Computational Thinking: A Digital Storytelling Perspective

Ruben R. Puentedura, Ph.D.

Dimensions of Computational Thinking

Computational Concepts	Computational Practices	Computational Perspectives
Sequences	Being Incremental and Iterative	Expressing
Loops	Testing and Debugging	Connecting
Events	Reusing and Remixing	Questioning
Parallelism	Abstracting and Modularizing	
Conditionals		
Operators		
Data		

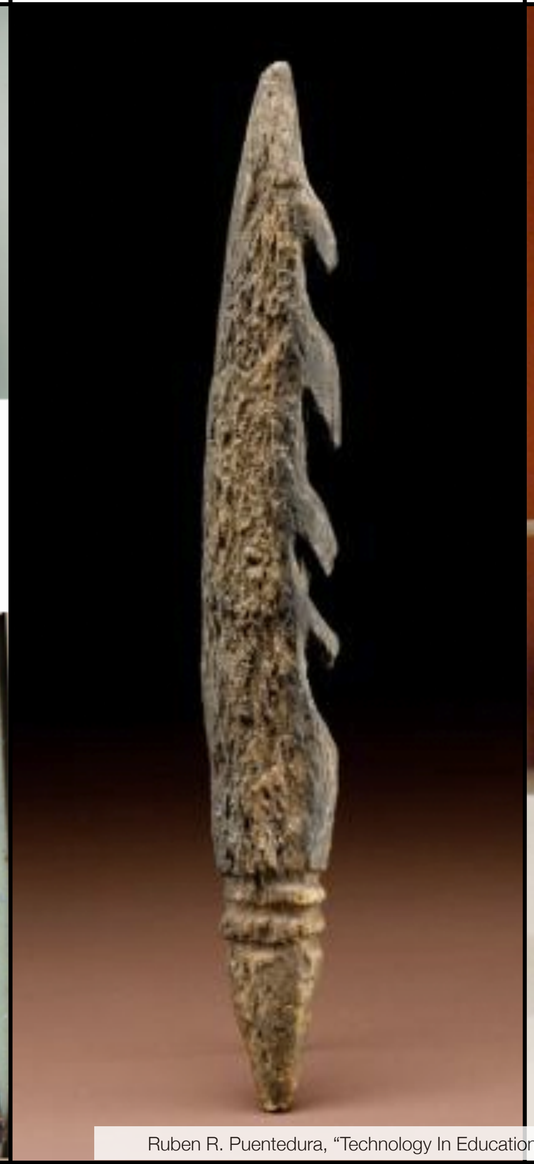
Brennan, Karen, and Mitchel Resnick. "New frameworks for studying and assessing the development of computational thinking." *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*. 2012.

Computational Thinking in Math and Science

Data Practices	Modeling & Simulation Practices	Computational Problem Solving Practices	System Thinking Practices
Collecting Data	Using Computational Models to Understand a Concept	Preparing Problems for Computational Solutions	Investigating a Complex System as a Whole
Creating Data	Using Computational Models to Find and Test Solutions	Programming	Understanding the Relationships within a System
Manipulating Data	Assessing Computational Models	Choosing Effective Computational Tools	Thinking in Levels
Analyzing Data	Designing Computational Models	Assessing Different Approaches/ Solutions to a Problem	Communicating Information about a System
Visualizing Data	Constructing Computational Models	Developing Modular Computational Solutions	Defining Systems and Managing Complexity
		Creating Computational Abstractions	
		Troubleshooting and Debugging	

Weintrop, David, et al. "Defining computational thinking for mathematics and science classrooms." *Journal of Science Education and Technology* 25.1 (2016): 127-147.

Social	Mobility	Visualization	Storytelling	Gaming
200,000 years	70,000 years	40,000 years	17,000 years	8,000 years



Ruben R. Puentedura, “Technology In Education: The First 200,000 Years” *The NMC Perspective Series: Ideas that Matter*. NMC Summer Conference, 2012.

Storytelling

Narrative sources; Narrative constraints

Pictorial vocabulary; Narrative transitions; Text/image integration

CDS Seven Elements; Montage structures

Narrative structures; Narrative flows

Ludic elements

Image Assembly → Sequential Art → Moving Image → Interactive Media → Interactive Fiction

TIME

PLACE

Infinite Canvas

CONNECTION

SOCIAL

Formal Definition of **Game** (Salen & Zimmerman)

"A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome."

Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2003.

Games and Fun

Successful Games	
Include These Items…	**…To Avoid**
Preparation before challenges	Results due to pure chance
A sense of a game space	The perception of the game as trivial
A solid core mechanic	The game not being perceived as a game
A range of challenges	The game being exhausted too quickly
A range of required abilities	The game being perceived as simplistic
Skill in using the required abilities	The game being perceived as tedious
Also Have…	**…Because**
Variable feedback	Players like to see greater skill result in greater rewards
Ways to accommodate beginners & experts	Beginners need not get clobbered, or experts “bottom feed”
A definite cost for failure	Players feel cheated by “never-lose” games

In Unsuccessful Games	
When Players Say…	**…They Mean**
The game is too easy	Game patterns are too simple
The game is too involved	They are uninterested in the info required to detect patterns
The game is too hard	Patterns are perceived as noise
The game becomes too repetitive	New patterns are added too slowly
The game becomes too hard	New patterns are added too fast
The game runs out of options	All game patterns are exhausted

Koster, R. *Theory of Fun for Game Design*. Paraglyph. (2004)

Twine – http://twinery.org

Twine is an open-source tool for telling interactive, nonlinear stories.

You don't need to write any code to create a simple story with Twine, but you can extend your stories with variables, conditional logic, images, CSS, and JavaScript when you're ready.

Twine publishes directly to HTML, so you can post your work nearly anywhere. Anything you create with it is completely free to use any way you like, including for commercial purposes.

Twine was originally created by **Chris Klimas** in 2009 and is now maintained by a whole bunch of people at **several different repositories**.

Download 2.1.3
For **Windows (32-bit)**, **OS X**, and **Linux (32-bit)**

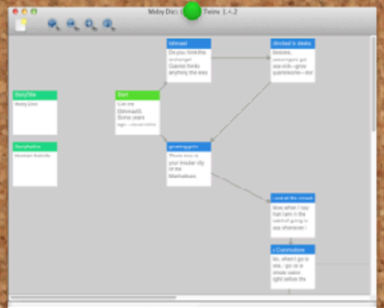
Use it online

Version 1.4.2 for **Windows** and **OS X** is also available.

