Computational Thinking: A Digital Storytelling Perspective

Ruben R. Puenteđura, Ph.D.
# Dimensions of Computational Thinking

<table>
<thead>
<tr>
<th>Computational Concepts</th>
<th>Computational Practices</th>
<th>Computational Perspectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequences</td>
<td>Being Incremental and Iterative</td>
<td>Expressing</td>
</tr>
<tr>
<td>Loops</td>
<td>Testing and Debugging</td>
<td>Connecting</td>
</tr>
<tr>
<td>Events</td>
<td>Reusing and Remixing</td>
<td>Questioning</td>
</tr>
<tr>
<td>Parallelism</td>
<td>Abstracting and Modularizing</td>
<td></td>
</tr>
<tr>
<td>Conditionals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Computational Thinking in Math and Science

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

---

<table>
<thead>
<tr>
<th>Social</th>
<th>Mobility</th>
<th>Visualization</th>
<th>Storytelling</th>
<th>Gaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>200,000 years</td>
<td>70,000 years</td>
<td>40,000 years</td>
<td>17,000 years</td>
<td>8,000 years</td>
</tr>
</tbody>
</table>

Narrative sources; Narrative constraints
Image Assembly
Sequential Art
Moving Image
Interactive Media
Interactive Fiction
Pictorial vocabulary; Narrative transitions; Text/image integration
CDS Seven Elements; Montage structures
Narrative structures; Narrative flows
Ludic elements

Infinite Canvas

Storytelling

Narrative sources; Narrative constraints
Image Assembly
Sequential Art
Moving Image
Interactive Media
Interactive Fiction
Pictorial vocabulary; Narrative transitions; Text/image integration
CDS Seven Elements; Montage structures
Narrative structures; Narrative flows
Ludic elements

Infinite Canvas

Storytelling

Narrative sources; Narrative constraints
Image Assembly
Sequential Art
Moving Image
Interactive Media
Interactive Fiction
Pictorial vocabulary; Narrative transitions; Text/image integration
CDS Seven Elements; Montage structures
Narrative structures; Narrative flows
Ludic elements

Infinite Canvas

Storytelling

Narrative sources; Narrative constraints
Image Assembly
Sequential Art
Moving Image
Interactive Media
Interactive Fiction
Pictorial vocabulary; Narrative transitions; Text/image integration
CDS Seven Elements; Montage structures
Narrative structures; Narrative flows
Ludic elements

Infinite Canvas

Storytelling
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.”

## Games and Fun

### Successful Games

<table>
<thead>
<tr>
<th>Include These Items…</th>
<th>…To Avoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation before challenges</td>
<td>Results due to pure chance</td>
</tr>
<tr>
<td>A sense of a game space</td>
<td>The perception of the game as trivial</td>
</tr>
<tr>
<td>A solid core mechanic</td>
<td>The game not being perceived as a game</td>
</tr>
<tr>
<td>A range of challenges</td>
<td>The game being exhausted too quickly</td>
</tr>
<tr>
<td>A range of required abilities</td>
<td>The game being perceived as simplistic</td>
</tr>
<tr>
<td>Skill in using the required abilities</td>
<td>The game being perceived as tedious</td>
</tr>
</tbody>
</table>

### Also Have…

<table>
<thead>
<tr>
<th>…Because</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable feedback</td>
</tr>
<tr>
<td>Ways to accommodate beginners &amp; experts</td>
</tr>
<tr>
<td>A definite cost for failure</td>
</tr>
</tbody>
</table>

### In Unsuccessful Games

<table>
<thead>
<tr>
<th>When Players Say…</th>
<th>…They Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>The game is too easy</td>
<td>Game patterns are too simple</td>
</tr>
<tr>
<td>The game is too involved</td>
<td>They are uninterested in the info required to detect patterns</td>
</tr>
<tr>
<td>The game is too hard</td>
<td>Patterns are perceived as noise</td>
</tr>
<tr>
<td>The game becomes too repetitive</td>
<td>New patterns are added too slowly</td>
</tr>
<tr>
<td>The game becomes too hard</td>
<td>New patterns are added too fast</td>
</tr>
<tr>
<td>The game runs out of options</td>
<td>All game patterns are exhausted</td>
</tr>
</tbody>
</table>

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 Krimon in 2009 and is now maintained by a whole bunch of people at several different repositories.
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

Editing “He’s a cement worker”

He’s a cement worker

+ Tag

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

Baker Street

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]]

Ask about the slipper

"That's where I keep my tobacco", Holmes replies.
[[Baker Street]]
Choices: Using a *macro* and a *hook*

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.

