

## MGPatcherTools – Introduction

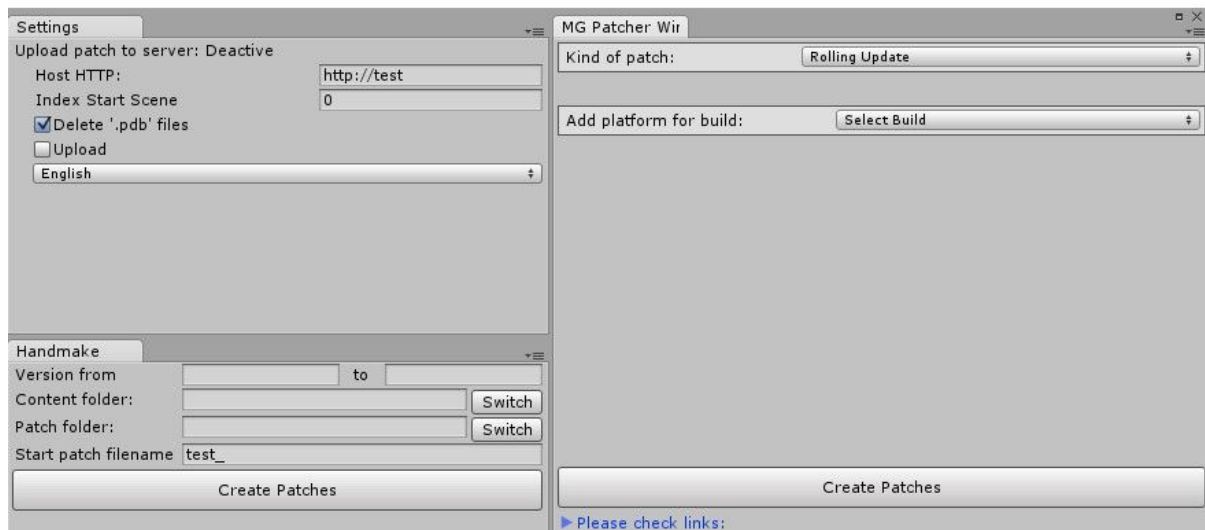
MGPatcherTools - This is a package of tools for fast and comfortable update your game. This patcher compares two versions of the game removes the missing part of the game in the initial version, the maximum compresses it and automatically loads the server. You can immediately create a patches on all platforms (Windows, MacOS, Linux, Webplayer). A huge expanse of possibilities. By purchasing this product you are unlimited to create your personal design patcher. Patcher works perfectly and in the UI and GUI.

## I. Working with MGPatcherBuilder.

### 1. Program interface.

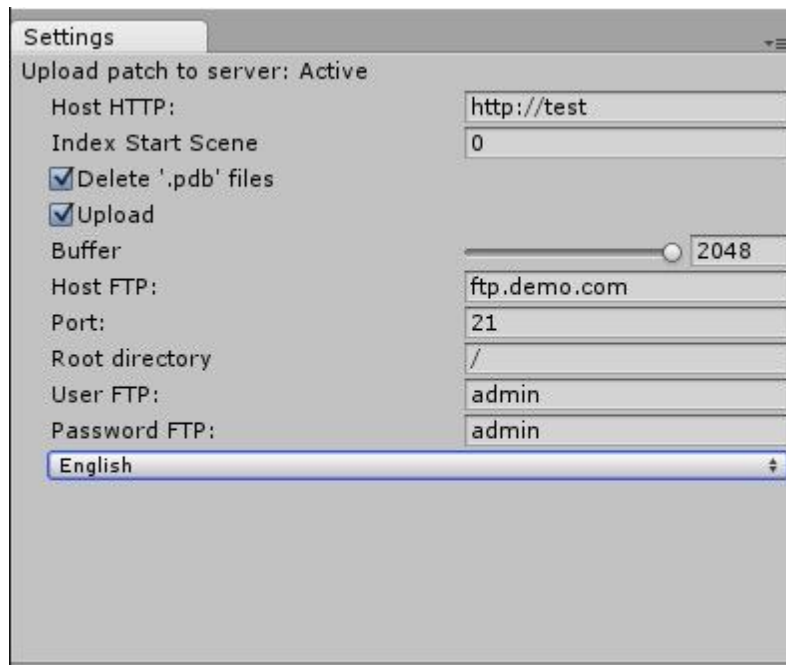
This tool has 3 working Windows..

1. MG Patcher Window
2. Settings
3. Handmake



1. MG Patcher Window - this window to configure the platforms. Here you can choose a few platforms and start auto patching.
2. Settings - this window to the global settings.
3. Handmake - this window allows you to manually compare 2 folders and generate a patch.

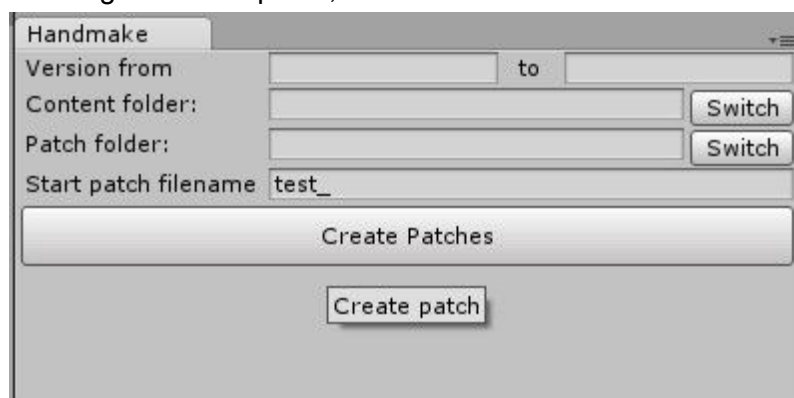
Let's see what needs to be set in this window:



1. HostHTTP(HTTPS) - Patcher to download files uses http(https). So, in this field you should enter the address to folder "path to the patches" and "path to versions".
2. Index Start Scene - Function LoadGameScene uses this number to load a scene. Function LoadGameScene used for the Play button.
3. Delete '.pdb' files - In order to remove unnecessary pdb files. This will not affect the game itself, but will reduce the size of the game..
4. Upload -For automatic upload file to ftp server.
5. Buffer - The size of the part to load.(2048-2MB).
6. Information about the ftp server.
7. The choice of interface language.

