Evaluation and Testing

Our final testing report can be found https://teamfractal.github.io/assessment4/testreport4.pdf, and covers both evaluation (as requirements acceptance testing) and testing.

Evaluation Methodology

Our methodology for determining whether our final product met the brief consists of checking the product against the requirements document. This document was initially pulled directly from the brief and an interview with the customer, and has been revised throughout the process by *DRTN*, and us to accommodate the requirements change in assessment 4. This means that we are confident that it completely and comprehensively covers the brief. For our product to be acceptable in line with the brief each and every requirement needs to have been fully met - there can be no partially met requirements. We continued to use *DRTN*'s methodology for doing this whereby each team member checks the finished product to ensure all the requirements are met, as we believe it to be a good way to make absolutely sure that all of the requirements are met. We used the spreadsheet provided by *DRTN* as a template for doing this. The evidence of this can be found at https://teamfractal.github.io/assessment4/reqtest4.html

Testing

For the purposes of testing our code, we defined “appropriate quality” as follows:

> The code should provide all the functionality mentioned in the brief, provide an easy to use and intuitive GUI and should never crash (e.g. on an uncaught exception).

We determined that this is a good definition of appropriate quality because it ensures that the finished product will be easily playable, and won’t have any unexpected behaviours. This is important as it should be easy for someone to pick up and play, and unexpected behaviours, including crashing, are a barrier to this.

To determine whether our finished code met this quality, we performed several different types of tests. We continued to use the testing methodology that we inherited from *Duck Related Team Name* as the product had been thoroughly tested throughout development to release ready standards, meeting this appropriate quality.

For our testing report (linked above), we reworded some of the descriptions, and removed some of the historical details about why the tests were needed in their current form. However, the methodology remains the same. For traceability, the original report is linked from ours. Accommodating the Assessment 4 changes was easy as the methodology we inherited was well thought out. The changes we made are as follows and are highlighted in the document:

Manual Testing - RoboticonQuest - PlayerTest():

Changed to include testing for up to 4 players.

GUI Tests:
Tests for capture the chancellor and player selection added.

All of these changes are highlighted in blue in the document.

All of the tests we ran passed, so we are confident that our code is of appropriate quality.

Requirements

The latest requirements can be found at https://teamfractal.github.io/assessment4/requirements4.pdf. Our product meets all the requirements as evidenced above in the evaluation section.