

# UPDC

## Unity to PokEngine Data Converter

### User Manual

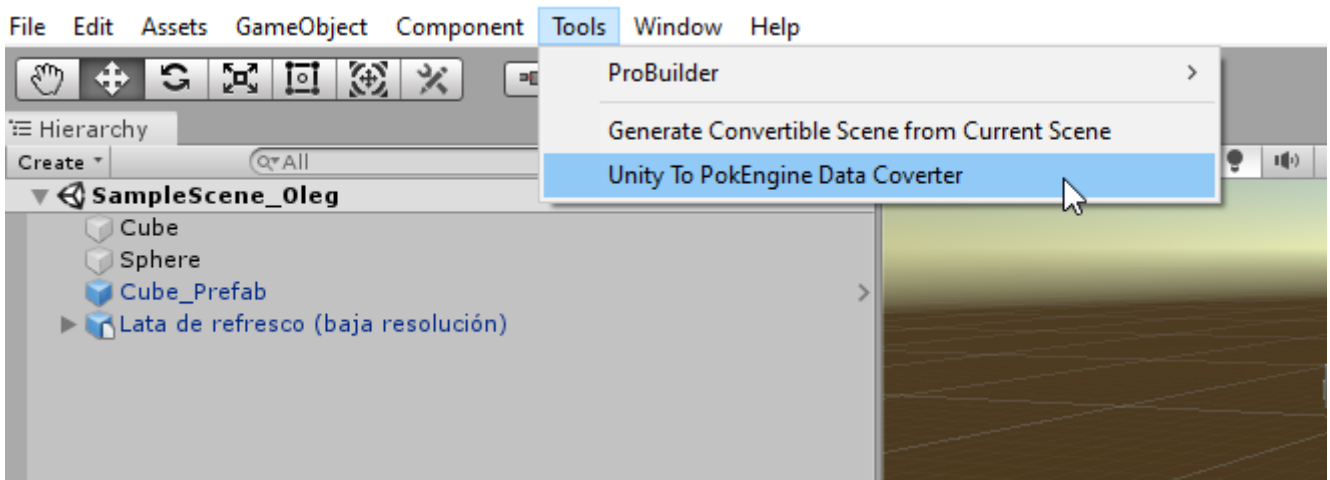
Oleg Loshkin

December 19, 2019

## 1 Introduction

The **Unity to PokEngine Data Converter**, or **UPDC for short**, is a Unity tool used to **export ScriptableObject** inheriting **.asset** files **into JSON** files that end with extensions that specify their type. These JSON files can then be read by the PokEngine's parser.

