Delusions of Dialogue: Control and Choice in Interactive Art
Author(s): Jim Campbell
Source: *Leonardo*, Vol. 33, No. 2 (2000), pp. 133-136
Published by: The MIT Press
Stable URL: http://www.jstor.org/stable/1576847
Accessed: 13-07-2017 20:38 UTC

DOCUMENT

Delusions of Dialogue: Control and Choice in Interactive Art

Jim Campbell

ABSTRACT

The author discusses the relationship of interactive art to the structure of the computer, in particular commenting on artists' choices in using different kinds of systems, programs and interface devices. He discusses the problems inherent in the reduction of an artist's concept to a mathematical representation, a transformation that is necessary to allow a work to be implemented on a computer. Discussing the potential of the computer to allow a work to be able to change and grow over time through the extraction and storage of information, he looks for new, untouched directions for interactive art.

These days more and more artists are using computers to create artworks. I would like to discuss here the relationship of the structure of the computer and what has been called "interactive art." Although understanding this structure is clearly not relevant to viewing art that incorporates a computer, I believe that some understanding of the way that a computer functions might help us to critically analyze the state of the art and examine why the art has so clearly not reached the level of transcending the technology (Fig. 1).

A computer can be thought of as an empty structure into which a concept is inserted. The concept—which must be represented in a mathematical way—is the program, which is made up of a series of algorithms that define the response of the system. Data is input, and the program reacts and produces an output selected from the vocabulary of that particular system. That output could be an image, a sound, a robot that jumps up and down or a change in room lighting. These outputs all represent the internal direction of the program.

I find it useful to put interactive work on a dynamic spectrum with controllable systems on one end and responsive systems on the other. In controllable systems the actions of the viewer correlate 1-to-1 to the reactions of the system. Interactive CD-ROMs are controllable systems and so, generally speaking, are games. In responsive systems the actions of the viewer are interpreted by the program to create the response of the system. Artificial life artworks fit at the extreme end of this side of the spectrum. This spectrum is fuzzy and often subjective and, more importantly, changes with a person's technological proficiency. If a work is responding in a predictable way and the viewers become aware of the correlation between their actions and the work's response to their actions, then they will feel that they are in control, and the possibility of dialogue is lost. The first time I walked through an automatic door at the supermarket I thought the door was smart and was responding to me. Now I step on the mat to open the door on purpose. The point is that often the first time an interface is experienced it is perceived as being responsive, but if the interface is experienced again it becomes controllable. The second time it is not a question but a command.

It is very hard to avoid the theme of control in computer art because computers are fundamentally designed to be controlling devices. The computer industry's goal of making computers and programs smarter is simply to make computers more efficient at being controlled by the user to get a job done. Why should they do anything else? This is generally what we want computers for: we want them to be passive slaves. One can see this in the software, hardware and interfaces that are currently being used. This model is fine until it collides with art.

For example, let us look at the concept of icons as interface devices. Icons are designed to be precise and accurate and discrete—on or off. They are designed to present a closed set of possibilities. They are not capable of subtlety, ambiguity or question. An interface of choice and control makes sense for a word processor, an information retrieval system or a game, but not as a metaphor for interactivity or dialogue.

I have often wondered why most interactive work feels contrived and designed for a calculated response, like bad art-school art. I have seen so many CD-ROMs and interactive video discs that made me feel like my interaction was completely scripted and predetermined within the pretext of a few choices. A painter can create a painting without consciously thinking about future viewers, and a filmmaker can create a film without being overly affected by predicting the audience's response, but it is almost impossible for an artist creating an interactive work to avoid trying to second-guess the viewer. How else can an artist design an interface without seeing it from the other side? One of the ways that artists can avoid this problem is to consider it from the point of view of the work itself, rather than trying to put themselves in the viewer's shoes. Instead of saying, "as viewer, what can I trigger?," why not say, "as program, what can I measure?" and then, "What can I reflect and what can I express based on some interpretation of the viewer's responses?" This way the work becomes a momentary but dynamic reflection of a thinking process. Because the artist does not write the viewer's side of the interaction, the viewer can respond in a more open way.

