

After having a team meeting, we thought about the various risks that we may face throughout the project and how we were to deal with them before they came up, and how we were to face them if we were to run into one. In the end, we made the decision to create a risk assessment form that consisted of these columns - **ID, Risk and Category, Information and Causes, Likeliness, Severity, Avoidance and Minimisation, and Risk Monitoring.**

To define and explain our columns -

- **ID** - Created so we will be able to easily reference each risk later on our reports.
- **Risk & Category** - Naming the actual risk and placing it in a category where we feel it fits.
- **Information & Causes** - A simple explanation of a given risk so we can apply it to a real-life situation, as well as what causes said risk.
- **Likeliness / Severity** - A ranking system for us to be able to quickly and easily identify the impact of a risk at a glance, and what the chances of it occurring are. 1 being "Very unlikely", 3 being "Fairly likely/Fair impact", and 5 being "Very likely/Severe impact".
- **Avoidance & Minimisation** - How a risk may be avoided entirely, and how we may recover from a risk with minimal losses.
- **Risk Monitoring** - Identify when a risk is most likely to happen, and how frequently we need to reflect on the project to see if we are at risk of a specific risk.

Considering the scale of the project, we felt that these sections were enough for us to be able to accurately identify potential risks without extra, unnecessary details that may confuse us later on in the assessment. On top of this, as we will be using an Agile method, this layout was satisfactory in letting us quickly update our risks as they develop and change with our requirements.

We also agreed that throughout the project, we will host meetings where we reflect back on the progress we have made and how this related back to the risk assessment, giving us the opportunity to monitor the progression of any risks and thus be able to plan ahead for them. Additionally, we will use this time to create a note of new risks that have appeared as the project progressed further.

For a given risk, the assessment needs to be to the point and describe the situation in a simple manner - if the risk is hard to understand then we are only going to be more likely to fall victim to it. In some cases, the columns may contain instructions on how to act in a risk, and what steps we should take to mitigate its effects.

We felt like this layout was a good compromise between being simple and easy to read, while also providing adequate information and thought into how we can respond to whatever situation we may find ourselves in.

We will also have a Risk Manager who will have a general overview of the risks and will keep track of all of the risks, while also hosting sessions where we discuss progress of our risks and if any new risks have appeared. The responsibility of any specific risk will be on the individual who is working in that field, so the requirements risks will be on those who are working on and keeping track of the requirements, people risks should be focused on by everyone in the group for themselves and others, etc.

ID	Risk	Risk Information & Causes	Likeli- - ness	Seve- - rity	Avoidance / Minimisation	Risk Monitoring
	<b>People</b>					
P1	Member missing / unable to work - short term	Someone may not be able to do their share of the workload due to illness and personal reasons. Can lead to issues reaching team goals.	2	3	Give notice if/when and why you're not going to be able to make it. The missing person(s) will then be updated on Messenger on what was discussed/agreed and what they have to do for their workload. Work will be on Drive or Github.	This can come up out of nowhere, so not much can be done, but note that illnesses are more likely during the Winter season, etc.
P2	Member missing - long term	It's possible that someone drops out of the course / University or is unable to attend due to a string of illnesses and personal issues.	1	5	Discuss the matter with the Course Leader, but otherwise, the designs/requirements may need to be simplified and the workload may need to be altered to avoid stressful situations.	Try to discuss how every module is going for each member to see if there are any issues.
P3	Member not doing work outside of lab	A member might only do work in SEPR labs, which can lead to us falling behind on our overall goals.	3	3-4	Discuss the matter with them to see why they aren't doing their work, potentially altering the workload if they're struggling.	Regular checks in meetings to see if assigned work is being done.
P4	Member motivation failing	Members may become worn and tired of the course / project and lose enthusiasm, so work quality and delivery may suffer.	3	2	Avoid hardships by planning ahead, otherwise things like complimenting people on their work and keeping each member's workload varied may help in the long run.	Take note if anything has happened in the group to make them feel like this, reflect how we feel like the project is going regularly.
P5	Friction between Members	Two or more members could have a falling out or argument, potentially leading to work not being done and meeting time being wasted.	2	2	Try to break up any conflicts or seek a compromise as soon as possible. Keep meetings civil, possibly seek advice from module leaders.	Keep note if anything is slowly building up over time. Discuss in meetings.
	<b>Tools &amp; Technology</b>					

