CRAFTY

v 1.2 Documentation

Contents:

Important Note:

We strongly advice on cleaning Crafty from the project (that also include the in-project folder created by Crafty) before downloading new version.

1. Introduction

Welcome to the tutorial and presentation section of the Crafty program, made by Developers for Developers.

Crafty is a two in one program that allows you to create patches from comparing two of your projects on the binary data level. Patches made from" Patchie" part of the Crafty will be able to change, replace and delete data that you don't want or need in your projects.

The second part of the Crafty program is" Launchie" it's an in game Launcher made from your own GUI design. No matter how complicated it is you have as much power over it as your knowledge on Unity GUI or other program like NGUI, iGUI and so on. Together these programs will check for updates and have instant reaction when there is patches ready to be downloaded.

And with the new very powerful outside installation system you can expect patches to be installed in perfectly correct order just like you would expect from triple AAA programs.

Crafty itself is a program that will work on Windows, Mac and Linux also in the future we plan to make it available for all mobile devices. We have under our sleeves many tricks and add-ons that we want to add to the Crafty: speeding it up, giving it new possibilities, and of course keep in mind that price of 50 $ will NOT be increase.

As Indie developers we understand low income of developers, we respect that and we will make sure to keep price as low as possible while not taking away anything from Crafty Project.

NOTE: Crafty works on both Indie and Pro version of Unity from Unity 3.5.7 and all version of Unity 4+.

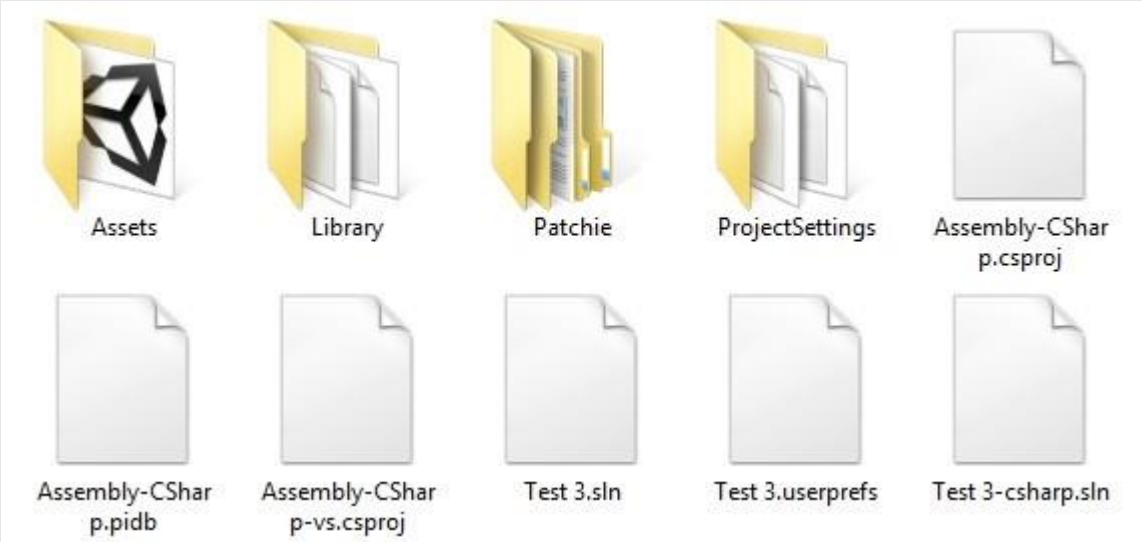
With that a said let's move on to the tutorial part.

2. Installing

The moment you will download Crafty it will be sent to the EDITOR folder please, do NOT move it from that location, it is there so that developed game will not copy the Crafty program, making loop error while creating the patches.

Crafty it self is a very smart program, it will create its own work place after you will turn it on for the first time (NOTE it will not work properly with a build in games if you don't turn on Patchie from the window first). It will create a folder in the project named" Patchie" so it will be easier for you to find, in it you will locate one more folder called "_current ". Both folders are very important and we strongly advice to not change their name.

Example of Patchie folder located in the game project.



Test patches created in the Patchie folder. (_output folder is the location of all your patches, more on that in the Patch Creation section).

Remember to build your games (that you want to use for patch creation) in" _current" folder, so that while making patches Crafty will know that it's Unity Game.