One of the consequences of this approach is that the work, like a painting and like a film, exists on its own. There is no "attract" mode. The work is not waiting for a person to complete it. In a way, the work becomes interactive not with people but with its environment. This is particularly important with work that exists in a public space. The degree to which a work feels like a game instead of a dialogue, or the

Jim Campbell (artist, engineer), 1161 De Haro Street, San Francisco, CA 94107, U.S.A. E-mail: <jcam@sirius.com>.

This paper was first presented as a lecture as part of the "Video Viewpoints" series at the Museum of Modern Art, New York, 1996. Manuscript solicited by Sonya Rapoport.

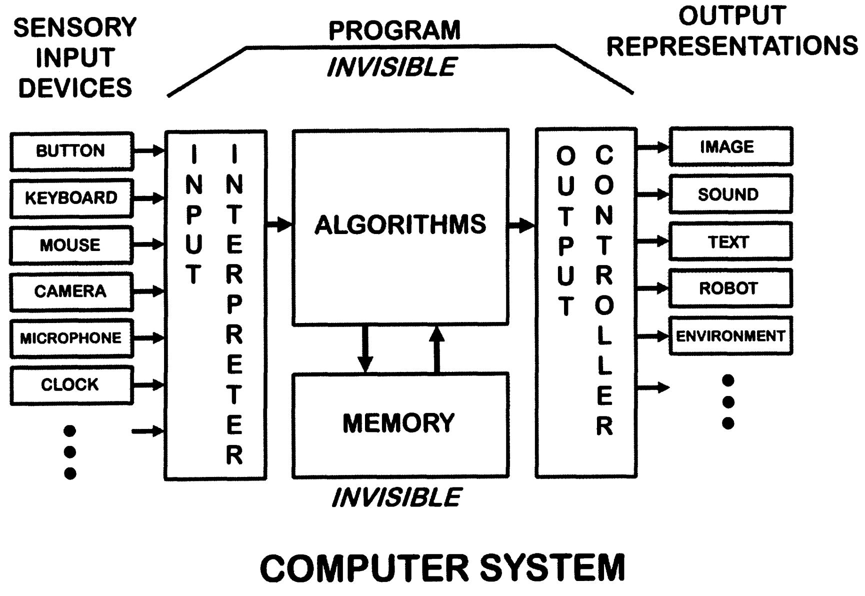

Fig. 1. A simplified diagram of the structure of a computer.

degree to which a work feels like an answer instead of a question, is the choice of the artist and not a limitation of the medium or the technology.

The Program

The program has three main functions: (1) to interpret the sensory input devices (the mouse, the keyboard, the microphone, etc.); (2) to control the memory (what to store, what not to store and what to retrieve); and (3) to select and control a response based on the interpretation of the sensory devices and the memories. I press the letter "k" on the keyboard, and a short time later a graphic representation of the letter k appears on the screen.

Programs are mathematical representations: they have to be defined mathematically. This brings interesting questions to the artistic process when an artist is forced to transform a concept, an emotion or an intuition into a logical representation. This is a difficult thing to do without trivializing the original concept. What often happens during this reductive and transformational process is that the subtlety in the work is lost simply because of the fact that things have to be defined with mathematical precision. A different approach is to start with an idea from technology, and let the work flow from the set of technological possibilities. This avoids the problem of finding a mathematical equivalent by starting with one, but certainly a problem with this approach is that it is difficult to take the work beyond self-referentiality. Often these works are only about the technology that they use, their processes and effects. Yet another way of avoiding this issue of needing to be precise is to use third-party programs to make the transformation from artist's concept to mathematical representation. This solution incorporates a whole new set of problems: the main one being that it is usually the third-party software that becomes the soul of the work—the "Photoshop effect" in which the software that is used was written to create the same effects or responses over and over again. Such software becomes not a tool but a palette of cliché symbols. However, third-party software that is specifically written for a particular work does allow for unique expression in the context of the work, because software is subjective in the transforming process. The way that a program is written has meaningful expression unless the program is performing a trivial function. For example, there will never be a universal program that truly understands a sentence, because sentence comprehension clearly has a subjective element to it. Any sentence comprehension program will exhibit the biases of its programmer within its interpretations.