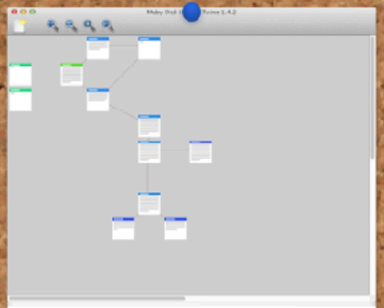
Do you love Twine?
Help support its development!

Wiki
tutorials, documentation

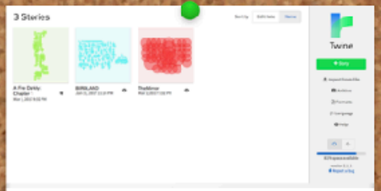
Forum
get help, share your work



Editing a story in Twine 1.4.



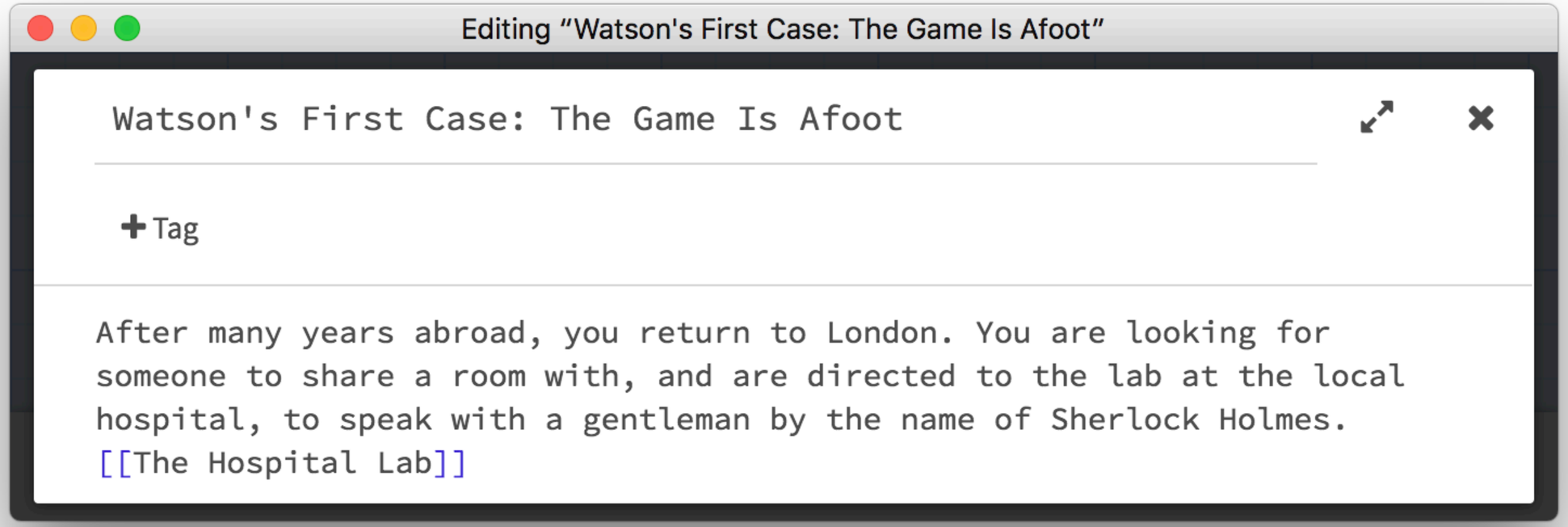
A bird's-eye view of a story map in Twine 1.4.



The story list in Twine 2.0.

Editing a story in Twine 2.0.

Setup


Editing “Watson's First Case: The Game Is Afoot”
Watson's First Case: The Game Is Afoot
+ Tag
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]]

Branching and inserting media

Editing “The Hospital Lab”

