HTML5 Canvas, User Illusions, and Game Flow

Larry J. Crockett
Augsburg College, Minneapolis MN, USA
crockett@augsburg.edu

Abstract: HTML5 is significant for game pedagogy for many reasons, but the most significant is Canvas, a 2d graphics API that supports apps from word processors to space adventures. Canvas enables interactive dynamic bitmapping, drawing, animation, and simulations such as 3d–scrolling backgrounds with parallax. Alan Kay’s concept user illusion names the “simplified myth” we employ when we use icons, scroll bars, folders and trash cans to use a computer. In Alan Kay’s terms, Canvas creates the user illusion of fully computable animation on many different devices. That is, Canvas enables unlimited development of user illusions. Mihaly Csikszentmihalyi’s venerable concept of “flow” gets new life if we view it through the lens of user illusions articulated in Canvas. Flow ensues when the emotions are not just contained and channeled, but fully aligned with the cognitive faculties to engage a user illusion in a compelling task. Games, in fact, are best understood as immersive user illusions designed to induce flow. As a result, more than any other recent technology, HTML5 Canvas makes immersion in game flow—and the user illusion that most enables learning—fully computable on any device that can display web pages, anywhere.

Keywords: bitmap, graphics API, architecture, consciousness, game flow, learning

1. Introduction

The arrival of HTML5, years in development with more years needed for fuller implementation, is justly celebrated. As a technology partly developed by free thinkers outside traditional HTML standards, it is widely appreciated that it represents a significant turn in the history of markup languages. Among other advances, error handling is improved, it has better support for video and audio, the need for problematical plugins is reduced, and its code is designed to be device–independent. In fact, HTML5 brings a rich array of advances that work together to enhance the user experience.

As a result, HTML5 is significant for web–based game development for many reasons, but I believe that the most significant is the Canvas element, a 2d graphics API enabling the development of apps from web–based graphics editors to space adventures. Since Canvas makes possible interactive dynamic bitmapping, drawing, animation, and simulations such as 3d–scrolling backgrounds with parallax, it is ready–made for networked gaming. Given this treasure trove of possibilities, we might imagine that Canvas is a complex technology with a daunting learning curve. In fact, it represents the best kind of technology: it is conceptually simple—in fact, its base technology is disarmingly unsophisticated, and yet its promise for sophisticated applications is exceptional. It has a relatively easy learning curve, but its potential for web games and game–based, web pedagogy is unprecedented.

This paper argues that the potential for this new (yet venerable) technology is best understood, perhaps surprisingly, in terms of two concepts now decades old. Alan Kay’s computer science concept of user illusion (1984) and Mihaly Csikszentmihalyi’s psychological concept of flow (2008) enable us to see how a simple technology can have such promise. Using work partly dating to Sweetser and Wyeth (2005), I will merge these two concepts and name the merged name, “game flow.” Game flow is the psychological state of complete immersion in a specialized user illusion, namely, a game. More than any other recent technology, I will argue, HTML5 Canvas makes game flow—immersion in a user illusion that most enables learning—fully computable on any device that can display web pages, anywhere.
**2. HTML5 canvas: persistence of vision from a low-tech bitmap**

HTML5’s Canvas element dates to Apple’s development of the Mac OS X Webkit, an open source browser engine, more than a decade ago. Webkit has a large market share, enhanced in part by its use in Kindle e-readers, iOS, and Android. More importantly, in 2006, the Web Hypertext Application Technology Working Group (WHATWG), formed partly in dissatisfaction with the World Wide Web Consortium’s slow development of web standards, adapted Canvas in the proposed specifications for the new HTML5. As a result, Canvas is part of the new “HTML” standard.

Aided by a scripting language such as JavaScript, Canvas supports dynamic rendering of 2D shapes, specifically, as mentioned above, bitmap images. Canvas consists of a drawable bitmapped region defined in HTML code with height and width attributes. As a simple, even low-level technology, it updates a bitmap without the aid of a higher-tech scene graph, a data structure common to more sophisticated, vector-based editors such as Adobe Illustrator and CorelDraw. JavaScript code can access the drawable region with a full set of drawing functions similar to those of other common 2D APIs. As a result, while Canvas uses a pedestrian bitmap technology, dynamically generated and manipulated graphics are possible. This means a web page can generate graphs, animations, and—notably—games.

