
Data Science

Computer Science

Capstone Report - Spring 2020

# Popularizing Distributed Systems

*Hongyi Li,*

*Hongyi Liu*

Supervised by

Olivier Marin

**Abstract**

*Many beginners find the abstractions and jargon in the study of distributed systems hard to comprehend. This is not conducive for beginners to form an intuitive understanding of the problems and algorithms. In this study, we propose dramatization as a pedagogical tool to help beginners understand distributed algorithms. We embed the concepts of distributed algorithms in a play set in the universe of Gulliver's Travels [1]. We present the drama in the form of a role-playing game (RPG). Our preliminary experiment shows that dramatization can improve beginners' understanding of distributed problems. However, we believe that there is still much to improve regarding the presentation of the drama.*

# 1. Introduction

Encompassing a wide range of topics, distributed systems is a difficult area of research as well as a difficult subject to study and understand, especially when it comes to distributed algorithms. This is because, first of all, computer scientists model and reason about the problems in distributed systems based on a set of assumptions. Knowing the assumptions as well as their implications is crucial to the proper understanding of distributed problems. However, many assumptions are not intuitive for people without any background in distributed systems to intuitively adopt. When they try to understand a distributed problem without the proper assumptions in mind, they are bound to get perplexed. Secondly, problems and algorithms are often formally defined with mathematical terms. A formal definition is perfect for reasoning about correctness, performance, and impossibility rigorously but it also introduces a huge amount of jargon, which even those who have formal training in discrete mathematics might struggle to grasp. For these reasons, distributed systems are usually high-level courses only open to graduate students or senior undergraduates, who are familiar with the relevant abstractions and capable of reasoning with algorithms formally and structurally.

On the other hand, distributed systems are increasingly commonplace nowadays. For example, recently we see a proliferation of blockchain technologies. People are fascinated by the decentralization that blockchain technology can bring because it can revolutionize many industries. Blockchain technologies are especially hot in logistics and financing industries. However, few people understand what's under the hood and the lack of proper understanding of the technology can lead to poor decision making. As Maurice Herlihy points out, "Many of the basic algorithms and techniques used in blockchains are best understood as variations on familiar algorithms and techniques from classic distributed computing" [2]. Maybe even an intuitive understanding of the fundamental concepts in distributed systems may help people form a more critical view of the new technologies in distributed systems, and make more informed decisions on whether to adopt them.

We believe this situation calls for the vulgarization of distributed systems. By saying 'to vulgarize', we mean to eliminate the usage of archaic jargon and translate the mathematical terms in formal reasoning to simple English. In short, vulgarization is to popularize knowledge without replacing formal reasoning. Popularization should make the knowledge of distributed systems accessible to whoever is interested so that they can intuitively adopt the assumptions

in distributed problems and follow the reasoning to reach a basic understanding of the problem, and learn about the capabilities and limitations of different algorithms and solutions.

## 2. Related Work

Our literature review consists of two sections: an overview of the existing pedagogical tools/methods for teaching distributed problems/algorithms, and an analysis of their evaluation methods. Some of the previous works related in this area are studies in various approaches in computer science teaching, be they in introductory courses or specifically distributed systems. Others look at applications of technology as teaching aid. They are not two independent fields, however. Pedagogical designs and computer-aided teaching can be interrelated in many ways.

Based on our findings, and because the prospect of conducting in-person activities during this semester does not look promising due to the COVID-2019 pandemic, we want to contribute to this field by developing pedagogical tools that can be easily deployed in online settings.

### 2.1. Overview of Previous Works

In this section, we first present the existing pedagogical tools we found during our literature survey. We roughly sort the tools into two categories by their formats: 1. pedagogical computer programs; 2. learning activities. We emphasize the design principles of these tools. Finally, we discuss the pros and cons of each format.

#### 2.1.1. Pedagogical Computer Programs

Some pedagogical programs and programming frameworks for distributed systems in the literature focus on the aspects of simulation and visualization of high-level abstractions of distributed problems and algorithms [3, 4, 5, 6], while the others provide a virtualized distributed environment to simulate real-world distributed systems [7, 8, 9].

[3, 5, 6] are algorithm simulation/visualization frameworks. All of these frameworks put an emphasis on providing a versatile library that can enable users to trace the execution of the algorithms and explore different scenarios, which is considered to be extremely helpful in understanding the problem and algorithms.

DAJ (Distributed Algorithms in Java) [3] allows students to interact with the states of processes, leading to a better understanding of the state changes of the algorithms. For example, [4] is a

simulation and visualization of the Byzantine Generals Problem developed in DAJ.

LYDIAN (Library of Distributed Algorithms and Animation) [5] has been designed to support students in understanding, developing and analyzing distributed algorithms. It provides a TCL/TK graphical user interface endowed with an animator mode to visualize the behavior of the algorithms. The user can either create their own protocol to simulate (and eventually to visualize) or import an animation trace file for visualization. EnViDiA [6], which is a part of the LYDIAN project, visualizes the execution of distributed algorithms by using the visualization enhancements offered by Virtual Reality technology. It addresses to represent the complex flow of information tied with the execution of a distributed algorithm in a way that even novices can easily develop a first understanding of the algorithm behavior.

ViSiDiA (Visualization and Simulation of Distributed Algorithms) [10] implements several types of distributed systems: synchronous or asynchronous ones, mobile agents and sensor networks. In addition, this framework allows users to implement their own algorithms using the Java programming language. It is also possible to define a distributed algorithm without writing any source code, through a graphical user interface (GUI) which enables users to draw the rewriting rules that correspond to the algorithm.

[7, 8, 9] try to supply distributed systems curricula with the experience of developing and evaluating distributed applications in real-world/virtual environments. These works stress the importance of a hands-on experience of distributed application implementation and distributed algorithmic problem-solving.

The teaching tool developed by Wein et al. [7] uses a virtualization layer to overcome many of the obstacles in resourcing, setup and configuration that make it difficult for instructors to work with substantial distributed system projects. On top of this virtualization layer is a novel gaming environment in which students work in teams to understand a nontrivial distributed system.

