بسمه تعالی

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

درس هوش مصنوعی

مسابقه اختیاری

زمستان ۱۳۹۶

مقدمه

با توجه به آشنایی با الگوریتم های مختلف جستجو در درس هوش مصنوعی، پروژه ای اختیاری برای استفاده از آموخته های این درس و حتی مفاهیم هوش مصنوعی که در این درس مورد بررسی قرار نگرفت طراحی شده است که در قالب پیاده سازی یک عامل هوشمند برای بازی اتلو ارایه شده است. برای این پروژه از یک چهارچوب متن باز استفاده شده که کل بازی در آن پیاده سازی شده است و در هر مرحله از بازی لیست حرکت های مجاز قابل انجام را باز می گرداند و کافی است برای طراحی عامل هوشمند خود، از بین حرکت های مجاز، هوشمندانه ترین حرکت را انتخاب کنید و درگیر مسائل پیاده سازی این بازی نخواهید شد.

چهارچوب ارایه شده قابلیت های دیگری از جمله رابط کاربری مناسب و امکان بازی در مقابل عاملی که طراحی کرده اید (انسان در مقابل کامپیوتر) در اختیار شما قرار می دهد که برای عیب یابی و ارزیابی عامل هوشمند شما بسیار کارآمد خواهد بود. در نهایت، بین گروه های شرکت کننده در این پروژه یک مسابقه برگزار خواهد شد که توضیحات بیشتر در مورد نحوه برگزاری و امتیاز دهی و تاثیر آن در نمره درس در ادامه بیان شده است.

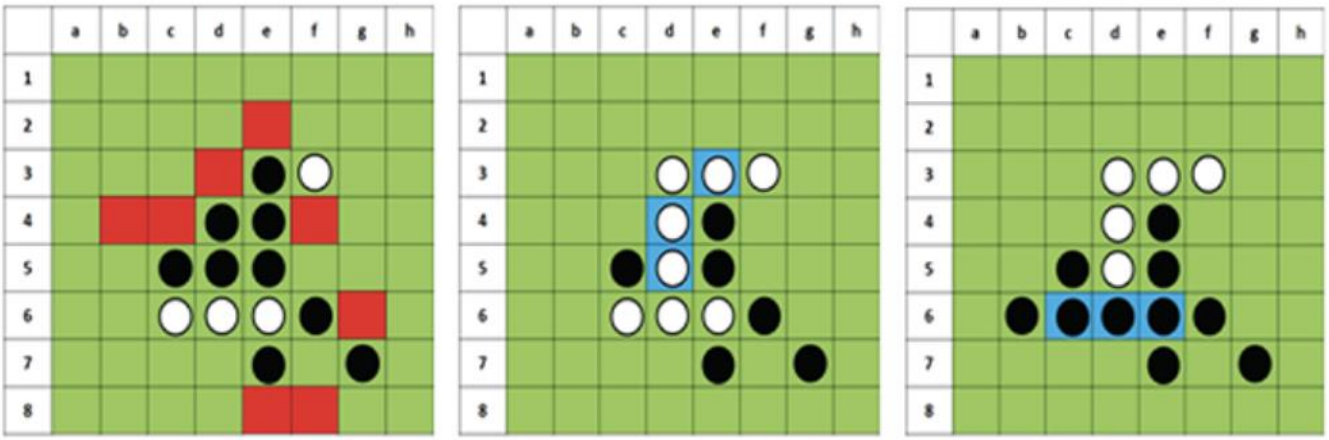
شرح بازی

همان طور که پیش تر توضیح داده شد، بازی در نظر گرفته شده برای این پروژه اتلو می باشد که به صورت دو نفره در یک صفحه مربعی ۸×۸ انجام می شود.

این بازی نسخه تجاری شده بازی Revesi می باشد و قوانین بازی به همراه نکات مفید در رابطه با استراتژی های بازی در این لینک وجود دارد.
در نسخه اتلو، در ابتدا چهار مهره در مرکز صفحه قرار گرفته‌اند که دو مهره از آن متعلق به هر یک از بازیکن‌ها می باشد که به صورت قطری قرار گرفته اند در حالی که در Revesi در ابتدای بازی صفحه خالی می باشد.