T1	Computer / Software crashes	A random crash may disrupt a member's workflow, while also potentially losing data / work in the process.	2	1-3	Back up regularly and push work regularly and keep everything on Drive/Github. Software like Word can also save work from crashes, etc.	Completely random, but some software may be known for crashing.
T2	Member not having correct software or tool to do work with	A member may not have access to a PC at home, or have the specs or space to download and run the tools we use to create the game, so they will not be able to do their work at home.	2	3	Discuss this beforehand to see if everyone can work in their own time at home. Otherwise, they may need to do this all in Software labs, and thus their workload may need to be specifically tailored for this.	Discuss at meetings to see if everyone can run any new software and if anyone will be unable to get to a computer system for a period of time.
T3	Difficulty in creating or obtaining assets for game	No one in the team is an artist or musician, but we will need art and music assets for the game. This means we may struggle to create them ourselves, or struggle to find free non-copyrighted assets to use.	4	2	The group shouldn't be awfully picky on the art or music we create, and members should start to research websites where such assets can be found in advance.	This will come up around the middle of the game's development, monitoring should start right before development begins.
T4	Code being of a poor standard	The code is poorly written and has no / limited comments, making additions and edits difficult for members.	2	3	Remind programmers to keep it up to an acceptable standard, have several people review the code before it's moved to the main branch.	Discuss code quality all the way through its development in meetings.
T5	Code's version control not being kept up	A member may not use Git as intended, leading to copies of the main code being created, all containing different levels of completed work.	2	4	Make sure everyone knows how Git and Github works, and say in Messenger if you have updated something that others need to be aware of.	Discuss the code's version control all the way through its development in meetings.
	<b>Requirements</b>					
R1	Changes in Requirements	The team may receive new requirements after a meeting with the stakeholders, meaning that we will have new tasks to	5	2	Respond quickly by understanding what the new requirement entails and make a clear plan in how to implement it into our project. Keep a	Discuss any new requirements in meetings and how they affect old

		focus on, while older requirements may no longer be relevant.			clear plan of all requirements.	requirements regularly. Someone may be in charge of requirements.
R2	Creative differences between members	In discussing ideas, two or more members may have clashing ideas where neither one wants to back down, causing a stalemate where the idea discussion halts.	4	2	A large portion of this will be finding a compromise, which may be suggested by another person in the group. Don't forget the KISS principle.	This can come up throughout the entire development, so discuss ideas in meetings and find an agreement.
R3	Unacceptable quality of work created	Work, notably reports and code, may be created at a standard that's unacceptable and thus will have to be fixed or completely remade by another member.	2	5	Review and read through each piece of work submitted by other members to see if it's of acceptable quality. If it's of poor quality, discuss the matter with them and the rest of the team.	Discuss this regularly whenever a piece of work is handed in.
R4	Non-requirements being focused	One or more members may focus on implementing something that isn't necessary and isn't a requirement, wasting time and effort while we still need to finish off compulsory work.	2	4	Make sure everyone understand what the requirements are and what they have to implement. If each member and what they're doing is kept track of, this can be detected early and fixed.	Someone with a specific role, potentially a leader role, may need to keep track of what work is being done by everyone else.
R5	Running out of time	As there's a lot of work to do, it's possible that we could run out of time and have to submit unfinished work, or cram work at the end.	3	5	Using the Scrum method, split up work into pieces and make sure we achieve them in good time in our sprints. Keep track of what's on our current assessment, when it's due, and what's on the next one, in case we need a headstart.	Keep track of deadlines and the work that was finished/not finished on time. May need to assign this risk to specific role.
R6	Requirement being a lot more difficult to achieve	We could be overambitious and take on requirements that are too difficult for us to implement, meaning that we will have to change requirements with the stakeholders.	3	3	This is more difficult to gauge and will be discussed and agreed on in meetings. Arrange meetings with stakeholders or discuss the situation with a lecturer if we are unable to find a way around the requirement.	This will come up throughout the development of the program, discuss progression in regular meetings.