|  |
| --- |
| CPSC 2720 Spring 2018 |
| BBG Adventure |
| Team Big Boys  Letter: G |
|  |
| **Gideon Richter – Justin Creig – Jesse Huss** |
| 16/02/2018 |

Table of Contents

[Revision History 3](#_Toc506213826)

[Introduction 4](#_Toc506213827)

[Project Management 5](#_Toc506213828)

[Team Organization 5](#_Toc506213829)

[Risk Management 5](#_Toc506213830)

[first step – identify 5](#_Toc506213831)

[second step – evaluate 6](#_Toc506213832)

[third step – eliminate 6](#_Toc506213833)

[Development Process 7](#_Toc506213834)

[Coding Conventions 7](#_Toc506213835)

[Code review 7](#_Toc506213836)

[first stage – personal 7](#_Toc506213837)

[second stage – peer 7](#_Toc506213838)

[third state – project 7](#_Toc506213839)

[team communication 8](#_Toc506213840)

[Software Design 8](#_Toc506213841)

[Design 8](#_Toc506213842)

[Design Rationale 8](#_Toc506213843)

[Appendices 8](#_Toc506213844)

[Appendix A 8](#_Toc506213845)

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision Date** | **Changes Made** | **Name** |
| 2/10/2018 | Added title page and logo | Gideon Richter |
| 2/11/2018 | Added introduction, team organization, code review guidelines, team communication | Gideon Richter |
| 2/12/2018 | Added risk management section | Jesse Huss |

# Introduction

BBG Adventure is a text-based adventure game developed as part of a group project for the course Practical Software Development & Design at the University of Lethbridge. The project has three major sections – Design, Implementation, and Testing. The members of the Big Boy team are: Gideon Richter, Jesse Husse, and Justin Creig.

The adventure plot revolves around a lost father, who, in search of directions approaches an abandoned manor house – only to never return. It is the Hero’s quest to enter the house and find out what happened.

Upon entering the house, the Hero discovers a mad scientist who has transformed the father into a duck with his Electro-object-polymorphizer. Unfortunately, after duck-ifying the father, the machine blew up and parts were sent flying throughout the grounds. Too busy with his other mad-scientist obligations, the Hero is enlisted to search for the three missing parts and return them.

In their search of the house, the Hero encounters rats, ghouls, and school of piranhas while traversing a dark, scary, dark-scary cellar, a spooky gravesite, and manor grounds.

Of course, not every adventurer will return from their adventure – be warned, there are many ways to die in this adventure, or end up locked in the house forever.

But let’s not dwell on failure, there are many ways to succeed! The Hero is provided with a save/load system that allows them pick up where they left off, or travel through time after coming to an untimely end. Furthermore, an in-game action manual provides the hero with all necessary (or unnecessary) text actions that the game can parse.

This document serves as an overview of the design process of the game thus far. The topics include:

* Project Management: Team Organization, productivity, and risk management
* Development Process: Code review and conventions, issue tracking, and team communication
* Software Design: UML class/sequence diagrams, use cases, design principles and patterns

# Project Management

Provide a description of and address any foreseeable problems. TODO: Intro: Summarize contents of sections subsections.

## Team Organization

A close up of a logo

Description generated with very high confidenceThe Big Boys team will be organized as a democratic or open structured team, where all members will have the same opportunity to dictate and participate in team activities.

Figure 1: Example democratic team structure

The team members will have the following roles:

* Team Lead: Gideon Richter; organize and keep everything on track
* Design Lead: Justin Creig; ensure adherence to good object-oriented design
* Quality Assurance Lead: Jesse Husse; ensures implementation of design specifications

Furthermore, all team members will fill the role of Software Developer and Software Tester, while contributing to project documentation.

## Risk Management

As with any project, there are risks involved that can hinder the completion of the project. Foreseeable risks will be dealt with in three steps. These steps help ensure that as many potential problems are dealt with before they occur.

### first step – identify

1. Identify the risks.
2. Identify what aspects of the project will be impacted as well as severity of the impact.
3. Identify likelihood of that risks occurring.
4. Categorize the risks. Categories may include but are not limited to:
   1. Time
      1. Scheduling issues/Busy schedules
      2. Underestimation in time to complete an aspect of the project.
   2. Technical
      1. Major design changes needed.
      2. Realize parts of project are out of scope.
   3. People
      1. Loss of a team member
      2. Family emergencies/ Illnesses
      3. Lack of knowledge or skill
      4. Not Contributing

### second step – evaluate

1. Evaluate potential risks based on likelihood and severity of the impact.
2. Identify the most prominent risks.
   1. Most severe impact and most likely to occur
   2. Less severe but likely to occur
   3. Severe but unlikely to occur
   4. Minimally severe and least likely to occur

### third step – eliminate

1. Remove the risk
   1. Removing as many high severity or likely risks as possible is key for maintaining a productive project.
   2. Easy risks to remove can include:
      1. Design flaws caught early. i.e. Out of scope additions
      2. Lacking knowledge or skill. Discuss before hand to gain a sense of what each team member is capable of and assign tasks from there.
2. Reduce the risk
   1. Reducing the severity and/or likeliness of a risk occurring is second to removing the risk altogether because although the risk is still there, it becomes less of an issue and is much more manageable. Examples include:
      1. Deal with scheduling prior to starting the project. i.e. Use a scheduling software to see when each team member is free. Conflicts may still arise later but the risk was reduced.
      2. Discuss overall difficulty and size of the planned project to determine if it is attainable.
3. Plan for risks
   1. Plan to deal with any risks that may still occur. This may include:
      1. Creating internal deadlines that provide sufficient time before official deadlines for team review/catch up.
      2. Break project up into parts. i.e. Key functionality/requirements, features that can be added once core of project is complete if time allows.

# Development Process

## Coding Conventions

TODO: JUSTIN

## Code review

Code review will occur in three stages. These stages help ensure that code entering the production codebase of the game is up to standard and is bug-free.

### first stage – personal

Prerequisites:

* Code should compile
* Code should be free of warnings
* Code should follow project coding conventions
* New functionality should be accompanied by (non-exhaustive) unit tests
* Existing unit tests broken by new code should be fixed, replaced, or deleted

If prerequisites are met:

1. Test local merge with up-to-date develop branch and ensure no conflicts
2. A GitLab pull request should be made develop branch
   1. with sufficient description
   2. with no merge conflicts
   3. with no unnecessary/temporary files
   4. and assigned to at least one other group member

### second stage – peer

1. Visually review changes
2. Locally checkout branch
3. Locally merge reviewed branch with up-to-date develop branch
4. Run unit tests
5. Merge pull request into develop branch

### third state – project

* Periodically, all new changes are reviewed again and merged into master branch
* This ensures that the master branch is always stable not directly modifiable (unless in extraordinary circumstance)

## team communication

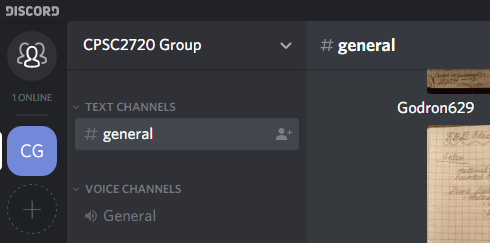
Team communication will occur mainly through a discord channel. Other methods such as texts and emails are also acceptable. Communication regarding code review should take place on GitLab pull requests and issue tracking.

Figure 2: discord channel

# Software Design

## Design

## Design Rationale

# Appendices

## Appendix A