**VIETNAM NATIONAL UNIVERSITY**

**UNIVERSITY OF SCIENCE**



**TOPIC**

**Project 02**

MEMBERS

Vũ Công Minh 18127155

Lư Ngọc Liên 18127046

**Course: Artificial Intelligence**

**Ho Chi Minh City– 2020**

# Contents

[Contents 4](#_Toc49786557)

[Assignment Plan 5](#_Toc49786558)

[Environment to compile and run the program 6](#_Toc49786559)

[Estimating the degree of completion level for each requirement 7](#_Toc49786560)

[Instruction 8](#_Toc49786561)

[Algorithm explanation 10](#_Toc49786562)

[Comparison between 2 strategy mode 15](#_Toc49786563)

# Assignment Plan

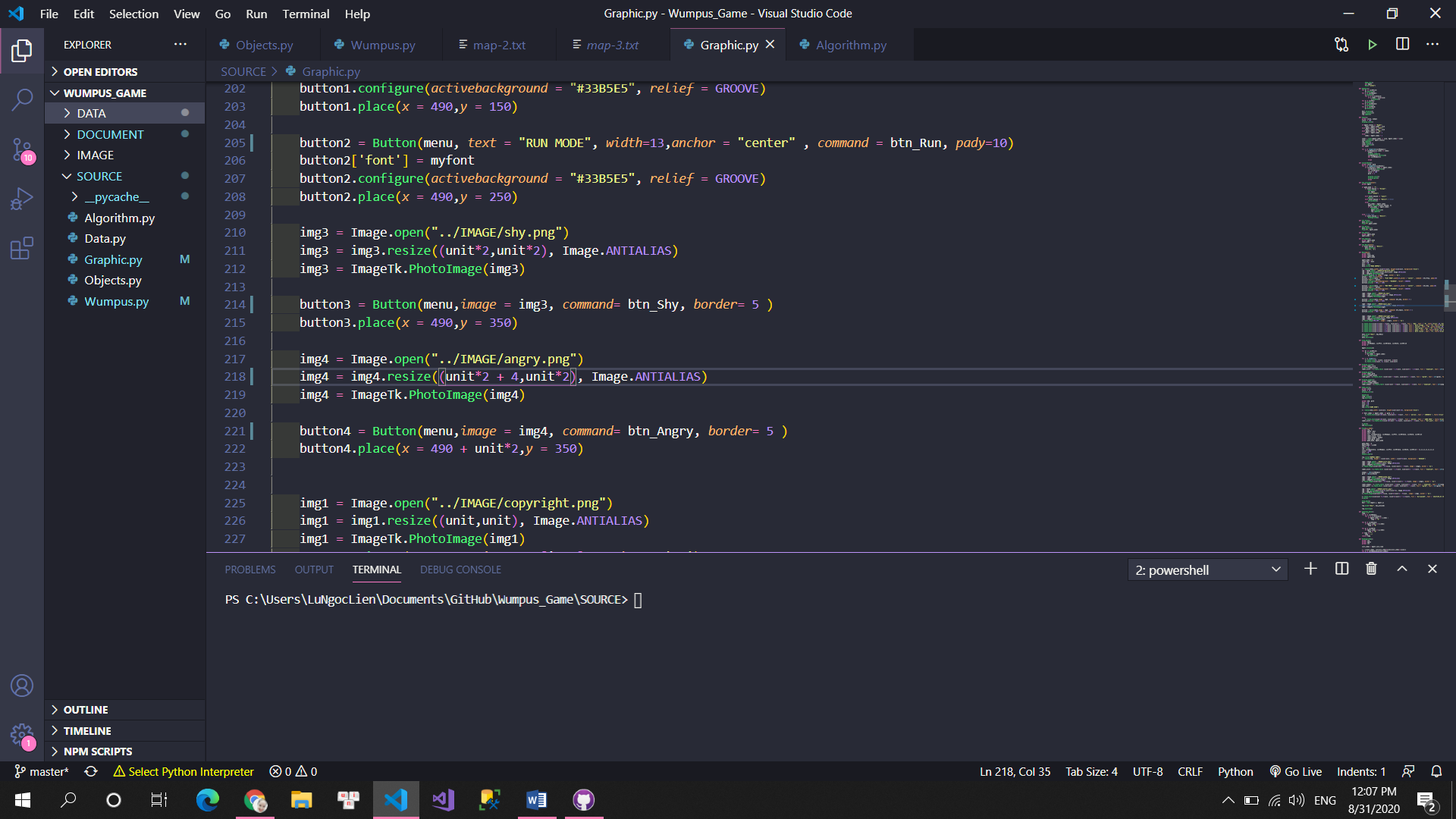
|  |  |  |  |
| --- | --- | --- | --- |
| Student ID | Student Name | Tasks | Percent |
| 18127046 | Lư Ngọc Liên | - Handle Input  -Display text, buttons, images  -Shy Agent  - Collect gold  - A\* search  - 5 static mazes | 100% |
| 18127155 | Vũ Công Minh | - Create class objects  -Aggressive Agent  - Shoot wumpus  -Handle clauses  - Image for objects  - Random maze | 100% |