We might be surprised, given the characterization of Canvas as “pedestrian,” that Canvas has generated the excitement it has. In fact, when detailing the breakthroughs that HTML5 purportedly represents, Canvas is often mentioned first. There are more than a half dozen published books dedicated to Canvas (Benson, 2014; Campesato, 2012; Fulton, S and Fulton, J., 2013, 2014; Geary, 2012; Hawkes, 2012; Rowell, 2011) and many more substantial articles. But this attention may only add to the puzzle: Why would this low-level, bitmap technology, which generates only evanescent images, using a graphics technology which has been around for decades, generate such excitement? As a case in point, Canvas is metaphorically a canvas with real-world canvas properties. For example, if we paint over a red flourish with blue, the red disappears since, unlike Photoshop, there are no layers and there is no memory of what went before.

![Figure 1](image.png)

**Figure 1:** Canvas consists of an empty rectangle with no border by default. Here a rectangle is created to fill most of the space, using fillRect(), a method in the Canvas API. Canvas coordinates can be transformed, with the origin rotated arbitrarily, a major source of Canvas’ power.
It has been often observed that “simplicity is the soul of design.” Notably, the logician Alfred North Whitehead (2012, 83) urged us to “seek simplicity.” The rejoinder he added is, “but distrust it.” In an age of sophisticated graphics cards, efficient algorithms, and vector graphics formats such as scalable vector graphics (SVG), we still might be puzzled by the simplicity of a blank-screen bitmap in the latest, heralded markup language—a canvas element has no content, as illustrated in Figure 1, not even a border of its own. Since Canvas is so simple, we might take Whitehead’s counsel and express our doubts.

More spadework may assuage the deepening puzzle. A canvas is a resolution-dependent bitmap in a web page which we can program in JavaScript to do real-time graphics. Much of the power of HTML5 comes from its integration with JavaScript and Cascading Style Sheets. Creating a canvas context means adding the `<canvas>` element to an HTML document:

```html
<canvas id="mycanvas" width="300" height="150"></canvas>
```

Height and width specify the size of the canvas and an id is specified to get DOM-enabled script access. Each canvas has a context, which provides an application programming interface or API for drawing shapes, rendering text, and sculpting images. To create a context, we pass a string “2d” as an argument, which creates a context object, whose methods enable drawing on our canvas:

```javascript
var context = canvas.getContext('2d');
```

The 2D Context provides objects, methods, and properties to draw and manipulate graphics on a canvas drawing surface. Figure 2 illustrates the simplicity, yet power of the Canvas context, and its API. Just a few lines are needed to complete the face and the code is significantly self documenting. In fact, this is a robust API which explains much of the appeal of Canvas.

```
31 context.beginPath();
32 context.strokeStyle = "red"; // color
33 context.lineWidth = 3;
34 context.arc(200, 50, 20, 0, Math.PI, false); // draw smile
35 context.stroke();
36
37 // eyes
38 context.beginPath();
39 context.fillStyle = "blue"; // color
40 context.arc(190, 45, 3, 0, Math.PI * 2, true); // left eye
41 context.fill();
42 context.arc(210, 45, 3, 0, Math.PI * 2, true); // right eye
43 context.fill();
```

**Figure 2**: Canvas Stick Man. Drawing the stick man takes more than 13 lines but just 13 lines will draw the smile in the face, and the eyes. The code invokes methods that are part of the API.

Of course, games usually require more than static images. Animation, not surprisingly, turns out to be one more user illusion, in this case an optical illusion generated in our principal illusion generator, the brain. Psychologists call this pervasive phenomenon—created by slightly different images with a small enough difference in time between them to induce the illusion—persistence of vision (Hardy, 1920). Persistence of vision explains why the black spaces
between successive frames in a film, for example, are not perceived and why we see pixelation on a monitor screen relatively late and reluctantly. A related idea, long known, is the phi phenomenon. It names the illusion of continuous motion perception when viewing discrete objects if they are arranged the right way. In fact, as illustrated in Figure 3, Jeremy Hinton’s Lilac–Chaser (2005), a sequence of images with no motion in any of the individual images can cause us to perceive movement. We experience at least three different illusions as we watch it. It’s all a rotating bitmap our mind’s eye construes as complex motion.

![Figure 3: Lilac–Chaser generates several illusions: (1) rotating purple balls; (2) a green ball chasing around the circle; and (3) the purple balls disappearing. http://cdn.bigshotcamera.com/images/fun/lilacChaser_lilac.gif.](http://cdn.bigshotcamera.com/images/fun/lilacChaser_lilac.gif)

Simple bitmaps can therefore create complex user illusions, given persistence of vision. The phi phenomenon, evidenced in the biopsychological matrix of ordinary human perception, closes the deal. Bitmaps are constituted digitally in discrete mathematical representations, just as the Lilac–Chaser is comprised of discrete, static images, but can be experienced as continuous, fluid, and malleable. Consequently, the fact that Canvas generates a bitmap is no impediment to a human brain already geared to render discontinuous bitmap changes into a fluid user illusion. We need no training for such perception since it is built in from the beginning. The wisdom of using a low-tech bitmap becomes even more evident: bitmaps, despite their simplicity, lend themselves to the illusions our brains are built to find.

