Risk Assessment and Mitigation

[Changes in blue]

It was found that generally, the risk assessment table was very comprehensive, with clear and detailed descriptions of both the risks and mitigations. Some changes have been made to the risk assessment table, to increase the clarity of each risk.

The old “Likelihood/Severity” column has been split into two columns. When reading the document we found that the single column could cause confusion, for example: #8 had the Likelihood/Severity entry “high”, which could imply that it was very likely to happen. This is clearly not the case, so changes needed to be made.

Some changes have been made in the “Description of Risk” column. For most risks, the description was very clear, though in some cases changes had to be made. In these cases either the description had skipped over some details, or the description was ambiguous. No changes were made to the “Mitigation” column. Each mitigation was appropriate and detailed, giving us confidence in the stability of the project as a whole.

At the bottom of the table, change risks have been added. These were added to prepare for any issues that may occur due to the handover process at the beginning of this assessment.

ID	Description of Risk	Likelihood	Severity	Mitigation
Technology Risks				
1	Using a complex programming language to develop the product might be harder for some members of the project team to work with, which may mean unnecessary extra time spent on work.	**MEDIUM**	**LOW**	In order to prevent this from happening, the group should mutually agree to use a language that all members are comfortable with using. The more experienced members of the team should help the less experienced with understanding the basics of the programming language.
2	Implementation of the specification may be difficult with the programming language chosen. This will mean time will have to be spent finding and adjusting to a different language.	**LOW**	**HIGH**	In order to avoid this happening, the language chosen should be able to implement all of the points in the specification.
3	If the hardware and software are prone to bugs and are “slow”, they may be inappropriate for developing a system. This can affect development, especially during the coding and testing stages.	**LOW**	**MEDIUM**	In order to reduce the time wasted on finding bugs, the code should be kept as simple and clean as possible, (i.e easy to understand, change and test) with good documentation of code. We should use existing libraries where possible as well as keep the code modularised. The hardware used should at least be fast enough to develop the game in.
4	If the architecture is too complex to be implemented, the architecture design will have to be simplified or modified. This will mean time wasted on re-designing.	**MEDIUM**	**MEDIUM**	In order to prevent this from happening, the architecture should be designed so that it follows the standard format, so that all team members are familiar with it.
People Risks				
5	If the skill level of the team members are not sufficient enough for the development of the project, this could mean either a product that is not of the quality expected by the client, or a product that won’t be met by the deadline due to said members learning the required skills whilst developing the product.	**LOW**	**MEDIUM**	In order to prevent this from happening, the development team will be required to learn the languages and techniques necessary to develop the game. If any members of the team are particularly experienced with a facet of the development tools, they should instruct those who are not as well versed. Also to avoid the product not being finished, individual deadlines should be set to check if each member is keeping up with their work, if not then other team members can help.
6	Unless planned, a missing team member (potentially multiple) from the project will have an	**MEDIUM**	**MEDIUM**	In order to maximise efficiency, the missing member should contact the rest of the team immediately to confirm if

	effect on the schedule of the project because that person is not available to do the work assigned to them.			their absence will affect their work. If so, the missing member's work should be delegated among the team.
7	Disagreements between members may delay project deadlines as time may be needed to agree on something. In the worst case, it may cause members to leave.	LOW	MEDIUM	In order to avoid any delays, if there is a disagreement, it should be brought to the attention of all team members and a decision should be made democratically.
8	If a team member leaves permanently, that will mean their workload has to be shared amongst the remaining team members.	LOW	HIGH	In order to finish the work within the deadline, the work should be delegated. An individual member should not be heavily relied upon.
Tools Risks				
9	Lack of understanding of the support software tools used in the process by team members will slow down development of the product, which will affect deadlines.	MEDIUM	MEDIUM	In order to minimise the amount of wasted time, experienced team members should help the less experienced members understand the tools. Less experienced team members should also seek online tutorials or documentation for tools if needed. Also the more experienced members should work with the others to make sure they understand how to use the tools.
10	Any important updates for the support tools used whilst in development may mean members have to adjust to new interface arrangements or other significant changes in the software.	MEDIUM	LOW	Due to SEPR being such a small project, unless updates are absolutely necessary, all tools should be kept on the same version until the project is finished. If an update is mandatory, team members will have to help each other adjust to any major changes in the software, or seek online help.
11	If any of the support tools fail to function in any way, or do not work as well as anticipated, this will slow down development as a fix will have to be found or a new tool will have to be used.	LOW	MEDIUM	If the performance of a particular tool has come to the attention of most of the team, a replacement tool will have to be found as soon as possible.
Requirements				
12	Changing of requirements may mean the team has to be ready to modify the project to suit the new changes otherwise the wrong product will be delivered.	MEDIUM	LOW	All team members should be prepared and be made aware that changes in requirements are expected. If a requirement changes, the team should discuss what changes to the system must be made accordingly.
13	Requirements that are incomplete, ambiguous or untestable will lead to the wrong product or something the client was not expecting.	MEDIUM	MEDIUM	If a requirement has come to the attention of most of the team due to being ambiguous and unclear, the client should be consulted to either remove it or change it to be more precise.
Estimation Risks				

14	Inaccurate or insufficient planning of the tools required to complete the project could lead to an incomplete product or poor quality outcome, because of wrong estimations in resources.	**LOW**	**MEDIUM**	All the necessary tools and required software should be identified early on in the development process. If any additional tools are needed, the team should be notified and an agreement should be made about whether it is needed.
15	Inaccurate or insufficient planning of the project schedule will lead to missing deadlines because of wrong estimations in time.	**LOW**	**HIGH**	Sufficient time should be given for each piece of work when planning and milestones. Deadlines should not be underestimated and team members should agree with the time allocated for each piece of work.
Process Risks				
16	Development of the project without an appropriate, defined process, could lead to inconsistency in approach, producing a more fragile product.	**LOW**	**MEDIUM**	A suitable, appropriate development process should be mutually agreed upon by all team members.
17	Team members may not follow the desired process, either because they prefer not to or they don't know how to. This can mean it will be harder to coordinate which members are doing what in the team.	**LOW**	**MEDIUM**	Team members should follow the agreed process and try to do the work assigned to them. If team members do not know the principles of the process, they should ask more experienced team members or seek information online.
Change Risks				
18	Without proper investigation, the chosen codebase could have issues that prevent the game from running.	**MEDIUM**	**MEDIUM**	The team could download the potential team's source code and run it before choosing the codebase to work with.
19	The code could have unforeseen issues that make it difficult to extend the game without major refactoring of the code.	**MEDIUM**	**HIGH**	If the other team offers technical assistance, make full use of it early in development to understand what measures they took to allow extensibility.
20	The code style of the previous team is different to ours.	**LOW**	**LOW**	Read through the code initially and understand the pattern and code style decisions they make before making changes.