Assignment3

Kohei Tateyama (S6427214), Ewen Gay-Semenkoff (S6475899) Università di Genova

***Abstract*—This report evaluates the performance of a robot simulation designed to collect tokens in a randomly set environment. The simulation was run multiple times to measure execution time and success rate. Data analysis revealed significant differences in these metrics, which were compared with a peer's results to benchmark performance. The findings highlight the impact of different coding strategies on the robot's efficiency and reliability.**

***Index Terms*—Keywords, keyowrds, keywords**

I. Introduction

IN the first semester, we developed a robot simulation to collect tokens placed in an environment. This project introduced us to basic robotic algorithms and environment interaction. In this assignment, the focus shifted to evaluating the performance of the simulation by measuring execution time and success rate through repeated trials.

We ran the simulation multiple times, recording execution times and calculating success rates based on token collection. The results were analyzed to identify performance patterns and compared with a classmate's implementation to highlight differences in coding strategies. These tests were conducted by two students, **Kohei Tateyama** (S6427214) and **Ewen Gay-Semenkoff** (S6475899), from Robotics Engineering at the University of Genoa. This comparison provided insights into the efficiency and reliability of our robot simulation, informing potential improvements for future iterations.

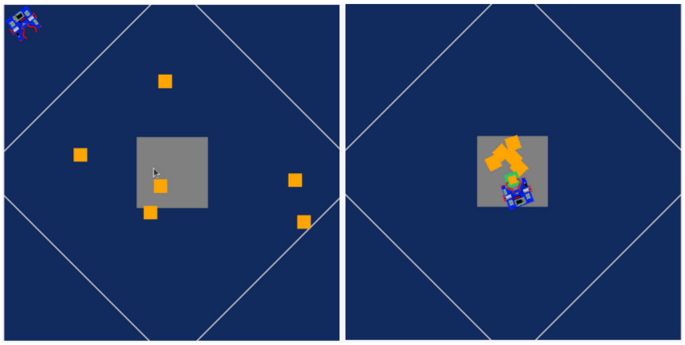


Fig. 1: Before and after the executiong of the code

II. Setup

Figure 1 illustrates the initial and final states of the robot simulation environment. The left side of the figure shows the environment with tokens randomly placed. The right side depicts the environment after the robot successfully gathered all the tokens, demonstrating the effectiveness of the robot's algorithm in navigating and collecting the tokens.

Figure 2 showcases the randomness of the token placement within the environment. This figure emphasizes the variability in the initial conditions of each simulation run, highlighting the challenges the robot algorithm must overcome to achieve consistent performance.