## 2 Accessing the Tool



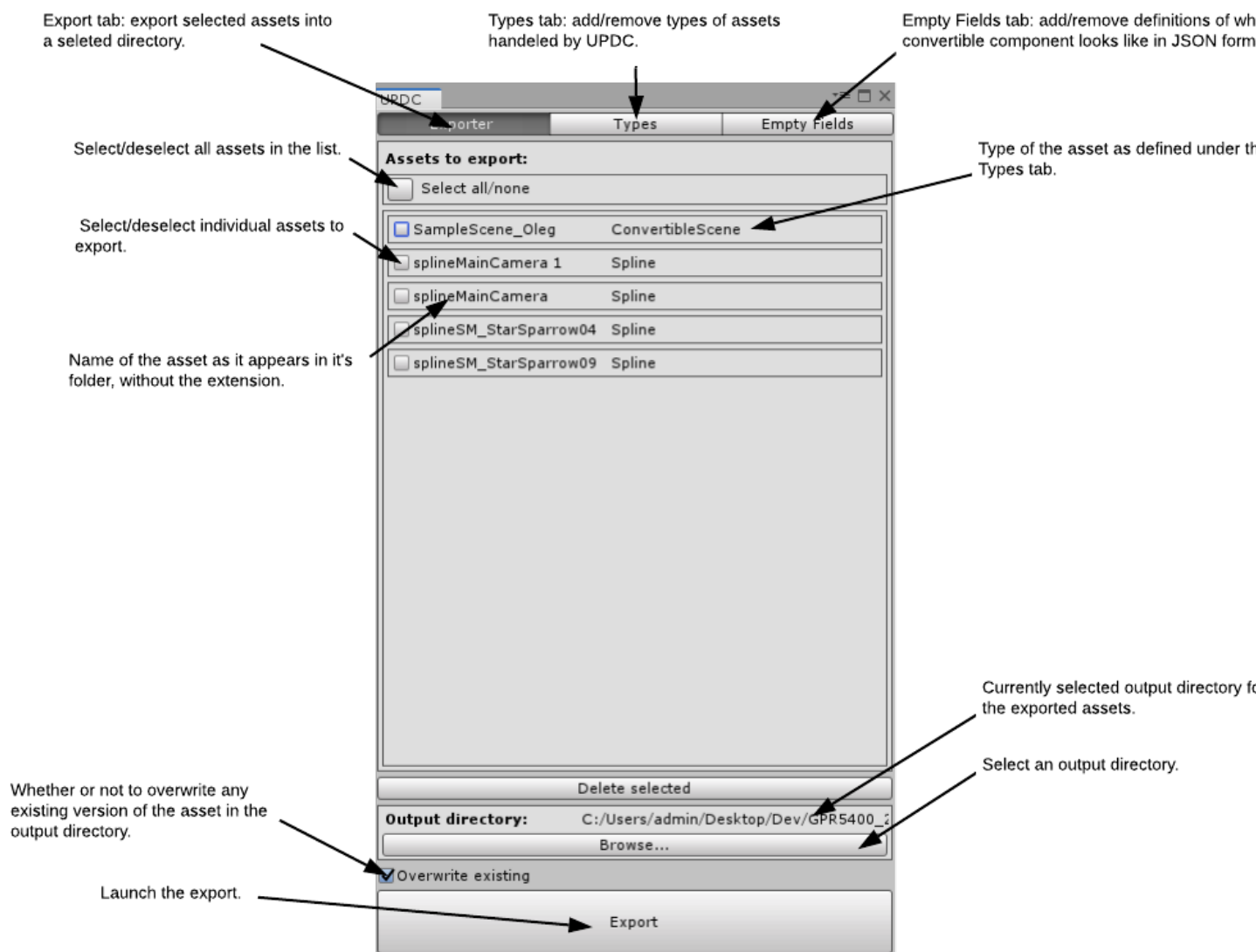
## 3 Folder Hierarchy



UPDC generates folders for every convertible type that are defined for it in the "Types" tab. These folders must contain the corresponding type of .asset files for the tool to be able to export them.

The folders located in "Assets/Data" with a name corresponding to a convertible type are folders containing assets that are not linked to any particular scene. Other folders are named after scenes and inside are folders for convertible assets that are linked to that scene.

