# Recommendations for Commercial Implementation

Were this project to be implemented commercially, several areas of the project management, program design and product itself would likely need to be improved. Firstly, the accountability in terms of development would need to be improved and visibly better. During the project, little auditing was done on code submissions, and it was very possible for people to modify code, upload it to the project and nobody notice this has happened, which may have led to problems. In a commercial implementation, a clearly defined process of submission, testing and approval would need to be considered to ensure that no unwanted or dangerous changed are committed to the main build. Specifically, somebody other than the developer of the code in question would need to be responsible for approving code for submission into the main build.

Secondly, better documentation of the application’s design and implementation would be needed. Many aspects of the design were never detailed anywhere, such as class interactions, state transitions and general program design. This often left other team members having to work out and guess for themselves what other people’s work did. In a commercial application, the client would fully expect suitable documentation to exist when work is done for them, which was not the case with our product. While we did have a general outline for user-interface design, the class interactions behind it were never discussed and this potentially impacted on the final quality of the application.

Next, the final sprint where we intended to implement multiplayer had to be dropped three-quarters of the way through. This was due to the discovery that the multiplayer aspect that we’d agreed on implementing, in the design we had proposed, didn’t appear to be feasible in practice. The absence of a feasibility study was particularly noticeable in this sprint, with the final build suffering significantly with the absence of what was considered to be the most important feature. While a work-around was developed to ensure we still met the brief, this fell far below what we had set out to achieve with our application and ultimately could have been easily avoided. This part of the Agile paradigm would have be a significant help to reduce the risk of this issue occurring again were a commercial implementation of our game created.

Finally, in a commercial implementation for our product, much more thorough testing would be required. As much of the code was done in individual units and incrementally added, and no clear testing process was defined, we ended up in a situation where only individual part of the code had been tested. However, the whole of these individual parts together was never tested nearly as thoroughly, and this created a major risk of significant bugs going into the final build unnoticed. In a commercial application, a clear testing process would need to be set out and adhered to before each build, with particular attention given to the interaction between individual units and less emphasis on the individual units themselves.