Risk 1

Group 2 - Vikingz

Damian Boruch

Tommy Burholt

Elliott Bryce

James Butterfield

Ren Herring

Zhenggao Zhang

Sharlotte Koren

Risk Management Process

Our risk management process involved 4 steps: risk identification, risk analysis, risk planning and risk monitoring.

The first step was to identify relevant risks that we might encounter during the project. In this phase it was important to eliminate any risks with negligible consequences or very low probability of occurrence. To help us identify risks, we used our past experiences working in team projects, both within the university and elsewhere. We also read through section 22.1 in *Software Engineering* [1] to get expert advice on risk management processes. To start we simply brainstormed some ideas, but then we began to categorise the risks into product related risks and human related risks.

We then began our risk analysis where we assigned each identified risk a severity and likelihood rating. Two team members were involved in this process and they each individually gave a ranking and the average score was taken.

After the analysis we had a good understanding of the known risks we might encounter during our project. We then had to plan avoidance and mitigation strategies. To help avoid risks, we made sure to not have any critical dependencies with a low bus factor. Some mitigation strategies we employed were:

- Bi-weekly meetings to promote a close working environment
- Created documents for internal todo lists to keep everyone up-to-date on current tasks
- Cloud sharing to reduce risk of important documents being lost.

As a group we also created contingency plans, such as using 3rd party sound and graphic assets if we run out of time or fail to make our own during the project timeline.

The final step of our risk management process is risk monitoring. This is an ongoing step throughout the project. We made sure that all team members were re-assessing risks related to their parts of the projects. It is the responsibility of the owner of a risk to make sure that the likelihood and severity of the risk is kept up to date.

Explanation of risk register columns

- ID: a unique risk ID so that the risk can be referenced easily elsewhere in the project
- Type: Human or Product. Human risks relate to issues with members of the team. Product risks relate to hardware, software or documentation issues.
- Description: a short description of what the risk is.
- Likelihood: Low, Med or High. The probability of a risk occurring.
- Severity: Low, Med or High. How damaging a risk would be if it occurs.
- Impact: an explanation of the impact this risk would have if it occurs.
- Mitigation: our planned strategy to avoid this risk and what to do if it occurs.
- Owner: the team member(s) responsible with tracking and updating this risk.

Risk Register

ID	Type	Description	Likelihood	Severity	Impact	Mitigation	Owner
R1	Human	Sound designer becomes unavailable	Low	Med	Our plan is to design in-house sound for the game. If the sound designer becomes unavailable, this would leave the project without sound which would fail the UR_SOUND requirement.	As a contingency we researched 3rd party sound packs, so that we could implement them quickly if needed.	Tommy
R2	Product	Estimated time taken to complete deliverables is underestimated	Med	High	This could lead to us not handing in our project on time and missing the deadline. This would cause us to lose marks and potentially fail.	We made an internal deadline of one week before the official deadline for all tasks (apart from testing code and updating the website) so that we were aware of any delay well before the due date.	Elliott
R3	Human	Member(s) of the team become ill or unavailable during project	Low	Med	Losing a member of the team would increase workload on all other members and would require a rework of responsibilities.	We decided that if only 1-2 members became unavailable we would be able to cope with the extra workload. If more members become ill we will ask for consideration from the teaching staff.	Zhenggao
R4	Product	Faults in 3rd party code/graphic extensions	Med	Low	This could cause bugs in our code that we don't have the capabilities to fix, reducing our project quality.	Due to LibGDX being a popular framework there are many options of packages we can use. If we find detrimental faults in a package, we will research and switch a separate package.	Damian
R5	Human	Conflict within group (fall outs, arguments)	Med	Low	This could hinder group productivity for all members of the team - not just those who are involved in the conflict. This could cause us to miss deadlines or not get tasks done.	If 2 members of the team have an argument we will separate their responsibilities to keep them mostly 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 members

ID	Type	Description	Likelihood	Severity	Impact	Mitigation	Owner
R6	Human	Poor code quality	Med	Med	This might put other teams off from carrying on our project as the code won't be maintainable - making us miss out on bonus marks	The reality of software development is that we can't guarantee that every piece of code will be of the best quality. We decided that it's important that we check over other peoples commits and bring up problems as they happen so that we can keep a high standard of quality.	All members
R7	Human	Poor write-up quality	Med	High	This could cause the project to not make sense to markers, meaning we get a worse result.	Multiple members of the team will check over each deliverable, including a team member not majorly involved in that deliverable to avoid bias.	All members
R8	Product	Bugs in program	Med	Med	This would make us lose marks in the implementation section of the project. Severe bugs could make the game unplayable and not allow us to meet the UR_EXPERIENCE requirement.	Create automatic tests throughout development and rigorously test before any deadline	Ren

References

[1] I. Sommerville, *Software engineering*. Boston: Pearson/Addison-Wesley, 2004.