The Hospital Lab

+ Tag

```
<img
src="http://upload.wikimedia.org/wikipedia/commons/c/cd/Sherlock_Holmes_P
ortrait_Paget.jpg">
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

Editing "He's a cement worker"

He's a cement worker

+ Tag

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

Editing "He's a brewer"

He's a brewer

+ Tag

"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.

Choices: Take 1

Editing "Baker Street"

Baker Street

+ Tag

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.
[[Ask about the slipper]]
[[Ask about the rent]]

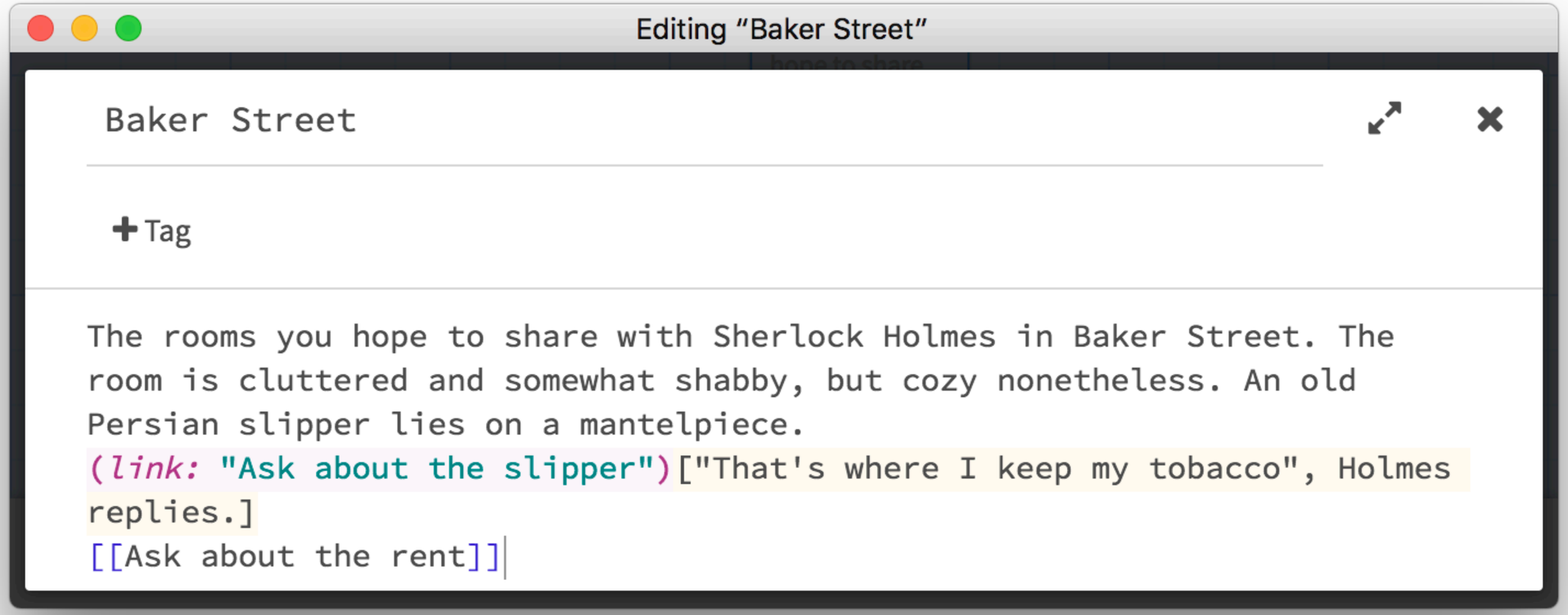
Editing "Ask about the slipper"

Ask about the slipper

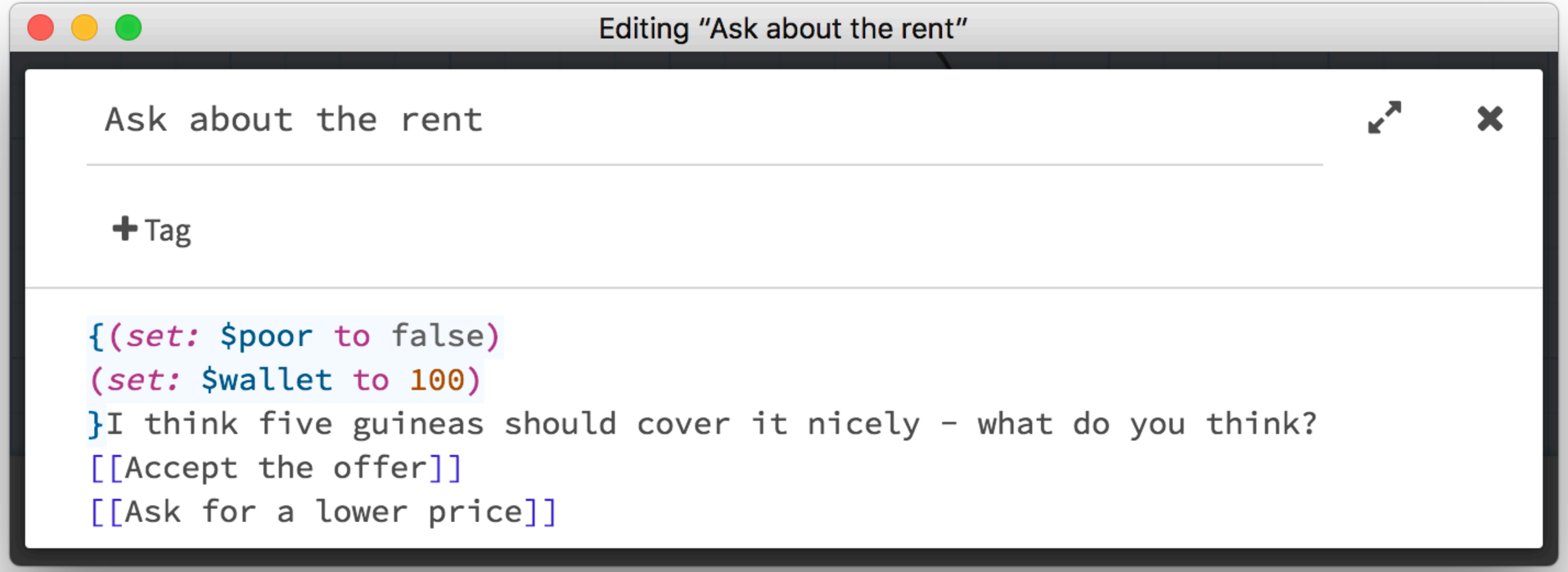
+ Tag

"That's where I keep my tobacco", Holmes replies.
[[Baker Street]]

Choices: Using a *macro* and a *hook*



Creating variables and collapsing whitespace



Using variables and *if/else* macros

Editing "Ask for a lower price"

Ask for a lower price

+ Tag

```
"Well... I suppose I could take just 3 guineas..." Holmes replies
dubiously.
[[Accept the offer]]
(set: $poor to true)
(set: $wallet to 10)
```

Editing "Accept the offer"

Accept the offer

+ Tag

```
"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.]}
[[After Dinner]]
```

Including links within hooks and displaying variables

Editing "After Dinner"

After Dinner

+ Tag

```
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?]] ]}
```

Editing "You must be exhausted - we can talk more in the morning."

You must be exhausted - we can talk more in the morning.

+ Tag

