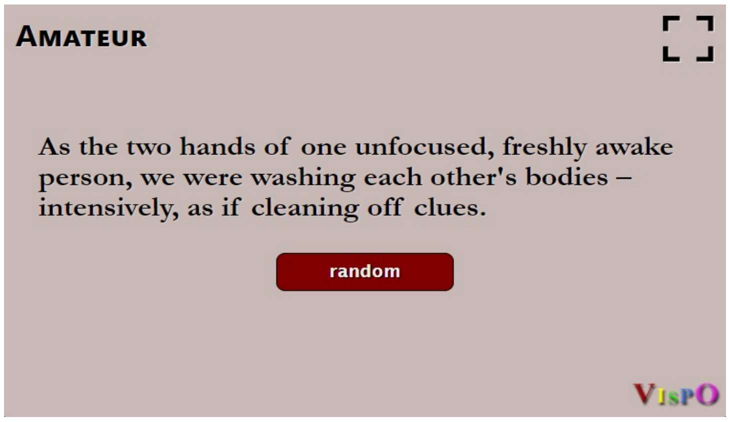
# Multi-dimensional, Non-linear “Amateur”

Jim Andrews
vispo.com



As the two hands of one unfocused, freshly awake person, we were washing each other's bodies – intensively, as if cleaning off clues.

# Multi-dimensional, Non-Linear “Amateur”

Jim Andrews
vispo.com

Kirill Azernyy asked if I was interested in collaborating on an interactive fiction. He’d written a story called “Amateur”, he said, and felt it would profit from an interactive treatment where the reader, at any time, would see only one paragraph of the story, and could advance the display to another paragraph of the story—but to a *random* paragraph of the story, not necessarily the next paragraph.

That struck me as fairly easy to create, from a programming and design point of view, and intrigued me as a reading experience. It’s elegant and simple. So we went ahead with that, and in a few days, we had a prototype. We talked about it, back and forth, and soon we had a ‘completed’ version.

Several months after 'completing' “Amateur”, I suggested we add a new feature. The reader was only able to advance the text *randomly*. Clicking the red button below the story paragraph causes the program to display a *random* paragraph from the story. There was no way for the reader to advance to the *next* paragraph in the story. There was no way to read the story *linearly* except if you read the full text (which is linked from the app) or read the story summary, which is displayable in the app. Which is kind of charming, ie, mandatory non-linearity. Kirill and I talked about the linear ‘next’ feature, decided to try it out, and I subsequently made the story text *itself* clickable; this advances the displayed text to the *next* paragraph.

So, now, the reader can advance linearly *or* non-linearly, depending on whether they click the story text itself or the red button below the story text.

I find this new feature adds interesting dimensions to the experience. I found myself sometimes advancing linearly, sometimes randomly. You come to understand the text much better, and get a stronger sense of the sequentiality but also non-linearity of the underlying story.

By reading linearly, you see how “Amateur” itself advances or proceeds. It's quite non-linear, advancing from sentence to sentence with intriguing yet not exactly narratively-normal connections and development, often—though there is indeed a story that even has a displayable summary in the app itself.

Kirill has mentioned how the story proceeds sentence by sentence with emphasis on every sentence itself being worthy of extended reading or contemplation. In relation to the whole story, of course, but also in and of itself, as a poetic statement of some sort. That’s an important part of the pleasure of reading this text.

I read “Amateur” linearly, for a few clicks, and then clicked the red button a few times, and thereby experienced fully random sequencing of paragraphs. I compared the random experience of reading the text with the linear experience. It is at that point that I appreciated the non-linear structure of “Amateur” itself, Kirill’s story, although I had previously read the full text through, linearly.

Of course, linear texts can have non-linear sequencing of paragraphs or many another such textual element, such as scenes, sentences, phrases, words, or even letters, *sequenced non-linearly*.

As a programmer, one learns that non-linear data structures, such as trees, can be and are coded as *arrays* that are ultimately *linear sequences* yet can represent *multi-dimensional* structures. Non-linearity can be coded in linear structures, or *multi-linear* structures (linear structures one after another). Multi-dimensional structures are often made of multiple lines. In poetry. In visual art.

One does not really understand either linearity or non-linearity until one understands this. Writing typically consists of a linear sequence of symbols, yet can encode *multi-dimensional*, *non-linear* structures. We build worlds up from uni-linear, 1-dimensional strings. Called, for instance, sentences.

In mathematics, n-dimensional vectors are coded as a (linear) sequence of n values in ordered n-tuples. As in the 4-tuple (xPosition, yPosition, zPosition, time) or (“dog”, 4, {0}, {a:1}), each of which has 4 dimensions. Each component of a tuple constitutes a dimension of the vector space under consideration. We can think of a vector as a multi-dimensional structure, although it is coded as a sequence of values.

The experience of both random sequencing (via clicking the red button) and linear sequencing (via clicking story text) shows you they’re similar, but different reading experiences. “Amateur” is already tending toward randomness in its narrative sequencing. When you randomize something that’s already random, the result can be close to the original. This makes for two interesting ways to proceed in reading the story/text—because they feel almost the same in their narrative disjunction—one isn’t ‘better’ than another, in the case of this text, “Amateur”.

The experience of being able to navigate like this is not unknown in the reading process—books let us proceed randomly, and skip quickly through the text, nimbly, and read pretty much as non-linearly as we want.

