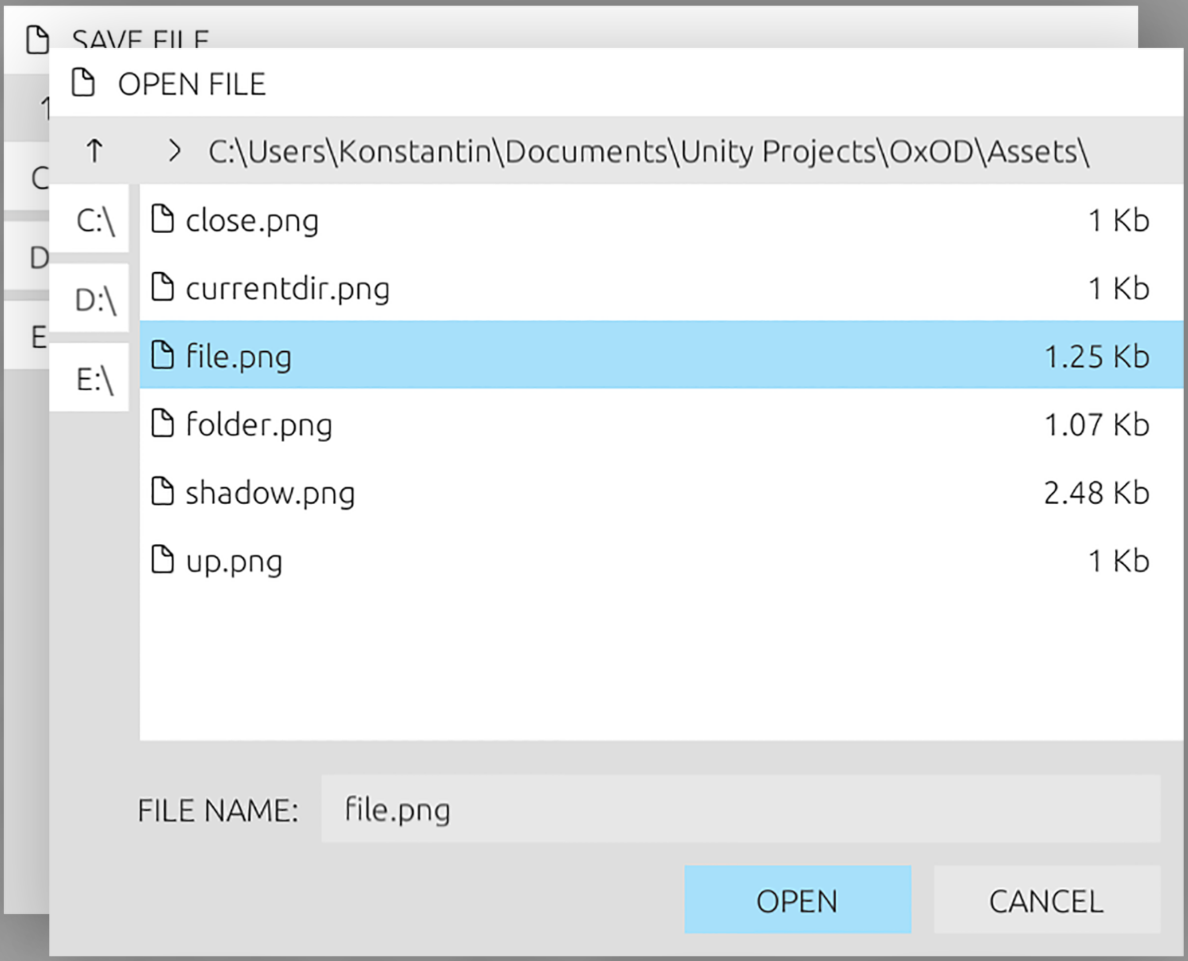


C:\Users\Konstantin\Documents\Unity Projects\OxOD\Assets\

close.png	1 Kb
currentdir.png	1 Kb
file.png	1.25 Kb
folder.png	1.07 Kb
shadow.png	2.48 Kb
up.png	1 Kb

