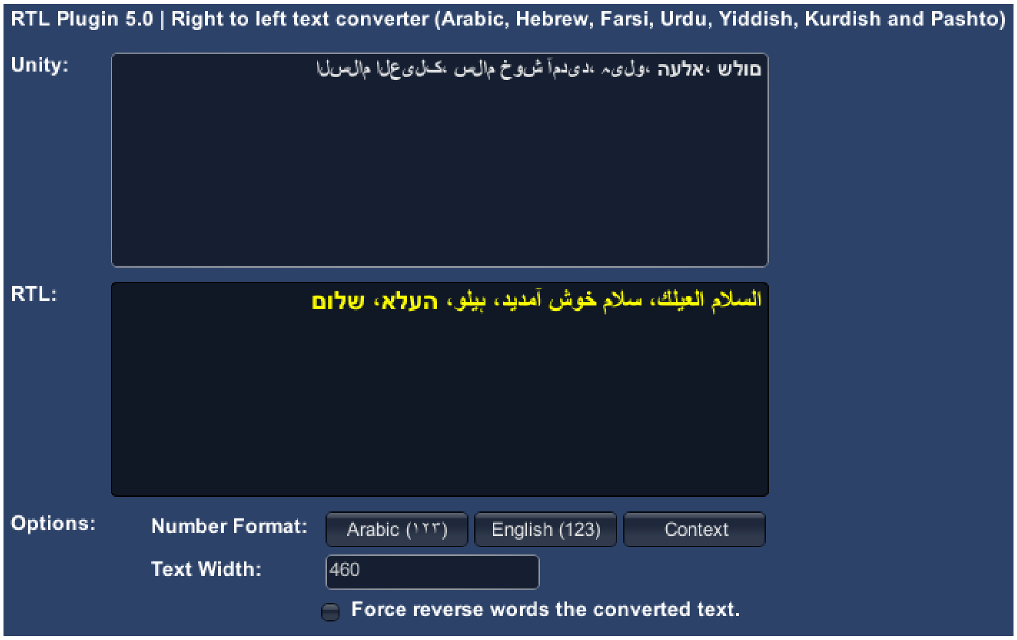
Raymon RTL 5.0.x

RTL plugin is a very powerful and fast text converter for generating `Arabic, Hebrew, Farsi, Urdu, Yiddish, Kurdish and Pashto`[1] right to left text. The result from this plugin can be used in Unity3D, Adobe Photoshop, Flash, 3Ds Max, Maya and other designing software or game engines. However the main focus of the plugin is to produce well-format right to left texts for Unity3D game engine. (If you would use this plugin in web services, web application back-end or mobile applications please contact us raymongame@gmail.com)



RTL Plugin 5.0 | Right to left text converter (Arabic, Hebrew, Farsi, Urdu, Yiddish, Kurdish and Pashto)

Unity: םולש ،אלעה ،ولیہ ،دیدمآ شوخ مالس ،کلیعلا مالسلا

RTL: السلام العيلك، سلام خوش آمديد، ہيلو، העלא، שלום

Options: Number Format: Arabic (١٢٣) English (123) Context

Text Width: 460

○ Force reverse words the converted text.

Features

RTL is awesome because of:

1- Supporting most right to left languages each separately or mixed with English or any left to right languages.

2- Supporting different number formats (English, Arabic and Farsi) like ۴۵۶ and ٤٥٦.

