابع به این شکل است که صفحه بازی و حرکتی که قرار است انجام شود را از برنامه دریافت میکند و بعد از انجام حرکت و تغییر مهره های دیگر صفحه بازی جدید را به برنامه برمیگرداند

:score_Check

کار این تابع امتیاز دهی به زمین بازی با توجه به شمردن تعداد مهره های هردو بازیکن میباشد.

علاوه بر آن که اختلاف تعداد مهره های زمین را حساب میکند به ازای مهره هایی که قابل حرکت نیستند نیز امتیاز اضافه تری به آن بازیکن اختصاص میدهد که این مساله موجب میشود الگوریتم موجود در آن برای انتخاب های بعدی بهتر عمل کند و تصمیم بهتری بگیرد. این وزن ها از روی استراتژی های معروف برای بازی اتللو مقداردهی شده است

:play_can_position_Get

این تابع در هر نوبت زمین بازی را دریافت کرده و حرکت های مجاز موجود برای بازیکن را به آن برمیگرداند. همانطور که در بخش قوانین بازی گفته شد حرکات مجاز صرفا حرکاتی هستند که موجب گرفتن امتیاز و برگرداندن مهره حریف شوند

:neighbors_enemy_Get

این تابع در واقع یک تابع کمکی برای تابع قبلی به جساب می آید. ورودی این تابع زمین بازی و مختصات مخصوصی از آن است و خروجی آن نیز مختصات مهره های دشمنی است که در همسایگی آن قرار دارند.

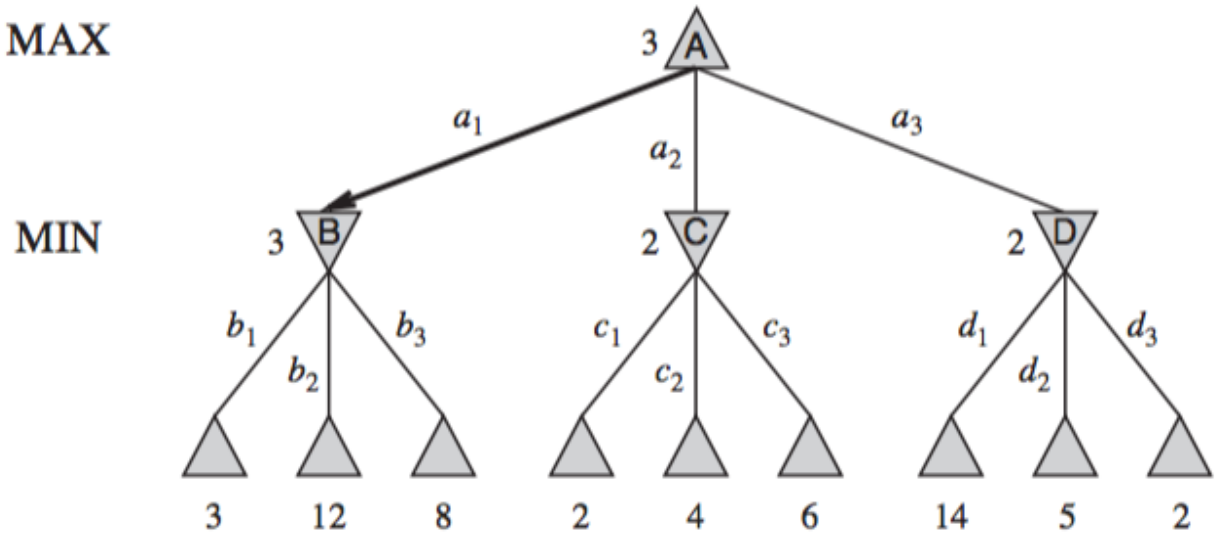
:Minimax

تابع اصلی برای تصمیم گیری کامپیوتر برای حرکت بعدی این تابع میباشد. این تابع علاوه بر خود دو تابع کمترین مقدار و بیشترین مقدار را نیز دارد که هر بار به ترتیب پشت هم صدا زده میشوند تا زمانی که به عمق مطلوب برسند و سپس با استفاده از تابع چک کردن امتیاز زمین که بالاتر توضیحات آن داده شد مقدار امتیاز زمین را برمیگردانند. این تابع ها به صورت بازگشتی عمل میکنند تا به عمق مطلوب برسند و از این طریق بهترین حرکت موجود در این حالت را به کاربر برمیگردانند

پیاده سازی این تابع از روی کتاب راسل انجام شده است و به این شکل عمل میکند که هر بار از لایه زیرین خود میخواهد که ماکسیمم مقداری که از مینیمم لایه های زیر به دست آمده را برگرداند. و با توجه به عمقی که درخت

دارد این عملیات را چند بار پشت هم تکرار میکند تا به عمق مطلوب برسد. وقتی به عمق مطلوب رسید تابع مینیمم یا ماکسیمم مورد نظر مقداری برمیگرداند و به همین ترتیب لایه های بالایی پر میشوند و مقداری را به لایه بالاتر خود بر میگردانند تا به لایه اولیه برسد و این لایه با توجه به مقدار های ممکن بهترین انتخاب را برای آن نوبت از بازی پیدا میکند.