```
You agree with Holmes, and go to bed. Before turning in, you check your
wallet: you have $wallet guineas left for the rest of the month -
clearly, you will need to look for a job soon.
```

Using variables in calculations

Editing "How about a few rounds of backgammon?"

How about a few rounds of backgammon?

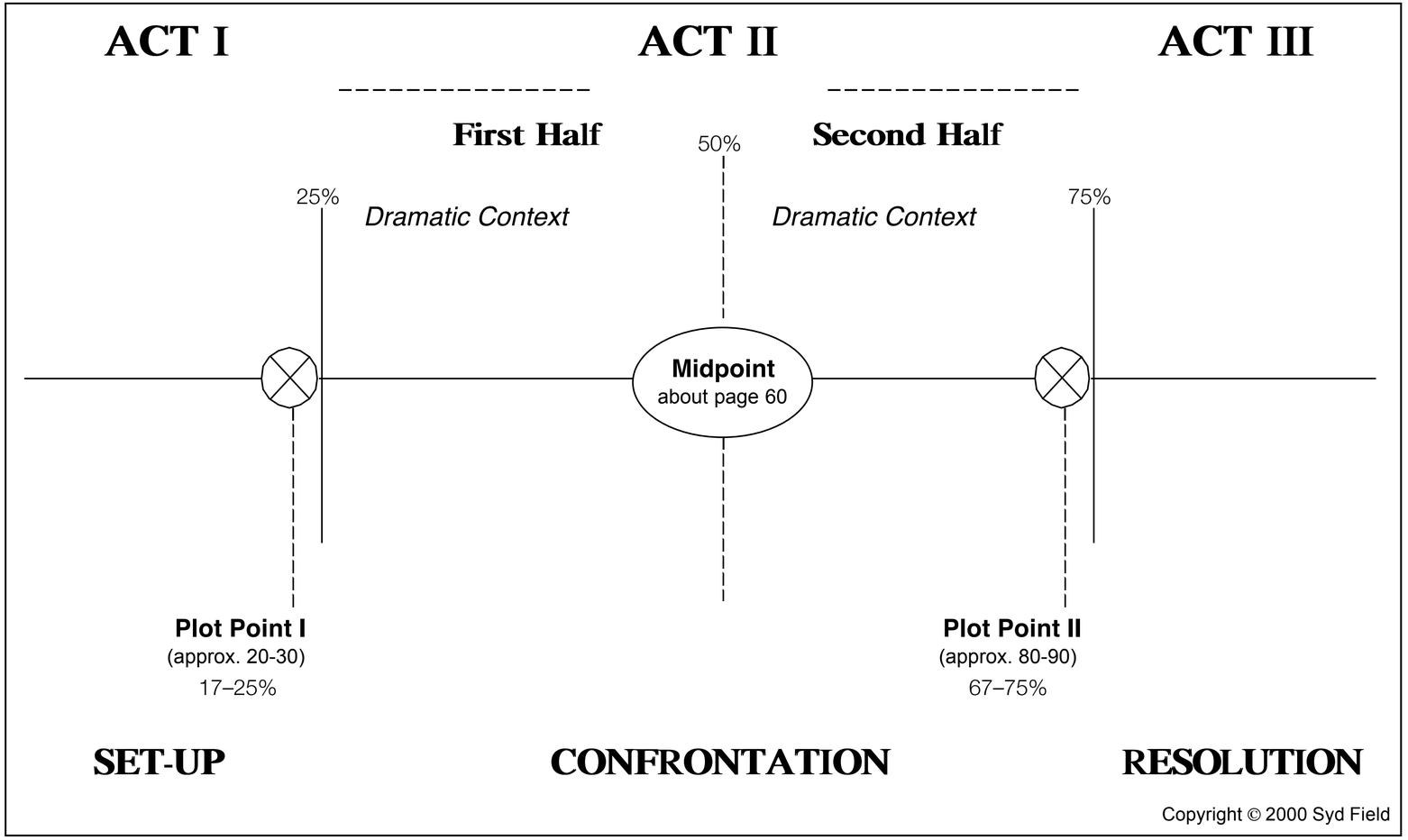
+ Tag

