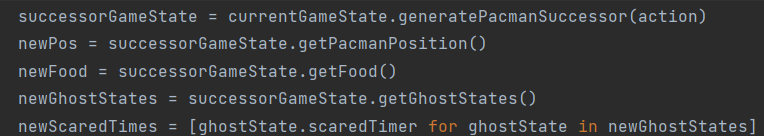
**گزارش فاز دوم پروژه مبانی هوش مصنوع کاربردی**

1. **عامل عکس العمل:**

ابتدا متغیر های زیر را حساب میکنیم:



در قدم بعدی چک میکنیم که آیا پکمن فاصله امنی با روح ها دارد یا خیر، برای چک کردن این مطلب فاصله منهتن موقعیت فعلی پکمن و روح ها را محاسبه کرده و مینیموم آن را بدست می آوریم، حداقل فاصله پکمن با روح ها باید از حد امن بیشتر باشد

minGhostDistance = min([manhattanDistance(newPos, state.getPosition()) for state in newGhostStates])

چک میکنیم آیا این عمل موجب افزایش امتیاز شده است یا خیر، برای چک کردن این مطلب امتیاز موقعیت وارث را از موقعیت فعلی کم میکنیم

scoreDiff = successorGameState.getScore() - currentGameState.getScore()

چک میکنیم آیا این عمل موجب نزدیک تر شدن به غذای های نزدیک پکمن شده است یا نه

pos = currentGameState.getPacmanPosition()  
nearestFoodDistance = min([manhattanDistance(pos, food) for food in currentGameState.getFood().asList()])  
newFoodsDistances = [manhattanDistance(newPos, food) for food in newFood.asList()]  
newNearestFoodDistance = 0 if not newFoodsDistances else min(newFoodsDistances)  
isFoodNearer = nearestFoodDistance - newNearestFoodDistance

موقعیت فعلی را ذخیره میکنیم

direction = currentGameState.getPacmanState().getDirection()

حالا ارزیابی را آغاز میکنیم، اگر پکمن فاصله کافی با روح هارا نداشته باشد اصلا عمل خوبی نداشته ایم پس:

if minGhostDistance <= 1 or action == Directions.STOP:  
 return 0

اگر عمل پکمن موجب بیشتر شدن امتیاز آن شده باشد بهترین عملکرد را طبیعتا داشته ایم:

if scoreDiff > 0:  
 return 8

اگر به غذاهای نزدیک، نزدیکتر شده ایم:

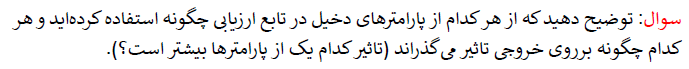
elif isFoodNearer > 0:  
 return 4

اگر عمل ما در راستای جهت فعلی بوده :

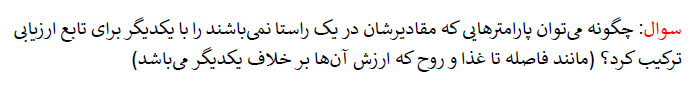
elif action == direction:  
 return 2

و درغیراین صورت عمل ما به سود خاصی نرسیده پس :

else:  
 return 1



در توضیح کد کامل توضیح داده شده و تاثیر افزایش امتیاز منطقا بیشتر است زیرا به دنبال همین مطلب در بازی هستیم.

ما پس از محاسبه این مقادیر دانه به دانه اینهارا چک میکنیم و درصورتی که در ارزیابی مفید واقع شوند( چه کم و چه زیاد بودنشان) به آنها نمره میدهیم

1. مینماکس

برای پیاده سازی این بخش ابتدا در تابع getAction دو زیر تابع تعریف میکنیم:

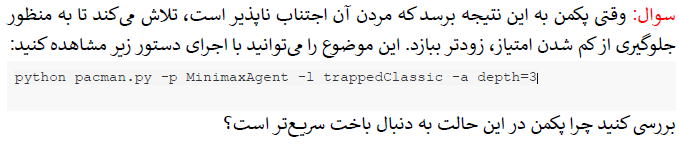
def minValue(state, agentIndex, depth):  
 legalActions = state.getLegalActions(agentIndex)  
 if not legalActions: # NO legalActions means game finished(win or lose)  
 return self.evaluationFunction(state)  
  
 # When all ghosts moved, it's pacman's turn  
 if agentIndex == state.getNumAgents() - 1:  
 return min(maxValue(state.generateSuccessor(agentIndex, action), depth) for action in legalActions)  
 else:  
 return min(minValue(state.generateSuccessor(agentIndex, action), agentIndex + 1, depth) for action in  
 legalActions)

ابتدا لیست legal actions را برای یک اجنت میگیریم، اگر legal action ای موجود نبود یعنی بازی تمام شده چه برد و چه باخت داده باشیم. گفته بودیم مقدار ایندکس پکمن 0 است پس چک میکنیم اگر تعداد اجنت ها منهای 1 برابر ایندکس باشد یعنی نوبت پکمن است و دیگر روحی در بازی نیست پس حداقل مقداری که تابع maxValue برمیگرداند را برای آن میخواهیم وگرنه یعنی روحی در بازی هست پس حداقل مقدار را برای آن برمیگردانیم

def maxValue(state, depth):  
 legalActions = state.getLegalActions(0)  
 if not legalActions or depth == self.depth:  
 return self.evaluationFunction(state)  
  
 return max(minValue(state.generateSuccessor(0, action), 0 + 1, depth + 1) for action in legalActions)  
  
bestAction = max(gameState.getLegalActions(0),  
 key=lambda action: minValue(gameState.generateSuccessor(0, action), 1, 1))  
return bestAction

در تابع maxvalue، legal actionsرا برای پکمن پیدا میکنیم روند مشابهی با تابع قبلی پیش میرود اما اینجا حداکثر مقداری که تابع minValue برای حرکت پکمن پیدا میکند را برمیگردانیم

نهایتا بهترین عمل برگردانده میشود.



