Design Document

Connor Ahearn, Micah Arndt and Kevin Chan

Table of Contents

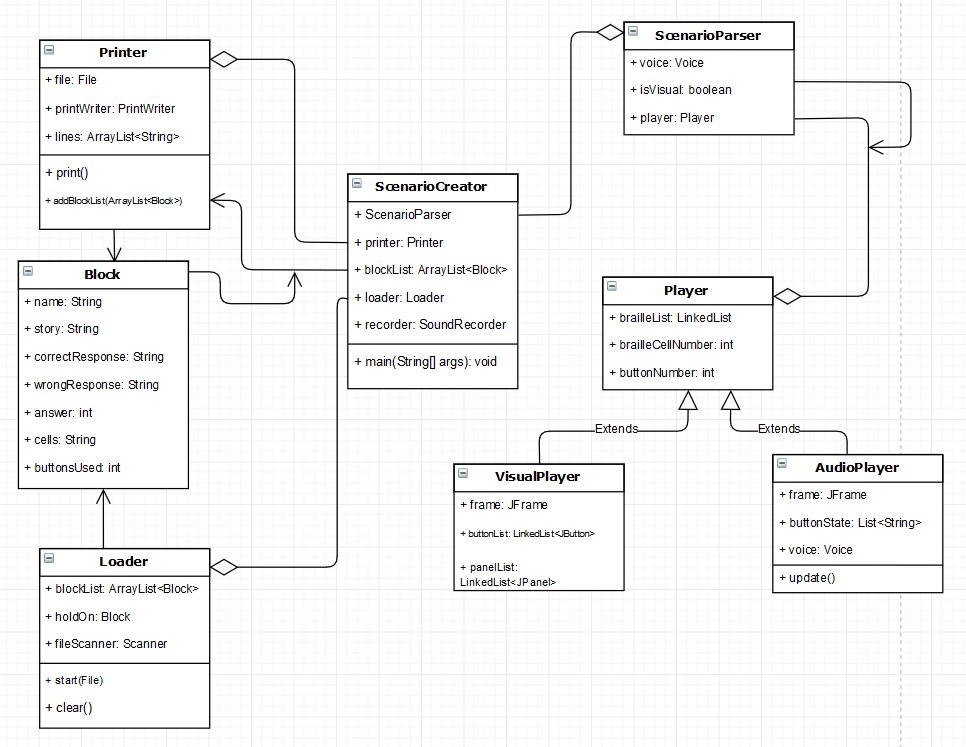
1. Introduction
2. Class Diagram
3. Sequence Diagram
4. Introduction

The purpose of this document is to describe the way the application operates at a high level. This is done using a class diagram and a sequence diagram.

The runtime of the application is much easier understood if you look at it from the perspective of **ScenarioCreator**. ScenarioCreator is the class with the main method, it has all the GUI’s for creating scenarios along with code to launch testing scenarios in our app. From start-up to close, if the Authoring App is running, ScenarioCreator is running. Through Aggregation, ScenarioCreator (As seen in the class diagram) uses every class in the application, although not always directly.

1. Class Diagram

The class diagram for the Authoring app can be seen below:



While looking at the diagram above, its important to note that some classes didn’t make it onto the diagram. These classes were omitted to make the diagram easier to understand and were deemed less important than the ones you’re seeing above. The omitted classes are listed below:

* BrailleCell.java
* BrailleCellPanel.java
* BrailleInterpreter.java
* SoundRecorder.java
* ToyAuthoring.java
* All Exceptions used throughout the program

Most of these classes are unessential to understand at a high level or are straight forward enough that they don’t need to be in the diagram.

