

## بازی دوز Tic-Tac-Toe

با استفاده از MiniMax، یک هوش مصنوعی را پیاده سازی کنید که بازی tic-tac-toe را بصورت بهینه بازی کند.



## شروع

کد توزیع را از <https://cdn.cs50.net/ai/2020/x/projects/0/tictactoe.zip> دانلود و آن را از حالت فشرده خارج کنید. پس از ورود به دایرکتوری پروژه، دستور `pip3 install -r requirements.txt` را در ترمینال اجرا کنید تا مازول پایتون مورد نیاز برای این پروژه (pygame) نصب شود.

## درک مساله

دو فایل اصلی در این پروژه وجود دارد runner.py و tictactoe.py. tictactoe.py حاوی منطق انجام بازی برای انجام حرکتهای بهینه است. runner.py نیز حاوی کدهای اجرای رابط گرافیکی بازی است. وقتی تمام توابع مورد نیاز در tictactoe.py را کامل کردید، می‌توانید python runner.py را اجرا و در برابر هوش مصنوعی بازی کنید. اکنون tictactoe.py را باز کنید. ابتدا سه متغیر O، X و EMPTY را جهت نمایش حرکتهای ممکن صفحه (board، صفحه بازی) تعریف می‌کنیم.

تابع initial\_state وضعیت اولیه صفحه را باز می‌گرداند. برای این مساله، صفحه را به عنوان مجموعه‌ای از سه لیست (که نشان‌دهنده سه ردیف صفحه هستند). تعریف می‌کنیم در این مجموعه مقدار هر یک از اعضای لیست می‌تواند حاوی سه مقدار O، X یا EMPTY باشد. در ادامه نیز توابعی وجود دارند که شما باید کامل کنید.

## مشخصات

ابزار خودکاری به کارکنان در چک کردن محدودیتهای زیر کمک می‌کند. اگر هر یک از این موارد به‌درستی انجام نشود، ماژول‌هایی به غیر از موارد مجاز را وارد کنید یا توابعی را به‌صورت غیرمجاز تغییر دهید، فایل تحویل داده شده، نمره قبولی دریافت نمی‌کند.

توابع player، action، result، winner، terminal، utility و minimax را کامل کنید.

تابع player باید وضعیت صفحه را به عنوان ورودی دریافت کند و نوبت بازیکن را بازگرداند. (X یا O)

در وضعیت آغازین بازی، X حرکت نخست را انجام می‌دهد. سپس با هر حرکتی که انجام می‌شود، بازیکن هم تغییر می‌کند. در صورتی که صفحه پایانی به عنوان ورودی ارائه شده باشد (یعنی بازی پایانه یافته باشد)، تمام مقادیر بازگشتی قابل قبول است.

تابع actions باید مجموعه‌ای (set) از تمام حرکت (action) هایی را که می‌توان روی صفحه‌ی با وضعیت مشخصی انجام داد، بازگرداند.

هر یک از حرکات باید به عنوان زوج (i,j) ارائه شود که در آن i متناظر با ردیف حرکت (۰ یا ۱ یا ۲) و j نشان دهنده سلول در ردیف متناظر با حرکت مورد نظر است. (این متغیر هم برابر ۰ یا ۱ یا ۲ است)

حرکت‌های ممکن شامل تمام سلول‌های روی صفحه هستند که مقدار X یا O ندارند.

در صورتی که یک صفحه پایانی (terminal board) به عنوان ورودی ارائه شود، تمام مقادیر بازگشتی قابل قبول هستند. تابع result مقدار board و action را به عنوان ورودی می‌گیرد و باید وضعیت صفحه جدید را بدون تغییر صفحه اصلی بازگرداند.

اگر action شامل اقدام قابل قبولی برای صفحه نباشد، برنامه‌تان باید [exception](#) برگرداند.

وضعیت بازگردانده شده باید صفحه‌ای باشد که از صفحه ورودی اصلی حاصل می‌شود و به بازیکنی که نوبت اوست، اجازه انجام حرکت در خانه ای (input برنامه) که انتخاب کرده، بدهد.

مهم است که صفحه اصلی بدون تغییر بماند زیرا مینی‌مکس باید چندین وضعیت صفحه مختلف را در حین محاسبه لحاظ کند؛ پس به روز کردن تنها یک خانه از board، نشان‌دهنده اجرای دقیقی از تابع result نیست. احتمالا باید پیش از انجام تغییر، [کپی عمیقی](#) از صفحه تهیه کنید.

تابع winner باید board را به عنوان ورودی بگیرد و برنده صفحه را (در صورت وجود) بازگرداند.

اگر بازیکن X برنده شده باشد، تابع شما باید X را بازگرداند. اگر بازیکن O برنده شده باشد، تابع شما باید O را بازگرداند. می‌توان بازی را با سه حرکت متوالی در جهت‌های افقی، عمودی و قطری برنده شد.