در ادامه کمی بیشتر راجع به کلیت الگوریتم تصمیم گیری که در این جا استفاده شد توضیح میدهیم. و با چند مثال آن را نشان خواهیم داد.



درخت بالا یک درخت ساده است که با الگوریتم استفاده شده عمل میکند. در این درخت ابتدا لایه دوم درخت به لایه زیرین خود نگاه میکند ومینیمم آن را پیدا میکند. اگر شاخه سمت چپ را در نظر بگیریم مینیمم موجود در آن برابر ۳ میباشد. بنابرین مقدار ۳ در نود سمت چپ لایه دوم قرار میگیرد. نود های بعدی هم همینکار را انجام میدهند و هرکدام مقدار مینیمم لایه زیرین خود را انتخاب میکنند. وقتی که این لایه کامل شد لایه بالایی که ماکسیمم را میخواهد بین نود های لایه زیرین ماکسیمم را انتخاب میکند و بنابرین مقدار سمت چپ که برابر ۳ است را انتخاب کرده و مسیری که این نود دارد را انتخاب میکند

شبه کد این الگوریتم را میتوانید در عکس زیر که از کتاب راسل برداشته شده است مشاهده نمایید

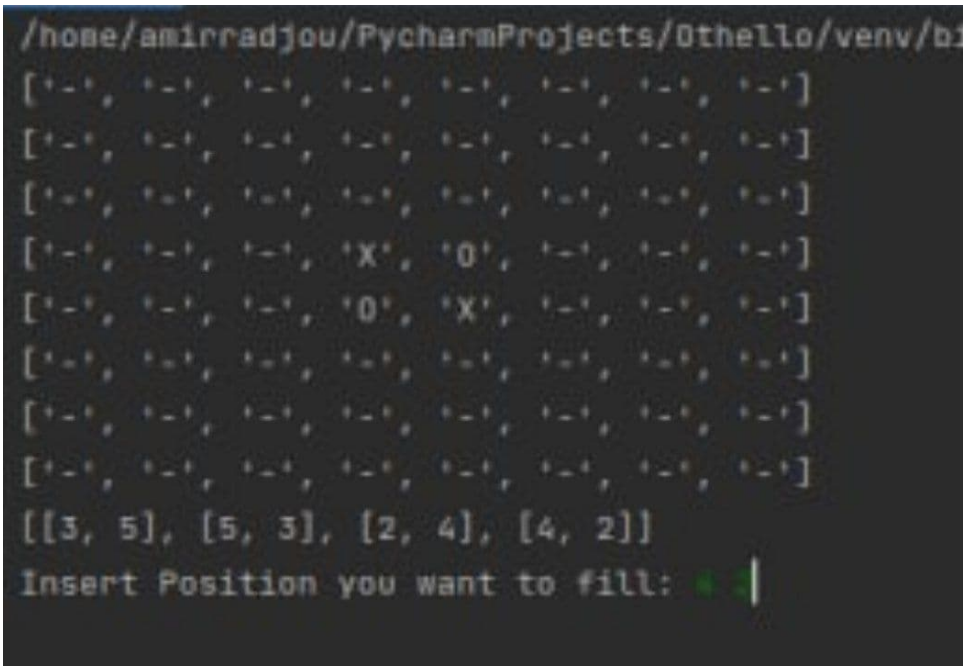
```
function MINIMAX-DECISION(state) returns an action
  return arg max_{a ∈ ACTIONS(s)} MIN-VALUE(RESULT(state, a))
```

```
function MAX-VALUE(state) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← −∞
  for each a in ACTIONS(state) do
    v ← MAX(v, MIN-VALUE(RESULT(s, a)))
  return v
```

