The University of Western Ontario

London, Ontario, Canada

Department of Computer Science

# CS 4482a – Video Game Programming and Engine Development

**TOOLS 2** – Language and Localization

## 1 Introduction

Not everybody in the world speaks English. It's very likely that, if you want your game to be successful, you will release your game in North America, Japan, Europe, and all manner of foreign country. Usually you will want to ensure the gamers in these countries understand what your game is all about; this is where localization comes in. You must create a system that will not only ease editing of text, but also allow the game to switch seamlessly between languages. Your system should do the following:

1. Use the Unity3D Editor API to create your tools.
2. Define a set of *ScriptableObject* classes that will support 1...N languages containing 1...N strings each.
3. Write a collection of *CustomEditor*s that make it easier to create, add to, and edit Languages and their key-value pairs.

   **Note**: You may choose to author your data outside of Unity. However, if you must implement a *ScriptedImporter* [1] that will generate equivalent *ScriptableObject*s (*.asset*) with *CustomEditor*s to visualize your localization data.
4. Use the Unity UI system to create a text box that utilizes your language data. You can reference string by name, id, or other.
5. Your text box contents must update when you change the selected language of your engine.

   **Hint**: Provide a language selector in your custom editor that changes the language.

## 2 Miscellaneous

If you would like to write your editor tools in another engine, SDK, language, or otherwise, I am open to other possibilities. Please send me an e-mail and we'll discuss alternatives.

[1] *ScriptedImporter* - https://docs.unity3d.com/Manual/ScriptedImporters.html