WebGLJsDialogTool

Asset that allows implementing dialog windows for browser functions.

Steps to use:

1. Import “ShigeDev.WebGLJsDialogTool” namespace in each class/script you want to use it.

```
using  ShigeDev. WebGLJsDialogTool;
```

2. Add the code "WebGLJsDialogTool." and the method you want where you want the functionality to be.
    - WebGLJsDialogTool.DisplayJsAlert(param).
    - WebGLJsDialogTool.DisplayJsPrompt(param).
    - WebGLJsDialogTool.DisplayJsConfirm(param).
    - WebGLJsDialogTool.DisplayJsDialog(param).

Scripting API

Public static Methods

DisplayJsAlert(string alertMessage)	Display a dialog with an optional message, and to wait until the user dismisses the dialog into the browser. The alertMessage parameter is the text to be displayed in the dialog.
DisplayJsPrompt(string promptMessage)	Display a dialog with an optional message prompting the user to input some text, and to wait until the user either submits the text or cancels the dialog into the browser. The promptMessage parameter is the text to be displayed in the dialog. This method returns a variable nullable of type string. It is the text inputted by the user in the dialog.
DisplayJsConfirm(string confirmMessage)	Display a dialog with an optional message, and to wait until the user either confirms or cancels the dialog into the browser. The confirmMessage parameter is the text to be displayed in the dialog. This method returns a variable of type bool. It is the option chosen by the user in the dialog.

DisplayJsDialog(string dialogId)	Displays the dialog as a modal. The dialogId parameter is the id of the dialog to be displayed.

DisplayJsAlert

To call the static method DisplayAlert you must write the name of the static class "WebGLJsDialogTool", then the name of the method “DisplayJsModal” and add a string variable as parameter.

```
WebGLJsDialogTool.DisplayJsAlert( "This  is  alert");
```

Code example:

```
using  System. Collections;
using  System. Collections. Generic;
using  UnityEngine;
using  ShigeDev. WebGLJsDialogTool;

public  class  Example  :  MonoBehaviour
{
  public  void  ShowAlert()
   {
          WebGLJsDialogTool.DisplayJsAlert( "This  is  alert");
   }

}
```

DisplayJsPrompt

To call the static method DisplayJsAlert you must create a string type variable which will store the value that the method will return, then write the name of the static class "WebGLJsDialogTool", the name of the method “DisplayJsPrompt” and adding a string type variable as a parameter. Note: The string variable in which the result of the method is to be stored must be nullable.

```
string? userInput = WebGLJsDialogTool.DisplayJsPrompt("Write something");
```

Code example:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using ShigeDev.WebGLJsDialogTool;

public class Example : MonoBehaviour
{
  public void ShowPrompt()
   {
      string? userInput = WebGLJsDialogTool.DisplayJsPrompt("Write something");

   }

}
```

DisplayJsConfirm

To call the static method DisplayJsConfirm you must create a bool type variable which will store the value that the method will return, then write the name of the static class “WebGLJsDialogTool”, the name of the method “DisplayJsConfirm” and add a string type variable as a parameter.

```
bool result = WebGLJsDialogTool.DisplayJsConfirm("Select OK or Cancel");
```

Code example:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using ShigeDev.WebGLJsDialogTool;

public class Example : MonoBehaviour
{
  public void ShowConfirm()
   {
      bool result = WebGLJsDialogTool.DisplayJsConfirm("Select OK or Cancel");
   }
}
```

DisplayJsDialog

To use the DisplayJsDialog static method in a C# script, you must enter the name of the static class WebGLJsDialogTool in a code block, then the method name "DisplayJsDialogl," and add a string variable as a parameter. Finally, in the index.html file of your WebGL game compilation, you must add the dialog with an ID.

```
WebGLJsDialogTool.DisplayJsDialog( "dialogId");
```

C# code example:

```
using System. Collections;
using System. Collections. Generic;
using UnityEngine;
using ShigeDev. WebGLJsDialogTool;

public class Example : MonoBehaviour
{
  public void ShowDialog()
   {
          WebGLJsDialogTool.DisplayJsDialog( "dialogId");
   }
}
```

html code example(index.html):

```
<dialog id ="dialogId">
        <p>This is dialog.</p>
        <button>Close</button>
</ dialog>
```