

Tutorial para suporte a multi- idiomas em jogos através do Construct 2

Alunos: Franklin Mateus Mendonça Barbosa

João Victor da Silva Justino

Lucas Emmanuell da Silveira Gonçalves

Turma: PJD – 3M

Tutorial Jogo com Multi-Idiomas

1. Introdução

O dado trabalho tem como objetivo explicar como se faz o mecanismo de múltiplos idiomas em um jogo. Para isso, documentaremos os principais passos que devem ser tomados para criar essa mecânica no motor de jogos “*Construct 2*”.

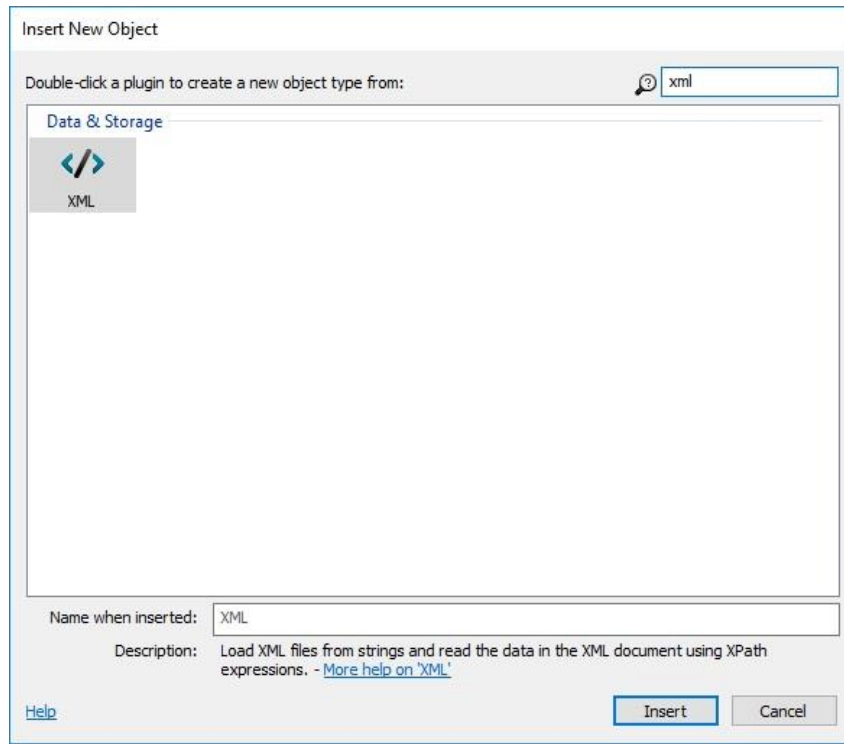


2. Criando arquivo de texto

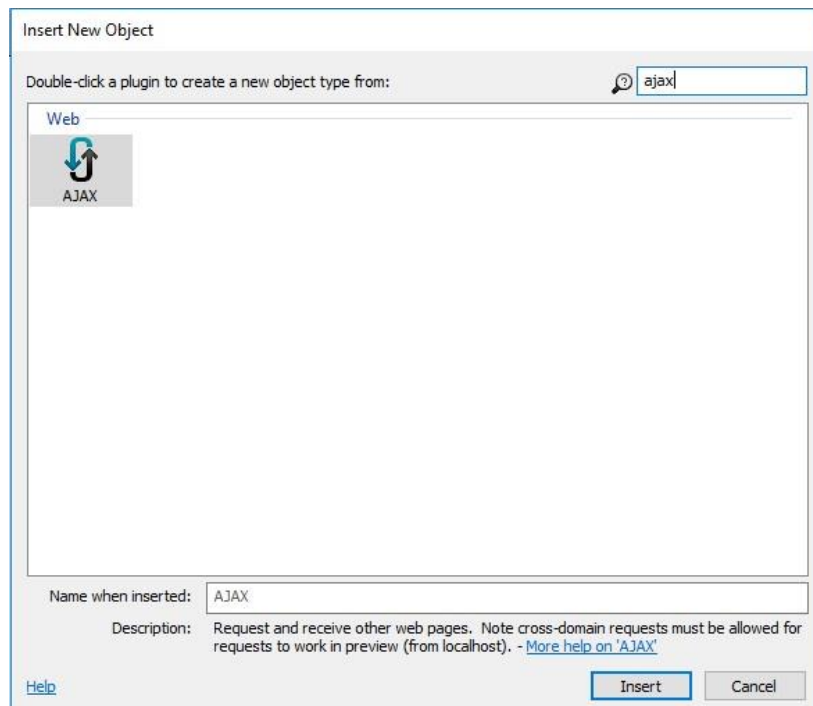
O primeiro passo para criar um mecanismo de múltiplos idiomas é escolher os idiomas que serão utilizados. Para esse trabalho, usaremos 3: Português do Brasil, Inglês e Espanhol, que são línguas que somos mais familiarizados.