به طور خلاصه هر بازیکن باید یک مهره را بر روی تخته در جایگاهی قرار دهد که تعدادی از مهره های حریف (حداقل یک مهره) به صورت افقی، عمودی یا مورب بین مهره جدید و مهره های قبلی بازیکن مورد نظر قرار گیرد. پس از آن تمام مهره های میانی، هم رنگ مهره ی ابتدایی و انتهایی می شوند.

برای مثال در شکل زیر، سمت چپ تمام حرکت های مجاز برای بازیکن با مهره سفید با رنگ قرمز نشان داده شده اند. پس از انتخاب خانه 3d، خانه هایی که در شکل وسط به رنگ آبی مشخص شده اند از رنگ سیاه به رنگ سفید تبدیل می شوند و بعد از آن بازیکن با مهره سیاه خانه 6b را انتخاب کرده و خانه هایی که به رنگ مشکی تغییر کرده اند در شکل سمت راست نشان داده شده اند:

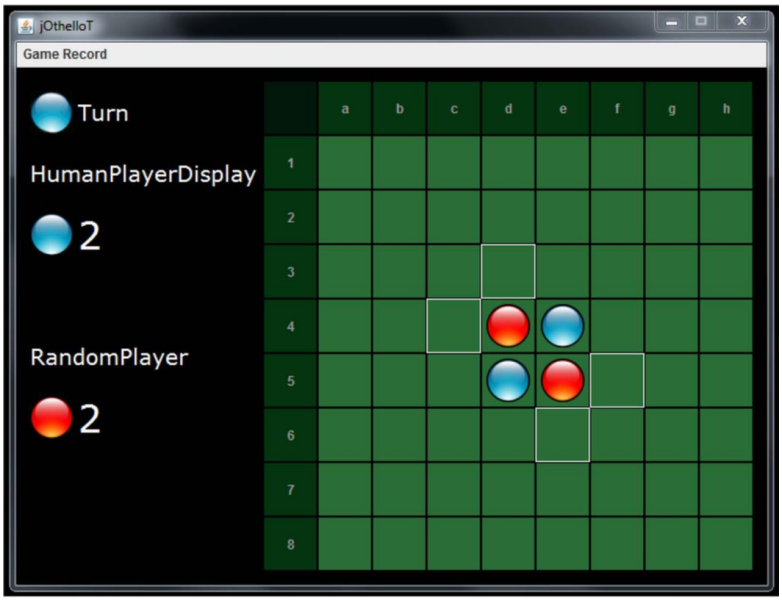


اگر بازیکنی نتواند حرکتی انجام دهد، بازی با حرکتی توسط حریف ادامه پیدا می کند و در صورتی که هیچ یک از بازیکنان نتوانند حرکتی انجام دهند. (حرکت مجازی ممکن نباشد و یا تمام صفحه اشغال شود)، بازی به اتمام می رسد. در چنین شرایطی برنده کسی است که دیسک های بیشتری در صفحه دارد.

آماده سازی اولیه

برای استفاده از این چهارچوب، کافی است که آن را در IDE مورد نظر خود Import کنید. برای اطلاع از این که آیا این کار به درستی انجام شده است، متد main داخل کلاس Game را اجرا کنید. در صورت موفقیت قادر خواهید بود در مقابل عامل رندم بازی کنید و علاوه بر اطمینان از درستی Import کردن کد، با بازی هم آشنا شوید.

تصویر زیر، وقتی که متد main کلاس Gameبه درستی اجرا شود، نشان داده خواهد شد.



