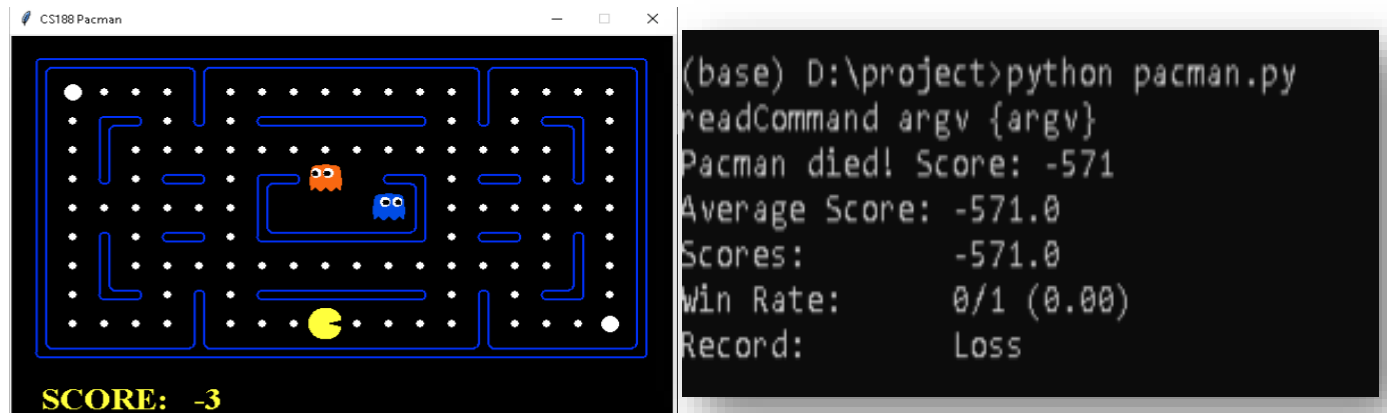


## Project output

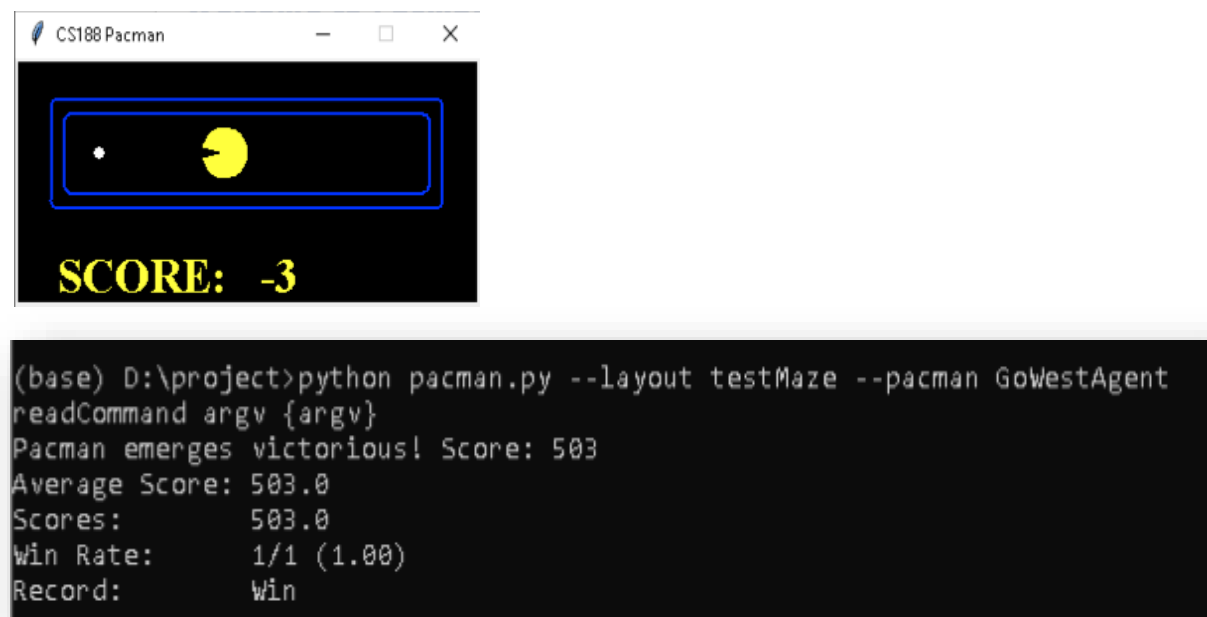
```
python pacman.py
```

OUTPUT:



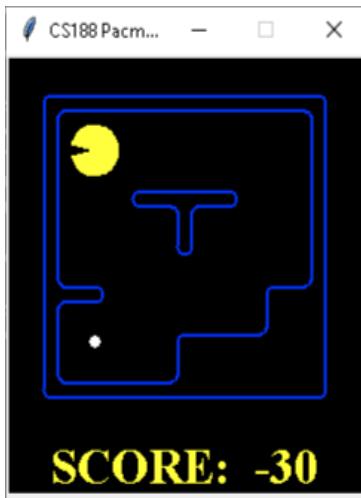
```
python pacman.py --layout testMaze --pacman GoWestAgent
```

OUTPUT:



```
python pacman.py --layout tinyMaze --pacman GoWestAgent
```

OUTPUT:

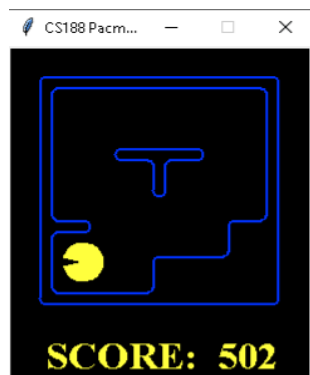


```
(base) D:\project>python pacman.py --layout tinyMaze --pacman GoWestAgent  
readCommand argv {argv}  
(base) D:\project>_
```

## Question 1: Finding a Fixed Food Dot using Depth First Search

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch
```

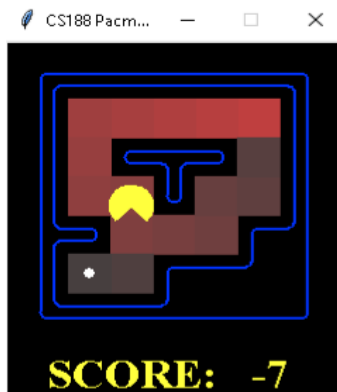
OUTPUT:



```
(base) D:\project>python pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch
readCommand argv {argv}
[SearchAgent] using function tinyMazeSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 8 in 0.0 seconds
Search nodes expanded: 0
Pacman emerges victorious! Score: 502
Average Score: 502.0
Scores:      502.0
Win Rate:    1/1 (1.00)
Record:      Win
```

```
python pacman.py -l tinyMaze -p SearchAgent
```

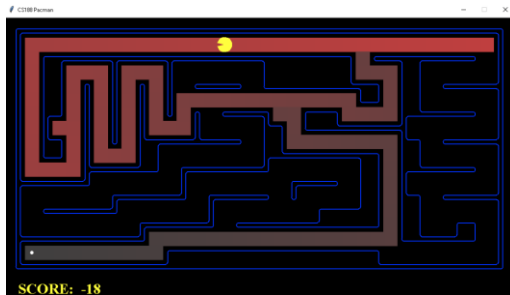
## OUTPUT:



```
(base) D:\project>python pacman.py -l tinyMaze -p SearchAgent
readCommand argv {argv}
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 10 in 0.0 seconds
Search nodes expanded: 15
Pacman emerges victorious! Score: 500
Average Score: 500.0
Scores:      500.0
Win Rate:    1/1 (1.00)
Record:      Win
```

```
python pacman.py -l mediumMaze -p SearchAgent
```

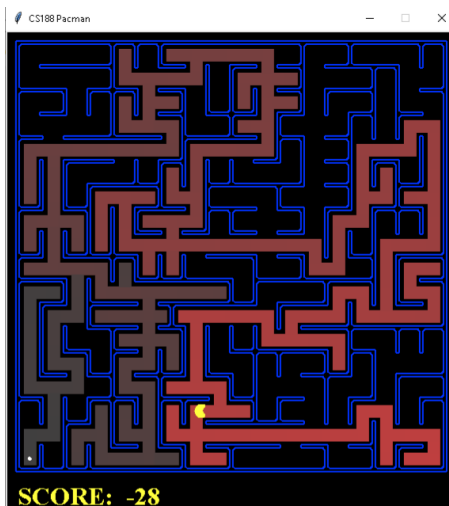
## OUTPUT:



```
(base) D:\project>python pacman.py -l mediumMaze -p SearchAgent
readCommand argv {argv}
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 130 in 0.0 seconds
Search nodes expanded: 146
Pacman emerges victorious! Score: 380
Average Score: 380.0
Scores:      380.0
Win Rate:    1/1 (1.00)
Record:      Win
```

```
python pacman.py -l bigMaze -z .5 -p SearchAgent
```

## OUTPUT:

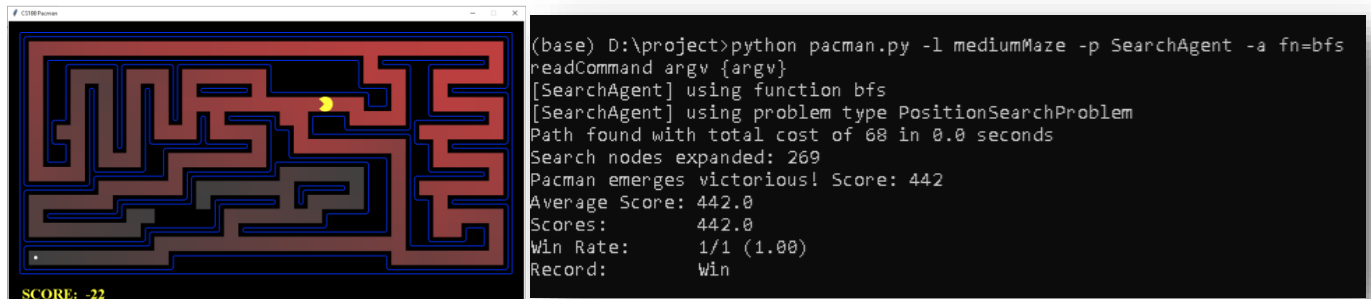


```
(base) D:\project>python pacman.py -l bigMaze -z .5 -p SearchAgent
readCommand argv {argv}
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.0 seconds
Search nodes expanded: 390
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores:      300.0
Win Rate:    1/1 (1.00)
Record:      Win
```

## Question 2 : Breadth First Search

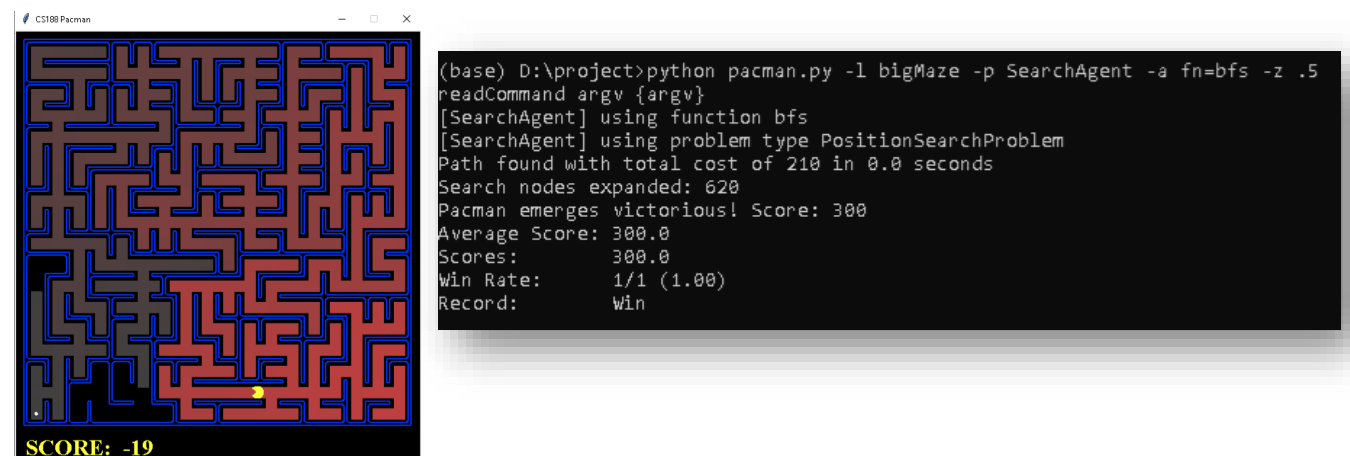
```
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
```

OUTPUT:



```
python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
```

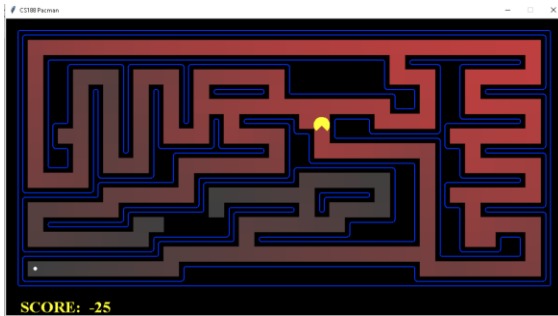
OUTPUT:



### Question 3: Varying the Cost Function

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
```

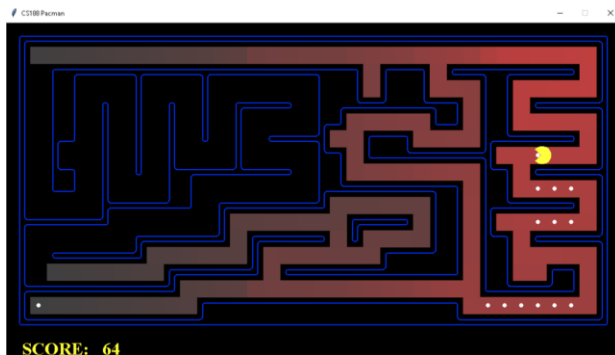
OUTPUT:



```
(base) D:\project>python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
readCommand argv {argv}
[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
```

```
python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
```

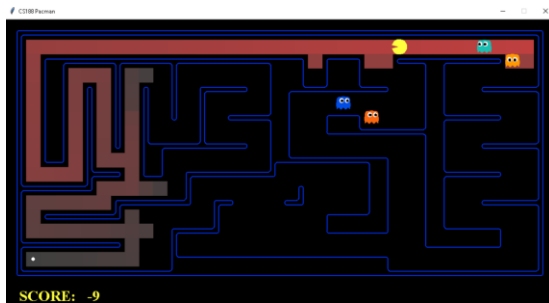
OUTPUT:



```
(base) D:\project>python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
readCommand argv {argv}
Path found with total cost of 1 in 0.0 seconds
Search nodes expanded: 186
Pacman emerges victorious! Score: 646
Average Score: 646.0
Scores:      646.0
Win Rate:    1/1 (1.00)
Record:      Win
```

```
python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
```

OUTPUT:

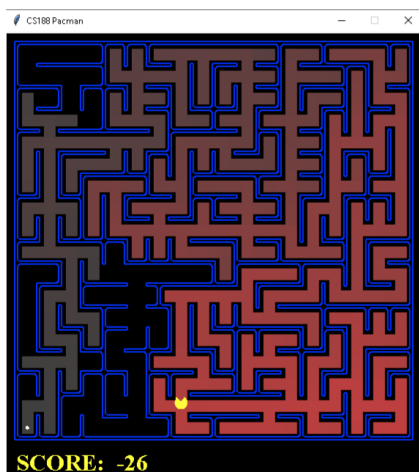


```
(base) D:\project>python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
readCommand argv {argv}
Path found with total cost of 68719479864 in 0.0 seconds
Search nodes expanded: 108
Pacman emerges victorious! Score: 418
Average Score: 418.0
Scores:      418.0
Win Rate:    1/1 (1.00)
Record:      Win
```

## Question 4: A\* search

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a
fn=astar,heuristic=manhattanHeuristic
```

OUTPUT:



```
(base) D:\project>python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
readCommand argv {argv}
[SearchAgent] using function astar and heuristic manhattanHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.0 seconds
Search nodes expanded: 549
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores:      300.0
Win Rate:    1/1 (1.00)
Record:      Win
```

فاطمه خدادادی\_97143013