In [8], the author addresses the problem of lacking hands-on access to distributed computing platforms in undergraduate distributed systems courses. Through the use of shared computing testbeds, such as PlanetLab and GENI, and open-source technologies, such as Xen and Hadoop, students can perform large-scale, distributed experimentation even at small colleges.

In [9], the author introduces a distributed and configurable task environment in which students can program AI agents to solve various distributed problems. The configurable task environment allows students to focus on new concepts instead of learning a new environment every time.

### 2.1.2. Learning Activities

The Active Learning Classrooms (ALC) is studied by [11, 12, 13]. These three studies show that the physical environment of the classroom does not have a substantial influence on the learning effects. It is the inverted classroom method adapted in ALC that to a larger extent facilitates learning. However, these experiments were conducted in CS1 and CS2. Students taking theses courses could be be less familiar with computer science and may have less interest and confidence in learning it. In distributed systems, however, we believe that students are generally CS or related majors who know the basics of data structures, algorithms and have programming experience. Therefore, such an approach may not be directly applicable to teach distributed systems.

CAROUSEL (Collaborative Algorithm Representations Of Undergraduates for Self-Enhanced Learning) [14] is a platform where students create a representation of algorithms in forms like graphics, texts, etc. and share with their peers. The authors found that creating representations and posting them is helpful for learning. Graded Group Activities (GGAs) conducted in [15] are a set of learning activities where students take exams in a group after having finished them individually. According to this research, the performance of students in the exams as a group is correlated to their final exam scores. These two methods, along with the active learning approach share an emphasis on peer instruction (solving problems together, sharing thoughts, etc.) which could be utilized in our study.

In [16], Sivilotti and Pike introduce several kinesthetic activities for learning distributed systems e.g. non-deterministic sort and stablizing leader election. In addition to detailed activities, key implementation notes are listed in their work. Using actors could also be a way to enhance the learning of distributed algorithms [17]. However, the authors raise the hypothesis that using virtualization in place of human animation might work just as well. We might also divide students into actors and audience and see if acting out the algorithms to learn could enhance learning as well. In addition, as we are now in a special period when the possibility of face-to-face teaching/learning is dramatically challenged, we could well focus on a virtual approach from the beginning.

## 2.2. Evaluation Methods

In this section. we look at how some of the authors evaluate their projects in their papers. We discuss both quantitative and qualitative approaches. From this, we extract a set of possible metrics for assessing our progress.

### 2.2.1. Quantitative Evaluation

Many of the studies use test scores as a key variable [15, 11, 12, 13, 14]. Finding whether there is a correlation between a factor and test scores is a straightforward way of evaluation. Such a correlation can be found through regression analysis. Controlled group experiments are done in such cases to ensure accuracy.

Other relevant statistical data are collected besides test scores, such as drop rates and failure rates[13]. These could be useful in conducting large-scale experiments; however, as within the time of this study (less than a semester) we cannot find many students to participate, they will not be adopted in our study.

While often used to gain insight into subjective feedback, questionnaires can also be quantified to gauge participants' evaluation. [17] converts students' feedback to statistics to show how students feel about the approach in different aspects.

### 2.2.2. Qualitative Evaluation

Qualitative evaluations of pedagogical tools/methods can rely on observations of student's performance during sessions as well as post-session feedback collected via interviews and questionnaires.

For the qualitative evaluation of pedagogical computer programs, ease of use (learning curve of the tool) and whether students can find something unexpected from the interaction are the most important criteria. The questions in feedback questionnaires should relate to these issues.

For learning activities, participants' interaction during the activities is a good indication of their understanding of the problem. For example, in [18], researchers assess which stage of the problem-solving process the student was in by observation. In [17], researchers consider the student's spontaneous discussion of solution and proof of correctness during the activities as indicators of improvement in understanding. Therefore, student engagement in the activities is vital.

## 3. Solution

The solution we propose to vulgarize distributed systems is dramatization. Presenting distributed algorithms in the form of fables is not something unheard of. The fault-tolerant consensus problem is known as the Byzantine General Problem. Turing award laureate Leslie Lamport presented the famous Paxos consensus algorithm as running a part-time parliament on a fictional Hellenistic

island called Paxos [19]. Although the original Paxos paper is known to be hard to read and comprehend, anthropomorphism of the elements in a distributed problem allows people to relate to the problem through its similarities with a real life situation, and to understand the abstractions better. What's more, assumptions of the problem can be convincingly embedded in the story as the setting of the fictional universe. Authors might use a story to introduce the problem and then formalize the problem using mathematical language, from which they can derive observations and conclusions and propose algorithms to solve the problem.

Dramatization takes this idea a step further: the problem is not only introduced in a story, but we try to present the reasoning and solution through the plot. The characters in the play can come up with the solution through conversation and monologue and perform the solution so that the audience can also see the simulation of algorithms. Additionally, uncertainties and non-determinism are common in distributed algorithms. Failures of participants or the communication channel can complicate a lot a seemingly simple problems. Novices often do not realize or anticipate this. In the same time, uncertainties and plot twists make for good drama and the audience can discover unexpected failures of algorithms through unexpected turns of the plot.

### 3.1. Our Dramatization Project

Among all the theories/algorithms, we choose the Two-Phase Commit Protocol (2PC) [20] as our first topic. The failure and recovery schemes in the 2PC protocol are thoroughly discussed in [21]. Our preliminary goal is to produce a play to illustrate the 2PC algorithm as well as its crash and recovery intuitively and engagingly. The story of Gulliver and the small humans [1] came to our mind.