FILE NAME: file.png

Unity UI

OXOD

Open/Save File Dialog System

USER MANUAL

v. 1.5-2017.11.11

TABLE OF CONTENTS

ABOUT

At first, thank you for purchasing OxOD! OxOD — simple, modern and lightweight open/save file dialog system for Unity UI. Displays a dialog box that prompts the user to select a file, then returns it's full path. Easy to integrate in the existing project. Supported platforms: Windows, Mac, Linux. Note that OxOD requires Unity 4.6 or later. If you're using older version of Unity, you must update it.

If you have any questions or suggestions, please contact: oxy949@gmail.com

DEMO

There is a demo scene, located in "Assets/OxOD/Demo/" folder, named "DemoScene".

VERSION CHANGES

• Release 1.1:

+ Added maxSize and saveLastPath options;

* Some bug fixes

• Release 1.2:

* Fixed bug with maxSize parameter

• Release 1.3:

* Improved documentation

* Fixed bug with saveLastPath parameter

• Release 1.4:

* Added SelectFolder option for dialogue

* FileSelector: Renamed OnFileSelected() Event → OnDialogueEnded(string result)

• Release 1.5:

* Now works in a world space canvas and is thus suitable for Virtual Reality applications

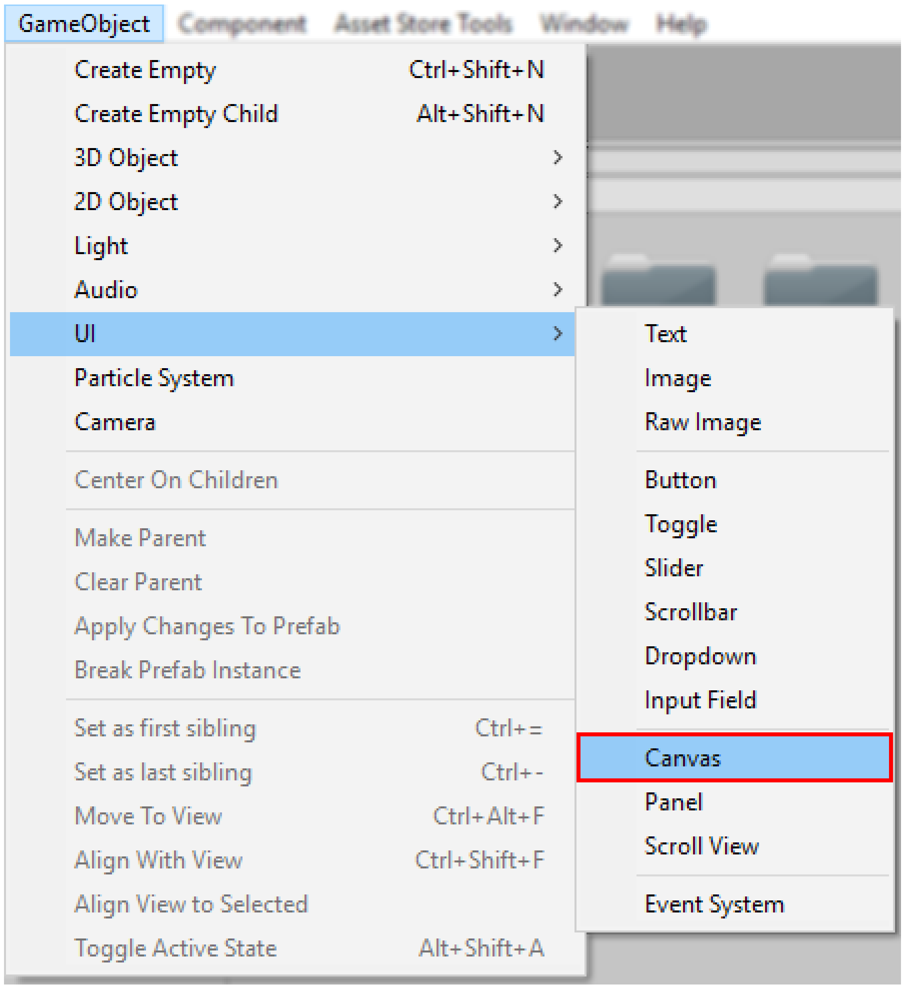
KNOWN BUGS

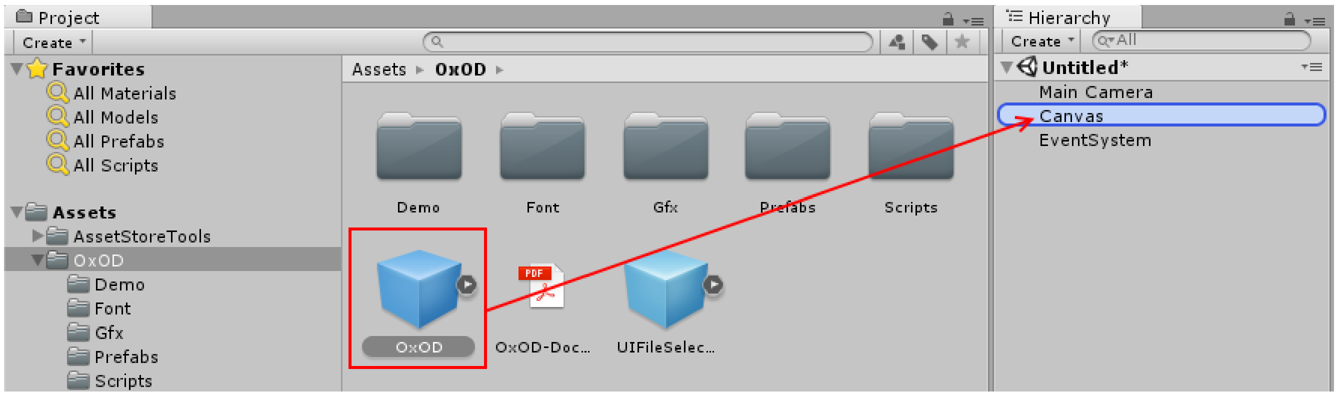
• None

USAGE WITHOUT CODING

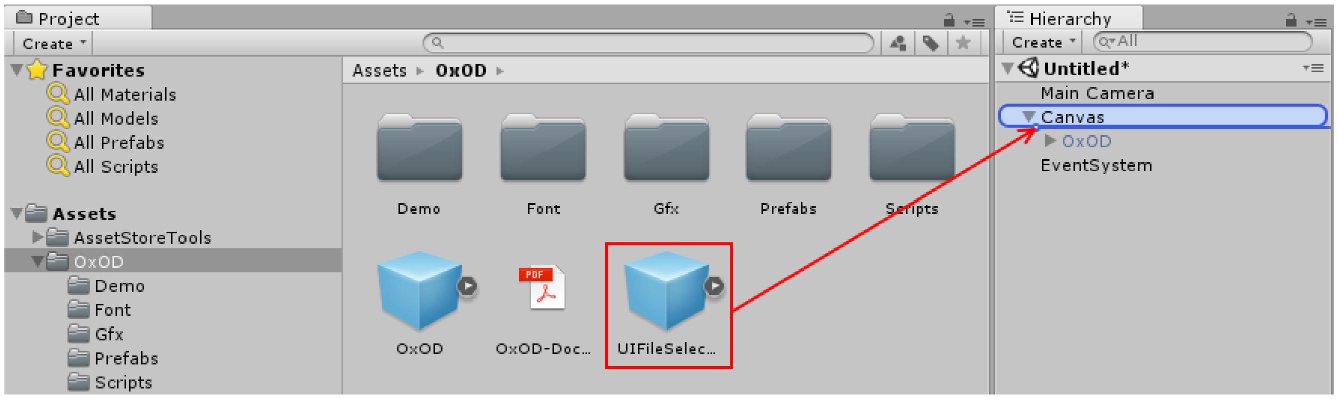
Plugin is easy-to-use.

