Deceptive AI: Path-Planning using Strategies Inspired by Stage Magic

STUDENT: XIAOYUE MA

STUDENT ID: 878899

SUPERVIOR: Michael Kirley & Peta Masters

COMP90055 – Computing project (Semester 2, 2019)

Research project: 25 credit points

I certify that - this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text. - where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the School - the thesis is 5339 words in length (excluding text in images, table, bibliographies and appendices).

Abstract*—As a higher intelligent creature, humans think that they will not be easily deceived. Therefore, to test the degree of deception of different strategies for humans, this project uses turn-based game Mind Trail to explore. Each strategy is designed for the AI agent in Mind Trail, who play deceptively against a human opponent. As for the human opponent in Mind Trail, several naive human behaviours are applied to against the AI agent.

***Keywords*— Deceptive AI, human, Mind Trails**

I. Introduction

This project is based on the Mind Trail game in unity WebGL platform by C# language, and output logfile as txt file. In this report, we will illustrate the definition of deception and describe the concrete situations of the game. Based on the specific deception definition, some assumptions of this game are proposed. Then, after carrying out the relevant simulations and analysing the performance of the deceptive parameters, a discussion of the results will be given.

There are several definitions of deception. Firstly, According to Vrij's description, deception could be enjoyed as a deliberate attempt, potentially successful, without prior notice, to increase the income of the communicator [1]. The feature of this attempt is that although it is untrue for the communicator, the communicator believes that it is correct. However, Da Waal supports claims that deception articles as an incorrect or false image of knowledge, intention or motivation, which is beneficial to the communicator itself [2]. Moreover, there is a simpler definition of deceptive behaviour by Bond and Robinson, which defines deceptive behaviour as a false communication that tends to help communicators[3]. In this report, the straightforward definition of deceptive behaviour for AI carry out in this project. This definition refers to the capacity for some software to act as the medium, where some artificial intelligent agents would be manipulated to believe, resulting in spreading or accepting false information [4].

Mind Trails is a turn-based game for two players. The game environment is created by Ruilin Liu (shown in Fig. 1) [5]. The game contains a game board, within some cells are located as anchors, and four token-generator located at nodes outside the grid. Player A controls could control one or more shuttles (only control one in this project), which are used to collect tokens (red or non-red) with a limited number (one each turn in this project). Shuttles can also be used to move tokens from grid cell to grid cell or to swap a token that's in the shuttle for one already on the grid. The goal of Player A is to design and build a winning path. At Player A's turn, it moves the shuttle to the random token generator, collects one or more coloured tokens, and deposits them in the grid. Player B is aiming to design and prevent Player A from winning. At B's turn, it blocks an unoccupied tile in the game board, resulting in effectively making it inaccessible and removing it from the board.

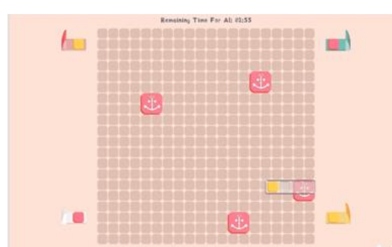

Fig. 1. Mind Trail game scence

The basic rule of this game is that when the tokens are deposited on the board by Player A, they will turn into white, so player B cannot see (can only remember) what colour they are. Nonetheless, in order to show the process clearly, although in game process, all the tokens will turn over into white colour, the original colour of some tokens will be shown in this report. Moreover, in most cases, Player A creates a real path of red tokens but tries to deceive Player B into blocking an alternative fake path. If Player B blocks a tile on Player A's real path from one anchor to another, Player A will reschedule and find an alternate path between the same two anchors.

II. Design Definition

There are some design explanations in this part, such as Path, narrative region and deposit methods.

A. Path

The path $\pi = n_0, n_1, n_2, \ldots, n_k$ is a contiguous sequence of the nodes, the tile units or anchors, so each consecutive pair (n_i, n_{i+1}) is connected in 4 ways, in horizontally or vertically (not diagonally). The concrete detail of the path is shown in Fig. 2.

The winning path is such a path that n_0, n_k $(n_0 \neq n_k)$ are aiming anchors' positions, and red tokens will locate in each tile n_i for $0 < i < k$, that is, a red four-way continuous sequence from one anchor n_0 to another n_k. The cost of the path is calculated as the total number of connection tiles between aiming anchors in that path. An optimal path is the path shares the lowest cost. As Fig. 3 shows below, the optimal path cost is cost(π)=2, calculating by the minimum number of connection tiles between aiming anchor.

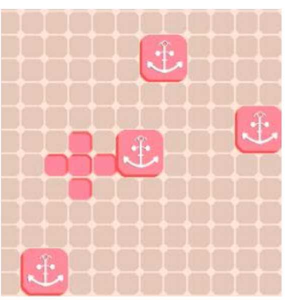

Fig. 2. Show the path definition

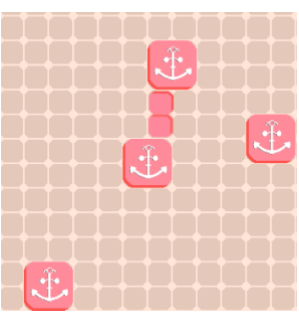

Fig. 3. Show the cost

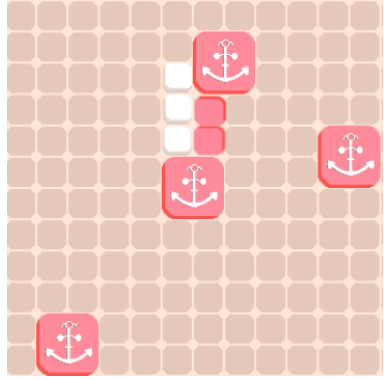

Fig. 4. Show the fake path and true path

What is more, there are two main types of paths in this game, the fake path and true path. As shown in Fig. 4, the white tokens make up the fake path and the red tokens constitute the real path.

B. Narrative region

In figure x, it represents the definition of the narrative region between anchors. It is observed that in the case where there are any two anchor positions in the game grid, every tile in the rectangular block is located at the diagonally opposite corners. In this project, we call this rectangular block as the narrative region for the two anchors, where Player A could find many optimal/true paths in this area. Such as, in Fig. 5, the lower right corner is seen as the location of the anchor.

We could give the assumption that A1= (0,5), A2= (7,7) and A3= (2,0). The narrative region (A1, A2) includes all cells from (0,5) to (7,7), that is, from $(min_{col}(A1, A2), min_{row}(A1, A2))$ to $(max_{col}(A1, A2), max_{row}(A1, A2))$. The narrative region (A2, A3) includes all cells from (7,7) to (2,0), that is, from $(min_{col}(A2, A3), min_{row}(A2, A3))$ to $(max_{col}(A2, A3), max_{row}(A2, A3))$. As we could see from figure x the red circle and blue circle are the narrative regions for (A1, A2) and (A2, A3).

