

Risk Assessment and Mitigation

"Mathochist Studios" Cohort 4, Team 11

Euan Cottam
Charlie Thoo-Tinsley
Harri Thorman
Will King
Zach Moussallati
Aiden Turner
Marcus Williamson
Joshua Zacek

Approach

Our group adopted the use of a risk register using Software Engineering^[1] as a guideline which was applied to our risk management strategy. Risks were identified collaboratively during the requirements and plans to mitigate these risks were reviewed at our bi-weekly meetings. At each of the meetings we would evaluate each risk based on the likelihood and impact using a simplified 3-point scale: Low, Medium and High and help to see which risks needed to be prioritised.

The justification for our risk management plan was to accurately: *identify, analyse, mitigate, monitor and dedicate ownership of each* unique risk that happened during the development of our project.

Process

1. Identification - Thinking of different types of risks that may come up in the project as well as adding new risks which appeared throughout the project.
2. Analysis - What is the likelihood of the occurrence and how it will impact the project.
3. Mitigation - What can be done to minimise the impact of the risk and help prevent this happening again in the future.
4. Monitoring - Bi-weekly review of the risk register and adding new risks if any were identified.
5. Ownership - Each risk was assigned to a team / team member to oversee.

After this process was completed, the risks were continued to be monitored by each unique owner and discussed at the bi-weekly meetings to see how progress was being made on each of the risks.

Risk Register

1. Risk ID - A numerical ID given to each risk which helps to track risks throughout the project.
2. Type - What type of risk it is. It is split into three distinct categories: Business, Product and Project.
3. Description - A short write up of what each risk is about.
4. Likelihood - The chances of one of these risks happening or recurring throughout the project.
5. Severity - How much of an impact can be done each risk will make to the project.
6. Mitigation - Solutions that can be done to either prevent the risk happening again or minimise the effect.
7. Owner - Who will be responsible for each risk.

Risk ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	Product and Project	Version control conflicts when multiple members edit at the same time	L	M	Effective communication amongst the group to prevent conflicting versions	Aiden

R2	Project	Members absent for various reasons	H	H	Reassignment of tasks and roles + work to be done on a shared drive to track progress	Euan
R3	Project	Underestimating time required for key and core game mechanics	H	H	Setting weekly specific goals, progress is discussed at meetings	Josh
R4	Project	Assets were all made by the group not sourced which took a while	H	L	Will took the lead and his other initial workload was disseminated	Will
R5	Project	JAR does not run consistently across Windows / Mac / Linux	L	H	Avoided OS specific dependencies	Aiden
R6	Business	Misinterpreting client expectations for events/difficulty amongst the group	L	M	Documented the client meeting and clarified with each other	Charlie
R7	Project	Poor team communication resulting in missed meetings and unclear tasks	M	M	Bi-weekly set meetings + table showing what everyone's tasks and roles are	Josh
R8	Product	Gameplay bugs affect core mechanics	H	M	Manual weekly testing done by all of the group	Marcus
R9	Product	Low code quality / inconsistent style	M	M	Coding style standardised and reviewed weekly	Marcus

R10	Product	Lack of early user feedback before handover	L	M	Early testing amongst friends and peers, feedback documented	Charlie
R11	Project	Risks were not identified at the start of the project	M	M	Risks recorded and identified throughout the project	Harri
R12	Product	Lack of prior experience with LibGDX	H	L	Those with experience took the lead	Aiden
R13	Product	Performance issues across different hardware	H	L	Tested across multiple different hardware	Marcus
R14	Project	Documentation inconsistent with the code (Not updated UML Diagrams)	M	H	Documentation done in tandem with implementation	Zach
R15	Project	Game files lost due to corruption	L	H	Regular copies of the game saved	Aiden
R16	Project	Members falling behind due to prioritising other academic workload / other commitments	H	L	Good communication and workload shared accordingly	Josh
R17	Product	Dependencies and 3rd party libraries become incompatible / unavailable	M	H	Alternative options considered and downloading copies of the dependencies	Aiden

Bibliography

[1] Sommerville, Ian, "Software Engineering," Pearson Education, 2015