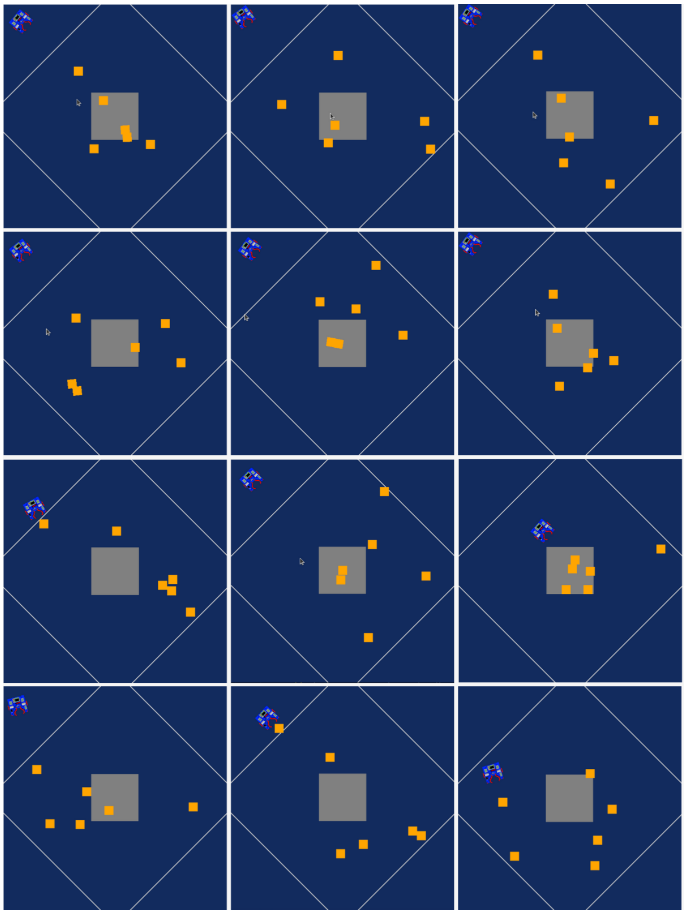


Fig. 2: The setup

III. Results

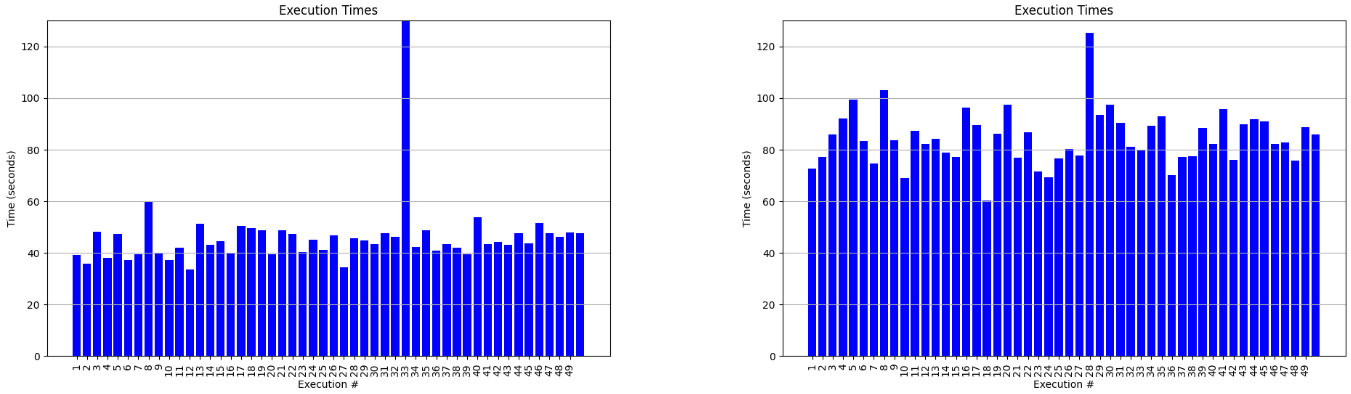
A. Comparison of the execution time

Figure 3a presents a bar graph of the execution time for each attempt of Kohei's code. The simulation was run 50 times, with each bar representing the execution time for one run, illustrating the variability and consistency in performance.

Figure 3b presents a bar graph of the execution time for each attempt of Ewen's code. Similar to Figure 3, this graph shows the execution times over 50 runs, allowing for a comparative analysis of performance between the two implementations.

B. Comparison of the success rate

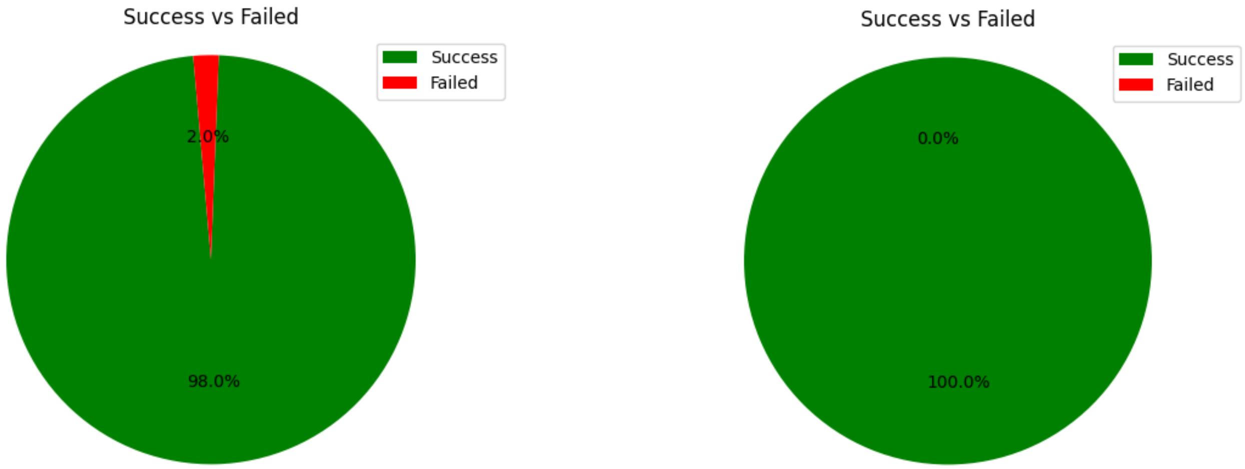
Figure 4 shows the success rates of the two codes in a pie chart. The left side represents **Kohei's code** with a success



(a) Execution time of Kohei's code

(b) Execution time of Ewen's code

Fig. 3: Comparison of Execution Times



(a) Success rate of Kohei's code

(b) Success rate of Ewen's code

Fig. 4: Comparison of Success Rates

rate of 49 out of 50 runs, equating to **98%**. The right side represents **Ewen's code** with a success rate of 50 out of 50 runs, achieving **100%**.