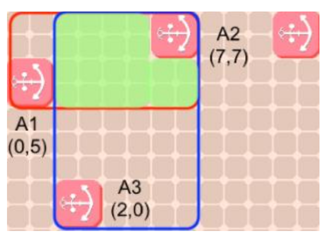

Fig. 5. Show the narrative region and overlap situation

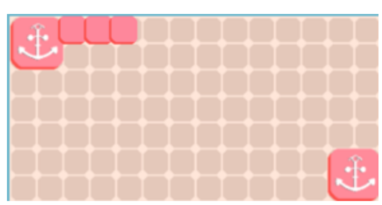

Fig. 6. Show the sequentially way

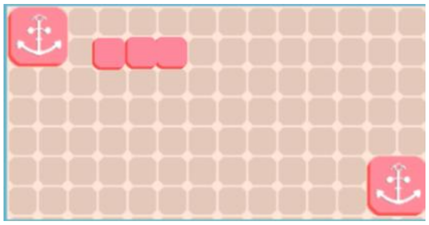

Fig. 7. Show the contiguously way

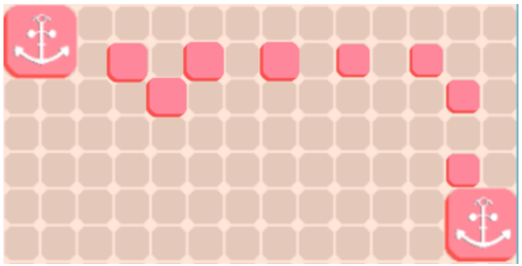

Fig. 8. Show the non-contiguously

In the narrative region definition that needs attention is that although the lower corner is regarded as anchor's position, the scope boundary of the anchor needs to be considered. For instance, A2= (7,7), when calculating the narrative region, it is necessary to consider the upper left corner in A2 (8,8). Thus, the narrative region between A1and A2 is from (0,5) to (8,8).

What is more, the green area in Fig. 5 is to describe the overlapped region between the narrative region (A2, A3) and the narrative region (A1, A2), which are applied in simulation circumstance two.

C. Deposit approaches

Next, through observation, the main responsibility of A is to decide for red tokens and non-red tokens. There will illustrate three methods of deposit the reds: sequentially, contiguously, and non-contiguously approaches. On the other hand, for non-reds, there are four types of locations to place them, elsewhere, target anchor's narrative region, non-target anchor's narrative region, and overlapped narrative region between multiple anchors.

1) Red token: Given a path $\pi = n_0, n_1, n_2, \ldots, n_k$,where n_0 and n_k are anchors.

a) Sequentially : The tokens will deposit sequentially on π, placed in order starting from one of the anchors. That is to say, each token is placed at a position n_i, such that $i = j + 1$ and the token previously stored on the path was located at location n_j. If there is no such position, for instance, this is the first token to be deposited on the path, the relevant token will be released at position n_1 (Fig. 6).

b) Contiguously : Under the contiguous circumstance, tokens will deposit contiguously on π without gaps. That means each token is placed at the position n_i, such that either position n_{i-1} is occupied or position n_{i+1} is occupied. If no such location exists, like this is the first token to release on the path, a random value i ($1 < i < k-1$) will be selected (Fig. 7).

c) Non-contiguously : As for the non-contiguous situation, tokens will place discontinuously on π leave as much gap as possible for as long as possible. That is, each token is stored in a random position n_i , where n_i is a node on π such that n_{i-1} and n_{i+1} are both empty. If there is no such location, for example, at least half the nodes on the path are already occupied, the token is randomly placed in any unoccupied location on the path (Fig. 8).

2) Non-red token

a) Elsewhere : Tokens will be located at a random unoccupied location on the board (Fig. 9).

b) Target anchor's narrative region : In Fig. 10, it shows the situation that non-red tokens will be randomly deposited at the target anchor's narrative region (the red circled area).

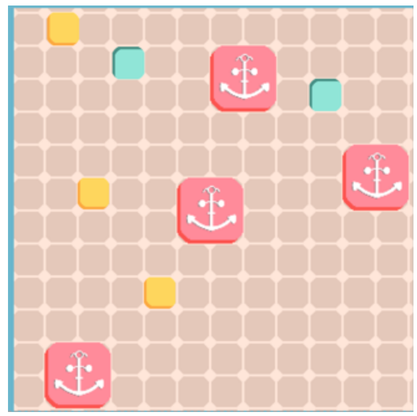

Fig. 9. Show the random position

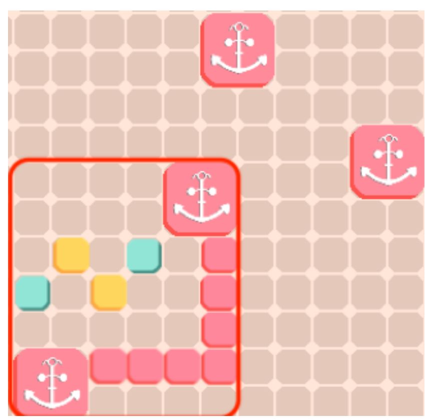

Fig. 10. Show target anchor's narrative region

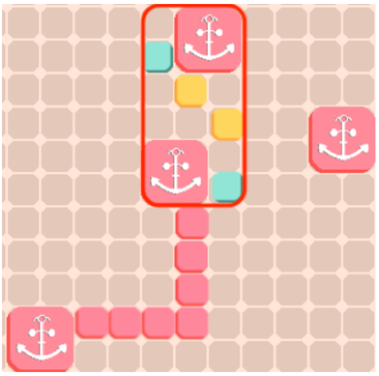

Fig. 11. Show the non-target anchor's narrative region

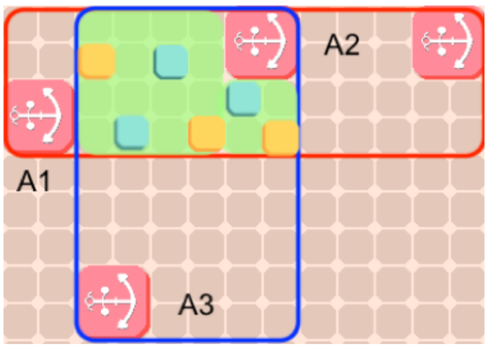

Fig. 12. Show the overlapped narrative region

c) Non-target anchor's narrative region: In Fig. 11, non-reds will be randomly placed at the area in the red circle, because this area is non-target anchor's narrative region. The two anchors connected by red tokens are targets.

d) Overlapped narrative region between multiple anchors: In this situation, tokens will be stored at the overlapped region (the green area in Fig. 12) randomly.

