Distributed Tic Tac Toe: Peer-to-Peer Team-Based Gaming System

Panagiotis Antoniou, Emily Band, Paulius Dilkas, Dafin Kozarev, Joshua Styles

***Abstract*—This article describes a Distributed Multiplayer Tic Tac Toe software game written in Java using the RMI API. While previous implementations attempting to serve a similar purpose exist, none of them boast good design or efficiency and neither of them deliver the same functionality. The developed piece of software allows clients to play locally or globally a game of the classic tic tac toe - but with a twist. Players are separated into teams and decide on what the next move should be by voting. The implementation is derived from a design that follows the distributed applications guidelines and is prepared to handle corner cases such as client disconnects for example.**

I. Introduction

Coding nowadays is the main occupation of loads of people and it is also a past-time for just as many. Most people get their first steps in writing code following the local single-threaded model. Often, it can be difficult to beginner programmers or even those with more experience to abstract the key knowledge they have obtained using this model and apply it in a different way. Distributed software requires a new look on the system the number of devices has gone up and with it so has the number of processes and other concerns. Some concepts have now changed and need to be dealt with differently one such example is mutual exclusion. We believe programs that serve as examples and focus on simple local tasks turned into distributed ones can help many people trying to make this step with a hands-on experience of what they are preparing for. Seeing theory in practice is a proven technique and there is evidence for this in every university course that features practical work. Therefore, we have chosen to build a distributed game of tic-tac-toe a game known to possibly everyone. By restraining from introducing additional complexity as part of the goal of the project, the focus is shifted towards the implications of the distributive aspect of the system and the way they are handled. The finished product also has another purpose it is an actual game that can be played and as such it can bring joy to people and increase their curiosity about computers.

II. System Design

III. Implementation

IV. Evaluation

V. Related Work

While other multiplayer distributed games already exist, they have not been created with this projects task in mind and therefore do not serve the same purpose. They can be split into two categories very unprofessional and too professional. For example, hobby implementations exist that deal with distributed problems in a bad fashion. Similarly, a single-player-per-team distributed tic-tac-toe game in Java can be found online that does not comply with any design guidelines and puts all functionality in a single class and as such cannot teach students good practice and might confuse them further about distributive programming [1]. Separately, very in-depth materials that provide state-of-the-art practices can be found [2] but there is a high possibility that they might be incomprehensible to a beginner reader looking to tackle basic distributed concepts to begin with. Furthermore, none of the similar implementations residing on the Internet take advantage of the Java RMI API which must be considered a good starting point for distributed systems learners for it to be taught as part of our degree and is a requirement for this project. Instead, they use different sets of other technologies which can further confuse their reader and introduce a steeper learning curve.

VI. Conclusions and Future Work

References

[1] L. Kotthoff, “LLAMA: leveraging learning to automatically manage algorithms,” arXiv, Tech. Rep. arXiv:1306.1031, Jun. 2013. [Online]. Available: http://arxiv.org/abs/1306.1031

[2] R. Diestel, *Graph Theory, 5th Edition*, ser. Graduate texts in mathematics. Springer-Verlag, 2016, vol. 173.