IV. Conclusion

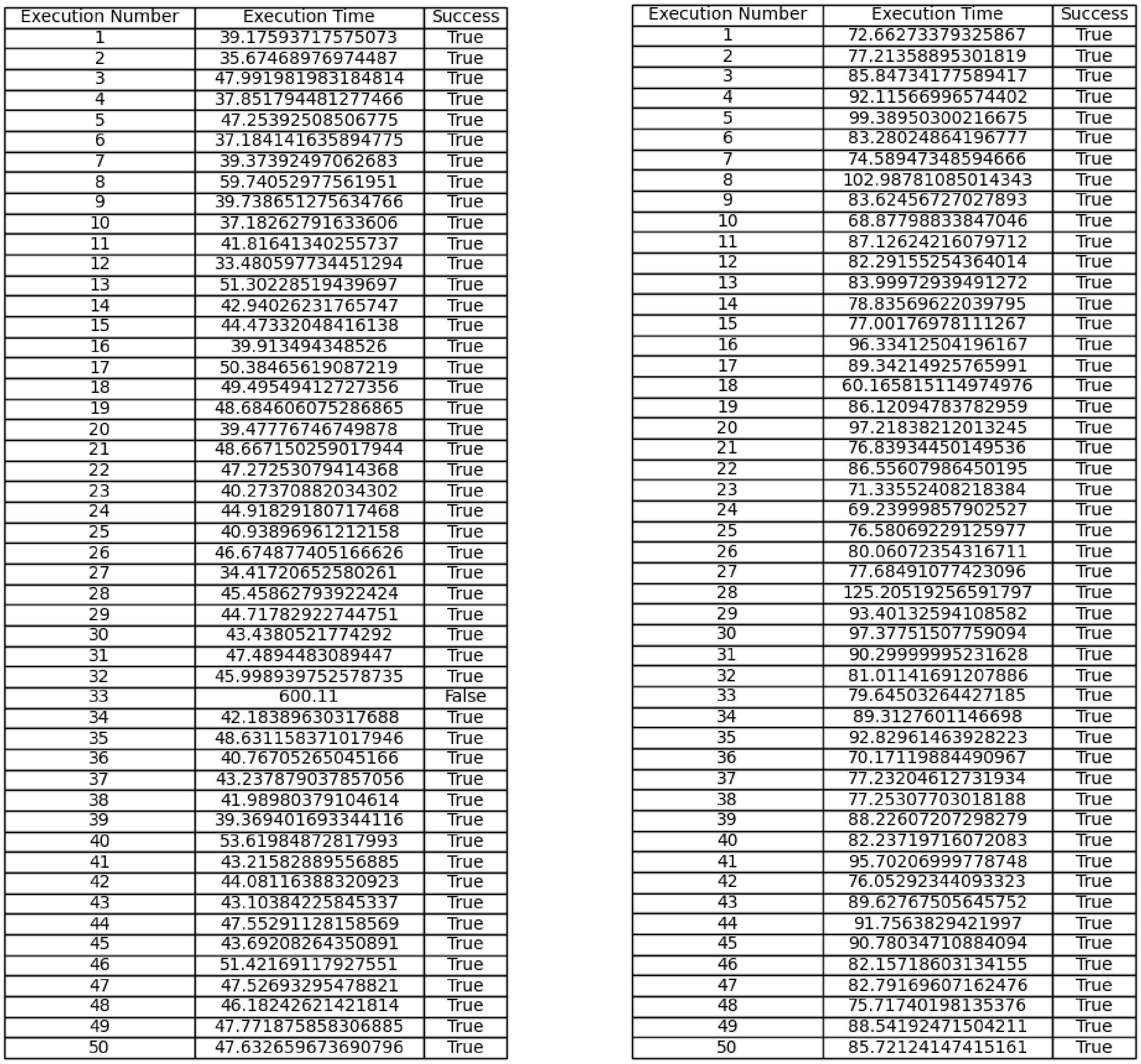
In this study, we evaluated the performance of two robot simulation codes designed to collect tokens in a randomized environment. Kohei's code achieved a success rate of 98%, successfully collecting tokens in 49 out of 50 runs. This indicates a high level of accuracy but also highlights room for improvement in the algorithm to achieve 100% success. On the other hand, Ewen's code demonstrated robustness with a perfect success rate of 100%, successfully completing the task in all 50 runs.

