Risks Assessments and Mitigation

Group 27 BlackCatStudios
Jack Vickers, Hubert Solecki, Jack Hinton, Sam Toner, Felix Seanor, Azzam Amirul Bahri

Taking over from Group 30 Triple 10
Kelvin Chen, Amy Cross, Amber Gange, Robin Graham, Riko Puusepp, Labib Zabaneh

The Risk Management Process

Triple 10's risk management process began with risk identification, where they brainstormed potential risks that may have posed issues during project development. After briefly assessing each risk, those with a lower probability of occurrence or severity were dismissed so that less time was wasted monitoring unnecessary risks. Continuing with their analysis, they discussed each risk and determined a likelihood and severity rating. Following analysis, they moved on to risk planning, which involved the development of strategies to prevent or mitigate each risk, then distributed them amongst the team so that each member monitors the risk during development, updating their possibility of occurring and impact as necessary. Below shows their risk register which outlines any risks identified, listing their likelihood, severity, owner, and mitigation/avoidance strategies. In the risk register, they included an ID which allowed each risk to be easier to reference throughout the documentation, the mitigation column to create a plan for the worst-case scenario and an owner to manage this risk to prevent and mitigate.

The Risk Register

This is Team Triple 10's risk register for assessment 1:

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1.1	Project	Programmers become unavailable	L	M	We have 3 programmers so one should always be free	Labib
R1.2	Project	Game can't run on several platforms	L	M	Ensure we develop the game to be supported on several operating systems	Robin
R1.3	Project	Can't resize game window	L	H	Ensure we develop a method to resize	Robin
R1.4	Technology	Choosing the wrong library for our game	L	H	Look at the advantages and disadvantages of each library and ensure we choose the right one for our game	Kelvin
R1.5	People	Game too hard/easy	M	L	Ensure game balance is thoroughly explored and tested	Riko

R1.6	Team	Poor distribution of work	M	H	Organise what we're all doing together	Amy
R1.7	Team	Not having clear set goals	M	M	Set clear goals so we all know what we're doing	Amy
R1.8	Project	Program is littered with bugs making it unplayable	M	H	Have a good game design and clean code that makes debugging the game quick and easy	Labib
R1.9	People	Player unsure of how to play the game or what controls to use	M	H	Use standardised controls such as "wasd" and have instructions on screen to help users play the game	Robin
R1.10	Project	Loss of game files or information files	M	H	Make sure that we always have a backup of any files we're working on (especially the important ones!)	Amber
R1.11	People	Not meeting deadlines	L	H	Always have someone that's checking progress on a weekly basis and making sure that workflow is on track	Amber
R1.12	Project	Not meeting assessment requirements	L	H	The first things to implement in the game are the functionalities needed to complete the requirements.	Kelvin
R1.13	Project	Infrequent maintenance of the website	M	M	Make sure that as part of each weekly meeting we add a plan or add directly to the website	Riko
R1.14	Project	Report format unclear and hard to read/ not enough detail	L	H	Ensure report is reviewed by the group and meets assessment	Amy

		about our project			requirements	
R1.15	Project	Poorly designed website	L	M	Follow HCI design rules and heuristics	Robin
R1.16	Project	Inappropriate assets or assets we don't have a licence to use	L	H	Make sure that all assets fit the theme of the game and come from a trusted source	Kelvin
R1.17	Team	Badly managed risk management	M	H	Make sure that everyone keeps on top of the risks they're owners of to make sure that all risks remain well observed	Amy

