Risk Assessment and Mitigation

Risk Assessment and Mitigation

Group 9

Lewis Ramsey
Toby Rochester
Henry Sanger
Remi Shaw
Ethan Spiteri
William Timms
Antonio Tiron

[1] After considering the potential impacts possible risks could have on our project our team decided to use the following risk management process to ensure the successful delivery of our system within the time constraints provided:

1. **Risk Identification**
    a. As a team, identify risks which could potentially cause negative consequences. To keep the process simple and time efficient, risks which are extremely unlikely to occur, or have negligible consequences are ignored.
2. **Risk Analysis**
    a. After risks are identified, each risk must be assigned a likelihood of occurring, and a severity (potential impact).
    b. Likelihood is categorised by Low, Moderate or High and Severity is categorised by Insignificant, Tolerable, Serious or Catastrophic.
    c. Both are colour coded in the Risk Register below, for a quick understanding of which risks are of most significance, and therefore, require the most monitoring and mitigation planning.
    d. The register is updated whenever either a new risk is introduced or when a given risk's Severity or Likelihood changes.
3. **Risk Planning**
    a. This stage concerns our strategies for handling these risks given their likelihood and severity. Some risks can be avoided entirely (avoidance strategies), while others have an unavoidable chance of occurring, in which case we must implement mitigation strategies or contingency plans.
4. **Risk Monitoring**
    a. Each risk is assigned an owner, who is solely responsible for reassessing its likelihood and severity, as well as taking proactive actions to minimise the risk's impact.
    b. All risks are regularly monitored by the assigned owner throughout the development of the system.

Below is a Risk Register, with a row for each risk, and columns denoting each risk's ID (for purely organisational and reference purposes), Type (described below), Description (a short description of the risk), Impact (the effects of a risk), Likelihood (as 2.b), Severity, (as 2.b), Mitigation (as 3.a, the process of reducing the likelihood of a risk occurring and minimising its impact) and its Owner (4.a).

Risks are classified into the following types based on what they affect:

- **Project Risk:** Affects the project schedule, members or resources, but not the final product.
- **Product Risk:** Affects the product quality or totality.
- **Project and Product Risk**: Affects both the project and product.

Risk Register

ID	Type	Description	Impact	Likelihood	Severity	Mitigation	Owner
1	Project	Individual team members are ill.	Entire team falls reversibly behind schedule.	Moderate - High	Tolerable	Unpreventable, but the team must spread the workload and maintain a bus factor of at least 2 for all components.	Zhenggao
2	Project	Major rework of third party software rendering it incompatible.	Disastrous disruption to the project, especially if this occurs towards the latter stages of its production.	Low	Catastrophic	Only using software that has been out for an extended period of time and making sure to keep at least one local copy of previous software version in event of updates.	Damian
3	Project	Team Turnover (any team member leaving mid-project).	Entire team falls (possibly irreversibly depending on the stage of the project) behind schedule.	Low	Serious - Catastrophic	Unpreventable, but the team must remain on or ahead of schedule at all times nonetheless to reduce the chance of falling irreversibly behind, were this to happen.	James
4	Product	Loss of some functionality of the system on smaller screens.	The game may not be playable on some laptops or smaller monitors.	Moderate	Tolerable	Carry out usability and functionality testing of the system on different monitor, window or resolution sizes.	Ren
5	Product	The game is slow on systems with low hardware specifications	The game may be unplayable, or at least unenjoyable.	Low	Tolerable	Although exceedingly low chance for this small game, testing on different specification systems will be carried out nonetheless.	Tommy

ID	Type	Description	Impact	Likelihood	Severity	Mitigation	Owner
6	Product	The game has extremely low resolution on larger screens (for showcasing).	The game cannot be shown to large audiences.	Moderate	Tolerable	Testing on large screens will be carried out, and, if the team deems necessary, efforts will be made to fix the issue if present.	Ren
7	Product	Unexpected (minor) bugs in the codebase.	Depending on the severity of the bugs, this could mean the game is unsuitable for its intended purpose.	Moderate	Tolerable	Although small bugs may be unavoidable, mistakes in the code which would cause the game to be unplayable (and hence be of catastrophic severity) are detectable with vigorous testing.	Damian
8	Product	Unexpected (major) bugs in the codebase.	Depending on the severity of the bugs, this could mean the game is unsuitable for its intended purpose.	Low	Serious	Perform both automated and user testing to identify these bugs before.	Sharlotte
9	Product	A lack of a united clear vision of the end product by all team members.	The game may not suit the user requirements, or misunderstandings between members of the development team may slow production.	Moderate	Serious	By holding regular group meetings with all group members, establish a clear vision from the beginning, and document the user requirements, system functionality and mechanics early on, using tables, charts and diagrams.	Elliott
10	Product	Team members lack experience with some technologies or software	The team (most likely on the programming side) falls behind schedule.	Moderate	Serious	Identify the skills and gaps of team members at an early stage, and assign roles based on these strengths and weaknesses.	James

ID	Type	Description	Impact	Likelihood	Severity	Mitigation	Owner
11	Project and Product	3rd party libraries/ game engines are faulty, or otherwise not fit for purpose.	The programming team may fall behind schedule as a new 3rd party asset is sought.	Low	Serious	Only using reputable, well established 3rd party assets which have well documented guides and can be trusted to function correctly for a system of this scale.	Damian
12	Project	Arguments/conflict between team members	This could hinder group productivity for all members of the team and cause us to miss deadlines.	Moderate	Tolerable	If two members have an argument we will give out seperate responsibilities to keep them apart. If members cannot stand to be on the team together then we can speak to the module staff to find a better resolution.	All team members
13	Project and Product	Intellectual Property Infringement - used a third party asset in a way that the licence does not allow for.	The system must be rewritten, using new third party assets to replace the unlicensed one.	Low	Serious	Exceedingly unlikely, however, can be avoided by vigorously checking the licences of any third party assets before utilising it within the project.	Elliott
14	Project and Product	Requirement Changes	The system may need to undergo significant changes to meet these new requirements.	Moderate	Serious	Can be mitigated easily by staging regular meetings and consistent communication with the consumer, checking uncertaintanties with them where necessary.	James
15	Project and Product	Late completion of dependencies which core features are reliant upon.	Such core features either cannot be completed or are delayed, leading to a late completion date or	Low	Serious	When creating the project schedule, identify such critical dependency - feature relations early on and plan for their timely completion. Also plan to finish project a week	Tommy

ID	Type	Description	Impact	Likelihood	Severity	Mitigation	Owner
			incomplete system.	🟩	🟥	before submission	
16	Project and Product	Inadequate time for polishing appearance.	The game's release date doesn't allow time for refining the usability or GUI of the game.	High	Tolerable	Buffer time must be built into the project schedule, to allow time for just refinements after the core functionality is complete.	Tommy

References

1. Ian Sommerville, *Software engineering*, 10th ed. Boston: Pearson Education Limited, pp. 644-652, 2015