Updated Requirements

Due to the brief we were given in assessment 1 not changing moving onto assessment 2, there was little to change in the requirements specification. We also had no communication from the client regarding a change in requirements or a growth in the scope of the project.

From the feedback we gained from assessment 1, we decided that the layout of the requirements system should remain the same as there was no downside to the system layout and modifying the layout is an unnecessary task. We also felt that the tabular layout definitely prevented our team from restricting the requirements too much, which could have caused the requirements to be difficult to implement if restricted too much. It also limited how much we could discuss the pass criteria and discussion of risk which allows for these areas to be clearer and more concise, avoiding any ambiguity when it comes to fulfilment of the fit criteria. The only thing we added to the presentation is a very short overview of what approach it is we are using for the requirements.

Most of the changes made involved rethinking some of our requirements making them less ambiguous. This allows the team to know when the requirement has been fulfilled. We also went about separating multiple requirements bundled up into individual requirements so that the pass criteria are clearer to understand and the requirements are less ambiguous to all team members. This also allows the team to know when the requirement has been fulfilled. Due to the separation and addition of new requirements, the referencing system may differ slightly from the previous document, however, these changes clearly shown in the updated requirements document http://limewire.me/docs//assessment2/Req2.pdf. This keeps the trace of changes obvious.

The main issue we could find with our current set of requirements was the ability to test some of the requirements. In particular, requirement **2.7** (referenced from Req1) which we found was difficult to test as different group members may have different opinions on what is deemed as ‘aeshetically pleasing’.

The requirements we broke down into multiple requirements were **1.4** and **1.7** (referencing from Req1). We felt that it would be much easier to test these requirements when separated by giving them more individual pass criteria. This was our main motivation for splitting these requirements up. Having these requirements broken down will definitely help the development process as it naturally makes satisfying them a more iterative process.

We have also added a new requirement (1.14). We felt that this was definitely a feature that we looked past and that it would be impactful enough to include in the requirements list. We feel that the requirement will affect gameplay as it allows for clearer decision making when the player is taking a turn and as of now, there is no risk associated with this requirement.

The updated requirements for assessment 2 can be viewed here:
http://limewire.me/docs//assessment2/Req2.pdf

Update to Method and Plans

As it worked well for us during the first assessment we will be continuing to use the scrum software development technique. However due to the fact that a large amount of time was over Christmas and the team couldn't regularly meet we decided to have one large sprint over the holidays and one small sprint to finish everything off back during term time.

We are still continuing to use google drive for any documentation that needs sharing between the group and then a GitHub to hold the main program as well as branches where team members can share what they have been working on. The issue we found with git, however, is that it is not very collaborative especially not in real time and struggles to keep up with the google drive. To counter this we used a tool called Atom. This is a collaborative programming software which allows users to work on the same piece of code in parallel to each other. We chose Atom because it has embedded git control. This is discussed in the updated method selection and planning document on the webpage. We also still regularly use Facebook messenger to keep in touch amongst the group and still use intelliJ for our primary IDE (however, occasionally use Atom on isolated functions for debugging).

We have however decided that using the task management website monday.com didn't contribute much to keeping the group on time with tasks as people knew the time frames which they were working to and what tasks needed to be done. It was also very inconvenient to keep renewing the free trial that we were using to view the plan and at some points, it would not allow us to edit the document due to not wanting to upgrade to a paid plan. In turn, we are using the Gantt chart to keep on schedule and communicating through the group messenger chat so that we all know what has been done and what is remaining to be done. This also flags up whenever there is a risk of falling behind schedule as we look over the chart during frequent group communication sessions and meetings. We felt that using the gantt chart was better than the previous method as we could print out the gantt chart and give all team members a copy then annotate the chart with which jobs belong to which team member, and update as information came through from the group Messenger and meetings. We felt that the gantt chart breaks down jobs in a logical fashion and gives individuals smaller workloads which are more realistic in the time frame given by the schedule.

For the most part, team roles remain the same as everybody was satisfied with their roles. There was however 2 alterations made. Firstly, Andrew became the architecture designer as the head developer was putting almost all of his time into the code and Andrew was the next person down the line in terms of understanding the code and how the game would need to be built up, as the secondary developer. Because of this, Andrew took on the role of designing the architecture for the code so that he and Ed could being implementing it immediately. The role of the lead tester was then given to Luke because, as he is designing the majority of the requirements, it naturally made sense for him to also run the black box tests based off these requirements. This switch was able to happen due to the flexibility in our team. Upon analysing our feedback from assessment 1, we decided to add an explanation, and justification, of why we planned out the project schedule in the way that we did. We felt that this would give a better idea of the methods we used when deciding the order to complete a task as well as justifying the time frame which we gave these tasks. The plan reflects how we wanted to keep things current in our mind and prevent having to go back and relearn/remember things we did such as tests and debugging weeks after the event. This shows in the plan due to tasks running in parallel in most cases.

The updated plan can be viewed here http://limewire.me/docs//assessment2/Plan2.pdf

Updated Risk Assessment

There were little changes to the risk assessment we submitted in assessment 1 as the same risks were still appropriate for assessment 2. This was largely due to the requirements remaining constant for the project across both assessments. Another contribution to the fact that we didn't want to alter the risk assessment document much from assessment 1 is that all group members clearly understood the layout of the risks as well as how they could be mitigated and if they did occur, how highly they would impact the project. All of the colour coding is consistent with the Risk1 document. The labelling and categorisation of risks was also agreed to be clear and easy to reference so this has also remained the same.

The main alteration we have made to the risk assessment document is creating a column in the table for 'risk ownership'. We found that, despite all of us understanding the possible risks and severity of them, it wasn't initially clear who had to worry about each individual risk. This was an area which was also brought up in our feedback which we reviewed in detail. Despite their now being ownership of each risk, the nature of our team hierarchy still remains reasonably flat so everyone needs to keep all the risks in mind. The ownership does, however, give a most likely owner of each risk allowing a specific member of the group to be able to monitor their specific risk. This should result in fewer risks materialising.

We added two risks to the project risks part of the risk assessment specification both with appropriate mitigation strategies, likelihood and impact. The first risk (**PJ11**) was something we felt was overlooked in the first assessment as we were all present in the same location throughout the whole of assessment 1. It is now something we know needs monitoring over the rest of the project lifetime as the project development lifespan is reasonably high (all university terms). Because the project is planned to continue over the whole academic year, it is highly likely that it will materialise, hence its likelihood being labelled high. The second risk added (**PJ12**) was also something we didn't consider enough in assessment despite some discussions taking place about the matter. It is key that when sharing code in future iterations through the GitHub, all the code must be set up in a way which it can be configured quickly on all the team members machines irrespective of what operating system they are using. As we foresee that code will be getting pushed to and pulled from the Git frequently, it's a risk that has a lot of opportunities to materialise, however, assuming that we know the IDE well, this should be easy to mitigate after initial set up.

To keep track of how the risks developed throughout the development process, we have now included a date above items in the 'status' column so that it can be monitored when the status of a risk changes and when it is resolved (changes back to potential). We agreed that anybody can add items into the risk status column, however, it is primarily the job of the risk owner to monitor the risk, therefore, it will more likely be a specific individual altering their respective owned risk status'.

The updated risk assessment document can be viewed here:
http://limewire.me/docs//assessment2/Risk2.pdf