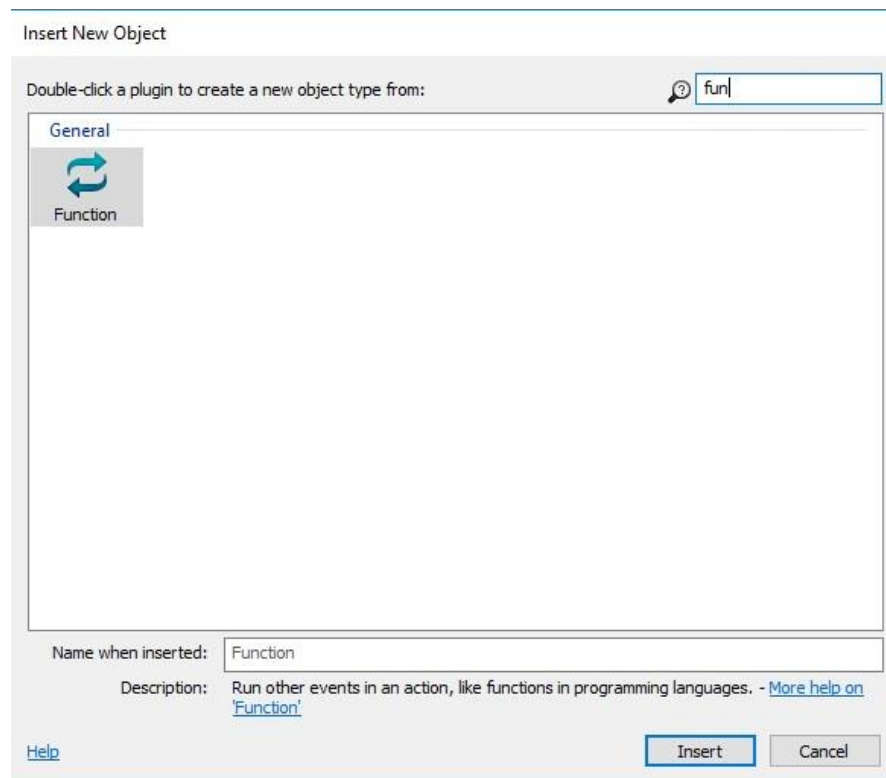
Devemos, agora, adicionar alguns plug-ins ao projeto. O primeiro, e mais importante deles, é o **XML**, que serve para ler ou analisar algumas informações em arquivos XML. Esses arquivos XML são arquivos de texto estruturados que servem para compartilhar determinadas informações. Nós os faremos mais a frente.



