5. Risk Assessment and Mitigation.

Group Number: **Team 20**

Group Name: **Gourdo Ramsay**

Group Members:

Lauren Waine
Megan Bishop
Tommy George
Bartek Grudzinski
Davron Imamov
Nathan Sweeney

Risk Management process:

Foremostly, we needed to identify the risks posed to the project. This was done by discussing with the team about potential risks and also evaluating their relative impact on the project. We based this on our experiences with previous projects and what problems we faced, when delivering work as a team. Considering the different risk types also helped us to think about different areas of potential risks so that all areas were covered.

For the risk register, each risk is given its own unique risk ID. This makes it easier to reference and not be confused with other risks. The risks are also ordered by their ID to make it easier to search through them. Accompanying the risk ID is a description of the risk. This explains what the risk is in a single and simple full sentence to make it straightforward to read and understand. Adjacent them is a risk type. This categorises what types of risks can happen and also makes it clearer what risks may affect which areas of the project. There are four types of risks identified: the product risks affect the overall integrity of the product produced; the project risks affect the time and assets of the project; the technology risks are risks solely from the underlying hardware and software being used and the business risks consider the effect on the client.

Each risk identified has a varying chance of happening and severity of consequences, represented by its likelihood and impact. This is important because it can change what mitigation strategies should be used based on these variables. A system of low (L), medium (M) and high (H) is used to denote the level of likelihood and impact pertaining to a risk as it is intuitive to understand and more complex systems aren’t necessary. Each of these are colour coded to increase clarity and quickly identify the more prominent risks.

The mitigation strategies, which are courses of action used to either lessen or avoid the consequences of a risk, are kept simple and as the project isn’t an essential system. This makes it easier to know what to do if the problem occurs as there’s already a plan in place and be more aware of the problem generally. Risks of low likelihood, can often be eliminated with avoidance strategies. Whereas with high likelihoods, the risk is probably unavoidable and a part of the changing and growing of a project so must be mitigated by being flexible and communicating changes within the group. Risks with low impact shouldn’t affect the project greatly and can be worked around; however, with high impact risks, contingency plans should be used to prevent a critical failure of the project.

Finally each risk should be monitored in order to identify if the project is being affected. Each risk is given to an owner, whose responsibility is to re-assess their risk’s likelihood and impact. This is based generally on their expertise of the project as they will be more aware and have more knowledge about the risks pertaining to their section. Each owner will be prompted to re-asses their risks at each group meeting. This will prevent risks from being neglected and also allow for any new risks to be raised.

Risk Register:

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	Product	Only one person is assigned to make the website (causing a high bus factor).	L	H	- Work closely as a team to check up on progress each week. - Have shared documentation so others can carry on the work.	Davron
R2	Product	Other incomplete deliverables (excluding website as previously mentioned).	L	H	- Use redundancy so multiple people are assigned to each section. - Also has shared documentation.	Tommy
R3	Project	Accidental deletion of project files (Human Error).	L	H	- Use software e.g. GitHub and google docx that save cloud versions of documents. - Have a local backup copy.	Nathan
R4	Technology	Project files are deleted (Technology error).	L	H	- Use shared files that are automatically saved online. -Make regular commits.	Tommy
R5	Project	Requirement changes and errors.	H	M	- Check in with the client to ensure they are happy with the project's direction. - Keep code modular so it's easier to implement changes.	Lauren
R6	Project	Project delays.	H	M	- Have weekly meetings to track progress. - Use a Gantt chart to create a realistic	Bartek

					schedule. - Add buffer time to absorb any setbacks.	
R7	Technology	Errors with IDEs and libraries.	L	M	- Use well documented and maintained programs so they are easier to debug and less likely to have errors. - Have multiple coders in communication so they can help each other with errors.	Lauren
R8	Product	Remaining bugs in code, when the program is submitted.	L	M	- Ensure that all of the main functionalities work by doing test playthroughs to limit any bug's severity.	Megan
R9	Project	Handover issues in change of project management.	M	M	- Ensure everything is well documented and modular (particularly code). - Ensure everything can be transferred easily (upload code and website to github so it can be copied and transferred).	Lauren
R10	Business	Very high competition may make our game obsolete.	M	H	- Listen to the customer and cater to what their target audience wants to ensure a good user experience.	Megan
R11	Product	The game may only work on certain laptops or	L	M	- Try running the program on	Davron

		screen sizes.			multiple laptops and computers. - Ensure that the program can adapt to different screen sizes.	
R12	Project	Code conflicts when commiting and updating each other's code.	M	M	- Have a clear project structure through UML diagrams. - Do small rather than large commits to limit conflict.	Nathan
R13	Project	Group conflict on project direction and implementation.	M	L	- Discuss project ideas and specifications before implementation to prevent confusion and frustration.	Bartek
R14	Project	Altering current code could cause the implementation to no longer work.	M	L	- When trying a new idea, create a branch so the main project is unaffected but different methods can be tried. - Otherwise roll back to a previous workable version of the code.	Megan