Risk Assessment and Mitigation

Galin Dzhumakov

Aisyah Firoz Khan

Faran Lane

Samuel Nicholson

Jack Polson

Alana Witten

5. Risk assessment and Mitigation

Part a)

Before blindly coding the project, it is vital to consider the risks that are involved in the project as a whole. Ensuring that each risk has a mitigation plan in place to reduce the effect on the project and avoid problems occurring in the future. Identifying the risks that are involved with the project was the first thing on our agenda. First and foremost, we conducted some research [1] into areas that we should consider when creating potential risks. The article highlighted key points to consider when generating risks, since the team has limited experience with software engineering risks. Initially, we had a discussion as a team to construct a list of all potential risks. As such, each team member was given the opportunity to voice their opinions, allowing a comprehensive list to be developed. Unfortunately, with this approach, there may be risks that get included that are redundant. To eliminate this danger, we reviewed the list and removed any such risks. For example, there was a risk discussed where a team member didn't complete their work, but this is quite vague since it depends on the reason, thus making it hard to create a mitigation plan.

Following the list of risks, we had to analyse each one to determine its likelihood and severity. We each determined values independently for each risk, calculating averages for final numbers. Importantly, each risk needed a mitigation plan to limit its impact on the overall project. Completing this step required the team to meet and collaboratively determine what the best strategy for the risk would be. Consequently, it allowed members to look at the problem from different perspectives, ensuring that the members are critically evaluating the plans.

Following the development of the risks, we would need to correctly and effectively format the risk register so it's easily accessible and readable. It is critical that the risk ID naming be consistent throughout the project in order for risks to be easily identified. Through research it appears there sometimes can be the need for a risk breakdown structure, as a group we believe this is not a necessity for our project. This is because, given the nature of the project, there are not an overwhelming number of risks, so we do not need a breakdown. In fact, having a risk breakdown structure may overcomplicate the risk register.

Part b)

Project

Risk	Risk ID	Likeli-hood (L/M/H)	Severity (L/M/H)	Mitigation	Status	Owner
A group member misunderstands their task and completes the wrong thing	R_PROJECT_01	M	M	We will meet regularly and stay in contact through messaging to ensure we all are clear on what we are doing, and if a different task gets completed, we just mark that one as done.	Not occurred	Everyone
A member leaves the project	R_PROJECT_02	L	M	If this should happen, we will be in contact with the lecturers and module leader.	Not occurred	Everyone
The network fails before a member has committed their work and therefore they cannot keep up to date with the remote code which makes it challenging to merge their changes in	R_PROJECT_03	M	M	We make sure we commit often and try to keep pull requests a reasonable size, not too big. But if this occurs, then we can review the merge conflicts together and work out the best way to resolve it. We will also try to make sure that our pieces of work are self contained and have minimal areas that may have merge conflicts.	Not Occurred	Everyone
A member becomes unavailable for an extended period of time and cannot communicate this	R_PROJECT_04	M	M	Regular communication should mean that we all know where we are at, and if someone becomes unresponsive, we will notice this. Attempts will be made to contact the member using alternative forms of communication, and if that does not work we will let the lecturers and module leader know,	Not Occurred	Everyone

Risk	Risk ID	Likeli-hood (L/M/H)	Severity (L/M/H)	Mitigation	Status	Owner
				while distributing up the work between the rest of us.		
A member is overworked and becomes burnt out	R_PROJECT_05	M	M	Our regular meetings should help ensure that work is assigned evenly. We will also try to make sure that there is a second person assigned as backup, so that if people don't do their main task, there is still someone who can do it and it does not all fall on one person.	Not Occurred	Everyone
We misjudge the time it takes to complete the project, and do not finish it before the deadline	R_PROJECT_06	M	H	We make a schedule of when certain pieces of work should be done so we have a rough way to measure progress. Then throughout the project with regular meetings we can see what parts are on schedule and what we need to catch up on. There is also buffer time at the end to help mitigate this risk.	Not Occurred	Jack
A task relying on the completion of another task before it can commence	R_PROJECT_06	M	M	Having an awareness of dependencies between tasks as well as communication between team members as to when tasks have been completed. Sticking to set due dates of particular tasks will also help to mitigate this risk.	Occurred – 17/12/22 Solved – 17/12/22	Faran
Two or more members dropping out	R_PROJECT_07	L	H	Communicate with module lead, discuss requirements changes and distribute new workload evenly	Not Occurred	Everyone