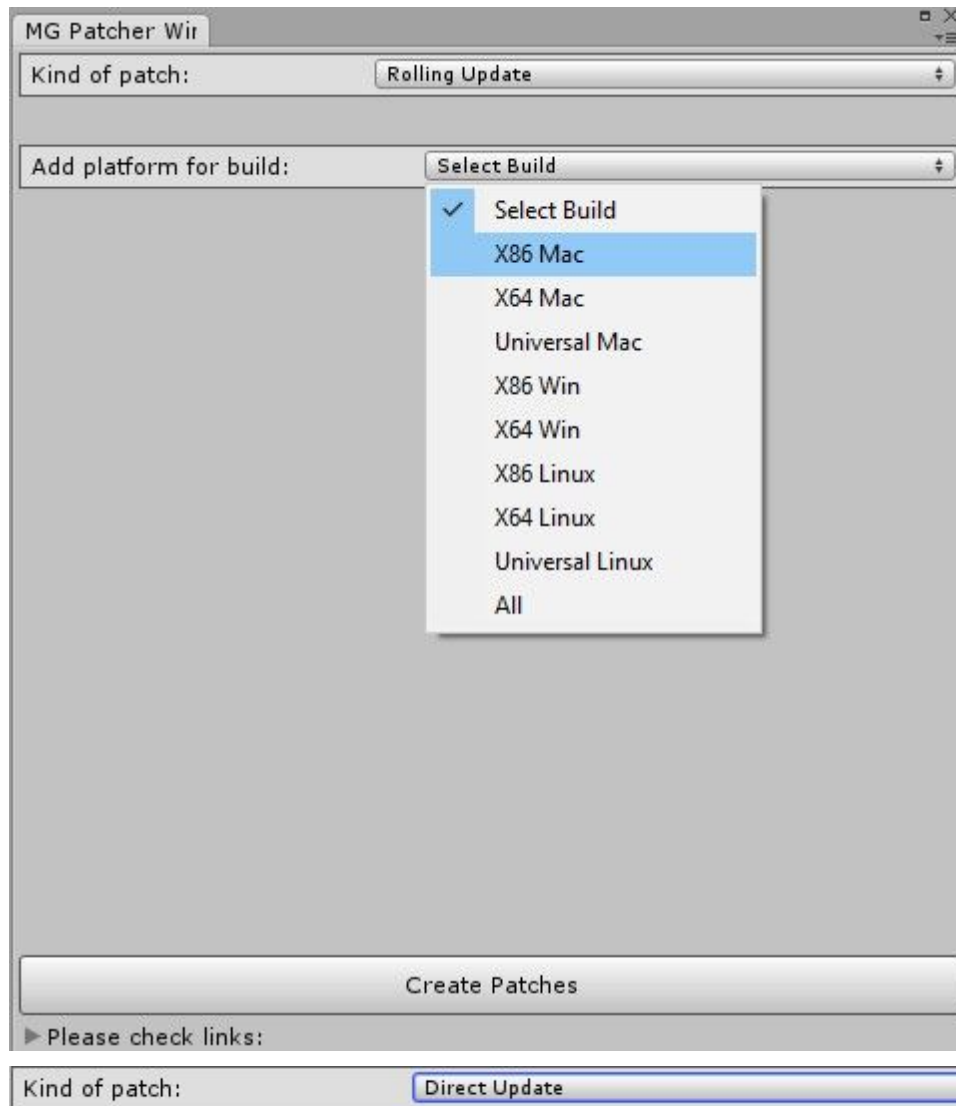
After setup, be sure to save the project.

Briefly talk about useful stuff, Handmake. To create the patch manually, you can enter the name of the first folder in the "Version from", the name of the second "to". Next, in the "Content folder" to choose a folder location, and in the "Patch folder" where to save the resulting file. "Start patch, the filename is the initial file name.



After entering the required information press "Create Patches".

Moving on to more complex parts. Let me explain how to use this tool.



Choice between incremental patches (RollingUpdate) and direct (DirectUpdate) patches. The incremental approach will take less time to create the patches, but more time to apply them. It is the other way around for direct patches.

---

*MGPatcherBuilder incremental update:*

*Old versions:*

*-1.0.0.0*

*-1.2.0.0*

*-1.2.1.0*

*New version: 1.3.3.0.*

*Patches: 1.2.1.0\_1.3.3.0.*

*MGPatcher incremental upgrading from -1.0.0.0 to 1.3.3.0:*

*Patches that are downloaded to the user computer:*

*-1.0.0.0\_1.2.0.0*

*-1.2.0.0\_1.2.1.0*

*-1.2.1.0\_1.3.3.0*

---

*MGPatcherBuilder with direct update:*

*Old versions:*

*-1.0.0.0*

*-1.2.0.0*

*-1.2.1.0*

*New version: 1.3.3.0.*

*Patches:*

*-1.0.0.0\_1.3.3.0*

*-1.2.0.0\_1.3.3.0*

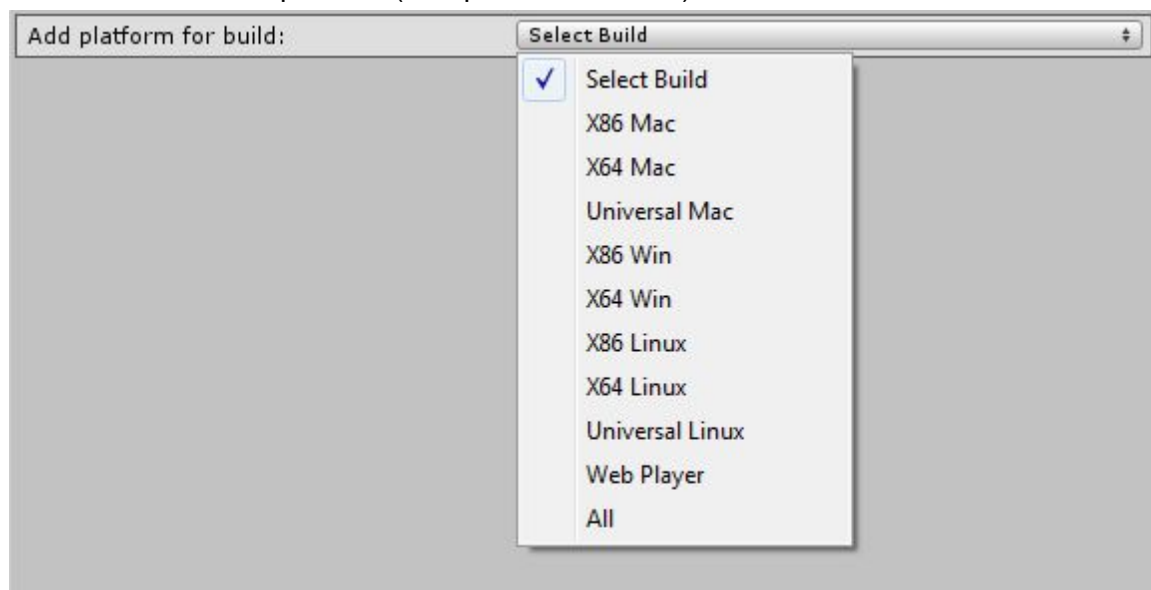
*-1.2.1.0\_1.3.3.0.*

*MGPatcher in direct update from version -1.0.0.0 to 1.3.3.0:*

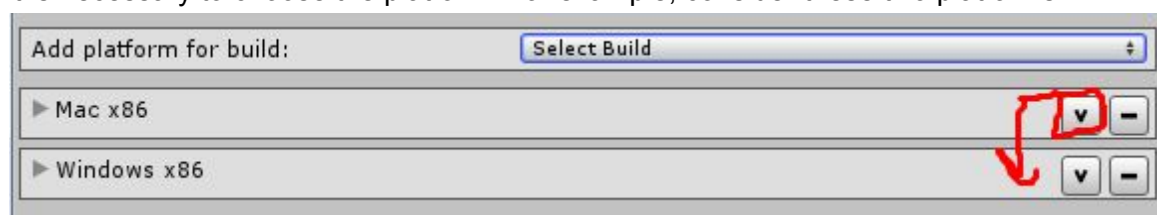
*Patches that are downloaded onto the user's computer: 1.0.0.0\_1.3.3.0.*

---

Next item is to add a platform (Add platform for build).

