

قسمت دوم : بازی Pentago

متین پس از اینکه موفق شد با هر سختی‌ای بود امتحان ریاضیات مهندسی را بگذراند، به خانه برگشت و تصمیم گرفت به بازی کردن بپردازد. او که به بازی‌های فکری و بازی‌سازی علاقه زیادی داشت تصمیم گرفت تا بازی Pentago را به کمک الگوریتم Minimax پیاده‌سازی کند. از آنجایی که او هنوز مهارت برنامه‌نویسی ندارد، دوباره از شما برای پیاده سازی این بازی کمک خواسته است.

توضیح بازی

بازی Pentago یک بازی دو نفره استراتژیک است که روی یک صفحه 6×6 انجام می‌شود. این صفحه به چهار بلوک 3×3 تقسیم شده است. بازیکنان به نوبت یک مهره (سفید یا سیاه) در خانه‌های خالی صفحه قرار می‌دهند و سپس یک بلوک 3×3 را 90° درجه (ساعتگرد یا پادساعتگرد) می‌چرخانند.

- **هدف:** ایجاد یک خط افقی، عمودی یا مورب متشکل از ۵ مهره از رنگ خودتان.
- **چرخش بلوک:** چرخش بلوک می‌تواند موقعیت مهره‌ها را تغییر دهد و ممکن است باعث ایجاد یا از بین بردن خطوط پنج‌تایی شود.
- **پایان بازی:** اگر یک بازیکن ۵ مهره متوالی بسازد، برنده است. اگر صفحه پر شود و هیچ برنده‌ای وجود نداشته باشد، بازی مساوی است.



پیاده سازی

هدف شما پیاده سازی الگوریتم Minimax است. کد بازی به شما داده شده است اما این کد کامل نیست و شما باید بخش های TODO را کامل کنید. شما باید تابع minimax را کامل کنید که در واقع پیاده سازی الگوریتم minimax برای این بازی است.

شما می توانید برای تمیزی کد خود، متد و توابع دیگری را به کد اضافه کنید اما بهتر است تغییری در بخش های دیگر کد ایجاد نکنید و این بخش ها ثابت بمانند. برای الگوریتم minimax خود، به یک تابع heuristic برای ارزشیابی هر یک از حالات نیاز دارید.

برای استفاده از کد کافیست یک نمونه از کلاس PentagonoGame با توجه به آرگومان های مد نظر بسازید و پس از پیاده سازی تابع minimax، تابع play را در نمونه ی خود صدا زده و نتیجه ی بازی به صورت زیر برگردانده می شود:

1: بازیکن برنده شده است.

0: بازی مساوی شده است.

1-: کامپیوتر برنده شده است.

همچنین برای استفاده از رابط گرافیکی به منظور پیاده سازی راحت تر، می توانید پرچم ui را هنگام ساخت نمونه True کنید. می توانید از تابع print_board نیز برای بررسی صفحه بازی در هر مرحله استفاده کنید.

بررسی نتایج

برای درک کامل الگوریتم کد را با عمق های مختلف و بدون هرس 50 الی 100 بار اجرا کنید و میانگین زمان و شانس پیروزی و میانگین تعداد نودهای دیده شده را برای هر عمق حساب کنید.

هرس آلفا و بتا: برای افزایش سرعت کد و کاهش نودهای دیده شده، هرس آلفا و بتا را به کد اضافه کنید و موارد ذکر شده در بخش قبل را مجدداً بررسی کنید.

در نهایت نتیجه بازی را برای 100 بار بازی کردن در مقابل حریف رندوم را نشان دهید.

سوالات

1. آیا میان عمق الگوریتم و پارامترهای حساب شده در بخش بالا روابطی می بینید؟ بررسی کنید که عمق

الگوریتم چه تاثیراتی بر روی شانس پیروزی، زمان و گره های دیده شده می گذارد.

2. آیا می توان ترتیب دیدن فرزندان هر نود را به گونه ای انتخاب کنیم که بیشترین هرس را داشته باشیم؟

اگر جواب شما مثبت است روش خود را توضیح دهید و در غیر این صورت توضیح دهید که چرا این

عمل امکان پذیر نیست.

3. Branching Factor را توضیح دهید و بگویید که با پیشرفت این بازی چه تغییراتی می‌کند؟
4. توضیح دهید که چرا به هنگام هرس کردن الگوریتم بدون از دست دادن دقت خود سریع‌تر می‌شود.
5. چرا در حالاتی که حریف به صورت شانسی عمل می‌کند (مانند این پروژه)، استفاده از minimax بهینه‌ترین روش نیست؟ چه الگوریتمی می‌تواند جایگزین این الگوریتم باشد؟ توضیح دهید.

نکات پایانی

- دقت کنید که کد شما باید به نحوی زده شده باشد که نتایج قابلیت بازتولید داشته باشند.
- توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید. از ابزارهای تحلیل داده مانند نمودارها استفاده کنید. حجم توضیحات گزارش شما هیچ گونه تاثیری در نمره نخواهد داشت و تحلیل و نمودارهای شما بیشترین ارزش را دارد.
- سعی کنید از پاسخ‌های روشن در گزارش خود استفاده کنید و اگر پیش‌فرضی در حل سوال در ذهن خود دارید، حتما در گزارش خود آن را ذکر نمایید.
- پس از مطالعه کامل و دقیق صورت پروژه، در صورت وجود هرگونه ابهام یا سوال با طراحان پروژه در ارتباط باشید.
- نتایج، گزارش و کدهای خود را در قالب یک فایل فشرده با فرمت AI_CA2_[stdNumber].zip در سامانه ایلرن بارگذاری کنید. به طور مثال AI_CA2_810100999.zip
- محتویات پوشه باید شامل فایل پاسخ‌های شما به سوالات کتبی، فایل jupyter-notebook، خروجی html و فایل‌های مورد نیاز برای اجرای آن باشد. از نمایش درست خروجی‌های مورد نیاز در فایل html مطمئن شوید.
- توجه کنید این تمرین باید به صورت تک‌نفره انجام شود و پاسخ‌های ارائه شده باید نتیجه فعالیت فرد نویسنده باشد. در صورت مشاهده تقلب به همه افراد مشارکت‌کننده، نمره تمرین 100- و به استاد نیز گزارش می‌گردد. همچنین نوشته نشدن کدها توسط هوش مصنوعی نیز بررسی می‌شود!

موفق باشید