ARTIFICIAL INTELLEGNECE

Lab 03

Abstract

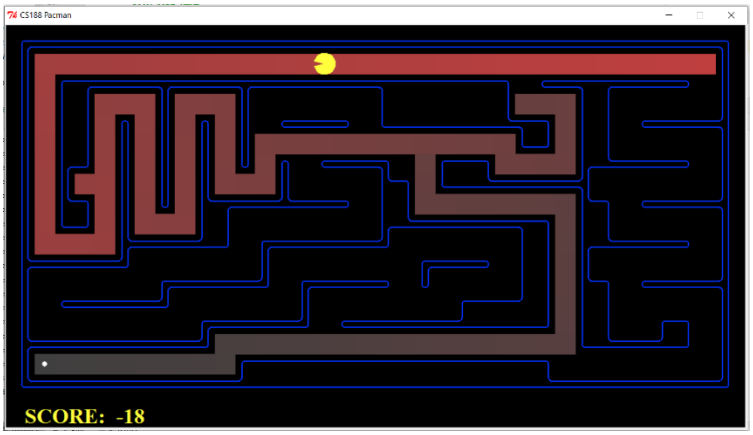
Depth First Search, Breath First Search and Varying Cost Function

Syed Muhammad Akbar & Muhammad Haris Owais Ahmed

Msyed.bscs17seecs

TASK 01

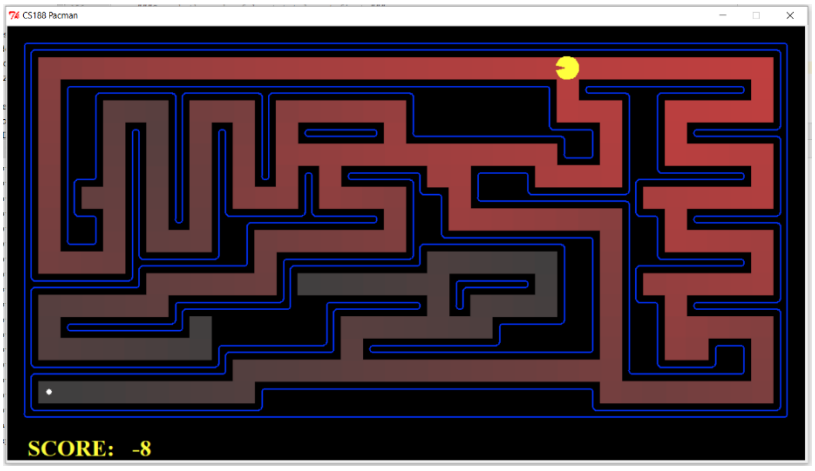
```
import util

Next = util.Stack()
Visited = []
current_position = problem.getStartState()
Next.push((current_position, []))
Visited.append(current_position)

while Next.isEmpty() == 0:
    current_position, actions = Next.pop()

    for next in problem.getSuccessors(current_position):
        next_position = next[0]
        next_direction = next[1]
        if next_position not in Visited:
            if problem.isGoalState(next_position):
                return actions + [next_direction]
            else:
                print(Next)
                Next.push((next_position, actions + [next_direction]))
                Visited.append(next_position)
```




```
<util.Queue instance at 0x0000000003949788>
<util.Queue instance at 0x0000000003949788>
<util.Queue instance at 0x0000000003949788>
<util.Queue instance at 0x0000000003949788>
Path found with total cost of 68 in 0.3 seconds
Search nodes expanded: 267
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win
```