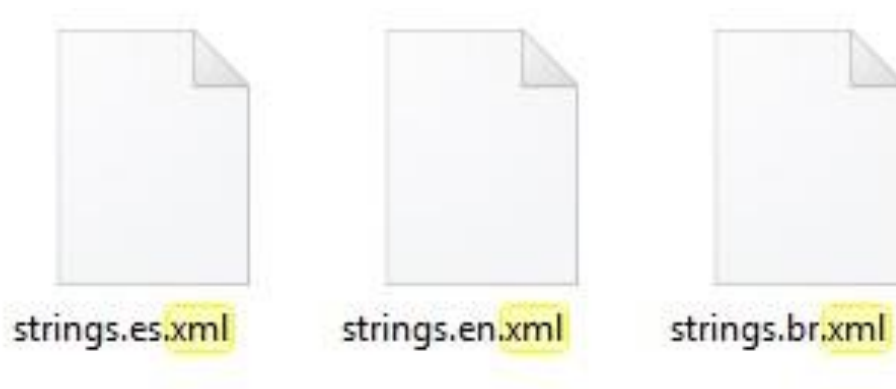
Também devemos adicionar o plug-in **AJAX**. Ele tem a função de requisitar e ler as informações que estão contidas nos arquivos XML.




Por último, adicionaremos o objeto **Function**. Esse objeto servirá para organizar funções que usaremos no suporte a múltiplos idiomas. Com elas, evitamos a duplicação de ações ou eventos desnecessários.



Logo após a inclusão desses objetos, devemos criar um dicionário de texto para cada uma das linguagens. Esse dicionário de textos deve ser feito no formato XML, que é um protocolo da internet recomendado para gerar linguagens. Como temos três linguagens, devemos criar três arquivos XML, um para cada linguagem. O formato de arquivo deve ser **strings.xx.xml**, onde o **xx** será o idioma que queremos usar, sendo **br** para português, **en** para inglês e **es** para espanhol. Ele deve ser feito no bloco de notas e depois deve ser alterado para o leitor de XML do seu computador



Agora basta criar as bibliotecas dos dicionários como nas imagens abaixo. Esses dicionários são formados por ID's e por valores, sendo os ID's aqueles que estão entre aspas simples e os valores os que estão entre "> <".


 strings.br.xml - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
<?xml version="1.0" encoding="UTF-8"?>

<strings>
  <string id='play'>Iniciar</string>
  <string id='configuration'>Configuração</string>
  <string id='credits'>Créditos</string>
  <string id='atention'>Cuidado com o que se esconde no escuro</string>
</strings>
```

XML português


 strings.en.xml - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
<?xml version="1.0" encoding="UTF-8"?>

<strings>
  <string id='play'>Start</string>
  <string id='configuration'>Configuration</string>
  <string id='credits'>Credits</string>
  <string id='atention'>Beware of what lurks in the dark</string>
</strings>
```

XML inglês

 strings.es.xml - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

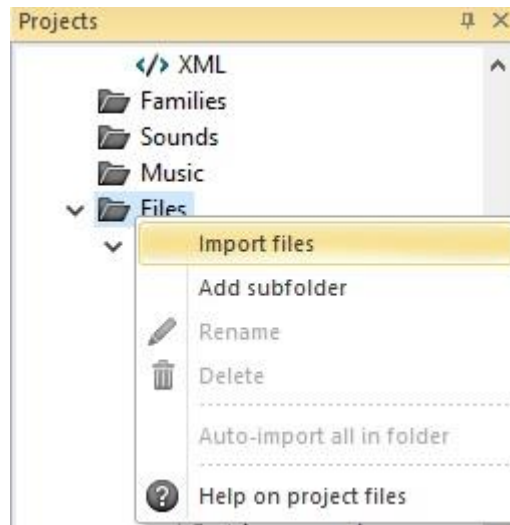
```
<?xml version="1.0" encoding="UTF-8"?>

<strings>
  <string id='play'>Empezar</string>
  <string id='configuration'>Configuraciones</string>
  <string id='credits'>Creditos</string>
  <string id='atention'>Cuidado con lo que se esconde en la oscuridad</string>
</strings>
```

