HW2 report

B05901084 電機四 劉容均

1. 簡介

我的betterEvaluationFunction和ReflexAgent的evaluationFunction的使用了相似的演算法，差別在於evaluationFunction只能計算到下一回合的可能性，和動態的局面變化，better是計算靜態state的局勢優劣。Score的計算方式如下：

將score拆解成三個子項目，w是權重weight，調整權重可以改變各項目的優先度。e則是各項目的evaluation，之中也有不同的權重，接下來會一一說明計算方式，

1. food

場上食物越少，分數越高；若食物數量相同，則離最近食物的距離越近，分數越高。實作上採用扣掉場上食物數量，同時扣掉最近食物的距離。兩種分數也設有權重，兩權重比值必須夠大，避免發生下一個食物太遠時，pacman為了最近食物距離的分數而不吃下已在面前的食物。例如設100：1，當下一個食物距離大於100步時，不吃的分數會比吃的分數高。

1. ghost

ghost分為active和scared，由於ghost只有在靠近pacman時需要考慮逃跑或追逐，太遠的active ghost不需要考慮，太遠的scared ghost也不用花心力去追，因此兩者採用exponential的方式計算evaluation，距離active ghost愈近分數急遽下降，距離scared ghost愈近分數急遽上升。其中active ghost 的decayFactor較大，因為pacman和ghost的速度相同，即使在非常近的距離才開始逃跑也不易被追上，所以距離3以後影響分數的幅度極小。當吃下capsule轉為scared ghost，追逐scared ghost優先度變成第一，因此設置最大的權重。

這邊會遇到一個難題，當pacman即將吃下ghost時，因為ghost會變回active ghost，並回到重生點，此時靠近scared ghost而上升的分數會突然消失，導致pacman停下不吃ghost。我想到的方式是以active ghost數量調整分數，補償消失的scared ghost分數，但此時又會遇到因為吃capsule會讓active ghost減少，active ghost數量分數消失而不吃capsule，最後以capsule的數量調整分數，補償active ghost數量分數，吃ghost的機制才正常運作。此外，由於與ghost的互動攸關存亡，這部分額外設置了幾種判斷機制，避免找死的情況發生。例如不能在重生點吃ghost，不然active ghost會重生在旁邊；當ghostState.scaredTimer倒數快到時，放棄追逐。

1. capsule

capsule在遊戲中非必須吃光的東西，但要衝高分數，在ghost接近時吃capsule反追非常重要，因此使用以下方式計算capsule分數：

其中判斷是否有機會先一步走到，判斷距離， 的參數可以判斷ghost遠近，在適合的時機吃下capsule。此分數在同時靠近ghost和capsule時影響幅度大過active ghost接近減去的分數，使反殺的優先度高過逃跑。

1. minMazeDistance

由於mazeDistance是計算兩點距離，在尋找最近食物時，需要執行相當於場上食物數量的bfs，相當耗時，因此我重寫FoodSearchProblem，將goal state改成第一個遇到的食物，因為bfs由淺到深的搜尋，可以確保這是最近的食物。MinMazeDistance呼叫這個problem，並回傳路徑長度。

1. 各項目權重

根據優先度，當ghost極遠時，是影響分數的主要因素，且capsule也不一定要吃。當ghost靠近時，逃跑變為優先，而同時也靠近capsule時，吃capsule為優先，變成scared ghost後追逐變為最優先。

1. 討論

這次作業的優化重點放在生存，因為死亡的懲罰太大，其次才是利益最大化。經過一連串改良和優化，最終pacman的勝率近97%（-n 30 -f），場上有兩隻ghost時，平均分數達1500以上，觀察失敗案例，幾乎都是夾殺，原因是depth不夠深，無法算到另一端的ghost。也有一些未實作進作業中的想法，例如在吃完scared ghost前不吃下一個capsule、減少跑進死路的機會（這次作業自己沒有使用到wall的功能），讓分數能達到最高。此外還有一些難題尚未克服，例如因為計算深度關係，pacman會提早迴避ghost，這是reflex agent不會有的情況，因為reflex agent只算到下一個action。

1. 結果