The small-human empires, Lilliput and Blefuscu are hostile to each other because of their different ways of cracking eggs. In our story, one of the empires wants to restore peace with the other by abolishing its law regulating how to break eggs. This way we make Gulliver the coordinator and the two small-human empires the participants in the 2PC and make up the settings in a way that they satisfy all the key attributes of the protocol: (1) The two empires want to reach an agreement (i.e. consistency), namely abolishing their laws about breaking eggs and restore peace; if the agreement is not reached somehow, they should remain the same, that is, it should not happen that one empire abolishes its law but the other does not. (2) The two empires do not communicate directly with each other because there is little trust between them; instead they turn to Gulliver whom they think is a wise, fair man (i.e. elected coordinator). (3)

The two empires do not necessarily agree with the proposal; even if they do, their messengers could die on their way (i.e. message lost), or the empire could be at an emergency state e.g. civil war, natural disaster, etc. so that they are unable to learn anything from the outside (i.e. site failure). In order to analyze the single-site failure, we come up with a new setting: a mystic typhoon generating in the middle of two islands, which will only hit one of them, and Gulliver moves to one of the empires, so that only one empire could be in chaos.

We are cooperating with a playwright so that we can expect the story to be presented with dramatic, intriguing lines rather than plain, uninteresting dialogues. We provided the playwright with the summary of the plot as writing prompt and the playwright developed scenes, characters, conversations and all the other details one would expect in a play. At the end of our capstone project, we have finished the scripts for the failure free scenario (a fault-free execution of the 2PC protocol), the message-losts scenario (network partition) and the single-site-failure scenario (one site crash). The play script is hosted on Google Drive and accessible with this link: (https://drive.google.com/file/d/1RJE5P1kjuPaDperII7NyOo7JHi1MJa4R/view?usp=sharing).

## 3.2. Presentation of the Drama

Ideally, we would hire actors and direct the play. But the reality is that we must conform with the imperative of social distancing during the COVID-19 pandemic, which makes a live play virtually impossible. Moreover, directing a play is really not our expertise and we do not have connections to actors like the authors of [17] do. Thus, we turn to another media of storytelling – computer games. In comparison with a theatrical play, a computer game is more interactive and can be easily distributed. It does not require public gathering, either.

We therefore surveyed a few tools for developing computer games.

ChengGuang or 66RPG [22] is an RPG community which also provides an official tool of developing RPGs. No programming background is needed for developing a game, since many built-in modules as well as tutorials are available. Its limitation, however, is that such games can only be deployed on their website. Also, the website and development tools are all in Chinese, which makes it difficult to use or maintain for non-Chinese speaking students/faculty.

HuanJing or EvkWorld [23] is like an offline version of 66RPG. One can choose to export the game as a mobile app or as an H5 game. However, there is no English support in the development tools, either.

RPG Maker [24] allows to build old-Nintendo-style RPG games. There is large freedom and

| Tool | Pros | Cons |
|---|---|---|
| 66RPG | Easy to learn, robust functionalities available | Limited to their platform, only in Chinese |
| EvkWorld | Portable, robust functionalities | Only in Chinese |
| RPG Maker | Large freedom, various functionalities | Map-oriented, character movement focused |
| Evennia | Open-source, large freedom | Text only, steep learning curve |
| Ren'Py | Open-source, supports graphics and audio | Needs visual art |

Table 1: Comparison of RPG development tools.

good graphic support. It is more map-oriented rather than scene-oriented.

Evennia [25] is a Multi-User Dungeon (MUD) engine based on Python. It makes classic command-line style dungeon games. The game content consists only of text. It is open-source and needs some programming background; consequently it has a larger learning cost.

Ren'Py [26] is an RPG engine based on Python. It is also open-source and needs some programming background, but is relatively easier to learn. Like the previously mentioned RPG tools, it supports character portraits, background image and audio. It does not come with many pre-set modules, however; and as we cannot find any character portraits of Gulliver and the small people, we would have to find an illustration artist to develop the visual arts.

As is mentioned above, we could have developed a game using an all-Chinese tool, but that would have made it much more difficult to maintain and would also have limited the user group. Therefore, we had to abandon 66RPG and EvkWorld despite the robust and easy to use functionalities they provide. RPG Maker focuses more on maps and character movement which we do not need. The choices left were Ren'Py and Evennia. Considering that we did not have anyone who can develop visual art and that time is limited, we decide to develop our game using Evennia. We recognize that Ren'Py has great potentials in user experience, so we hope to explore this path in the future.

### 3.3. Implementation

Like any MUD game, the world of Evennia consists of rooms (Figure 1), exits that connect the rooms, and characters. Players interact with the world by entering textual commands. In our game, rooms are mapped to the scenes in our play scripts where the plots take place (Figure 2). Players can proceed from one scene to another by taking the exits. Entering a new room will trigger the plot to unroll, and characters will start to deliver the lines in the play script as if they

were the actors in the play. Character names, lines, actions are encoded in different colors so that players can easily differentiate them. After the scene is finished, players can interact with the characters to learn about the current state they are in and what message they are expecting (Figure 5) so that they can move to the next state. This feature is designed to simulate a finite state machine. One room can have multiple exits, which allows branching of plots. For example, our player can choose to see what happen when a message is lost at a certain point of the protocol (Figure 4), but the failure branches are invisible to the players until they finish the story line where the protocol runs smoothly without any failures. This is to make sure that they are familiar with how the protocol runs in normal operation. The flowchart overview of the game is shown in Figure 5.

Considering that MUD is a niche category in the world of gaming, we also develop a tutorial to help players get the hang of the gameplay (Figure 6). The game is deployed on the Internet (http://maxee.cn:4001) and any one can play it using a browser. We invite the reader to try it for themselves.