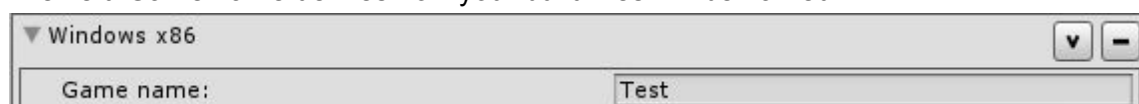


It is necessary to choose the platform. For example, consider these two platforms.

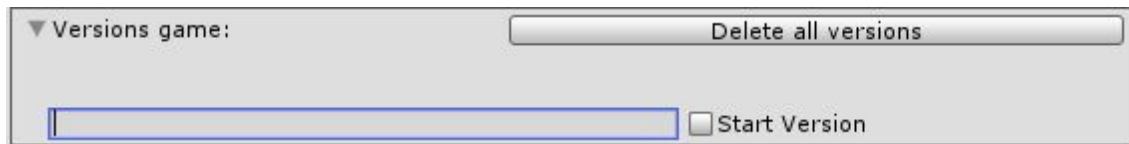


Each platform has its own configuration. For each selected platform there are two buttons – an arrow and a minus. Pressing the arrow copies the configuration for that platform from the top to the configuration of the platform below it. Minus removes the platform, but the platform configuration is not deleted.

The field Game name defines how your build files will be named.

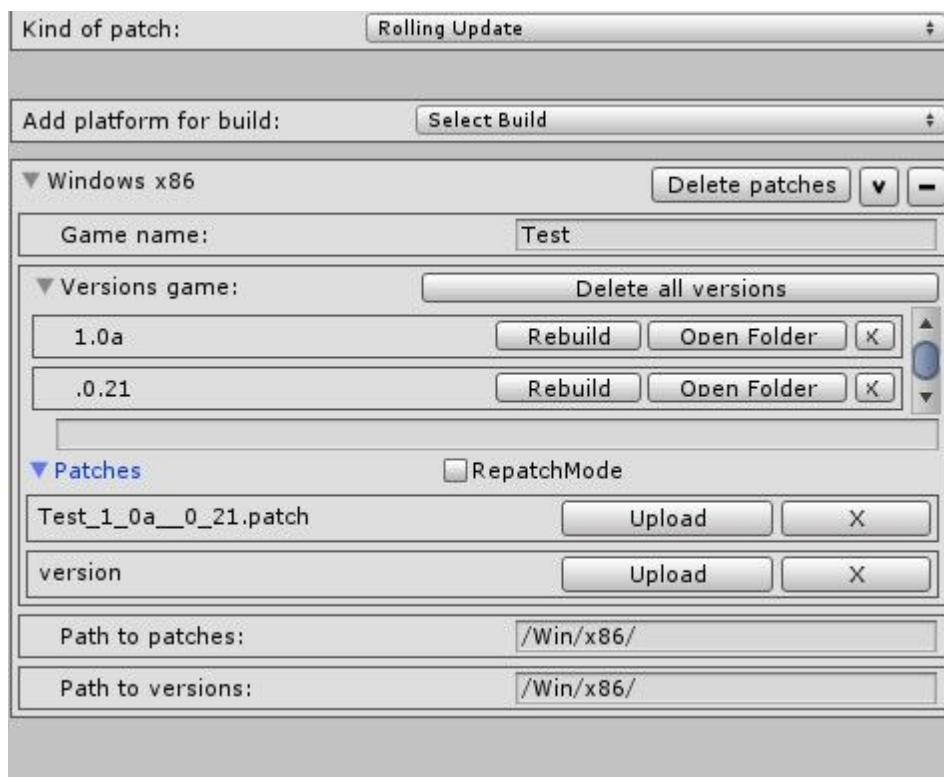


Set the name of the application without any format extensions (.exe, .apk, etc.).



Under “Versions game” you will find information on all versions that have been created for your game.

To create a version, you must write a version in the field. If you want to start, check the version launch version (Version of the start is active only if the version is not specified). The delete button deletes all versions all versions.

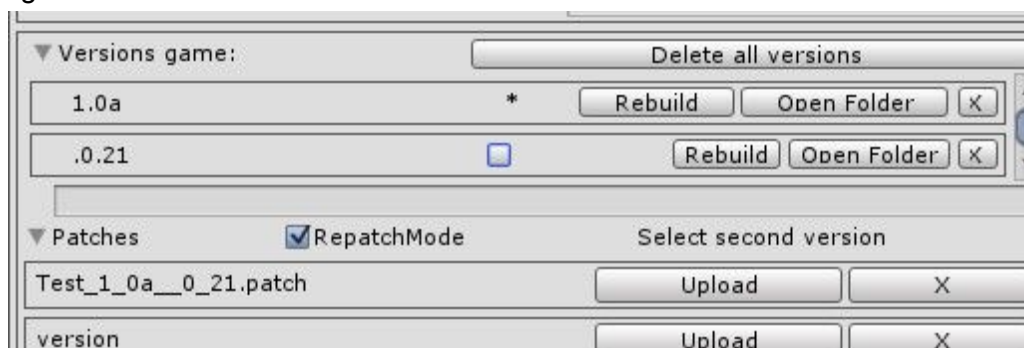


Delete patches- remove all patches of this platform from that computer.

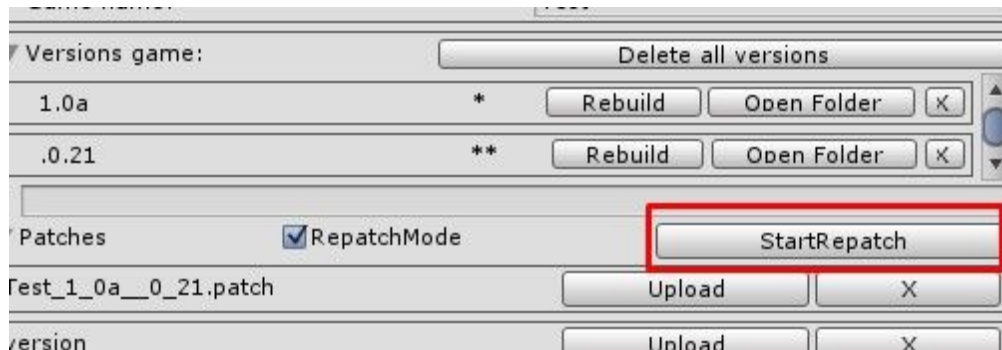
Rebuild- recompiled version of the game.