There is no good way of defining what a program is. A program might be described mathematically as a series of algorithms that choose a new state based on the current state, the past states and the current set of inputs. In this definition, a program has direction. Using an anthropomorphic analogy one might say that a program controls its own time by responding to its senses. It has motivation. Another aspect of a program is that it is completely invisible to the viewer. The viewer can only infer meaning from the program. This trait of invisibility is where the power of illusion lies, and invisibility associated with direction or motivation is the combination of characteristics that cause us to project attributes of life onto or into a computer. It is difficult for viewers not to project intelligence into a program that has "meaningful" responses to their actions.

I did an experiment a while ago to try to show how the simplest of meaningless processes could be combined to imply meaning. I created a second cursor on the computer screen that acted like a shadow of the regular cursor. I then added delay to this second cursor and noise to its coordinate position on the screen. The result was as though this second, shadow cursor were following the first cursor around, as if it were alive. The simplest interpretation would suggest that delay implies thinking and that adding randomness to delay implies volition. Of course there was no life, there was only the sum of some meaningless characteristics of life. The characteristic of following may be a characteristic of something alive, but it is also a characteristic of the behavior of a meaningless computer algorithm. We are still in the illusion stages of computer technology. I have wondered what the extrapolation of the willing suspension of disbelief will mean in our relationship to computers.

Expressive meaning within the program is an important part of an interactive work. During a conversation, a person's words, facial expressions, tone of voice and type of eye contact all point to the ideas and feelings that this person is attempting to communicate. A person's words and behavior provide a window to his or her conscious and subconscious ideas, memories and motivations. The underlying aspects of what is going on inside one's head are certainly an important part of what is being communicated. Analogously, during interaction with a computer, the control and display of the images, audio and text by the computer program all point to the hidden aspects of the program itself. I think this happens whether one wants it to or not, simply because the program is responding to the present. If the program is trivial, then an aspect of the communication will seem trite.

Another way to understand why there is meaning within a program is to look at some older mediums. If one is watching a film or looking at a painting, the images that one sees reference the past in a static way. In an interactive work the images that one sees are dynamically referencing the past.

If the new element particular to the medium of film was time, then I think that the new element to interactive art is the present. And it is the program that connects the present to the past.

There is one very simple thing that a computer program can do that our minds can not. Flip a coin. The ability of a program to make a truly arbitrary decision, an unmotivated decision, is often used to model many naturally occurring processes, but usually it is an inaccurate model. Typically the only characteristic in common with the process being modeled is unpredictability. Irrational behavior, for example, is unpredictable, but it is anything but arbitrary. If many irrational interactions occur within a communication, and the actions point to the same set of hidden motivating forces, they will begin to reveal what these motivations are. A number of random actions will point in many different directions, creating nothing but confusion. Unpredictability need not be confusing but can actually be revealing if the right models are used in the programs.

Most programs are trivial with randomness thrown in to make them seem complex. What this does is make the communication shallow and confusing.

Sensory Inputs

People think of interfaces mainly as a way of getting discrete and accurate information into the computer, because computers process numbers. Input interfaces are ways of converting the real world into numbers (digitizing), but the world is continuous, not discrete. Going back to the idea that interactive works can be put on a spectrum from controllable to responsive, interfaces can be put on a similar spectrum with command on one end and measurement on the other. These usually correspond to discrete interfaces on the command end and continuous on the measurement end. Just where an interface fits on this spectrum has as much to do with its perceived structure as with its implementation.

For example, if a foot switch is mounted under a carpet near a video monitor, and a viewer walks up to the monitor and steps on the carpet, the switch is closed and triggers an image and sound to start playing on the monitor. When the viewer leaves the carpet, the image and sound stop. If the viewer wants to see the image again he or she will step on the carpet. The viewer is not interacting with the image or the program behind it, but merely with the foot switch. There is no dialogue. It is a discrete interface: the switch is on or off; the image is on or off.

