

استاد درس : دکتر مهدی قطعی

استاد کارگاه : بهنام یوسفی‌مهر

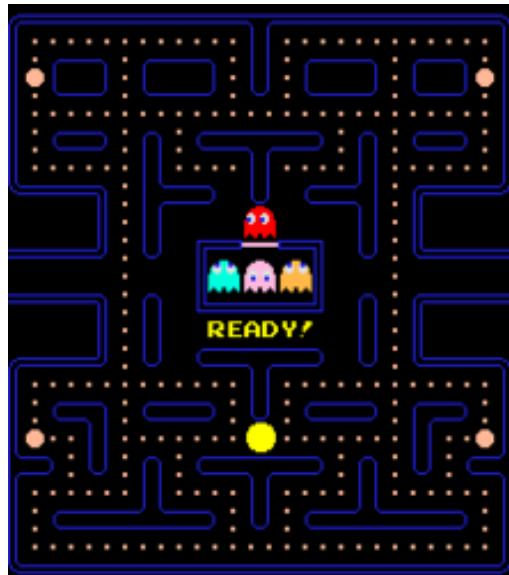
بهار ۱۴۰۳

هوش مصنوعی و کارگاه

راهنمای تمرین سری سوم



PACMAN بازی



شکل ۱: محیط بازی PACMAN

بازی PACMAN بازی‌ای کلاسیک است که یک صفحه بازی به اندازه 26×36 همراه با مواعنی می‌باشد. هدف این بازی خوردن تمام نقاط موجود در صفحه با کمترین تعداد حرکت توسط PACMAN می‌باشد. حرکت در چهار جهت بالا، پایین چپ و راست برای PACMAN مجاز می‌باشد اما توجه داشته باشید که نمی‌تواند از روی موائع عبور کند؛ برای مثال در صورتی که به سمت راست حرکت کند اما سمت راستی مانع وجود داشته باشد، در این صورت سر جای خود باقی خواهد ماند. همچنین چهار روح در این بازی با نام‌های Inky، Blinky، Pinky و Clyde موجود می‌باشند که PACMAN نباید به آنها برخورد کند و در صورت برخورد با روح‌ها، یکی از جان‌هایش را از دست می‌دهد. علاقه‌مندان جهت مطالعه بیشتر پیرامون این بازی جذاب می‌توانند به [این لینک](#) مراجعه نمایند.

اهداف تمرین و نحوه پیاده‌سازی

در این تمرین شما باید عامل‌های هوشمند^۱ این بازی از جمله PACMAN و روح‌ها را پیاده‌سازی کنید. در همین راستا شما باید الگوریتم‌های Minimax، هرس آلفا-بتا^۲ و Expectimax را پیاده‌سازی کنید و همچنین سعی کنید بهترینتابع ارزیابی^۳ را طراحی کنید.

در این تمرین هدف این است که کامپیوتر به عنوان یک عامل هوشمند نقش PACMAN را داشته باشد و جهت حرکات را تصمیم بگیرد، توجه داشته باشید که هدف این تمرین بازی توسط کاربر نمی‌باشد و برنامه ورودی‌ای ندارد. بازی به صورت یکی در میان بین PACMAN و روح‌ها می‌باشد به این صورت که در ابتدا PACMAN بازی را شروع کرده و در یک جهت حرکت می‌کند، سپس روح اول در یک جهت حرکت کرده و پس از آن روح دوم حرکت می‌کند و سپس دوباره PACMAN حرکت می‌کند و همین روند تا زمانی که بازی تمام شود ادامه می‌باید. در هر مرحله که نوبت PACMAN می‌باشد باید یک حرکت انجام دهد اما در صورتی که در کنارش مانع باشد می‌تواند به سمت مانع حرکت کند و در واقع در جای خود باقی بماند. برای روح‌ها نیز این مسئله صادق است و در هر مرحله باید حرکت انجام دهنده و در صورت وجود مانع می‌توانند به سمت آن حرکت کنند.

¹ Rational Agent

² Alpha-Beta Pruning

³ Evaluation Function

راهنمای تمرین سری سوم

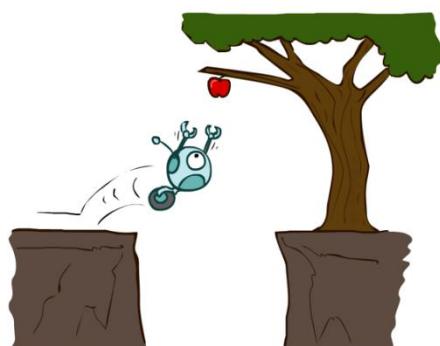
به جهت ساده‌سازی انجام پیاده‌سازی، جلوگیری از انجام وضایف حاشیه‌ای مانند پیاده‌سازی گرافیک بازی و همچنین هماهنگی بیشتر، شما عزیزان باید پیاده‌سازی‌های خود را در قالب کد پروژه PACMAN کورس آموزشی CS188 دانشگاه برکلی^۴ انجام دهید که از طریق [این لینک](#) می‌توانید به آن دسترسی داشته باشید. جهت اطلاعات بیشتر درمورد این قالب می‌توانید از طریق [این لینک](#) به سایت کورس آموزشی CS188 مراجعه نمایید.

