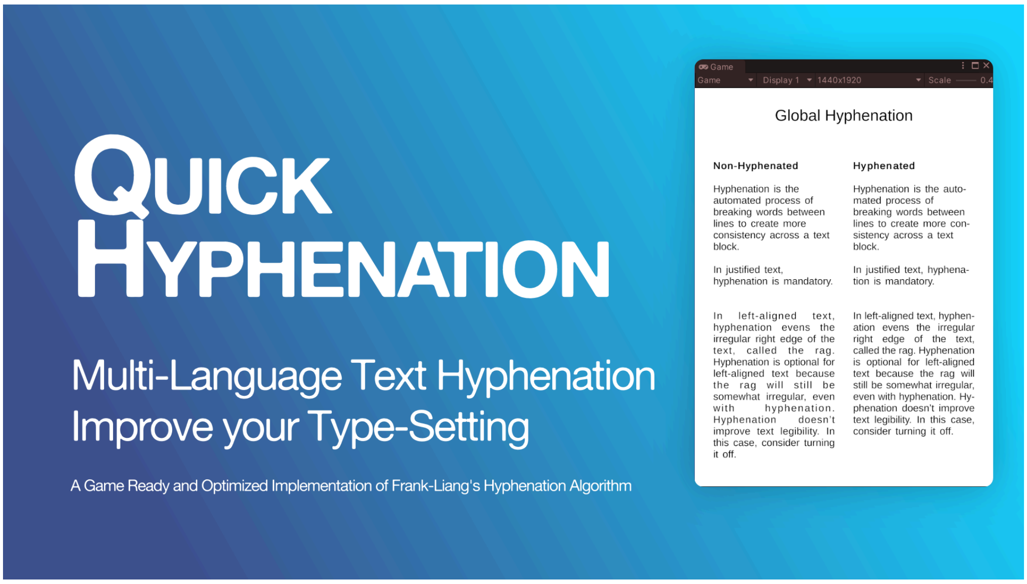
Quick Hyphenation


Quick
Hyphenation
Multi-Language Text Hyphenation
Improve your Type-Setting
A Game Ready and Optimized Implementation of Frank-Liang's Hyphenation Algorithm
Game
Game
Display 1
1440x1920
Scale
0.4

Global Hyphenation

Non-Hyphenated

Hyphenation is the automated process of breaking words between lines to create more consistency across a text block.

In justified text, hyphenation is mandatory.

In left-aligned text, hyphenation evens the irregular right edge of the text, called the rag. Hyphenation is optional for left-aligned text because the rag will still be somewhat irregular, even with hyphenation. Hyphenation doesn't improve text legibility. In this case, consider turning it off.

Hyphenated

Hyphenation is the automated process of breaking words between lines to create more consistency across a text block.

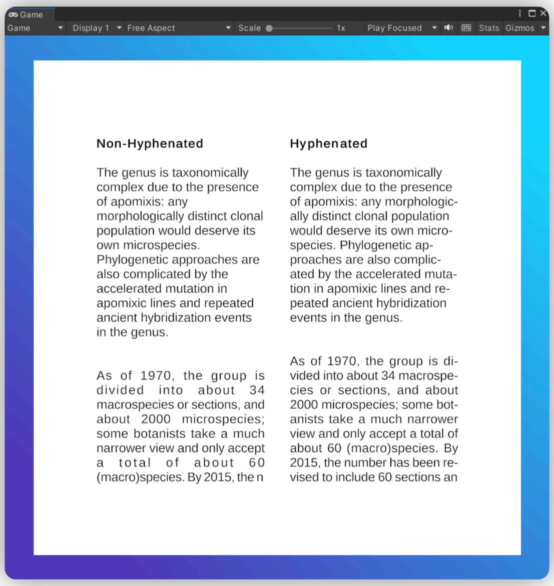
In justified text, hyphenation is mandatory.

In left-aligned text, hyphenation evens the irregular right edge of the text, called the rag. Hyphenation is optional for left-aligned text because the rag will still be somewhat irregular, even with hyphenation. Hyphenation doesn't improve text legibility. In this case, consider turning it off.

Quick Hyphenation allows you to hyphenate your strings and Text Mesh Pro text fields in Unity, allowing for a massively improved typesetting quality and look. Quick Hyphen is based on [Frank Liang's Hyphenation algorithm](), and can use any set of pre-computed patterns.