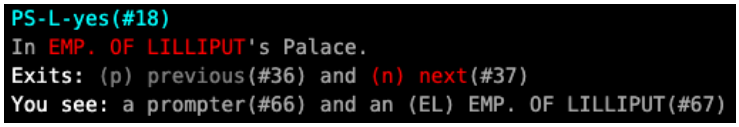


Figure 1: A Room

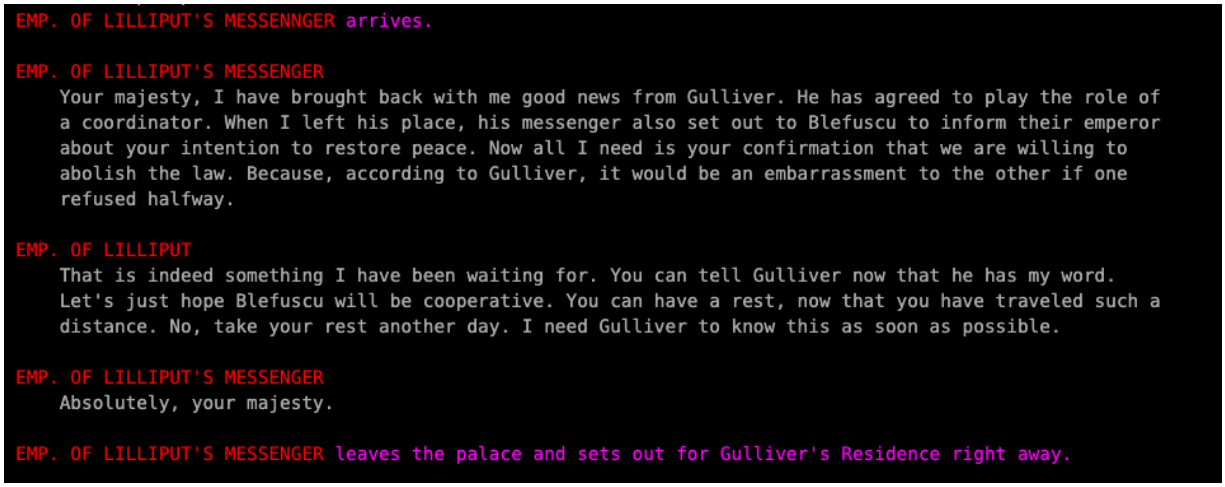


Figure 2: A Plot Talking Place

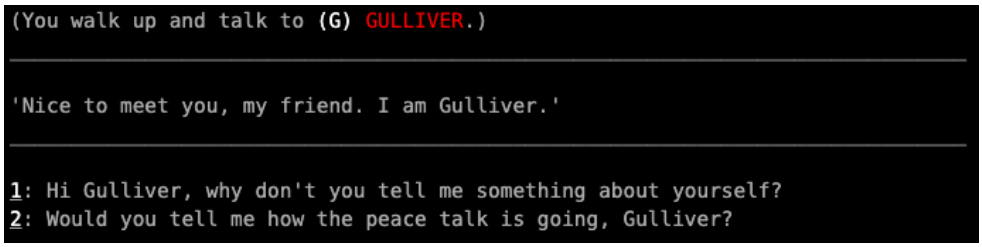


Figure 3: Interacting with an NPC

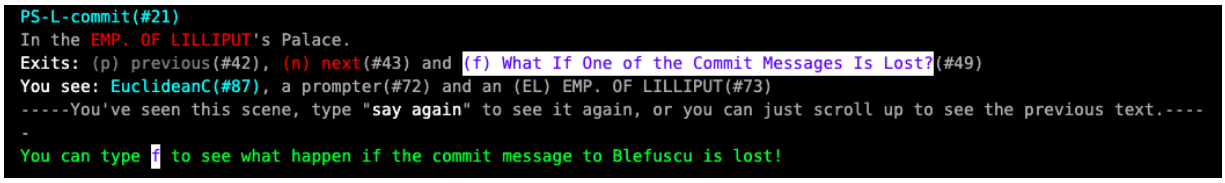


Figure 4: A Ramification Point in the Game

## 4. Results and Discussion

In order to see if our method can actually have the desired effects, we need to test it among students and observe how much understanding of the 2PC protocol they have gained. Quantitative analysis should be conducted to measure the efficacy; qualitative/subjective feedback should also be collected so that we can spot problems and improve our product.

### 4.1. Experimental Setup

We designed a pre-game quiz (Appendix A), a post-game quiz (Appendix B), and a feedback questionnaire (Appendix C).

The pre-game quiz tests the participant's understanding of the Two Generals Problem [27] which is the base form of the problem that the 2PC tries to solve, including (1) how mutual agreement is reached when messages are guaranteed to arrive and (2) whether there is a solution for the problem when messages are sometimes lost (there is not).

The post-game quiz tests (1) whether the participant has learned about the goal of 2PC (consistency) from the game and (2) whether they understand that consistency cannot be guaranteed in such kinds of problems when messages do not reach their destination.

A questionnaire is attached to the post-game quiz to collect feedback of the game. The questionnaire collects feedback on (1) whether the subject is familiar with this kind of game, (2) the theme/setting of the story, (3) user experience during game play, (4) the tutorial for the game, etc.