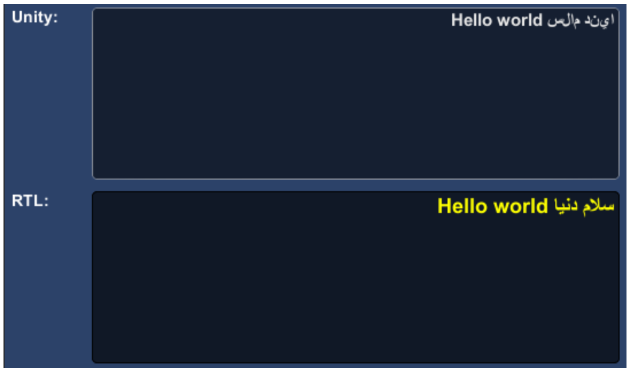
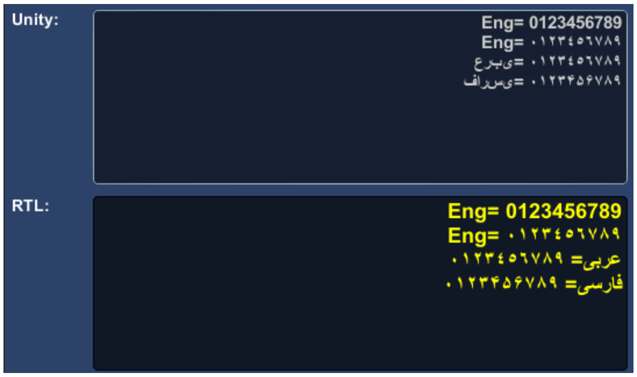
3- Supporting simple math equation conversion. ۴+۵=۹.

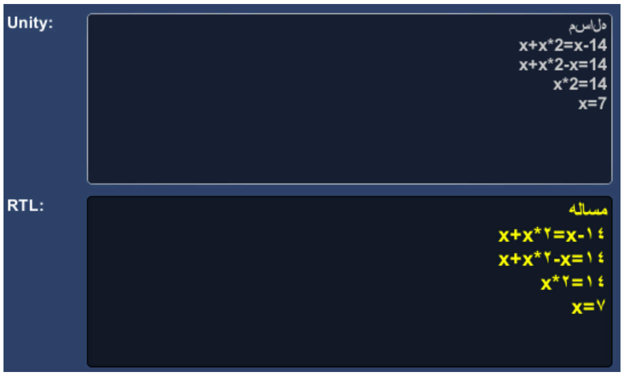
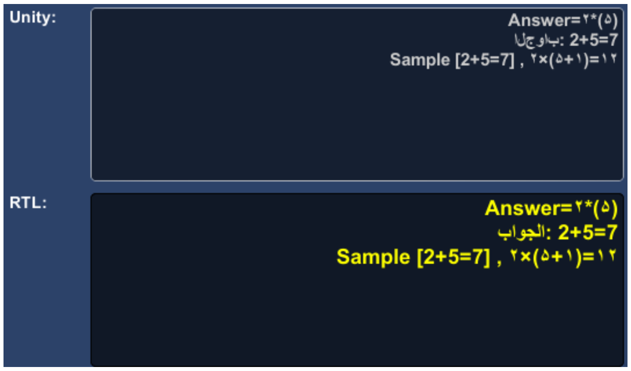
4- Supporting combinations of numbers, punctuations and texts[2] in one function call.

[1] RTL doesn't support yet these characters in Pashto language: څ، ځ، ښ، ږ، ډ، ټ، ګ، ڼ، ړ.

[2] Not all complicated combinations of punctuation, parentheses, RTL and LTR altogether are supported.

Mixed languages support	Different number format conversion


Unity:
Hello world ايند مالس
RTL:
Hello world سلام دنيا
	
	Unity:
	Eng= 0123456789
	Eng= ٠١٢٣٤٥٦٧٨٩
	ىبرع= ٠١٢٣٤٥٦٧٨٩
	ىسراف= ۰۱۲۳۴۵۶۷۸۹
	RTL:
	Eng= 0123456789
	Eng= ٠١٢٣٤٥٦٧٨٩
	عربى= ٠١٢٣٤٥٦٧٨٩
	فارسى= ۰۱۲۳۴۵۶۷۸۹

Mathematic	Mix of punctuation, text and numbers
 Unity: هلاسم x+x*2=x-14 x+x*2-x=14 x*2=14 x=7 RTL: مساله x+x*٢=x-١٤ x+x*٢-x=١٤ x*٢=١٤ x=٧	 Unity: Answer=٢*(٥) باوجلا: 2+5=7 Sample [2+5=7] , ٢×(٥+١)=١٢ RTL: Answer=٢*(٥) الجواب: 2+5=7 Sample [2+5=7] , ٢×(٥+١)=١٢

How to use

RTL Converter is a class library (.dll) you can add to your game projects. This class library contains two main static methods including `Convert()` and `ConvertWordWrap()` returning the converted RTL text. Each method has several parameters to let developers adjust the conversion. You can setup number formats and word-wrap parameters when you call RTL methods. In addition, there are a few helper methods available such as `IsRTL()` and `IsRTLNumber() to help you get the most from right to left game features.`

Main Methods:

- `Convert(…):`

  This method converts and returns the well-format right to left text.