***Self rating:***

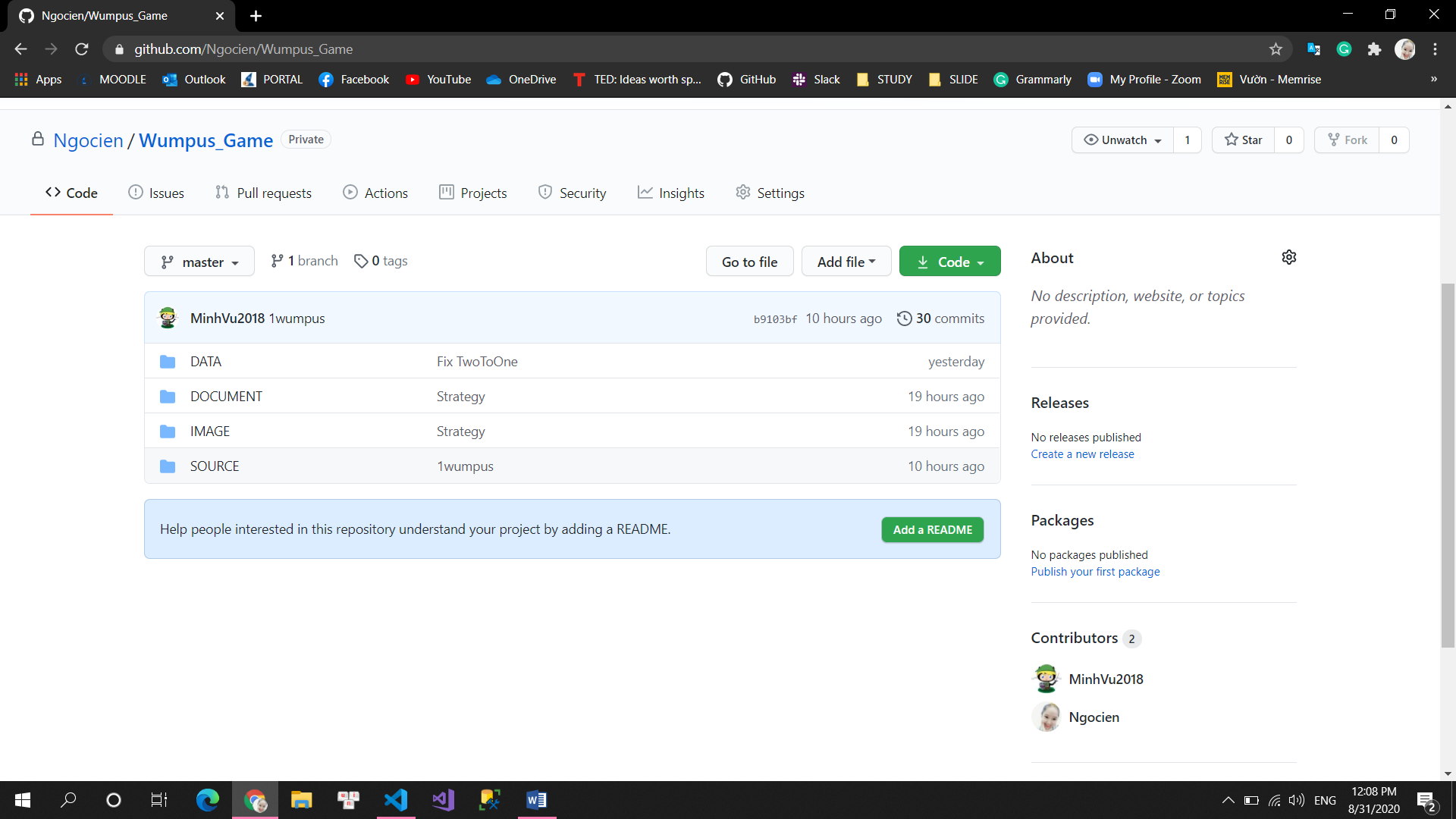
Project: 100% overall.