Don't worry you can call your game whatever you want, you don't have to call it (for example) Penkura version. 1.0.0 BETA, you can just call it whatever you wish. After Patches will be made or when you will feel that the Project folder is getting too big for your taste, just delete previous versions of the game. Only the Current version is important because from that version you will created update patches. More information's about the "_current" folder will be given in the next part of our tutorial (Patch Creation).

3. Patch Creation

Crafty is under the Window's button at the top of Unity, in it you will find two types of Patchie systems, first one is the Patchie (which you will use to update your game) and PatchieCompat (This is Compatibility patchie which is used to update project that already use Crafty 1.0.5 to Crafty 1.1) when you turn it on a window will appear, allowing you to create the gaming versions, and the overall patches (also the first time opening" Patchie" will create a Patchie folder in your project where all the data will be stored).

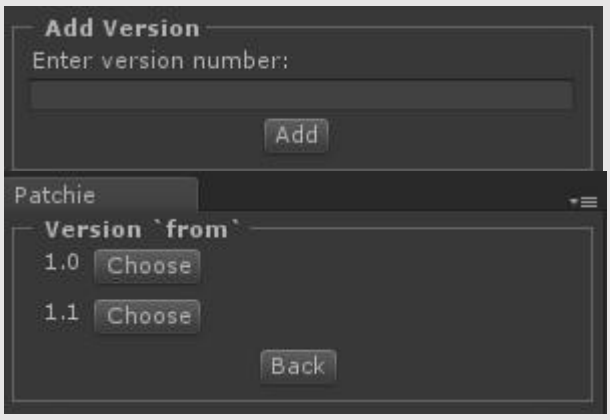
For Video version of this tutorial please click HERE. (Please note that video was made using the older version of Crafty plug-in and it may be out of date).

This is the Patchie window, as you see there is information that the game should be built in the " Patchie/_current " besides that there are two Versions Placements "From" and "To", that's the place where you will insert your builds that you generated from your project, you will use them to create patches from the differences between them.

Under that is the most important part, Packet Size.
Here you need to specify the binary size search limitation. Simply put this is the amount of bits Patchie will skip each time in search for changes till it will compare all the files between the versions. The smaller the number the more complex search, ending in smaller patches but at the cost of the time.

We recommend to not specify smaller packet size than 16 or bigger than 36 000.

Side note:
Don't be worried we already created some security for you, so you won't be able to create two of the same versions or create any patches (or game version) without a name.



The moment you will build a new version of the game the Add Version will change to" Enter version number" this is the Version of your game. In this tutorial we will call it 1.0 and 1.1 to make it easier to understand.

After creating two versions type of your game just press the Choose button, the window will change so that you can choose between the builds you created, choose the two from which you wish to create the Patch. Remember the "FROM" version is the previous version and "TO" is the next version.

Side note:

As it will be stated in the" Launcher creation" section, the version name of the patch needs to be the same as the version name in the launcher. This will be explained later on.

