

Risk Assessment and Mitigation

Team 15

Aous Abdulla

Byron Morris

Eddie Barun

Ethan Griffiths

Theo Toth

Weiham Zhu

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1 ▾	Product ▾	Crashes in game due to bugs on Open Day	M ▾	H ▾	Allocate an adequate amount of time for testing and prototyping	Theo Byron Eddie Aous Ethan Weihan
R2 ▾	Technologies ▾	Problems with resolution, framerate, etc	L ▾	M ▾	Considering it is a 2D game, the scale function can be used	Theo Bayren Ethan
R3 ▾	Technologies ▾	Bug fixes and updates for game	L ▾	H ▾	Keep testing the game and provide gameplay to third parties in order to get the feedback needed and respond quickly in order to fix the bugs before presenting.	Theo
R5 ▾	Product ▾ Project ▾	Time constraint until release day which causes development of games to be made with haste	H ▾	H ▾	Allocating a specific time to test it and work on quality assurance	Theo Eddie Aous
R6 ▾	Technologies ▾	Incorrect score or score displayed incorrectly	L ▾	H ▾	Having all group members involved in testing the game after it finishes	Theo Byron Eddie Aous Ethan Weihan
R7 R7 ▾	Product ▾	Potential wrong registration	L ▾	L ▾	Test areas where errors are expected to	Theo

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
		of buttons (where the borders of things need to be)			happen (borders of items, different cursor actions, etc)	
R8 ▾	Product ▾	No fun factor	L ▾	H ▾	Add more customer interactions in game and make them more rewarding (Higher amount for money and prices) as well as adding vibrant colours and "balancing the art and technology"	Theo Byron Eddie Aous Ethan Weihan
R9 ▾	Project ▾	One of the participants becomes unavailable	L ▾	H ▾	Organise how to distribute the missing participant/s work amongst the rest of the group	Theo Byron Eddie Aous Ethan Weihan
R10 ▾	Project ▾	Jobs aren't divided based on skill set	L ▾	H ▾	Have group meetings and clearly dictate strong suits and weaknesses	Theo Byron Eddie Aous Ethan Weihan
R11 ▾	Project ▾	Inadequate identification of requirements	M ▾	H ▾	Having whole group meetings to discuss the requirements and researching which requirements apply to the project	Theo Byron Eddie Aous Ethan Weihan

M. Schmalz, A. Finn and H. Taylor, "Risk Management in Video Game Development Projects," *2014 47th Hawaii International Conference on System Sciences*, 2014, pp.

4325-4334, doi: 10.1109/HICSS.2014.534.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6759136>

The risk assessment began with a discovery phase, where possible risks of any kind were identified and recorded. After, a meeting was organised with the customer to specify a few details, expectations and requirements and figure out the setting and device that the game will be played in. Next, we spent the first few meetings discussing and recording the risks on a document. After initially identifying the risks, we decided to discuss new risks as well as changes to existing ones in all future meetings as the project progressed. This ensured full team cooperation and feedback on the risks and proved to be efficient as it was simpler to assign team members to risks based on their skills through the feedback.

Next, a risk register was designed for the risks identified. The group used simple classifications rated “low, medium, high” as we found it to be appropriate for a project of this size. At the time of creating the table, some time had passed since the discovery phase and the team was more familiar with the project, so many risks were reviewed and assigned their own likelihood and severity. This process took a few meetings to complete, however this guaranteed that the categorizations were as accurate as possible. Next, the decision of ownership was devised and written in the table.

The group decided on the ownership based on the likelihood of the risk and its complexity. This meant that a low possibility risk which is simple can be taken by one person, and since all of the risks are initially discovered and discussed, the mitigation of ‘division of labour based on skills’ made the ownership process quick and efficient. On the other hand, if a risk was higher than “low”, a minimum of two members were assigned to it and increased if the complexity was higher than the others. This helped mitigate several low categorised risks quickly as time was limited with risks such as verifying and checking keybinds.

A recurring theme throughout the project with this group has been efficiency without sacrificing quality. This meant that mitigation factors were dealt with as mentioned above by likelihood and complexity, however no time was to be saved if quality wasn't maximised. This in itself was a mitigating factor to the time constraint until the open day at the CS department. One risk that has occurred though, in which time efficiency was challenged was if one of the engineers became unavailable, which becomes more common in times of holidays, where most of the group members were travelling. This was quickly rectified, however, before the holiday as the frequency of meetings allowed us to assign a minimum of two people and catch them up on the information beforehand which minimised and shrunk the risk of a part of the project being incomplete at the specified time.

Furthermore, we eliminated the risk of incompatible code when different aspects of the code were divided between the group members. This was mitigated by organising the game code on github and utilising the merge request feature. This meant that at least two members had to test and approve the amended/added code before the merge process could occur.