Risk Assessment and Mitigation

Our team followed the Project Risk Management technique for Risk Assessment and Mitigation. It consists of four main stages:

1. **Risk Identification:**
   During the meetings, we brainstormed the potential risks and issues we could encounter while developing the game. To make this process more easier and organised, we decided to make categories:
   - Technology
   - People
   - Requirements
   - Estimation

2. **Risk Analysis:**
   Each risk was then assessed for its likelihood(probability that it would occur) and its severity(how badly it would impact). We used a simple scale to measure it:
   - Low (1)
   - Moderate (2)
   - High (3)

3. **Risk Planning:**
   For every potential risk, we came up with specific avoidance and mitigation strategies to reduce its likelihood and contingency plans to minimize its severity.

4. **Risk Monitoring:**
   On every meeting (twice a week), the owners of each risk would re-assess the situation (likelihood and severity) of the risk. And update the log sheet so all the team members are aware if any changes happen.

Risk Register

ID	Category	Description	Likelihood	Severity	Mitigation	Owner
R1	Estimation	Delays in the project timeline caused by tasks taking longer than expected or unexpected issues during development	2	2	Allow more time for tasks, and keep checking on other members to check if they need help or are falling behind schedule	Max
R2	People	Project delays or workload issues if a team member leaves or becomes sick	1	2	Assign multiple people to each role so there are still enough to complete the tasks	Sarina
R3	People	Conflicts or differences in opinion within the team	2	2	Resolve by having clear communication, listening to both opinions, then decide by majority	Maddie
R4	People	Not having enough knowledge or experience with the technology or software we are planning to use	3	2	Research the software before use e.g. youtube or tutorials, and ensure that all team members are confident in using it	Fatima
R5	People	Unequal contribution distribution	2	2	Track contribution transparently, ask if others know how to contribute or need help	Kaleb
R6	People	Team members not attending meetings or communicating	1	3	Set clear expectations for each member, reach out if someone becomes unresponsive	Alex
R7	People	Team members not being available during set meeting times	2	2	Plan to meet at a different time to ensure that all members can be included in discussions	Cory
R8	Requirements	Misinterpretation or lack of clarity of project requirements	2	2	Ask our client more questions about their requirements	Kaleb
R9	Requirements	Modifications to the originally defined project requirements	2	1	Keep in contact with our client so they can inform us of any changes, and keep good version history so we can modify from earlier versions if necessary	Fatima
R10	Requirements	Project doesn’t meet the client’s criteria	2	2	Create a checklist for all the requirements to ensure that	Max

					all are completed to a high standard	
R11	Requirements	Game plan is too complex to complete within the given time	2	2	Keep realistic expectations, and align our plan with the requirements	Sarina
R12	Technology	Software malfunction or unexpected behaviour due to coding or logic errors	3	1	Test code frequently to check for bugs and errors, any changes should be documented	Alex
R13	Technology	Problems in development if a needed software library isn't available or stops working properly	1	3	Have multiple libraries available to use	Cory
R14	Technology	Failure of a laptop to operate as expected	1	1	Store data in a shared github repository so it can be used from any device	Maddie
R15	Technology	Problems in managing and tracking changes in project files or code	1	2	Keep a clear project history and ensure that we update all documents including logbook as we progress	Sarina
R16	Technology	Data loss	1	2	Maintain good version history so that if significant data is lost it can be restored to the most recent version	Maddie
R17	Technology	Code being able to be altered by players to enable cheating	1	2	Create robust code to minimise cheating ability	Cory
R18	Technology	Audio / visual assets not loading correctly or being incompatible	1	2	Test asset loading early on, research compatible formats and standards	Alex
R19	Technology	Integration issues when combining individual parts	2	2	Define coding conventions early and integrate frequently	Max
R20	Technology	Sound, music, or visual asset copyright issues	2	3	Ensure that we use free assets which are not copyrighted, or create our own	Kaleb