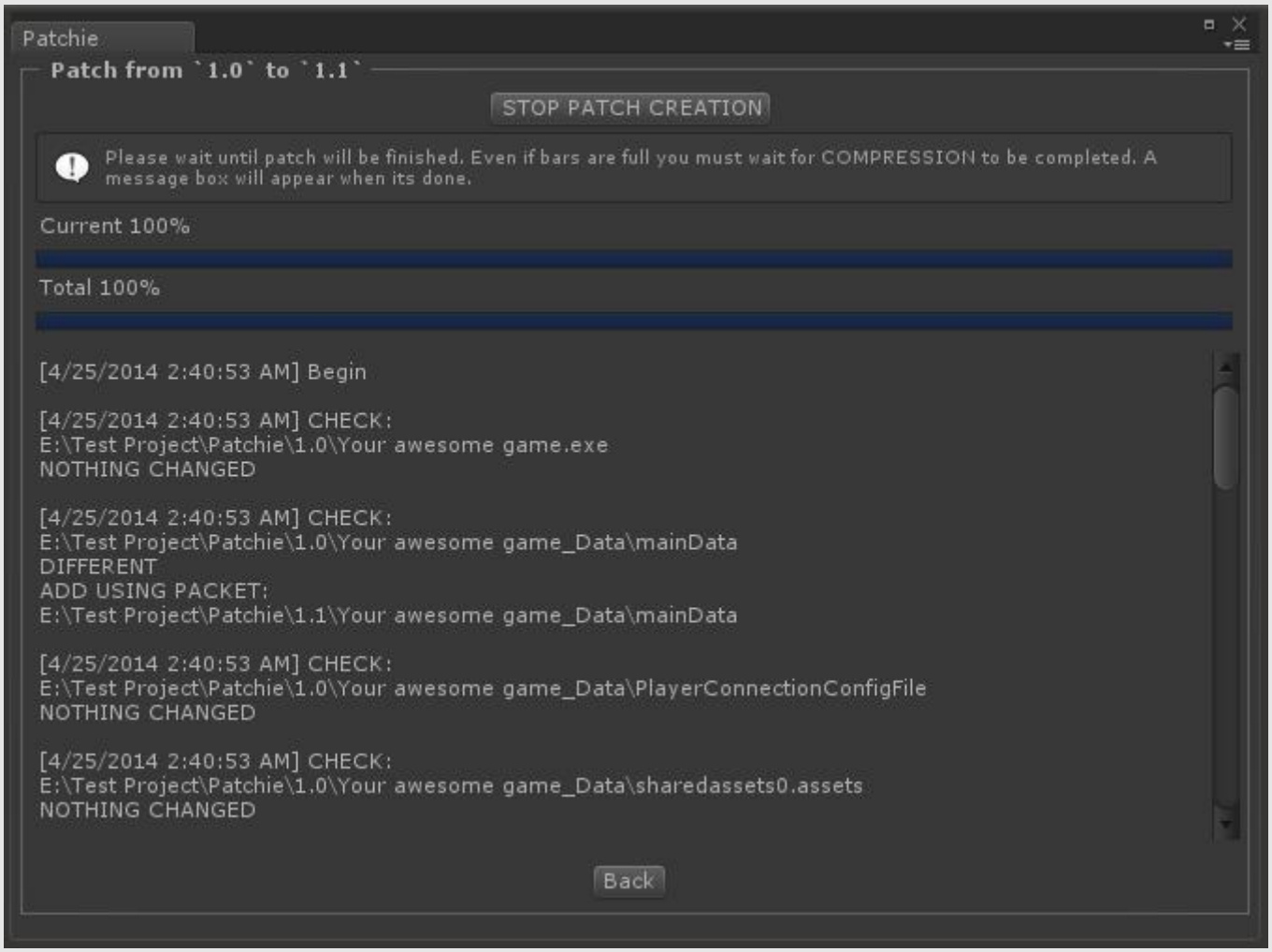
After choosing which versions/builds you want to use to create the patch a new window will appear where you can write all your “Release notes”, those are the Patch changes that will appear in the game while player will download the patch, last part is to click “Make” Button and you are done.

Two blue lines will appear that will indicate how long you have to wait till your patch will be finished.

Your patch will be nicely compress and will appear in the” Patchie” folder in your Project Files, there will be also a TXT file with your patch release note, it was made that way so you can change the information, even after the patch creation.

Below you will find more detailed information on patch creation.

In the window it self you will see not only progress bars (informing you on the patch creation process) but also a button allowing you to stop patch creation. NOTE: you can stop patch creation only when patch is being created NOT when it’s being compressed.



Also while patch is being created you will have full access to all information on what Crafty is actually doing with your builds. At the end of each line you will see one of six messages which are:

ADD – Crafty located a difference in file.

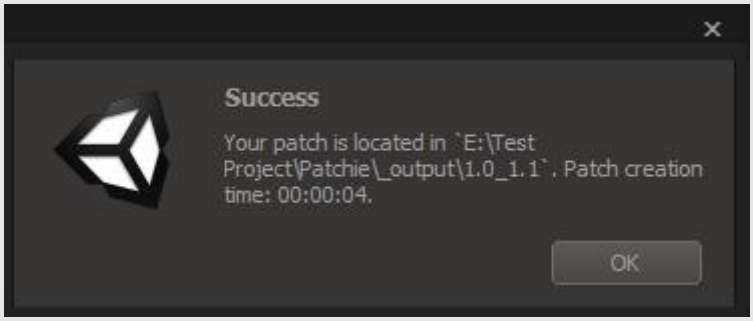
CHECK – Crafty is checking the file for any difference.

REMOVE – Crafty removes a file as it is not located in the next version.

NOTHING CHANGED – this means that Crafty didn't found any changes between the files located in both builds.

DIFFERENT ADD USING PACKET – this message means that Crafty located a change between the builds and will add them to the patch using packet system.

