

# Localization and Multiple Languages

One of the unique features available within Orx is the ability to switch languages in game using locale features. While many game engines allow a user to program using locale to some degree, Orx lets you switch languages back and forth in real-time.

An example of this could be a graphical adventure game that features characters, onscreen text, digital speech, and music. Locale switching would allow you to completely switch out all graphics, text, speech and music with a single function call without altering the game-play.

All data is defined by you, and so your regions or nations could be aliens/humans, orcs/elves. You could even think of it as a kind of themeing.

Its a pretty amazing feature that comes out of the box.

This tutorial will be done in two steps:

1. Create a simple game with characters with their own look, speech and music.
2. Add in a second locale, and be able to switch between them.

## Starting a new project

Begin with a blank project created using the [init tool](#).

Name your project something like `the-conversation` (or whatever you wish).

Our game will be about two polite gentlemen having a conversation. They continually greet each other all day.



Eventually, we want to be able to switch their nationality, as the player may prefer to play the game in their own country and language.

## French Assets

Let's begin with the French assets, building the French version first.

Frenchman sprite for the data/texture folder:



Speech samples for the data/sound folder:

[bonjour1.ogg](#)  
[bonjour2.ogg](#)

Music for the same folder:

[je-te-veux.ogg](#)

Delete the default config Object and FX sections. We wont need those.

Set up a smaller resolution for this demo by changing the Camera Frustum:

```
[MainCamera]
FrustumWidth    = 800
FrustumHeight   = 600
```

Also turn off the smoothing effect to suit our pixel graphic style:

```
[Display]
Smoothing = false
```

Create the default Person object in the config with:

```
[Person1]
Graphic         = @
Texture         = typical-frenchman.png
Pivot           = center
Scale          = 5.0
Position        = (-100, 0, 0)
KeepInCache     = true
```

The above config uses section inheritance. You can [read more about it here](#).

Create a second person from the first, but one that stands on the other side and faces the other way:

```
[Person2@Person1]
Position        = (100, 0, 0)
Scale           = (-5, 5, 1)
```

Next, create a Scene object that will be the root to hold all the other objects. Then place both Person objects in it's Childlist.

```
[Scene]
ChildList = Person1 # Person2
```

Create the scene in the game in code. This will create all the child objects onscreen. You can just change the existing Object creation with Scene, from:

```
orxObject_CreateFromConfig("Object");
```

to:

```
orxObject_CreateFromConfig("Scene");
```

Create all the text objects to go onscreen. You will need text objects for:

1. The speech
2. The player instructions
3. The current language indicator

Create these as:

```
[ToggleNationalityObject]
Graphic = ToggleNationalityGraphic
Scale   = 2
Position = (0, 120)

[ToggleNationalityGraphic]
Text = ToggleNationalityText
Pivot = center

[ToggleNationalityText]
String = "PRESS SPACE TO CHANGE NATIONALITY"

[CurrentNationalityObject]
Graphic = CurrentNationalityGraphic
Scale   = 2
Position = (0, 160)
Color   = (0, 128, 255)

[CurrentNationalityGraphic]
Text = CurrentNationalityText
Pivot = center

[CurrentNationalityText]
String = FRENCH

[SpeechObject]
Graphic = SpeechGraphic
Scale   = 2
Position = (-70, -100)
Color   = (0, 128, 255)

[SpeechGraphic]
Text = SpeechText
Pivot = center

[SpeechText]
String = Bonjour
```

And now place them into the scene by adding them to the Childlist:

```
[Scene]
ChildList = Person1 # Person2 # ToggleNationalityObject #
CurrentNationalityObject # SpeechObject
```

Next is the music. Define it with:

```
[BackgroundThemeMusic]
Music      = je-te-veux.ogg
Volume     = 0.5
Loop       = true
KeepInCache = true
```

And add it to the scene with:

```
[Scene]
ChildList = Person1 # Person2 # ToggleNationalityObject #
CurrentNationalityObject # SpeechObject
SoundList = BackgroundThemeMusic
```