XML espanhol

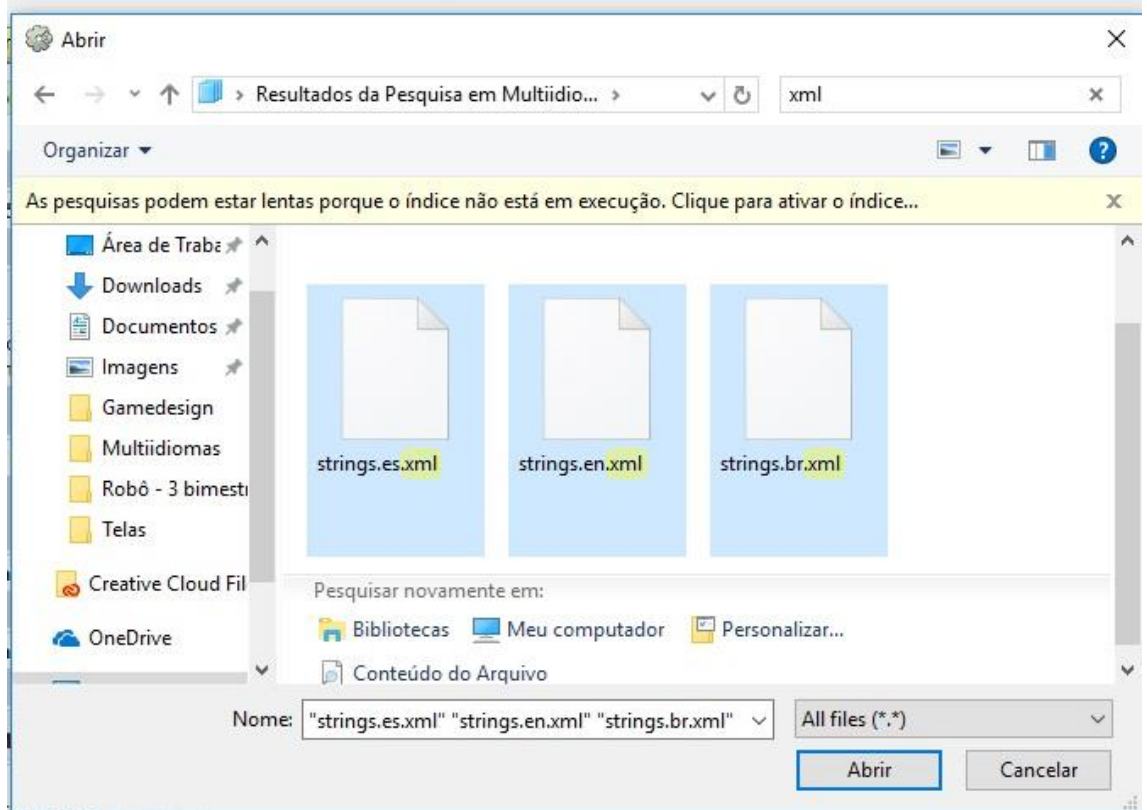
3. Criando o código

Após criar e alterar o arquivo, devemos abrir o arquivo e importá-lo para o Construct 2. Ele deve ser importado a partir da pasta **Files**.