DIFFERENT ADD – Crafty have located a difference between the builds and will add it directly to the patch.



When Crafty will finish patch creation a window will appear informing you that the patch is created, it will show the location of the patch and patch creation time as you can see below.

NOTE: If you didn't get this message window it means that patch is not yet completed. Even if progress bars are full it takes time to compress the files which is not possible to provide process of from inside of Unity.

Patches will be installed outside of the game/project to prevent any errors while merging files. A small window will appear with installation process. After patches will be installed game will turn it self on automatically without the need of manually starting the game.

This is the End of this Tutorial Part, your Patch is ready to be sent to the game. Let's move on to the launcher creation.

4. Launcher Creation.

Launchie is a second part of the Crafty program that allows you to create an in-game launcher that is connects with your website or homemade server.

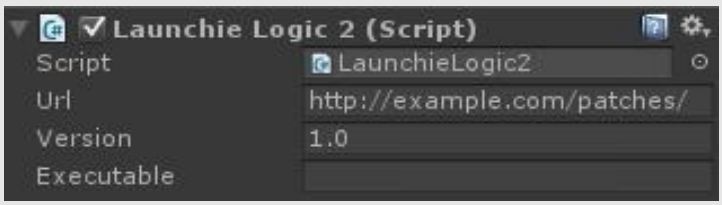
For Video version of this tutorial please click HERE. (Please note that video was made using the older version of Crafty program and it may be out of date).

Since Crafty 1.0.4 there is a prefab of Launchie in your Crafty folder. This allows you to simple drag and drop Launchie in to your scene. (It's ready to use Launcher with very basic GUI made purely in Unity)

In the prefab you will have two important scripts connecting your launcher with the Crafty and server.

Note: that it is important to keep both of the script in one game object. Those scripts are, LaunchieGUI and LaunchieLogic. Thanks to this new solution you will be able to update Crafty without damaging your work that you put in to adjusting LaunchieGUI to your game/project.

After connecting Launchie with your scene (note that we strongly advice to connect Launchie to your main menu scene) you will have three empty bars to fill.



Url which represents address link to your folder with the patches and versions.txt on your website.

NOTE: Remember to always start http:// in the URL.

Versions which represents the version number of your BUILD (it's very important that the version number used here in Launchie Logic be exactly the same as the versions numbers used in versions.txt)

Executable which is simply the name of your game/project. This information is used by the out-side patching system. For example Test.exe (for Windows) Test.x86 or Test.x42 (for Linux) Test.app (for Mac). Remember to add .exe for Windows builds, .x86 or .x42 for Linux build and .app for Mac builds.

Note: Keep in mind that the version in the LaunchieLogic needs to be exactly the same as the version name you are using while creating the patch.

At the end of the LaunchieGUI script will be located a simple GUI code that represents the main graphic part of your Launcher. Note that it uses a build in Unity GUI functions which means that you have 100 % power over the look of your Launcher. The more knowledge of Unity GUI you have the more advance Lanchers you can create.

Since Crafty 1.2 you will have not one but three test scene created by us to showcase numbers of features to make it easier for you to decide how you want your Launcher to work.

First test scene will present our typical launcher where you need to allow Crafty to download the patch by simply clicking "download" button.

Second test scene will automatically download all patches and turn the game off to install them correctly which after will turn game back on.

Third test scene will showcase our in-game error log, THIS SCENE WAS BUILD TO NOT WORK PROPERLY. You will notice that URL is incorrect, this is intentional error to showcase in-game error log.

Note: LaunchieGUI will be adjusted to the new GUI available in the Unity 4.6.

That is the end of the Launchie setup (for a basic users) .

All information below are for more advance users that want to reprogram Launchie for much more complicated purposes.

Launchie(string url, string currentVersion) - constructor, it creates instance of Launchie class and setup location of patches and current version of game

```
// url and currentVersion can be hardcoded of taken from public vars
Launchie l = new Launchie( url, currentVersion );
```

bool Check() - method for checking server if there is any patches to download. After result is cached to reduce web requests.

```
bool can_download = l.Check();
```

bool ForceCheck() - same as above, but it force check for patches (don't use cached result)

```
// 1 web request
bool can_download = l.Check();
bool can_download2 = l.Check(); //will be same as can_download

// 2 web requests
bool can_download = l.Check();
bool can_download2 = l.ForceCheck(); //can be other than can_download
```