Updates to Risk Documentation after Takeover & Contingency Plan

- BlackCatSudios decided to continue on using Triple 10's conventions since they are similar to ours in terms of the risk register, risk mitigation and member assignment.
- We created a new risk register for assessment 2, merging some of Triple 10's relevant risks with ours from assessment 1.
- Based on our experience in assessment 1, we have also decided to update our risks, such as the risk of inadequate Git knowledge, where we have changed the risk's severity and likelihood to low due to an increased knowledge and experience. This was completed for all the risks in our new risk register.
- Additionally, we have added further risks related to assessment 2 specifically regarding understanding Team Triple 10's code base, using a change log and other unique features related to assessment 2.
- We continue to use our current mitigation and contingency strategy since it worked well for us in the last assessment; we maintain a high "Bus Factor" for all our risks and continue to actively work on contingencies if and when risks occur. We enforce good communication, teamwork and honesty and actively avoid redundancy in our work. Furthermore, we ensure our work is recoverable through the continued use of version control software such as Google Drive and Git.
- We also enforce quality assurance on all work and peer review our work to ensure that it is up to the best possible standards.
- To add, we have added additional responsibilities where needed for new risks. As before, risk owners must ensure they are aware of their own risks, understand their contingency plans and actively report when these risks occur so that they can be recorded and dealt with by the team as stated. All members have the option of updating the risk register when necessary. Again, good communication and keeping to convention is encouraged to ensure the reduction of further risks and quick action to occurring risks is put in place.
- We dynamically assess and deal with risks through the weekly updates on our website along with our weekly gantt chart screenshots so that all members are aware of the risks that are currently occurring and understand how to deal with them. We discuss the risk that we are currently aware of due to circumstances in the team and with the project before providing an effective mitigation according to our mitigation plan and current circumstances. These are stated in our weekly updates so that the entire team were aware of the circumstances of the project and could actively help and avoid any further complications.

BlackCatStudios' risk, likelihood and severity categories

Type of Risk		Likelihood Categories		Severity Categories	
Project	Affects the project schedule and the completion of work	Low	Unlikely to occur/ something that we do not expect to occur	Low	Does not affect the project too much
Product	Affects the quality, robustness, and completion of the game and its code	Medium	Moderately likely/something that could occur	Medium	Affects the problem but does not need immediate reaction
Technology	Affects the technology and devices we use during development and run the game on once it is complete	High	Very likely to occur/something we expect to happen	High	Affects the problem severely and needs immediate reaction.
Business	Affects our team, the popularity of our game, and the customer's opinion of the game.	N/A		N/A	

BlackCatStudios' Risk Register for Assessment 2

Risk ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R2.1	Project	Any member of the team missing during meetings or periods of time during the assessment period	M	M	Minimum bus factor of 2 on each project section; good communication encouraged; minutes taken at every meeting so that members are up to date	Hubert Solecki
R2.2	Project	Weekly updates for the website unable to be made due to Jack Vickers being ill	L	L	Minutes that can be referred to, potentially someone else can take over. Mkdocs is easy to use and well-documented so would be easy for another team member to understand.	Jack Vickers
R2.3	Project	Group leader falls ill or is unable to manage	L	M	Both Felix and Hubert are taking leadership over the group, and the dev team generally knows what they're doing	Felix Seanor, Hubert Solecki

R2.4	Project	Exam deadlines near the end of the assessment 2 period	M	H	Collaboration to reduce workload encouraged, project planning and project sector planning encouraged; time management encouraged; good communication encouraged, regular team meetings during the Spring term	Sam Toner
R2.5	Project	Code being deleted, losing progress	L	H	Make sure regular backups are made	Felix Seanor
R2.6	Project	Unable to meet in person	M	L	Use communication tools such as Discord to hold meetings or allow members unable to be there to still be apart; book meeting rooms; collectively decide on times/dates that suit everyone	Everyone
R2.7	Project	Missing files from merging of GitHub	L	H	Get the files from team Triple 10 by emailing them personally	Azzam Amirul
R2.8	Project	Changes to deliverables and code forgotten	H	M	Use google docs comments in deliverables to make note of changes and highlight changes in the code using comments and author tags	Sam Toner

R2.9	Project	Plan dynamically changes based on member requirements, deadlines and any other factors potentially affecting decided plan	M	M	Dynamic and malleable plan enforced so that it can be fit around members' commitments and any potential events occurring during already set plan; good communication encouraged; collaboration and adaptability encouraged	Everyone
R2.10	Project	Group conflicts	L	M	Group ensured to create healthy relationships before project started; leadership structure was discussed and implemented; project plan and group conduct discussed and implemented	Hubert Solecki
R2.11	Project	Game is not completed in time, some features missing	L	H	Estimates of how much development is remaining can be made each week and if needed, additional effort and time can be put in over the Easter break	Felix Seanor