```
{(set: $lostmoney to (random: 0,8)+2)
(set: $wallet to $wallet - $lostmoney)
}You agree, and a few hours later your wallet is $lostmoney guineas
lighter, leaving you with $wallet guineas for the rest of the month. You
retire to bed, impressed by Holmes' gaming prowess.
```

Some Text Formatting Options

Formatting	Source Code	Appears As
Italics	//text//	*text*
Boldface	''text''	**text**
Superscript	meters/second^^2^^	meters/second^2
Horizontal line	---	_____

Basic Screenplay Design



S. Field, *Screenplay: The Foundations of Screenwriting, Third Edition*. Dell Publishing. (1994)

Joseph Campbell: The Hero's Journey

- Three parts to the journey:
  - **The Departure**: the hero is called to adventure
    - Someone is in need of aid, and the hero is called upon to help
  - **The Initiation**: the hero undertakes a journey (physical or spiritual) to reach the goal that will secure the needed aid
    - The hero undergoes a process of change
  - **The Return**: the hero accomplishes their task, and aid is rendered
    - The hero receives some reward

Campbell, J. *The Hero with a Thousand Faces*. Princeton Univ. Press. (1972)

The Detailed Journey

- **The Departure**:
  - The Call to Adventure
  - Refusal of the Call
  - Supernatural Aid
  - The Crossing of the First Threshold
  - Belly of The Whale
- **The Initiation**:
  - The Road of Trials
  - The Meeting with the Giver of Life
  - Encounter with Temptation
  - Atonement with the Giver of Laws
  - Apotheosis
  - The Ultimate Boon
- **The Return**:
  - Refusal of the Return
  - The Magic Flight
  - Rescue from Without
  - The Crossing of the Return Threshold
  - Master of the Two Worlds
  - Freedom to Live

Vladimir Propp: Character Roles

- **Main Characters:**
  - Protagonist (Hero)
  - Antagonist (Villain)
  - Dispatcher
  - Donor
  - Helper
  - Person Sought-For
  - False Protagonist (False Hero)
- **Supporting Characters:**
  - Family Members
  - Connectors

Propp,V. *Morphology of the Folktale*. Univ. of Texas Press. (1968)

Character Functions

Introduction		
#	*Function*	*Example*
1	Absentation	A member of the family absents him/herself.
2	Interdiction	An interdiction is given to the hero.
3	Violation	The interdiction is violated.
4	Reconnaissance	A villain makes an attempt to get information.
5	Delivery	The villain gets information about the victim.
6	Trickery	The villain tries to deceive the victim.
7	Complicity	The victim is deceived.

The Body of the Story		
#	*Function*	*Example*
8	Villainy	The villain causes harm to a family member. - *OR*
8a	Lack	A family member lacks or desires something.
9	Mediation	A misfortune is made known, the hero is dispatched.
10	Begin Counteraction	The hero (seeker) agrees to counteraction.
11	Departure	The hero leaves home.

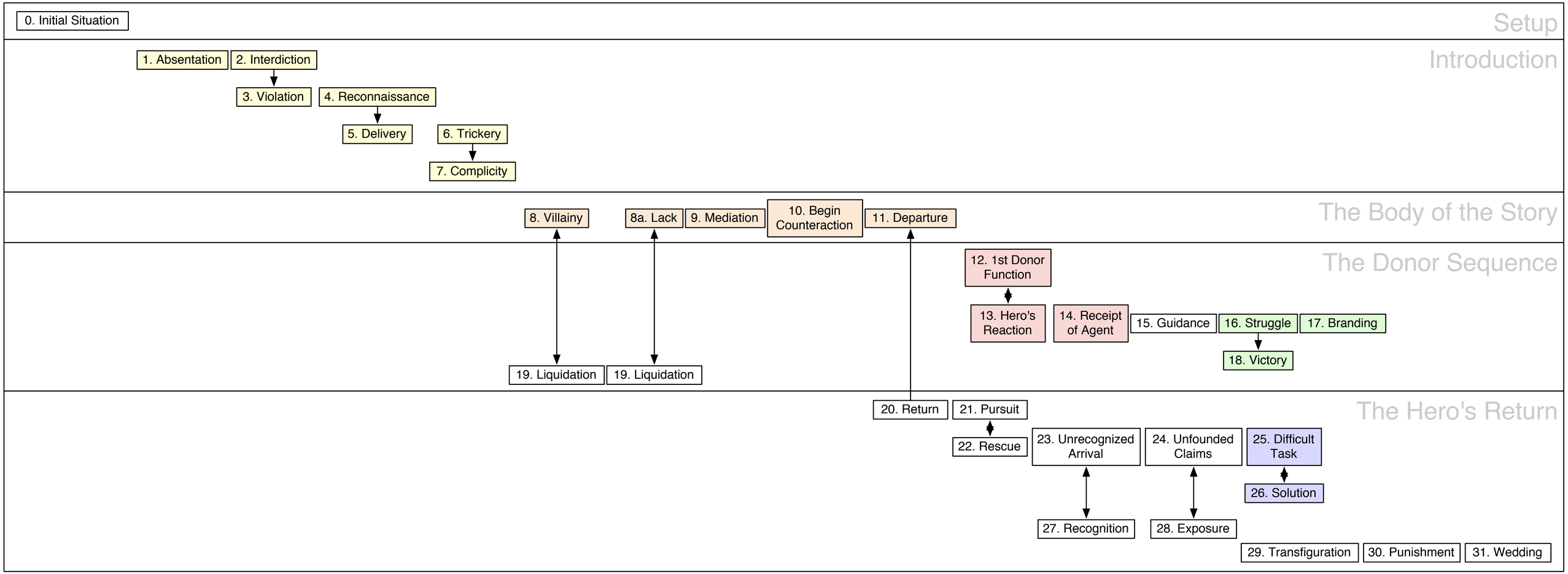
Notes:
- 12–14 can also occur as a block prior to the 8–11 block;
