Project 01

1. **Group member**

* 1651025 – Nguyễn Trung Nghĩa
* 1651049 – Chu Đức Khánh

1. **Assignment plan**

* **Level 1:** We use BFS start from destination (food position). We use another 2d array to save path length: b[i][j] is min path length from [I,j] to destination; another 2d array to save previous position of min path: pre[i][j] is coordinate of places which Pacman go to before go to [i][j]. Finally, we just base on pre[i][j] to get min path.

Time complexity: O(m\*n)

Space complexity: O(m\*n)

* **Level 2:** The same as level 1 (we can treat still monster like wall)
* **Level 3:** We use DFS start from initial Pacman position. We use another 2d array to check whether Pacman has been certain position or not. Algorithm stop when Pacman have visited all places.

Time complexity: O(4m\*n)

Space complexity: O(m\*n)

* **Level 4:** Haven’t done yet

1. **Environment**

User can compile by GDB and run program in Linux, Microsoft Windows 7, 8, 10.

Program had written by C++, compiled and debugged by GDB.

1. **Requirements review**

|  |  |  |
| --- | --- | --- |
| No. | Specifications | Scores |
| 1 | Finish level 1 successfully. | 20% |
| 2 | Finish level 2 successfully. | 20% |
| 3 | Finish level 3 successfully. | 10% |
| 4 | Haven’t done level 4 | 0% |
| 5 | Graphical demonstration of each step of the running process. You can demo in console screen or use any other graphical library.  Run pacman.exe | 10% |
| 6 | Generate at least 5 maps with difference in number and structure of walls, monsters, and food.  See in maps.txt | 10% |
| 7 | Report your algorithm, experiment with some reflection or comments.  See 2(assignment plan) above. | 20% |
| Total | | 90% |

1. **References**