Since the time of the split between rationalism and empiricism in Western philosophy in the European Enlightenment, roughly 1600 to 1800, we have often assumed that sense data are precede interpretation and that the brain’s task is to interpret sense data after the eye receives them. But this is indefensible for a number of reasons. Consider the well–known Necker Cube in Figure 4, dating to the work of Louis Necker in the early 19th century (Bruno et al, 2006). It is a wire frame cube with no identifiable depth cues yet we interpret it as having depth in one or two ways (switching requires conscious effort) without needing study or deliberate interpretation. It might be argued that the Cube’s lack of depth cues means we need time to make an interpretation after seeing it. But it turns out the interpretation infuses the perception from the beginning. Experiments with rotating Necker cubes suggest the human visual system can readily identify cues and patterns to make a continuous illusion (Harrison and Backus, 2014). Our brain, therefore, is a dedicated creator of useful illusions.
To the point, the psychology of the Necker Cube and the Lilac-Chaser suggests we do not and cannot see the world “as it is,” whatever that might mean philosophically. Interpretation cannot be usefully distinguished from the sensing of “data”; we don’t first sense, then interpret, to get perception. Interpretation is not an after-the-fact overlay on sensation. Our brains interpret from the outset of perception, seeking meaningful patterns and building—in the terms used here—a user illusion, a short-hand way to interpret otherwise overwhelming or ambiguous data. Here’s the point. User illusions are not tangential to human perception or consciousness, they are central, even defining. Without them, we can navigate neither the natural nor the formal world. It is the topic considered next in greater depth.

3. User illusions: graphical user interfaces and consciousness

Alan Kay’s concept, “user illusion,” is now several decades old since it debuted in his notable “Computer Software” (1984). Kay’s polymathic article anticipated by several decade today’s raging controversies over “digital humanities” (Kirsch, 2014). A user illusion is the “simplified myth” we employ in order to be able to navigate an environment which would otherwise be too complex. As illustrated in Figure 5, a user illusion such as a graphical user interface traffics in an array of metaphors—in this case, the pointers, file folders and scroll bars on a desktop—so that we can interact successfully with a computer comprised of an unending sea of 1s and 0s manipulated by blindingly fast hardware. Such metaphoric user illusions, it should be noted, are not a luxury or evidence of shallow understanding since they are ineluctable to novice and master alike. It’s not that we don’t have time for the binary representations, or we could use the binary if we worked hard. We are unable cognitively to deal directly with binary data this vast.
Computers, for Kay, bear the same relationship to computing as musical instruments do to music—we make a puerile mistake if we suppose that a guitar, for example, is music or that a computer is computing. We play the strings of a guitar to produce music, Kay would argue, just as we manipulate a GUI with a touchpad so that the computer computes something. Computer hardware enables computing but is not computing. Of course, the materials of software are the simplest possible, 1 and 0, but even the bits are not the computing any more than the strings on the guitar are the music. Instead, it is their choreographed dance that comprises computing. Aesthetics matter in elegant design. Just as it is not the individual notes that determine whether music is inane or beguiling, it is the choreographed arrangement of the bits that determines whether a Canvas animation in a web game bores, intrigues, or befuddles.

As a result, just as we cannot avoid the user illusion of interpretation in everyday life, we have no choice but to interact with a computer through some kind of user illusion. Just as we can view Newton’s laws to be “useful fictions” we use to navigate the natural world in physics, even though they are literally false (Cartwright, 1983), so the user illusion on screen helps us make our way computationally without needing to be an accurate representation. Sitting at the computer scrolling through a document using a scroll bar, for example, the user employs a software-generated metaphor drawn from everyday life. The GUI user illusion presents a friendly array of metaphors, namely icons, pointers, and scroll bars that can be manipulated to get things done. The user illusion of a graphical user interface, for Kay, is in fact a theatrical context. With his use of terms such as “user illusion” and “simplified myth,” Kay therefore puts a glossy computational varnish on the Bard’s lasting insight, “all the world’s a stage.”

