Social Contact

an interactive work of political philosophy

Pierre Depaz – MFA Thesis

Social Contact is a game about exploration.

Social Contact is a computer-aided artistic rendering of our social selves.

Social Contact is a gateway into the labyrinth of process, populated by our assumptions.

Social Contact is an interactive work of political philosophy. The intent was to take the medium of computing and apply it to existing principles that have been embedded in our societies More specifically, I wanted to look at how the ability of computers to create emergent states from an existing set of specific instructions can act as a mirror for our own thoughts. The player is given the affordance to manipulate the software into shaping uncertain patterns. She is then presented with those results, which often don't conform to her expectations . The game features two sets of conflicting instructions. As one setting is picked over another, a mental model is being formed around the expected result from that choice. The first set of instructions is the behavior expected from the player's perspective. However, as the software itself interprets these instructions, it does so in a very different, mathematical and logical, way. These settings, as embodiment of choice, are interpreted in an essentially different way. From this difference a space for thought is created.

Despite being the result of the interaction between the human and the machine, Social Contact chooses not to look primarily at either of them. The subject of Social Contact is deliberately political. The increasing disinterest for political matters is a defining aspect of the current generation, usually characterized by the very fact that those expressing that disinterest, were born before the popularization of microcomputers. There is a natural tendency to take things for granted when one has not witnessed their coming into being. This lack of passion for political thought in western democracies might stem from us taking them for granted, and even their multiple limitations do not appear to trigger conscious philosophical thought and speculation. Presenting a certain version of political thought as a game -the most emotional genre of interactive software- is therefore an attempt at offering a way out of the status quo.

About this preface

This document is presented in conjunction with the code, as a provider of contexts. The first context acts as an explanation of the developer's intent and objectives. This context only serves as an addition to the running simulation, and is in no way an essential component to anyone's interaction with the simulation. The second context, however, is the following print out of the source code. As the exact replication of the set of instructions which allow the game to exist as it is and does, it is an essential component to its existence. In the same sense that a specific piece of music, while being much more than just sheet music, could not exist without it being written down. Because of its highly logical structure and syntax, source code (and the act of programming behind it) is often looked at through a functionalist lens, praising beauty and worth as a function of its efficiency. And yet, code is also writing. Not only is it worthy as the product of cooperation and collaboration between multiple individuals, its semantic richness becomes more obvious when it is, as in this case, the work of a single person.

This printed out copy of Social Contact's source code is therefore presented here as an opportunity to see a more exposed version of what constitutes interactive software. As a piece of subjective writing, it is a testament to the rarely-admitted bias, and humanity, of software. While a fairly complex system, the simulation itself doesn't use overly technical or high-level programming concepts, which would tend to mask personality with best practices. It has been subject to editing, rearranging, and editing, to allow for both an ease of reading by anyone, as well as a clearer declaration of intent. When needed, comments (preceded by '//' are used as clarifications as well as stylistic elements).

About design

One of the definitions of modern play is 'to voluntarily overcome obstacles'. Playing, we want to both figure out the system in which we're playing, and figure out how to become agents in these systems. Designing a game is designing a system for others to interact with . As creative or commercial works, digital game designs can either tend to be explicitly intended to provide instant, standard gratification or, on the contrary, as politically neutral as possible, for commerce's sake. And yet, all systems still represent certain general ideas, a certain perspective on the world. Choosing to allow people to interact in certain ways while presenting a particular goal is a heavily political act, and the rules laid by the designers find their social equivalent in the laws passed by human beings...

Political philosophy shares several traits with game design, with treaties of the enlightenment acting as rulebooks for our societies. Interacting with a computer simulation is having the opportunity, for an instant, to both become lawmaker and law-abider. Particularly in digital games, we willingly examine a possibly obfuscated system, testing the rules until they show their limitations. Through this rigorous examination, we start to see the cracks, the biased assumptions on which this system was built. Social Contact is a look on political thought in general, and on the individual political thoughts of both the player, and the designer.