Open Folder - opens the folder with the version.

RepatchMode - allows perati in mod personday patches. After you select version 2 in the right order.



next press StartRepatch



Upload - upload a file to the server.

The sequence of actions to create 1 version:

1. To set the scene in BuildSettings.
2. To modify settings in Settings window.
3. Add platform.
4. Specify the name of the game.
5. Specify the name of the version.
6. If you're using patcher as an external program, select Start Version
7. Check the availability of links in "Please check links".
8. Press Create Patcher

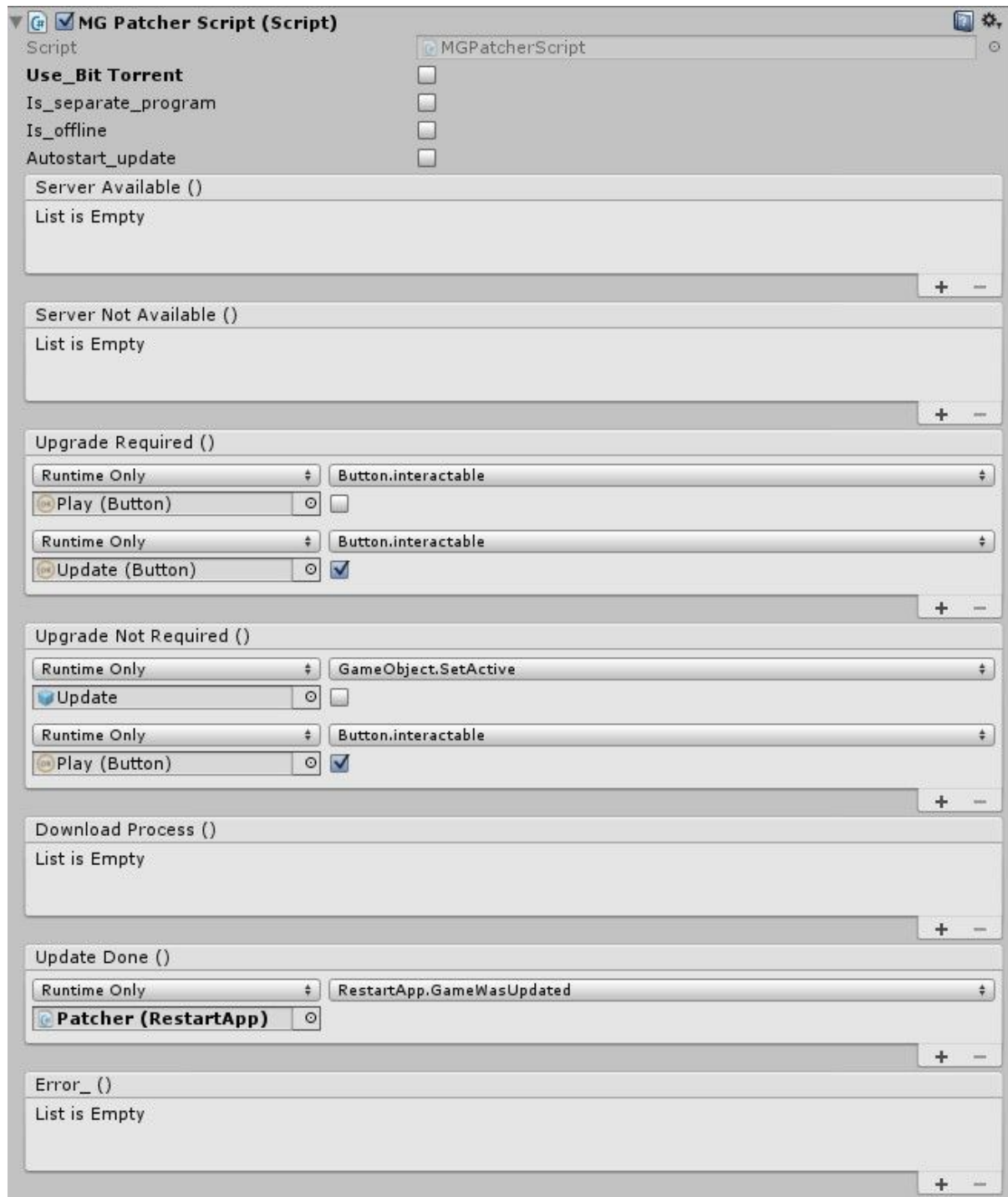
## 2. Possible errors.

1. 1. The file has not been uploaded to the server because of an **error**. Any problems connecting to the server (check the settings). Either an error when creating the patch (unlikely).
2. 2. **ERROR:** Please add the scene **in** the build **settings**. There is a scene in the scene settings (add at least one scene).

## II. Working with MG Patcher.

### 1. Program interface.

To work on any Game Object necessarily need to add a script MG Patcher Script.



Use\_Bit Torrent- when to use MGTorrent.

Is\_separate\_program - when using a third-party application.

Is\_offline - when you need to MGPatcher worked without the Internet.

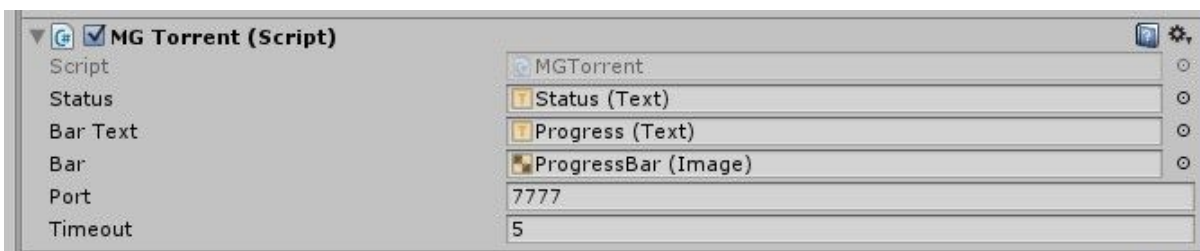
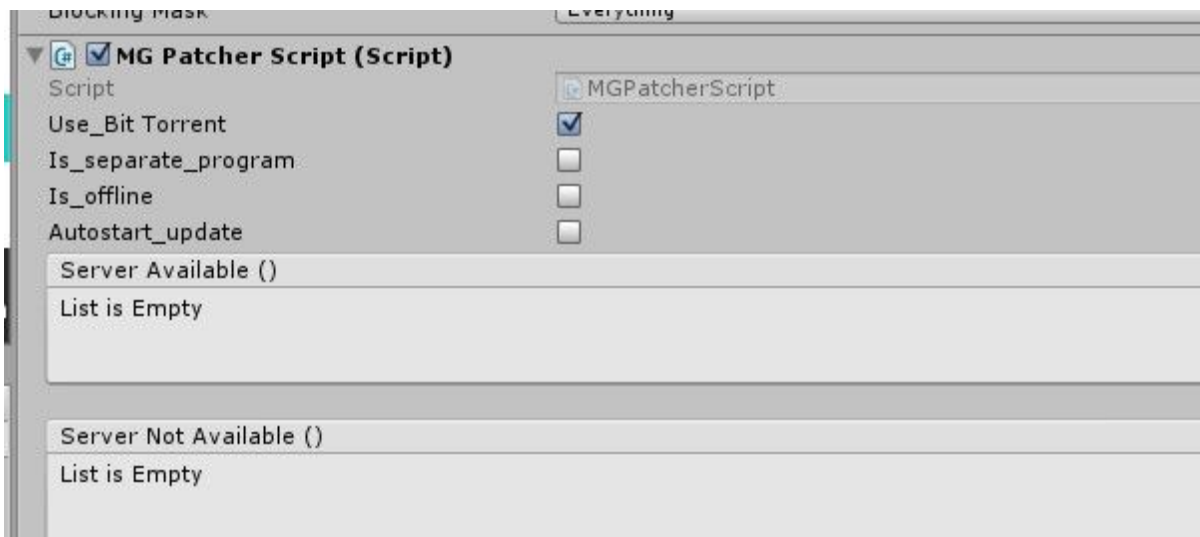
Progress Patch - required to work MGPatcherScript.