- `ConvertWordWrap(…):`

  This method converts the input text to proper word wrapped right to left text to be used in Unity3D.

Note: There are some specific structures used by RTL like UnityEngine.GUIStyle which is only available in Unity3D game engine.

Parameters:

Here is a list of parameters used in these methods:

- `originalText:`

    The original text you would like to convert.

- `numberFormat:`

    This parameter helps to adjust the output number format:

    `Context (default)`: Number format will be the same as it is in the original text.

    `Arabic:` Force the numbers appear in Arabic format (٠١٢٣٤٥٦٧٨٩).

    `Farsi:` Force the numbers appear in Farsi format (۰۱۲۳۴۵۶۷۸۹).

    `English:` Force the numbers appear in English (`0123456789`) format.

- `isLTRContext:`

    You can set this parameter to true if the original context is in English (or any other left to right language) and it contains only a few right to left phrases. In other words if you need to fix the format for a few words in an English context you need to set this parameter to true and pass the whole text as input to RTL methods.

    Note: If the input context is in a right to left and it has only a few English words or there's no left to right words at all in the text, this parameter should be set to false.

    (Default value is `false`)

- `elementWidth:`

    Width of GUI element which will be used to show the converter text in game. This is useful because Unity game engine doesn't support word wrap for right to left languages. This parameter lets RTL override Unity's calculations and correct word wrapping.

- `guiStyle:`

    This parameter defines the GUI style which will be used to render the converted text. This is required when you would have converted text word wrapped.

    The default value for this parameter is set to `Unity's default GUIStyle.`

- `font:`

    This parameter indicates the font used to render converted right to left text. The default font is Arial.

- `fontSize:`

  The size of the font, based on the sizes set in any word processor. The default font size is 12.

- `fontStyle:`

  This parameter indicates the font style used to render converted right to left text.

Usage examples

By calling the following simple function, a well-formatted right to left Arabic text will be returned. It’s now ready to be rendered in Unity game engine or in any other applications.

```
string originalText = "السلام و علیک";
string convertedText = RTL.Convert(originalText);
```

The following example converts the text with word wrapping effect. The result will be ready to use in Unity 5.0 UI elements as well as in OnGUI function.

```
string converted = RTL.ConvertWordWrap(originalText, 150, myFont, 12, FontStyle.Bold, 
RTL.NumberFormat.English, false);
```

The following example uses Unity's GUILayout to show converted right to left text.

```
GUILayout.TextField(RTL.Convert(originalText),GUILayout.Height(150),GUILayout.Width(250));
```

It’s always a good idea to use the RTL plugin or any computational function outside of OnGUI(). Generally, Update() method is always a better choice. You can store the result of the function in a variable, then use that variable in OnGUI() function if needed.

More examples:

1-
   ```
   //Get RTL text with more options and adjustments (no word-wrapping – Arabic text)
   convertedText = RTL.Convert(InputText, RTL.NumberFormat.Context, false);
   ```

2-
   ```
   //Get RTL text (with word-wrapping – default GUI style)
   convertedText = RTL.ConvertWordWrap(InputText, 200, null);
   ```

3-
   ```
   //Get RTL text (word-wrapping enabled based on element width 300px and rtlStyle)
   convertedText = RTL.ConvertWordWrap(InputText, 300, skin.customStyles[0],
   RTL.NumberFormat.Arabic, false);
   ```

If you have any questions please feel free to contact us: [raymongame@gmail.com]

Good Luck,

Raymon Game