It’s been said that, too often, interactivity is merely a glorified page-turner. In “Amateur”, we have *two* glorified ‘page turners’. One, the red button, advances not to the next page but to a random paragraph of the Amateur text. Clicking the story text, on the other hand, advances to the next paragraph. You don’t get the overview of where you are in the story, unless you have read the summary and/or the full texts, both of which are displayable. Books give you a great overview from moment to moment of where you are in the book, as you hold the material object in your hands. But the Amateur app lets you navigate from paragraph to paragraph with nimble randomness or linearity. It’s a fun, revealing, unusual way to read the text—*this text*, anyway, because of its sentence-to-sentence poetic and its complexity but disjunctive narrative (and other) sequencing.

Kirill wondered if readers would have any motivation to proceed randomly via the red button, given the possibility of proceeding normally/linearly via clicking the story text. Previously, their only option was to proceed randomly. Mandatory non-linearity. My feeling is that, at first, the big red button will attract more clicks than clicking the story text cuz people won’t initially understand the difference, or even that the story text is clickable. They’ll click the red button to advance randomly, because it’s the obvious object to click to change stuff. Although the cursor does change to a pointer when the reader mouses over the story text, indicating clickability. But not many people will pick up on that, either.

Most people won’t *ever* understand that the story text, when clicked, advances to the next paragraph, and the red button advances to a random paragraph. It is stated in the Help, but they won't read it or they won't understand it if they do read it. But some who enjoy the experience of reading and clicking the red button will go on to click "Amateur" at top left. And then possibly "Help". At which point, having by then seen the story text interface, they will understand the Help. They will then understand that they can proceed linearly or randomly.

I expect that, like me, many of them will then proceed with a mixture of both, because it's fun and interesting to do so. And because it seems like that's what we're encouraging them to do. And we are. Interactivity does involve this sort of dialog between the reader and the authors.

A different issue Kirill and I discussed was that it would be good to make "Amateur" more understandable for mobile readers. Thing is, when you use a *mouse* in Amateur, you see tooltips that pop-up when you mouse-over a clickable element. Tooltips allow one to 'explain' or 'hint' at functionality. But on mobile devices, where one uses *touch*, not a *mouse*, tooltips do not appear. Consequently, mobile readers were at something of a disadvantage regarding explanatory/hinting material to help them understand the functionality and accompanying concepts.

The best solution seemed to be to introduce another screen or layer, a Help layer, that consists of only two short sentences:

> "Click story to remember next paragraph.
> Click button to remember random paragraph."

This screen can be summoned via the "Help" option that appears when you click "Amateur" at top left. That brings the number of different screens/layers in the app to four.

It was at this point, when another new screen needed to be added, that I decided to rewrite "Amateur"'s programming to use my ZIndexMgr.js code, which I've used in several projects in which there are multiple screens but just one HTML page. I've used it in artsbirthday.net, Neonio, Slidvid, Sea of Po, and possibly other projects. ZIndexMgr.js is code I've made publicly available via Github. It's a z-index manager. There's a video about it, if you like.

This code is crucial to my being able to develop the sort of apps I want to develop, ie, ones that have one HTML page but multiple screens that must be displayed. Basically, it's a matter of being able to have multiple divs that function as layers, and ZindexMgr.js makes managing these layers coherent and easily doable. ZindexMgr.js is useful when the z-index order of a bunch of layers is changing unpredictably at run-time. ZindexMgr.js provides methods to bring a layer to the front or send it back etc. Once again, we're dealing with the non-linear, only concerning layers, not paragraphs. Oh wait, the layers are coded in divs. Which aren't too far off being paragraphs.

"Amateur" includes another feature I've included in all my recent work: it goes fullscreen on the first click, and provides a fullscreen button that lets the reader exit/enter fullscreen. Previously, I've supplied a fullscreen button, but haven't gone fullscreen on the first click, regardless of *what* they click on; previously, it only went fullscreen after the viewer clicked the fullscreen button. Going fullscreen is important to art in the browser. The browser has enough chrome to be a distraction from the art. When people view my art, I want their full attention. Also, it looks better when it's fullscreen, and how it looks is not insignificant to my work.

The browser itself does not let developers go fullscreen until the viewer first clicks on something/anything.  This is presumably to prevent hackers from going fullscreen immediately upon loading the page, and supplying no easily identifiable way to exit the fullscreen. Though pressing the Esc button will exit fullscreen mode regardless of whether there's an onscreen button for it or not. But, OK, most people don't know that, even though a browser message appears after going fullscreen that tells the reader they can exit fullscreen via the Esc key.

Even so, iOS, in some cases, does not go fully—or, sometimes, at all—fullscreen when it is invoked. Sometimes there's still a horizontal bar at the top of the screen with a button on it, to exit fullscreen. Control freaks? Oh yes.

The history of fullscreen in the browser, going back to 1997, is one of developers not being given the same latitude as in normal apps to go fullscreen. And, even when they can go fullscreen, the real control freaks and audience monopolizers have insisted on retaining some of the screen real-estate.

But now I'm grousing about developer difficulties and art difficulties over the years. Let's get back to "Amateur".

I have enjoyed working on this piece and reading "Amateur" linearly and non-linearly. Reading this way, in turn non-linear, linear, non-linear, linear, etc reveals an interesting document/story structure, and the story grows in a cumulative, gradual way. Kirill has created a compelling literary experience.