void DownloadReleaseNotes() - method for download release notes for patch

```
// because it's async function, you need use
// setOnProgress / setOnDone methods

// l.setOnProgress it's not needed, due to small size of Release Notes
l.setOnDone( OnDownloadReleaseNotesDone );
l.DownloadReleaseNotes();

---

void OnDownloadReleaseNotesDone()
{
  Debug.Log( "Release Notes:" + l.getReleaseNotes() );
}
```

double getProgress() - returns progress for current task (DownloadReleaseNotes, Download or Extract)

```
l.getProgress();
```

double getSize() - returns size of patch

```
l.getSize();
```

void Extract() – method for download patch

```
// because it's async function, you need use
// setOnProgress / setOnDone methods

l.setOnProgress( OnDownloadProgress );
l.setOnDone( OnDownloadDone );
l.Download();

---

void OnDownloadDone()
{
  Debug.Log( "Patch" + l.getAvaliableVersion() + " has been downloaded." );
}

void OnDownloadProgress( double progress )
{
  Debug.Log( "Downloading patch" + l.getAvaliableVersion() + ": " + progress + "% com-
plete." );
}
```

void Finish() – method for apply patch

```
// Finish isn't async function.
l.Finish();

// after patch you should exit game
Application.Quit();
```

string getAvaliableVersion() - returns highest version of available patch

```
l.getAvailableVersion();
```

string getReleaseNotes() - returns release notes

```
l.getReleaseNotes();
```

bool debug – this field is responsible for create crafty_error.txt files when something gone wrong

```
// if there will be error no file will be created
l.debug = false;

// create file on error
l.debug = true;
```

void setOnDone(OnDone handler) - sets OnDone event handler for DownloadRelease-Notes, Download and Extract methods

```
l.setOnDone( handlerForOnDone );

---

void handlerForOnDone()
{
  // do something when current task ( DownloadReleaseNotes, Download or Extract ) finish
}
```

void setOnProgress(OnProgress handler) - sets OnProgress event handler for DownloadReleaseNotes, Download and Extract methods

```
l.setOnProgress( handlerForOnProgress );

---

void handlerForOnProgress( double progress )
{
  // for ex. display progress of current task
}
```

void setOnError(OnError handler) - sets OnError event handler. OnError event is triggered regardless of value of debug field

```
l.setOnError( handlerForOnError );

---

void handlerForOnError( Exception ex )
{
  // handle any errors
}
```

void AddHttpHeader(string name, string value) - add custom header to all requests

```
l.AddHttpHeader( "My-Header-Name", "Value" );
```

void RemoveHttpHeader(string name) - remove custom header

```
l.RemoveHttpHeader( "My-Header-Name" );
```

void CleanHttpHeaders() - remove all headers added via AddHttpHeader

```
l.CleanHttpHeaders();
```

An example can be found in packed/Crafty/LaunchieGUI.cs

5. PatchieAPI

To allow more professional use of Crafty we added PatchieAPI to allow patch creation from code.

void AddVersion(string name) - prepare version of game

```
Patchie.API.AddVersion( "1.0" );
```

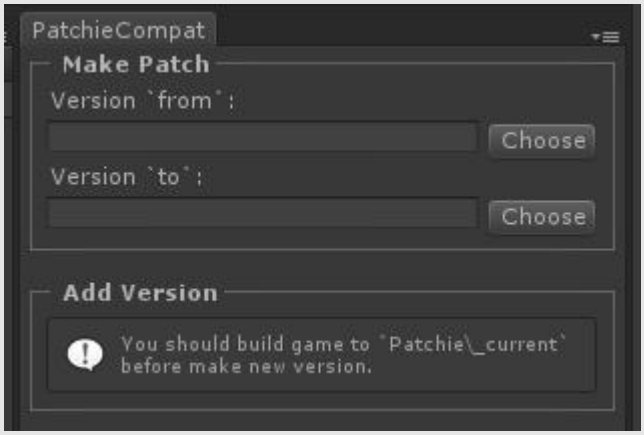
void MakePatch(string from, string to, string description, bool compat, int packetsize [,bool synchronous = true]) - make patch from 2 versions of game

```
//
// from        - older version of game
// to          - newer version of game
// description - release notes for patch
// compat      - if "from" version was builded using Crafty 1.0.5
// packetsize  - used when "compat" is set to false. It specify size of packages of file
//               when comparing two files
// synchronous - if true, it will block script until patch will be finished
//
Patchie.API.MakePatch( "1.0", "1.1", "Release notes for patch 1.1.", false, 512);
Patchie.API.MakePatch( "1.0", "1.1", "Release notes for patch 1.1.", false, 512, false);
```

6. Updating from Crafty 1.0.5 to 1.1

NOTE: **This step is for Crafty 1.0.5 users only**. Allowing them to update their game from Crafty 1.0.5 to Crafty 1.1. If you are a new user please skip this step and proceed in using Crafty 1.1.