(link: "Ask about the slipper")"That's where I keep my tobacco", Holmes replies.
[[Ask about the rent]]
Creating variables and collapsing whitespace

Ask about the rent

+(set: $poor to false)
(set: $wallet to 100)
}I think five guineas should cover it nicely - what do you think?
[[Accept the offer]]
[[Ask for a lower price]]
Using variables and \textit{if/else} macros

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

"Splendid! I'll let Mrs. Hudson know", Holmes replies.
\texttt{(if: $\text{poor}$)[Holmes is clearly sympathetic to your current financial woes, and offers to buy dinner. You gratefully accept.]}\texttt{(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

---

**After Dinner**

+ Tag

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

```plaintext
{(if: $poor)[ [[[You must be exhausted - we can talk more in the morning.]]]}
(else:)[ [[[How about a few rounds of backgammon?]]]}
```

---

**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

How about a few rounds of backgammon?

{(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

<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;text&quot;</td>
<td>text</td>
</tr>
<tr>
<td>Superscript</td>
<td>meters/second^[^2^[[^2</td>
<td>meters/second^2</td>
</tr>
<tr>
<td>Horizontal line</td>
<td>---</td>
<td>___</td>
</tr>
</tbody>
</table>
Basic Screenplay Design

ACT I

- First Half
  - 25% Dramatic Context
  - Plot Point I
    (approx. 20-30)
    17–25%

ACT II

- Second Half
  - Midpoint
    about page 60
  - 50% Dramatic Context

ACT III

- 75%

Plot Point II

(approx. 80-90)
67–75%

SET-UP

CONFRONTATION

RESOLUTION

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

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
## Character Functions

### Introduction

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

### The Donor Sequence

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

### The Body of the Story

<table>
<thead>
<tr>
<th>#</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Villainy</td>
<td>The villain causes harm to a family member.</td>
</tr>
<tr>
<td>8a</td>
<td>Lack</td>
<td>A family member lacks or desires something.</td>
</tr>
<tr>
<td>9</td>
<td>Mediation</td>
<td>A misfortune is made known, the hero is dispatched.</td>
</tr>
<tr>
<td>10</td>
<td>Begin Counteraction</td>
<td>The hero (seeker) agrees to counteraction.</td>
</tr>
<tr>
<td>11</td>
<td>Departure</td>
<td>The hero leaves home.</td>
</tr>
</tbody>
</table>

### The Hero’s Return

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

**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).
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

Introduction

The Body of the Story

The Donor Sequence

The Hero's Return

0. Initial Situation

1. Absentation
2. Interdiction

3. Violation
4. Reconnaissance

5. Delivery
6. Trickery

7. Complicity

8. Villainy
8a. Lack
9. Mediation

10. Begin Counteraction
11. Departure

12. 1st Donor Function

13. Hero's Reaction
14. Receipt of Agent
15. Guidance
16. Struggle
17. Branding

18. Victory

19. Liquidation
19. Liquidation

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
Linear Storytelling

Branching Stories
Controlled Branching
Modular Storytelling Example

Player A
- First Story Step
  - Perm. 1 Dyn. 1
  - Perm. 2 Dyn. 2
  - Perm. 3 Dyn. 3

Player B
- First Story Step
  - Perm. 1 Dyn. 3
  - Perm. 2 Dyn. 1
  - Perm. 3 Dyn. 6

Player A
- Perm. 4 Dyn. 6
  - Perm. 5 Dyn. 5
  - Perm. 6 Dyn. 4

Player B
- Perm. 4 Dyn. 4
  - Perm. 5 Dyn. 2
  - Perm. 6 Dyn. 5
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:)
Arrays

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

Loops

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

Conditionals

(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]

Parallelism

(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)]

User Input

(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]
sentimood: AFINN sentiment analyzer for the browser

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

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:

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

Then you can do cool things like this:

```javascript
var analyzer = sentiment.analyzer();
result = sentiment.likelihood();
```
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>