TASK 02

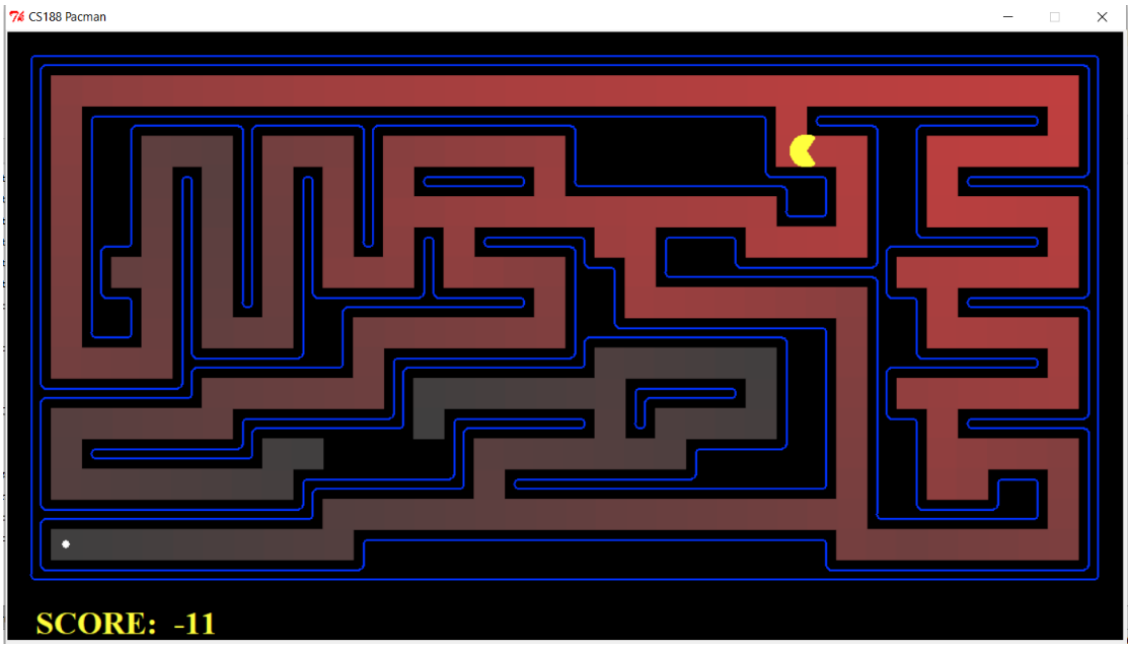
```
import util

Next = util.Queue()
Visited = []
current_position = problem.getStartState()
Next.push((current_position, []))
Visited.append(current_position)

while Next.isEmpty() == 0:
    current_position, actions = Next.pop()

    for next in problem.getSuccessors(current_position):
        next_position = next[0]
        next_direction = next[1]
        if next_position not in Visited:
            if problem.isGoalState(next_position):
                return actions + [next_direction]
            else:
                print(Next)
                Next.push((next_position, actions + [next_direction]))
                Visited.append(next_position)
```




```
<util.Queue instance at 0x00000000039B9788>
<util.Queue instance at 0x00000000039B9788>
<util.Queue instance at 0x00000000039B9788>
<util.Queue instance at 0x00000000039B9788>
Path found with total cost of 68 in 0.2 seconds
Search nodes expanded: 267
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win
```

TASK 03

```
import util import heapq

def _update(Next, item, priority):
    for index, (p, c, i) in enumerate(Next.heap):
        if i[0] == item[0]:
            if p <= priority:
                break              del Next.heap[index]
            Next.heap.append((priority, c, item))
            heapq.heapify(Next.heap)
            break      else:
        Next.push(item, priority)

Next = util.PriorityQueue()
Visited = []
current_position = problem.getStartState()
Next.push((current_position, []), 0)
Visited.append(current_position)

while Next.isEmpty() == 0:
    current_position, current_direction = Next.pop()

    if problem.isGoalState(current_position):
        return current_direction

    if current_position not in Visited:
        Visited.append(current_position)

    for next in problem.getSuccessors(current_position):
        next_position = next[0]
        next_direction = next[1]
        if next_position not in Visited:
            _update(Next, (next_position, current_direction +
[next_direction]), problem.getCostOfActions(current_direction +
[next_direction]))
```



```
(venv) D:\Assignments\AI\Lab03\search>python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.0 seconds
Search nodes expanded: 269
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win
```