ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	Project	Scope creep.	M	M	Set clear requirements.	Billy
R2	Technology	Inconsistent libraries.	L	H	Find workarounds online, if there aren't any then reevaluate.	Jacob
R3	Project	Team member becomes ill or otherwise unavailable.	L	H	Keep communication between group members so tasks can be taken over by other team members.	Billy
R4	Technology	Poor documentation	M	M	Iteratively document code as it's made; Javadocs at the least.	Jacob
R5	Product	AI produces art that breaches intellectual property.	L	H	Reverse google image search ai produced art. Use ai art sparingly only where royalty free art isn't available.	Billy
R6	Product	Library contains a bug.	M	M	Prefer libraries that are open source.	Sam
R7	Product	Compatibility issues from using outdated 3rd party libraries	M	M	Check developer release notes.	Sam
R8	Project	Poor quality assurance	M	H	Ensure good code testing.	Sam
R9	Technology	Team member accidentally deletes some code	L	H	Use version control system.	Billy
R10	Project	Two team members work on the same thing at the same time (ie duplicate work.)	M	L	Communicate what each member is working on, assign tasks each week so everyone works on different things.	Billy
R11	Project	Team members disagree on the correct appraoch to a deliverable.	H	M	The lead for each member of the project makes the final desiscion, this must be respected by all team members.	Billy
R12	Product	3rd party application works inconsistently accross team members' machines	M	H	Find another suitable application for the task. If not available, assign the task to a team member whose machine supports the application.	Jacob
R13	Technology	Underuse of libgdx and other 3rd party libraries	L	M	Read docs fully so that team members don't implement features that already exist.	Jacob
R14	Business	Unclear licensing of assets	L	H	Do not use assets that have unclear licensing.	Jacob
R15	Project	Architecture disagreeing with implementation	M	L	Use scrums to discuss which version is better (the architecture/implementation), and change the worse one.	Jacob
R16	Project	Game being too hard	M	L	Get people to test the game, and see whether they can complete it in 5-10 minutes fairly consistently. If not, make the game easier	Jacob
R17	Project	User interface/Controls are unintuitive	M	H	Include controls on the main menu so that users understand all of the things that can be done in the game	Jacob
R18	Resource	Shortage of skilled developers	M	H	Implement knowledge sharing and have regular meetings to check on the implementation's progress	Leuay
R19	schedule	Underestimation of development time leading to missed deadlines	M	H	Have comprehensive project planning and allow a buffer for unforseen challenges	Leuay
R20	Quality	Bugs and glitches affecting the gameplay	M	H	Implement testing tools and have the game played for quality assurance	Leuay
R21	External	Changes in dependancies policies/regulations	L	H	Regular monitioring and compliance checks, also diversifying our dependancies	Leuay
R22	People	Poor communication among team members leading to misunderstandings, conflicts, or inefficiencies	H	M	Claryfiying roles and responsibilities and encouraging open dialogue	Leuay
R23	Competition	Other teams hearing unique project ideas and implementing them	L	H	Ensure all members communicate unique ideas outside of practical rooms to avoid ideas leaking	Leuay
R24	Cultural	Misrepresenttion of a particular demographic	L	H	Factor in culturalsensitivity and ensure the product cant be misinterpreted in any way	
R25	Localisation	Not developping the game with the player engagement in mind-(game for comp sci student)	L	M	Conduct player research and design engaging , relevant game mechanics	Leuay
R26	Technology	Platform Compatibility Issues	L	H	Conduct early platform analysis and design a game optimized for average PC specifications	Leuay

R27	Ethical	Game encourages/permits an unhealthy lifestyle	M	M	Do not add many unhealthy options for the player to take, and penalise the player for making any remaining ones	Billy
R28	Project	Inadequate error handling	M	H	Use exception handling and employ recovery mechanisms	Sam
R29	Project	Technical debt accumulation	M	M	Prioritise code quality, and constant relation back to user requirements	Sam
R30	Project	Memory dependency issues	M	M	Outline flow of data in user interaction, to highlight any possible memory oversights	Sam
R31	Project	Code refactoring causes implementation issues	H	M	Use incremental refactoring	Sam