Risk Assessment and Mitigation

Introduction

Risk management is an important part of any project, we must prepare for what could happen during the course of the project in order to be able to quickly recover and stay on track. The risks which are shown below are relevant to the particular SEPR context and take into account the scale of the project and aim to cover only risks which are realistic within this context.
The risks are here presented in a tabular format. This table is split into 6 columns; the first column gives the risk an identification number (ID) for easy reference in our requirements and also if a risk does happen and we need to resolve it. The second column describes the risk itself. The third column gives an estimated likelihood of the risk occurring. To indicate the likelihood of each risk occurring we have use a high medium and low rating which is then also colour coded:

Likelihood	
Green	Low likelihood, this risk is not likely to occur. Roughly a 25% chance although some extreme risks could be a lot lower.
Yellow	Medium likelihood, there is an equal chance of the risk occurring or not occurring. Roughly a 50% chance.
Red	High likelihood, there is a good chance that this risk will occur . Roughly a 75% chance.

These colours were chosen as a low likelihood is generally preferable to a high likelihood. The fourth column describes the impact the risk would have on progress in the project. The fifth column shows the severity of the impact using this colour coordination:

Severity	
Green	Low severity, may mean a few hours extra work but nothing major.
Yellow	Medium severity, could add up to a week of extra work and may threaten a deadline.
Red	High severity, a major set back which could affect the whole project.

These colours were chosen due to their natural association with severity which is well known. The sixth and final column details how we will aim to avoid such a risk and deal with it.
The overall table is split into sections which group together similar risk such as software risks. Each section is then ordered by severity, highest first. Equal severity is ordered by likelihood. This table will be regularly consulted in an attempt to monitor the risks and try to ensure they do not occur and catch them early if they are occurring.

Lorem Ipsum

Table of risks

Software risks

ID	Description	Likelihood	Impact	Severity	Mitigation
1	Our game may be slow or unresponsive.	Medium	No one will want to play the game.	High	Improve efficiency of the game wherever possible and regularly check performance
2	Software library doesn't work or lacks a feature e.g. has a bug that stops the game from working, or is missing a feature required for the game to work.	Medium	We would struggle to implement the feature we want to add. We would also use up lots of time trying to solve the issue.	Medium	Test the elements of the library you plan to use beforehand. Also, make sure the library has an active community surrounding it and that bugs are fixed quickly. If it was early stages we could get a new library but this would require us to rewrite our code to work with the new library.
3	Code is hard to understand and seems too complex.	Low	Could cause bugs and makes bug fixing harder. Slows down the productivity of the group.	Medium	Use meaningful variable names and plenty of comments,both in code and in commit messages. Make sure code is reviewed by the majority of members before it gets merged into the repository.
4	Conflicts in git. Different members changing the same code.	High	May need to move code around and even rewrite.	Low	Make sure people work on separate elements by assigning them to different tasks and if not then make use of Gits tools.
5	Our own software doesn't work as intended.	High	Will need to bug fix. Loss of time and potentially productivity if that function or feature is the bottleneck of the game.	Low	This is a normal part of software development. We all make mistakes. However, before code is approved by the group we will use unit testing that will test key functions of the game as we develop them meaning that should a function break we will know about it before it's merged.

Hardware risks

ID	Description	Likelihood	Impact	Severity	Mitigation
6	Personal computer breaks long term or is lost.	Low	Could lose work and be unable to work.	Low	Ensure work is saved online to google drive cloud service and that code is stored on github. Department PC's should be accessible most days and have all the tools we need.
7	Personal computer crashes while working.	Medium	Potentially will have lose work, meaning you lose time doing it again.	Low	Save regularly, google docs[2] will do this for us. Regularly commit code to personal branches so that it stored elsewhere other than your PC .