# Environment to compile and run the program

Environment to compile and run the program: Visual Code

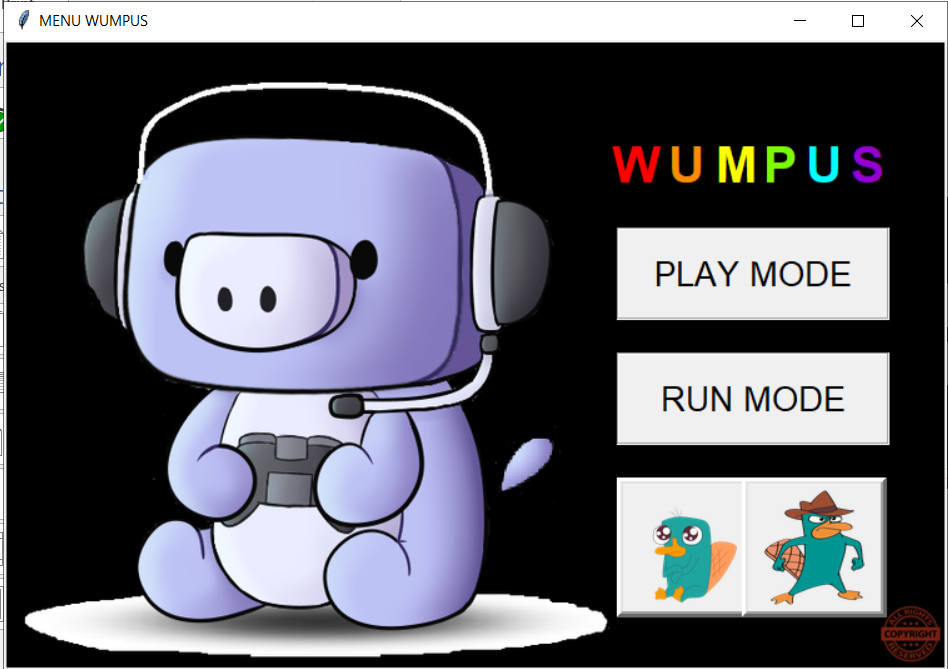


Version control: GitHub



# Estimating the degree of completion level for each requirement

1. Finish problem successfully with 2 modes and 2 strategies for auto-run mode: 100%
2. Graphical demonstration of each step of running process: 100%



1. Generate 5 maps input and implementation generate Wumpus World randomly with difference structures such as position and number of Pit, Gold, Wumpus: 100%
2. Report our algorithm and compare the result of each strategy: 100%

# Instruction

To play the game, you should follow steps below.