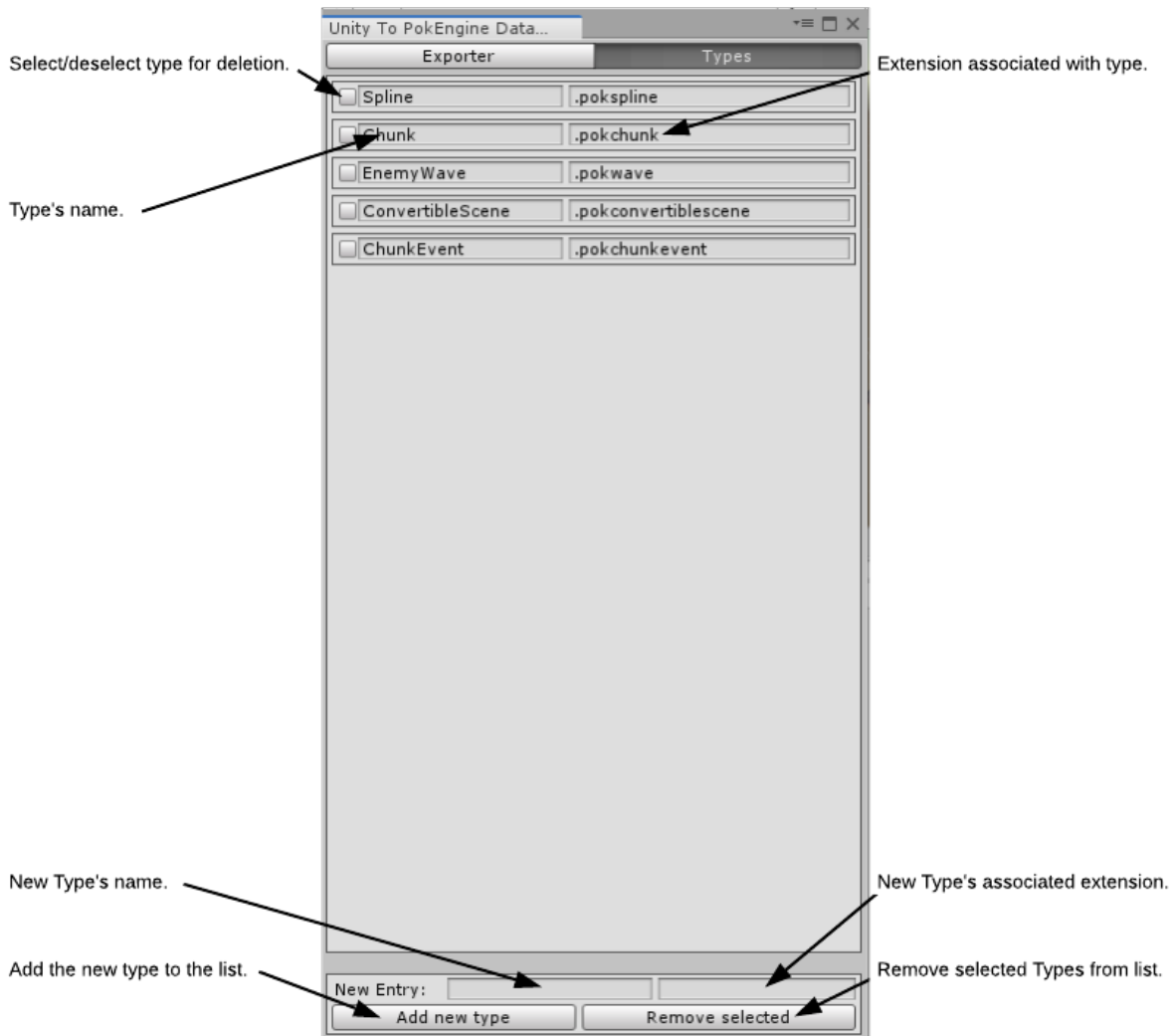
## 4 The Exporter Tab



The tool is separated between **three tabs**. In the **Exporter tab**, you can **select the .asset files located in the Assets/Data/... folders you wish to export**. You can **select the output directory** for the files generated. By default, the tool outputs the files into Assets/Editor/UPDC/DefaultOutputDir.

Upon exportation, **directories will be generated in the output directory** to mirror the structure seen in Assets/Data. Finally, you can choose whether to **overwrite any existing files** or not.

## 5 The Types Tab



The **Types tab** contains a list of **tuples that define a type's name and it's corresponding extension**. The type's name is used for generating appropriately named folders and to identify the asset's type. The extension is appended to the generated JSON's name for the PokEngine's parser to identify the asset's type.

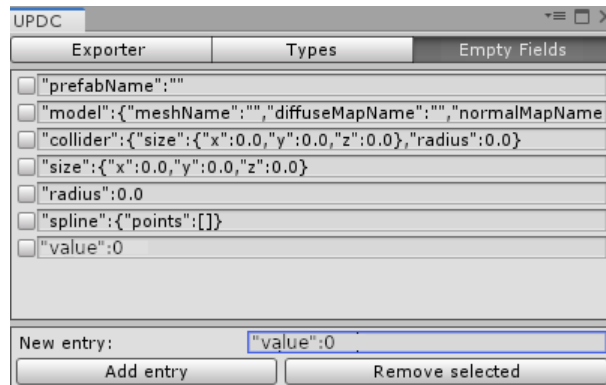
**To integrate your newly created class with UPDC, simply enter it's name and it's extension in the "New entry" fields and press the "Add new type" button. A folder for your new .asset files will be automatically generated in Assets/Data and the files will be visible in the "Export" tab.**

**To remove a type, simply select it in the list and press the "Remove selected" button.**

## 6 The Empty Fields Tab

Unity's **JSONUtility** used for exporting **ScriptableObject**s fills **any uninitialized fields** of a **ScriptableObject** with **default empty values**. This adds a lot of **useless contents in the output JSON file**. **UPDC** allows you to **define substrings to remove from the final JSON file**.

Add there the definitions of what uninitialized fields of your component look like if you wish them removed from the final result.



This results in a JSON file containing only the initialized values like so:

```
1 {
2     "myData":
3     [
4         {"value": 25.458742},
5         {"value": 0},
6         {"value": 0},
7         {"value": 0}
8     ]
9 }
```



```
1 {
2     "myData":
3     [
4         {"value": 25.458742}
5     ]
6 }
```

## 7 Integrating with UPDC

**UPDC** converts **.asset** files of types **inheriting from the ScriptableObject** class.

If you wish to export your data using **UPDC**, you will therefore need to **create your ScriptableObject** inheriting **class** as such:

```
1 using UnityEngine;
2
3 public class MyScriptableObject : ScriptableObject
4 {
5     public string someData;
6     public int moreData;
7 }
```

Instances of your **ScriptableObject** inheriting class will need to be stored in **Assets/Data/yourTypeFolder** or **Assets/Data/someScene/yourTypeFolder** for **UPDC** to manage them. **This must be done on your side** as **UPDC** does not provide any automatic way to put your **.asset** files into their corresponding folders. Next, you will need to **define your new class with UPDC** via the **"Types"** tab and **add any empty fields you wish to remove in the "Empty Fields" tab**.