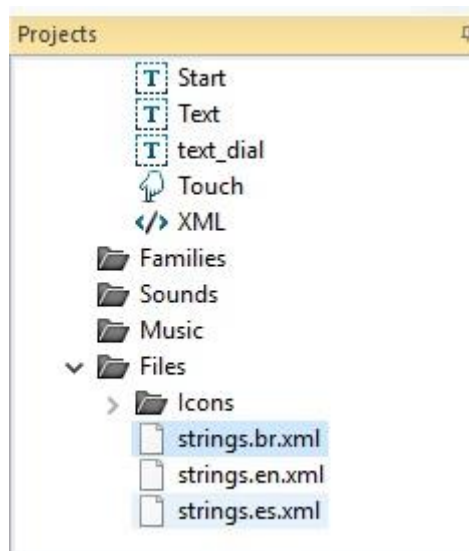


Importando arquivos

Construct 2 - multiidioma.cpx - EventMenu



Selecionando arquivos XML



Confirmando que os arquivos foram importados

Importados os arquivos, iremos agora criar uma variável global, no *event sheet*, que armazenará a língua inicial. A essa variável daremos o nome de **Idioma**. Logo após, criaremos uma função que vai carregar os idiomas dos arquivos XML (**ProcurarLinguas**) através do AJAX. Logo após faremos um rastreo (Comando do AJAX *on completed*, com a tag **Rastreo**) do idioma que está sendo usado e, dependendo de qual for, vamos modificar os textos, através de uma outra função **MudarTexto**, que vai receber uma outra função a qual nomearemos **GetText**. Essa função vai modificar os valores dos ID's nas diversas linguagens.

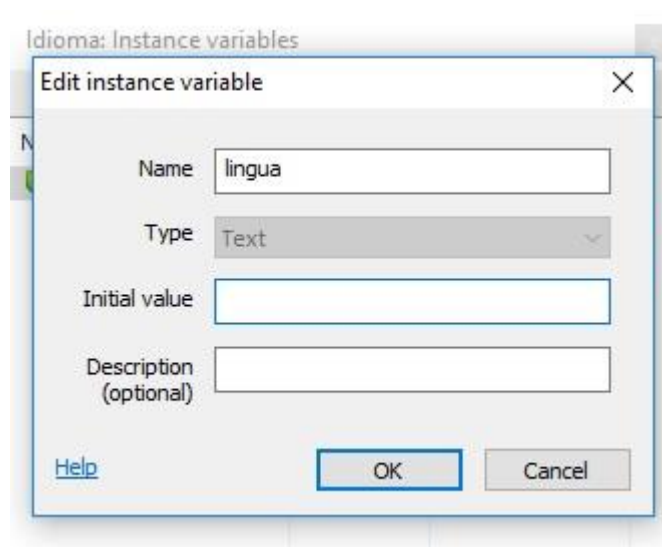
bucar idioma			
Fun...	On "ProcurarLinguas"	AJAX	Request "strings."&Idioma&".xml" (tag "Rastreo")
		Add action	
AJAX	On "Rastreo" completed	XML	Load XML document from string AJAX.LastData
		Function	Call "MudarTextos" ()
		Add action	
Fun...	On "MudarTextos"	Start	Set text to Function.Call("getText","play")
		Config	Set text to Function.Call("getText", "configuration")
		Credits	Set text to Function.Call("getText", "credits")
		Aviso	Set text to Function.Call("getText", "attention")
		Add action	

Por último, faremos a função **GetText** que é responsável por levar um ID de string ao **MudarTexto** e então retornar um valor que esteja em algum dos dicionários XML. O **GetText** é essencial e se estrutura como mostrado na imagem abaixo.

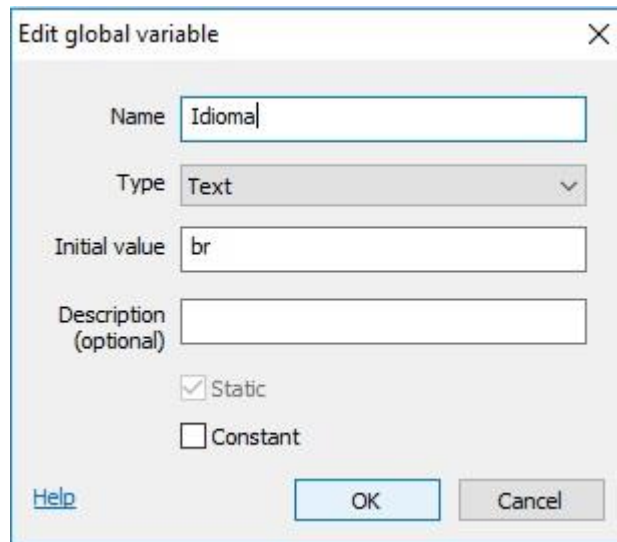


Além de criar as funções de mudar o idioma, devemos fazer com que o jogador possa escolher esses idiomas. Para isso, criaremos um grupo **Língua**.