Next is the actual conversation. Define the speech with:

```
[Hello1]
Sound      = bonjour1.ogg
KeepInCache = true

[Hello2]
Sound      = bonjour2.ogg
KeepInCache = true
```

Then, using a timeline track switch on and off the alpha of the speech text object to make it appear/disappear, move its position and play the sounds:

```
[ConversationTimeline]
= Object.SetPosition ^ (-70,-100) # Object.SetAlpha ^ 1.0 # Object.AddSound
^ Hello1
1 = Object.SetPosition ^ (70,-100) # Object.AddSound ^ Hello2
2 = Object.SetAlpha ^ 0.0
4 =
Loop      = true
KeepInCache = true
```

Add the Timeline to the SpeechObject so that the speech will be animated:

```
[SpeechObject]
Graphic = SpeechGraphic
Scale = 2
Position = (-70, -100)
Color = (0, 128, 255)
TrackList = ConversationTimeLine
```

Compile and run. Our two French gentlemen will start having a lovely conversation.

That is the basic game.

Now to make the game switchable to Australia mode. This is the easy part.

## Australian Assets

We'll get the Australian version of the assets in now. These will replace the French ones.

Australian sprite:



Speech samples:

[gidday1.ogg](#)  
[gidday2.ogg](#)

Music:

[binda\\_polka.ogg](#)

Begin by defining the two languages for the game: French and Australian.

```
[Locale]
LanguageList = French # Australian
```

Next, define those languages as sections containing your keys for the various assets or text:

```
[French]
Hello = Bonjour
Image = typical-frenchman.png
HelloSound1 = bonjour1.ogg
HelloSound2 = bonjour2.ogg
BackgroundMusic = je-te-veux.ogg
LanguageText = FRENCH
```

```
[Australian]
Hello          = Giddy
Image          = typical-aussie.png
HelloSound1   = giddy1.ogg
HelloSound2   = giddy2.ogg
BackgroundMusic = binda_polka.ogg
LanguageText  = AUSTRALIAN
```

Finally, go to each config and replace the absolute asset paths and text with the named keys from language sections. These keys are called using the special \$ symbol:

```
[Hello1]
Sound = $HelloSound1
KeepInCache = true

[Hello2]
Sound      = $HelloSound2
KeepInCache = true

[BackgroundThemeMusic]
Music      = $BackgroundMusic
Volume     = 0.5
Loop       = true
KeepInCache = true

[Person1]
Graphic    = @
Texture    = $Image
Pivot      = center
Scale      = 5.0
Position   = (-100, 0, 0)
KeepInCache = true

[CurrentNationalityText]
String = $LanguageText

[SpeechText]
String = $Hello
```

Excellent, the localization is now added to all our parts of the config. But there is no way to switch locales. The final step is to define the Space Bar as the key to switch between languages. First the key in the config:

```
[MainInput]
KEY_ESCAPE    = Quit
KEY_SPACE     = SwitchLanguage
```

Define a variable at the top of the code to keep track of the current language:

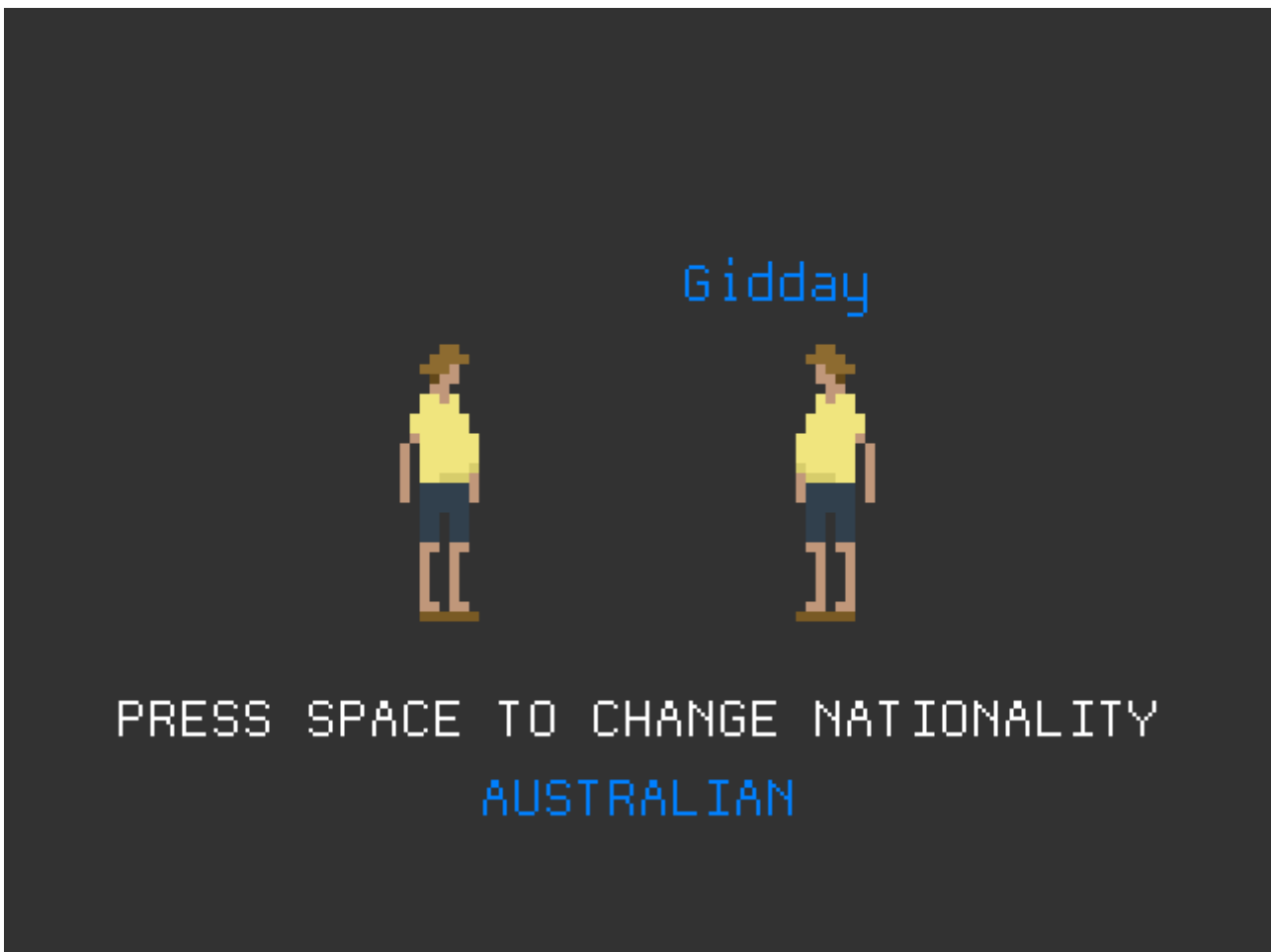
```
int currentLanguageIndex = ;
```

Then in code, in the Run ( ) function, switch the locale:

```
if (orxInput_HasBeenActivated("SwitchLanguage"))
{
    currentLanguageIndex++;
    if (currentLanguageIndex > 1){
        currentLanguageIndex = ;
    }

    orxLocale_SelectLanguage(orxLocale_GetLanguage(currentLanguageIndex));
}
```

Compile and run. Now you can switch between the two languages with the Space Bar. Everything in the game switches over in an instant. Pretty impressive stuff.



Acknowledgements, permissions and thanks:

- Thank you to Norimichi Nagasaka for allowing me to use his accordion performance of Je Te



Veux. His Youtube channel is at: <https://www.youtube.com/user/VAccordion>

- Thanks to Trevor Brennan: <http://trevorbrennan.com> for the graphics.
- Thanks to FullyBugged, Dom M and Matt R for the voices.

From:

<https://orx-project.org/wiki/> - **Orx Learning**

Permanent link:

<https://orx-project.org/wiki/en/tutorials/localization>

Last update: **2019/01/02 15:04 (17 months ago)**

