Deliverable: Risk assessment and mitigation

Group number and name: Group 10 - Uptown Func()

Group member names: Freddie Cooper, Brandon Stahl, Matthew Tsang, Ryuchie Tjhuanda, Joel Weston, Ben Williams

a)
Risks are potential problems that we need to address as they could harm our project by delaying development or affecting the quality of the final product. To avoid this, we have produced a table of risks below so we can better anticipate what could go wrong in the project.

In the table below, we have included a scale of the risks based on how likely they are to happen and how severe their effects would be if they were to happen, with ‘L’ being rated as low, ‘M’ being medium and ‘H’ being high. Along with that we also explored ways to resolve or mitigate the effects that the risks would have on our development. We decided to design our risk table in this way because it would tell the reader all of the important information about a specific risk while being easy to understand.

Before putting the risks into a table, we sorted them into categories. The categories we have used are technology, people, product and project. Technology relates to technical risks to do with coding, these would include bugs and any software related issues. People risks are the risks that come from team members, such as missing team members and conflicting time schedules. Product risks affect the quality of the final product, an example would be accidentally including a feature that's not supposed to be there in the final game. Lastly, we have project risks, these are risks related to the organisation of the group itself and development of the project, an example would be requirement changes for the final product. We used these categories because it covers all risks that we could think of perfectly. Another reason is because their names would give the reader a good idea of the risk while not being too vague.

Before starting our project, to set up the development process, we gave out roles, decided on weekly meeting times and chose the right Java libraries. One of the main things we discussed was risks to the project, due to this meeting being early on in the project, most of the risks we came up with were in the project category. These risks have more to do with the organisation of the group and possible tools we will be using. It is important to address the risks at this stage since any of them occurring early on in the project could have effects much later and could also setback the project by weeks or affect the quality of the final product.

As development continued, we managed to include more risks to the project. Most of the risks we found at this stage are more towards the technical side such as bugs, git merge conflicts, etc. The risks at this stage are less serious compared to the organisational risks near the start, with the worst effect being a slight delay in the project by a few days or a slight drop on the quality of the product.

b)

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	Project	Choosing the wrong Java library	L	H	Research on the best libraries to use, libGDX	Freddie
R2	People	Team members become unavailable	L	M	By having more members than needed, development can still move smoothly even in the absence of a member	Ben
R3	People	Team member being behind on work	M	M	Flexible schedule so another team member could fill in	Isaac
R4	Project	Multiple team members working on the same thing at once (i.e two maps being made)	M	L	Organise and give out tasks so that no one is accidentally working on the same thing	Ryuchie
R5	Technology	Merge conflicts on github, code produces errors on other devices	M	M	Every team member works on their own branch, ‘git pull’ before making any changes	Ben
R6	Technology	Incompatible code, code for one aspect of the game could break code for another thing	M	H	Separate git branches for all scripts, merge to main branch once done	Matthew
R7	Product	Camera zoom in/out used for debugging still included in final product	L	L	Double check the product before releasing	Brandon

R8	Technology	Newer versions of LibGDX/Java causes code to stop working	L	H	Don't update to newer versions	Joel
R9	Project	Requirement changes	L	H	modular code to make it easy to change	Matthew
R10	People	Conflicting time schedules make meetings hard to plan	H	L	Delegate work outside of meetings, keep in contact regularly online	Freddie
R11	Project	Improper allocation of resources	M	M	Flexible team members so they could be allocated to work on different parts of the project	Joel
R12	People	Team member conflict	L	H	Try to have another team member mediate, if it doesn’t work then the module lead intervene	Freddie
R13	People	Lack of skilled team members in certain fields	M	L	Assign people based on their strong suits	Isaac
R14	People	Lack of clarity in instructions in the team	L	L	Keep a note of the instruction, ask when unsure	Ryuchie
R15	Technology	Unclear code, no comments or bad format	L	L	Comment on code, follow standard coding style and format	Matthew
R16	Project	Final product does not follow client requirements	L	M	Careful write up of requirements	Brandon

R17	Project	Lack of documentation and online support for tools we use	L	M	Do research on tools before deciding to use them	Ryuchie
R18	Product	Insufficient testing time	L	M	Finish project with time to spare, test regularly throughout development	Ben
R19	People	Lack of collaboration between team members	L	H	Weekly meeting in person, message each other regularly	Joel
R20	Product	Unclear requirements from client could result in unsatisfactory product	L	M	Don’t ask open-ended questions to the client, be specific	Brandon