Lorem Ipsum

Risks with people

ID	Description	Likelihood	Impact	Severity	Mitigation
8	A team member leaves the module or even the course.	Low	They may have only access to their work, also the rest of the team will have more to do.	High	As above store online but also try to keep each other motivated to avoid this.
9	A team member is ill/away for a week or two.	High	They might have been skilled in a certain area that no other member can do well.If they have the only access to work may get behind from it.	Medium	Hard to avoid, but we should store work online where everyone can access. If we work in pairs to complete tasks then there will be less of a chance of having one person who knows the most about one area.
10	Arguments within the team.	Medium	Disrupts the work of the team and prevents us moving forwards. Also, unpleasant for the team as a whole.	Medium	Try to avoid conflict but if necessary have proper debates perhaps using a mediator, do not keep issues hidden.
11	Lack of communication.	Medium	Tasks may be done twice or not done at all.	Medium	Keep strong communication using the tools we plan to use.
12	A team member does not do their work.	Medium	Could disrupt other members work and could make the other team members annoyed.	Low	Don't give members too much work or work they cannot do, ensure that the team communicates well and regularly meets up to discuss how the work is going.

Risks with tools

ID	Description	Likelihood	Impact	Severity	Mitigation
13	Google drive servers stop working.	Low	Could lose/lose access to work that is stored there.	Medium	Store work locally , and on other services.
14	Central git repository[1] is lost in some way.	Low	Temporarily lose access to it.	Low	Keep up to date local copies so can be easily restored. We could host our own local copy should github go down.
15	Website hosting fails.	Low	Users lose access to the website.	Medium	The website files are stored on github and every team member has a local copy of the repository on their computer so we could bring the site back up on a different server. The site is also protected by cloud-flare[3] who will provide a cached version of the site if our host were to go down.

Lorem Ipsum

Requirements risks

ID	Description	Likelihood	Impact	Severity	Mitigation
16	Not including a requirement which is required by the customer.	Low	We let the customer down and have failed them.	High	Make sure key requirements are elicited from the customer so they get what they want.
17	A requirement could change/ be added.	High	May need to rewrite code or add extra code to account for it. Extra time will be needed.	Medium	Our software architecture must be flexible and able to be changed easily.
18	Stating a requirement that we cannot actually achieve.	High	Let down the customer and also waste time.	Medium	Be sensible when deciding requirements, be sure you can achieve them.
19	Ambiguity in requirements.	Medium	May end up making something which is not what was originally intended.	Medium	Ensure requirements are clear and check any ambiguities with the customer.
20	Choosing requirements that the customer doesn't really want.	Medium	Waste time which could be spent on requirements they did want.	Low	Ensure you know which requirements the customer really wants and which can be ignored.

Estimation risks

ID	Description	Likelihood	Impact	Severity	Mitigation
21	Expect the team or a team member can do more than they actually can.	Medium	Work is not done or is done to an insufficient standard.	Medium	Give tasks that people can do and if they can't then help them. When working on difficult tasks work in pairs to complete the task meaning individual team members don't feel as overwhelmed by the task
22	We may underestimate how long it will take to do some work.	Medium	Work ends up taking longer than expected or not done to the standard it could be done. This could cause other areas of the project to suffer	Medium	Set realistic timings to do work and be realistic on how long a task will take. Account for unforeseen delays in our plan adding time where we can catch up.
23	Be too pessimistic about what we can achieve.	Medium	We end up with a product which is not as good as it could have possibly been.	Low	Push our limits but also stay realistic and within the requirements. If we have extra time then we can use it to enhance the product.
24	Distribute tasks incorrectly.	Low	Team over/under worked.	Low	Distribute tasks appropriately and tell others if feel over/under worked.

Bibliography

[1] GitHub [online] Available https://github.com [Accessed 01/11/2016]
[2] Google Drive [online] Available https://www.google.com/drive/ [Accessed 01/11/2016]
[3] Cloud Flare [online] Available https://www.cloudflare.com/ [Accessed 01/11/2016]