

## SOFTWARE ART CATEGORY

-----  
transmediale.04  
-----

jury statement by Geoff Cox, Casey Reas & Kate Rich:

'Software Art' has become a fashionable term and so caution is recommended. This recent attention is due in no small part to the transmediale festival and its previous jury statements (with the software art category in its fourth iteration) but more recently due to other cultural events that grant critical attention to the activity of programming and the material of 'code' (the 'Readme' festival and its associated 'Runme' software art repository, the Ars Electronica festival 'Code', the CODEDOC exhibition at the Whitney and later at Ars - to mention only a few recent examples). There has been a flurry of discussion around the reception of these events over the last few months particularly on mail-lists (on nettime and rhizome) and presentations at conferences and associated proceedings. Thus, any statement on 'software art' at this juncture must be underwritten by the concern that this might be a further stage of the institutionalisation of a set of emergent practices (and its commodification), unless a critical approach is engaged simultaneously. There is some danger of making fixed definitions here in a way that contradicts the very principles of software as something 'generative'; that is always in progress, and on execution produces unpredictable outcomes, that remains in a process of 'becoming' in other words. Whereas some observers would place emphasis on 'software', others tend to privilege 'art' rather than investigate the contradictions that arise when the two terms are placed together. Similar contradictions arise with the idea of a competition in this regard where collaborative, free and open source principles are a guiding ethos for much of the work. Is a software art competition simply a contradiction in terms?

The transmediale festival definition goes some way to counter reductive tendencies in forcing together technical and cultural modes. It reads: 'Software is not to be understood as a functional tool... but as a generative means for the creation of machinic and social processes'. Without doubt, software is a thoroughly cultural phenomena but there is clearly need for more critical work in this area of software production. The recent book by Matthew Fuller 'Beyond the Blip' (2003) is useful in this connection in calling for a criticism of 'software culture' that does not operate at some distance from practice but that takes account of practice, and that is a practice in itself of course. He does this through presenting examples of practices and categories that are not exclusive but simply ideas in progress - here the subtitle of the essay helps our understanding of the status of his project: 'some routes into "software criticism," more ways out'. He offers three categories, briefly summarised as: firstly, 'critical software' designed to undermine normalised understandings of the operations of software itself; secondly, 'social software' developed and changed through social networks of users and programmers, that emerges from a different set of social relations such as those of the 'open source' community; and thirdly, 'speculative software' that reflexively investigates itself, what he calls the 'reinvention of software by its own means... as mutant epistemology'. Criticism of the categories themselves is encouraged as a normal part of the expected critical work to be engaged. For our task, we take this as a useful backdrop to our selection process - not least as a recommendation for a reflexive approach to the 'Software Art' category, this jury statement, and our competition selection (and any lingering uneasiness about our roles as imagined 'experts' in this respect).

To contest whether the term or category 'software art' is useful is a good starting point for any statement such as this, and a recognition of how 'new media' terms become normalised and historicised by the establishment (through festivals, conferences, publications and so on). Whilst recognising we are unavoidably part of this dubious process, we maintain that 'software art' can be useful in itself in revealing these contradictory tendencies.

To discuss software, its relation to hardware appears fundamental to an understanding of its processes and operations. Perhaps one challenge for software art is to consider whether it is

possible to conceive of software without hardware - both in describing a technical relation and a metaphorical one. The three shortlisted works for the software art award all challenge easy distinctions: whether a computer is needed to execute the work (socialfiction's .walk), how software might be extended beyond itself to satire (Conglomco.org and The Carbon Defense League's Re-code), and to avoid prosecution (Robert Luxembourg's The Conceptual Crisis of Private Property as a Crisis in Practice) - all interestingly collaborative and/or anonymously attributed and demonstrating certain ambiguities. Some of the work we did not finally select stresses these issues in other directions: the importance of hardware (Olaf Val's Micro-controllers), software in the public realm (Marius Watz's Drawing Machine) and the use of software overtly as a tool (Heath Bunting's Cube tasks - in this work, a simple application allows for the translation of individual to-do lists at an artist-run collective into an online, machine-readable system).