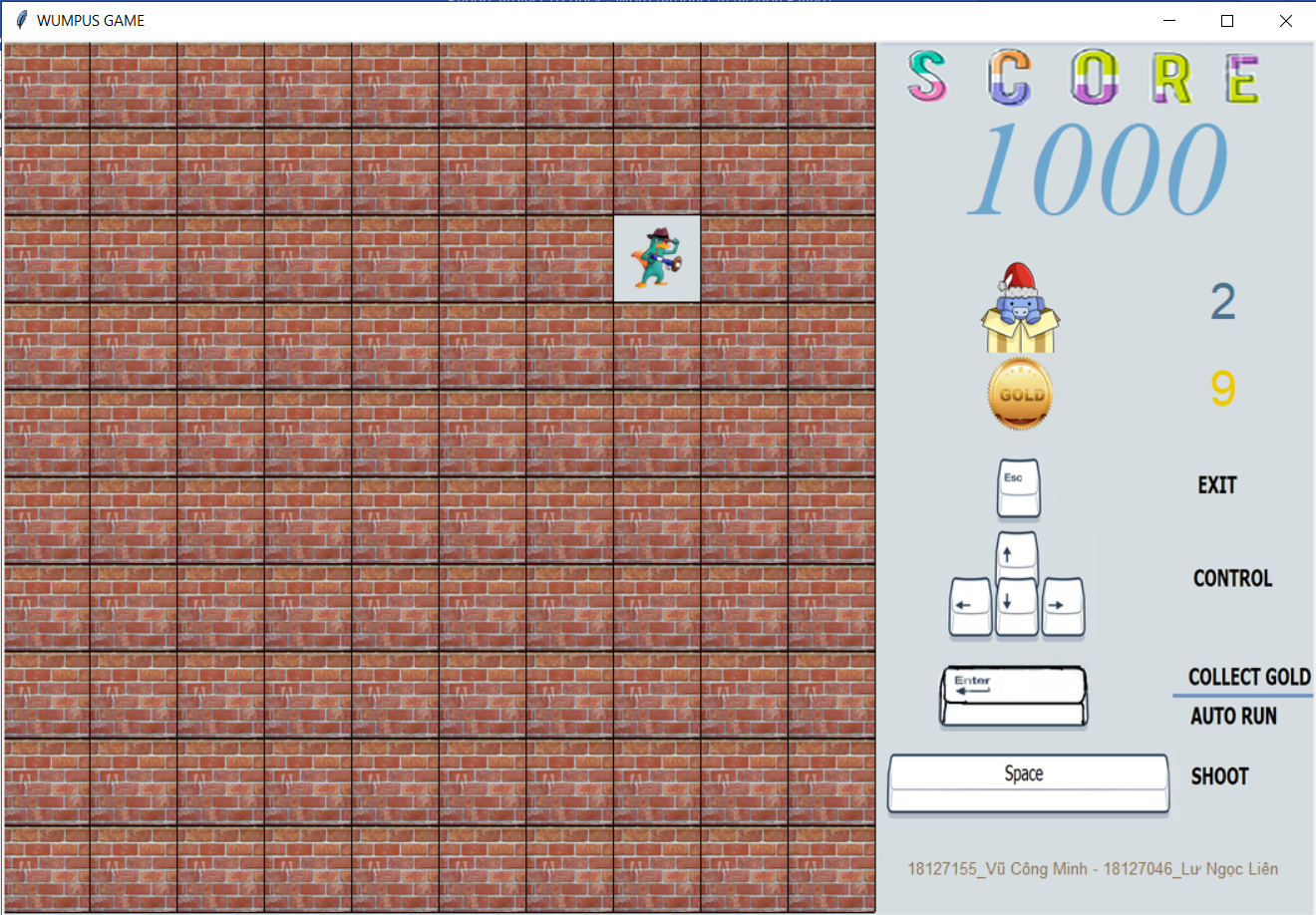
1. You have to choose what strategy mode (agent mode) you want to play by clicking one of 2 pictures below.

* The left one is we call shy agent. The agent will try to collect all golds and remain score is as high as possible.
* The right one is we call aggressive agent. The agent will kill wumpus as soon as they have a confident prediction.

1. After choosing strategy mode, you will choose the game mode by clicking one of 2 buttons. We have 2 for you.

* The up one is the play mode. That means you will control the agent, kill wumpus and collect gold by yourself.
* The down one is the run mode. That means the agent will move automatically, collect gold automatically and kill wumpus automatically.

Choose agent mode

1. Then you will see:

Blue color: Number of wumpus

Yellow color: number of gold

Initial score will be 1000

* Run mode: Press Enter to play
* Play mode:
* Press Enter to collect gold
* Esc to exit
* Space to shoot

1. Finish Game: Congrats or Game Over



# Algorithm explanation

* What we use for algorithm:
* Current node: tuple(index, state, True/False)
* If there is a pit or wumpus, it will be False
* Else it will be True
* List Predicted: stored nodes at the current node with and operator ()
* It will be update after current node add to List Visited
* List Visited: stored nodes visited
* Add current node to list visited
* KB: stored list visited predicted

|  |  |  |  |
| --- | --- | --- | --- |
| S |  | B | P |
| W | BS | P | B |
| S |  | B |  |
| A | B | P | B |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 4 | 8 | 12 |
| 1 | 5 | 9 | 13 |
| 2 | 6 | 10 | 14 |
| 3 | 7 | 11 | 15 |

* Step 1:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| A |  |  |  |

* List Predicted: [(3,’-‘,True), (2,’-‘,True), (7,’-‘,True)]
* Current node: (3,’-‘,True)

Update to Visited

* List Visited: [(3,’-‘,True)]

Update to Predicted

* List Predicted: [(2,’-‘,True), (7,’-‘,True)]
* KB: [(3,’-‘,True), (2,’-‘,True), (7,’-‘,True)]
* Step 2:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| A | B |  |  |