D. Achievement

Due to need simulation to collect the relevant feedback from different deceptive strategies, several changes are made in the original game version.

1) Autonomous Player B agents: First of all, in the previous game version, Player B is a real human player. Thus, when it comes to Player B's turn, the game need manually operate by the human player. After clicking the relevant position in the game board, the position will turn grey. However, it is not suitable to simulate automated experiments. Therefore, to process deception experiments though the game, B is developed as an automatic proxy, which imitates three naive human behaviours. That is, instead of pausing manual input, allow the response to proceed automatically. Furthermore, the game state is passed to Player B, such as the deposited tokens' positions and their colours. Then, Player B returns to the suggested blocking position and adds the block to

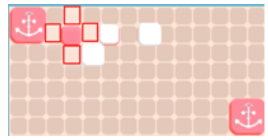

Fig. 13. Block positions adjacent to the most recent red token

the game state, which is then passed to Player A. The concrete details of B's types are shown as follows (one block per turn).

a) Player B will naively randomly block at the tiles adjacent to the most recent red token (the area of the red square in Fig. 13). If no such position exists, for example, at A's turn, A deposited a non-red token, a random position on the board will be blocked

b) Player B will naively block at the tile near the anchor closest to the most recent red token location (the area of the red square in Fig. 14). Meanwhile, to improve Player B's performance, B will choose the position where is included in the anchor's narrative region, composed by the anchor closest to the aiming anchor. For example, A3 and A1 are Player A's target anchors. At B's turn, the anchor which closest to the most recent red is A1, and A2 is the nearest anchor from A1. Therefore, the positions near A1 which is included in the narrative region between A1 and A2 will be blocked. If no such position exists, for example, at A's turn, A deposited a non-red token or all the tiles in target narrative region are occupied, a random position on the board will be blocked.

c) Player B will naively prevent a position near densest concentration of red tokens (the area of the red square in Fig. 15). However, the precondition for this method is more than three red tokens are deposited on the board. If less than three reds on the grid or no available positions, a random position on the board will be blocked.

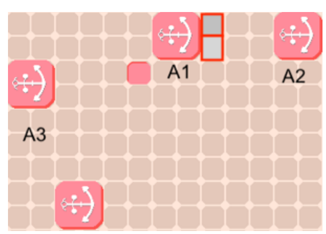

Fig. 14. Block positions adjacent to anchor closest to most recent red token placement

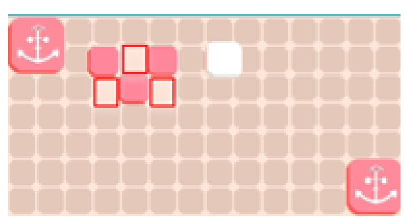

Fig. 15. Block positions near densest concentration of red tokens

Fig. 16. Show the change type way in option menu

2) Player A types: Similarly, eight types of Player A are designed to test different deceptive approaches on two dimensions, the random anchors situation (Table I) and overlapped narrative region position in 10 tiles (Table II). The check mark indicates how the corresponding Player A determines the red tokens and non-red.

The tester could change Player A and B types from option menu. (show in Fig. 16)

Furthermore, in the original version, Player B can be the winner only under the time out circumstance. Thus, another win option has been added in the game, if the path between the selected anchor pair is completely blocked, the game will announcement Player B is the winner.

TABLE I. THE RANDOM ANCHORS SITUATION

	Red sequential	Red contiguous	Red Non-contiguous	Non-red elsewhere	Non-red narrative region	Non-Red fake path	probability 0.5
A1	✓			✓			
A2		✓		✓			
A3			✓	✓			
A4			✓			✓	
A5			✓		✓		
A6			✓		✓		✓

TABLE II. OVERLAPPED NARRATIVE REGION POSITION IN 10 TILES

	Red Non-contiguously	Non-red overlap Narrative region	Non-red Narrative region	Probability 0.5
A7	✓		✓	
A8	✓	✓		✓

3) Shuttle start time: There is an option to fill in the departure time of the shuttle- shuttle start time in Fig. 17. That is, if multiple shuttles are used, this option could control each shuttle start moving timestamp. Such as, each shuttle can be set to start moving at the same time as the first shuttle, or to start

Fig. 17. Show the shuttle time enter

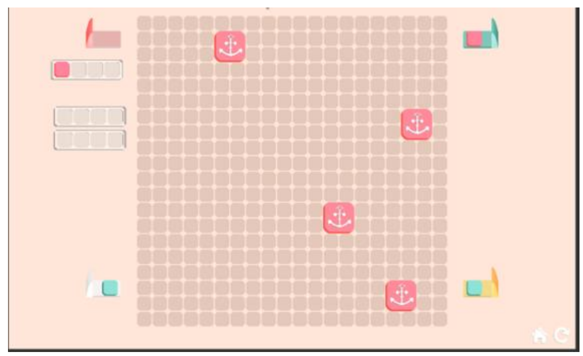

Fig. 18. Show the shuttle start time

moving a few seconds later than the first shuttle. It is easier for this function to provide follow-up testing in the departure time of the shuttle. As Fig. 17 shown, 3 shuttles and 2 second are entered; thus, in Fig. 18, after the first shuttle begins to move for two seconds, the remaining two shuttles will move.

4) Parameters and calculation: For games, it is important to build the relevant environments to test and run the simulations, and more importantly, analyse the results to continuously improve the performance of both side of the game. As for this deceptive game, the main parameters of the analysis came from the three ways and were expanded into serval dimensions. At first, deceptively will be measured in three ways: if Player A wins, how many red tokens are successfully added to the real path before A is blocked by Player B for the first time; and the ratio between blocks on the real path and blocks on the false path, which is recorded at the end of the game. However, as mentioned above, by changing the rules of the game, in addition to the timeout, Player B could be a winner under other situations. Thus, the original measurements are no longer suitable. To facilitate this, extend the original parameters to the arguments below. To describe and show the block position clearly, the block color will be changed from grey to green and some figures are only references.