1. Drag and drop “OxOD” prefab from “Assets/OxOD/” folder into the scene's root canvas (usually named “Canvas”). If you don’t yet have root Canvas object (and EventSystem for it), create it from Unity menu: GameObject → UI → Canvas;

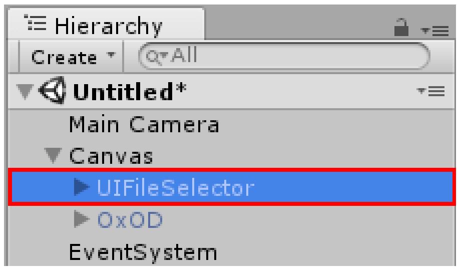




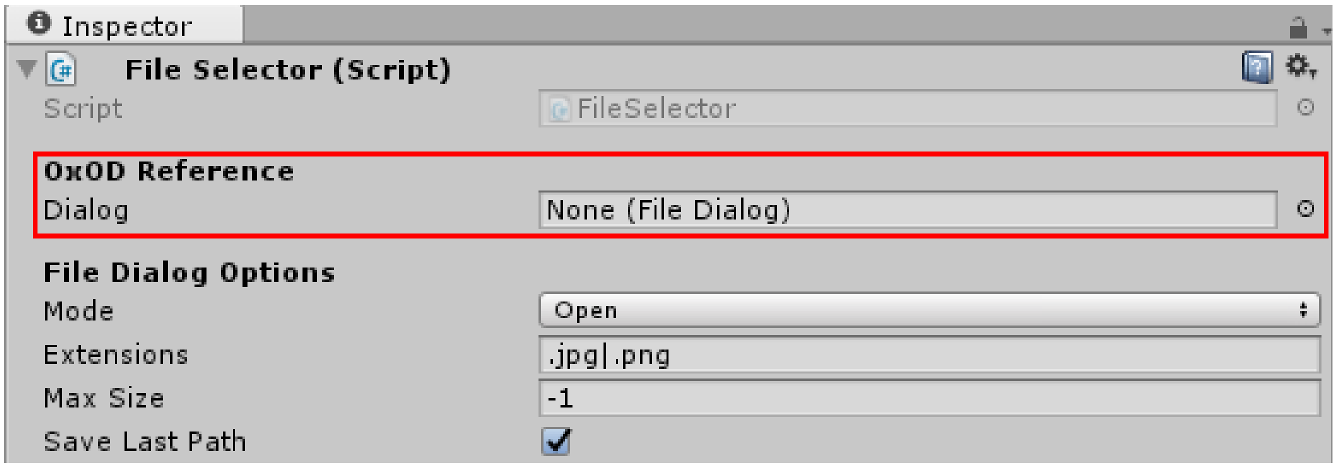
2. Drag “UIFileSelector” prefab from “Assets/OxOD/” folder to your UI form (or create new one (or just drop it to the root Canvas));



3. Select created instance of “UIFileSelector” prefab and in the Unity Inspector window:



3.1 Drag and drop “OxOD” prefab instance (from scene objects, not from Assets) to the “Dialog” field;


Inspector

File Selector (Script)

Script FileSelector

OxOD Reference

Dialog None (File Dialog)