* List Predicted: [(2,’-‘,True),(7,’-‘,True), (3,’-‘,True), (6,’-‘,True), (11,’-‘ True)] (1)
* Current node: (7,’B’,True)

Update to Visited

* List Visited: [(3,’-‘,True), (7,’B’,True)] (2)

Update to Predicted based on the last visited node

From (1) and (2), with the index unvisited. We can see index 6 has different state. So we create a function to solve this with a, b is tuple has the same index

if (not a[2]) and (not b[2]) and (a[1] in b[1] or b[1] in a[1]):

if len(a[1]) < len(b[1]):

return a

else:

return b

return (a[0],'-',True)

* List Predicted: [(2,’-‘,True), (3,’-‘,True), (6,’P‘,False), (11,’P‘, False)]
* KB = [(2,’-‘,True), (7,’B’,True), (3,’-‘,True), (6,’P‘,False), (11,’P‘, False)]
* Step 3:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| S |  |  |  |
| A | B |  |  |

* List Predicted: [(2,’-‘,True), (3,’-‘,True), (6,’P‘,False), (11,’P‘, False), (1, ‘-‘, True)]
* Current node: (2,’S’, True)

Update to Visited

* List Visited: [(3,’-‘,True), (7,’B’,True), (2,’B’,True)]

Update to Predicted based on the last visited node

* List Predicted: [(3,’-‘,True), (6,’-,True), (11,’P‘,Fasle), (1, ‘W‘, False)]
* KB: [(3,’-‘,True), ), (7,’B’,True), (2,’B’,True) , (6,’-,True), (11,’P‘,Fasle), (1, ‘W‘, False)]
* Step 4:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| S |  |  |  |
| A | B |  |  |

* List Predicted: [(2,’-‘,True), (5,’-‘,True), (6,’-‘,True), (11,’P‘, False), (1, ‘-‘, True), (10,’-‘,True)]
* Current node: (6,’-’, True)

Update to Visited

* List Visited: [(3,’-‘,True), (7,’B’,True), (2,’B’,True), (6,’-’, True)]

Update to Predicted based on the last visited node

* List Predicted: [(2,’-‘,True), (5,’-‘,True),(11,’P‘, False), (1, ‘-‘, True), (10,’-‘,True )]
* KB: [(3,’-‘,True), (7,’B’,True), (2,’B’,True), (6,’-’, True),(5,’-‘,True),(11,’P‘, False), (1, ‘-‘, True), (10,’-‘,True )]
* Step 5

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | BS |  |  |
| S |  |  |  |
| A | B |  |  |

* List Predicted: [(2,’-‘,True), (5,’-‘,True),(11,’P‘, False), (1, ‘-‘, True), (10,’-‘,True ), (4,’-‘,True), (9,’-‘,True)]
* Current node: (5,’BS’, True)

Update to Visited

* List Visited: [(3,’-‘,True), (7,’B’,True), (2,’B’,True), (6,’-’, True), (5,’BS’, True)]

Update to Predicted based on the last visited node

* List Predicted: [(2,’-‘,True),(5,’B’,True), (11,’P‘, False), (1,‘W‘, False), (10,’-‘,True ), (4,’WP‘, False ), (9,’PW‘, False)]

#Chuối

* Kill
* Update visited stench around wumpus
* KB: [(2,’-‘,True), (5,’B’,True), (11,’P‘, False), (1, ‘W‘, False), (10,’-‘,True ), (4,’P‘, False ), (9,’P‘, False), (3,’-‘,True), (7,’B’,True), (6,’-’, True)]
* When we predicted all the next node at the current node is wumpus or pit. We will use A\* search to find the shortest path from current node to the node in List Visited to remain score as high as possible. (shy agent)
* When we predicted the next node at the current node is sure a wumpus (double check), kill wumpus and move to wumpus node (aggressive agent). Else use A\* search to find the shortest path from current node to the node in List Visited.
* When at the current node has gold, we collect the gold.
* When we shoot the wumpus, all stench at top, down, left, right of the wumpus will be deleted and update to the List Predicted by this function above.
* After update visited list and predicted list, the agent send a signal to graphic request its display.

# Reference

# Comparison between 2 strategy mode

|  |  |  |
| --- | --- | --- |
| Maze | Shy agent | Aggressive agent |
|  | 250 | -260 |
|  | 610 | -360 |
|  | 710 | -700 |
|  | 1000 | -200 |
|  | 1330 | 1200 |