UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

Risk Assessment and Mitigation
Cohort 2 - Group 16 (Skloch)

Group Members:

Charlotte MacDonald
Hollie Shackley
Luis Benito
Kaustav Das
Sam Hartley
Owen Gilmore

Originally by: Noah Forryan, Lewis Morgan, Naufal Tun Thamanian, Tom Owen, Dan Manby, José Fernandes of Group 19

We divided the risk assessment process into identification, analysis, planning and maintenance.

Identification
For this step we felt it was important that any risk a team member thought of should be recorded, regardless of its severity or likelihood. So we came together and wrote any and all ideas down into a Google sheet. This process was very important as it created an open atmosphere where team members did not feel pressured to only write down “correct” ideas. Because of this, we were able to generate more ideas that could then be assessed and potentially mitigated, thus making our project better covered overall.

At the end of this stage we had a long list of risks, some of which would need to be mitigated and others that were minor enough in likelihood or severity that they did not need so much attention. However, at that time about the categories they belonged to.

Analysis
This was the step where we examined each risk to determine how much mitigation it would need. We understood that a key principle in risk management is that one cannot cover every possible risk. We knew that we would have to divide our time between the risks. The first step for this was to come up with a system to accurately show a risk’s likelihood and severity and then combine them into a “risk score”. Originally, we assigned a risk a 1-5 score for likelihood and severity and then summed those two to get this. However, we felt that this overplayed the danger of risks that only scored highly in one of the two metrics. To fix this, we multiplied the two scores together. Therefore, a risk with low likelihood and high severity would score higher than a risk with moderate likelihood and severity.

Once we had worked as a group to assign scores to each risk we had a clear picture of which ones we needed to focus on. Anything with a risk score less than 6 was deemed safe enough to ignore, allowing us to focus on the more dangerous risks. This is because a score of less than 6 means that at least one of likelihood and risk are low, so it is not a significant concern. To further help with this, we assigned a colour to each score. We used the RAG system as it is an intuitive method to show levels of danger, at a glance.

Planning
Now that we knew which risks to focus on we began looking at how to mitigate them. The first step for this was to assign ownership to each risk. This was decided based on the specialism within the team. If a risk related to coding the game it was assigned to Tom and Dan, if it was risk or requirements based it was assigned to Noah and Lewis, if it was asset related it was assigned to Naufal and Jose. It was important that people took ownership of the risks to do with their specialism as they have greater knowledge in that area to be able to avert the risk. Each risk had two owners, in case one was unavailable. By building redundancy into the risk mitigation itself we further secured our project’s success. From here, the pairs of owners developed a way to mitigate the risks assigned to them.

Maintenance
We made another table to show how the risks have changed over the course of the project. During every group meeting, we discussed changing risks. These were changed in the updated

risk assessment. We did this for optimal risk assessment accuracy.

Risk Assessment Format

The format of our risk assessment is: ID (the name of the risk, written in a way that is easily understandable), Type (the area the risk impacts), Likelihood, Severity and Risk Scores (as described above), Mitigation (the plan to deal with the risk) and Owners (who is in charge of that risk mitigation). The format of the updated risk register is: ID, reason for change, date.

Updated Risk Register (Changes made by Group 16 [4])

Priority risk matrix:
X = Impact level, Y = Probability level

5	5	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
1	1	2	3	4	5
	1	2	3	4	5

Project Manager	Hollie and Luis
Product Owner	Owen and Sam
Team Leader	Charlotte and Kaustav

ID	Risk class	Risk description	Impact description	Impact Level	Probability Level	Priority Level	Prevention/Mitigation Strategy	Owner	Reassessment Date
1	Project	(Concentration risk) Dependency on a single key team member for a critical task	Delay in task completion, potential project failure	5	3	15	Documentation of critical processes. Each critical task will have more than one person overseeing/contributing to it to avoid having any single point (person) of failure. Conduct regular knowledge-sharing sessions, ensure backups are in place for key roles.	Project manager	Biweekly
2	Project	(Scope Creep) Continuous addition of new features beyond the initial scope of the project.	(Resource drain). The effort would be disproportionate to the marks given for the particular task and would be considered a waste of time and resources.	2	5	10	To make sure the team is not adding or changing features that shouldn't be added or changed. (Change control process). Document all the necessary features that are needed to be added and changes that are requested by the client.	Product owner	Weekly
3	Project	(Communication failure) There has been a conflict between group members and communication has broken down.	Decreased productivity, team morale issues and dysfunctionality between team members	3	2	6	(Mitigation strategy for ID:1) Establish conflict resolution protocols, encourage open communication channels. Conduct regular team check-ins, address conflicts promptly	Team leader	Weekly