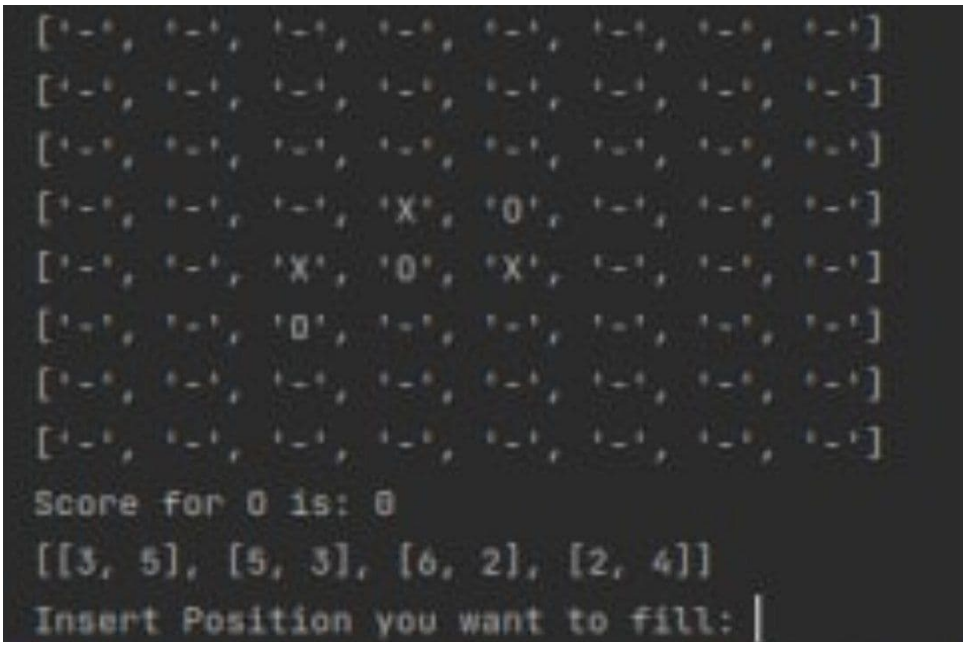
```
function MIN-VALUE(state) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← ∞
  for each a in ACTIONS(state) do
    v ← MIN(v, MAX-VALUE(RESULT(s, a)))
  return v
```

روند بازی:

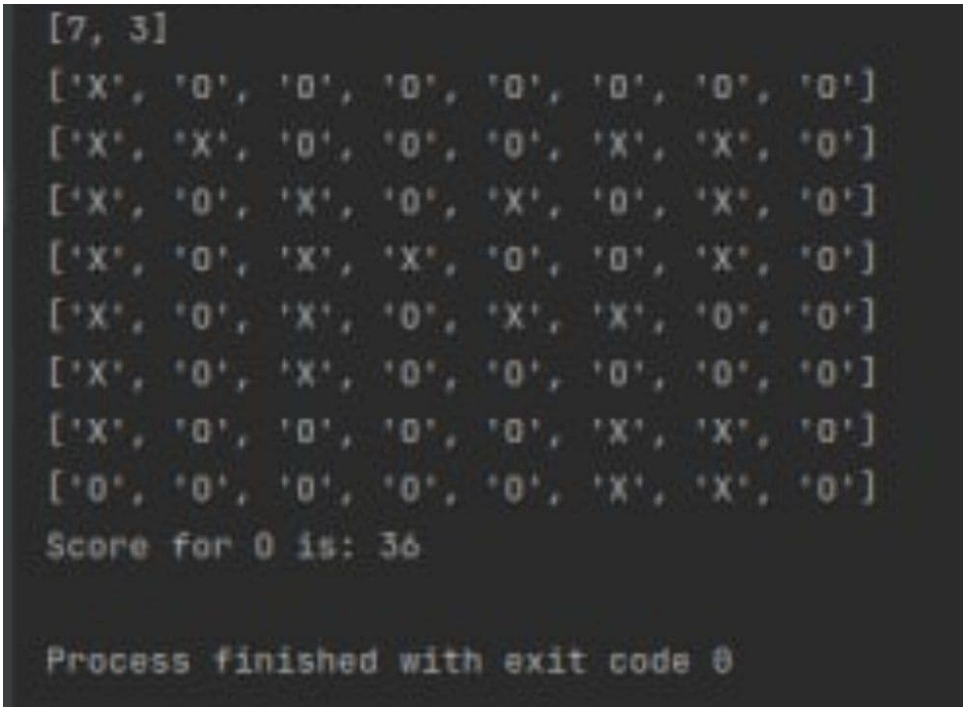
حال به بررسی کد بازی و روند اجرای آن میرویم. در عکس زیر حالت شروع بازی را مشاهده میکنید که حالت شروع بازی اتللو میباشد. در اینجا برنامه حرکات قابل اجرا را در زیر برای شما نمایش داده است و شما میتوانید از بین حرکات موجود یکی را انتخاب کرده و نوبت خود را بازی کنید.



بلافاصله بعد از اینکه نوبت خود را بازی کنید زمین بازی تغییر کرده و به شما نمایش داده خواهد شد. و سپس کامپیوتر توسط الگوریتمی که توضیح داده شد بهترین عمل تا عمقی که به آن داده شده را انجام میدهد.



در عکس پایین زمین بازی بعد از انجام حرکت نهایی توسط حریف و اتمام بازی را مشاهده میکنیم.



کد بازی را میتوانید از لینک زیر مشاهده نمایید:

این کد در حال ارتقا یباشد و سعی دارم تا رابط گرافیکی را نیز به آن اضافه کنم.

https://github.com/amirradjou/Othello

Reference:

جزوه جستجوی تخاصمی دکتر موسوی:

https://skatgame.net/mburo/ps/evalfunc.pdf

کتاب هوش مصنوعی راسل:

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.49.7258&rep=rep1&type=pdf