Lappeenranta University of Technology
CT30A9300 .NET Code Camp

Word Cards

Code Camp Report

Lappeenranta, Feb. 2, 2011.

Group 6

Contents

Introduction

This paper documents Word Cards, an interactive learning software created during the 2011 .NET Code Camp, held in Lappeenranta University of Technology. The paper is structured into four parts: it first explores the pedagogic problem behind the software and then proceeds to explain the solution and its implementation. The last part summarizes the project in addition to proposing future improvements.

The Problem

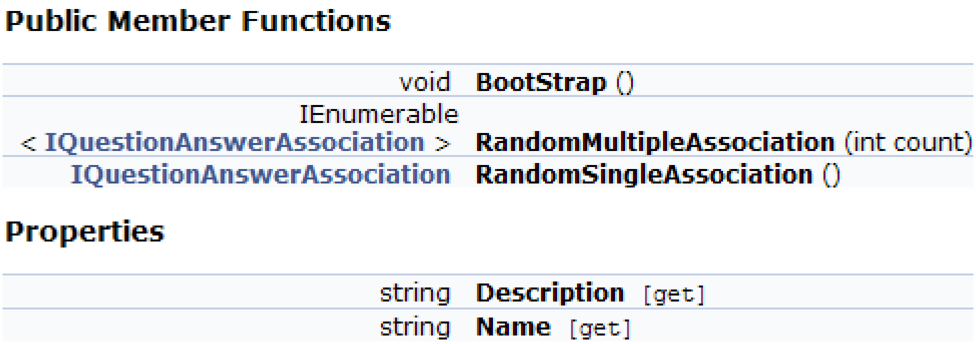
Words are a fundamental part of any language. Knowing the meanings of individual words not only enables us to communicate but also makes it possible to consume and produce written material. Integral part of this learning process is the ability to bind words with different meanings. This learning process should be made accessible through the use of interactive software that enables learners to associate words with descriptions or images.

The Solution

To learn new words, the learner should be able to associate the words with their meanings, either through other languages, words or images. Word Cards is a learning game that enables this learning process by providing a framework for dynamically loading association dictionaries that provide association mappings. The mappings can be presented as combinations of words and images, e.g. from image to image or from image to word.

The Implementation

Word Cards is implemented as .NET class libraries with initial graphical user interface (GUI) implementations in Windows Presentation Foundation (WPF), Windows Forms (Winforms) and Silverlight. Thus, also the core framework has .NET (CLR 4.0) and Silverlight (version 4) implementations. The framework provides a pluggable model for dynamically loading dictionaries by scanning assemblies for predefined interfaces, shown in figure 1 and figure 2. This allows developers to create rich dictionaries that can harness different association sources. For example, during the code camp, a REST web service based dictionary was created to fetch words and their descriptions in real-time from an online dictionary.



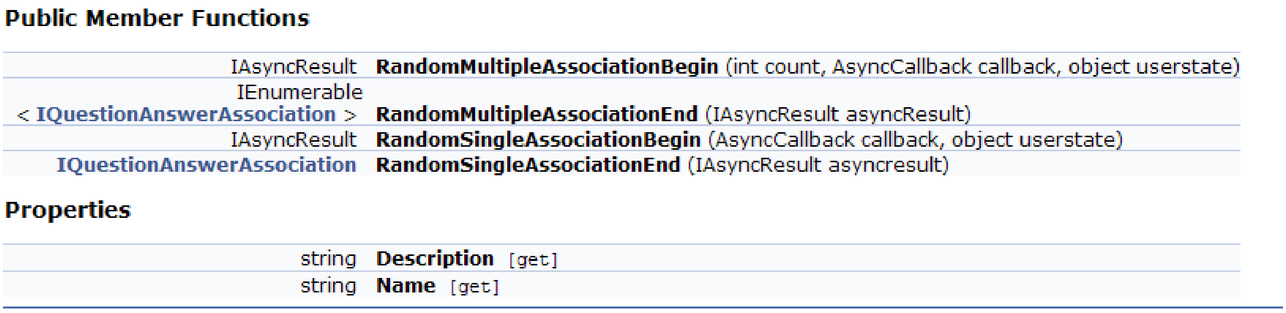
Public Member Functions

void	**BootStrap** ()
IEnumerable < IQuestionAnswerAssociation >	**RandomMultipleAssociation** (int count)
IQuestionAnswerAssociation	**RandomSingleAssociation** ()

Properties

string	**Description** [get]
string	**Name** [get]

Figure 1. Interface for .NET 4.0 dictionary.