می‌توانید فرض کنید حداکثر یک برنده وجود دارد (یعنی در هیچ صفحه‌ای هر دو بازیکن نمی‌توانند با سه حرکت متوالی برنده شوند زیرا این وضعیت صفحه قابل پذیرش نیست).

اگر بازی برنده نداشته باشد (مثلا چون بازی هنوز در جریان است یا مساوی تمام شده)، این تابع باید مقدار None را بازگرداند. تابع terminal باید board را به عنوان ورودی بپذیرد و یک مقدار boolean را که نشان می‌دهد بازی تمام شده است یا خیر، بازگرداند.

اگر بازی تمام شده باشد (چون کسی بازی را برنده شده یا تمام سلول‌ها پر شده‌اند اما کسی برنده نشده)، این تابع باید مقدار True را بازگرداند.

در غیر این صورت اگر بازی تمام نشده باشد، تابع باید مقدار False را بازگرداند.

تابع utility باید board پایانی (یکی از وضعیت هایی که بازی پس از آن ادامه پیدا نمی کند) را به عنوان ورودی بپذیرد و مطلوب بودن صفحه را بازگرداند.

اگر X بازی را برده باشد، مطلوبیت برابر ۱ و اگر O برنده شده باشد، مطلوبیت برابر -۱ است. اگر بازی مساوی شده باشد، مطلوبیت صفر است.

می توانید فرض کنید utility صرفاً زمانی روی صفحه فراخوانده می شود که terminal(board) برابر True باشد.

تابع minimax باید board را به عنوان ورودی بپذیرد و حرکت بهینه بازیکن را بازگرداند.

حرکت بازگردانده شده باید شامل اقدام بهینه (i,j) ای باشد که یکی از اقدامات مجاز روی صفحه است. اگر چند حرکت به یک اندازه بهینه باشند، تمام آنها قابل قبول است.

اگر board از نوع terminal board باشد، تابع minimax باید مقدار None را بازگرداند.

برای تمام توابعی که board را به عنوان ورودی می پذیرند، فرض کنید که این صفحه قابل قبول است (یعنی لیستی شامل سه ردیف که هر کدام سه مقدار X، O یا EMPTY دارند). نباید تعریف های از پیش نوشته شده در توابع را تغییر دهید.

پس از این که تمام توابع به درستی اجرا شدند، می توانید python runner.py را اجرا و با هوش مصنوعی بازی کنید. از آنجا که در بازی دوز اگر هر دو طرف بهینه بازی کنند نتیجه مساوی خواهد شد، پس هرگز نباید بتوانید هوش مصنوعی را شکست دهید (البته در صورتی که بهینه بازی نکنید هم بازنده خواهید شد)

## راهنمایی ها

اگر می خواهید توابع خود را در فایل پایتون دیگری تست کنید، با دستور `from tictactoe import initial_state` می توانید آنها را انتقال دهید.

می توانید توابع کمکی دیگری نیز به `tictactoe.py` بیفزایید؛ به شرطی که نام آنها با نام توابع و متغیرهای موجود در ماژولها تداخل نداشته باشد.

هرس آلفا-بتا اختیاری است اما می تواند هوش مصنوعی را کارآمدتر کند.

## چطور ارسال کنیم؟

ممکن است کد شما در شاخه `ai50/projects/2020/x/tictactoe` نباشد و در زیر دایرکتوری دورتری مثل `tictactoe` یا `projects0b` باشد. ببینید اگر اعضای تیم ما سعی کنند به `https://github.com/me50/USERNAME/blob/ai50/projects/2020/x/tictactoe/tictactoe.py` دسترسی پیدا کنند که `username` همان نام کاربری گیت‌هاب شماست، فایل شما باید دقیقاً در آن آدرس حاضر باشد. اگر فایل شما در زمان نمره‌دهی در آن آدرس نباشد، تیم ما نمیتواند به آن نمره بدهد و فایل ثبت شده شما نمره‌ای دریافت نمی‌کند.

1. وارد [این لینک](#) شوید، و به حساب گیت‌هاب خود وارد شوید، بر روی `Authorize cs50` کلیک کنید. سپس، مواردی را که نشان می‌دهد مایلید کارکنان با مجوز دوره به تمرین‌های ارسالی شما دسترسی داشته باشند را علامت بزنید و سپس بر روی `Join Course` کلیک کنید.

2. [گیت را نصب کنید](#) و اگر تمایل داشتید `submit50` را هم نصب کنید.

3. اگر `submit50` را نصب کرده اید، خط زیر را اجرا کنید:

```
submit50 ai50/projects/2020/x/tictactoe
```

در غیر اینصورت، با استفاده از گیت، تکلیف خود را در <https://github.com/me50/USERNAME.git>، `push` کنید،

`USERNAME` همان نام کاربری اکانت گیت‌هاب شما روی شاخه‌ای به نام `ai50/projects/2020/x/tictactoe` است

4. [این فرم](#) را نیز پر کنید.

می‌توانید با رفتن به آدرس <https://cs50.me/cs50ai> پیشرفت خود را مشاهده کنید.