به نام خدا

دانشکده مهندسی برق و کامپیوتر دانشگاه تهران

هوش مصنوعی، ترم پاییز 98-99

پروژه بازی ، مهلت ارسال: تا یکشنبه ۱۲ آبان

طراحان پروژه: محمدرضا یزدانی‌فر، صدف صادقیان، نازنین صبری

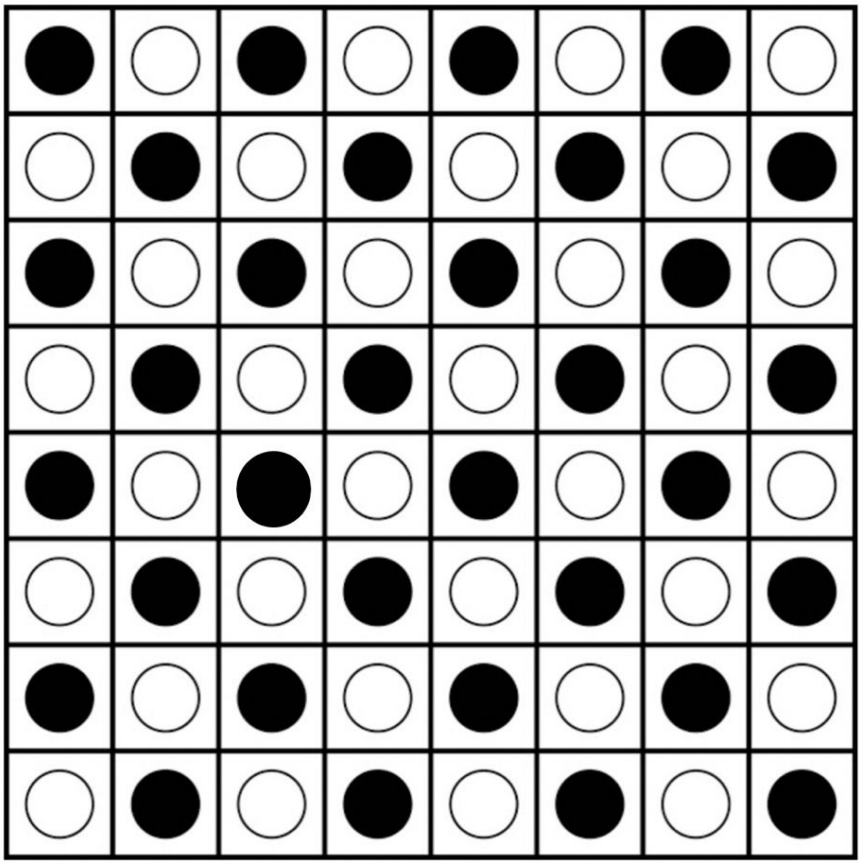
پروژه بازی

مقدمه

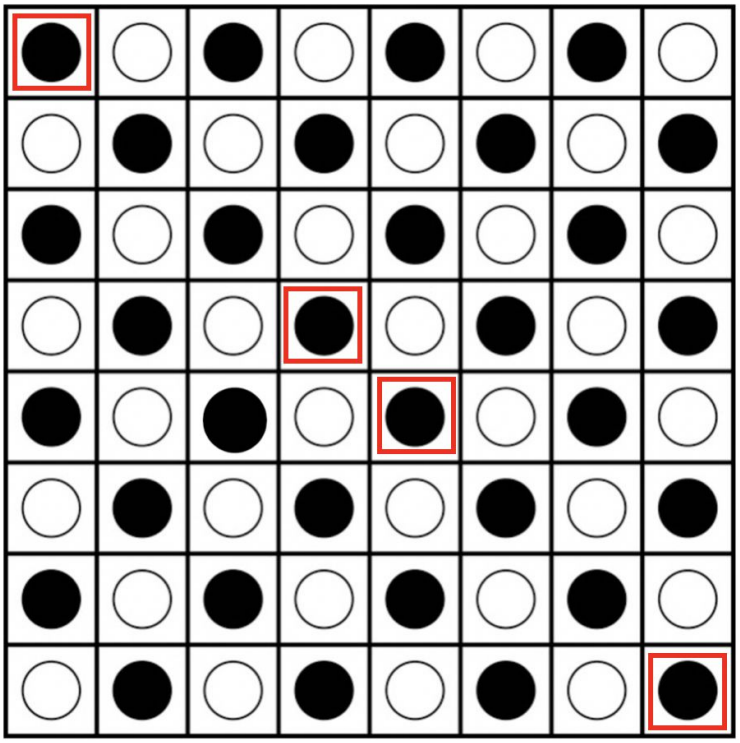
هدف این پروژه آشنایی بیشتر شما با بازی‌ها است. در ادامه به توضیح بازی که در این پروژه برای شما در نظر گرفته شده و قوانین این بازی می‌پردازیم.