Reflex Agent

برای اولین بخش این تمرین، شما باید کلاس ReflexAgent را تکمیل نمایید تا عامل واکنشی توانمندی تولید کنید که هم موقعیت غذاها و هم موقعیت روح‌ها را در تصمیم‌گیری‌های خود در نظر بگیرد. جهت اطمینان حاصل کردن از اینکه این عامل را به درستی پیاده‌سازی کرده‌اید، از دستور زیر استفاده نمایید. عامل شما باید بتواند به راحتی از پس زمین `testClassic` برپایید:

```
python pacman.py -p ReflexAgent -l testClassic
```

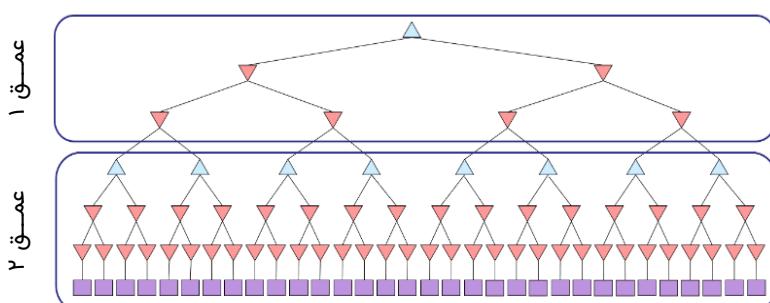
توجه داشته باشید که عامل واکنشی‌ای که در یک وضعیت^۵ است، به حرکت‌هایی^۶ که از آن وضعیت می‌تواند انجام دهد امتیاز می‌دهد. در ادامه خواهیم دید که در نقطه مقابل این رویکرد، عامل‌هایی هستند که به خود وضعیت‌ها امتیاز می‌دهند.



شکل ۲: Reflex Agent

الگوریتم Minimax

برای این قسمت شما باید در فایل `multiAgents.py` و در کلاس `MinimaxAgent` ارائه شده، یک عامل جستجوی تخصصی^۷ را پیاده‌سازی کنید. در این بازی هدف این است که PACMAN در هر مرحله از حرکت با استفاده از این الگوریتم تصمیم‌گیری کند، پس هر بار باید درختی از



شکل ۳: درخت Minimax بازی PACMAN با دو روح

وضعیت‌های ممکن بکشد از آنجایی که در این بازی دو روح داریم پس بعد از هر گره^۸ در الگوریتم، دو گره در سطح بعدی باید min باشند که به ترتیب مربوط به روح‌های اول و دوم می‌باشند. همچنین کد شما باید بتواند درخت Minimax را تا عمق دلخواه پیمایش کند. امتیاز برگ‌های درخت Minimax را با تابع `self.evaluationFunction` ارزیابی کنید. درخت `Minimax` این بازی به شرح مقابل می‌باشد:

راهنمای تمرین سری سوم

برای اطمینان حاصل کردن از درستی پیاده‌سازی تان، می‌توانید با دستور زیر از AutoGrader این قالب استفاده نمایید:

```
python autograder.py -q q2
```

در پیاده‌سازی صحیح الگوریتم جستجوی تخاصمی Minimax، کاملاً طبیعی است که PACMAN در برخی از اجراهای برنامه بازنده شود و این موضوع مشکلی ندارد.

در زمین‌های بزرگ مانند OpenClassic یا MediumClassic (زمین بازی پیش‌فرض)، مشاهده خواهید کرد که PACMAN در زنده PACMAN ماندن مهارت بالایی دارد، ولی از طرفی در بُعدن بازی ضعیف است. در بسیاری از موقعی PACMAN بدون هیچ پیشرفته در امتیازش به دور خود می‌چرخد، حتی ممکن است بدون دلیل حول یک غذا بچرخد بدون اینکه آن را بخورد زیرا که نمی‌داند بعد از خوردن آن غذا به کجا برود. در صورت مشاهده چنین رفتاری توسط PACMAN، نگران نشوید 😊

هرس آلفا-بتا

اکنون برای این قسمت شما باید در فایل AlphaBetaAgent.py و در کلاس multiAgents.py ارائه شده، یک عامل جدید طراحی کنید که با استفاده از هرس آلفا-بتا، به صورت بهینه‌تر درخت Minimax را پیمایش می‌کند.