See https://hyphenation.org/ for a list of available patterns under a permissive open source license, usually MIT or LPPL.

Here is a quick side by side comparison of a text in Unity using hyphenation to create more balanced line breaks:



Non-Hyphenated

The genus is taxonomically complex due to the presence of apomixis: any morphologically distinct clonal population would deserve its own microspecies. Phylogenetic approaches are also complicated by the accelerated mutation in apomixic lines and repeated ancient hybridization events in the genus.

As of 1970, the group is divided into about 34 macrospecies or sections, and about 2000 microspecies; some botanists take a much narrower view and only accept a total of about 60 (macro)species. By 2015, the n

Hyphenated

The genus is taxonomically complex due to the presence of apomixis: any morphologically distinct clonal population would deserve its own microspecies. Phylogenetic approaches are also complicated by the accelerated mutation in apomixic lines and repeated ancient hybridization events in the genus.

As of 1970, the group is divided into about 34 macrospecies or sections, and about 2000 microspecies; some botanists take a much narrower view and only accept a total of about 60 (macro)species. By 2015, the number has been revised to include 60 sections an

Contents

1 Getting Started

1.1 Text Mesh Pro Hyphenation

To get started, simply add the component “Hyphenation Processor” to an object with a Text Mesh Pro Component. The text will now be hyphenated using the global hyphenation settings, and default to US- English hyphenation rules. Text Mesh Pro will use our inserted soft-hyphens to break words where necessary.

1.2 In Script Hyphenation

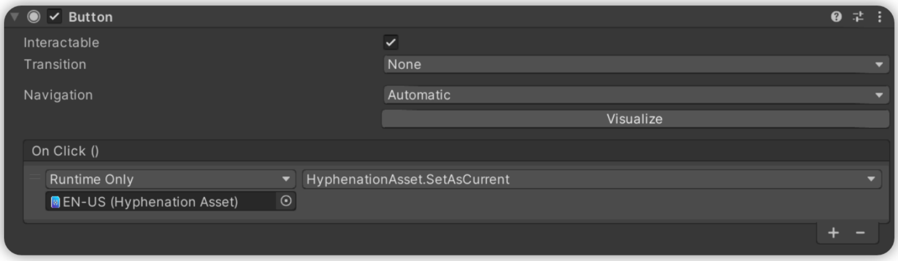
You can also hyphenate any of your strings via C# using the extension functions: `Hyphenate()` and `Hyphenate(hyphen string)`, which will return the hyphenated source text.

Example:

```
string foo = “Hello World”.Hyphenate();
```

1.3 Global Settings

Via Scripting:



☑ Button

Interactable ☑

Transition: None

Navigation: Automatic

On Click ()

Runtime Only — HyphenationAsset.SetAsCurrent

EN-US (Hyphenation Asset)

Simply reference the Hyphenation Asset and call the function “SetAsCurrent” to set this asset as the new global asset.

Via Code:

To Change the global Hyphenation language, simply call

```
Hyphenation.SetupAsync(string languageID);
```

or:

```
Hypenation.Setup(HyphenationAsset asset);
```

languageID corresponds to the name of a Hyphenation Asset to be loaded asnychronously. Ideally you would call this function during the setup stage of your game, before you show the first text.

Here you can also change the global hyphen symbol.

1.4 Local Custom Hyphenation

If you want to support multiple languages in the same scene, you can use the Custom Hyphenation Processor, which allows you to assign a Hyphenation Asset to be used, and a custom hyphenation string to be inserted, should you want to.

1.5 Hyphenation Assets

Hyphenation assets are located in the Hyphenation Assets folder. Each Asset corresponds to a language, allowing for varieties of languages and hyphenation rules, such as UK-English and US-English. Each of these assets is based on open source patterns created by amazing people from all over the world. Along with settings you will also find the necessary **license and copyright information**, usually MIT or LPPL.

I have included Hyphenation Assets for the languages:

Language	License
Danish	MIT
Dutch	LPPL 1.0
English UK	MIT
English US	Custom*
French	MIT
German	MIT
Italian	MIT
Polish	MIT
Portuguese	BSD 3
Russian	LPPL 1.2
Spanish	MIT
Ukrainian	LPPL 1.2
Welsh	LPPL 1.0

* ”Copyright (C) 1990, 2004, 2005 Gerard D.C. Kuiken
Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.”

