**Artificial Intelligence Spring 2022 Group Game Project Spec (Pacman)**

* The time limit for each move is 0.04 seconds.
* If your program return answer out of time, your role will continue the control of last step.
* At the end of game, the total score will print on the command line board.

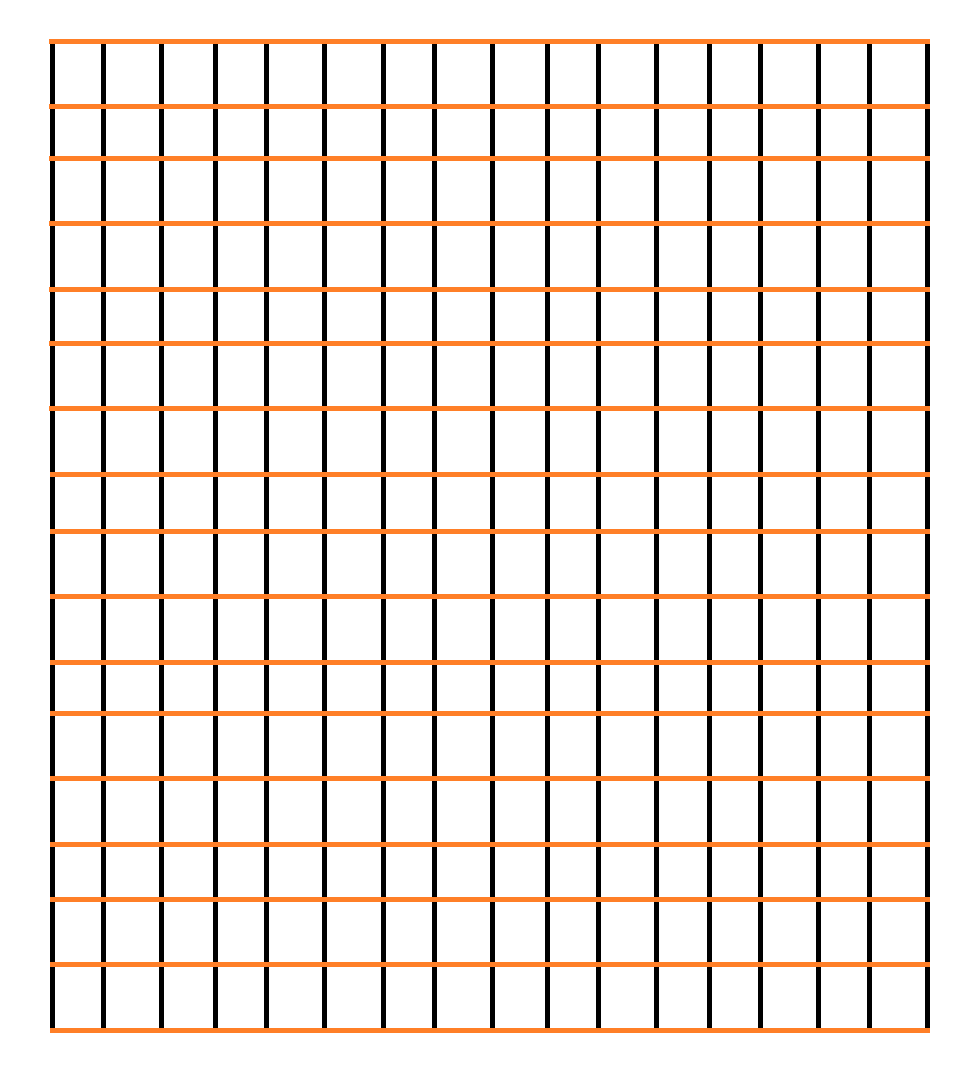
**About programming:**

You are allowed to use Python (3 or above)/C++ (14 or below) to code your program. TAs will provide code including TCP socket and a basic code template. For simplicity, you only need to fill in the function provided in template code.

* General
  + You don’t need to worry about the connection between your program and the tournament judge if you follow the template code. The template code provides a function GetStep for you to fill in. It has three parameters (playerStat, ghostStat, propsStat) and the returned object contains the current control by your program.
  + Function GetStep will be called every time step and the control made by your program will be verified in server. That is, you don’t need to worry about the situation your control is not valid.
  + Parameter **playerStat** and **otherPlayerStat** contain the status of player which is stored in list of list (python) and array (C++); playerStat: **[x, y, n\_landmine, super\_time]**
    - x and y mean position of player in gameboard.
    - Landmine means landmine the player having.
    - super\_time means the remaining super time for the player.

otherPlayerStat is same as playerStat but it is 3\*4 array for other three players.

* + Parameter **ghostStat** contain the status of ghosts. It contains [[x, y],[x, y],[x, y],[x, y]]. X and y mean position of ghosts.
  + Parameter **propsStat** contain the status of all kinds of eaten object. PropsStst shape is [[type, x, y] \* n] for total n props. For type, 0 is landmine, 1 is power, 2 is pellet, 3 is bomb. X and y mean position of object.
  + Parameter **parallel\_wall** and **vertical\_wall** is the array that describe the wall in the game map. Parallel is a 16 \* 17 array and vertical\_wall is 17 \* 16 array. In these two arrays, 1 means there is a wall and 0 means there is a road.   
    Below is the explanation image, orange line is parallel\_wall and black one is vertical\_wall.



* Python
  + **STcpClient.py**: This code includes TCP functions. You should not change it except that you must change the number at line “idTeam = -1” into your team number “idTeam = <team\_number>”. We will announce the team number later.
  + **Sample.py:** This is template code which includes function GetStep to be filled by you.
  + We will run your code in python 3.6 or above with win10 operating system. Your source code should be able to run directly by python interpreter.
* C++
  + **STcpClient.h**: This code includes TCP functions. You should not change it except that you must change the number at line 18 “idTeam = -1” into your team number “idTeam = <team\_number>”. We will announce the team number later.
  + **Sample.cpp**: This is template code which includes function GetStep to be filled by you.
  + Notice that template code is written in Visual Studio 2017 with win10 operating system and includes OS-depended code (winsock for TCP) and compiler depended code (#pragma). You can compile your program following this website <https://docs.microsoft.com/zh-tw/cpp/build/walkthrough-compiling-a-native-cpp-program-on-the-command-line?view=msvc-170> .
  + We will run your code in win10 operating system. DO NOT submit the whole project directory. Instead, just submit STcpClient.h and Team\_yourTeamnumber.cpp.

**TAs strongly recommend you to use the same compiler and operating system as mentioned above to ensure there won’t be any unexpected problems.**