In a second example, 100 foot switches are mounted in a row under a carpet to create a position detector that measures a viewer's distance from the same video monitor as above. The system can differentiate 100 possible distances from 0 feet to 20 feet. The image is at maximum brightness, and the sound is at maximum volume when the viewer is 20 feet away, but as the viewer walks closer the image and sound fade, reaching nothing when the viewer is 1 foot away. A viewer will find that the optimal positions for image intelligibility and sound intelligibility will be different. Different viewers might respond in any of a number of ways—oscillating between these two optimal positions, finding a compromise or prioritizing—but the important point is that their actions will be based on what they are seeing and hearing, not on where their feet are. This example illustrates the fundamental difference between discrete interfaces and continuous ones: namely that in discrete interfaces the interaction is between the viewer and the interface, and in continuous interfaces the interaction is between the viewer and the work or the program.

Even though the above interface is discrete, as all digital representations by definition are, it will be perceived as being continuous because the difference between any two of the 100 levels is imperceptible. Of course, one could display the distance as a number between 1 and 100 on the screen along with the image, and this would turn the perceived continuous interface into a discrete one—causing the viewer to interact with the number, as with a slider bar on a Macintosh or Windows program.

Interfaces that involve discrete choices leave little room for intuition. Discrete choices generally cause the viewers to look for a logical reason to make the correct choice based on what they think the consequences might be. Unless it is a game, there is no correct choice.

It has been my experience that intuitive interaction through an interface can only be possible if that interface is able to understand any input of its type. For example, if the interface to the computer is word recognition, then the computer should have a reasonable understanding of anything that might be said to it, not just a few words. If the interface to the computer is a distance-measuring device then the computer should understand distance in any direction that it is approached from, not just from straight on. If an interface has holes in it with regard to its structure, then it will be disregarded by a viewer simply as metaphorical, and any interaction that does occur between a viewer and the work will get stuck at the interface. The interaction that occurs will be between the viewer and the interface, not between the viewer and the work or the program, as I suggested earlier with discrete interfaces. A transparent interface is a continuous one that is perceivably complete within its type of structure.

Memory

Like the program, the memory in a computer is also invisible, even more so because information must travel through the program to get into or out of memory. The process that connects the real world to the internal memory must involve transformation but may or may not involve interpretation. It usually does not. For example, a moving image is stored as a moving image and later is played back as the same moving image. In this case, transformation takes place at the input device, the camera system, by digitizing the image, and at the output device by un-digitizing the image and displaying it. This process is simply the regurgitation of raw data. This does not have to be the case. The current structure of the computer allows for the possibility of interpreting an input and subsequently storing this interpretation in memory. The original data need not be stored at all. The potential of the computer to be able to extract information from an input and store it, not as raw data but as associated data, is one of the fundamental characteristics that allows for a work to be able to change and grow with time, even changing its vocabulary along the way. To me this is one of the most exciting and unique possibilities in computer art, and very little work has been done in this area. The possibilities exist for works that perceivably never repeat themselves. Works that respond to their environment not just in a short-term way, but in a long-term way, unpredictably and meaningfully (easier said than done).

Conclusion

The difference between an interactive game and an interactive work of art is not just in the subject matter. It is also in the program and interface, which are important parts of the expression of a work. Artists working with computers will continue to be at odds with the models and directions of the multimedia industry.

Interactive art that uses a computer is still in its developmental stages. The computer is certainly the first medium in history in which the expression of an emotion or a concept has to be reduced to a mathematical form. Probably the only meaningful dialogues that occur while interacting with a work are between the viewers and themselves in the form of feedback systems. The limitations that we are up against at this point are no longer technological. Possibly, as we begin to understand more about communication, it will be possible to express not merely a thought *of* the past, but a thinking process *in* the present.

Manuscript received 20 August 1999.

Jim Campbell's electronic artworks have been shown internationally and are included in collections of museums such as the San Francisco Museum of Modern Art. In 1992 he created one of the United States' first permanent public interactive video artworks in Phoenix, Arizona. As an engineer, he holds more than a dozen patents in the field of image processing and is currently working on HDTV-related products at Faroudja Laboratories, Sunnyvale, California.