Architecture, Design and Coding standards

Team 7 April 19, 2020

Team Members
Matteo Corsi
Riccardo Convertino
Tommaso Osti
Riccardo Scardoni

Document Control

Change History

Revision	Change Date	Description of changes
v1.0	19/04/2020	Initial release

Document Storage This document is stored in the project documentation's GitHub repository at: https://github.com/Fr0stBob/2020-5Ai-team7-Corsi/tree/master/docs

Document Owner Convertino Riccardo is responsible for developing and maintaining this document.

1 Introduction

The Team 7 architecture document was designed to illustrate and identify the high-level architecture systems used to design and implement the Web Server application. The document contains an overview of the classification of the components used.

2 Components Classification

2.1 Presentation Layer

Purpose: To display forms, controls and images to the user to create a fluid and efficient user experience.

Specific Nature: The presentation layer will be in charge of displaying appropriate images and menus to the user. When a user clicks a menu on the GUI, the code corresponding to that event will be called. This layer will also be in charge of the spawning of appropriate threads. The need of spawning extra threads is due to the fact that the main thread of the app will be watching for event clicks, but we also need another thread constantly running to get the users current position.

Subcomponents: Image Viewer – That it is used during the Walking Tour and during the Interactive Map mode of the application. Its responsibility is to display the appropriate image as determined by the Landmark held in the Current State.

2.1 Database Layer

Purpose: This layer is in charge of storing data in persistent storage.

Specific Nature: This layer will consist of SQL files. These together will be our database management system. These SQL files will store all the data needed for the application.

2.1 Data interaction Layer

Purpose: This level is used to pass data from the database to the application.

3 Design Goal

Our priorities for the design are:

- The project must minimize complexity and development effort.
- The project must take into account the development environment, which consists of 1 small team with complementary skills working in time and space. Ideally, the project should lead to 3-4 components of equal size and complexity. If the components have well-defined interfaces each team member can work independently by coding the interfaces of the other components. The concerns of each component should be narrowed down so that each member can specialize in a particular technology or skill.

Coding Standards

1 Source file basics

1.1 File name

The source file name consists of the case-sensitive name of the scope of the file, plus the extension.

1.2 File Encoding: UTF-8

Source files are encoded in **UTF-8**.

2 Source file structure

A source file consists of, **in order**:

1. License or copyright information, if present
2. Package statement
3. Import statements
4. Contents of the file depending on the extension

3.1 License or Copyright information, if present

If license or copyright information belongs in a file, it belongs here.

3.2 License or Copyright information, if present

The package statement is **not line-wrapped**.

3.3 Import statements

3.3.2 No line-wrapping

Import statements are **not line-wrapped**.

3.3.3 Ordering and spacing

Imports are ordered as follows:

1. All imports in a single block.

If there are both static and non-static imports, a single blank line separates the two blocks. There are no other blank lines between import statements.

4 Line-Wrapping

When code that might otherwise legally occupy a single line is divided into multiple lines, this activity is called *line-wrapping*.

There is no comprehensive, deterministic formula showing *exactly* how to line-wrap in every situation. Very often there are several valid ways to line-wrap the same piece of code.

5 Braces are used where optional

Braces are used with if, else, for, do and while statements, even when the body is empty or contains only a single statement.

6 Block indentation: +4 spaces

Each time a new block or block-like construct is opened, the indent increases by four spaces. When the block ends, the indent returns to the previous indent level. The indent level applies to both code and comments throughout the block.