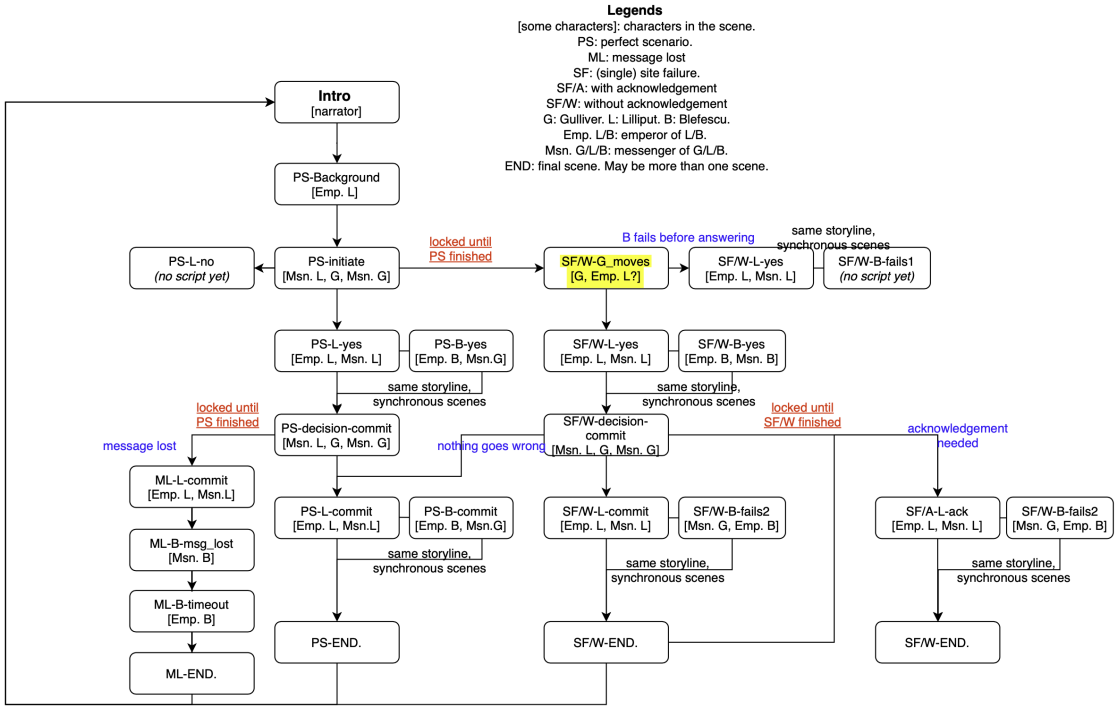


Figure 5: Overview of the Game

The quizzes and the questionnaire are available as appendices at the end of the paper. The reader is welcome to try the questions by themselves. It is recommended that they follow the same procedure as the test subjects: take the Pre-Game Quiz, then play the game, followed by the Post-Game Quiz.

We use some of the Marvel characters to model the Two General's Problem. We do not have any cooperation with Marvel, and do not plan to use this for any commercial purposes.

## 4.2. Quantitative Analysis

### 4.2.1. Data Processing

We begin processing the data by exporting the responses as csv files.

For answers to questions that has a standard answer, the data point value is set to 1 if the answer is correct and 0 if the answer is not. For other questions, positive answers are marked 1 and negative 0 (e.g. RPG Exp). In some cases a middle-ground answer (Gulliver Exp and Game_Storytelling) like "maybe" will be marked 0.5, which only happens in the Feedback Questionnaire. We preserve the answer values of scaled question (e.g. Confusion), and manually process textual feedback answers.

Specifically, we have different rules for grading some of the questions. The reader might need

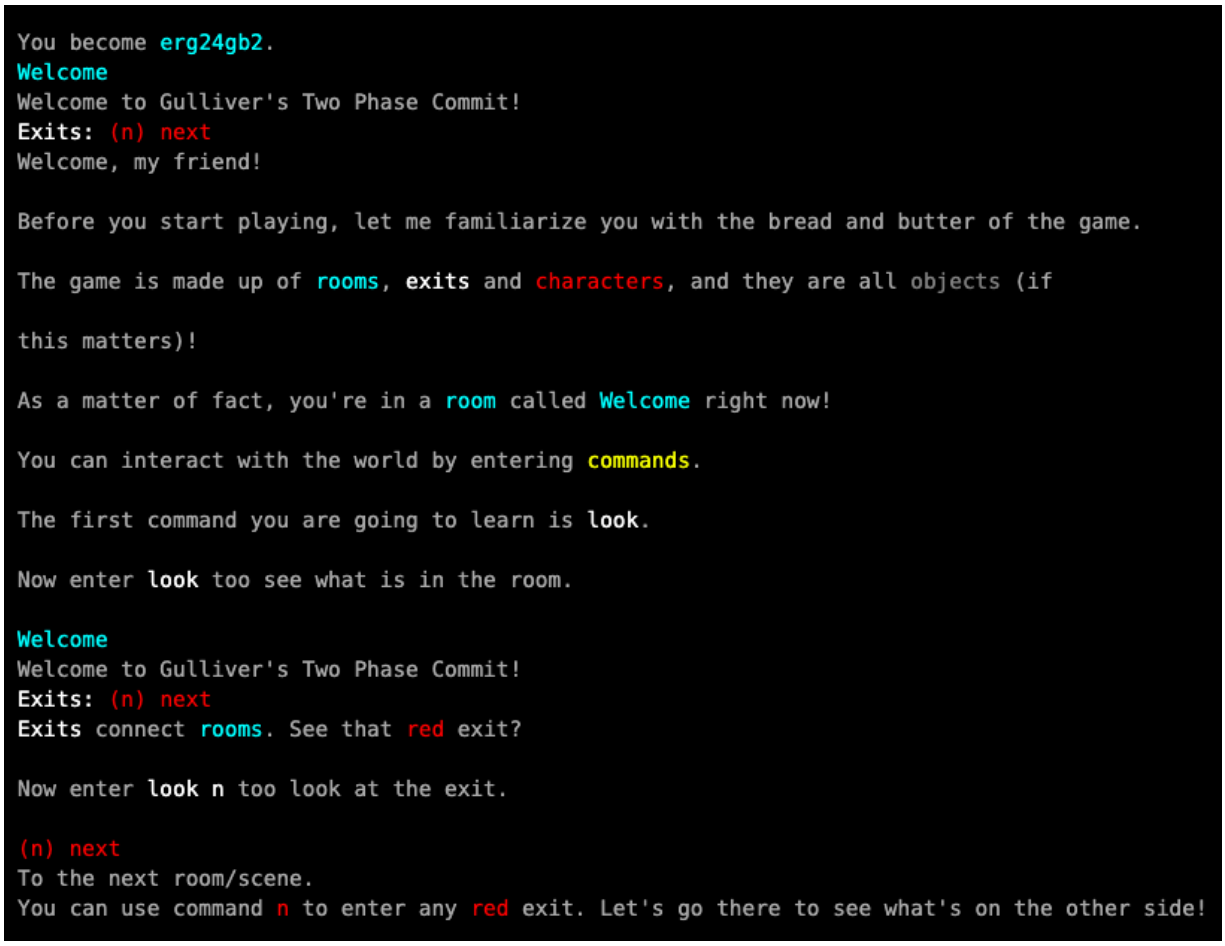


Figure 6: A Snippet of the Tutorial

to refer to the appendices before they continue reading.