- 23–24 and 25-26 can also occur prior to 19;
- 17 can occur between 25 and 26.
- Moves can end on functions other than 31 (e.g., 14, 19, 20, 22).

The Donor Sequence		
#	*Function*	*Example*
12	1st Donor Function	The hero is tested by a donor of a magical agent.
13	Hero's Reaction	The hero reacts to the agent or donor.
14	Receipt of Agent	The hero acquires the use of the magical agent.
15	Guidance	The hero is led to the object of search.
16	Struggle	The hero and villain join in combat.
17	Branding	The hero is branded.
18	Victory	The hero defeats the villain.
19	Liquidation	The initial misfortune or lack is liquidated.

The Hero's Return		
#	*Function*	*Example*
20	Return	The hero returns.
21	Pursuit	The hero is pursued.
22	Rescue	The hero is rescued from pursuit.
23	Unrecognized Arrival	The hero, unrecognized, arrives home or elsewhere.
24	Unfounded Claims	A false hero presents unfounded claims.
25	Difficult Task	A difficult task is proposed to the hero.
26	Solution	The task is resolved.
27	Recognition	The hero is recognized.
28	Exposure	The false hero or villain is exposed.
29	Transfiguration	The hero is given a new appearance.
30	Punishment	The villain is punished.
31	Wedding	The hero is married and ascends the throne.

Moves and Other Elements

- A **move** is defined as any development from *Villainy* or *Lack* until a terminal function (which may be *Wedding*, or any allowable prior function).
- Relationships between moves:
  - Moves can follow each other sequentially;
  - One move can be embedded within another (e.g., the first part of move 1 is followed by move 2, which is then followed by the second part of move 1);
  - One move can split into two separate moves, which are then resolved sequentially;
  - Two moves can have a common ending.
- Other elements:
  - Connectives
  - Motivations
  - Branching items
  - Function results: positive, negative, negative with punishment
  - Treblings


Setup
0. Initial Situation
Introduction
1. Absentation
2. Interdiction
3. Violation
4. Reconnaissance
5. Delivery
6. Trickery
7. Complicity
The Body of the Story
8. Villainy
8a. Lack
9. Mediation
10. Begin Counteraction
11. Departure
The Donor Sequence
12. 1st Donor Function
13. Hero's Reaction
14. Receipt of Agent
15. Guidance
16. Struggle
17. Branding
18. Victory
19. Liquidation
19. Liquidation
The Hero's Return
20. Return
21. Pursuit
22. Rescue
23. Unrecognized Arrival
24. Unfounded Claims
25. Difficult Task
26. Solution
27. Recognition
28. Exposure
29. Transfiguration
30. Punishment
31. Wedding

"Authentic" Russian Folktale, Outline Generator v1.0
(AKA – Proppian, Folktale Outline Generator v1.0)

0: α (alpha) — initial situation

Select functions: (function 8 and/or function 8a are required)

- [ ] 1: β (beta) — absentation

- [ ] 2: γ (gamma) — interdiction w/3
- [ ] 3: δ (delta) — violation w/[2]

