Risk Assessment and Mitigation

Cai Hughes<cabh500@york.ac.uk>
Ben Slater<bs1463@york.ac.uk>
Adeola Adeniji<aa3098@york.ac.uk>
Mathew Riedy<mr1723@york.ac.uk>
Riad Kasmi<rmk526@york.ac.uk>
Simon Konieczny<sk2144@york.ac.uk>

Our team's risk management for ENG1 involves a detailed four-step process: 1) identifying, 2) assessing, 3) treating, and 4) monitoring risks to proactively address potential challenges in our software development. By systematically evaluating and prioritising risks, assigning clear ownership, and implementing targeted mitigation strategies, we aim to navigate uncertainties effectively. This approach ensures we're prepared for issues, maintaining project momentum towards successful completion.

Step 1: Risk Identification

In the first step of the risk management process, the team systematically identified threats and risks to the project timeline and the product itself. These were identified through brainstorming sessions, personal judgement, and SWOT analysis. By combining the knowledge of everyone on the team, a comprehensive list could be created. The risks covered a wide range of problems, such as technological, project and product-related, as well as security.

Step 2: Risk Assessment

Once the risks were identified, it was possible to assess them in detail. This involved establishing each risks' type, likelihood of occurrence, impact on the project, and owners as well as writing a brief description. These factors were decided based on expert judgement and the research that the owners did on the subject area. By quantifying the likelihood and impact, the team (especially the owners) could be more aware of potential threats and be mindful to avoid them, or be able to pick the path with least harm to the project.

Step 3: Risk Treatment

In this step the team developed strategies to address the identified risks. This involved appropriate measures to mitigate risks from occurring as well as prioritising risks with high likelihood and impact over others. Risks deemed less severe were treated with less urgency. By establishing ownership of risks team members knew what they were responsible for and could take steps to mitigate specific risks through their work.

Step 4: Risk Monitoring

The final step of the process involved continuous monitoring and tracking of the identified risks, the people in charge of specific areas of the project (the owners of various risks) continuously paid attention to potential pitfalls and mitigated risks. The risk register is also a live document in the sense that it is continuously updated to changing and unforeseen circumstances. The team's clear communication, especially within the task divisions is an integral part of maintaining this iterative process and ensuring the project remains resilient to uncertainties.

The format of the team's risk register was shaped by the risk management process. Each identified risk has an identifier, enabling easy communication when discussing risks, a type, description, likelihood, impact, mitigation strategy, and finally the owner(s). This covers all the key information regarding ownership and managing of the risk. The format incentivises the team members to iteratively monitor and react to potential risks and collaborate on their mitigation.

The Risk Register of the Team is shown below, enabling the team to coherently and effectively manage and mitigate potential risks, as explained above.

Risk ID	Type	Description	Likelihood	Impact	Mitigation Strategies	Owner
R1	Technology	Technical challenges with learning how to use new software tools (LibGDX)	**Medium**	**High**	- Early prototyping and testing. - Consider alternative tools if necessary.	Cai/Ben
R2	Project	Scope creep	**Medium**	**High**	-Regularly review project scope with stakeholders; implement change control processes.	Riad/Simon
R3	Technology	Hardware/Software failure	**Medium**	**Medium**	-Ensure regular backups; use quality hardware; maintain software updates and patches.	Cai/Ben
R4	Security	Data loss or breach	**Low**	**High**	- Implementing robust data backup and security measures (version control on GitHub)	Cai/Ben
R5	Project	Team Collaboration Issues	**Medium**	**Medium**	- Schedule regular team meetings - Regular communication on the Discord server	Riad/Simon
R6	Product	Poor user reception	**Medium**	**High**	- Regular testing on target hardware specifications - Optimization for lower-spec devices	Mathew/Adeola

R7	Technology	Incompatibility with target hardware	**Medium**	**High**	- Regular testing on target hardware specifications - Optimization for lower-spec devices	Cai/Ben
R8	Project	Lack of Stakeholder Engagement	**Low**	**High**	-Regular updates and reviews; seek feedback; address concerns.	Riad/Simon
R9	Product	Poor User Experience (UX) Design	**Medium**	**High**	-Conduct user research - Iterative design process; usability testing.	Mathew/Adeola
R10	Technology	Difficulty in scaling the game	**Low**	**High**	- Design with scalability in mind	Mathew/Adeola
R11	Project	Delay in Development Milestones	**High**	**High**	- Implement agile project management. - Regularly review project timeline and adjust as needed (done during the twice-weekly team meetings)	Riad/Simon
R12	Product	Overcomplexity Leading to Player Frustration	**Medium**	**High**	- Simplify game mechanics where possible. Offer hints or tips for challenging parts of the game.	Mathew/Adeola Ben/Cai
R13	Product	Not meeting user expectation	**Medium**	**High**	- Engage in continuous user feedback. Iteratively improve the product based on feedback.	Mathew/Adeola
R14	Project	Inaccurate Requirements Gathering	**High**	**High**	-Using prototyping to validate requirements.	Riad/Simon