

Risk Management Plan

Assignment 4 – Final Project Documentation

INFT575 Preparation of Final Business Case

Chris Murphy (CIT182880)

25 June 2019

Table of Contents

Risk Management Review Policy.....3

Risks, Mitigation Strategies & Contingency Plans.....4

 High-level risks.....4

 Medium-level risks.....4

 Low-level risks.....7

Risk Management Review Policy

The identified risks as well as the associated management strategies and contingency plans shall be reviewed, evaluated, and altered if need be on the 14th of August. This is the point at which development of non-documentation assets will be roughly halfway complete (after the base retro shaders are finished and before the retro post-processing shader development has started), and will be a good point to assess the effectiveness of the Risk Management Plan before it is too late for any changes to be meaningful.

Risks, Mitigation Strategies & Contingency Plans

High-level risks

Project files are deleted/lost:

Type: Infrastructure risk

Probability: Moderate

Impact: Extreme

Risk: High

Mitigation – the most obvious way to prevent this risk is to keep backups of all project files and assets. The main method of backup for this project will be using GitHub as source control – the repository in which the project is stored should be updated at the end of each development session to ensure that if the local copy is damaged/lost, only a small amount of development time will be lost alongside it. For additional protection, a secondary backup of the project should be kept on an external portable drive and updated every week.

Contingency – if files or assets are lost, the most important thing to remember is that the end product must be functional – due to the nature of the product as a set of assets with instructions, if there isn't enough time to develop all the planned assets than the number developed for the final product could be reduced in order to keep to the development timeline. This way, even if the final product ends up smaller in scope than intended all the components will at least be fully functional.

Medium-level risks

Project isn't finished on time:

Type: Time risk

Probability: Unlikely

Impact: Major

Risk: Medium

Mitigation – the project timetable/milestones have been built to allow an 18-day buffer of unassigned time at the end of the project to compensate for any slips in the project deadlines. Additionally, the development milestones have set dates which will make it obvious early if the project is starting to slip.

Contingency – see contingency plan for above risk.

Final product is overly buggy:

Type: Contractual risk

Probability: Unlikely

Impact: Moderate

Risk: Medium

Mitigation – a basic test plan will be created and applied throughout product development to help maintain quality throughout the life cycle.

Contingency – the product will have to be patched and fixed post-launch

Product is not explained/tutorialised effectively for new users:

Type: Contractual risk

Probability: Moderate

Impact: Moderate

Risk: Medium

Mitigation – 30 days of development have been dedicated specifically to developing assets which help explain and demonstrate effective use of the product. These include:

- Thorough internal and external documentation of all code
- External tutorial documentation explaining the use of each asset in the package
- An example Unity scene which demonstrates the implementation of each asset in the package

Contingency – any feedback regarding the product not being easy to understand should be analysed and any post-release patches should include measures to help the user learn and implement the product.

Product is copied and resold by someone else:

Type: Legal risk

Probability: Rare – Impact: Major – Risk: Medium

Mitigation – there are a couple of mitigation strategies that can be employed to help prevent this issue, the first being that all code written during the project should include a copyright disclaimer in the header to outline the applicable copyright laws. Additionally, GitHub recently changed their free version of source control so that free repositories can be made private, rather than having to pay a subscription to prevent them being private – this will help protect the code from any potential thieves.

Contingency – the Unity Asset Store has a process for dealing with asset theft. This should be utilised in order to get any stolen assets taken down as quickly as possible.

Product doesn't work on low-end hardware:

Type: Contractual risk

Probability: Unlikely

Impact: Major

Risk: Medium

Mitigation – I'm looking to buy a new laptop soon, so to help ensure that the product functions properly on low-end hardware the testing plan will also include running performance tests on this new laptop.

Contingency – any feedback regarding the product not running correctly should be analysed and any post-release patches should include measures that help fix this issue.

Product doesn't work on non-Windows versions of Unity:

Type: Contractual risk

Probability: Unlikely

Impact: Major

Risk: Medium

Mitigation – the testing plan will include running tests on borrowed computers running both Mac OSX and Linux alongside the Windows-based development machines to ensure all versions are functional.

Contingency – any feedback regarding the product not running correctly should be analysed and any post-release patches should include measures that help fix this issue.

Product isn't accepted for submission to the Unity Asset Store:

Type: Financial risk

Probability: Rare

Impact: Major

Risk: Medium

Mitigation – the best mitigation strategy is to read through the submission requirements for the Unity Asset store and ensure that the product complies fully before submitting it for review.

Contingency – if the product is rejected, the Unity Store reviewer will likely give the reason why. This/these issues should be addressed and the product re-submitted for review.

Market is too niche for product to make a profit:

Type: Financial risk

Probability: Likely

Impact: Low

Risk: Medium

According to a quick/very basic estimation, the amount of development hours put into the project will be worth roughly \$2,000 on Australian minimum wage. If the product is priced at \$20 per license, that means that at least 100 sales are required just to break even on

development costs. Additionally, because this product is aimed at a niche market it's difficult to estimate the potential sales range with any kind of accuracy. Luckily, the main aim of this project is to make a functional product that can be sold, rather than making a profit – as such, there aren't any negative impacts that would result if the product doesn't make a profit.

Low-level risks

Development hardware breaks during project:

Type: Infrastructure risk

Probability: Rare

Impact: Moderate

Risk: Low

Mitigation – take care of computer/s during development. The spare time allocated at the end of the project timeline will help soften the blow of any hardware issues that take more than a day to fix.

Contingency –replace any hardware that needs fixing.