See https://hyphenation.org/ for a list of all open source patterns and available languages.

1.6 Adding Languages

To add hyphenation patterns for a language, check out hyphenation.org for free and open source hyphenation patterns.

Right click and create a Hyphenation Asset in your Hyphenation Assets folders.

Make sure to diligently review and port the license, author and copyright information of any pattern asset you might select.

After having inserted the author and copyright notes from the file, you can go ahead and import it via ULR, or by pasting its text and pressing the corresponding button in the patterns tab, and Quick Hyphen will automatically import the patterns and exceptions for you.

1.7 Adding Exceptions

You can add and remove exceptions in the hyphenation asset using the provided text field and buttons. This allows you to handle edge cases, or exclude words from being hyphenated altogether. Simply insert the word with a - (minus hyphen)at the spots it should hyphenate in the input field, and it will be handled accordingly.

By adding a "." at the start or end of the word, you ensure that the exception only applies if the word starts with it, or ends with it.

Example:

Cheesecake

Exception = Che-ese

Cheesecake => Che-esecake

Exception = .Che-ese.

Cheesecake => Cheesecake

2 How It Works

This plugin implements Frank Liang's hyphenation algorithm with considerations for game engines, focussing on minimal garbage generation and fast hyphenation times.

It's based on a set of patterns in the following format

.pat5 = .0p0a0t5, which split apart means .pat = {0,0,0,5}

Wherein "." marks a word boundary.

For each part of a word, each set of available patterns needs to be checked, and can then be applied to the word's result.

Word

.w .wo .wor	w wo wor	o or ord	r rd rd.	d d.	.

.word .word.	word word.	ord.			

If a number entry of a pattern has a higher result than the currently stored value at that position, it must be replaced by the new value of the pattern. At the end, uneven values determine the points where the hyphen can be inserted.

I have optimized the lookup formula by using a node tree, to eliminate unused paths early on, and avoided garbage generation wherever possible.

2.1 Garbage Generation

Currently, the only garbage being repeatedly generated when calling the function is the returned string. That does not mean Quick Hyphenation uses no memory.

If the inserted string length exceeds the starting allotted 1024 characters in length, internally arrays and string builders will resize themselves once, to make more space. Resizing again at 2048 characters and so on. This leads to a new array instantiation, and brief garbage creation. If a string has caused the arrays to grow to over 1024 * 1024 * 4 characters in length, they will be reset to 1024 at the end of the hyphenation. At this point, I assume hyphenating that amount of text will not be done every frame.

2.2 Memory Allocation

Each language requires a different number of patterns to function, some over 30,000, some under 200. This dictates the size of the patterns' collection in memory, on the disk, and the complexity of the search tree.

Currently, the largest file is German, which takes up about 1,460 KB on the disk, and will likely take up around 5,000 KB in Memory.

2.3 Text Mesh Pro Integration

Text Mesh Pro uses soft-hyphens to break up words where it could benefit the layout of the text. This massively improves the look of your text fields. Quick Hyphenation works nicely with Text Mesh Pro by placing these soft-hyphens whenever Text Mesh Pro notices a change in text. For this, it uses the ITextPreprocessor interface, and assigns itself as the text preprocessor of the given TMP_Text. This may create problems with other plugins that use the same “slot”. Let me know should you run into any difficulties, and we can figure out a solution together.

3 Supported Platforms

No platform specific code is used in this plugin, it should run without problems on every platform.

Note: Some Platforms may not use `Resources.Load()` or `Resources.LoadAsync()`. In these cases, load the Hyphenation Asset of your choice manually, and call the function `Hyphenation.Setup(HyphenationAsset asset)`

4 FAQ

Q: My text seems to break correctly but the hyphen itself is not visible.

A: Sometimes Text Mesh Pro auto-sizing combined with bold text can in *very rare cases* misplaced the hyphen “-”. Sadly this happens on TMP’s side and there is not really much we can do to fix it. Try turning off auto-sizing.

5 Contact and Support

If you encounter bugs, have feature requests or questions, feel free to reach out at support@neoludic.games.

Best wishes, David