For the first question (Q1) of the Thanos (Two Generals) problem in the Pre-Game Quiz, for which there exists a solution, if the answer is "Yes." but the explanation is incorrect (examined manually), it is marked 0. If the answer is "I think there is one, but I don't have a clue", it is marked 2, considering the possible difference in the testing subjects' understanding of the problem. In the second question (Q2 and Q2_again) of the Thanos problem and the same repeated question in the Post-Game Quiz, where there is no possible way out, since both answers are positive and thus incorrect, they will both be marked 0. For the question regarding the common goal (Common Goal) of the Thanos problem and the Gulliver story, if one answers "consistency", "synchronization", "mutual agreement", or provides any similar description, it is marked correct, the data point set to 1. Other answers such as "peace", "coordination" or "cooperation" are marked incorrect, the data point set to 0.

For example, for questions [Q1, Q2, Common Goal, Q2_again], if one's answers are ["Yes" (but explanation incorrect), "I think there is one", "consistency", "No"], the response will be

| | N | Minimum | Maximum | Mean | Std. Deviation |
|---|---|---|---|---|---|
| Background | 27 | 0 | 1 | .33 | .480 |
| Q1 | 27 | 0 | 2 | N/A | N/A |
| Q2 | 27 | 0 | 1 | .33 | .480 |
| Common Goal | 27 | 0 | 1 | .67 | .480 |
| Q2_again | 27 | 0 | 1 | .44 | .506 |
| RPG Exp | 27 | 0 | 1 | .56 | .506 |
| Gulliver Exp | 27 | .0 | 1.0 | .370 | .4056 |
| Confusion | 27 | 1 | 10 | 6.26 | 2.782 |
| Game_Storytelling | 27 | .0 | 1.0 | .611 | .3490 |
| Q2_again - Q2 | 27 | -1.00 | 1.00 | .1111 | .50637 |
| Valid N | 27 | | | | |

Table 2: Descriptive Statistics of the Responses

processed as [0, 0, 1, 1]. For answers ["I think there is one", "No", "peace", "Yes"], the response will be processed as [2, 1, 0, 0].

Table 2 shows the descriptive statistics of the responses.

#### 4.2.2. Analysis and Interpretation of the Data

First of all, we see a rise in the mean score of Q2 before and after the game, from 0.33 to 0.44. 18 out of 27 test subjects thought there was a solution to the Two General Problem before they played the game. After they played the game, five of them realized the impossibility of solving the problem.

Secondly, for those who answered Q1 correctly, the increase of Q2 correctness after the game is even more significant (See Table 3). Analysis of the variance shows that the positive effect of answering Q1 correctly on the improvement of Q2 post-game is significant (passing the tests of Between-Subjects Effects with 0.01 significance). One of the purposes of Q1 is to examine whether the subject read the problem statement correctly and has formed an understanding of the problem setting. We assume that anyone who read the problem carefully can get Q1 right. If our assumption is sound, then the data indicates that our game can help understand the impossibility of solving the Two General Problem, provided that one follows the instructions carefully. In constrast, we see that for those who answered Q1 incorrectly, the game has a negative effect on their Q2 score. We think that they might be somewhat confused by the phrasing of the questions or the question in general.

Thirdly, the Confusion has a relatively high mean value of 6.26. We examine the influence of whether having knowing the game's reference to Gulliver's Travel (Gulliver Exp) and whether

Dependent variable: Q2_again - Q2

| Q1 | Mean | Std. Deviation | N |
|---|---|---|---|
| 0 | -.3333 | .51640 | 6 |
| 1 | .0000 | .00000 | 7 |
| 2 | .3571 | .49725 | 14 |
| Total | .1111 | .50637 | 27 |

Table 3: Q2_again - Q2 dependent on the value of Q1

having played RPG before (RPG Exp) on the level of Confusion. Analysis of Variance shows that only the product of the two values has a significant influence on the level of confusion (see Figure 7). The data suggest that to improve user experience and reach out to a broader audience, we must not only adopt a more popular, game format but also base the dramatization on a better-known story.

Dependent Variable: confused

| Source | Type III Sum of Squares | df | Mean Square | F | Sig. | Partial Eta Squared |
|---|---|---|---|---|---|---|
| Corrected Model | 80.638[a] | 5 | 16.128 | 2.809 | .043 | .401 |
| Intercept | 954.303 | 1 | 954.303 | 166.244 | .000 | .888 |
| RPG_exp | 9.394 | 1 | 9.394 | 1.637 | .215 | .072 |
| Gulliver_exp | 10.336 | 2 | 5.168 | .900 | .422 | .079 |
| RPG_exp * Gulliver_exp | 37.904 | 2 | 18.952 | 3.302 | .057 | .239 |
| Error | 120.548 | 21 | 5.740 | | | |
| Total | 1259.000 | 27 | | | | |
| Corrected Total | 201.185 | 26 | | | | |

a. R Squared = .401 (Adjusted R Squared = .258)

Figure 7: Univariate Analysis of Variance(Confusion) - Tests of Between-Subjects Effects

## 4.3. Qualitative Analysis and Discussion

Besides the quantitative analysis, the textual feedback from the test subjects allows us to evaluate our method qualitatively. Out of 27 test subjects, 20 gave textual feedback on what confused them during the game including one answer being “Nothing", so we have 19 effective answers here. Some answers expressed more than one element that confused them.

Many of the test subjects were confused by the gameplay (10 answers). Responses include the game being somewhat linear with not enough branches to choose, and the interface having only text with no graphics, which also brings problems with, for example, text rolling speed and input source. Of the concerns regarding the gameplay, most of them are related to the interface of the

game (8 answers).