Risk	Risk ID	Likeli-hood (L/M/H)	Severity (L/M/H)	Mitigation	Status	Owner
Inability to communicate with previous developers that have worked on the project	R_PROJECT_08	M	M	Try numerous attempts to make contact with the other team, trying to communicate with different members. If unsuccessful, contact module leaders to inform them and ask for guidance.	Not Occurred	Alana
The client of our game is unavailable to contact	R_PROJECT_09	M	M	Make numerous attempts to contact our client, allowing reasonable time for a reply before contacting another stakeholder.	Not Occurred	Jack
The deliverables worked on previously by different developers are inaccessible	R_PROJECT_10	L	H	Contact previous developers in an attempt to obtain accessible versions of the deliverables.	Not Occurred	Alana

Product

Risk	Risk ID	Likeli-hood (1-5)	Severity (1-5)	Mitigation	Status	Owner
A member does not understand the use or purpose of a variable (e.g. confusing variable names)	R_PRODUCT_01	M	M	When working on code, we will be working on separate branches, and when we are done with that task, we will merge using pull requests. These pull requests can then be reviewed by someone else on the team, and that should catch confusing variable names. Though if something is still confusing, we can message the group or the person who worked on that part previously to ask for clarification.	Not Occurred	Everyone
Assets that are being used are not allowed to be included in open source projects	R_PRODUCT_02	L	M	Before using any assets we will check the licence for usage rights. If that is clear and says that we can, we will use it. If it says we cannot use it, then we will not. If it is unclear, then we can either look for something else, or we will get in contact with the asset creator to ask for permission.	Not Occurred	Galin
A library being used becomes unavailable or deprecated in the middle of the project	R_PRODUCT_03	L	H	We will try to choose popular open source libraries that have a large community around them to help avoid this risk. However, should this still happen, we can check to see if the last released version is sufficient for our use case, which it should be, or we have a look for alternatives that would also work.	Not Occurred	Faran
Files get corrupted	R_PRODUCT_04	L	H	Making use of version control systems to access backups to restore any	Not Occurred	Galin

Risk	Risk ID	Likeli-hood (1-5)	Severity (1-5)	Mitigation	Status	Owner
				corrupted files.		
Misunderstanding pre-written code/poorly written code	R_PRODUCT_05	M	M	Working through code, using javadoc, comments and unit tests to understand the intended purpose. If still unclear, go through the GitHub repository finding who wrote the code and contact the developer directly.	Not Occurred	Faran
Creating and using unsatisfiable unit tests	R_PRODUCT_06	M	M	Only creating relevant and appropriate unit tests with the intention of testing every feature of the feature.	Not Occurred	Sam
Bugs within obtained code	R_PRODUCT_07	M	M	Detect bugs within code through the use of unit tests and mitigate the risk of them affecting the game by removing them and replacing them with good code.	Not Occurred	Faran
Game/Features of the game are incomplete or missing	R_PRODUCT_08	M	M	Detect through code, documentation and testing any features of the game that are potentially incomplete and decipher how to implement these elements.	Not Occurred	Everyone
Having the hardware compatibility to run the endless mode	R_PRODUCT_09	L	H	Implementing guards that stop the game from crashing and having an end condition after a finite time where the user is idle and not interacting to prevent the game from running forever.	Not Occurred	Everyone
The user being unable to access and change the json file	R_PRODUCT_10	L	M	Ensure the json file is in a safe state before accessed and that the user is given the permission to read/write this file when the game begins in order to save their result.	Not Occurred	Galin