شبه کد زیر الگوریتمی که برای پیاده‌سازی این قسمت نیاز دارید را تشریح می‌کند:

$$\alpha := \text{Max's Best Option on Path to Root}$$

$$\beta := \text{Min's Best Option on Path to Root}$$

```
def max_value(state, α, β):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor, α, β))
        if v > β return v
        α = max (α, v)
    return v
```

```
def min_value(state, α, β):
    initialize v = +∞
    for each successor of state:
        v = min(v, value(successor, α, β))
        if v < α return v
        β = min (β, v)
    return v
```

برای اطمینان حاصل کردن از درستی پیاده‌سازی تان، می‌توانید با دستور زیر از AutoGrader این قالب استفاده نمایید:

```
python autograder.py -q q3
```

راهنمای تمرین سری سوم

A- الگوریتم Expectimax

الگوریتم‌های Minimax و آلفا-بتا عملکرد خیلی خوبی دارند، ولی هر دوی آنها فرض را بر این گذاشته‌اند که در مقابل حریفی بازی می‌کنیم که تصمیمات بهینه می‌گیرد. بدیهی است که در اکثر مواقع این اتفاق نمی‌افتد. در ادامه این تمرین، شما باید در فایل `multiAgents.py` و در کلاس `ExpectimaxAgent` ارائه شده، عاملی را پیاده‌سازی کنید که بتواند انتخاب‌های نیمه بهینه^۹ حریف را مدل کند. جهت آشنایی بیشتر با این الگوریتم می‌توانید به [این لینک](#) مراجعه نمایید.

برای اطمینان حاصل کردن از درستی پیاده‌سازی تان، می‌توانید با دستور زیر از AutoGrader این قالب استفاده نمایید:

```
python autograder.py -q q4
```

معیارهای سنجش تمرین



- توضیحی کلی از بازی و نحوه‌ای که بازی را پیاده‌سازی کرده‌اید بدهید. برای مثال کلاس‌هایی که تکمیل کرده‌اید و متودهایی که برای پیاده‌سازی بازی استفاده کرده‌اید را توضیح دهید.
- هدف از پیاده‌سازی این بازی، یادگیری الگوریتم‌های Minimax، هرس آلفا-بتا و Expectimax می‌باشد، پس نحوه انتخاب عامل براساس این الگوریتم‌ها را در یک وضعیت توضیح دهید.
- تابع ارزیابی‌ای که تعریف کرده‌اید را توضیح دهید و علت تعریف تابع به این صورت را نیز شرح دهید.
- تا عمق دو و سه درخت پیش بروید و تغییرات در عملکرد PACMAN را به طور کامل تحلیل نمایید.
- با توجه به تصادفی بودن حرکت روح‌ها، بازی را چند مرتبه اجرا کرده و حرکات PACMAN و تصمیماتی که در شرایط خاص می‌گیرد را با توجه به الگوریتم تحلیل کنید.
- علاوه بر بخش عملی، پاسخ دادن به سوالات تئوری این تمرین نیز بخشی از ارزیابی شما عزیزان خواهد بود و در ارائه مجازی تمرین سوم، باید به پاسخ سوالات تئوری نیز تسلط بالایی داشته باشید. این سوالات از طریق درایو درس و گروه تلگرامی در اختیارتان قرار گرفته است.
- تمیز و خوانا بودن کد اهمیت بالایی دارد و در ارزیابی نهایی از شما عزیزان تاثیر مثبت خواهد گذاشت.
- گزارش کار باید در قالب پایان‌نامه رسمی دانشگاه که در گروه تلگرامی درس فرستاده شده و به صورت فایل PDF تحويل داده شود.
- عزیزان توجه بفرمایید که استفاده از **مدل‌های زبانی** (مانند ChatGPT) برای پاسخ دادن به تمارین **غیر مجاز** می‌باشد.

راهنمای تمرین سری سوم

مهلت تحويل



دانشجویان محترم دقت بفرمایید که مهلت تحويل تمرین سری سوم تا ساعت ۲۳:۵۹ روز ۱۵ فروردین ماه میباشد. لطفاً برای تحويل بهموقع گزارش کار خود، برنامه ریزی مناسبی داشته باشید. علاوه بر بخش عملی، این تمرین شامل هفت سوال تئوری نیز میباشد که از طریق [این لینک](#) PDF میتوانید به آنها دسترسی داشته باشید. درنهایت شما عزیزان باید فایل PDF گزارش کار، فایل پیادهسازی zip و فایل multiagent.zip پاسخنامه سوالات تئوری را در قالب یک فایل ZIP و از طریق [این لینک](#) در سامانه کوئرا بارگذاری نمایید.

در صورت هرگونه مشکل و یا ابهام میتوانید با تدریسیاران در ارتباط باشید و یا اینکه به ایمیل درس سوالات خود را ارسال نمایید:

aut.ai.spring@gmail.com

☞ موفق و سر بلند باشید ☜