IDT 2013: McLean High School

# System State Recognition and Modeling

## Requirements & dOCUMENTATION

### Akshay K.

### peter O.

### Drew s.

# Introduction

Team Name: mhs1\_winter2013

Team Members: Akshay Karthik, Peter Ott, Drew Sorrels

## Running the system

The Application can be run in two different ways.

1. Double Click the executable JAR file (“IDT\_2013.jar”)
2. Build it from Source
   1. Open the project in eclipse via (File>Import)
   2. Press F11 under default settings or whatever your assigned build action is.

To use the application:

1. Click "File" In the top left to open the drop down dialogue box.
2. Click "Open Images" and a file explorer should appear.
3. Highlight all of the images to analyze.
4. Click "Open" and the application will analyze the images.
5. The application will display the changes as well as write a report in plain text format that is written to the image directory.

## Understanding the report

The report follows a very specific format:

# Requirement TRaceability

First, the examples given were analyzed by hand, documenting the specific state changes we expected our application to recognize. Then our application was tested on both SEQUENCE\_1 and SEQUENCE\_2 and the reports were compared to determine classification correction.

In order to improve our development and testing process, we utilized the popular Java Testing framework JUnit. We developed three sets of tests which were run on each compile, reducing the time it took to fix errors and improving developer performance. These tests are included in the tests folder and require JUnit to run. To run these tests, open the project in eclipse, select “AllTests.java” and press build.

1. Unit Tests: These tests assured that each class, data container, or utility method worked as promised
2. System Testing: These tests assure that each component works by itself
3. Integration Testing: These tests ensure that each component works in conjunction with other components to perform its function.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Source Code | Test File | Description |
| The system shall accept a sequence of png images of size 1280x1024 as input. | Config.ini  Config.java | ConfigTest.java | We have a configuration class that stores default values such as the size of the image. It will return display an error when the images are the wrong size. |
| The system shall report back to the user information regarding transition changes between images in the sequence. | Report.java | ReportTest.java | It will display the text in the application as well as store a save file in the application directory. |
| The system shall quantify transition changes in the report. | Recognizer.java | RecognizerTest.java | The application shall return ”Bounds” to quantify location of changes. |
| The system shall classify transition changes in the report. | Recognizer.java  ClassificationType.java | RecognizerTest.java  ClassificationTypeTest.java | The system will return enumeration “ClassificationType” which will be displayed in the report. |
| The system shall describe the rectangular boundary of each transition change that is found (either graphically or with text) as output. | Bounds.java | BoundsTest.java | Bounds is a container that describes |
| The system shall recognize window opening/closing. | WindowStateProcessor.java | WindowStateProcessorTest.java | A binary |
| The system shall recognize window moving/resizing. | WindowStateProcessor.java | WindowStateProcessorTest.java |  |
| The system shall recognize menus opening/closing. | WindowMenuProcessor.java | WindowMenuProcessorTest.java |  |
| The system shall recognize menu item selection. | WindowMenuProcessor.java | WindowMenuProcessorTest.java |  |
| The system shall recognize application area updates. | WindowChangeProcessor.java | WindowChangeProcessorTest.java |  |
| The system shall recognize window title bar clicks. | WindowChangeProcessor.java | WindowChangeProcessorTest.java |  |
| The system shall recognize window title updates. | WindowChangeProcessor.java | WindowChangeProcessorTest.java |  |
| The system shall recognize desktop icon changes. | DesktopTaskbarChangeProcessor.java | DesktopTaskbarChangeProcessorTest.java |  |
| The system shall recognize taskbar updates. | DesktopTaskbarChangeProcessor.java | DesktopTaskbarChangeProcessorTest.java |  |
| The system shall ignore updates to the clock (time and date). | DesktopTaskbarChangeProcessor.java | DesktopTaskbarChangeProcessorTest.java |  |