File Dialog Options

Mode Open

Extensions .jpg|.png

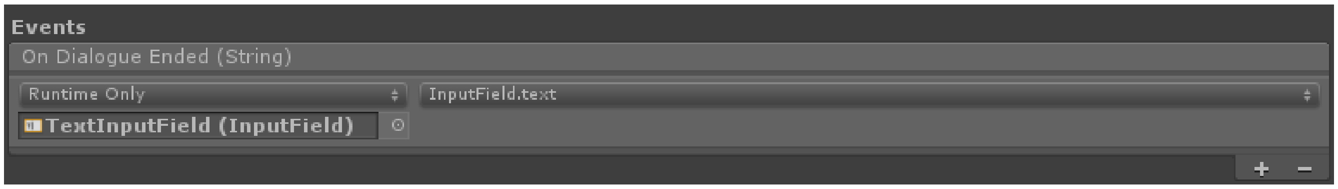
Max Size -1

Save Last Path ✓

3.2. Edit File Dialogue Options:

- Swich mode (Open/Save file/Select folder)
- Edit list of the allowed extensions, splited by "|" symbol. Example: `".png|.jpg|.gif"` (only for file Open/Save file modes)
- Set maximum file size in bytes (only for "Open" mode) (-1 – unlimited)

3.3. Link any action you want to "OnDialogueEnded (String result)" event (more info at: https://docs.unity3d.com/Manual/UnityEvents.html) or simply use "public string result" for your needs (see the demo scene for examples);

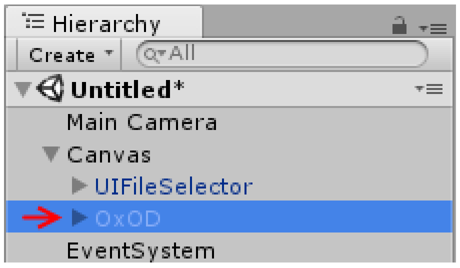

Events

On Dialogue Ended (String)

Runtime Only InputField.text

TextInputField (InputField)

4. Make sure the OxOD prefab is on top of other elements (last child of Canvas object in scene);



5. Done! Now you ready to launch created scene.

USAGE WITH CODING (NORMAL)

1. Repeat 1st and 4th steps from "USAGE WITHOUT CODING";

2. Open any MonoBehaviour (Unity) script and add public reference to the OxOD.FileDialog:

```
using OxOD;
public FileDialog dialog;
```

3. Create new Coroutine (IEnumerator) with "yield return StartCoroutine(dialog.Open()" for async file selection for opening, "yield return StartCoroutine(dialog.Save()" for async file selection for saving and "yield return StartCoroutine(dialog.SelectFolder()" for async folser selection;

4. After dialog box closed, use "dialog.result" to get full path for selected file or folder. Note, dialog.result will be null, if dialogue was cancelled;

5. Invoke created Coroutine (IEnumerator) from script by StartCoroutine(CoroutineName());

6. Link "OxOD" prefab instance with field from created script (see 3.1 from "USAGE WITHOUT CODING").

Example:

```
public IEnumerator SelectFile()
{
    Debug.Log("[FileSelector] Starting file dialog");

    yield return StartCoroutine(dialog.Open());

    if (dialog.result != null)
    {
        Debug.Log("[FileSelector] Dialog ended, result: " + dialog.result);
         // Your interactions with dialog.result here
    }
    else
    {
        Debug.Log("[FileSelector] Dialog canceled");
    }
}

public void OnSelectFileButtonClick()
{
      StartCoroutine(SelectFile());
}
```

ADVANCED PARAMETERS

“OxOD.FileDialog.Open()” and “OxOD.FileDialog.Save()” have equals parameters (except “maxSize”):

```
Open/Save(string path = null, string allowedExtensions = null, string
windowName = "OPEN FILE", Sprite windowIcon = null, long maxSize = -1, bool
saveLastPath = true)
```

Where:

- `path` – default full path to file/folder
- `allowedExtensions` – string of the allowed extensions, splited by “|” symbol. Example: `".png|.jpg|.gif"`
- `windowName` – window title
- `windowIcon` – Sprite icon (null – default file icon)
- `maxSize` – Maximum file size in bytes (only “Open” parameter) (-1 – unlimited)
- `saveLastPath` – Save last directory’s path to PlayerPrefs "OxOD.lastPath". Used as default path, when “path” parameter is not set