The influence of Kay’s idea of a user illusion extends beyond the pages of the original article. Fourteen years later, for example, Tor Nørretranders published his (1998), which extends Kay’s term to human consciousness. According to Nørretranders, we do not and cannot have immediate experience of the world. Metaphorically speaking, of course, perception requires processing time. The brain does not record what is out there in the world but, instead, given...
sensation produced in intentionally shaped dialogue with the environment, actively produces conscious awareness, with a lag-time that is small but real. As suggested in Figure 6, the brain does the seeing, using its user illusion interface that looks specifically, given its purposes. As a result, consciousness is also a user illusion, one that is a necessary condition to our experience and shapes that experience.

**Figure 6: User-illusory vision.** The brain does not passively record, then interpret, what the eyes transmit. Instead the brain does the seeing, using its user illusion interface that searches, given purposes—its user illusion is deeply teleological. If we want to know the forecast, we look past other icons on a screen, effectively not seeing them. As our defining user illusion, consciousness drives perception, framing and illuminating our experience.

To put it another way, according to Nørretranders, consciousness is the brain’s user illusion, it is the brain’s GUI. Just as a graphical user interface rides atop the computational cauldron in a computer, so consciousness is necessarily a theatrical user illusion, effectively a “thin client” GUI which enables us to navigate a complex environment and sustain our belief that we have a continuous identity (Johnson, 1998). Neither the graphical user interface nor the thin client of consciousness chiefly represent. Instead, it is more accurate to say that they usefully construct the theatre in the mind. They spare us debilitating sensational complexity so we can accomplish things we wish to do, allowing us to preclude the irrelevant in our experience, given our goals. This character of our user-illusioned consciousness is important to the discussion of flow below.

Why is consciousness like a thin client? Our brains are subjected to a million bits of information every second, as Nørretranders sees it, from a variety of sources which we cannot accommodate. In fact, Nørretranders argues that consciousness can deal with fewer than 50 bits per second. Whether he has these debatable numbers exactly right does not matter. It is whether he has the order of magnitude about right, and it seems he does. The brain must be ruthlessly selective. In fact, if it is to survive, the human brain must construct its own user illusion that is consciousness. Each second, each of us constructs an exquisitely refined, carefully constructed, emotively saturated state we deem “consciousness.” Consciousness therefore is less about cataloging information about the “real world” than constructing an image, a Canvas–like bitmap, an emotively rich theatrical construction that enables us how to decide what to do next. The human mind, in fact, is the ultimate user illusion, since it seduces us into thinking we perceive reality rather than an interpreted, theatrically scripted, highly selective construction of it.

Here is the larger result of this section that I claim moves us to an unexpected conclusion. Since our user-illusioned consciousness is our unavoidable interface to a world that is too complex to perceive directly, there is a parallel between Canvas and human consciousness: both are in
the business of building user illusions, bitmaps, as it were, modified more or less in real time, for specific purposes. In Kay’s terms, Canvas creates the user illusion of fully computable animation inside a web browser that can sustain the illusion across unlimited kinds of devices. A Canvas bitmap and human consciousness, therefore, have more in common than we might have expected.

4. Game flow and learning on the edge of chaos

Mihaly Csikszentmihalyi’s well-known concept “flow” rivals Thomas Kuhn’s term “paradigm” (Kuhn, 1996) in its use, overuse, and abuse. But it still has considerable power, especially if we view it through the lens of user illusions articulated in Canvas, to help us understand the psychology of web-based games. Human emotional realities and cognitive needs are paramount if we are to engage in successful game flow that enhances learning. Flow ensues when the emotions are not just contained and channeled, but fully aligned with the cognitive faculties to engage a user illusion. Games, in fact, are best understood as immersive user illusions designed to induce psychological flow.

Games are formal systems of logic designed to produce supra-rational, even magical experience. Game mechanics are rational in the sense of being systems of logic (computers are logic systems), but such systems are capable of producing extraordinary flow in people. Players engage significant learning curves and often endure considerable frustration in order to experience flow. A good game designer, in fact, can develop one game that can create endless kinds of flow (Sylvestre, 2013, 39–41). The complex systems idea of emergence explains how this is possible. Simple mechanics interact with the player to generate an effectively infinite variety of complex experiences designed to induce flow.

Flow is complete immersion such that the emotions and cognitive abilities work in tandem, given that the player’s skills are up to the task of coping with the challenge at hand. Flow ensues when the emotions are not just contained and channeled, but fully aligned with the cognitive faculties. Any emotion which distracts the mind from full attention blocks the emergence of flow since it is single-minded immersion in an appealing activity. Remember the point above that our user-illusioned consciousness does not chiefly represent; instead, it constructs the theatre in the mind—a point critical to the concept of flow. In playing a musical instrument, all that matters is “being one with the music and expressing emotion” (Farmer, 1999). Flow, therefore, is single-minded, emotionally rich, undistracted immersion in an appealing challenge.