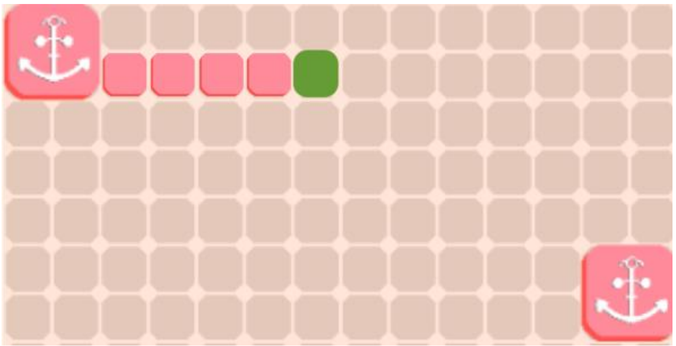

Fig. 19. Show a block on real path

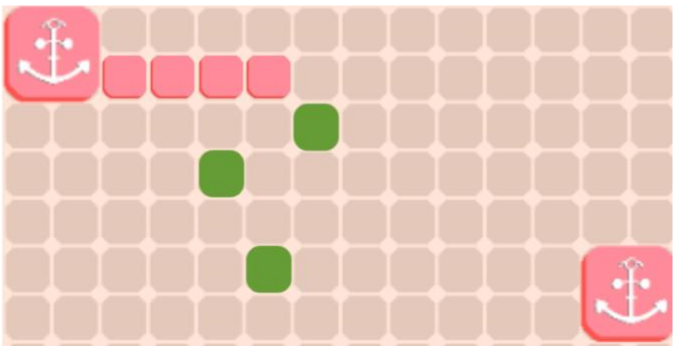

Fig. 20. Blocks on the narrative region

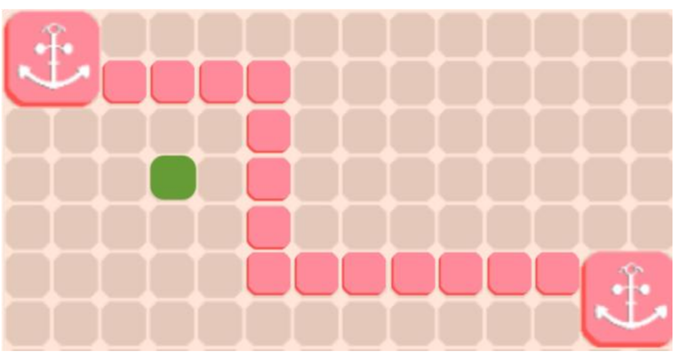

Fig. 21. Point-to-polyline distance

- Total_RealPath_Blocks: At the end of the game, the total number of blocks on real path deposited by Player B. In Fig. 19, the green tile is to show a block on real path.
- Total_FalsePath_Blocks: The total number of blocks made on a false path at the game over by Player B. However, for some test situations, there is no default false path on the board, thus, this parameter seems useless if no false path is explicitly planned.

```
Shuttle 0 collects in Generator 1 color: 0
Shuttle 0 moves to (6, 2)
Shuttle 0 deposits at (6, 2), index: 0
Shuttle 0 moves to (-3.5, 10)
Player blocks (6.0, 1.0, 0.0)
The distance between real path and block is :1
The distance between fake path and block is :15
Shuttle 0 moves to (22.7, 3.5)
Shuttle 0 collects in Generator 3 color: 0
Shuttle 0 moves to (5, 2)
Shuttle 0 deposits at (5, 2), index: 0
Shuttle 0 moves to (-3.5, 10)
Player blocks (4.0, 2.0, 0.0)
The block is on the real path
Shuttle 0 moves to (22.7, 3.5)
Shuttle 0 collects in Generator 3 color: 0
Shuttle 0 moves to (5, 1)
Shuttle 0 deposits at (5, 1), index: 0
--- AI Win ---
AI Turn Count: 10
Remaining Time For AI: 112.6191 seconds
Deposit in total real path is 3
Deposit in total real narrative is 9
Deposit in total blocks is 10
Deposit in total false blocks is 0
The AI replan times is :3
```

Fig. 22. Show the output

- Total_NarrativeRegion_Blocks: The total number of blocks deposited on the aiming anchor's narrative region. In Fig. 20, the green position is to show blocks on the narrative region.
- Replan_Time: The number of times replanned by Player A at the game over.
- Total_Blocks: The total number of blocks made at all locations at the end of the game.

Also, for further testing, the distance calculation of the block location, at each Player's turn, between the true path and false path is performed. Due to in some cases, the path is not a straight line, but an irregular polyline. To simplify the calculation, the polyline is considered to consist of different straight-line segments. Therefore, the point-to-polyline distance is the minimum distance from the point to the different straight-line segments. For instance, in Fig. 21, the distance between the path and the block position is 1.

The output of the part at the end of the game is shown below(Fig. 22): As shown in Fig. 22, after each game is over, the parameters mentioned above will be recorded in the txt file; and at each Player B's turn, the distance between blocking position and the real/ fake path will be calculated and recorded.

III. ASSUMPTION

According to the definition of the deception given in the Introduction part and common sense, several hypotheses are proposed.

By analysing the design of Player A in Table I and Table II, as the number increases, the complexity of the game increases, thus, the degree of confusion about Player B is likely to increase. What is more, there are 8 types of Player A in this project, according to this hypothesis, no matter which Player B is applied in the game, the eighth Player A could acquire the highest winning rate.

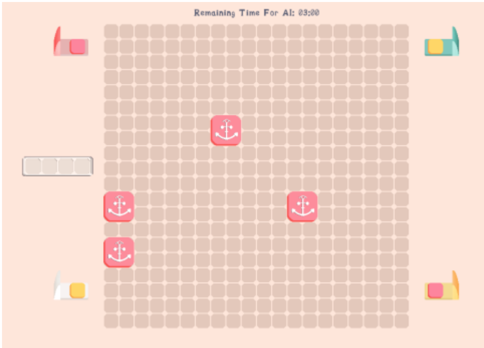

Fig. 23. The initial situation in Task 1

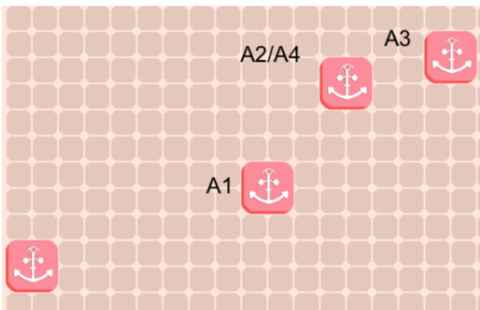

Fig. 24. The target anchors

Similarly, among the three types of B, there may be one type that can achieve the highest winning ratio, no matter which Player A is applied in the game.

Player A is more likely to win the game.

IV. Simulation and Results Analysis

This project is mainly focus on two dimensions, the random anchor position and overlap narrative region (in 10 tiles) with 3 types player B and 8 types player A. Each experiment of Player A and B was tested in 10 times.

In those tasks, the game environment should be initialised with grid size (20*20), number of shuttles (1 shuttle), number of tokens colours (yellow and red), number of anchors (4 anchors) all at minimum values (show in Fig. 23).

A. Task 1 random anchor position

In this task, there are 6 types of Player A (Table I). This task aims to use path-planning to build a connection of red tokens between two anchors, testing how easy it is for Player B to win.

