فرهاد امان ۹۹۳۱۰۰۶

۲- الف) تابع normalize در این بخش فراخوانی می‌شود تا اطمینان حاصل کنیم که مجموع احتمالات در self.beliefs برابر یک است. در واقع، self.beliefs نماینده احتمالاتی است که با توجه به مشاهدات قبلی و مکان فعلی پکمن، به موقعیت‌های مختلف احتمالی ارواح نسبت داده می‌شود. اما به دلیل اعمال عملیات ضرب بر روی این احتمالات در دوره زمانی، ممکن است مجموع احتمالات دیگر برابر با یک نباشد و مقادیر در self.beliefs به طور غیرمعقولی تغییر کنند.

ب) در واقع beliefs نماینده توزیع احتمالی فعلی ما در مورد موقعیت احتمالی ارواح است. در ابتدا، beliefs توزیع احتمالی یکنواخت بر روی همه موقعیت‌های ممکن ارواح دارد. سپس، با هر مشاهدهٔ جدید، beliefs بروزرسانی می‌شود.
در تابع beliefs ، observeUpdate بر اساس مشاهده فاصله‌ای که به صورت نویزی به دست آمده است و موقعیت فعلی پکمن، به روزرسانی می‌شود. برای هر موقعیت ممکن برای ارواح (شامل موقعیت زندان)، احتمال بروز وقوع مشاهده فعلی در آن موقعیت محاسبه می‌شود و با احتمال فعلی موقعیت، ضرب می‌شود. به عبارت دیگر، هر موقعیت در beliefs در نظر گرفته شده و به آن احتمال جدیدی نسبت داده می‌شود که با توجه به مشاهده فعلی، این احتمال بروز وقوع مشاهده را نشان می‌دهد.

۳- الف) کلاس DiscreteDistribution یک مدل برای توزیع‌های احتمالی و وزن‌های مربوط به مجموعه‌ای متناهی از کلیدهای گسسته است. این کلاس از کلاس dict ارث‌بری کرده و به عنوان یک وراثت دیکشنری عمل می‌کند، که کلیدها را به همراه مقادیر مرتبط نگه می‌دارد.

__getitem__ عملکرد خود را از کلاس والد dict ارث‌بری کرده و برای دریافت مقدار مرتبط با یک کلید مشخص استفاده می‌شود. در صورتی که کلید در دیکشنری وجود نداشته باشد، مقدار آن را به صفر تنظیم می‌کند و سپس مقدار مربوطه را بازگردانی می‌کند.

copy این متد یک نسخهٔ کپی از توزیع را برمی‌گرداند. با استفاده از dict.copy، یک کپی از دیکشنری اصلی ایجاد می‌شود و سپس این کپی را در یک نمونه جدید از DiscreteDistribution برمی‌گرداند.

argMax این متد کلیدی را با بیشترین مقدار در توزیع پیدا می‌کند و آن را برمی‌گرداند. در صورتی که توزیع خالی باشد، None را برمی‌گرداند.

total این متد مجموع مقادیر برای همهٔ کلیدها را برمی‌گرداند.

normalize این متد توزیع را به نحوی نرمالیزه می‌کند که مجموع مقادیر همهٔ کلیدها برابر یک شود. در صورتی که مجموع توزیع صفر باشد، هیچ تغییری اعمال نمی‌شود. با استفاده از self.total، مجموع مقادیر را محاسبه کرده و سپس تمام مقادیر را بر اساس این مجموع نرمالیزه می‌کند.

sample این متد یک نمونه تصادفی از توزیع را برمی‌گرداند و کلید مربوطه را با توجه به وزن‌های مرتبط با هر کلید انتخاب می‌کند. این انتخاب بر اساس احتمال وزن‌ها صورت می‌گیرد. برای این کار، ابتدا تمام نمونه‌های تصادفی را ایجاد کرده و سپس تعداد وقوع هر کلید را شمارش می‌کند تا بتواند احتمال تقریبی وزن‌ها را محاسبه کند.

ب) ساختار شبکه بیزین در پروژه وابستگی‌های بین متغیرها را به صورت گرافیکی نشان می‌دهد. این گراف نشان می‌دهد که کدام متغیرها به کدام متغیرها وابسته هستند و ارتباطات آن‌ها را نشان می‌دهد. در شبکه بیزین، گره‌ها نمایانگر متغیرها هستند و یال‌ها نمایانگر وابستگی‌ها بین متغیرها می‌باشند. این وابستگی‌ها نشان می‌دهد که چگونه اطلاعات و تغییرات در یک متغیر ممکن است تاثیرگذار باشد و متغیرهای دیگر را تحت تاثیر قرار دهد.

گره نشان دهنده متغیر موقعیت فعلی روح‌ها است. این متغیر می‌تواند مقادیر مختلفی مانند موقعیت‌های ممکن روح‌ها را به عنوان دامنه داشته باشد. این گره می‌تواند وابستگی به خود داشته باشد (یعنی موقعیت فعلی روح تاثیرگذار بر موقعیت بعدی روح باشد) و همچنین می‌تواند به گره مشاهده نیز وابسته باشد (یعنی مشاهده‌ها تاثیرگذار بر موقعیت فعلی روح‌ها باشند).

گره مشاهده نشان دهنده متغیر مشاهده‌ها است، که می‌تواند شامل مقادیر مختلفی از مشاهده‌های ممکن مانند فاصله صداشده توسط روح‌ها باشد. این گره می‌تواند به خود وابستگی داشته باشد (یعنی مشاهده فعلی تاثیرگذار بر مشاهده بعدی باشد) و همچنین می‌تواند به گره موقعیت روح و تصمیم نیز وابسته باشد.

گره تصمیم نشان دهنده متغیر تصمیم‌ها است، که می‌تواند شامل مقادیر مختلفی از تصمیم‌های ممکن مانند جابجایی بین انواع تصمیم‌ها باشد. این گره می‌تواند به گره مشاهده وابسته باشد (یعنی مشاهده تاثیرگذار بر تصمیم‌ها باشد) و همچنین می‌تواند به گره موقعیت روح نیز وابسته باشد.

این ساختار شبکه بیزین به ما اجازه می‌دهد تا وابستگی‌ها و تاثیرات بین متغیرها را به صورت واضح درک کنیم و با استفاده از این اطلاعات، استنتاج و تحلیل‌های لازم را برای ردیابی روح‌ها در بازی پکمن انجام دهیم.