Consequently, flow is most likely to occur when a person is unreservedly engaging a task. The Buddhist notion of “monkey mind” (Pizer, 2007), being pulled in multiple directions at once, stimulated by unchosen, multiple stimuli—unproductively—is flow’s antithesis. Passive activities or activities associated with “multi-tasking” inhibit the emergence of flow—an important consideration for those students who imagine that multi-tasking can be studying. Illustrated in Figure 7, falling out of the “flow channel” because a challenge is either too hard or too easy, entails one of two results: 1) being overtaken by anxiety, if the challenge is too hard; or 2) dropping into boredom, if the challenge is too easy.
Figure 7: Flow channel between anxiety and boredom. Players can stay in the flow if their skills are a close match for the challenge at hand. If a player’s skills improve faster than the game can adapt in presenting challenge, the player will drop into boredom. If new challenges overwhelm developing skill, the player is overcome by anxiety, and drops out of the flow.

Important for present purposes, a player in the flow completely loses track of what we take to be the real world. Since, as we saw above, consciousness is a user illusion from the start, the emergence of game flow is psychologically simpler than we might imagine since it is a matter of swapping the conventional psychological user illusion for that of the game. Notably, it is not a matter of decamping from the real world to a fictional world. Such experience is a full swap of user illusions, from the user illusion, out-of-game experience to the user illusion, in-game play. In other words, game flow confirms that such play is genuine play, but also genuine experience, in a game world. Game flow ensues in the flow channel (Figure 7), when the player's growing competency can match the growing difficulty of the game's challenges, customized in real time in response to play (Carr et al, 2005, 56).

Another way to characterize the opposite of flow is with the physics-derived notion of entropy, which is the typical state of consciousness, a condition that most people find neither useful nor enjoyable, because it is psychologically disassociative, disabling, yielding no identifiable purpose or direction. Csikszentmihalyi (2008) argues that the battle against entropy is a battle for the self since entropy is conscious disorder and disaffection, a malaise of the soul. In terms of Stephen Wolfram’s (1994) categories, entropy corresponds to Class 3, chaos. Being in the flow corresponds to Class 4, in fact, on the thin, volatile edge between order and chaos. Flow-defeating boredom corresponds to Class 2, periodic order, with no challenge or purpose.

There are additional computational analogues to the psychological concept of flow. For example, a Class 4 cellular automaton will often develop strands in runs analogous to flow. Figure 8 illustrates several strands of flow, most of which stop and two of which continue. A Class 4 cellular automaton such as Rule 110 is provably a universal computer which is computationally irreducible. There is no shortcut to reduce the number of computations needed to simulate the Rule. Any game past a certain level of complexity will similarly be computationally reducible and impossible both to predict in terms of how the game will go and whether or not the player will drop out of the flow.

In related terms, learning occurs in the ephemeral space between certainty and uncertainty, analogous to the “edge of chaos” between Wolfram’s Class 2 (order) and Class 3 (chaos) systems, yielding Class 4, complexity, illustrated in Figure 8. On this precipice between order and chaos, when the player/learner is in the flow, with rapt attention and no distractions,
learning occurs optimally. Good games productively sharpen this edge and beckon participation in the adventures, the risks, and the rewards, as few other environments do.

![Image](image.png)

**Figure 8: Class 4 CA simulating psychological flow.** The light strands on the left stop, simulating the stoppage of flow into entropy, while the two larger ones on the right continue. There is no predicting Class 4 behavior, since it is irreducible, and there is no predicting when player flow will cease.

A canvas element driven by JavaScript, CSS and related APIs, and game engines delivers full, Class 4 computability whose limitations are those of the designer and programmer, not the environment itself. As a sophisticated API onto a simple, quickly computed bitmap that is analogous to the edge of chaos, Canvas is the optimal combination of power and simplicity. The way we simulate cellular automata, of course, is with bitmaps that offer exactly this universal computability.

5. Conclusion

We have only begun to scratch the surface in terms of the relationship of games to learning but we get a good start when we reach an understanding of the relation of game flow to learning. Learning out of the flow is uninspired, difficult, and shallow. Learning in the flow is inspired, much easier, and deeper. Games offer richer possibilities for flow than most other human activities, certainly most conventional pedagogy. Of all the recent technologies that are widely available and free, HTML5 Canvas offers the most accessible, learnable technology that supports game development. Web pages are as ubiquitous as the wide variety of devices that display them—at affordable prices. As a result, HTML5 Canvas makes immersion in game flow—and the user illusion that most enables learning—fully computable on any device that can display HTML5 web pages, anywhere.

References


page 10