Public Member Functions

IAsyncResult	**RandomMultipleAssociationBegin** (int count, AsyncCallback callback, object userstate)
IEnumerable < IQuestionAnswerAssociation >	**RandomMultipleAssociationEnd** (IAsyncResult asyncResult)
IAsyncResult	**RandomSingleAssociationBegin** (AsyncCallback callback, object userstate)
IQuestionAnswerAssociation	**RandomSingleAssociationEnd** (IAsyncResult asyncresult)

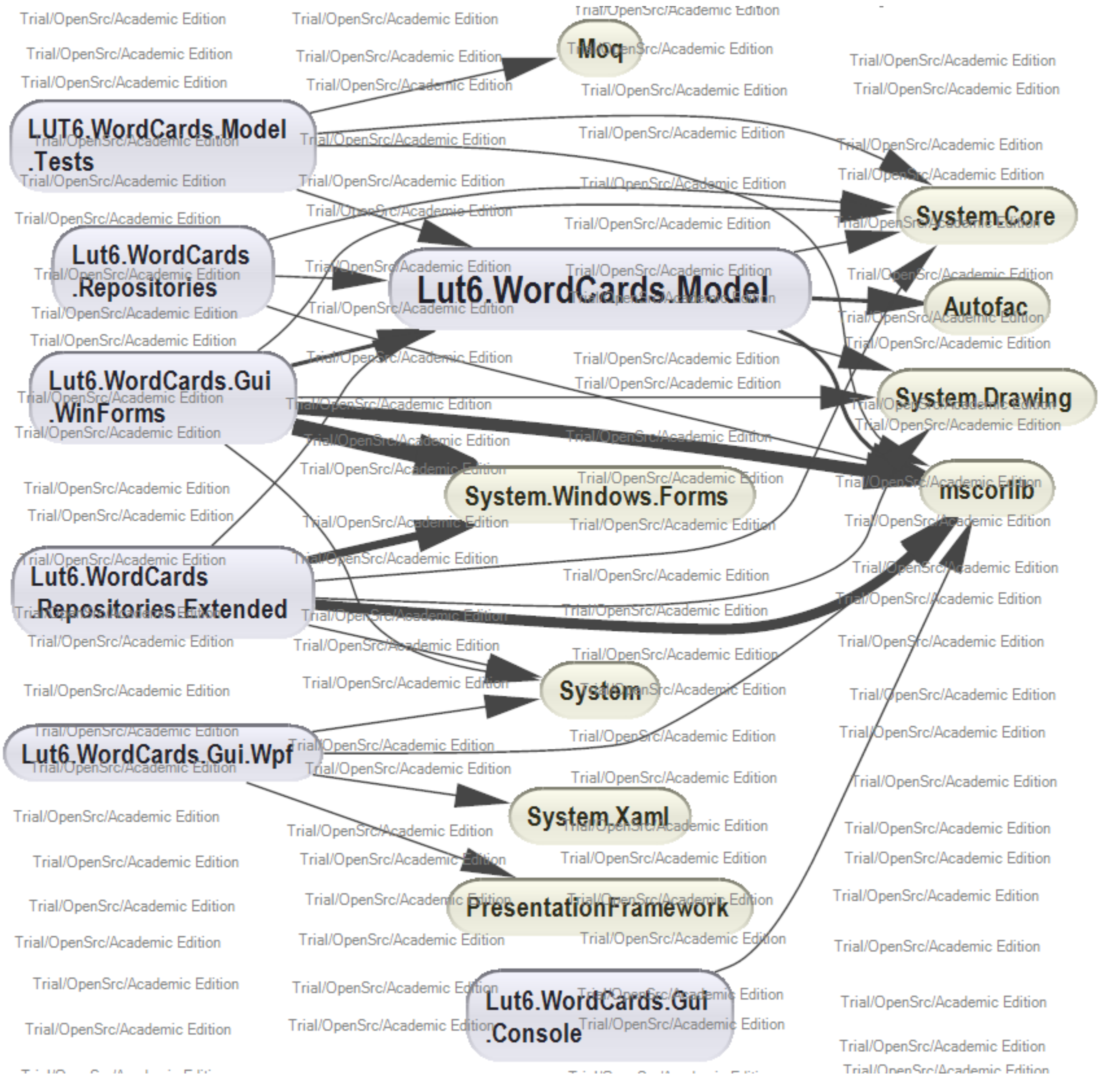
Properties

string	**Description** [get]
string	**Name** [get]

Figure 2. Interface for Silverlight 4 dictionary.

In addition to .NET 4.0 base class libraries (BCL) and Silverlight 4 that were used in the base framework implementation, Windows Forms, WPF and Silverlight were used in the GUI creation. Finally, Autofac was used as a dependency injection container (DI container).

