

Risk Assessment and Mitigation

Group: Group 10

Names:
Haaris Altaf
Casper Co
Will Garavelli
Kamso Osuji
Krishna Rajamannar
Diyar Savda
Sam Storrs

Risk Assessment and Mitigation

Risk management involves assessing potential risks and creating mitigation plans to minimize their impact. This is essential for any project, as it helps the team prioritize tasks and work with less stress. Risks can change over time, so having a live risk register ensures that updates and responses can be made promptly. For our project, we follow a four-stage process: Identify, Analyse, Plan, Monitor. This iterative approach allows each risk to be continually assessed and controlled, with all team members actively participating in monitoring and reporting changes. Since ENG1 is a small, non-critical software project, we keep our likelihood and impact measures simple, using a straightforward Low/Medium/High scale. This ensures risks are easy to evaluate and respond to without overcomplicating the process. This has been influenced by the lecture “Project planning and risk management” in the week 2 folder provided on the VLE.

Identification. Firstly, risks can be organized into three categories: Product (quality, performance and the outcome of the game), project(affecting the schedule, delivery, and resources) and requirements (capturing the client's ideas) . These categories ensure a broad coverage of the entire project and are sufficient for a project of this size. Some of the risks are broken down further into sub-risks, which are indicated by a decimal point.

Analysis. For each risk in the registers below, we can assign 2 changeable qualities: “Likelihood” (Low/Medium/High) and “Impact” (Low/Medium/High). Both have an easy-to-read scale that will be consistent and clear for all risks, along with indicating the urgency for each risk to be managed and resolved first.

Planning. Every risk, regardless of impact/likelihood, will have a mitigation plan in place that can be used if a risk should present itself, but those that are more complex will be given more thought-out and descriptive. Each risk is also assigned an “owner”, which is someone who is named as responsible for the task (but every member of the team can contribute). This means they are responsible for the monitoring process and ensuring accountability.

Monitoring. The risk register can be used live as some of the “likelihood” and “impact” values will be adjusted if the risk is less prominent or vice versa. This can be done by any team member (but actively done by the owners) at key milestones of their respective roles and as a whole team when in regular meetings. Monitoring, by definition, requires good communication and dependency is on all of us.

Our team is divided into two groups: a generalist team to handle the heavier documentation tasks and a hybrid team to focus on the software. This structure helps prevent a low ‘bus factor’ (too much responsibility on one person) and ensures an even distribution of workload.

Risk Register Format

ID - will be unique and allow for breaking down

Description - clear explanation of the risk

Likelihood - low/medium/high

Impact - low/medium/high

Mitigation - the steps to reduce or recover from risk

Owner - The team or individual responsible

Systematic Tabular Presentation of Risks (Risk register)

Product - (quality, performance and the outcome of the game)

ID	Description	Likelihood	Impact	Mitigation	Owner
1	Website becomes inaccessible	Low	High	We are hosting on GitHub, which is reliable and tested regularly	Hybrid Team
2	A library becomes unusable	Low	High	We did research into all libraries and backups	Hybrid Team
3	The library is slower than anticipated	Low	Medium	We checked similar games as a benchmark	Hybrid Team
4.1	Github branch problems	Low	Medium	Regular commits and pushes, descriptive commit messages, local backups, and using GitHub revert if needed	Hybrid Team
4.2	Two developers are working on the same task	Low	Medium	Communication between the team will ensure that different parts of the code are worked on	Hybrid Team
4.3	GitHub tag is not labelled as complete	Low	Low	Regularly check and verify tags; online files allow easy monitoring	Everyone
4.4	Github Merging/commit issues	Low	High	GitHub has a history feature, so it can be restored and then carefully merged	Hybrid Team
4.5	GitHub becomes inaccessible	Low	High	Maintain regular backups and schedule team meetings to coordinate access	Hybrid Team
5.1	licensing/copyright issues with assets	Low	Low	Use only royalty-free or licensed assets	Everyone
5.2	Assets become faulty	Low	Low	Keep multiple copies and replacements	Everyone
6	Maze has a major flaw	Low	Medium	Either redesign maze or fix the flawed section	Maze designer

Project - (affecting the schedule, delivery, resources)

ID	Description	Likelihood	Impact	Mitigation	Owner
1	Someone becomes unavailable and so cannot do their task	Low	Medium	We have split the team into groups to tackle the project in different ways	Everyone
1.1	Implementation	Medium	High	The devs have good communication, and so will have to help each other out	Hybrid Team
1.2	Requirements	Low	Medium	We will have people from the generalist team fill in	Generalist Team
1.3	Timelines/diagrams	Low	Medium	If not made, others can fill in, but the timelines are accessible to all	Generalists Team
1.4	Risk management	Low	Medium	This is a live document, so it can be updated anytime	Generalist Team
1.5	Architecture	Low	Medium	The Hybrid or generalist team can check through the documentation and write it up	Everyone
2	A file is inaccessible as its a local file	Low	Medium	We have a Google Drive and GitHub to allow unrestricted access to all resources	Everyone
3	Diagram software becomes unavailable	Low	High	We have backups and justification for all of the software choices	Everyone
4	We miss-update the timeline	Low	High	Regular group meetings where we update the timeline together and everyone is filled in	Everyone
4.1	We miss an internal deadline	Medium	High	The two mitigations for this would be ensuring that we chase up the responsible people, or we delegate more	Everyone

				people to the task	
--	--	--	--	--------------------	--

Requirements - (capturing the client's ideas)

ID	Description	Likelihood	Impact	Mitigation	Owner
1	Client proposes a change	Low	Low	We would have to ensure this change can be done without breaking the rest of the game	Everyone
2	We miss one of the game requirements	Low	High	In our regular meetings, we discuss the current trajectory of the project and next steps	Hybrid Team
2.1	Misprioritization of requirements/elements of the game	Low	Medium	In the requirements document, each element of the project is given a priority and those with the highest will be done first	Hybrid Team
3	Misinterpretation of the requirements	Low	Low	Meetings with the client for any requirement that is not crystal clear to us will be discussed	Everyone
3.1	Technical implementation causes trouble	Medium	Medium	Before any coding is done, each step will be thoroughly planned beforehand and checked for feasibility	Hybrid Team
4	Half made a non-functional requirement	Low	Low	Before these are planned, the timings will be taken into account so the user doesn't have a bad experience	Hybrid Team