توضیحات دقیق تر بازی

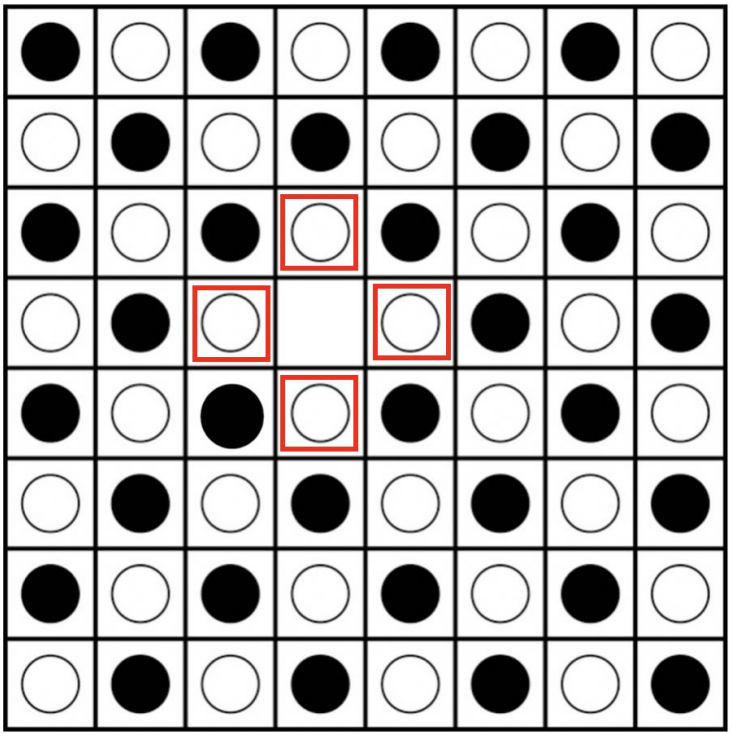
این بازی در یک برد ۸ در ۸ با مهره‌های سیاه و سفید بازی می‌شود. مراحل بازی در ادامه توضیح داده شده‌اند.



- بازیکن رنگ سیاه بازی را با حذف کردن مهره‌ای سیاه از یکی از چهار مربع مرکزی یا یکی از چهار مربع گوشه تخته شروع می‌کند. این معادل موقعیت‌های (۰ ، ۰) ، (۳ ، ۳) ، (۴ ، ۴) یا (۷ ، ۷) می‌باشد.

