



اعضا:

علی پورقیصری

طاها داوری

محمدامین مولوی زاده

فاز سوم پروژه

توضیحات:

این پروژه پیاده سازی بازی Connect Four با استفاده از کتابخانه PettingZoo می باشد. بازی بین دو عامل انجام می شود و این کد با استفاده از الگوریتم minimax با هرس آلفا-بتا پیاده سازی می شود. زیرا اینگونه محیط های دو عاملی با توجه به الگوریتم minimax به راحتی قابل حل و پیاده سازی می باشند. توابع پیاده شده در این کد به صورت زیر می باشند:

تابع count_sublists

در این تابع زیر مجموعه های لیست ورودی که شامل عدد یک که نشان دهنده مهره بازیکن می باشند محاسبه می شوند. این لیست ها در یک لیست ذخیره شده و برگردانده می شود.

تابع get_material_score

این تابع در واقع کار امتیاز دهی به حالت های مختلف صفحه را می کند. در ورودی یک حالت از صفحه، یک بازیکن و یک شرط اتمام می گیرد. حال برای آن بازیکن مهره های کنار هم را محاسبه می کند که این مهره های شامل: دو مهره کنار هم و سه مهره کنار هم می باشند. که این حالات می توانند به صورت سطری، ستونی و قطری باشند. در آخر با دادن ضریب به تعداد دفعات تکرار این مهره ها در کنار هم، یک امتیاز را بر می گردانند. همچنین اگر شرط اتمام صحیح باشد؛ به این معناست که اتمام بازی در این تابع چک می شود. به این صورت که در سطر و ستون و قطر دنباله ۴ تایی را جستجو می کند. اگر پیدا کرد مقدار True و در غیر این صورت مقدار False را بر می گردانند.

تابع heuristic

تابع heuristic امتیاز heuristic را برای یک حالت بازی معین که با آرگومان «observation» نشان داده می شود، محاسبه می کند. تابع دو آرگومان می گیرد: «observation» و «player». آرگومان observation یک دیکشنری حاوی وضعیت فعلی بازی است، در حالی که آرگومان "player" یک عدد صحیح است که نشان می دهد heuristic برای کدام بازیکن محاسبه می شود.

متغیر "my_material" با استفاده از تابع "get_material_score" روی امتیاز عامل هوشمند (یعنی "player_0") تنظیم می شود. متغیر "opponent_material" با استفاده از همان تابع روی امتیاز بازیکن مقابل تنظیم می شود.

سپس متغیر "value" بر روی تفاوت بین امتیاز حریف و امتیاز نماینده تنظیم می شود. این مقدار مزیت یا ضرر نسبی حالت بازی فعلی را برای عامل نشان می دهد. مقادیر مثبت نشان دهنده یک مزیت برای عامل است، در حالی که مقادیر منفی نشان دهنده یک نقطه ضعف است.

در نهایت، متغیر "value" به عنوان مقدار اکتشافی برای وضعیت فعلی بازی برگردانده می شود.

تابع possible_moves

تابع possible_moves وضعیت فعلی بازی Connect Four را مشاهده می کند و فهرستی از حرکات ممکن را که می توان روی محیط بازی انجام داد را برمی گرداند.

تابع minimax

تابع 'minimax' پیاده سازی الگوریتم minimax با هرس آلفا-بتا است. هدف این الگوریتم تعیین بهترین حرکت برای یک بازیکن در یک بازی دو نفره با جمع صفر مانند Connect Four است.

این تابع ابتدا بررسی می کند که آیا جستجو به حداکثر عمق خود رسیده است یا اینکه بازی به پایان رسیده است. اگر چنین است، مقدار اکتشافی وضعیت فعلی را برمی گرداند.

اگر نوبت بازیکن با هدف حداکثر کردن امتیاز آن برسد، تابع در تمام حرکات ممکن تکرار می شود و به صورت بازگشتی خود را با حالت جدید حاصل از هر حرکت فراخوانی می کند. سپس بهترین حرکت و مقدار متناظر آن را بر این اساس به روز می کند.

هرس آلفا بتا برای کاهش تعداد حالت هایی که نیاز به ارزیابی دارند استفاده می شود. آلفا نشان دهنده حداقل امتیازی است که بازیکن حداکثر کننده تضمین شده است که به آن دست یابد. بتا نشان دهنده حداکثر امتیازی است که بازیکن حداقل تضمین شده است که به آن دست یابد. اگر در هر نقطه ای از جستجو، مقدار بتا کمتر یا مساوی با آلفا شود، به این معنی است که حالت های باقی مانده نیازی به ارزیابی ندارند زیرا بر تصمیم نهایی تأثیر نمی گذارند. بنابراین، جستجو را می توان هرس کرد.

اگر نوبت بازیکن کمینه کننده باشد، عملکرد به طور مشابه اما با مقایسه های مخالف برای به روزرسانی بهترین حرکت و مقدار متناظر آن پیش می رود. بهترین حرکت برگشتی نشان دهنده حرکتی است که بازیکن باید انجام دهد تا به بالاترین امتیاز ممکن (یا کمترین امتیاز ممکن بسته به نوبت بازیکن) دست یابد.

تابع make_move

این تابع یک حالت و یک حرکت و همچنین بازیکن را می گیرد و حرکت را با توجه به ماتریس هر بازیکن روی محیط اعمال می کند. سپس اگر آن ستون از مهره پر شده باشد؛ `action_mask` آن حرکت را صفر می کند که دفعات بعدی انتخاب نشود.

تابع undo_move

این تابع دقیقاً خلاف کاری که تابع `make_move` انجام میداد را انجام میدهد برای اینکه حرکاتی که برای محاسبه ارزش حالات به دست می آوریم نباید در محیط باقی بمانند.

توجه شود که در صورت عدم تفهیم درست به کد مراجعه شود. کد به صورت کامل کامنت گذاری شده است.