پروژه ی دوم درس مبانی هوش مصنوعی - پاییز 1400 – علی نوروزی 9831067

قسمت اول تابع evaluation function:

در این قسمت قرار شد که یک evaluation function پیاده سازی شود. این تابع به استیت فعلی پکمن و اکشنی که انجام میدهد بستگی دارد.

پیاده سازی بدینصورت میباشد که در ابتدا به وسیله ی Action داده شده، successor game state که استیت بازی بعد از انجام همان action میباشد.

حال نزدیک ترین فاصله ی ghost فعال بدست آورده شده است و سپس چک میشود که اگر فاصله کمتر مساوی یک بود یک مقدار منفی بینهایت را برگرداند. در غیر این صورت معکوس فاصله تا نزدیکترین غذا برگردانده میشود. چرا که فاصله ی غذا با پکمن رابطه ی عکس دارد و هرچه فاصله کمتر باشد مقدار evaluation بیشتر میباشد.

قسمت دوم جستجوی minimax:

برای انجام این جستجو سه تابع کمکی نوشته شده است. سه تابع مطابق الگوریتم ارائه شده در اسلاید های درس میباشد. البته تغییرات زیادی را داشته اند چرا که در اینجا تعدا ghost ها بیش از یک میباشد. در نتیجه آرگومان ghostIndex اضافه شده است.

تابع اولی minValue میباشد که این تابع توسط گوست ها صدا زده میشود. در این تابع بر روی اکشن ها پیمایش میشود و هر سری successor game stateها به تابع value داده میشود و مینیمم گرفته میشود. نحوه ی پیاده سازی تابع Value در ادامه شرح داده خواهد شد.

همانند تابع minValue یک تابع maxValue وجود دارد که بر روی اکشن ها پیمایش میکند و successor game state به تابع value داده میشود تا مقدار محاسبه شود. این تابع تنها توسط پکمن که دارای agentIndex=0 میباشد اجرا میشود.

تابع Value یک تابع رابط هست که هر سری مشخص میکند که کدام یک از توابع minValue یا maxValue صدا زده بشود. در ابتدای این تابع، یکی به مقدار AgentIndex اضافه میشود و بر تعداد agentها مود گرفته میشود تا افزایش شماره ی Agentها به صورت چرخشی اتفاق بیوفتد.

سپس بررسی میشود که اگر شماره ی Agent برابر با صفر بود یعنی اینکه ایجنت پک‌من میباشد و میخواهد تابع maxValue را صدا بزند. قبل از آن نیز بررسی میشود که اگر به عمق depth رسیده ایم، تابع evaluation صدا زده شود و مقدار آن برگردانده شود. در غیر این صورت تابع maxValue صدا زده میشود.

حال اگر agentIndex بزرگتر از 0 بود آنگاه یعنی نوبت یکی از ghostها میباشد و باید تابع minValue را صدا بزند.

این کار صدا زده شدن maxValue و minValue تا جایی ادامه می یابد که نوبت پک‌من شود، یعنی agentIndex برابر با صفر شود و همچنین به عمق دلخواه رسیده باشیم. در این موقع به هنگام صدا زده شدن تابع value چک میشود که اگر نوبت پک-من بود و به عمق Depth رسیده ایم مقدار evaluation را برگرداند.

حال با توجه به اینکه تابع maxValue مقدار عددی را برمیگرداند و ما action میخواهیم و نه عدد! در تابع getActions اولین مرحله ای که پک‌من تابع max را صدا میزند را بصورت دستی نوشته ام. یعنی بر روی هر کدام از ساکسسورها پیمایش انجام میشود و آن اکشنی که ماکزیمم value میدهد را در یک متغیر ذخیره کرده ام. پس از انجام این پیمایش اکشنی که بیشترین مقدار minimax را دارد، return شده است.

جواب قسمت دستورکار پروژه برای این قسمت:

با توجه به اینکه در جستجوی مینیماکس، ایجنت ghost بهترین عمل خود را انجام میدهد، پس agent پک‌من در مواقعی که راه فراری وجود نداره سریعا خودش را به روح نزدیک میکند و خودکشی میکند!

قسمت سوم alpha beta pruning:

همانند شبه کدی که در دستورکار قرار گرفته شده، در minValue و maxValue شرط های alpha و beta چک شده. همچنین با توجه به اینکه اولین مرحله ای که پک‌من تابع max را صدا میزند را بصورت دسته پیاده کرده ام(توضیحش را در قسمت دوم داده ام) این شرط مربوط به alpha و beta را در تابع getActions نیز نوشته ام.

قسمت چهارم جستجوی expectimax:

این سوال یک تفاوت ریز با سوال دوم دارد. در اینجا با توجه به اینکه ایجنت ghost به صورت تصادفی عمل میکند و احتمال وقوع هریک از اکشن هایی که ghost انجام میدهد برابر میباشد، در تابع minValue که توسط ghostها صدا زده میشود، همه ی valueها جمع زده شده اند و میانگین آنها بدست آورده شده؛ یعنی مجموع Valueها بر تعداد اکشن های ghost تقسیم شده است.

جواب سوال دستورکار پروژه برای این قسمت:

با توجه به اینکه در جستجوی مینیماکس، ایجنت ghost بهترین عمل خود را انجام میدهد، پس agent پک‌من در مواقعی که راه فراری وجود نداره سریعا خودش را به روح نزدیک میکند و خودکشی میکند!

اما در expectimax با توجه به احتمالی که برای اکشن ghost وجود دارد، پک‌من شانس خودش را امتحان میکند و در یک سری از مواقع موجب بردن بازی توسط پک‌من میشود.

قسمت پنجم betterEvaluationFunction:

در این قسمت باید یک تابع evaluation را تنها بر اساس استیت فعلی بنویسیم.

ابتدا فاصله تا نزدیک ترین ghost فعال و همچنین فاصله تا نزدکترین ghost ترسیده به دست آورده شده. سپس بررسی شده که اگر فاصله تا نزدیکترین ghost فعال کمتر مساوی 2 باشد، یعنی پک‌من باید احساس خطر کند و مقدار منفی بی نهایت برگردانده میشود. سپس بررسی میشود که scared ghost یافت شده و اگر یافت شده بود، score فعلی به علاوه ی 5برابر

معکوس فاصله تا نزدیکترین scared ghost برگردانده میشود. در اینجا فاصله ی scared ghost با پک‌من رابطه ی عکس دارد و هرچه فاصله کمتر باشد مقدار evaluation بیشتر میباشد.

سپس فاصله تا نزدیکترین غذا بدست می‌آید. اگر فاصله تا نزدیکترین غذا برابر با صفر بود، مقدار مثبت بی‌نهایت را برمیگرداند.

در غیر اینصورت Score فعلی به علاوه ی منفی 8 برابر فاصله تا نزدیکترین روح فعال و به علاوه ی 4برابر فاصله تا نزدیکترین غذا برگردانده میشود. در اینجا ضریب فاصله با روح بیشتر از ضریب فاصله با غذا میباشد و این به دلیل اهمیت بیشتر و خطری است که روح فعال، پک‌من را تهدید میکند.