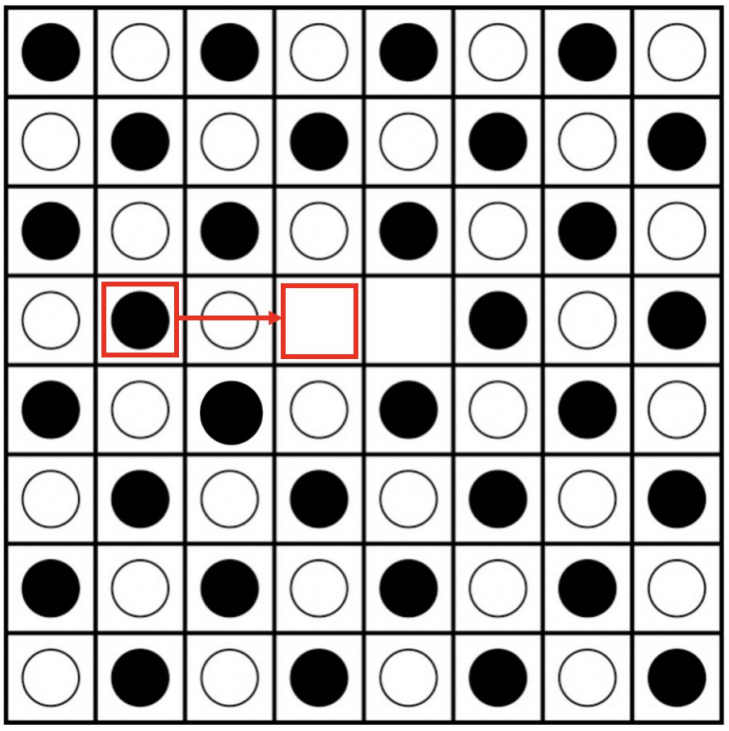


- بازیکن رنگ سفید یک مهره را از مجاورت فضای ایجاد شده توسط حرکت قبلی بازیکن سیاه، حذف می‌کند. برای مثال اگر فرض کنیم بازیکن سیاه مهره‌ای از موقعیت (۳ ، ۳) را حذف کرده‌است حال بازیکن سفید باید یک مهره از بین جایگاه‌های (۳ ، ۲) ، (۳ ، ۴) ، (۴ ، ۳) ، (۲ ، ۳) را حذف کند.

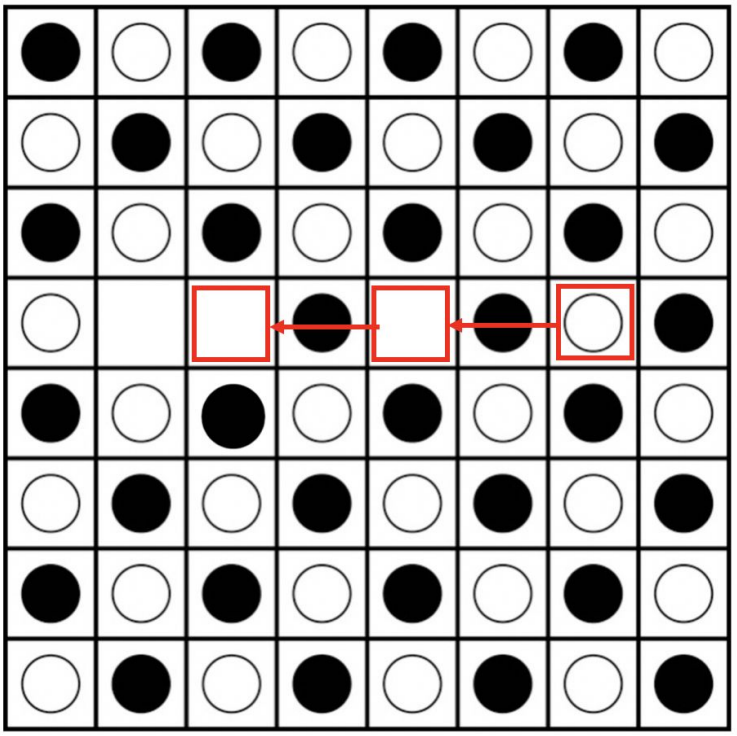


- پس از انجام این مرحله فاز زدن مهره‌ها شروع می‌شود و بازیکن رنگ سیاه دارای اولین نوبت است.

- یک بازیکن در نوبت خود مهره‌ای از رقیب را می‌زند به این شکل که یکی از مهره‌های خودش را برمی‌دارد، بصورت عمودی یا افقی از روی یکی از مهره‌های رقیب مجاورش می‌پرد، در جایگاه خالی در طرف دیگر قرار می‌گیرد و آن مهره‌ی رقیب را که از رویش پریده‌است، می‌زند.



- در صورت امکان و تمایل بازیکن، او می‌تواند از همان مهره یکسان استفاده کند تا از روی مهره‌های بیشتری از رقیب و در همان جهت بپرد.



- زمانی که بازیکنی در یک نوبتش تعدادی پرش انجام می‌دهد، جهت پرش مهره ای که منتقل می‌شود نباید عوض شود. در واقع همان مهره باید تعدادی پرش در فقط یک جهت ثابت انجام دهد. (مهره در یک خط منتقل شود.)
- اگر بازیکن فکر کند به ضرر اوست، لازم نیست تمامی پرش های ممکن (و در نتیجه زدن‌های مهره‌های حریف) را انجام دهد.
- زمانی که یک بازیکن حرکت یا حرکت‌هایش را انجام داد نوبت به بازیکن دیگر می‌رسد، این کار تا زمانی ادامه پیدا می‌کند که بازی به پایان برسد.
- بازی زمانی به پایان می‌رسد که یک بازیکن دیگر نتواند حرکتی انجام دهد و این بازیکن بازنده محسوب می‌شود و رقیب او برنده بازی است.
- تعداد مهره‌های زده شده هیچ تاثیری در برد و باخت بازیکنان ندارد.