In this script you need to add different graphical elements.

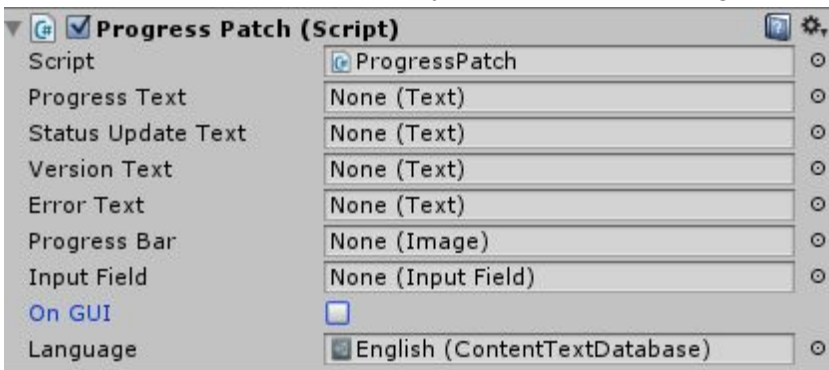
Autostart\_update - if you check, after starting the game, the app will automatically update the game.

Here is an example using MGPatcherScript with MGTorrent





The second script, which definitely need to add this Progress Path.



Progress Text - This UI Text to display the update status in%.

Status Update Text - This UI Text to output process.

Version Text - This UI Text to display the version.

Error Text - This UI Text to display errors.

Progress bar - This UI Image status updates to display graphically.

Input Field - For entering (You will need to MGDDownloader).

OnGUI - includes all information displayed in the GUI.

Language - select the language. Press and select the desired language.



## 2. The update process.

Let's look at the update process.

As soon as you run the game patcher loads into memory resources (.asset) and assign values to variables of these resources. Next patcher unloads these resources. This is to ensure that the resources of the new version could take the place of the old version of the resource. Since post File 2 stream is not possible.

After that, the patcher will download the file from the server version and reads a list of versions that are in it. If a new version of the program is looking for a suitable server patch file, it downloads and updates. Further, if the update is successful, the program runs the scene that you have selected the check box to create the patch.

## 3. The Separate Program

**This function** is mainly used to make the launcher. To work you need to create a build with any name you like after checking the information in MGTPatcher.asset. Next you need to create the file "CurrentVersion.txt" where to specify the current version. Now you can create the patches. The game files must be in the same folder where the patcher program(launcher).

## 4. Possible errors.

1. There **is** no information about the **current** version! - Golf current\_version\_string in MGTPatcher.asset empty.
2. The server version of the file **is** missing! - The server version of the file is missing.
3. Patch file **is** incorrect: {0}. - Patch file is corrupted when you create or incorrectly downloaded.
4. Invalid patch: {0}. - Patch file is corrupt.
5. Patch broke client: {0}. - Patch violated the program.
6. Version file **is** incorrect: {0}. - The file is corrupted version, may have been downloaded properly.

## 5. Creating a new patcher.

Let's create a new scene.

Create eight UI objects:

- 2 images.
- 4 texts.
- 2 buttons.

Call them as it is shown in the Image:

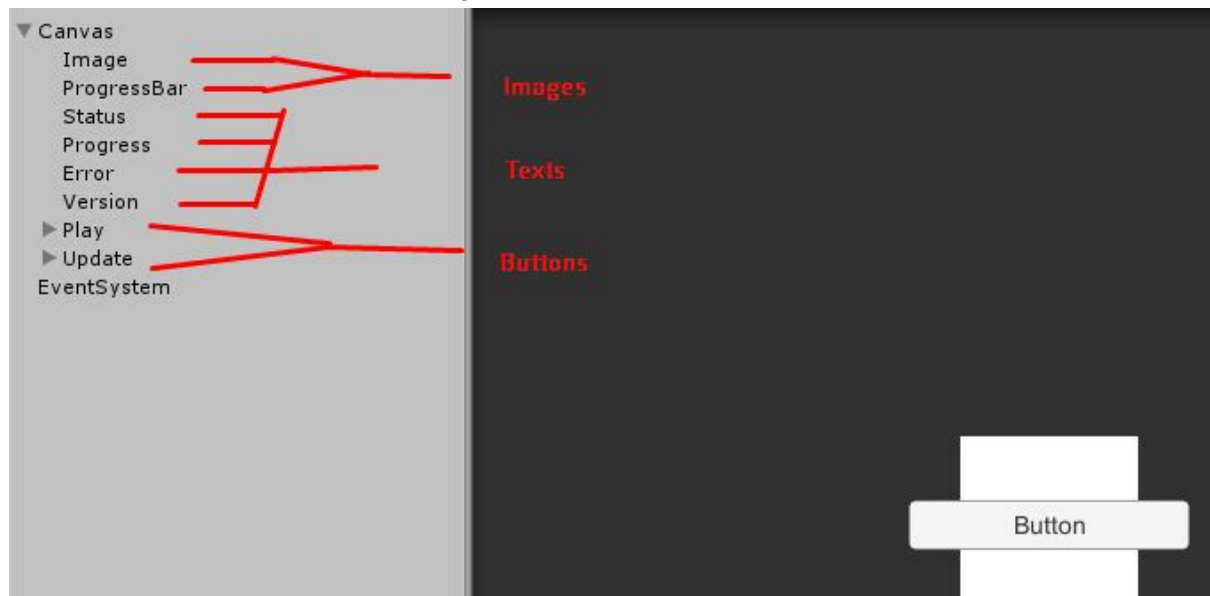
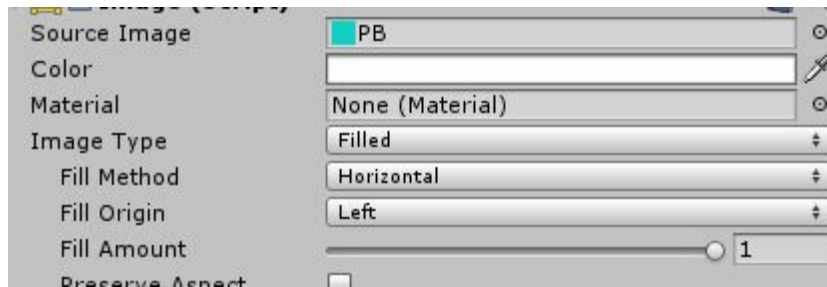