The target anchors can be selected as follows. Randomly choose A1, select A2 to make it one of the rest anchors closest to A1. A3 is randomly selected, where A3 ≠A1 and A3 ≠A2. Select A4 to make it one of the three anchors nearest to A3, which may be the same as A1 or A2. Path a is the shortest path between A1 and A2, and path b is the shortest path between A3 and A4 (Fig. 24). The relevant target are randomly created.

In Test 1 (shown in Fig. 25), Player A1 in Table I will be applied. That is, at Player A's turn, starting with the anchor,

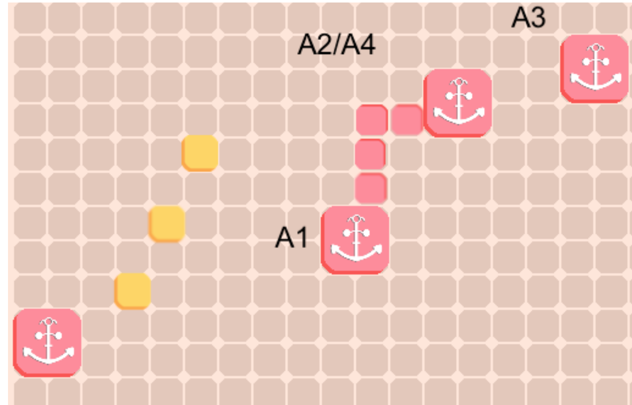

Fig. 25. Show Test 1 situation player A

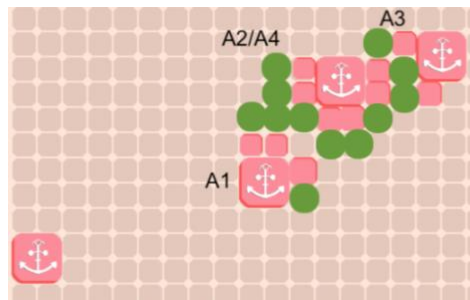

Fig. 26. Show Test 1 Player B as the winner

place the red tokens in order on the real path. Meanwhile, checking availability of path a at each A's turn. If there are no new path a, the real path will change from path a to path b. If any node on path a has been blocked, A should generate a new path a. If the shuttle collects non-red tokens, distribute them at any other random grid location. If no available path a or b, Player B will be the winner (shown in Fig. 26). As seen from figure, all the path between (A1, A2) and (A3, A4) are blocked, so Player B will win the game.

- Result (Table III and Table IV):

TABLE III. B WINS RATIO IN TEST 1

Player types	Win ratio
B1	0.5
B2	0.1
B3	0.3

TABLE IV. RELEVANT PARAMETER UNDER A WINS IN TEST 1

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A1	0.7	8	1.952380952	5.238095238	1.19047619	1.952380952

In Test 2, Player A2 in Table I will be applied (show in Fig. 27). That is, Player A will place the reds contiguously on the real path, such as not from the anchor. The rest of the operations are the same as Test 1.

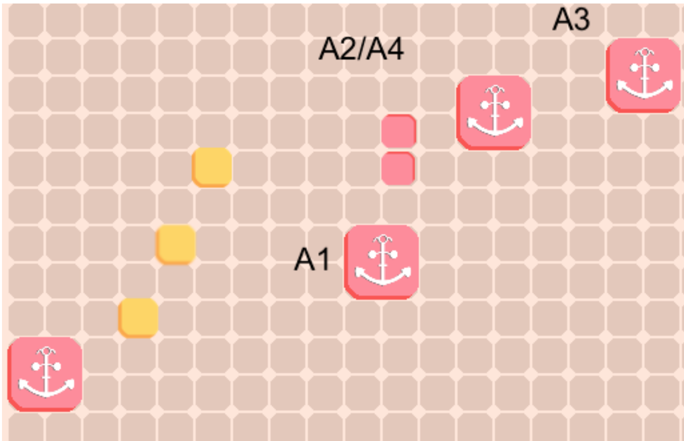

Fig. 27. Show Test 2 Player A

- Result (Table V and Table VI):

TABLE V. B WINS RATIO IN TEST 2

Player B types	Win ratio
B1	0.2
B2	0.1
B3	0.5

TABLE VI. RELEVANT PARAMETER UNDER A WINS IN TEST 2

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A2	0.73333333	9.454545455	1.5	5.5	1.909090909	1.5

In Test 3, Player A3 in table 1 will be applied (shown in Fig. 28). That is, Player A will release the red tokens non-contiguously on path a. The rest of the operations are similar to Test 1.

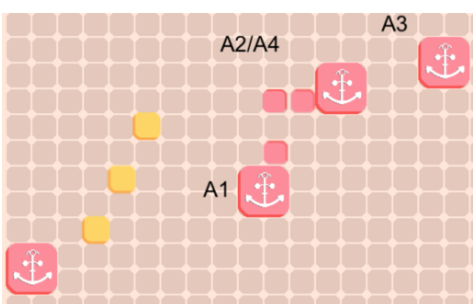

Fig. 28. Show Test 3 Player A

- Result (Table VIL and Table VIII):

TABLE VII. B WINS RATIO IN TEST 3

Player B types	Win ratio
B1	0.5
B2	0.2
B3	0.6

TABLE VIII. RELEVANT PARAMETER UNDER A WINS IN TEST 3

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A3	0.56666667	9.941176471	2.764705882	6.235294118	1.882352941	2.764705882

In Test 4, Player A is similar to Test 3 in red token deposition but distributes non-red tokens non-contiguously on path b (show in Fig. 29).

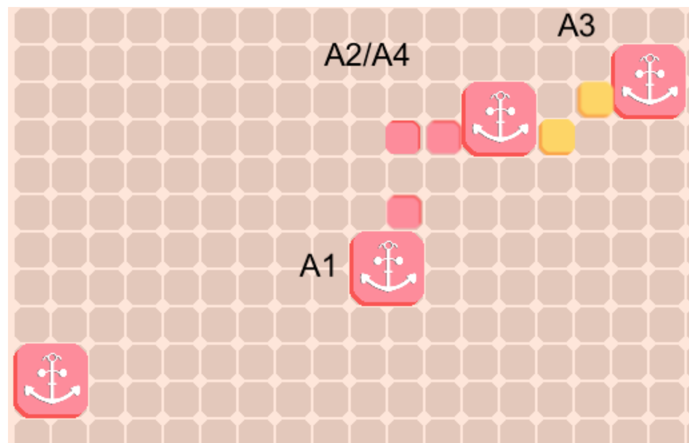

Fig. 29. Show Test 4 Player A

- Result (Table IX and Table X):

TABLE IX. B WINS RATIO IN TEST 4

Player B types	Win ratio
B1	0.6
B2	0.6
B3	0.5

TABLE X. RELEVANT PARAMETER UNDER A WINS IN TEST 4

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A4	0.43333333	7.307692308	1.923076923	4.692307692	0.769230769	1.923076923

In Test 5, Player A is the same as Test 4 (show in Fig. 30) but distributes non-reds at a random location within the narrative region for anchors (A3, A4).

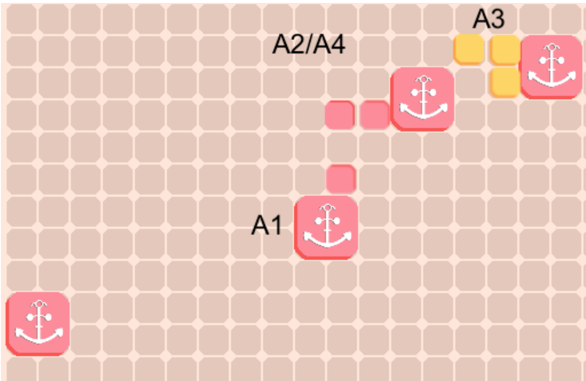

Fig. 30. Show Test 5 Player A

- Results (Table XL and Table XIL):

TABLE XI. B WINS RATIO IN TEST 5

Player B types	Win ratio
B1	0.6
B2	0.5
B3	0.3

TABLE XII. RELEVANT PARAMETER UNDER A WINS IN TEST 5

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A5	0.53333333	6.4375	1.125	3.875	1.25	1.125

In Test 6, Player A6 in table 1 will make decision in the game (show in Fig. 31). The probability in table 1 is to increase the difficulty of the game, further confuse Player B. That is, at each Player A's turn, the game will flip a coin. If on even, A will deposit the red token non-contiguously on the relevant path; if on odd, the red token will be placed at any random grid location.

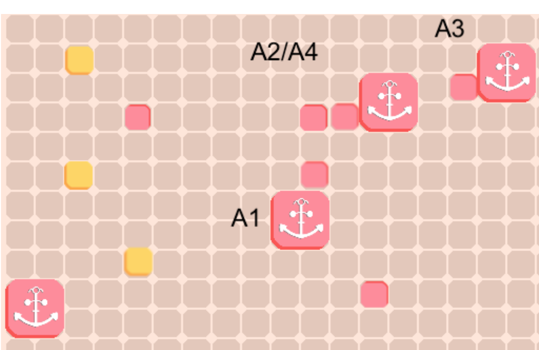

Fig. 31. Show Test 6 Player A

- Results (Table XIII and Table XIV):

TABLE XIII. B WINS RATIO IN TEST 6

Player B types	Win ratio
B1	0.7
B2	0.4
B3	0.9

TABLE XIV. RELEVANT PARAMETER UNDER A WINS IN TEST 6

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A6	0.33333333	8.7	2.3	4.6	4.7	2.3

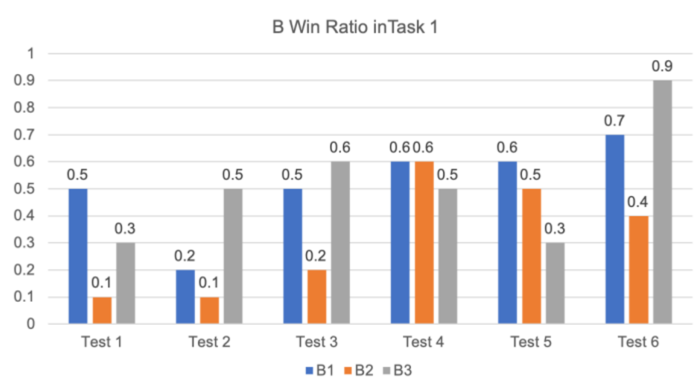

Fig. 32. Show the B win ratio in Task 1

In the Fig. 32, for all the tests in Task 1, Player B1 maintains a stable winning rate, about 60%. As for B2, it scores the highest winning percentage, 60% in the fourth Test, and achieves the similar winning ratios in Test 5 and Test 6 (50% and 40% respectively); and the rest of the test is maintaining around 15%. For B3, the prominent case is that its winning percentage reached 90% in Test 6, and its win rate is higher than 50% expect for Test 5 (30%) and Test 1 (30%).

B. Task 2 overlap narrative region (in 10 tiles)

This task creates a deceptive environment with ambiguity; thus, Player B could not determine which pair of anchors are the target pair, connected by red tokens. Base on the definition of the narrative region from Introduction part, this task will take advantage of the fact that, where there are multi-anchors, many narrative regions will overlap, testing whether the extent of regional overlap will increase deceptively.

The four anchors A1, A2, A3, A4 are selected such that each narrative region for each possible aligns overlaps with the rest by>=n tiles where n= 10. The shortest path between A1 and A2 is path a. As shown in Fig. 33, the narrative region (A1, A2) is the overlaps area, and coincidence degree is more than 10 tiles. The relevant anchors created in the tests should satisfy the premise that the coincidence is greater than 10 tiles.

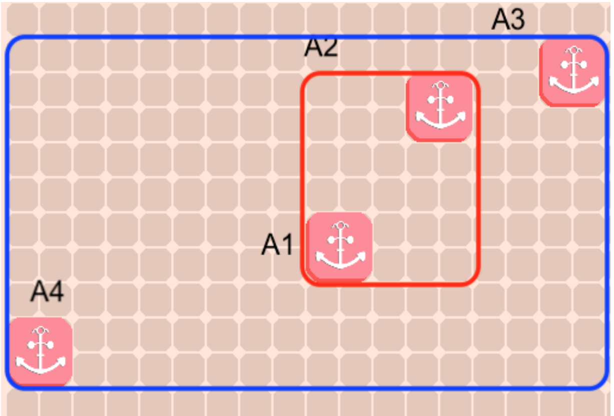

Fig. 33. Show Test 7 and Test 8 situation

Test 7 is trying to mask the path with occasional ambiguity. For example, anchors have been chosen to overlap their narrative areas, so it is inevitable to build and design the real path. At Player A's turn, the red token will be placed discontinuously on path a. The non-reds will be distributed within the narrative region for anchors (A1, A2), except for the elements on path a (Fig. 34).