The focus of the program can be seen in 3 main areas:

* ScenarioCreator – The runtime of the entire program runs through this
* Block – The data type universally used throughout the program’s working parts
* ScenarioParser – The Testing of scenarios is all done through this program

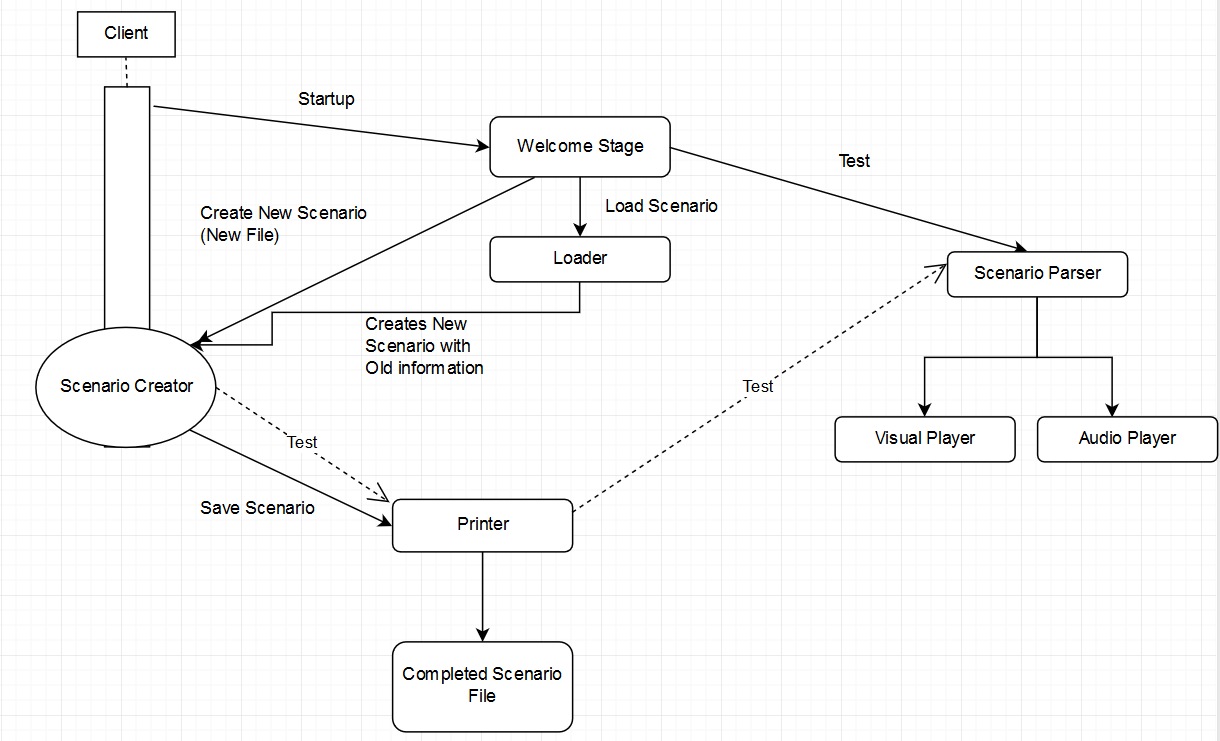
Blocks are the building block of scenarios in the application. The time frame of the box speaking up to the user pressing a button for the scenario is all stored in a block. This makes each scenario just a collection of blocks played in sequence by the braillebox. Having a standard datatype makes transferring data throughout the 3 main classes that use it (ScenarioCreater, Loader and Printer) much easier in practice.

ScenarioCreator was originally going to be the GUI used to create scenarios, but it evolved into being the central class of the application. All scenarios are written within its GUI, and advanced features and functions are completed via aggregation of other classes. Printer.java is used for creating new files, and Loader.java is used for loading existing files into ScenarioCreator.

ScenarioParser is central to testing scenarios. This code was provided to us by the professor.

1. Sequence Diagram

The sequence diagram of the Authoring App can be seen below.



The rectangle above / through Scenario Creator is the portion of the application that runs through that class that isn’t necessarily to do with scenario creation. It takes you to the welcome stage GUI where you’re given 3 options; New file, Edit file and Test file. Each of these options are arrows branching off the Welcome stage to the various areas of the program.