

System design document for Maze (SDD)

1	Introduction	2
1.1	Design goals.....	2
1.2	Definitions, acronyms and abbreviations	2
2	System design	3
2.1	Overview	
2.1.1	The model functionality	
2.1.2	Rules	
2.1.3	Unique identifiers, global look-ups	
2.1.4	Spaces.....	
2.1.5	Event handling	
2.1.6	Internal representation of text.....	
2.2	Software decomposition	
2.2.1	General	
2.2.2	Decomposition into subsystems	
2.2.3	Layering.....	
2.2.4	Dependency analysis.....	
2.3	Concurrency issues	
2.4	Persistent data management.....	
2.5	Access control and security	
2.6	Boundary conditions	
3	References	6

Version: Maze 0.1

Date:

Author:

Matilda Andersson

Karin Wibergh

Johan Ärlebrandt

Jonathan Johansson

1 Introduction

1.1 Design goals

1.2 Definitions, acronyms and abbreviations

- GUI, graphical user interface.
- Java, platform independent programming language.
- MVC, a way to partition an application with a GUI into distinct parts avoiding a mixture of GUI-code, application code and data spread all over.

2 System design

2.1 Overview

The application is decomposed into these parts:

- Main - Contains the main class to start the application

- Model - Contains all the model classes for the MVC

- View - Contains all the classes that display something, the View part in MVC

- Controller - Contains all the control classes for the MVC

- Util - Contains IO-classes and constants

2.2 Software decomposition

2.2.1 General

Package diagram. For each package an UML class diagram in appendix

2.2.2 Decomposition into subsystems

2.2.3 Layering

2.2.4 Dependency analysis

2.3 Concurrency issues

NA - single-threaded application.

2.4 Persistent data management

Information about the save slots and highscores is saved in a file called Game.ser.

2.5 Access control and security

NA

2.6 Boundary conditions

3 References

APPENDIX