**1.0 Introduction**

This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.

**1.1 Goals and objectives**

The goal with this software is to provide a fun and family oriented Trivia Game that is playable on most Windows systems.

**1.2 Statement of scope**

The current implementation allows game play to execute from a Windows console. A player will input the direction to move and an answer to a given trivia question. Output will consist of a text display of the maze (including player position and exit location) as well as text formatted to accommodate a given question.

**1.3 Software context**

This software will not be placed in any market, but will be submitted for grading for Team Macrosoft.

**1.4 Major constraints**

Ultimately we would like to implement a GUI or colored-text within the console. A database has not yet been implemented. We may not release this game for sale due to copyright issues of the trivia questions.

**2.0 Usage scenario**

This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.

This implementation presents the game in the format of a maze. The game player(s) must traverse from the entrance / beginning to the exit / end of the maze. To move in any direction (from one room to another) a player must answer a random trivia question. A correct answer allows movement and an incorrect answer bars any future access to that door. Good Luck!

**2.1 User profiles**

A user profile will consist of only a player(s) name.

**2.2 Use-cases**

* A user will be able to move through the maze, room by room, passing through a door.
* Upon acessing a door a user will be prompted with a trivia question and be provided a method for answering.
* To make the game interesting an incorrect answer will bar any future access to a given door.
* If no access to the entrance is available it is GAME OVER.
* If a user reaches the EXIT room they win.
* A user will be presented with end of the game statistics

**2.3 Special usage considerations**

If we decide to color text within the console we will have to include the required packages here.

**3.0 Data Model and Description**

This section describes information domain for the software

**3.1 Data Description**

Data objects that will be managed/manipulated by the software are described in this section.

**3.1.1 Data objects**

* Game
  + Will provide the implementation and setup of a new game
* Maze
  + Provides the interface for: accessing the maze, validating user inputs, verifying that the maze is still solvable, and drawing the maze for user-friendly reading
* Player
  + Provides the interface for: creating a new player by name, accessing player data, and storing player data statistics.
* Coordinates
  + Provides an interface for keeping track of a player's location within the maze
* Room
  + Serves as a data-wrapper for a collection of 4 door objects.
* Door
  + Provides an interface for: allowing / denying access to a door, retrieving the trivia item related to a door, retrieving explicit door values.
* Trivia Item
  + Serves as an abstract class to guarantee an interface the implements the following: retrieving and setting trivia data, providing a trivia question, providing a trivia answer
* Direction
  + Provides an enumeration type for simple identifying of a player's requested direction to move.
* Maze Builder
  + Provides a simple interface for creating and retrieving a new maze object.
* Trivia Factory
  + Provides an interface that implements the following: guarantee that no trivia question will be asked twice and allow retrieval of a trivia question.

**3.1.2 Relationships**

Relationships among data objects are described using an ERD- like form. No attempt is made to provide detail at this stage.

**3.1.3 Complete data model**

An ERD for the software is developed

**3.1.4 Data dictionary**

A reference to the data dictionary is provided. The dictionary is maintained in electronic form.

**4.0 Functional Model and Description**

A description of each major software function, along with data flow or class hierarchy (OO) is presented.

**4.1 Description for Function n**

A detailed description of each software function is presented. Section 4.1 is repeated for each of n functions.

**4.1.1 Processing narrative (PSPEC) for function n**

A processing narrative for function n is presented.

**4.1.2 Function n flow diagram**

A diagram showing the flow of information through the function and the transformation it undergoes is presented.

**4.1.3 Function n interface description**

A detailed description of the input and output interfaces for the function is presented.

**4.1.4 Function n transforms**

A detailed description for each transform (subfunction) for function n is presented. Section 4.1.4 is repeated for each of k transforms.

**4.1.4.1 Transform k description (processing narrative, PSPEC)**

**4.1.4.2 Transform k interface description**

**4.1.4.3 Transform k lower level flow diagrams**

**4.1.4.4 Transform k interface description**

**4.1.5 Performance Issues**

Special performance required for the subsystem is specified.

**4.1.6 Design Constraints**

Any design constraints that will impact the subsystem are noted.

**4.2 Software Interface Description**

The software interface(s)to the outside world is(are) described.

**4.2.1 External machine interfaces**

Interfaces to other machines (computers or devices) are described.

**4.2.2 External system interfaces**

Interfaces to other systems, products or networks are described.

**4.2.3 Human interface**

An overview of any human interfaces to be designed for the software is presented.

**4.3 Control flow description**

The control flow for the system is presented with reference to Section 5.0 of this document.

**5.0 Behavioral Model and Description**

A description of the behavior of the software is presented.

**5.1 Description for software behavior**

A detailed description of major events and states is presented in this section.

**5.1.1 Events**

A listing of events (control, items) that will cause behavioral change within the system is presented.

**5.1.2 States**

A listing of states (modes of behavior) that will result as a consequence of events is presented.

**5.2 State Transition Diagrams**

Depict the overall behavior of the system.

**5.3 Control specification (CSPEC)**

Depict the manner in which control is managed by the software.

**6.0 Restrictions, Limitations, and Constraints**

Special issues which impact the specification, design, or implementation of the software are noted here.

**7.0 Validation Criteria**

The approach to software validation is described.

**7.1 Classes of tests**

The types of tests to be conducted are specified, including as much detail as is possible at this stage. Emphasis here is on black- box testing.

**7.2 Expected software response**

The expected results from testing are specified.

**7.3 Performance bounds**

Special performance requirements are specified.

**8.0 Appendices**

Presents information that supplements the Requirements Specification

**8.1 System traceability matrix**

A matrix that traces stated software requirements back to the system specification.

**8.2 Product Strategies**

If the specification is developed for a product, a description of relevant product strategy is presented here.

**8.3 Analysis metrics to be used**

A description of all analysis metrics to be used during the analysis activity is noted here.

**8.4 Supplementary information (as required)**