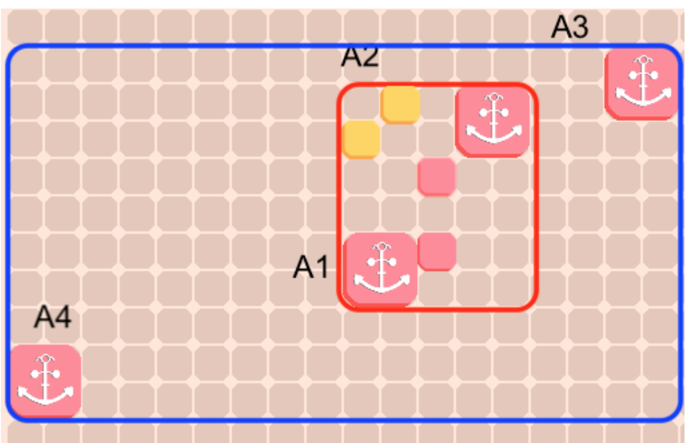

Fig. 34. Show A in Test 7

- Result (Table XV and Table XVL):

TABLE XV. B WINS RATIO IN TEST 7

Player B types	Win ratio
B1	0.7
B2	0.5
B3	0.3

TABLE XVI. RELEVANT PARAMETER UNDER A WINS IN TEST 7

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A7	0.5	14.33333333	3.4	8.533333333	2.266666667	3.4

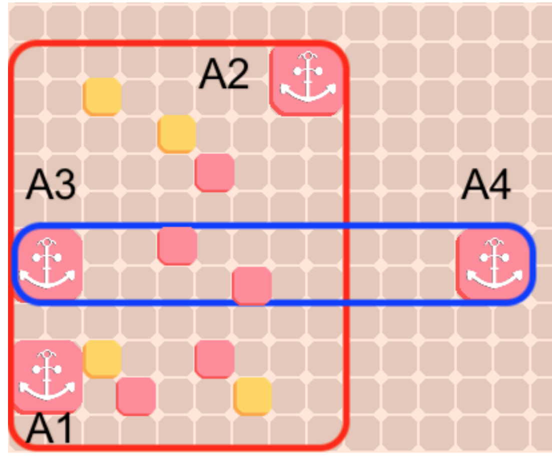

Fig. 35. Show A in Test 8

Test 8 attempts to cover the road in a deliberate method, adding extra reds to the overlapping region. In this test, the probability in Table II is to increase the deposition types for Player A, simultaneously confused Player B at the same time. That is, at each Player A's turn, the game will flip a coin. If on even, the red token will store non-contiguously on the specific path; if on odd, the red token will distribute at random location within the overlapped narrative region (show in Fig. 35).

- Result (Table XVIL and Table XVIIL)

TABLE XVII. B WINS RATIO IN TEST 8

Player B types	Win ratio
B1	0.9
B2	0.9
B3	0.8

TABLE XVIII. RELEVANT PARAMETER UNDER A WINS IN TEST 8

	RATIO	AVER TOTAL BLOCK	AVER BLOCKS IN REAL	AVER BLOCKS IN NARRATIVE REGION	AVER BLOCK IN FAKE	AVER REPLAN
A7	0.5	14.33333333	3.4	8.533333333	2.266666667	3.4

In Fig. 36, except for Player B3 in Test 7, the rest are generally achieving a pretty high winning percentage in Task 2, from 50% to 90%. As for Test 8 cases, it is not easy for Player A to win the game.

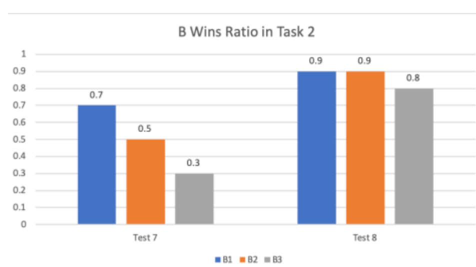

Fig. 36. Show B winning ratio in Task 2

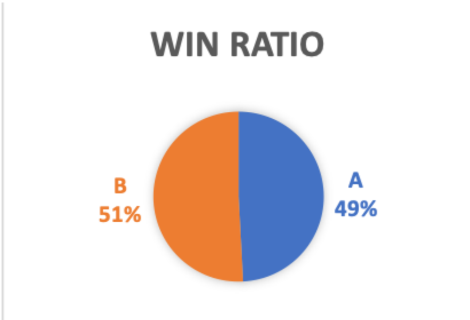

Fig. 37. Show Player winning ratio

V. DISCUSSION

This section is aiming to make the comparison of the simulation results, from the whole to the part, from the horizontal to the vertical, such as the total win ratio comparison and the percentages of each player types in the case where the player is the winner.

According to simulations' results (Fig. 37), it is found that Player A and Player B have similar equal probabilities to be the winner (49% and 51%, respectively), which is contradicted to the hypothesis 3.

Overall, from pie chart x, in all cases where Player A wins, A2 wins more times. That is, no matter which kind of Player B is applied in the game, A2 is a better choice for Player A to win the game.

As the bar chart Fig. 38 shows below, although it is assumed that with the increasing number of Player A, the more difficult of the game, the actual situation does not match the assumptions.

By observing the specific game process, it is found that although the game complexity increases with the number of Player A, one deposited token per turn limits the performance of Player A. Thus, the designed confusing behaviour of Player A sometimes could not play the role to confuse Player B; even in some test cases, it will achieve a waste of time substitution, resulting in the number increase, and the winning rate could not rise. Furthermore, because of the time consumption, for the higher numbered Player A, timeout is the most common reason for b to win the game.

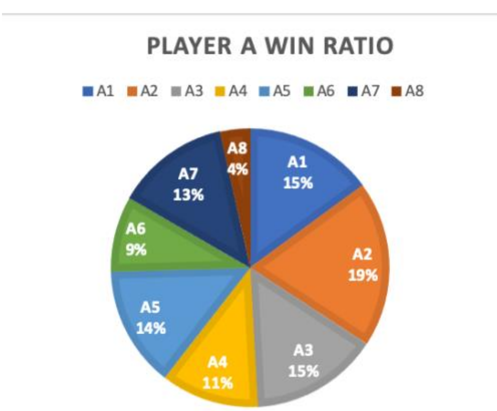

Fig. 38. Show Player A winning ratio in all cases where Player A wins

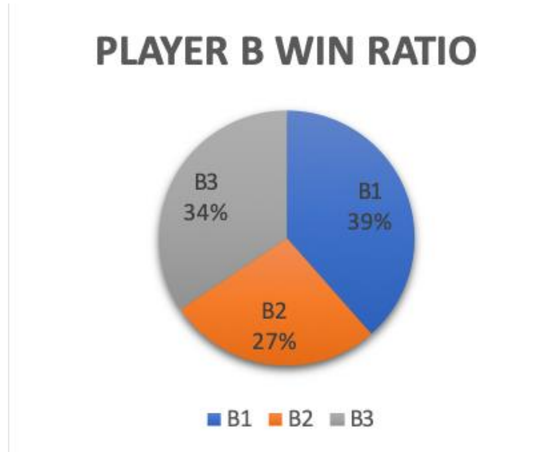

Fig. 39. Show Player B winning ratio in all cases where Player B wins

From overall view of pie chart Fig. 39, in all cases where Player B wins, the win ratio of each type is similar, and B1 shares slightly higher percentage. However, by observing the actual gameplay, each type has its own limitation to improve.

