UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

ENG1 Assessment 1
Group 18 - Octodecimal

Risk Assessment and Mitigation

Group Members:

Izz Abd Aziz

Phil Suwanpimolkul

Tom Loomes

Owen Kilpatrick

Zachary Vickers

Michael Ballantyne

Risk Management Process

To begin with, we allocated two team members (Tom and Owen) to individually carry out risk identification, thinking of any risks that the team may face during the software engineering process. They assigned each risk a type: "project" if it affects project schedule or resources, "product" if it affects product quality/completeness and "product and project" if it affects both. They also carried out risk planning, devising possible avoidance and mitigation strategies for each risk to prevent any severe consequences to the project. Once these ideas had been generated individually, they were collated into a risk list with each risk's summary and possible mitigation strategy. Then, during a group meeting, the other members of the team proofread this list to ensure they were accurate. They were then given an opportunity to add any other risks and mitigations that they thought had been missed.

Following this, we performed risk analysis. Every member of the team anonymously rated the likelihood and severity of each risk. Both were rated on a simple low-medium-high scale, as all group members would share a common understanding of each rating's meaning. We did this using a google sheet, enabling all of the results to be stored together and allowing us to calculate mode values for use in the finished risk register. This was done with the aim of generating the most accurate results, as our likelihoods and severities represent the opinions of all of the members of the team. For risk monitoring, we assigned each team member ownership of two risks, allowing each risk to be iteratively checked for changes in likelihood and severity.

The reason that the risk management process was carried out primarily by two members was to balance the workload of the team members across multiple tasks. Other members getting the chance to verify, amend and add to the initial risk list meant that everyone in our group was on the same page, and our final risk list represented the group as a whole. It also guaranteed that all possible risks had been identified. Anonymous voting meant members didn't feel peer pressure during rating, ensuring that the risk register accurately represented the group's views on risks' likelihood and severity. Risk ownership being distributed evenly created a fair split of workload, and this allowed each member to focus further on monitoring their assigned risks.

Tom and Owen finalised our risks into the risk register seen below, with each risk following the format: the unique risk ID, the risk's type, a brief description of the risk with its potential impact if the risk is realised, the likelihood of the risk, the severity of the risk, a mitigation strategy for the risk and the risk's owner.

The risk register is structured in a simple but effective way, with the aim being to make it easy to understand whilst having enough descriptive detail to properly cover the risks and the mitigation strategies. Giving each risk a unique ID made it easier to refer to specific risks. The colour coding of severity and likelihood helped to allow critical risks to stand out easily, so additional focus could be given to the likeliest and most severe risks.

Risk Register

ID	Type	Description + Impact	Likelihood	Severity	Mitigation	Owner
R1	Project	Main group member working on a deliverable becomes unavailable, meaning it can't be completed in the planned timescale.	M	H	Assign at least 2 group members to each deliverable, increasing bus factor and making it more likely the tasks will be completed.	Owen
R2	Project	Disputes about the game or deliverables cause delays in the progress of the project, as group members struggle to settle a disagreement/conflict.	M	L	Implement a system to resolve conflict when it occurs. Make sure debates are professional and don't become personal.	Zachary
R3	Project	Team members miss meetings and don't receive important information about progress and plans for the project. This may cause a lower quality output and delays.	M	L	Keep a meeting log, accessible by all team members, that tracks key information about meetings. Keep a list of current tasks in a shared online chat.	Zachary
R4	Project	Team debates over roles and responsibilities make meetings less productive. This leads to a delay in the project completion.	L	L	Assign roles to team members based on their preferences at the start of the project. Regularly review these roles (approx weekly). Make sure tasks are clearly assigned to specific team members.	Tom
R5	Project	Delays in gathering requirements, preventing the progress of multiple other areas of the project - primarily architecture and implementation.	H	H	Regularly checking that the requirements gathering progress is in line with the initially planned schedule and possibly assigning additional team members to the task if progress is falling behind.	Phil
R6	Product + Project	Customer interview changes the requirements and specifications for the project.	H	H	Creating a flexible outline of the project plan to begin with without going into too much depth, meaning that it is easily malleable to any requirements that the customer identifies during the interview.	Izz

R7	Product	There are problems with integrating libGDX engine as it does not work as intended, meaning that the project cannot be implemented as initially planned - potentially at the cost of some of the initial requirements.	M	H	Commence thorough research into the selected game engine in an attempt to find any pitfalls or bugs that other users may have flagged and ensure the limits and capabilities of the engine are well known.	Michael
R8	Product	A third party library used in the game's code is defective. This may lead to the product not meeting its requirements.	L	H	Use only well-known libraries with regular updates, as this will mean they're unlikely to be defective.	Michael
R9	Project + Product	Features are developed and implemented that are outside the scope of the requirements, wasting time and possibly leading to other requirements being missed or not property fulfilled.	M	M	Comprehensively define the scope of the requirements before any development has taken place and check any proposed features against this to make sure they are within the project's scope.	Phil
R10	Business	Using images with inappropriate copyright licences or AI art in deliverables may infringe on copyright law. These would require changing, leading to added workload and time spent on documentation or implementation.	L	H	Validate image's usage rights and look for appropriate licences before using them. If using AI art, validate that all of the artwork used does not have any likeness or similarity to any brand's imaging, logo or products.	Tom
R11	Product	Ineffective testing allows bugs to go undetected, compromising the quality and usability of the game.	M	M	Multiple team members should write and perform regular product tests, with the aim that any errors or missed areas for testing are uncovered so bugs can be rectified before submission to the client.	Owen
R12	Project + Product	Poor code documentation, which can lead to different developers misinterpreting the code potentially leading to errors. This leads to increased overheads as developers seek clarification from each other for ambiguous pieces of documentation.	H	M	Make use of a standardised documentation format so that all developers have a clearer understanding of the meaning and context behind pieces of documentation.	Izz