Admittedly partial (and trying to be polemical), our selection aims to reveal these tensions - we do not claim these works to be the best but simply that they offer a point of productive discussion in terms of the festival category definition. As might be expected, much of the work did not necessarily conform to this definition or our additional 'criteria' and desires - much of the submissions seemed somehow too familiar, reworking known paradigms (using google to insert 'live' data), homogeneous and rather locked into conceptual, cultural and geographical norms (most work by men from D and US). (For instance, few works submitted engaged with sound, use of sensors, mobile phone or handheld platforms, gps or location-based work, wireless networks, environmental issues, cartographies, genetics... whereas these are key issues for 'software culture' at large.) On the contrary, we were looking for work that expanded the category (beyond simply any expected google results), that challenged our expectations (and tastes), and expanded the potential of the terms and categories on offer. To do this in our collective terms, work had to operate critically and in a social context. We wish for software that has 'use-value', that engages its public, that is 'write-able', that reflects the intrinsic contradictory and potentially disruptive qualities of 'software art'. Our selection is thus not only partial but an attempt to foreground social, critical and speculative concerns in terms of the development of a discourse and critical practice around the category of 'software art' itself. The works highlighted by the jury reflects these interests in operating socially, through active participation and representing an expanded view of what constitutes 'software', 'art' and 'software art'. Somewhat exemplifying the festival slogan 'fly utopia', the short-listed works are not just critical, social and speculative but, in our view, productive in offering a program/programme of action.

/ / / / / / / / / /

Shortlisted works for Software Art Award:

-----

Conglomco.org and The Carbon Defense League, Re-code

<http://www.re-code.com/>

Re-code is a conceptual parody inspired by Priceline.com with visual reference to Walmart.com. It is a social hack exposing the software structures which dominate our lives and economy. In their words, they are trying to 'showcase the absurdity in a system that allows corporate theft to go unpunished while deeply criminilizing petty consumer theft'. They were able to generate strong media and corporate response with articles were published in USA Today, Wired News, salon.com and the project covered on stations such as BBCWorld and CNN. The strength of the work is the detail in which the entire system is considered and exposed. They are using the Web, the enormous UPC system and scanning equipment, scripts for generating UPC symbols, easy access to printers, databases, the advent of self-scanning machines, and even mental state of the checkout employees, realizing that the nature of their job causes them to not notice anomolies in the price of goods. It points out the strengths and weaknesses of all these systems and exploits them for social commentary. The simply built website, re-code.com, makes all the essential resources available for communicating their point and serves as a publishing

board, and vehicle for generating corporate distain. This uses a familiar form of satire by techno-art collectives such as RTmark and etoy but with does so with panache. The ultimate effect of such work perhaps remains in question.

///

Robert Luxembourg, The Conceptual Crisis of Private Property as a Crisis in Practice  
<http://rolux.net/crisis/>

The lengthy title indicates the critical trajectory of this work based on a quote from Hardt and Negri's Empire. Submitted to competition under the pseudonym of Robert Luxembourg, you are presented with a script (crisis.php), an explanatory text file (crisis.txt) and a screenshot (crisis.png) as a conceptual puzzle that together sits uneasily in an arts context. If you run the program it parses the screenshot into the full text of the novel 'Cryptonomicon' by Neal Stephenson. The project thus forms a neat conceptual loop between form and content addressing issues of encryption, privacy and intellectual property rights. It allows the reader to gain access to the novel in such a way that the author of the software remains within the legal constraints imposed on the author of the novel by the publishers. The work thus pushes the limits of the legal apparatus that Hardt and Negri see as underpinning the power structures of contemporary capitalism (Empire). Property rights are only infringed on execution of the script itself. The software on the other hand is distributed overtly as free, open software (under the terms of the GNU General Public License agreement). On closer examination, the work emerges from earlier works by textz.com/project gutenberg and the production of the software that lies behind the encryption process (pngreader v1.1) that works under the principle that images and texts have the same underlying code of zeros and ones. Any output that is encoded in such a way (using the pngwriter) can be decoded (using the pngreader) allowing for the covert distribution of copyrighted materials. The user is 'instructed' to perform an illegal act by running the php script by executing the simple instructions in the text file. Furthermore, the submission plays an ironic joke on the nature of competition itself, the commodification of software and the production of art (available expensively as a limited edition or free over the network) that we hope is somewhat enhanced by its inclusion as exemplar of good practice in software art. It unashamedly reveals contradictions and is thoroughly ideological.

