Localization Tool

Planning Document

Assignment 1

SP1 2017 Introduction to C# and Tools Development

Academy of Interactive Entertainment

Author Sam Cartwright

Contact helponline@aie.edu.au

Executive Summary

This document details the proposed architecture for a localization tool to be used to assist in game development.

This tool will allow the user to enter a word or phrase in English that will be used as text (user interface, dialogue, instructions, etc.) within a game. Users can enter translations in a number of different languages for a specific English entry.

The program will compile a local database of the data entered that can be imported into a game. Data will be saved in a Unicode text file, in comma separated format (csv).

Screen Mock-Ups

Main Window

Figure 1. Shows the main interface for the localization tool.

The mock-up shown in Figure 1 shows the main window for the tool.

Text can be entered via the table. New rows will automatically be added to the bottom of the table when all existing rows are full. Any empty row will automatically be deleted.

The user can scroll the table using the scroll bar on the right.

Buttons at the bottom of the window allow the user to save and load data files.

Dialogues

Saving and Loading dialog boxes are displayed when the save or load buttons (respectively) are pressed. Figure 2 shows the Save dialog box, and Figure 3 shows the Load dialog box.

Figure 2. The Save dialogue box.

Figure 3. The Load dialogue box.

If the user tries to load a file that is in the correct format or that has become corrupted, the Error dialogue box shown in Figure 4 will be displayed.

Figure 4. An Error message dialogue box displayed when loading fails.

Class Diagram

Figure 5. Class Diagram

The class diagram shows the main classes used in the tool.

The main program will store a collection of UIText objects, which contain the English word or phrase along with a collection of translations for the languages specified.

Ancillary classes for dialog boxes and other utility classes are not shown.

State Diagram

The State Diagram shown in Figure 6 shows the various program states and the transition between them.

The user will spend the majority of their time in the ‘Enter Text’ state, only transitioning to the other program states when saving or loading the application data.

Figure 6. State Diagram

Sequence Diagram

The sequence diagram shown in Figure 7 shows how UIText elements are created or updated in response to the user entering text via the main program user interface.

Figure 7. Sequence Diagram showing the updating of UIText objects in response to user input.

Save Data Format

Data will be saved in a comma-separated value format (.csv)

The first entry will be the English phrase or word, followed by translations in the desired languages. Where a translation for a specific language hasn’t been added, a space will be inserted instead.

An example of a data file saved by this program is shown in Figure 8.

Figure 8, Saved data format