نکات پیاده سازی

این پروژه باید تماما به زبان جاوا پیاده سازی شود. هر گروه، یک کلاس که باید از کلاس AbstractPlayer Extend شده باشد ایجاد می کند و constructor کلاس AbstractPlayer و متد play (int [] [] tab(را پیاده سازی می کند. Constructor برای مشخص کردن عمق جستجوی الگوریتم شما می باشد و Signature تابع play ،

pub l icBoardSquare play int) [] [] (tab

است. این متد در نوبت بازی بازیکن، از طریق شیء Game، صفحه بازی را در اختیار بازیکن قرار می دهد و شما می بایست منطق عامل هوشمند خود را در آن پیاده کنید.

یکی دیگر از متد های شیء Game، متد

getValidMoves ,tab [] [] int) getMyBoardMark(( )

است که لیست تمام حرکت های مجاز را برای بازیکن فعلی برمیگرداند. بنابراین شما تنها کافی است با توجه به وضعیت صفحه و حرکت های مجاز هوشمندانه ترین حرکت را انتخاب کنید.

برای اینکه بتوانید به طور دستی با عاملی که نوشته اید رقابت کنید یا آن را در رقابت با عوامل دیگر قرار دهید، کافی است کلاس بازیکن هایی که می خواهید در بازی شرکت کنند را به عنوان پارامتر در متدmain کلاس Game مشخص کنید. اگر هیچ پارامتری تنظیم نشده باشد به طور پیش فرض عامل رندم در مقابل عامل انسانی به رقابت می پردازد.

فایل های پروژه

کدهای این چهارچوب در سایت درس آپلود شده است. همچنین می توانید آن‌ها را از اینجا دانلود کنید.

شما در نهایت باید یک پکیج که شامل تنها یک کلاس با نام گروه شما است در سایت درس بارگذاری کنید.

نمره دهی

در حالت مسابقه، به ازای پیروزی در هر بازی ۳ امتیاز و تساوی ۱ امتیاز و باخت صفر امتیاز به عامل شما تعلق خواهد گرفت. تیم هایی که موفق به شکست بازیکن تصادفی و بازیکن حریصانه (بازیکنی که در هر مرحله حرکت با بیشترین سود را انتخاب می کند) شوند در مرحله نهایی مسابقه شرکت داده خواهند شد.

در مرحله نهایی، بین عامل های راه یافته به این مرحله یک مسابقه به شیوه تک گروهی برگزار خواهد شد که در آن هر دو عامل دو مرتبه با یکدیگر رو به رو خواهند شد با این تفاوت که شروع کننده بازی متفاوت خواهد بود. تیم‌های راه یافته به مرحله نهایی بر اساس امتیاز مرتب شده و به بهترین تیم 1نمره تعلق خواهد گرفت و رتبه های بعدی با کسر ۰,۱ نمره به ازاء هر رتبه امتیازدهی خواهند شد.

استفاده از الگوریتم‌های هوشمندی که در کلاس درس تدریس شده و یا استفاده از ایده‌های دیگر مرتبط با تکنیک‌های هوش مصنوعی امتیاز جداگانه‌ای خواهد شد که بر اساس گزارش ارسالی و بررسی کد تیم‌های مورد نظر در مورد آن تصمیم‌گیری خواهد شد.

خلاصه!

- باید برنامه‌ی خود را به زبان جاوا پیاده سازی کنید.
- تیم های شرکت کننده در مسابقه می‌توانند متشکل از یک یا دو نفر باشند. درصورت تشابه کد تیم‌ها با یکدیگر یا با منبع سومی، به تیم‌ها نمره ای تعلق نخواهد گرفت.
- درصورت استفاده از الگوریتم های هوشمند، گزارش مربوطه را نیز ضمیمه کنید تا کد شما در این خصوص نیز ارزیابی شود.
- تیم هایی که موفق به شکست بازیکن تصادفی و بازیکن حریصانه (بازیکنی که در هر مرحله حرکت با بیشترین سود را انتخاب می کند) شوند در مرحله نهایی مسابقه شرکت داده خواهند شد.
- تاریخ برگزاری مسابقه نهایی متعاقبا اعلام خواهد شد.
- کدها و مستندات خود را در یک فایل فشرده شده در سایت درس بارگذاری نمایید.

موفق باشید.