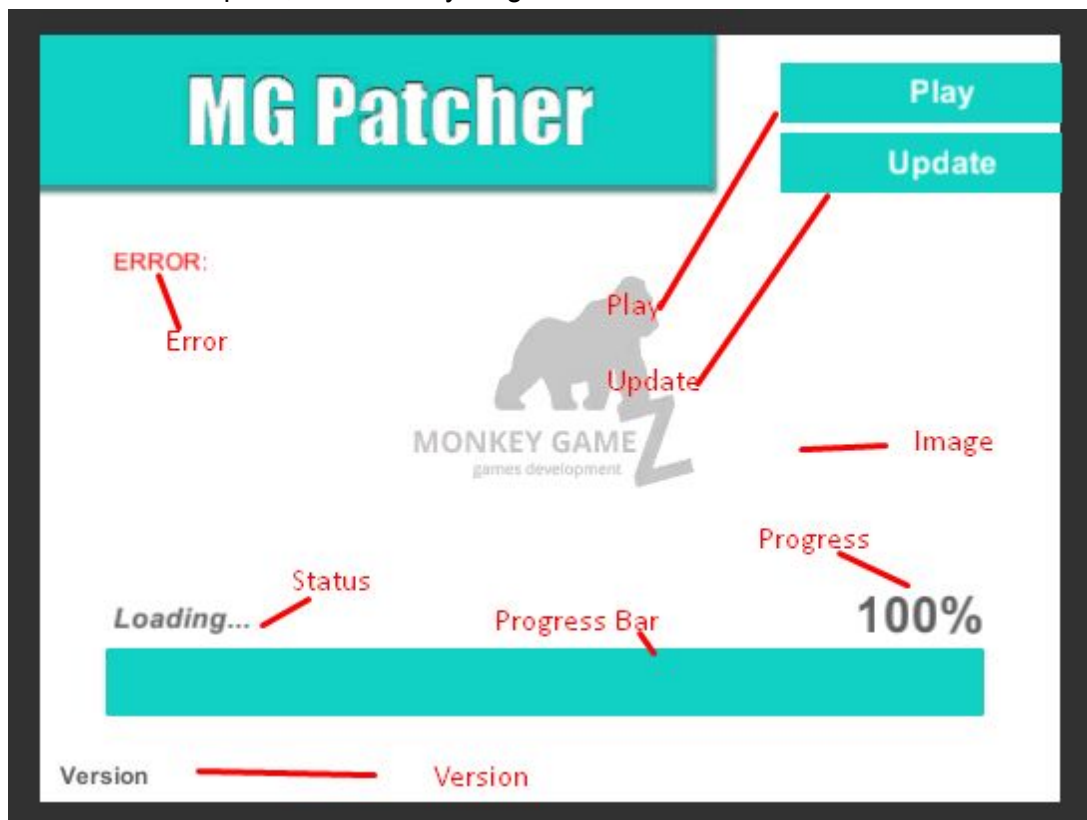
Image background so it will enhance it and add to it the image.

ProgressBar is our graphical progress. Adjust the size of the progress bar and add to it the image.

Set Progress Bar Image as shown below:



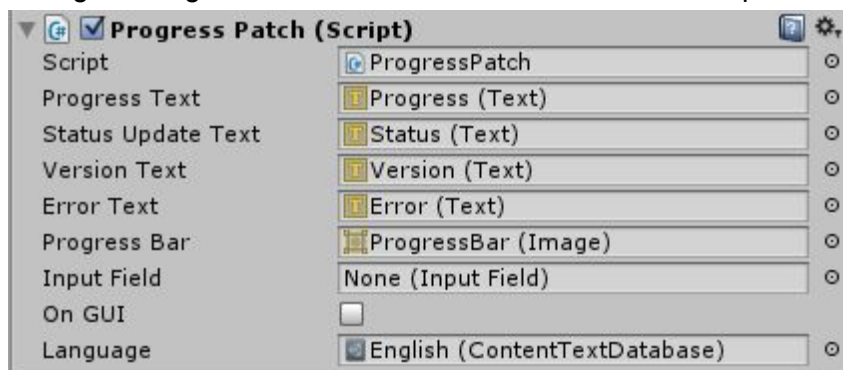
Then I will not explain, I think everything is clear. You should have this:



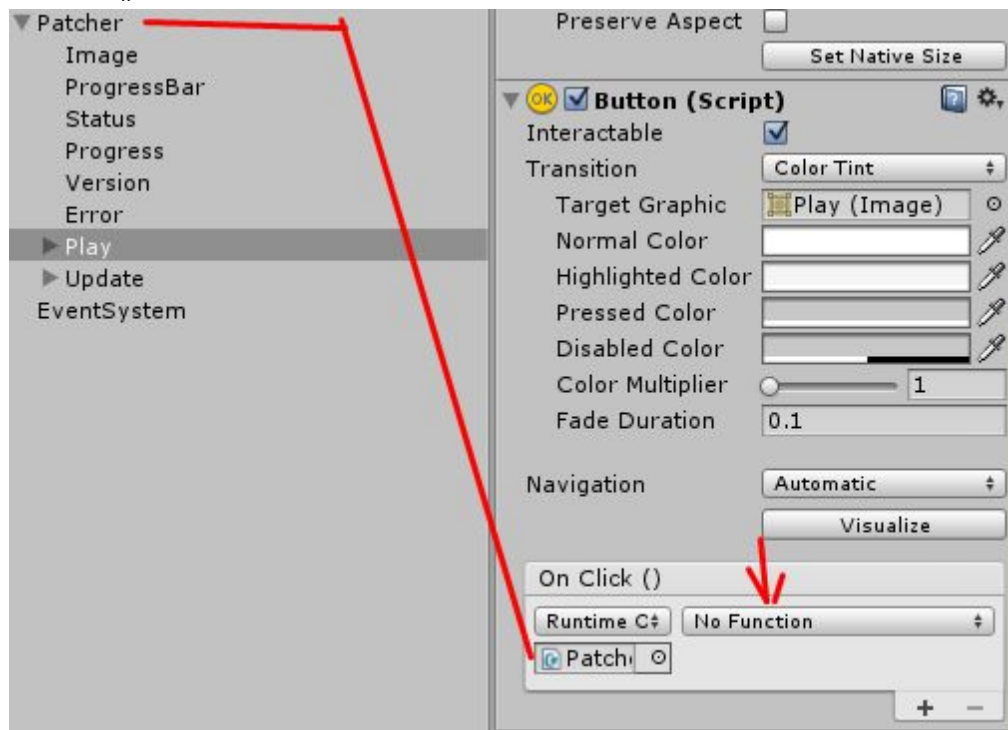
Rename Canvas in Patcher. Add to it 2 script:

- MGPatcherScript
- Progress Patch

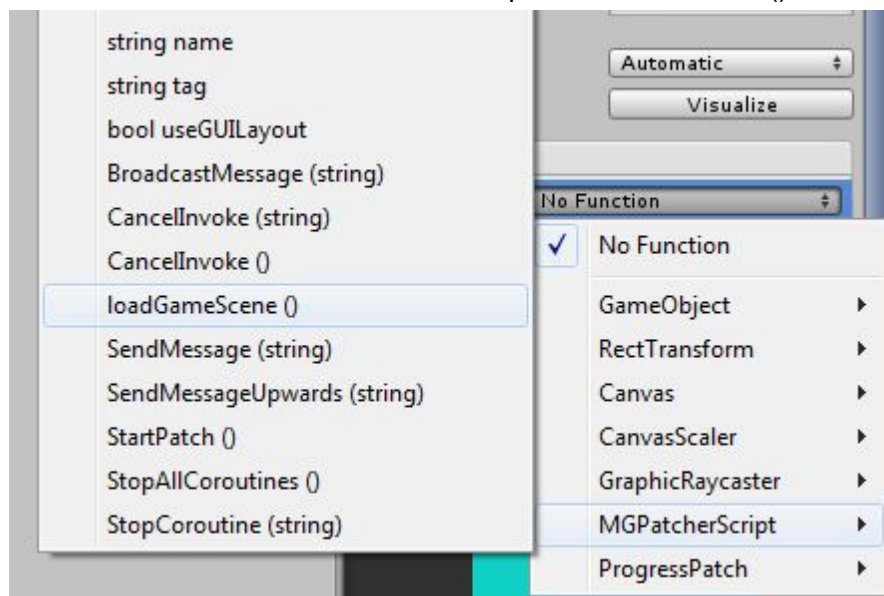
Configure Progress Patch. Add all UI elements in the script as shown in the picture:



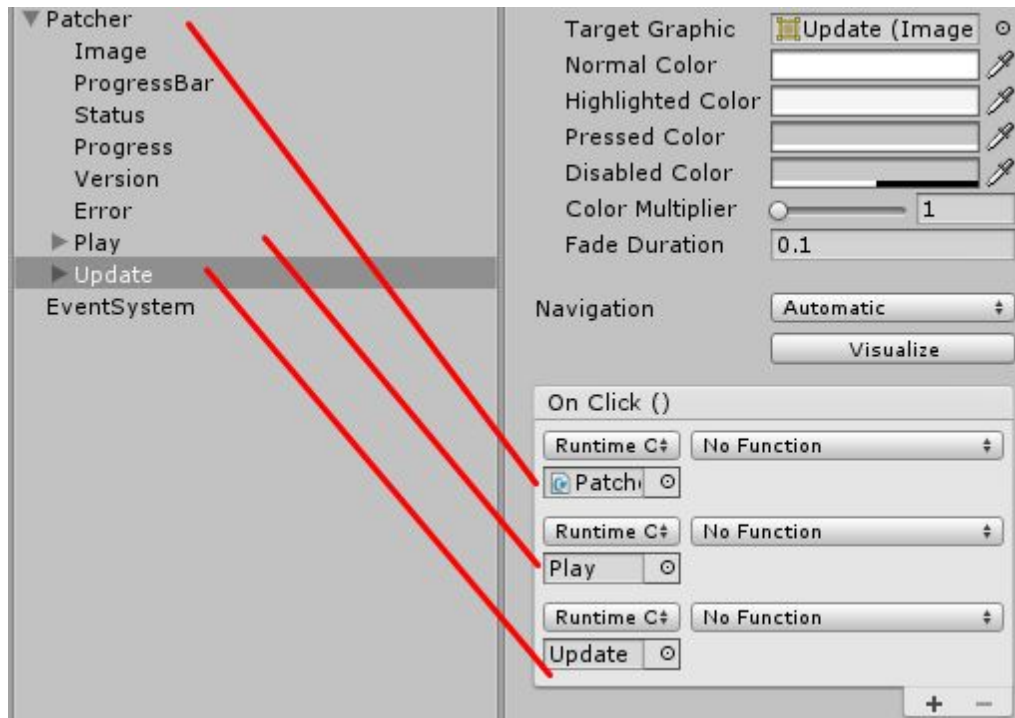
Select the Play add window in the On Click () and drag the GameObject Patcher in the OnClick () as shown below:



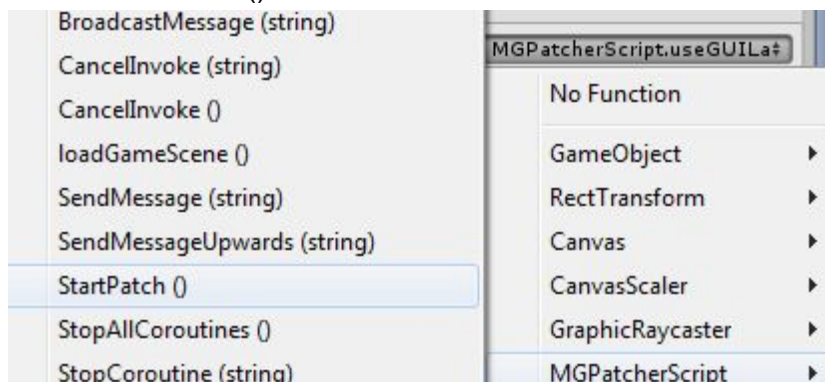
Click on NoFunction -> MGPPatcherScript-loadGameScene ().



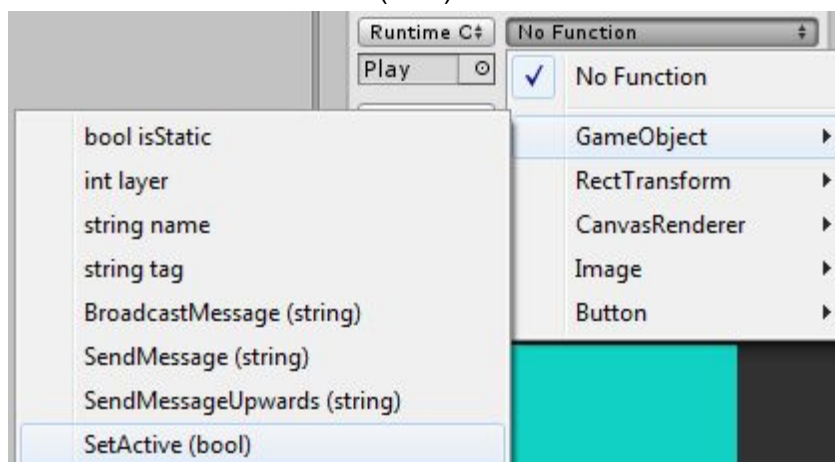
Select the Update button and edit OnClick () as shown below:



Select Start Patch () for Patcher.



