Oh What A Tangled Web We Weave: Interactive Storytelling Using Twine

Ruben R. PuenteDura, Ph.D.
Twine

Create your own interactive stories with Twine, the same tool used to produce the stories on this Web site.

Think Visually

Twine lets you organize your story graphically with a map that you can rearrange as you work. Links automatically appear on the map as you add them to your passages, and passages with broken links are apparent at a glance. As you write, focus on your text with a full-screen editing mode like Dark Room. Rapidly switch between a published version of your story and the editable one as you work.

Free As In Free

Stories you create with Twine can be used however you'd like. Because the final output is a single, small Web page, you can easily email a story to friends, post it on your Web site, or even distribute it on a CD-ROM. (You could use a floppy disk just as easily — stories take up that little space.) You can also use your stories for commercial purposes without restriction.

Twine is free to download and use, and you can share it with anyone you like. You can even modify the Twine, provided you release your own version under the GNU Public License.

Propellerheads Welcome

If you like working with a command prompt, you can also use teve, the command-line version of Twine. It's easy to incorporate into the toolchain you know and love, and syntax modes are available for some common text editors.

Seeing Is Believing

The screen casts below show how easy it is to create a story in Twine.

Download

- Twine 1.4.8 for Windows (74mb) graphical interface
- Twine 1.4.8 for OS X (28.4mb) graphical interface requires OS X 10.6
- twine 1.4.8 (370mb) command-line interface requires Python

Documentation

- Twine Reference
  A guide to writing stories with Twine and more.
- Twine Google Group
  Get help and share your work.
- Twine API
  Technical information on how the story engine works.

Syntax Modes

If you use a plain text editor to write, these will enhance your text so markup stands out.
Watson's First Case: The Game Is Afoot

After many years abroad, you return to London. You are looking for someone to share a room with, and are directed to the lab at the local hospital, to speak with a gentleman by the name of Sherlock Holmes.

[[The Hospital Lab]]
Sherlock Holmes stands here. He fixes you in his gaze, saying "Dr. Watson, I presume? You come at the right time. As a medical man, I could make use of your knowledge. A man was found unconscious yesterday in Frying Pan Alley. Some locals think he worked at a cement kiln, others at the local brewery. I've just mixed some residue from his clothes with phenolphthalein, and it turned pink. Dr. Watson, do you realize what this means?" You reply:

[[He's a cement worker]]
[[He's a brewer]]
Puzzles and Endings

"Excellent, Dr. Watson! I look forward to continuing our conversation at Baker Street."

"Dr. Watson, whatever your other merits as a doctor might be, you have clearly become rusty in your chemical knowledge. I do not think it would profit us to continue our conversation."

Sherlock Holmes turns his back to you, and you slowly walk away, wondering at what might have been.
Using actions and display

Baker Street

Title: Baker Street
Tags (separate with spaces)

The rooms you hope to share with Sherlock Holmes in Baker Street. The room is cluttered and somewhat shabby, but cozy nonetheless. An old Persian slipper lies on a mantelpiece.

<<actions "Ask about the slipper" "Ask about the rent">>

Ask about the slipper

Title: Ask about the slipper
Tags (separate with spaces)

"That's where I keep my tobacco", Holmes replies.

<<display "Baker Street">>
Setting a Bookmark and Creating Variables

I think five guineas should cover it nicely – what do you think?

[[Accept the offer]]
[[Ask for a lower price]]
<<set $poor = false>>
<<set $wallet = 100>>
Using Variables and *if/else/endif* Statements

```
Well... I suppose I could take just 3 guineas..." Holmes replies dubiously.

[[Accept the offer]]
<<set $poor = true>>
<<set $wallet = 10>>

"Splendid! I'll let Mrs.Hudson know", Holmes replies.
<<if $poor>>
Holmes is clearly sympathetic to your current financial woes, and offers to buy dinner. You gratefully accept.
<<else>>
Grateful to have a place to stay, you offer to buy dinner, and Holmes happily accepts.
<<endif>>
[[After Dinner]]
```
Displaying Variables

Holmes sits back in a worn yet comfortable armchair, and sighs. He then turns to you and says:

```
<<if $poor>>
[ [You must be exhausted – we can talk more in the morning.]]
<<else>>
[ [How about a few rounds of backgammon?]]
<<endif>>
```

You agree with Holmes, and go to bed. Before turning in, you check your wallet: you have `<<print $wallet>>` guineas left for the rest of the month – clearly, you will need to look for a job soon.
Using Variables in Calculations and *silently* to Eliminate Whitespace

Title
How about a few rounds of backgammon?

Tags (separate with spaces)

```twine
<<silently>>
<<set $lostmoney = Math.round(Math.random() * 8)+2>>
<<set $wallet = $wallet - $lostmoney>>
<<endsilently>> You agree, and a few hours later your wallet is <<print $lostmoney>> guineas lighter, leaving you with <<print $wallet>> guineas for the rest of the month. You retire to bed, impressed by Holmes' gaming prowess.
```
Some Text Formatting Options

<table>
<thead>
<tr>
<th>Formatting</th>
<th>Source Code</th>
<th>Appears As</th>
</tr>
</thead>
<tbody>
<tr>
<td>Italics</td>
<td>//text//</td>
<td>text</td>
</tr>
<tr>
<td>Boldface</td>
<td>&quot;&quot;text&quot;&quot;</td>
<td>text</td>
</tr>
<tr>
<td>Underline</td>
<td><strong>text</strong></td>
<td>text</td>
</tr>
<tr>
<td>Subscript</td>
<td>H<del>2</del>O</td>
<td>H₂O</td>
</tr>
<tr>
<td>Superscript</td>
<td>meters/second^^2^^</td>
<td>meters/second²</td>
</tr>
<tr>
<td>Monospace</td>
<td>{{{text}}}}</td>
<td>text</td>
</tr>
<tr>
<td>Horizontal line</td>
<td>----</td>
<td></td>
</tr>
</tbody>
</table>
Some Math Operators

• These operators are used “as-is” in expressions:
  • Logical Operators: eq, neq, gt, gte, lt, lte, and, or, not
  • Basic Operators: +, -, *, /, %

• These operators need to be prefixed with `Math`. and followed by an argument in parenthesis (`pow` takes two arguments, `max` and `min` take one or more arguments, and `random` takes no arguments):
  • Absolute Value, Rounding, Range, and Random Numbers: abs; round, floor, ceil; max, min; random
  • Powers and Logarithms: sqrt, pow, exp; log
  • Trigonometric Operators: sin, cos, tan, asin, acos, atan

• These constants need to be prefixed with `Math.`:  
  • Constants: PI, E, SQRT2, SQRT1_2, LN2, LN10, LOG2E, LOG10E

For more information and examples see: Mozilla Developer Network – JavaScript Reference – Math
Hippasus

Blog: http://hippasus.com/rrpweblog/
Email: rubenrp@hippasus.com
Twitter: @rubenrp

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License.