Risk Assessment and Mitigation

ENG1 Team 9

Jacob Dicken
Bertie Cartwright
William Croft
James Dovener
Henry Chan

This document has been modified for Assessment 2 to reflect the changes made by ENG1 Team 6:

- Hussain Alhabib
- Ellen Matthews
- Minnie Poon
- Jason Ruan
- Daniel Smith
- Owen Smith

Risk Management Process

The first thing our team did in the risk management process was to identify all the possible risks that could affect the development of the game. We did this by firstly coming up with some sections of the project (for example the requirements) and then as a team brainstorming as many risks as we could think of and noting them down under the current section. We then eliminated the risks we felt were extremely unlikely or inconsequential for a project of this scope. With the remaining risks we came up with a simple but clear description of what the risk was, so when we came back to assess the risks we would easily be able to understand what each risk meant. Splitting the risks into different types/sections helps us easily identify what each risk corresponds to and reference them when we are working on that part of the project.

After identifying all our initial risks we analysed each risk as a team and discussed their severity and likelihood. We decided that the likelihood would span from Low to Medium to High as we felt this was enough detail as we needed for the likelihood of each risk. We also decided that the severity should range from Low to Medium to High as we felt that this was also enough detail for the severity of each risk.

Now that we had analysed each risk we then created a plan on how to deal with each one. This plan had to include at least one way to avoid or mitigate each risk and if possible both. We felt that having both an avoidance and mitigation strategy for a risk was useful as we had a way to reduce the chance of the risk occurring and the impact it would have if it did occur. For some risks we also have a contingency plan which is the best course of action if the risk isn't avoided or mitigated and often entails falling back to earlier proven solutions.

Finally we assigned an owner to every risk, the owner's job was to assess the likelihood and severity of that risk at a frequency of at least once a week. Then the owners were to report any changes to the team during our weekly meeting so that we could agree as a team that these values needed to be changed and if any of our strategies to deal with the risk needed to change or be enforced. Who owned what risk was decided by what their main role was in the team and which type of risk this role was closely linked with (for example the project lead was assigned "project" type risks). This ensured that the owner of the risk was familiar with what the risk meant and its effects, meaning that they will be better able to monitor that risk over the course of the project.

All this above information about each risk was finally input into our risk register which is a table containing every risk along with their: ID, Type, Description, Likelihood, Severity, Avoidance/Mitigation/Contingency Plan, and Owner. This register allows our team to easily lookup any risk and have all the information about that risk clearly displayed to them. The Likelihood and Severity columns are even colour coded from green (for the lower values) to red (for the higher values) which made sure the information was even clearer to our team.

----------OUR EDITS----------

ID	Type	Description	Likelihood	Severity	Avoidance/Mitigation/Contingency Plan(s)	Owner
R1	Project	Team Illness	Medium	Medium	**Mitigation:** Maintain a bus factor of at least 2 for key tasks by ensuring multiple group members understand critical areas. For short term absences, tasks will be reassigned. For longer absences, regular check-ins will be conducted to redistribute workload.	Jason
R2	Project/ Product	A loss of access to the Shared Google Drive which contains all our documentation.	Low	High	**Mitigation:** As a team we will ensure essential files are backed up regularly on external storage devices/platforms or secondary cloud storage services to ensure minimal disruption.	Daniel
R3	Project/ Product	Miscommunication/ Unaware of Assignment	Medium	Medium	**Avoidance:** Set up a centralised communication platform like Slack for real-time updates and task discussions. **Mitigation:** Schedule weekly meetings to clarify assignments, record meeting minutes for reference, and update the Gantt Chart for accountability.	Jason & Daniel
R4	Project/ Product	Unfamiliar/ Overcomplicated Software	High	Medium	**Mitigation:** Before committing to software, discuss options as a team to ensure they align with team members' experience. If necessary, hold training sessions or offer each other support if unfamiliar tools are chosen.	Owen
R5	Product	A software the team is using suddenly implements a subscription for their service.	Low	High	**Mitigation:** The Project Lead would initiate a team discussion to explore alternative solutions and evaluate their suitability.	Jason
R6	Require	The customer changes	Medium	High	**Mitigation:** Ensure that the architecture of the project is modular and	Whole

	-ments	the requirements causing us to rewrite large parts of the game, possible under limited time.			when designing it, ensure the possibility of expansion and adding features is maintained.	Team
					Mitigation: Use an agile software development methodology, which will allow for flexible response requirements changes late into the project. This encompasses meeting with the customer frequently, which should reduce the number of the changes that arise later on.	
					Contingency: If there is not enough time to make the changes or the changes required are closer to a prior version we should fallback to that prior version of the solution.	
R7	Product	Skill Gaps	Medium	High	**Mitigation:** Identify team members' strengths and assign tasks accordingly. Furthermore, encourage collaboration on certain tasks to build skills and leverage skill sets.	Jason & Whole Team
R8	Require -ments	Requirements that are written don't correspond to what the customer wants.	Medium	High	**Avoidance:** Meet regularly with the customer and specifically discuss the user level requirements with them. Then carefully create the system requirements as a team based on the user requirements that have been gathered.	Daniel & Minnie
R9	Product	Product is unable to be built/run on the required hardware or Operating Systems.	Medium	High	**Avoidance:** Check with our customer if they have specific requirements for device specification or operating systems the project must be compatible with.	Owen
					Avoidance: Periodically review the performance of the project using benchmarking to ensure CI systems attempt to build the project for multiple OS targets.	
R10	Require -ments	Unclear or ambiguous requirements	Low	High	**Avoidance:** Make sure requirements are carefully written from the start.	Minnie
					Mitigation: If a team member feels they are ambiguous or unclear, make sure to clarify this with the rest of the team and the customer if necessary.	

R11	Product	Issues with the game and/or the playability of it	Medium	High	**Avoidance:** Ensure the game is tested in the early stages of development and feedback is gathered at intervals throughout.	Whole Team
					Mitigation: Address feedback and test results regularly throughout development to improve game quality.	
					Contingency: If feedback/errors are found late on in development, focus on those which affect the user experience the most.	