The story itself was a problem for some test subjects, too (7 answers). From the previous question, we find that many have not read Gulliver's Travels, though some know the Big-Endian, Small-Endian reference which we guess comes from their CS study. We did not expect so many to be unfamiliar with the story at all. What confused the test subjects include the names in the story, too many roles, and too much background information. Besides, one user did not understand the "meaning" of the story. We also received complaints about the script being too wordy and too much extra information is given.

We have also received some positive feedback, recognizing the story design and game development.

We also realize that there has been some limitation in the testing process.

The test subjects are limited to NYU Shanghai students, most of whom are computer science/data science seniors and students taking the Distributed Systems course this semester (Spring 2020). Besides the somewhat homogeneous composition of the test subjects, we have a relatively small sample size with only 27 valid responses.

There was not a strong incentive for test subjects to finish the test. For most of them, the reward for finishing the test was one extra credit for the Capstone Project/Distributed Systems course; a few others were interested in the project and offered to test it when we asked. There was not much guarantee that the test subjects took the quizzes/games seriously. One could have randomly selected answers and leave no textual answer, and quickly go through the game without paying much attention. In fact, 32 test subjects finished the Pre-Game Quiz but not the Post-Game Quiz.

We also feel that some questions in the quiz might have been confusing, especially for those who did not read carefully. Some of the answers showed that the test subjects had misunderstood the question: they either gave irrelevant answers or thought that there was no solution to the first Thanos question. This has possibly distorted the test result and made the data analysis less accurate.

## 5. Future Work

In our future work, we will consider expanding the test group to more students with a wider variety of backgrounds. If the condition permits, that is, if offline classes can resume, we will

consider conducting in-class testing sessions where test subjects can ask questions if they have any difficulty during the test. We should design the quizzes with more concise language and have them reviewed by someone unfamiliar with the study before we distribute them.

We will consider developing the game with Ren'Py which has a non-command line GUI. All we need extra is someone who can draw the portraits and background images for the game. For the story, we might work on other background settings than Gulliver's Travels, and consider using more concise language. We will consider giving the user more freedom to choose what happens and explore the possible consequences. This way, if the condition permits, we can have students play the game in class for more interactive experience.

## 6. Conclusion

Inspired by previous work, we decided to dramatize distributed algorithms as a way of vulgarizing them. We developed stories based on the background of Gulliver's Travels and deployed the stories on an RPG game. Despite the limitations of presenting the drama in the form of a text-based RPG game, our preliminary experiment shows that dramatization of distributed problems and algorithms can help with learning about the topic without using any mathematical terms.

There are still many things to be considered and worked on regarding our research. First of all, we need to find a more rigorous definition of intuitive understanding so that the learning result can be better measured. Secondly, we need to test whether dramatization can help understand the reasoning behind the conclusions. Thirdly, we need to make our presentation of drama more user-friendly and engaging to capture the interest of a broader audience. That being said, we are optimistic that more outcome is yet to come if we keep working on this topic.

## Acknowledgement

We would like to thank our supervisor, Professor Olivier Marin for being incredibly encouraging and supportive. Without your encouragement, assistance and guidance, our ideas could never have gone this far.

We would like to thank Jikai Zheng (CO'21 NYU Shanghai), our playwright, for writing the play scripts and helping us develop the characters and dialogues. Your creativity is amazing and we wish you all the best in your future career.

We would like to thank Chi Cheng (CO'20 NYU Shanghai) for helping us analyze the experiment data.

We would like to thank Professor Azza Abouzied at NYU Abu Dhabi, from whose lecture slides we drew inspiration for the presentation of the Two Generals Problem.

We would like to thank those who participated in our testing session. Your feedback is very valuable to our project.

## References

[1] J. Swift, *Gulliver's Travels*. Cambridge, UK ; New York N.Y. : Cambridge University Press, 2012.

[2] D. Weyns, "Software engineering of self-adaptive systems," in *Handbook of Software Engineering*. Springer, 2019, pp. 399–443.

[3] M. Ben-Ari, "Distributed algorithms in java," *ACM SIGCSE Bulletin*, vol. 29, no. 3, pp. 62–64, 1997.

[4] A. Tikvati, M. Ben-Ari, and Y. B.-D. Kolikant, "Virtual trees for the byzantine generals algorithm," *ACM SIGCSE Bulletin*, vol. 36, no. 1, pp. 392–396, 2004.

[5] B. Koldehofe, M. Papatriantafilou, and P. Tsigas, "Lydian: An extensible educational animation environment for distributed algorithms," *Journal on Educational Resources in Computing (JERIC)*, vol. 6, no. 2, pp. 1–es, 2006.

[6] P. Holdfeldt, B. Koldehofe, C. Lindskog, T. Olsson, W. Petersson, J. Svensson, and L. Valtersson, "Envidia: an educational environment for visualization of distributed algorithms in virtual environments," in *Proceedings of the 7th annual conference on Innovation and technology in computer science education*, 2002, pp. 226–226.

[7] J. Wein, K. Kourtchikov, Y. Cheng, R. Gutierez, R. Khmelichek, M. Topol, and C. Sherman, "Virtualized games for teaching about distributed systems," *ACM SIGCSE Bulletin*, vol. 41, no. 1, pp. 246–250, 2009.

[8] J. R. Albrecht, "Bringing big systems to small schools: Distributed systems for undergraduates," in *Proceedings of the 40th ACM technical symposium on Computer science education*, 2009, pp. 101–105.

[9] J. M. Hill and K. L. Alford, "A distributed task environment for teaching artificial intelligence with agents," *ACM SIGCSE Bulletin*, vol. 36, no. 1, pp. 224–228, 2004.

[10] W. Abdou, N. O. Abdallah, and M. Mosbah, "Visidia: A java framework for designing, simulating, and visualizing distributed algorithms," in *2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications*. IEEE, 2014, pp. 43–46.

