

SEPR Assessment 2 Feedback

Team Name: Duck related...

Implementation [15/20] Report [4/5] Code [11/15]	<p>Marks lost: the code is a good start, but would benefit from a further refactoring. For example, the main java file is large - it could easily be simplified by separating out the code for the GUI from the rest of the game logic. Additionally, more consideration could be given to maintainability. For example, encoding the three types of resource in a fixed-length array doesn't particularly lend itself to easy extension (suppose we ask you to add a new type of resource in assessment 4 - you may have to manually adjust all code using the array later, e.g., adding a new statement to a switch/case). Using a Factory pattern would be preferable.</p> <p>In the report, I was unclear as to the use of the term "missing" with respect to certain requirements: it's not really missing functionality but "not implemented" or "deprecated".</p> <p>Marks gained: commenting and documentation is clear and helpful in most places.</p> <p>The report is quite well done, with a clear and useful format and lots of justification. The tracing to requirements is well done for the most part.</p>
Architecture [16/22] Concrete [8/12] Justification [8/10]	<p>Marks gained: Clear link to diagram. Clear, succinct explanation of notation. You explain the target platform clearly. Changes are reported systematically, with good traceability to requirements and implementation platform.</p> <p>Marks lost: You do not explain the semantics of the dotted arrows in the diagram, or the <<Java>> stereotype. Your reporting is clear and concise, but does not give the bigger picture: HOW you got to this model from the abstract one. It is unusual to find a concrete architecture which has only added/deleted features of classes. If this is because your abstract architecture (which is not linked) was quite concrete already, you still need to comment on how the concrete builds on the abstract in this report.</p>
Testing [16/20] Summary [4/5]	<p>Marks gained:</p>

<p>Report [4/5] Evidence [8/10]</p>	<p>Extremely clear overview provided of unit testing - both of what test you have run, and why this testing strategy was appropriate to ensuring that requirements have been met.</p> <p>You have done a particularly nice job in evidencing how your tests helped to ensure that you were meeting requirements.</p> <p>Marks lost:</p> <p>Justification of testing design could have been improved through more discussion / *some* reference to the literature (citations, please!). For instance, you write that 'dedicated integration tests ... were deemed unnecessary as the system testing is the next logical step'. Why?</p> <p>Furthermore, in contrast to your excellent discussion of unit testing, some of the other kinds of testing that you conducted is strangely lacking in detail. For instance your discussion of system testing and usability testing needs fleshing out in order for another team to be able to replicate what you did! This is particularly highlighted in the evidence that you provide - for instance, you write that you tested to make sure that text was 'a suitable size'. What does this mean? There is no way for another group to adequately replicate this test!</p>
<p>Updated Docs [13/20] Requirements [4/6] Methods/plans [4/7] Risks [5/7]</p>	<p>Marks lost:</p> <p>Requirements: Report: could be clearer what sorts of reasons existed for requirements were changed e.g. previous statement overspecified; changes in understanding or scope (e.g. for assessment 2 specifically)</p> <p>Method: Web: the report states that part of the original report has been removed, but there is nothing in the document itself to show this. You can, for instance, highlight in red stuff that is no longer included, or use standard mark-up (e.g. in word or googledoc). Report: you could have spent more time explaining rationale and justification for changes, and less on details that are covered in the online documentation.</p> <p>Risks: Report: you give a systematic summary of changes but no overall sense of why things were changed - for instance, I imagine that many of these changes reflected suggestions/feedback from the first assessment, to give a better overall risk document: this would have been a useful observation (e.g. the following changes were made following feedback, and improve the presentation and manageability of the risk document). For the actual risk changes, again, some</p>

	<p>clear rationale for the changes is needed, alongside descriptions of what was done. It's unclear if anything changed in the light of your having encountered a risk, for instance.</p> <p>Marks gained: Requirements: Web: clear highlighting of changed requirements. Report: Clear factual commentary on changed requirements.</p> <p>Method: Web: clearly highlighted changes. Report: identifies main changes to tools and methods, as well as justifying some changes to the plan.</p> <p>Risks: Web: clearly highlighted changes to the deliverable. Systematic summary of changes.</p>
GUI report [3.5/5]	<p>Marks lost: the report is imprecise in several places. For example, what is meant by "usability" and "playability"? We're not overly concerned by how you define and use these terms (though you should do a little research) but we do want you to use the terms precisely and consistently. Interaction design <i>is</i> discussed but only in a few places; it would be better to discuss this consistently throughout the document. Use of green/red is a bad idea (if someone is colour blind, they're mostly likely unable to distinguish these colours).</p> <p>Marks gained: there is good discussion and explanations of the GUI layout and, overall, the general design decisions.</p>
Website [3/3]	<p>Comments Website is clean and tidy; absolutely fine.</p>