Although B1 seems to be more likely to win the game, if the red token is randomly placed or the aiming anchors are too close, it is hard for B1 to block on the true path.

What is more, since the distance of the randomly occurring target anchors is not fixed, sometimes B2, blocking the tile near the anchor, seems to be not effective. In addition to unfixed distance, B2 may make the wrong decision, resulting in an incorrect decision of the target anchor pair, as shown in Fig. 40. The aiming anchor pair is (A1, A3), but A2 is closer to A1, leading (A1, A2) is mistaken as the target. However, in addition to some cons, B2 has some significant pros. For example, enclosing the target anchor with blocks can effectively prevent the construction of the real path, so Player B could win the game.

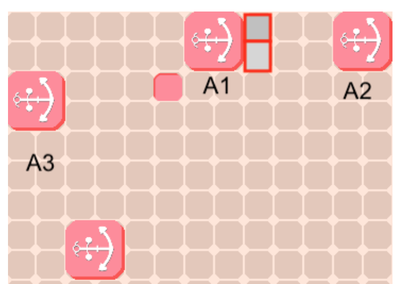

Fig. 40. Show the wrong determination

As for B3, the premise of dense determination is that there must be three or more red tokens on the board. If not, it can only randomly block. However, B3 could effectively avoid the confusion of some randomly placed red tokens. For example, B3 could reach 90% in Test 6 (shown in Fig. 41).

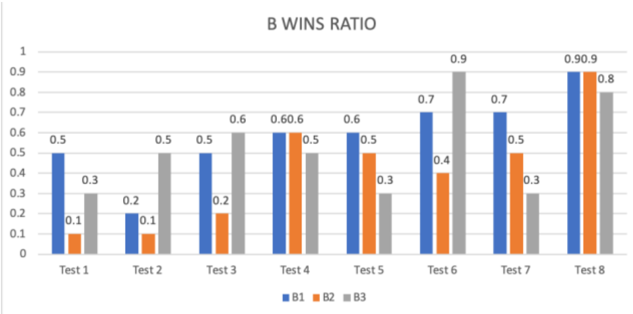

Fig. 41. Show the B wins ratio

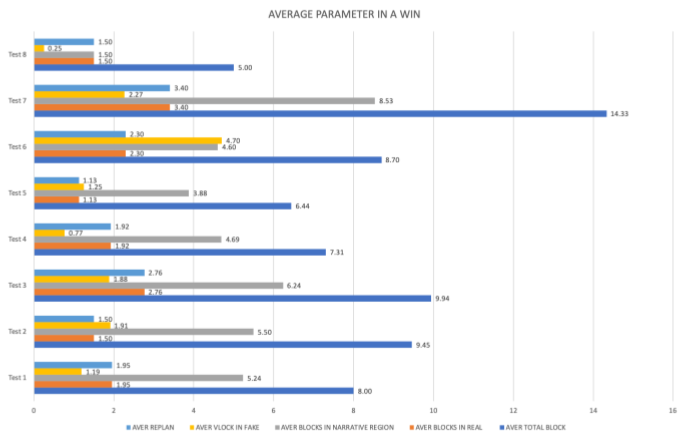

Fig. 42. average parameter in A win situation

From the Fig. 42, if the game can be controlled to end within 10 rounds, it seems easier for Player A to win the game. The following suggestions could increase the winning percentage of Player A by reducing the number of rounds of the game. For instance, we could narrow the distance between the target anchors, increase the production ratio of red tokens, and increase the load per shuttle.

Moreover, as for the higher numbered Player A, sometimes because of the lack of time, the time could be appropriately increased.

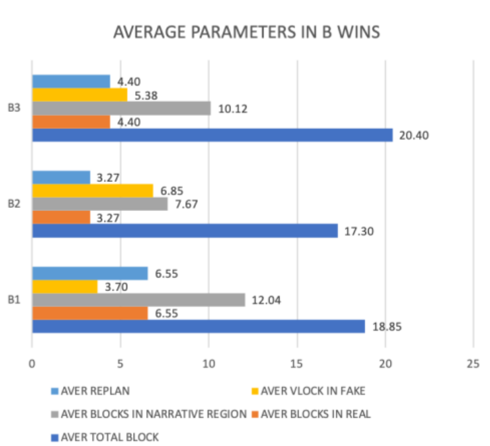

Fig. 43. average parameter in B win situation

By observing the Fig. 43, when the game round is more than 15 times, Player B is more likely to be the winner. Although combined with Fig. 42, the hypothesis one may not be true. However, from the game process, as for higher-numbered Player A, the reason why Player could be the winner is time out. Because of in the process of confusing B, such as the red token is thrown to a random location on the board, A will waste a certain amount of gaming time. Therefore, the suggestion that increase B's winning ratio is contrary to the suggestion of Player A. Like the anchor distance could be enlarged, decrease the production ratio of red tokens or given less gaming time for Player A.

Acknowledgment

Although this report was independently done by myself, it is inseparable from the help of Professor Michael and Dr Peta. They helped me get familiar with Unity and deception methods. Although some functions have been completed, due to time constraints, there are not good prior goals, such as machine learning agents and real manual testing. Hope you guys enjoy the deceptive world.

References

[1] J. Shim and R. Arkin, "Biologically-Inspired Deceptive Behavior for a Robot", 2019. .

[2] A. Vrij, "Detecting lies and deceit: the psychology of lying and implications for professional practice", *Researchportal.port.ac.uk*, 2019. [Online]. Available:https://researchportal.port.ac.uk/portal/en/publications/detecting-lies-and-deceit-the-psychology-of-lying-and-implications-for-professional-practice(29cc9a7d-d971-4f6f-9a36-a1f86b3e6ba0)/export.html.

[3] F. de Waal, "Intentional deception in primates", *Evolutionary Anthropology: Issues, News, and Reviews*, vol. 1, no. 3, pp. 86-92, 2005. Available: 10.1002/evan.1360010306.

[4]J. Shim and R. Arkin, "Biologically-Inspired Deceptive Behavior for a Robot", 2019. .

[5]O. Master, "Spider-eating Spiders", *Pustaqa.com*, 2019. [Online]. Available: https://pustaqa.com/spider-eating-spiders/.

[6]"notruilin/DeceptionGame", *GitHub*, 2019. [Online]. Available: https://github.com/notruilin/DeceptionGame.