- [ ] 4: ε (epsilon) — reconnaissance w/5
- [ ] 5: ζ (zeta) — delivery w/[4]

- [ ] 6: η (eta) — trickery w/6
- [ ] 7: θ / λ (theta /lamda) — complicity w/[6]

- [ ] 8: A — villainy w/[8a] 19
  and / or
- [ ] 8a: a — lack w/[8] 19
- [ ] 9: B — mediation, connective incident
- [ ] 10: C — beginning counteraction
- [ ] 11: ↑ — departure w/[20]

- ( ) DEF after ABC↑ only
- ( ) DEF before ABC↑ only
- ( ) DEF before and after ABC↑
- [ ] 12: D — first function of donor w/13
- [ ] 13: E — protagonist's reaction w/12
- [ ] 14: F — acquisition of magical agent

- [ ] 15: G — transference, guidance

- [ ] 16: H — struggle w/18
- [ ] 17: J — branding
- [ ] 18: I — victory w/[16] [17]

- [ ] 19: K — liquidation w/8
- [ ] 19: K — liquidation w/8a

- [ ] 20: ↓ — return w/11

- [ ] 21: Pr — pursuit w/22
- [ ] 22: Rs — rescue w/21

- [ ] 23: o — unrecognized arrival w/27
- [ ] 24: L — unfounded claims w/28

- [ ] 25: M — difficult task w/26
- [ ] 26: N — solution w/25

- [ ] 27: Q — recognition w/23
- [ ] 28: Ex — exposure w/24

- [ ] 29: T — transfiguration
- [ ] 30: U — punishment
- [ ] 31: W — wedding

Help

instructions
explanation
how it works

functions
- all
- paired
- grouped

characters
moves
other elements

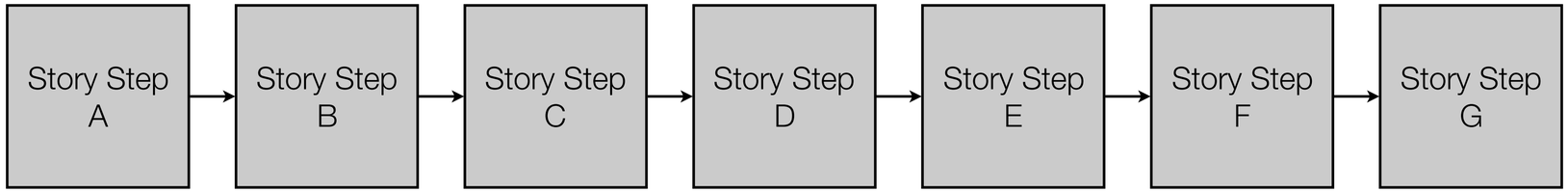
generate

clear

select all

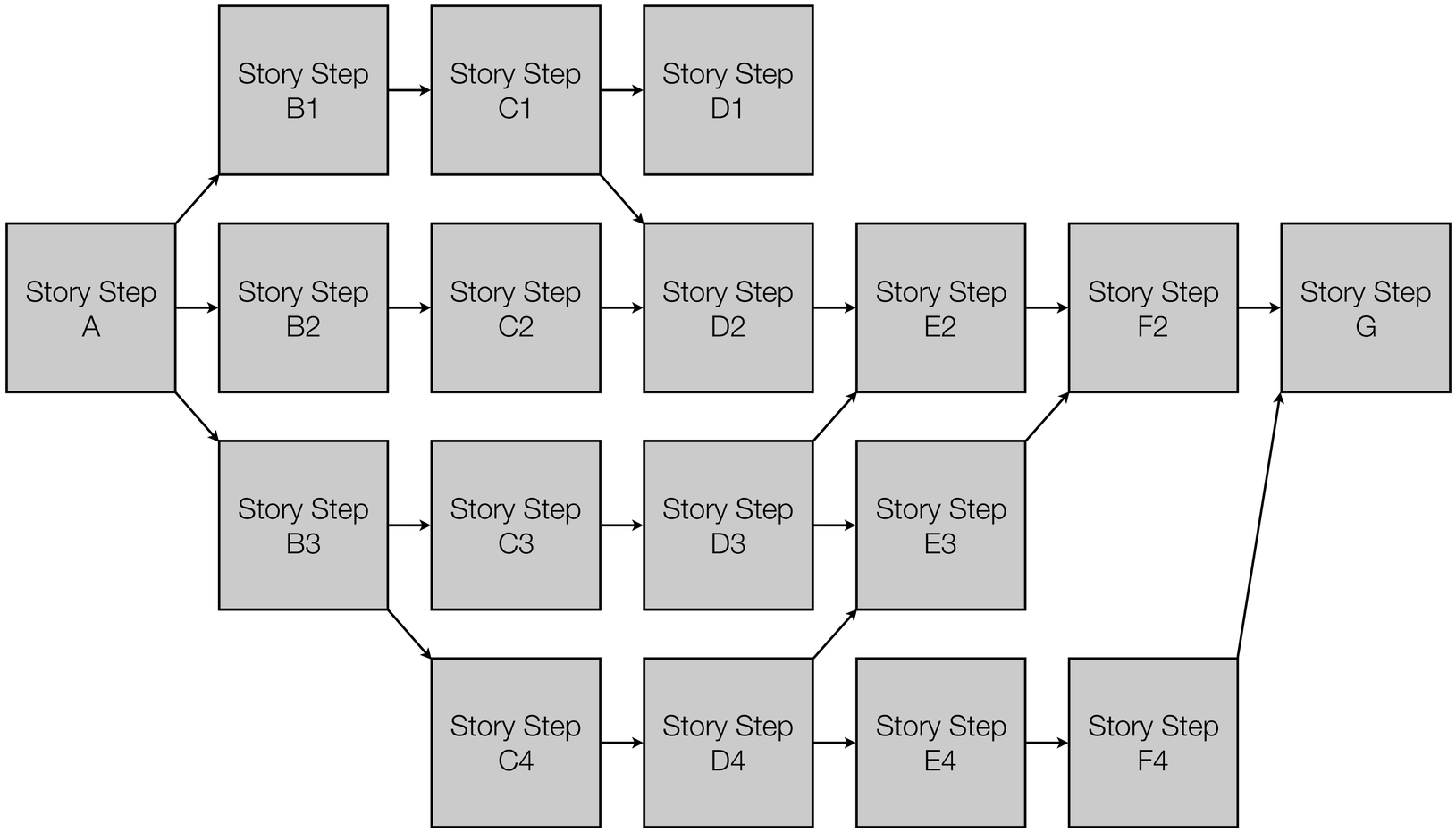
Proppian Folktale Outline Generator v1.0. Online at: http://www.stonedragonpress.com/vladimir_propp/propp_generator_v1.htm

Linear Storytelling

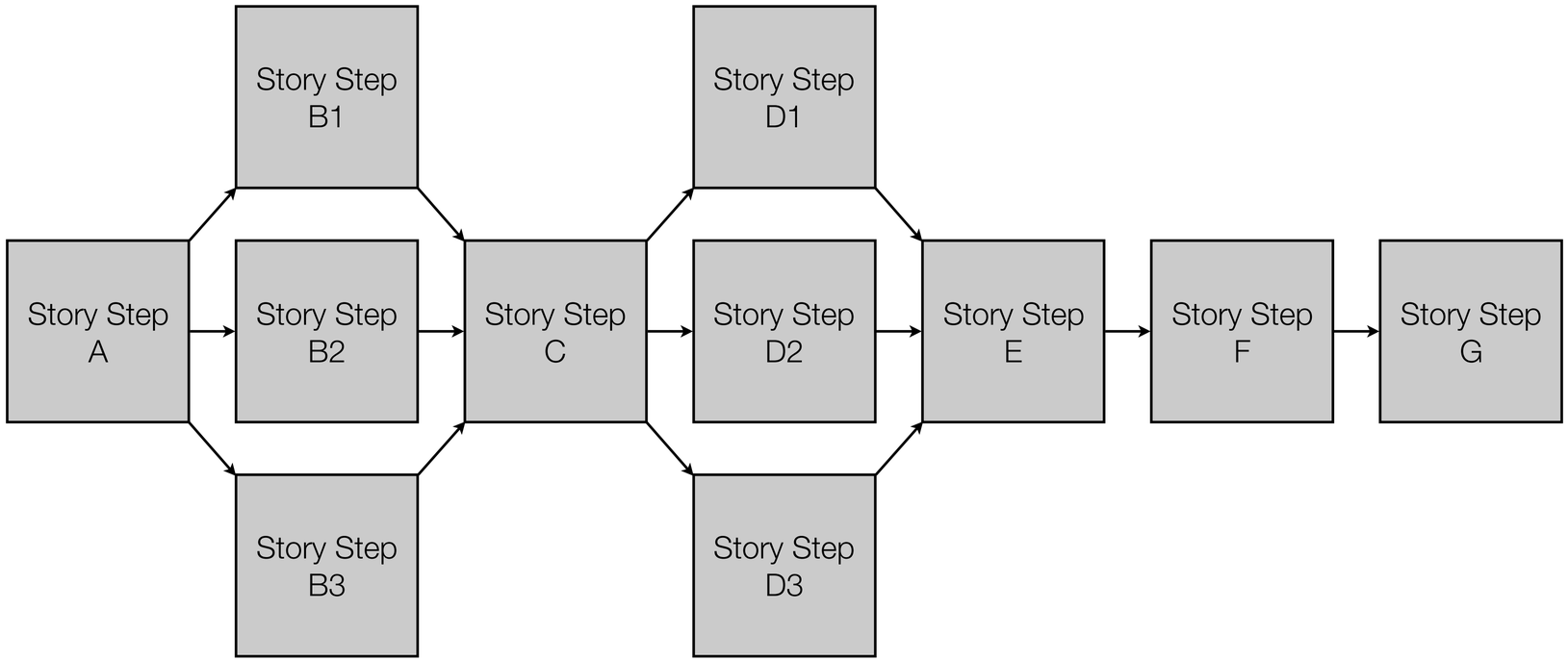


L. Sheldon, *Character Development and Storytelling for Games*. Thomson Course Technology PTR. (2004)

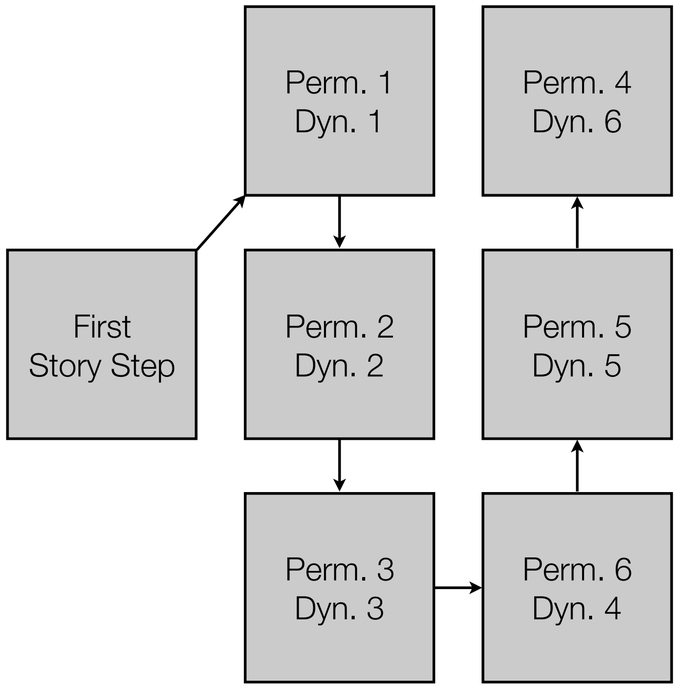
Branching Stories



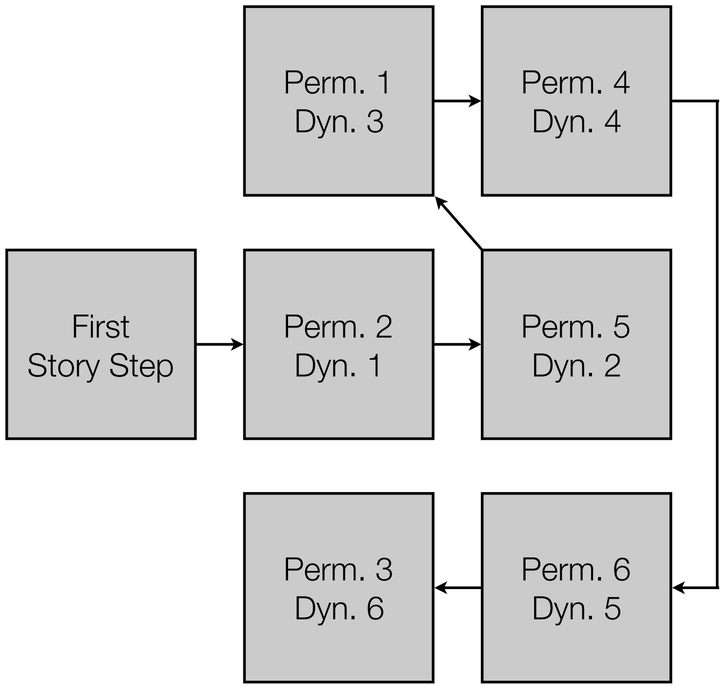
Controlled Branching



Modular Storytelling Example



Player A



Player B

Some More Macros

- Variables:
  - _mytvar: temp variable
  - $mypvar: persistent variable
- Data Structures:
  - Array: (a: "Fred", "Mary", "John", “Jane")
    - Example: (set: $namearray to (a: "Fred", "Mary", "John", "Jane"))
    - To create an array filled with consecutive numbers, use (range:1,5)
      - Example: (set: $numberarray to (range:1,5))
    - To retrieve a particular item in an array, use the item's number: (2) of $namearray
      - Example: (set: $firstfemalename to (2) of $namearray)
- Loops:
  - Over an array: (for: each _name, …$namearray)
  - Over a set of numbers: (for: each _i, …(range:1,5))
- Conditionals:
  - If: (if: $myvar is $myresult)[TheHook]
  - Else: (else-if: $myvar is $myotherresult)[AnotherHook]
  - Final Else: (else:)[TheLastHook]
- User Input:
  - Yes/No: (confirm: "The question")
  - Text: (prompt: "The request", "The default answer")
    - To convert text to a number: (num: (prompt: "The request", "The default answer"))
- Parallelism:
  - Live: (live: 0.3s)[TheChangingHook]
  - Stop: (stop:)

Editing "Arrays"

Arrays

+ Tag

