

Risk Management

Glossary

- **Risk** - A potential problem that may arise which can impact a project.
- **Risk Explanation** - An explanation of a potential risk.
- **Risk Threat Level** - The likelihood of the threat occurring.
- **Schedule** - The order and direction of the software projects development cycle.
- **Product** - The software project that I am creating.

Each risk is assigned one of three level ratings, indicating the probability of the negative outcome associated with the risk occurring during the project's development. These are as follows, in ascending order of probability:

1. **Low**
2. **Medium**
3. **High**

Assessment

Risk Category	Risk Explanation	Level
Schedule	Schedule is created with a best case scenario in mind, rather than being more realistic	High
	Sprints are not designed with other tasks in mind. I.E, Sprints are not filled with linked tasks that collectively make a function or a feature	Medium
	Issues that are not identified early in the project may impact meeting deadlines	Medium
	Incorrect prioritization of tasks leads to less crucial tasks being completed before core tasks, potentially stopping essential functions from being created with time constraints	Medium
	A similar game coming to market before this one can be completed	Low
Personnel	Personal strengths are misevaluated and time is required to adjust for this, affecting development time.	Medium
	Personal illness may occur and affect development time	Medium

Design Process	Research in relevant areas requiring more time than expected.	Low
	User testing finds major issues with the design of the application - Requires major rework.	Low
	Inability to test on a variety of different VR devices, with various specifications.	Medium
	Overambitious scope, resulting in incomplete project	Medium
	Scope creep, where interesting game mechanics may take too long to implement to be viable	Medium

Contingency Plans

Schedule is created with a best case scenario in mind, rather than being more realistic

Throughout the project, a pace must be met and should significant work not be met, the schedule should be reworked or the scope reduced to make it viable.

Sprints are not designed with other tasks in mind. I.E, Sprints are not filled with linked tasks that collectively make a function or a feature

During the sprint planning process, a sprint goal should be set. This goal should be to create a specific feature or function.

Issues that are not identified early in the project may impact meeting deadlines

A testing methodology should be used to test the code during each sprint, should issues not be identified during this time, there should be free time included within the project roadmap, to push work back a week should this become necessary.

Incorrect prioritization of tasks leads to less crucial tasks being completed before core tasks, potentially stopping essential functions from being created with time constraints

The project will be completed using agile methodologies, namely Scrum. Sprints should be organised based on getting core mechanics first, then iterate on this with less important tasks. Furthermore, the product backlog should be organised by how important each task is to the project. Furthermore, a minimum viable product will be confirmed and worked to before any features of the minimum awesome product is worked on.

A similar game coming to market before this one can be completed

Should this happen, it's unlikely the game is identical to the one being developed. Should this be the case however, steps will have to be taken to change the project, to make it reasonably different, as it's unlikely I can protect the IP at this time.

Personal strengths are misevaluated and time is required to adjust for this, affecting development time.

Significant research should be completed before the project begins, so that time can be allocated to effectively implement each feature. This should minimise the risk of this happening.

Personal illness may occur and affect development time

As previously mentioned, there should be time at the end of the product roadmap that can be used as a buffer, to push back work should sickness occur.

Research in relevant areas requiring more time than expected.

Significant research should be completed before the project begins, which should minimize the effect of this.

User testing finds major issues with the design of the application - Requires major rework

User testing sessions will occur throughout the lifespan of this project, testing individual components and features, rather than the project as a whole.

Inability to test on a variety of different VR devices, with various specifications.

The University provides several types of VR equipment for development, which should minimize the likelihood of one type of VR device having a different user experience. However, various PC configurations are unlikely to be able to be tested.

Overambitious scope, resulting in incomplete project

This should be noted during discussions with supervisor, if this occurs the scope may be reduced.

Scope creep, where interesting game mechanics may take too long to implement to be viable

Any interesting features will be added to the bottom of the product backlog, to be implemented should development time allow for it.