If you are a user of an older Crafty version 1.0.5 it is very important to update your project to newer version Crafty 1.1 by simply creating patch using the PatchieCompat that you can find in the Window / Crafty / Patchie Compat.



As you see it's simplify version of Patchie, without the specify packet size option. Simply choose version "from" and " to " create a patch and it's done.

Game updated using this patch will now accept patches created from Crafty 1.1.
From now on you don't have to use PatchieCompat ever again.

7. How to setup your server

This is the easiest part of this tutorial yet still very important as it will give you full control over all your patches and builds made using Crafty.

You should only create *versions.txt* and enter any available patches `[OLD_VERSION][SPACE][NEW_VERSION]`, for ex.:

```
1.0 1.1
1.1 1.2
```

It means that there is 2 patches `1.0 -> 1.1` and `1.1 -> 1.2`.

But if you add `1.0 1.2` at the end:

```
1.0 1.1
1.1 1.2
1.0 1.2
```

it means that there is patch `1.0 -> 1.2` and it's better to download 1 patch, than 2 patches `1.0 -> 1.1 -> 1.2`.

IF YOU WANT USE MERGED PATCHES LIKE ABOVE, REMEMBER TO ADD IT ON TOP OF FILE.

After configuration of *versions.txt* file is done, copy patches from your game directory *Patchie/_output*.

You should have something like this:

```
1.0_1.1
1.1_1.2
1.0_1.2

versions.txt
```

Patches can be hosted on private computer or on web server (it can be without any interpreters like: PHP, CGI, Ruby).

Side note:
Remember that there is only one spacebar between the patch version numbers.

If you target more platforms than one, you need to specify how to download patches for each of them.
This can be done in one of two ways.
First way:
All platforms will use same versions.txt file.
You need to apply prefix or suffix to all versions ex. win_10, lin_10, mac_10, etc.

For example:

```
win_1.0 win_1.1
win_1.0 win_1.2
lin_1.0 lin_1.1
lin_1.0 lin_1.2
mac_1.0 mac_1.1
mac_1.0 mac_1.2
```

Second way:
Each platform will use own versions.txt. To do this, you need to prepare different folder versions, ex.

http://www.example.com/patches/win ,
http://www.example.com/patches/lin ,
http://www.example.com/patches/mac ,

etc. Also there should be versions.txt and patches in folder that belongs to each platforms (just like you would do normally).

You have now mastered the Crafty program. Congratulations! Below you will find useful information on patch changelog, known bugs and of course any information you will need to contact us.

We hope that our product helped you as much as it helped us in creating our game/project. Remember this program like every other was made from developers for developers it means that if you will have any request on adding new features or improvements to Crafty just send us an email at support@penkura.com or post it on Unity thread located (HERE).

8. Changelog.

- 1.2
  + Entire documentation was rewritten.
  + 40 % of the Crafty code was made from scratch to introduce outside patching solution.
  + Added new security measure to make sure patches are installed properly.
  + LaunchieLogic now requires to set a name of the project that will be patch by Crafty.
  + Added Crafty API documentation.
  + Added notes on outside patching system to the documentation.
  + Added more detailed notes in documentation on patch creation process.
  + Patches will be downloaded all at once and installed in the correct order by the outside Crafty installation system.
  + Fixed BUG#0011 Some shaders were not installed properly by Crafty.
  + Added new parts of script in .dll in preparation for new .config system. (which will be available in Crafty 1.2.1).
  + Crafty will now automatically turn off the game and turn it back on after game is properly patched. (showcased in Test Scene two)
  + Added in-game error log option. (showcased in test scene three)
  + Added outside patch installation system with window that indicates installation process.
  + We added a new font for Crafty Linux versions to properly show launcher as Linux don't have standard Ariel font.
  + Added second test scene featuring automatic download process without player interaction.
  + Added third test scene featuring new addition to Crafty (in-game error log preview).