[11] Q. Hao, B. Barnes, E. Wright, and E. Kim, "Effects of active learning environments and instructional methods in computer science education," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 934–939. [Online]. Available: https://doi.org/10.1145/3159450.3159451

[12] T. Greer, Q. Hao, M. Jing, and B. Barnes, "On the effects of active learning environments in computing education," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 267–272. [Online]. Available: https://doi.org/10.1145/3287324.3287345

[13] A. Naeem Syeda, R. Engineer, and B. Simion, "Analyzing the effects of active learning classrooms in cs2," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 93–99. [Online]. Available: https://doi.org/10.1145/3328778.3366872

[14] T. Hübscher-Younger and N. H. Narayanan, "Constructive and collaborative learning of algorithms," in *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 6–10. [Online]. Available: https://doi.org/10.1145/611892.611919

[15] J. Kawash, T. Jarada, and M. Moshirpour, "Group exams as learning tools: Evidence from an undergraduate database course," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 626–632. [Online]. Available: https://doi.org/10.1145/3328778.3366889

[16] P. A. Sivilotti and S. M. Pike, "The suitability of kinesthetic learning activities for teaching distributed algorithms," in *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, 2007, pp. 362–366.

[17] B. Koldehofe and P. Tsigas, "Using actors in an interactive animation in a graduate course on distributed system," in *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 149–152. [Online]. Available: https://doi.org/10.1145/377435.377670

[18] J. Prather, R. Pettit, B. A. Becker, P. Denny, D. Loksa, A. Peters, Z. Albrecht, and K. Masci, "First things first: providing metacognitive scaffolding for interpreting problem prompts," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 531–537.

[19] L. Lamport, "The part-time parliament," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 277–317.

[20] B. Lampson and H. E. Sturgis, "Crash recovery in a distributed data storage system," January 1979. [Online]. Available: https://www.microsoft.com/en-us/research/publication/crash-recovery-in-a-distributed-data-storage-system/

[21] D. Skeen and M. Stonebraker, "A formal model of crash recovery in a distributed system," *IEEE Transactions on Software Engineering*, no. 3, pp. 219–228, 1983.

[22] Chengguang (66rpg). [Online]. Available: https://www.66rpg.com/

[23] Huanjing (evkworld). [Online]. Available: https://www.evkworld.com/

[24] Make your own game with rpg maker. [Online]. Available: https://www.rpgmakerweb.com/

[25] Evennia. [Online]. Available: http://www.evennia.com/

[26] The ren'py visual novel engine. [Online]. Available: https://www.renpy.org/

[27] J. Gray, "Notes on data base operating systems," in *Operating Systems, An Advanced Course*. Berlin, Heidelberg: Springer-Verlag, 1978, p. 393–481.

# Appendix

## A. Pre-Game Quiz

The test subjects are instructed to finish this quiz before playing the game.

1. Do you have any background in distributed systems?
   (a) I have taken/am taking a course in distributed systems, or have other experience.
   (b) No.

Read this story before answering the following questions:

Imagine Captain America and Iron Man are planning to fight Thanos. They are located in two mountains and cannot see each other. In the middle of the mountains, there is a valley where Thanos is at. Thanos is not undefeatable; if the two of them attack him at the exact same time, they can kill him. Otherwise, if either of them appears first, Thanos can one-shot him and then the other. The only way for them to communicate is a pigeon that flies across the valley.

2. Thanos doesn't know what the pigeon is for. All he is watching is whether his enemies have come. Can you come up with a way for the two heroes to kill Thanos? (Hint: you can use the pigeon more than once.)
   (a) Yes. (Specify your answer)
   (b) I think there is one, but I don't have a clue.
   (c) No way they can kill Thanos!
3. Now Imagine Thanos knows that the pigeon is the key to defeating him. Even though the pigeon is so fast that he can't notice it sometimes, there is a chance it gets spotted and shot. So whoever sends the pigeon doesn't know if the pigeon has arrived. Now can you come up with a way for the two heroes to attack at the same time? (Hint: even though the pigeon can get killed, if it's still alive, you can use it more than once.)
   (a) Yes. (Specify your answer)
   (b) I think there is one, but I don't have a clue.
   (c) No way they can kill Thanos!

The answer to Question 2 is yes. As long as one specifies a time to attack and the other agrees by sending an acknowledgment, they can attack.

The answer to Question 3 is no. For the other to know that one has received the message, one has to send an acknowledgment, but then the problem becomes the confirmation of the acknowledgment, and this could go on forever.

## B. Post-Game Quiz

The test subjects are instructed to finish this quiz after playing the game. After this section, they are asked to finish a questionnaire attached for feedback on the game.

1. In the story of Gulliver and the small humans and the story of Thanos and the superheroes, what do you think is the common goal they are trying to achieve? Either use one word, or one or two sentences to describe the goal. (Hint: think about what quality Gulliver is trying to make sure, and what condition must be satisfied so the superheroes can defeat Thanos)

2. Now, can you come up with a solution for the two superheroes in the second question of the Thanos problem, where Thanos might shoot the pigeon?

   (a) Yes. (Specify your answer)

   (b) I think there is one, but I still don't have a clue.

   (c) No! They can't kill Thanos.

The answer to Question 1 is consistency. Either both commit, or nobody commits.

The answer to Question 2 is no, as is explained in the pre-game quiz.

## C. Feedback Questionnaire

This questionnaire is attached to the post-game quiz.

1. Have you played a Dungeon-like game or an RPG (role-playing game) before?

   (a) Yes.

   (b) No.

2. Are you familiar with the novel Gulliver's Travels by Jonathan Swift?

   (a) Yes, I have read it before.

   (b) I haven't read it but I know the big endian, small endian reference.

(c) I know nothing about it.

3. From the scale of 1 to 10, how confused were you during the gameplay?

4. What confused you? ( the story, the gameplay, etc.)

5. From the scale of 1 to 5, rate the helpfulness of the tutorial.

6. What should we add to the tutorial?

7. Do you think our game is a good format for interactive storytelling?

   (a) Yes.

   (b) No.

   (c) Maybe.

8. Any additional comments?