IFC Importer README

The IFC Importer is a plugin for converting IFC / BIM -models into Unity prefabs on your local machine. It preserves the hierarchy and properties of the IFC file. The plugin also includes a work in progress IFC exporter.

Link to instructions video: https://youtu.be/GjQTLVUE2cM

FEATURES

- Import IFC building information models (BIM) into Unity as GameObject hierarchies
- Export IFC models from Unity into .ifc files. This feature is work in progress and supports editing properties.
- Works both on runtime and in the editor
- Imported IFC models retain their property, attribute, material, quantity, types and header data
- Search game objects based on their IFC id, presentation layer and element type
- Supports both IFC 2x3 and IFC 4
- Has been tested, and confirmed to work with IFC models exported from Autodesk Revit, Archicad, Tecla Structures, Allplan and Microstation Triforma
- Works with Unity 2019.x or later on Windows, Linux and Mac
- Local process with no cloud subscription needed
- Readme with simple C# examples
- Demo scenes
- Commented C# code
- Easy to use

INSTALLATION

1. Download the plugin from the Unity Asset Store
2. Download version 0.7 of IfcConvert for your operating system from https://ifcopenshell.org/downloads.html
3. Unzip the file you downloaded and copy the executable file "IfcConvert.exe" (Windows) or "IfcConvert" (Linux & Mac) to the IfcImporter folder in your project folder (on Windows by default to C:\UnityProjects\ProjectName\Assets\IfcImporter)

FILES FOR TESTING

- http://openifcmodel.cs.auckland.ac.nz/_models/030811DuplexModel-IFC-2011-05-05.zip
- http://openifcmodel.cs.auckland.ac.nz/_models/20160414office_model_CV2_fordesign.ifc
- http://openifcmodel.cs.auckland.ac.nz/_models/20160125Autodesk_Hospital_Parking%20Garage_2015%20-%20IFC4.ifc

Email support: leo.salomaa@arcventure.fi

IMPORTING IFC MODELS

Importing IFC models into the editor

1. Manually drag your IFC file into the Unity Assets folder in the editor
2. The conversion starts automatically. Please wait for it to finish.
3. Place the newly created prefab into the Unity scene from the Assets folder. NOTE: The conversion process may require a large amount of memory! If your computer runs out of memory during the process, Unity may crash. The IFC-Importer is unable to import IFC files that IfcConvert.exe is unable to process.

Importer settings

You can change a number of importer settings by opening the IFC importer settings window. Select "Edit/IFC Importer Settings..." to open the IFC importer settings window. With the runtime importer you can give the settings as an optional parameter as shown in the runtime importing example below.

Importing IFC models at runtime

IFC models can be imported into Unity at runtime using IfcImporter.RuntimeImport().

```
using System.Collections.Generic;
using UnityEngine;
using IfcToolkit;

public class RuntimeImportDemo : MonoBehaviour
{
    public string assetPath = "Path/from/project/root/";
    public string filename = "your_file.ifc";

    void Start()
    {
        //An optional parameter to toggle various importing option. All are true by default.
        Dictionary<string, bool> options = new Dictionary<string, bool>()
        {
            {"meshCollidersEnabled", true},
            {"materialsEnabled", true},
            {"propertiesEnabled", true},
            {"attributesEnabled", true},
            {"typesEnabled", true},
            {"headerEnabled", true},
            {"unitsEnabled", true},
            {"quantitiesEnabled", true},
            {"parallelProcessingEnabled", true},
            {"keepOriginalPositionEnabled", true},
            {"keepOriginalPositionForPartsEnabled", true}
        };

        //Various parts of the building are rootObject's children.
        GameObject rootObject = IfcImporter.RuntimeImport(assetPath+filename, options);
    }
}
```

EXPORTING IFC MODELS

Exporting IFC models from the editor

1. Select the root of an IFC GameObject in the editor.
2. Choose "Assets/Export selected IFC..." in the editor's top menu bar.
3. The IFC file will be saved in the project folder with the name "From_UnityEditor.ifc".

Exporting IFC models at runtime

IFC models can be saved from Unity as IFC files using IfcExporter.Export() assuming that the file has been imported using the IFC importer.

NOTE: Since Unity does not add IfcConvert.exe into your build automatically, you need to copy it there. You also need to ensure that asset paths and filenames match those in the build folder. In other words, if you are testing the Runtime importing demo scenes:

1. Create a folder named "Assets" in your build folder.
2. Copy IfcConvert.exe into the Assets folder
3. Create a folder named "IfcImporter" in the Assets folder
4. Create a folder named "Demo" in the IfcImporter folder
5. Copy demo_duplex.ifc to the Demo folder.

```
using UnityEngine;
using IfcToolkit;

public class RuntimeExportDemo : MonoBehaviour
{
    // This is a reference to the gameobject of an imported IFC model in the scene
    public GameObject ifcRootGameObject;

    void Start()
    {
        // Export edited IFC model to the Unity project's root directory
        IfcExporter.Export("From_ExporterDemo.ifc", ifcRootGameObject);
    }
}
```

EDITING IFC MODELS

IFC Importer Exporter includes functionality for editing the properties of IFC elements.

Editing IFC model properties

To edit the properties of an IFC element in the Unity editor, simply change the GameObject's IfcProperties component data. To do this on runtime, call the ChangeValue() function of an IfcProperties component as shown in the example below.

```
using UnityEngine;
using IfcToolkit;

public class RuntimeEditElementDemo : MonoBehaviour
{
    public GameObject ifcRootGameObject;
    public GameObject ifcElement;
```

```
    void Start()
    {
        // Change one of the element's properties
        ifcElement.GetComponent<IfcProperties>().ChangeValue("IsExternal", "true");

        // Export edited IFC model to the Unity project's root directory
        IfcExporter.Export("From_EditElementDemo.ifc", ifcRootGameObject);
    }
}
```