- 1.1.1
  + Fixed BUG#0010 Launchie was unable to work properly on Linux Universal builds.
  + Fixed BUG#0009 (Patchie sometimes added files to patch using absolute paths instead of relative paths)
  + Added button to stop patch creation on the patchy window.
  + Added PatchieAPI to use Patchie programmatically.
  + Added methods to manage headers in requests.
  + Added notification message while patch is being created.
  + Code clean up to speed up patch creation process.
  + Updated documentation with new informations on Multi-platform versions.txt.
  + Added new series of code preparing for outside Installation system (which will be added in Crafty 1.2)

- 1.1
  + Added download speed indicator.
  + Added patch size download indicator.
  + Added two types of download indicator to the script. (Left to download and already downloaded).

+ Added rounding system to the Launcher patch size indicator. (Allowing launcher to round the number in Bytes, KB, MB, GB)
+ Added support for files outside of Standard Unity build.
+ New compression method allow creating much more complex and smaller patches.
+ Added Patchie Compat, allowing users that already release their game using Crafty 1.0.5 to allow their game to Crafty 1.1.
+ Fixing Patch Creation indicator not showing properly patch creation process.
+ Added timer at the end of patch creation message window providing users with patch creation time.
+ Minor script clean up.
+ Added new step to the Crafty documentation "Updating from Crafty 1.0.5 to Crafty 1.1".
+ Updated example files
+ Added full console log at patch creation window allowing user to check patch creation process with all the necessary information.
+ Added percentage indicator at patch creation process.

❖ 1.0.5

+ Added support for Copy and Paste to Version number and release notes.
+ Fixed BUG#0004 (Wait until patch will be done error messages on mac version)
+ Fixed BUG#0005 (Patch could not be created if there are deleted files between builds.)
+ Fixed BUG#0006 (In rare cases Crafty is not able to create second patch version.)
+ Fixed BUG#0007 (In some cases project could present error log with unknown header.)
+ Fixed BUG#0008 (Incorrect matching game versions error appearing after downloading complex patch.)

❖ 1.0.4

+ Added Launchie Prefab.
+ New patch creation algorithm fixing many issues and errors appearing with the old algorithm.
+ Url and Version are now public.
+ Launcher is built on layers allowing much easier adjustment to the project and future updates. (Separated Logic and GUI)
+ New Documentation.
+ Dropped Support for Unity 3.4+ (now minimum Unity version is 3.5.7)
+ Fixed BUG#0003 (Loading bar not showing download process after downloading the patch)

❖ 1.0.3

+ Visual refresh of default Lauchie skin.
+ Fixed BUG#0001 (Crafty crashing on Windows 8).
+ _output folder is now inside the Patchie folder cleaning the build order.

❖ 1.0.1

+ Added Support for Unity 3.4+.

9. Known issues.

- ~~BUG#0011~~
  ~~Some shaders were not installed properly by Crafty.~~

- ~~BUG#0010~~
  ~~Launchie was unable to work properly on Linux Universal builds.~~

- ~~BUG#0009~~
  ~~Patchie sometimes added files to patch using absolute paths instead of relative paths .~~

- ~~BUG#0008~~
  ~~Incorrect matching game versions error appearing after downloading complex patch.~~

- ~~BUG#0007~~
  ~~In some cases project could present error log with unknown header.~~

- ~~BUG#0006~~
  ~~In rare cases Crafty is not able to create second patch version.~~

- ~~BUG#0005~~
  ~~Patch could not be created if there are deleted files between builds.~~

- ~~BUG#0004~~
  ~~Wait Until Patch will be done error massages on mac version.~~

- ~~BUG#0003~~
  ~~Loading bar not showing download process after downloading the patch.~~

- BUG#0002

  Rearranging scenes in Unity3d "Build settings" and/or changing the name of levels/scenes. May cause Crafty to add them to patch even if nothing had been changed.

- ~~BUG#0001~~

  ~~Crafty crashes on Windows 8~~

10. Support and Information

You can find our asset store by clicking the link below.

- Asset Store.

Our official website with update on all of our products and Penkura project can be found in the link below.

- Official Penkura website.

Thread on the Unity Forum about Crafty for Unity can be found in the link below.

- Unity Forum.

You can contact HEXTECH team by clicking the link below. (Penkura Team and HEXTECH team are one and the same from now on).

- Contact Us.

Our Youtube channel with Video Tutorials on Crafty Program can be found below.

- Youtube.