R2.12	Project	Lack of knowledge in any areas to be able to complete sections	M	M	Members encouraged to watch lectures and seek other resources for education before they begin a project section; communication for this is encouraged to ensure everyone is up to date on their education; collaboration encouraged; refer to plan for when certain sections will be started	Jack Vickers
R2.13	Project	Quality of work lower than expected	L	H	Quality assurance checks encouraged; regular peer reviews encouraged; good communication encouraged; checking project plan document encouraged	Sam Toner
R2.14	Project	Code style and naming conventions are not consistent	H	L	Java style checks will be incorporated with continuous integration and github actions so that checks are run when code is pushed to the repository. We will have regular checks to ensure code style is consistent and use auto formatting in IntelliJ to ensure consistency.	Jack Vickers
R2.15	Project	Program is littered with bugs making it unplayable	M	H	Create unit tests which cover the majority of the code.	Felix Seanor
R2.16	Project	Unit tests runs but the game doesn't run	H	H	Double check code base and documentation for the unit tests.	Azzam Amirul, Jack Vickers

R2.17	Project	Inappropriate assets or assets we don't have a licence to use	L	H	Make sure that all assets fit the theme of the game and come from a trusted source. Purchase the assets ourselves to avoid any issues.	Jack Hinton
R2.18	Project	Lack of understanding of the codebase	M	H	Read the deliverables written by Team Triple 10 and look at their javadocs. Spend time looking at how their code works before attempting to add to it	Felix Seanor
R2.19	Project	Difficulty taking over Team Triple 10's website	M	M	To mitigate this difficulty, Jack Vickers has changed all of the html files for the website to markdown files so that MKdocs can be used. This makes it easier to add things to the website. Liaise with Triple10 to obtain any other files or information needed.	Jack Vickers
R2.20	Project and product	Missing system and user requirements in the requirements documentation	M	L	Missing requirements can be easily added to the requirements document	Jack Vickers

R2.21	Product and project	Requirements changing during development process	L	H	Scrum Agile method and spiral lifecycle chosen to mitigate effect on project if requirements change dynamically; good communication encouraged; adaptability encouraged; dev team create an engine to easily adapt code if requirements change and items need to be modified, removed or added	Azzam
R2.22	Project	Programmers become unavailable	L	M	One of the other programmers will take over temporarily. Every team member will take part in development	Felix Seanor
R2.23	Product	Progress of development does not meet the initial requirement of the initial product brief eg: wrong language, not enough characters, etc.	L	H	Go over the product brief and initial requirements and cross check with the requirements prepared by team members to ensure the development of project does not fall back and a mistake is not dragged too far	Jack Vickers

R2.24	Tech	Developed game does not run on macos	L	M	We know that the game runs well on both linux and windows devices as we tested it on these systems when we took over from team Triple 10. We will use continuous integration tools and github actions to build versions of the game for windows, linux and mac systems. If we	Sam Toner
R2.25	Tech	Lack of knowledge and experience in Git and GitHub	L	L	Members watch lectures and practice using Git and Github as shown in the lectures and other resources online.	Jack Hinton
R2.26	Tech	Unfamiliar with LibGDX	L	M	Followed the tutorial and then implemented a custom engine over the top	Jack Hinton
R2.27	Tech	Libgdx not containing the correct tools for a goal	H	H	Research other ways to achieve the goal, with the best and most efficient way	Felix Seanor
R2.28	Tech + project	Crucial technology gets broken	L	H	As a team work quickly to replace the broken tech with temporary tech, in order to finish the project	Hubert Solecki
R2.29	Business	Project presentation does not portray how the game meets the criteria	L	M	Prepare presentation beforehand and double check the criteria for the presentation	Sam Toner

R2.30	Business	Poor distribution of work. Each team member does not roughly get an equal share of the marks	M	H	Organise what we're all doing together	Azzam
R2.31	Project	Someone has a personal emergency and becomes unavailable	M	H	Bus factor of 2 or more	Hubert Solecki
R2.32	Project	There is low coverage of the code in the code base.	M	H	Consistently run coverage reports in IntelliJ to keep track of progress and see which lines have been covered by tests and which have not. Adjust test plan and tests as necessary to cover this.	Jack Vickers
R2.33	Project	Some unit tests fail	H	M	If tests fail, ensure that the test is written in the way that requirements are stated and collaborate with developers to ensure knowledge of the test is up to date. If needed, fix bugs in the code base to ensure that tests meet given requirements. Run tests in debug mode to find error of test in game code and fix them at the source	Azzam Amirul
R2.34	Project	Not updating the change log, resulting in missing history	M	M	Quality assurance checks weekly, if not more regularly. Communicate if someone notices that a part is missing as soon as possible so that the missing part is easier to find	Everyone