طبیعتا اگر راه سودآوری نباشد صفر بهتر از ضرر کردن است بعبارتی با ادامه دادن پکمن تنها ضرر میکند و نتیجتا ضرر کمتر را ترجیح خواهد داد

1. حرس آلفا و بتا

در این قسمت بنابر کد موجود در دستور کار پیش میرویم، ابتدا تابع minValue را تعریف میکنیم:

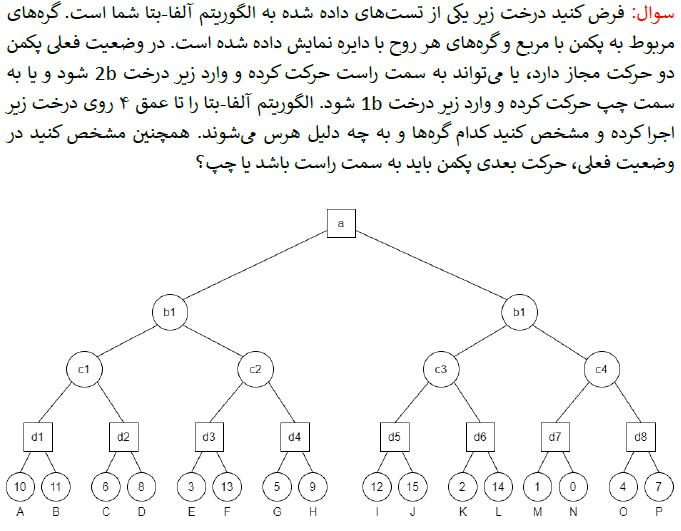
def minValue(state, agentIndex, depth, a, b):  
 legalActions = state.getLegalActions(agentIndex)  
 if not legalActions:  
 return self.evaluationFunction(state)  
  
 v = Infinity  
 for action in legalActions:  
 newState = state.generateSuccessor(agentIndex, action)  
 # Is it the last ghost?  
 if agentIndex == state.getNumAgents() - 1:  
 newV = maxValue(newState, depth, a, b)  
 else:  
 newV = minValue(newState, agentIndex + 1, depth, a, b)  
  
 v = min(v, newV)  
 if v < a:  
 return v  
 b = min(b, v)  
 return v

سپس تابع maxValue پیاده سازی میشود:

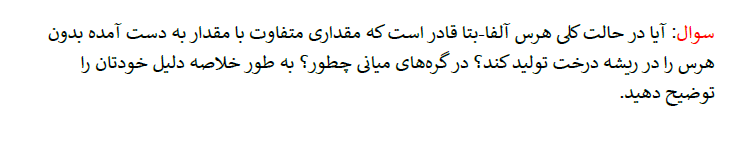
def maxValue(state, depth, a, b):  
 legalActions = state.getLegalActions(0)  
 if not legalActions or depth == self.depth:  
 return self.evaluationFunction(state)  
  
 v = -Infinity  
 # For enable second play pruning  
 if depth == 0:  
 bestAction = legalActions[0]  
 for action in legalActions:  
 newState = state.generateSuccessor(0, action)  
 newV = minValue(newState, 0 + 1, depth + 1, a, b)  
 if newV > v:  
 v = newV  
 if depth == 0:  
 bestAction = action  
 if v > b:  
 return v  
 a = max(a, v)  
  
 if depth == 0:  
 return bestAction  
 return v

سپس با فراخوانی تابع maxValue با توجه به اینکه پکمن میخواهد اجرا کند بهترین عمل را برمیگردانیم.

bestAction = maxValue(gameState, 0, -Infinity, Infinity)  
return bestAction



از چپ ترین شاخه شروع میکنیم بین 10و11 یازده را انتخاب میکنیم (مقدار حداکثری برای پکمن) در گره والد روح میخواهد حداقل انتخاب کند 11 بزرگتر از هشت است و شرط حرس در چنین شرایطی بزرگتر بودن شاخه است پس حرسی انجام نشده هشت بالا می آید نهایتا روح c1 8 را انتخاب میکند. هشت همچنان بزرگتر از 3 است پس حرسی اینجا هم هنوز انجام نمیشود بین 3 و 13 سیزده بالا می آید. در مرحله بعدی هم روح داریم پس 13 که بزرگ تر از 5 است حرسی انجام نمیدهد بین 5 و 9 نه انتخاب شده و بالا می آید و روح c2 9 را انتخاب میکند نهایتا روح b1 با مقدار نه بالا میرود. حال سراغ دوازده میرویم که بزرگتر از 9 بوده و همچنان حرسی انجام نمیشود، بین 12 و 15 پانزده بالا می آید و چون از 2 بزرگتر است باز هم حرصی نداریم 14 بالا رفته و روح c314 را بالا میبرد که بزرگتر از 1 است همچنان حرسی نداریم بین 1 و 0 یک بالا میرود 1 اما از 4 بزرگتر نیست پس شاخه مربوط به انتخاب d8حرس میشود و انتخاب c4 یک خواهد بود و b1 هم یک را بالا میبرد حال پکمن a بین 9 و 1 نه را انتخاب میکند.

پس انتخاب پکمن سمت چپ خواهد بود