4	Project	(Communication failure) Lack of communication causes multiple team members to do the same work	There will be multiple versions of the same work which will need combining/choosing between in a fair way to ensure everyone participates equally. Alongside delayed project timelines	3	3	9	Implement task tracking system, promote regular progress updates. We must establish clear task assignments, and encourage communication between members.	Project manager	Weekly
5	Project	Lack of communication causes a team member to do too much of the remaining work	There has not been equal participation and there is not enough remaining work to make it even.	4	3	12	Have regular workload assessments and promote open communication regarding task allocation. Conduct regular check-ins on workload distribution, provide support for overwhelmed	Team leader	Weekly
6	Project	A team member becomes temporarily absent for a specified period of time.	Delay in task completion and redistribution of workload	4	2	8	Document handover procedures and ensure clear task delegation. Cross-training of team members across various tasks could be the contingency plan, however the risk scales with team size, therefore evaluating the work ethic of each and every member should be feasible. Establish contingency plans for temporary absences UPDATE: Ensure the bus factor is >1, by ensuring multiple people understand each area.	Project manager	As required (during the time of absence)
7	Project	A team member becomes temporarily absent for an unspecified period of time	Uncertainty in task completion, increased workload for remaining team members and decrease in team morale	5	3	15	Regular check-ins with absent team member(s), distribute workload among remaining team members. Also (Mitigation strategy for ID:8)	Project manager and Team leader	Weekly, until return of absent member is confirmed
8	Project	A team member permanently drops out	We would only have 5 people which may put pressure on the rest of the team due to	5	3	15	Establish contingency plans for permanent drop outs of a single team member and consult module leader. Also, (Mitigation strategy for ID:7)	Project manager and Team leader	When and if it happens.

			increased workload. Also (Impact description for ID:8, 9)						
9	Project	2+ team members permanently drop out	We would only have 4 people which is not considered enough to complete the project Also (Impact description for ID:8, 9)	5	2	10	If some deliverables are dropped, the remaining team will work on the new, reduced deliverables. If all are dropped, consult module leader. Also, (Mitigation strategy for ID:7 & 10)	Project manager and Team leader	When and if it happens.
10	Project	A team member has been assigned too much work and reports they will be unable to complete the work on time	Delay in task completion.	5	4	20	Regular Evaluation and distribution of the workload according to the skillset of the members. Cross-training of team members across various tasks so that no concentration risk takes place.	Project manager and Team leader	Weekly
11	Project	A team member hasn't completed their work by the deadline and didn't report it	Project delays, compromised task dependencies	3	2	6	Clear reporting protocols, task tracking systems and establish reporting expectations in order to follow up on missed deadlines promptly	Project manager	Weekly (until its no longer a problem)
12	Product and project	There has been a drastic change in requirements	Increased workload, potential delays, scope creep	5	3	15	Robust change control process and immediate impact assessment. Also, negotiate changes with stakeholders.	Product owner	As needed, based on frequency of changes in requirement
13	Product and project	The change in requirements increases the workload by a large amount but the deadlines are not pushed back	Overworked team, Increased workload, compromises in terms of quality (lacklustre product)	5	3	15	Negotiate deadline adjustments and time extensions. Also assess the resource reallocation and workload.	Project manager	Weekly (until it's no longer a problem)
14	Product	Inadequate testing leads to issues with the product	Lacklustre product and customer dissatisfaction	5	2	10	Comprehensive testing strategy alongside an intricate benchmark for quality needs to be established. Also surveys to incorporate feedback.	Product manager	Monthly

15	Product	A tool relied on for a large portion of the project becomes unusable.	A new tool must be found and code rewritten which will require extra resources and increase workload.	4	2	8	Research all tools used extensively to ensure they seem reliable for the foreseeable future. Research alternates so they can be quickly implemented if necessary.	Product manager	Biweekly
16	Product	When changes are made to the code, it works on the changer's device but not on different hardware/software.	This could mean not all members of the team can run the code and develop it. It also means when the game is being marked, the module leaders cannot run the code.	4	2	8	Code should be tested on multiple different types of devices at every major change in the code to ensure it can be run on different devices. If it cannot, we can use github to revert to a previous commit.	Product manager	Weekly
17	Product	When merging two branches there are merge conflicts or similar issues	This means the current version of the code will not run/is incorrect or new code cannot be added to the main branch.	4	4	16	Analyse the code and git log to see how this has occurred and resolve any merge conflicts. If unsure, revert to the last commit.	Product manager	Weekly
18	Product	Poor quality code	Mostly backend work, but this can have serious impact on the grading process and in case future programmers choose to modify/fix/update the code to suit separate hardware.	5	2	10	Implement regular code reviews and pair programming, define a coding style guide, test code regularly (perhaps automatically after every commit). Also, use benchmarks, and various other quality checks to ensure the code is efficient, readable and bug-free as the final project.	Project manager	Weekly