```
(set: $namearray to (a: "Fred", "Mary", "John", "Jane"))
(set: $numberarray to (range:1,5))
(set: $firstfemalename to (2) of $namearray)
$namearray
$numberarray
$firstfemalename
[[Loops]]
```

Editing "Loops"

Loops

+ Tag

```
(set: $myarray to (a: "A","B","C"))
(for: each _myitem, ...$myarray)[_myitem<br/>]
(for: each _i, ...(range:1,3))[_i<br/>]
[[Conditionals]]
```

Editing "Conditionals"

Conditionals

+ Tag

```
(set: $myvar to -1)
(if: $myvar < 0)[$myvar is a negative number]
(else-if: $myvar is 0)[$myvar is zero]
(else:)[$myvar is a positive number]
[[User Input]]
```

Editing "User Input"

User Input

+ Tag

```
(set: $useranswer to (confirm: "Would you like to play a game?"))
(if: $useranswer)[
 (set: $userguess to (num: (prompt: "What number am I thinking of?","0")))
 (if: $userguess is 7)[You got it!] (else:)[Nope, that's not it.]
]
(else:)[OK, maybe another time.]
[[Parallelism]]
```

Editing "Parallelism"

Parallelism

+ Tag

```
(set: $stoprolling to false)
(live:0.3s)[
 (set: $dieroll to (random: 1,6))The current number is: $dieroll
 (if: $stoprolling)[(stop:)]
]
(link: "Stop rolling")[(set: $stoprolling to true)]
```

Features Business Explore Marketplace Pricing This repository Search Sign in or Sign up

soops / sentimood

Watch 2 Star 15 Fork 3

Code Issues 0 Pull requests 0 Projects 0 Insights

A minimal sentiment analyzer based on @thinkroth's "Sentimental" and written in CoffeeScript

6 commits 1 branch 1 release 1 contributor MIT

Branch: master New pull request Find file Clone or download

soops Update bower.json Latest commit 8d51e7a on Jul 1, 2015

LICENSE	Initial commit	2 years ago
README.md	Create README.md	2 years ago
bower.json	Update bower.json	2 years ago
sentimood.coffee	allow hyphenated words and words with zeroes (n00b)	2 years ago
sentimood.js	added bower package and compiled javascript	2 years ago

README.md

sentimood: AFINN sentiment analyzer for the browser

A more-or-less CoffeeScript browser-compatible port of @thinkroth's Sentimental, which was originally written in Node.

usage

After you install the package via Bower (with the command `bower install sentimood`) or just add it to your HTML document's `<head>`, you can initialize Sentimood like so:

```
sentiment = new Sentimood();
```

Then you can do cool things like this:

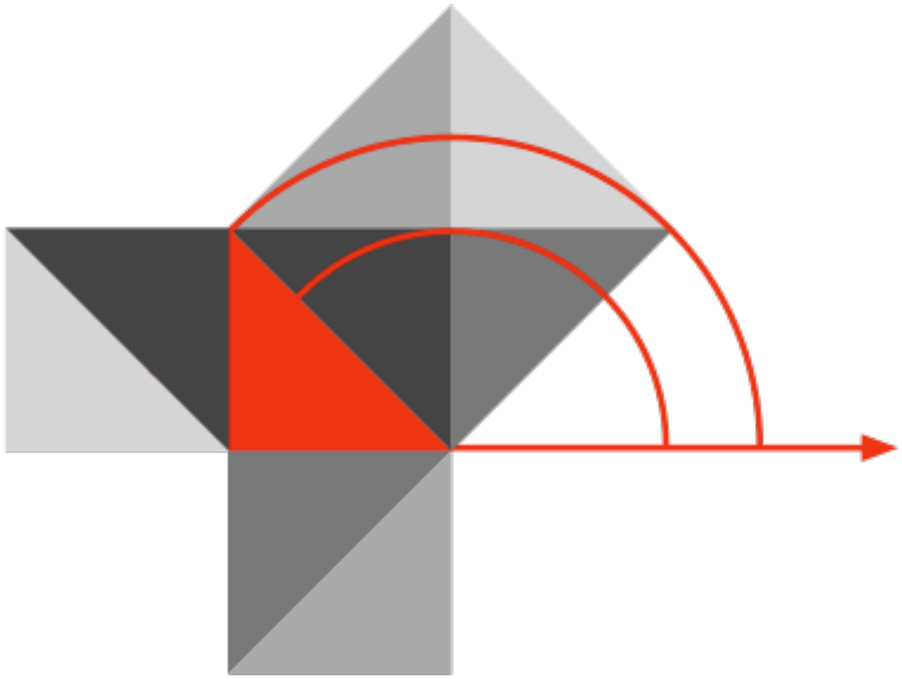
```
var analyze = sentiment.analyze(),
    positivity = sentiment.positivity(),
```

sentimood: AFINN sentiment analyzer for the browser - available online at: https://github.com/soops/sentimood

A Quick Sentiment Analysis

```
<script>
(paste sentiment.js code here)
</script>
<script>
 var sentimood = new Sentimood();
 var mytext = prompt("What do you think about the lamp?");
 var analysis = sentimood.analyze(mytext);
 var sentiresult = analysis.score;
 var customeroutput = alert("The customer sentiment analysis score
is: " + sentiresult);
</script>
```

Hippasus



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