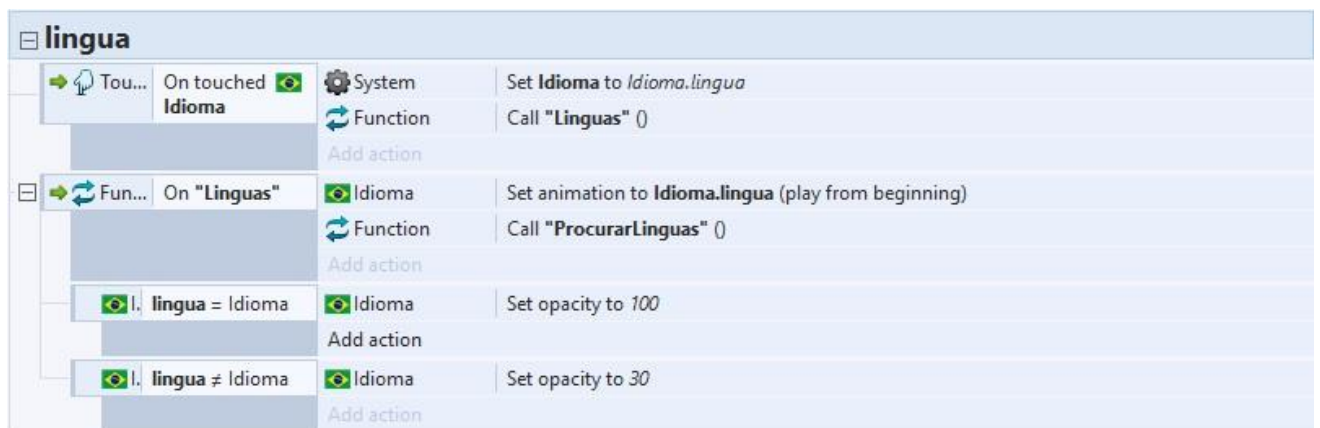
Antes de prosseguirmos no grupo, precisaremos criar uma variável de instância do objeto chamada **língua** que queremos que fique responsável por trocar o idioma – no caso desse jogo, em específico, são as bandeiras. Essa variável armazenará a informação do idioma ao qual o jogador prefere.



Também faremos uma variável global chamada **Idioma** que receberá os valores armazenados na variável *lingua* – optamos pelo valor inicial ser **br** por preferência própria, mas isso fica a critério de cada um.

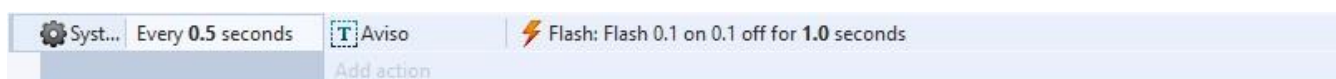


Agora, voltando ao grupo **lingua**, faremos um comando que, ao jogador tocar em certo ícone, será lido o valor de uma variável de instancia salvo no ícone, e, dependendo dessa variável de instancia, o comando vai carregar o arquivo de texto direto do dicionário de idiomas. Todos esses comandos vão depender do valor da variável global *Idioma*, que tem seu valor também modificado pela variável de instancia *lingua* de cada um dos ícones, sendo ela **br** para Brasil, **en** para EUA e **es** para Espanha. Um comando bônus é modificar a opacidade para deixar um pouco mais bonito visualmente, para isso, basta analisar qual o valor da variável *lingua* e mudar a opacidade da que está ativa para mais e das não ativas para menos.



4. Adicionais

Você pode deixar a mensagem de atenção piscando através do evento **Every x seconds** somado ao **Behavior: Flash**.



Além disso, você pode carregar sempre o valor inicial da linguagem, para que o jogo não comece com tudo apagado e só mostre os textos quando clicar em um dos ícones de linguagem.