Description of the process

Most of the theoretical foundations for our modern western democracies- which, through war and colonization, have expanded to the quasi-totality of the planet - were first formalized as political treaties between the sixteenth and eighteenth centuries, most notably through the works of John Locke, Thomas Hobbes, Jean-Jacques Rousseau and Jeremy Bentham. Political theory is the theory of our living together, and political philosophy is the constant questioning of how that is possible. Their political systems which define our modern understanding of 'democracy', 'republic', 'representative

assembly', are built first and foremost on the assumptions that these thinkers made about the nature of individuals, about our nature.

They were thinking about how we come together, how we interact with each other within our community, and how we deal with strangers, with those that have evolved in other communities, unbeknownst to us.

These ideas came to be in their own minds, through thought processes. This idea of the thought process is this exercise in trying to follow steps according to our own logic, this inner debate which requires significant intellectual discipline. It is particularly efficient when it comes to mediating, publicizing concepts others could relate to by following the same steps, and seems to be directly linked to the development of the printing press.

By delegating these intermediate steps between assumption and consequences of these assumptions, we can more easily gain a new perspectives on these beliefs. These processes have indeed the same, and perhaps a more rigorous discipline in terms of logic. One of the building blocks of programming, the if statement, is a clear manifestation of programming as structured thought, the same structured thought which constitutes philosophical practice. "If someone thinks in a certain way, then it must act accordingly".Computers take this maxim to its logical extreme, and will act only according to the way the programmer tells them to. It is worth mentioning that, even though bugs are often seen as limitations of the machine, of failures of the system, they actually are failures of the programmer, unable to express her thoughts in a way that would allow that machine to output the expected result. Because of their impartiality, computer simulations, as the digital incarnation of agent-based models allow for the rigorous exploration of our assumptions. An algorithm doesn't take into account the input's bias when outputting a result. Inputting assumptions about individuals does not always output expected social results.

About writing code

There is a certain pleasure in writing, and it persists in the act of writing code. Beyond just a representation of the certain mental models of whoever wrote it, code has spawned stylistic appreciation of programming. The word, and particularly the written word, by being the translation from a human concept to machine-readable data, seems to further increase the divide between signifier and signified. The very grammar of code, the flow of its prose, are representing very different things, bytes of data recombined through electrical current. This very physical limitation to what words can mean is something unique to writing code. There is a clear demarcation between intent and result as it passes through the filter of computing. It is a sort technical literature, and therefore has formal features which would be interesting to see unveiled more often.

Looking at the syntax of code, one can start decoding familiar elements of style that have been core of literature, and mirror's of the author's mind. For the following couple of lines of code, the result might be the same, and yet the process in which this result has been accomplished is semantically very different.

Let us start by introducing two variables, arbitrarily named 'agent' and 'love', as present in the source code of Social Contact. The former will be a representation of an individual and the latter will be a representation of love, brought into existence by having computer memory allocated to their being.

```
var agent;
var love;
```

We could make her fall in love by writing this:

```
agent.love = true;
```

This implies that love is actually a built-in feature of she, that it is a binary possibility which exists within every similar types of data (i.e. human beings). This feature can be declared to be true or false, but once it has been changed, it will continue to stay in that particular state, unless some external factor affects its state.

Or we could write it this way:

```
agent.love();
```

What is intended behind this phrasing is that love is a possibility which exists within each of us, but isn't a given. Love isn't suddenly true or false, it is an action which yields a given result, and whose output isn't immediately obvious. We are talking about her love, but do not know what is the qualitative aspect of that love.

Here is yet another way to make the agent fall in love:

```
love(she);
```

This syntax suggests that love is an action, a sphere of possibility independent from any individuals. However, for individuals to fall in love, they have to be realized through that action.

Code isn't built on separating different senses, either. Because everything is numbers, words can mean colors and colors can be sounds. Movement can be information and information can be rearranged endlessly. Social Contact takes advantage of this unique capacity by resisting the temptation to import any external assets, and writes any necessary audio-visual element featured by the executed code. The circles wandering in their flat world are words. The sounds they emit as they find each other are words, the stories told at each period are each time assembled through written instructions, and not just taken from a pool of pre-existing documents. Code is more than just instructions, it is a the actualization of style, shape, color, sound and political thought.

As one interacts with it, I can only hope that it fosters a renewed interest in how social systems work, how our beliefs take shape, how code runs; and, eventually, that it helps players to realize that all of it is neither irreversible or invisible, nature-like or god-like, but made and thought of by humans, and thus open to change.