گام پروژه

یک کلاس به نام MinimaxPlayer را پیاده سازی کنید که از کلاس Player و کلاس Game ارث‌بری کند. کلاس شما باید متد initialize و getMove را از کلاس Player پیاده‌سازی کند.

ابتدا agentخود را به شکلی بنویسید که با استفاده از درخت Minimax حرکت بعدی خود را مشخص کند و از یک evauation function خوب استفاده کنید. در ادامه برای بهبود agent خود از alpha beta prunning برای هرس کردن درختتان استفاده کنید تا تصمیم‌های بهتری بگیرد.

کدهای آماده

کدهای آماده را می‌توانید از اینجا دریافت کنید. در ادامه توضیحات لازم در مورد متدهای کلاس Game آمده‌است.

- متد **reset**: شرایط برد را به شرایط شروع باز می‌گرداند.
- متد **contains**: اگر سطر و ستون داده شده نشانگر یک مکان معتبر در صفحه باشد true را برمی‌گرداند.
- متد **opponent**: رنگ حریف را برمی‌گرداند.
- متد **distance**: فاصله بین دو نقطه را در یک خط عمودی یا افقی روی صفحه برمی گرداند. پرش های مورب مجاز نیست.
- متد **makeMove**: صفحه فعلی را با صفحه بعدی ایجاد شده توسط این حرکت به روز کنید.
- متد **nextBoard**: با توجه به حرکت یک بازیکن از (r1 ,c1) به (r2, c2) ، این حرکت را روی نسخه‌ای از صفحه فعلی اجرا می کند. در صورت عدم اعتبار حرکت ، یک GameError رخ می‌دهد. این متد یک کپی از صفحه را برمی گرداند ، و صفحه مورد نظر را تغییر نمی دهد.
- متد **openingMove**: در صورتی که حداکثر یکی از بازیکنان حرکت کرده باشد true برمی‌گرداند.
- متد **generateFirstMoves**: حرکات اختصاصی را برای حرکت اول بازی را برمی‌گرداند.
- متد **generateSecondMoves**: حرکات خاص را برای حرکت دوم بازی بر اساس جایی که اولین حرکت در آن اتفاق افتاده است برمی‌گرداند.
- متد **check**: بررسی می کند که آیا پرش با شروع از (r,c) و رفتن به جهتی که توسط دلتای ردیف (rd) و دلتای ستون (cd) انجام می شود ، امکان پذیر است.از factor برای بررسی بازگشتی جهش های متعدد در همان جهت استفاده می شود و همه‌ی جهش‌های ممکن را در جهت مشخص برمی گرداند.
- متد **generateMoves**: با استفاده از پیکربندی صفحه فعلی، تمام حرکت‌های قانونی را برای بازیکن داده شده ایجاد و برمی گرداند.

- متد **playOneGame**: با توجه به دو نمونه‌ی بازیکنان ، یک بازی بین آنها انجام می‌شود. در صورت پیروزی سیاه ، "B" را برمی گرداند، یا اگر سفید برنده شود "W" را برمی‌گرداند. وقتی show، درست است، هر حرکت در بازی را نشان داده می‌شود.

گزارش کار

در گزارش خود ابتدا به صورت کوتاه تمامی کاری که در پروژه انجام داده‌اید (از جمله الگوریتم‌های پیاده سازی شده) را بیان کنید. گزارش در عین خلاصه بودن باید جامع باشد. همچنین در گزارش خود به توضیح موارد زیر نیز بپردازید:

۱) تابع evaluation function که انتخاب کردید.

۲) آیا حرکات الگوریتم alpha-beta search نسبت به حرکات الگوریتم جستجوی minimax متفاوت است؟

۳) زمان حرکت را در الگوریتم alpha-beta و minimax search را اندازه‌گیری کرده و با یکدیگر مقایسه نمایید.

نکات پایانی

- هدف از تمرین یادگیری شما است لطفا تمرین را خودتان انجام دهید
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم درس مطرح کنید تا بقیه از آن استفاده کننده، در غیر این صورت ایمیل بزنید یا حضوری از یکی از طراح‌های پروژه بپرسید.

موفق باشید!