///

socialfiction.org, .WALK  
<http://www.socialfiction.org/dotwalk/>

It is claimed that: 'the technology will find uses of the street on its own'. In the tradition of psychogeography, social fiction invites active participation in the execution of algorithms for walking as the practice of 'generative psychogeography'. The website exists to distribute these machinic instructions and reveal some of the inevitable ambiguities of enacting something that appears deterministic when it is influenced by the subjective factors of human intervention. Walkers enact decision-making normally assigned to machine processes. There is no computer needed for this work, and in terms of the relationship between software and hardware, the walkers analogously extend the software to hardware, and to the platform of the streets (extended to the material world in opposition to 'immateriality'). The implementation of the algorithms is thus decidedly unreliable and can be seen to be generative. The code is interpreted by the public and executed in the streets allowing for unpredictable and chance encounters in the spirit of the reference to the psychogeographic practices it evokes ('drifting' leads to 'detournement', in other words - and yet we wonder whether the overt politics of the situationists is somewhat lapsed in this project). A simple stroll algorithm follows: '// Classic.walk; Repeat { 1st street left; 2nd street right; 2nd street left }'. This is both clearly understandable even to the non-specialist and wildly unpredictable in its outcomes. Complexity and conceptual ingenuity is evoked, not by the programmer, but by the execution of the program - unlike the 'closed' operations of much contemporary software (and much of the work submitted to the 'software art' category). The work manages to collide the

protocols of computing and something as ordinary as walking, collapsing the distinction between programming and fiction, as well as the contemporary interest in alternative cartographies (the practices of map-making/reading). It promotes the public understanding and interpretation of code outside the highly specialised language of computing and develops a new syntax or grammar for walking. In this way, dotwalk operates more like a programming language than stand-alone 'software art', with use-value (for instance, in 'search & rescue', devising an algorithm for covering every street in a given area). The idea is endlessly re-write-able in its collective distributed form and as a collaborative developmental model of practice in the open source tradition. The project history as well as the presentation code is available on the social fiction website for further use.

/// // // // //

--

works receiving honorary mention:

-----

Jonah Brucker-Cohen, Bumplist: An Email Community for the Determined

Bumplist is a mailing list that only allows a fixed number of subscribers (currently 6) so that when a new person subscribes, the first subscriber is automatically 'bumped'. In this way, it can be seen as a degenerate algorithm that operates on a set community of users and the accepted protocols. Thus, it forces a certain reflection on the technical and social conventions therein, inviting an active response to the structures of participation it sets up - subverting the accepted sense of what constitutes a 'Net community'. It is a curious intervention, taking an existing convention of a subscribable mailing list and then changes one rule for effect. The project plays with contradictions around openness and restriction, introducing the idea of scarcity into the 'limitless' resource-space of online communications. Ultimately, it questions the motivating factor in joining a list beyond the obsession determination of not being part of it - and not being bumped.

/// // // // //

Alex Galloway (RSG), How to Win "Super Mario Bros"  
<http://www.rhizome.org/rsg>

Alex Galloway's clever, obsessive adventure into the past of video games turns the 'joy' of gaming into quantitative data. The excitement of playing the game is reduced to an ASCII tablature for winning every level in each of the game's eight worlds. While some jury members (those with video game addictions) found this to be a perfect comment on the nature of gaming, others found it rather nihilistic. The project's strength lies in the thorough documentation of one person's subjective experience into a rational matrix. The tablature is augmented by hours of video clips, each showing the fingers on the game pad, tediously pressing buttons on the path to victory. While the video images from the game are not shown, the bright, childlike audio gives hints at the activities on the screen and provides the contrast to the monotonous button pressing.

/// // // // //

Margarete Jahrmann & Max Moswitzer, Nybble-Engine-Toolz  
<http://www.climax.at>

The origin of the project is in the gaming modification community, and nybble-engine-toolz is a modification of the 'c++kernel of the unreal-engine'. The project is notable for its ambition and desire to reconfigure our conventions for interacting with data creating new visualisations and interfaces - it is excessive, almost 'baroque' in terms of its realisation. Its highly-

stylized form merges with text data to create an aggressive visual assault. While some members of the jury felt the project obscured computational processes, others felt it liberated them from their purely formal and antiquated representations.

/ / / / / / / / / /

Yunchal Kim, void(traffic)

This is a spectacular visualisation of data traffic, proposed as an installation for the transmediale festival, demonstrating a transparency of process (running on a debian distribution of linux). This a neat sense of translation as a direct visualisation of the actual code being executed on the server in real-time - and thus cannot simply be considered as an example of ascii video where the characters are arbitrary. The code is displayed literally in waves of activity - with increased activity, the waves grow higher and vice versa. It is evocative and presents a technical process in a way that is easily understandable for the non-specialist. It is undoubtedly a highly-aestheticised experience (engaging the perceptive senses) - but this could also be considered in more negative terms as emphasising the 'spectacle' of new media and perhaps even placing the viewer in passive terms. Despite this potential worry, the work successfully combines technical and aesthetic ingenuity in a way that few of the entries to the software category manage.

/ / / / / / / / / /