Risk planning began by identifying all the foreseeable risks that may affect our project. This was done by reviewing the interview and brief given from the client to create a clear plan of user requirements (requirements tables) with priorities and dependencies, which let us make development decisions which were more likely to deliver all the key user requirements reducing the risk of not completing the project before the deadline. Risk reviews were conducted at the end of our group meetings allowing for team members to discuss the status of the risks they were managing or raise concern for new potential risks that were not already identified. This was possible due to the small scope of the project and the low complexity of all the required systems.

We made a risk register to keep a record of any risks we have already identified and their appropriate management strategy. Every risk is given a type, likelihood, severity, strategy and an owner. All our risks were categorized into types project, product, technology and business where project risks cover all concerns about executing the planned tasks on time before the deadline risks like losing a team member or needing time to learn Lib GDX. Product risks concern the possibility of faults in the final product produced such as the game's controls not being understandable or bugs found during merging different systems together in development. Technology risks are failures in the software or hardware that is needed to develop and maintain the game such as the risks of using libraries that might become unavailable during the development process or the loss of the availability of the website hosting our game and documentation. Business risks affect the whole project's completeness such as change in management or the project becoming obsolete before completion. Likelihood and severity are both scaled from Low-Moderate-High (L,M,H) this lets us estimate the impact of these risks to scale the necessary management strategy since a low likelihood risk could be avoided entirely and thus warrant planning any contingencies or mitigations. While a highly likely risk would need a mitigation strategy and/or a contingency plan depending on the severity of said risk. Likelihood and severity are colour coded to clarify the overall impact of the risk to emphasise the importance of reviewing its risk status. Owners were assigned to risks to help keep track of risk status more easily as it divides the task of risk monitoring amongst the team, saving more time for development and implementation of the game.

Risk Register

ID	Type	Description	Likelihood	Severity	Management Strategy	Owner
R1	Project	Development Team member becomes unavailable	L	H	All tasks can be re assigned if a member is not available to complete their task	Ben
R2	Technology	The final game is incompatible with clients hardware	L	H	Software is to be tested on multiple team members laptops to ensure functionality	Ben
R3	Product	Website becomes inaccessible	L	H	Website is regularly maintained and tested	Ali
R4	Project	Underestimated time for balancing	H	M	Separate balancing track extra playtests	Utkarsh
R5	Product	Bugs	M	H	Game is tested out to make sure there are little to no bugs	Utkarsh
R6	Business	Using copyrighted assets	L	M	Check copyright rules before using the assets	Fares
R7	Technology	Plugin/library deprecated mid-project	M	M	Choose supported libraries fallback plan	Ben
R8	Business	Music/SFX license gaps or expirations	L	H	Minimize/anonymize consent policy review	Utkarsh
R9	Business	Final delivery package rejected	M	H	Delivery checklist pre-acceptance dry run	Ali
R10	Technology	OS/driver update breaks graphics	L	H	Freeze image rollback plan	Utkarsh
R11	Technology	Engine upgrade breaks shaders/pipelines	M	M	Pin versions controlled upgrade window	Ben
R12	Project	Late joiner slows team due to onboarding gap	M	L	Onboarding doc buddy system	Gal
R13	Techn	Physics step tied to	M	H	Fixed timestep	Hux

	ology	framelate, leading to inconsistent gameplay			delta-time audits	
R14	Business	Stakeholder wants analytics report for showcase but data absent	M	M	Log minimal events now build basic dashboard	Gal
R15	Business	Inaccurate feature list in hand-in docs	M	M	Doc freeze verification pass vs build	Hux
R16	Product	Inaccurate scoring system	L	M	Simple scoring rules keep values in a data file for instant hotfixes	Chihun
R17	Business	External showcase build leaks unapproved content	L	H	Redacted build profile content flags	Hux
R18	Project	New to Git, leading to accidental history loss	H	H	Create branches protect main frequent small commits	Ali
R19	Project	Under-documented code becomes unmaintainable	H	M	Comment template short READMEs per system	Ali
R20	Product	Rooms feel empty due to content shortage	M	M	Procedural decorators. Reuse patterns with tweaks	Fares
R21	Technology	Antivirus flags unsigned executable	M	M	Zip build publish checksums alternative host	Hux
R22	Business	Using music from ‘free’ site without proof	M	H	Keep licenses/screenshots prefer CC0/library	Hux
R23	Project	Last-minute feature add breaks stability	M	H	Rollback to previous version re-evaluate the code	Ben
R24	Project	No clear backup of finished build	M	H	Store last 3 builds in hot and cold storage	Ali
R25	Project	Relying on beta engine features	M	M	Prefer LTS/released features fallback plan	Chihun
R26	Project	Inaccurate or ambiguous requirements	M	H	Thorough user requirements stage, performed by the	Hux

		gathered			whole team, with analysis done on interview transcripts	
R27	Product	Too many features added beyond initial scope	L	M	Clear method selection and planning to ensure all deliverables are reasonable within the time we have	Gal
R28	Project	Time management conflict between team's other course modules	H	H	Ensuring all work is split evenly between team members and no individual team member is overwhelmed	Gal
R29	Technology	University services fail (VLE, WiFi, local networks)	L	H	Ensure the project is non-reliant on any University hardware and local copies are frequently available	Fares
R30	Project	Poor team collaboration and miscommunication	M	M	Ensure that a clear outline of individual roles is set out, with each member having a clear idea of their own work and others	Hux
R31	Project	Conflicts between team members in potential decisions	L	L	Having a clear method of resolving any issues and conflicts	Gal
R32	Product	Gameplay unclear or objective not obvious	L	M	Clear tutorial screen implemented giving information about the game	Chihun