However, a significant difference was observed in the execution times of the two codes. Kohei's code had an average execution time of approximately **44.19 seconds**, demonstrating greater efficiency in speed. Conversely, Ewen's code, while robust, had a slower average execution time of approximately **84.41 seconds**. This indicates that although Ewen's algorithm is highly reliable, it requires optimization to improve its execution speed.

In conclusion, while Kohei's code shows promise in terms of speed, there is potential to enhance its accuracy. Ewen's code, despite its robustness, would benefit from improvements to reduce execution time. Future work should focus on optimizing the algorithms to balance both accuracy and efficiency, leading to more effective robotic simulations.

Appendix

Figure 5 shows the table with all the execution times for each attempt and also indicates whether each execution was successful or not. On the left is Kohei's code, and on the right is Ewen's code.



Execution Number	Execution Time	Success
1	39.17593717575073	True
2	35.67468976974487	True
3	47.991981983184814	True
4	37.851794481277466	True
5	47.25392508506775	True
6	37.184141635894775	True
7	39.37392497062683	True
8	59.74052977561951	True
9	39.738651275634766	True
10	37.18262791633606	True
11	41.81641340255737	True
12	33.480597734451294	True
13	51.30228519439697	True
14	42.94026231765747	True
15	44.47332048416138	True
16	39.913494348526	True
17	50.38465619087219	True
18	49.49549412727356	True
19	48.684606075286865	True
20	39.47776746749878	True
21	48.667150259017944	True
22	47.27253079414368	True
23	40.27370882034302	True
24	44.91829180717468	True
25	40.93896961212158	True
26	46.674877405166626	True
27	34.41720652580261	True
28	45.45862793922424	True
29	44.71782922744751	True
30	43.4380521774292	True
31	47.4894483089447	True
32	45.998939752578735	True
33	600.11	False
34	42.18389630317688	True
35	48.631158371017946	True
36	40.76705265045166	True
37	43.237879037857056	True
38	41.98980379104614	True
39	39.369401693344116	True
40	53.61984872817993	True
41	43.21582889556885	True
42	44.08116388320923	True
43	43.10384225845337	True
44	47.55291128158569	True
45	43.69208264350891	True
46	51.42169117927551	True
47	47.52693295478821	True
48	46.18242621421814	True
49	47.771875858306885	True
50	47.632659673690796	True

(a) Table of Kohei's Code Execution Times

Execution Number	Execution Time	Success
1	72.66273379325867	True
2	77.21358895301819	True
3	85.84734177589417	True
4	92.11566996574402	True
5	99.38950300216675	True
6	83.28024864196777	True
7	74.58947348594666	True
8	102.98781085014343	True
9	83.62456727027893	True
10	68.87798833847046	True
11	87.12624216079712	True
12	82.29155254364014	True
13	83.99972939491272	True
14	78.83569622039795	True
15	77.00176978111267	True
16	96.33412504196167	True
17	89.34214925765991	True
18	60.165815114974976	True
19	86.12094783782959	True
20	97.21838212013245	True
21	76.83934450149536	True
22	86.55607986450195	True
23	71.33552408218384	True
24	69.23999857902527	True
25	76.58069229125977	True
26	80.06072354316711	True
27	77.68491077423096	True
28	125.20519256591797	True
29	93.40132594108582	True
30	97.37751507759094	True
31	90.29999995231628	True
32	81.01141691207886	True
33	79.64503264427185	True
34	89.3127601146698	True
35	92.82961463928223	True
36	70.17119884490967	True
37	77.23204612731934	True
38	77.25307703018188	True
39	88.22607207298279	True
40	82.23719716072083	True
41	95.70206999778748	True
42	76.05292344093323	True
43	89.62767505645752	True
44	91.7563829421997	True
45	90.78034710884094	True
46	82.15718603134155	True
47	82.79169607162476	True
48	75.71740198135376	True
49	88.54192471504211	True
50	85.72124147415161	True

(b) Table of Ewen's Code Execution Times

Fig. 5: Comparison of Data Tables