The pluggable model in the framework implementation allowed for flexible design of user interfaces. Furthermore, it helped to achieve low afferent and efferent coupling as shown respectively in figure 3 and figure 4.



LUT6.WordCards.Model .Tests
Lut6.WordCards .Repositories
Lut6.WordCards.Gui .WinForms
Lut6.WordCards .Repositories.Extended
Lut6.WordCards.Gui.Wpf
Moq
Lut6.WordCards.Model
System.Windows.Forms
System
System.Xaml
PresentationFramework
Lut6.WordCards.Gui .Console
System.Core
Autofac
System.Drawing
mscorlib
Trial/OpenSrc/Academic Edition

Figure 3. Word Cards .NET 4.0 domain model and afferent coupling.

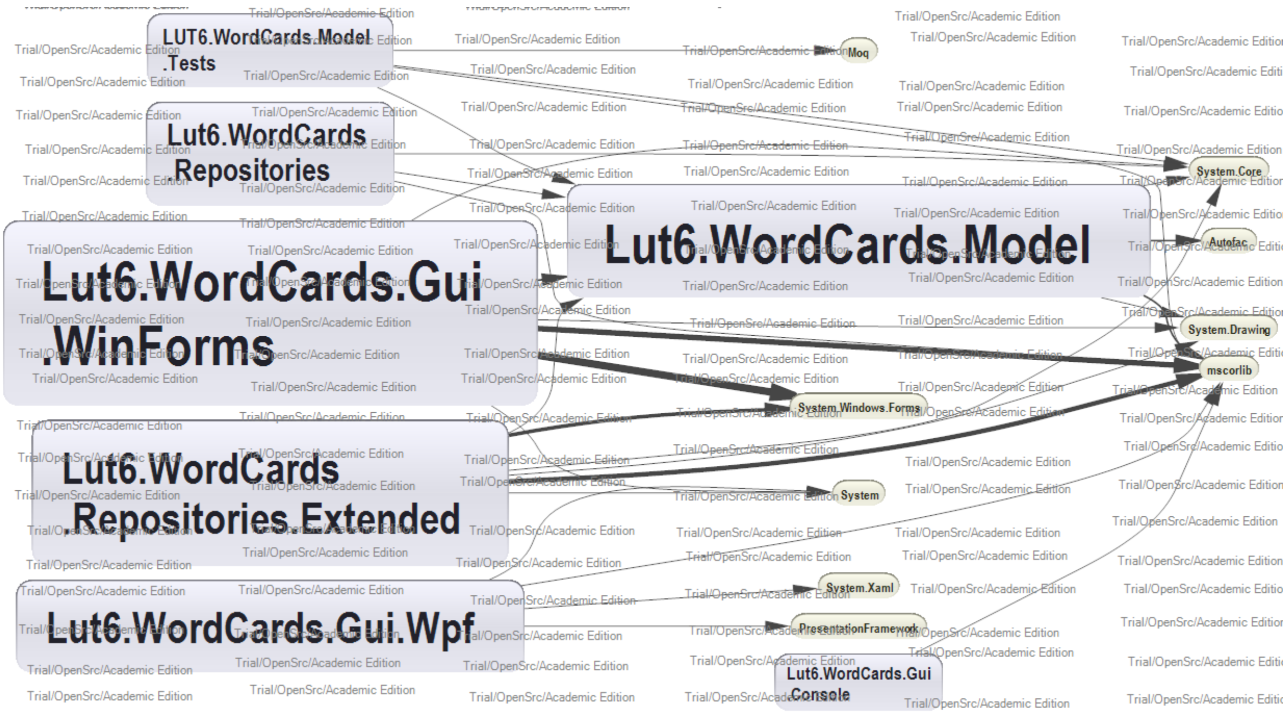

LUT6.WordCards.Model .Tests
Lut6.WordCards Repositories
Lut6.WordCards.Gui .WinForms
Lut6.WordCards .Repositories.Extended
Lut6.WordCards.Gui.Wpf
Lut6.WordCards.Model
Lut6.WordCards.Gui Console
Moq
System.Core
Autofac
System.Drawing
mscorlib
System.Windows.Forms
System
System.Xaml
PresentationFramework
Trial/OpenSrc/Academic Edition

Figure 4. Word Cards .NET 4.0 domain model and efferent coupling.

Summary

While only initial GUI implementations were done during the code camp, the base framework proved to be extensible and flexible as planned. More importantly, the solution demonstrated potential to assist the learning process of new words, thus providing a solution to the presented problem.

To take the project and learning process further, the GUI implementations should be finished. Furthermore, more dictionaries should be created to assist in learning different languages through the use of textual descriptions and imagery.