Ethics, Security, Legal and Social Impact
Colisee by Team A

Russley Shaw rfswf5@mst.edu
Quan Gip qgdgf@mst.edu
Michael LeDoux mmlv2c@mst.edu
Matthew Millerick mjmxd4@mst.edu
Gary Warren grwqk6@mst.edu
Tyler Kaufman tjk43b@mst.edu
Ryan Loeffelman rjlt3c@mst.edu

December 9, 2016

Contents

1 Introduction

Colisee was developed as a future replacement for ACM SIG-Game's *Arena* component for its MegaMinerAI Competition and thus, many types of users will interact with it: directly and indirectly. However, it must be noted that our implementation's privacy and ethical concerns are at the will of SIG-Game itself. Prior to the MegaMinerAI competition, when signing up on their web-site, they include a *terms and conditions* which must be accepted by the user.

Any major ethical concerns found in the Arena service itself, would have a small impact considering that the MegaMinerAI competition is a 24 hour competition and is limited to 100 to 200 individuals. This is very small on a realistic scale and any major issues can be handled personally and individually.

2 Possible Concerns

2.1 Externally Owned Code

The nature of the SIG-Game arena is that it compiles, runs, and stores the MegaMinerAI's competitors code. Often times, sponsors and other companies will join and compete in this tournament also. Some companies do not wish for SIG-Game to store their code longer than the duration of the competition. This resistance against wanting their code saved/distributed could be because of company specific implementation/code, licensing, etc. Our client, SIG-Game, relies a lot on sponsor contributions, so meeting their expectations is a high priority for them.

2.2 Client Privacy

With the population becoming more and more aware of privacy concerns, some users might find themselves worried about the privacy of the Arena implementation. Typically information about the users/teams would live in a database and the SIG-Game Web Server often hosts this information for other competitors to see.

2.3 Scheduling Fairness

It was a considerable internal debate in SIG-Game as to how to schedule individual teams. Possible solutions include scheduling based on ranking/score, other solutions include completely random. This decision was left up to Team-A and our implementation of the new Arena. SIG-Game's competitors often wish to see their AI play against AIs that are similar in skill, yet, when comparing two AIs it is unfair to a poor AI that might just so happen to be the perfect counter to a good AI. There is also the concern of competitors complaining about not playing enough games in total; often due to failing to compile code. Previous implementations would not play any games if a competitor's code failed to build in the Arena building environment.

3 Our Solutions

3.1 Externally Owned Code

One major feature of Colisee over previous Arena implementations is the use of *Docker*, a virtual containerization service. Docker containers can be easily created and destroyed. Because our implementation of the Arena uses Docker for all the individual components, at the end of the competition, the entire arena can be destroyed, thus ensuring SIG-Game's sponsors and other competitors that their developed code will not be distributed by SIG-Game, purposefully or accidentally.

Also because the code lives within containers, it is significantly more cumbersome to access the container file system than the host file system, making it more difficult for a malicious user to access individual's code over previous implementations

3.2 Client Privacy

In our implementation of the Arena, there is no concept of the user. Teams exist relatively anonymously because they are only identified by a unique identifier number and a unique name. The name of a team is not required to be personally identifying and if a user/team was concerned, they would have the ability to make their name whatever they wanted. All data from users is requested rather than collected, and therefore if a competitor does not give use the minimal required information, then they simply choose not to participate in the tournament.

For example, a team named *a932vds30kio* reveals very little about the specific team, where-as *RussShawTeam* could reveal more directly personal information about the users of the team. Also, the nature of the Arena is an administrative program, meaning no non-privileged users will have access to the web interface.

3.3 Scheduling Fairness

Between the two options presented to us (random and ranked scheduling), we decided to implement a random scheduling service. We felt this was fairer in a more absolute sense, where every team has an equal chance to play against another team.

Our service also only replaces the previous build if the current build succeeds. Therefore, as long as a single successful build exists, every team should be able to play games against other competitors. Other elements of the code have been unit tested extensiveness to ensure fairness in the implementation and its usage.

4 Future Concerns

The Colisee project is in an alpha state and the SIG-Game Arena development team is expected to pick this project up and continue development on it. New ethical issues may be introduced. Some possibilities for future concerns may include adding specific user information and deanonymizing teams. Also previous SIG-Game competitions allow signing in via Google Account information, so there is the potential for the Arena's database to inherit that information as well. This may be against the wishes of the competitor.

More issues arise when considering implementation of tournament scheduling algorithms. SIG-Game will need to determine if a triple elimination implementation is more fair and suiting than a swiss scheduling implementation for final tournament results. Beyond the simple choice of algorithm, is the verification of implementation. SIG-Game has had problems in the past with the implementation of scheduling algorithms that lead to dissatisfied competitors. This could be resolved by intense verification of algorithms through the use of unit testing.

Finally the issue of a transition to use in MST's Computer Science Introduction to AI's Chess Tournament will eventually become an issue. This will be a major ethical concern spike, as rather than being a client for simply SIG-Game, our implementation will eventually become a product used by another class, full of tuition paying students. Previous implementations of the Arena have struggled with performing in this environment and judging when Colisee crosses the threshold into becoming usable for this use case is a matter of judgment on the part of SIG-Game's Arena development team.

5 Conclusion

It is our held belief at Team-A that our software introduces no major ethical concerns. Any minor ethical concerns can either be realized as design decisions or items in SIG-Game's terms and conditions for entry to MegaMinerAI.

This lack of ethical concern is due in part to its state of completion. This product is not a professional product yet, as the purpose was not to have a turn-key solution. SIG-Game is responsible for the final professionalism of the product. The goal for this project was to begin development on the new Arena component and to eventually reach a state it could be handed off to future development teams. This project laid the ground work for future development and although issues may lie ahead, these are issues that SIG-Game will need to resolve, rather than Team-A.