For the rest select SetActive (bool) set false.

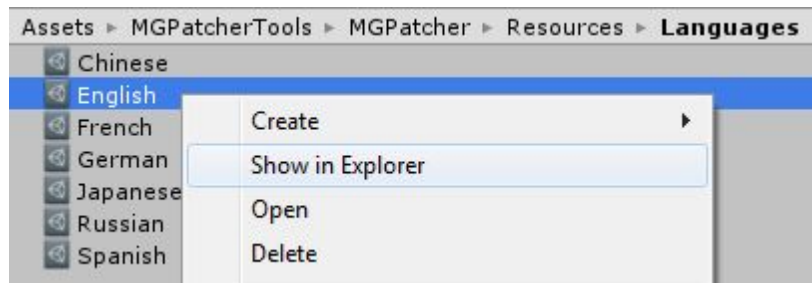


Save the scene.

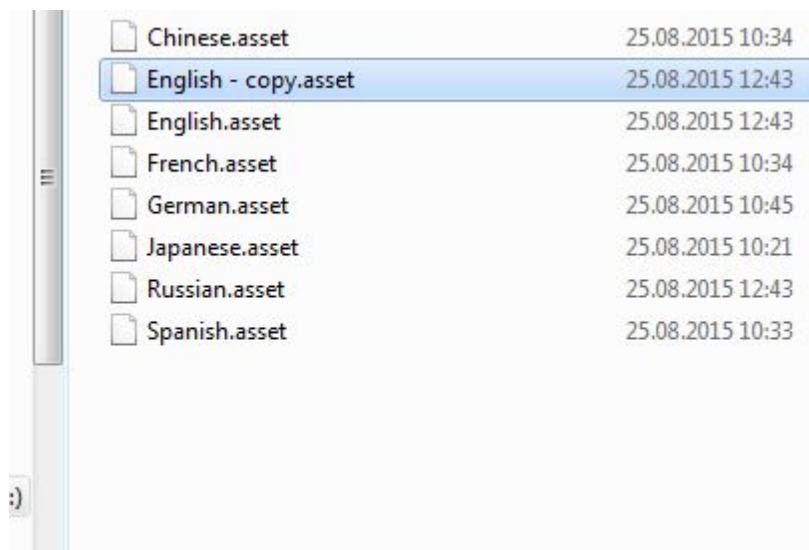
### III. Working with MGLanguages.

## 1. Creating language for MGPatcher, MGPatcherBuilder and MGDownloader.

To create a language go to "Languages". Go to the Explorer by clicking on the "Show in Explorer" ("Show in Finder" on Mac Os).



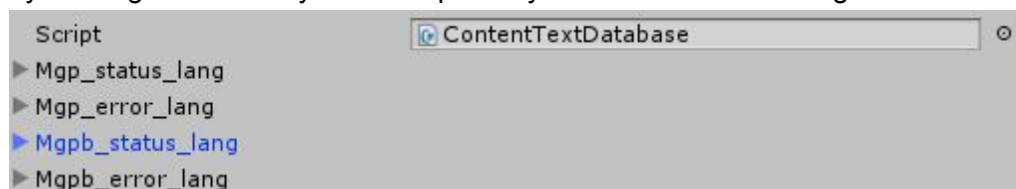
Duplicate any language and name.



I called "NewLanguage".



By clicking on it in Unity in the inspector you will see the following:



Mgp\_status\_lang - text status updates and MGPatcher MGDownloader.

Mgp\_status\_error - text error MGPatcher and MGDownloader.

Mgpd\_status\_lang - text interface MGPatcherBuilder.



Mgpd\_status\_lang - text error MGParserBuilder.

Change the line because we need you. Language for MGParser and MGDownloader ready. That he appeared in MGParserBuilder, do the following:

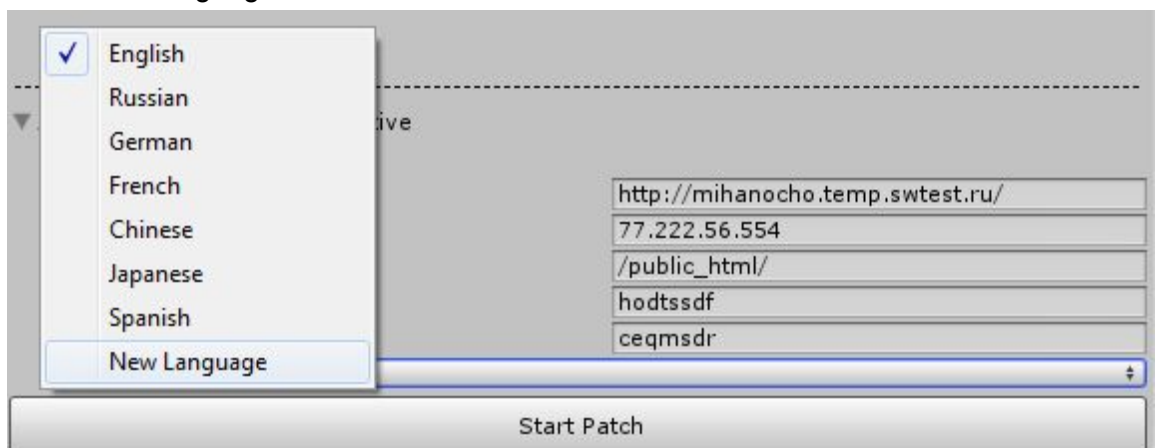
1. Open the script MGParserToolsDataBase.
2. Find enum Language and add a line with the name of your language.

```
public enum Language {  
    English,  
    Russian,  
    German,  
    French,  
    Chinese,  
    Japanese,  
    Spanish,  
    NewLanguage  
}
```

3. Open the script MGParserToolsScript. After about 507 line should be the following:

```
507 switch (mgptdb.lang) {  
508     case MGParserToolsDataBase.Language.English:  
509         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/English");  
510         break;  
511     case MGParserToolsDataBase.Language.Russian:  
512         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/Russian");  
513         break;  
514     case MGParserToolsDataBase.Language.German:  
515         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/German");  
516         break;  
517     case MGParserToolsDataBase.Language.French:  
518         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/French");  
519         break;  
520     case MGParserToolsDataBase.Language.Chinese:  
521         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/Chinese");  
522         break;  
523     case MGParserToolsDataBase.Language.Japanese:  
524         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/Japanese");  
525         break;  
526     case MGParserToolsDataBase.Language.Spanish:  
527         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/Spanish");  
528         break;  
529     case MGParserToolsDataBase.Language.NewLanguage:  
530         mgptdb.ctd = (ContentTextDatabase)Resources.Load ("Languages/NewLanguage");  
531         break;  
532 }  
533 }
```

4. Add a "case".
5. Done! Now language is in MGParserBuilder.





# Conclusion.

Thank you for what you bought this package. I hope it will be useful to you.  
I would be grateful if you leave a good tip on AssetStore.



**Sincerely, Michael A.**

**Authors: Michael A.**

*2015 All rights reserved*



