**Software Engineering Group Project**

**Design Specification**

|  |  |
| --- | --- |
| Author: | wgf |
|  |  |
| Date: | 2017-03-16 |
| Version: | 0.2 |
| Status: | Development |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Department of Computer Science

Aberystwyth University

Aberystwyth Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2017

Contents

[1 Introduction 3](#_Toc477426987)

[1.1 Purpose of this Document 3](#_Toc477426988)

[1.2 Scope 3](#_Toc477426989)

[1.3 Objectives 3](#_Toc477426990)

[2 Decomposition Description 3](#_Toc477426991)

[2.1 Programs in system 3](#_Toc477426992)

[2.2 Significant classes in each program 3](#_Toc477426993)

[2.3 Mapping from requirements to classes 3](#_Toc477426994)

[3 Dependency Description 4](#_Toc477426995)

[3.1 Component Diagrams 4](#_Toc477426996)

[4 Interface Description 4](#_Toc477426997)

[4.1 Class 1 Interface specification 4](#_Toc477426998)

[5 Detailed Design 4](#_Toc477426999)

[5.1 Sequence diagrams 4](#_Toc477427000)

[5.2 Significant algorithms 4](#_Toc477427001)

[5.3 Significant data structures 4](#_Toc477427002)

[6 Document History 4](#_Toc477427003)

# 1 Introduction

## 1.1 Purpose of this Document

The purpose of this Document is to show the design of the system for creation of the actual system. It will aid the system by providing a guide to the implementation. The document will

## 1.2 Scope

This document specifies the design of the system. It will describe the layout and design of the system.

## 1.3 Objectives

The objective of this document is to aid the implementation of the system. It will provide a guide to implementation the system through the use of sequence diagrams, significant algorithms, and significant data structures.

# 2 Decomposition Description

## 2.1 Programs in system

The system that we are designing is all included into one program.

## 2.2 Significant classes in each program

The GUI class is the graphical interface that shows displays all of the game running. It displays the board, with the islands and ports, the players’ locations, and the current score of all the players.

The Gameboard class controls all of the locations for the various spaces on the board, the location of the islands and the location of the ports.

The Player class keeps track of the player’s score, their homeport and their crew and chance cards. They are also assigned a ship object from the ship class which is linked through the owner private variable in the ship class

The Turntracker class keeps track of which players’ turn it is.

## 2.3 Mapping from requirements to classes

|  |  |
| --- | --- |
| Functional Requirement | Classes providing requirement |
| FR1 |  |
| FR2 |  |
| FR3 |  |
| FR4 |  |
| FR5 |  |
| FR6 |  |
| FR7 |  |
| FR8 |  |
| FR9 |  |
| FR10 |  |
| FR11 |  |
| FR12 |  |
| FR13 |  |
| FR14 |  |
| FR15 |  |
| FR16 |  |

# 3 Dependency Description

## 3.1 Component Diagrams

# 4 Interface Description

## 4.1 Class 1 Interface specification

# 5 Detailed Design

## 5.1 Sequence diagrams

## 5.2 Significant algorithms

## 5.3 Significant data structures

# 6 Document History

# 

|  |  |  |
| --- | --- | --- |
| Version | Changes | Changed by |
| 0.1 | Initial creation | WGF |
| 0.2 | Basic addition | WGF |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |