Among Black Boxes and Maze-building Rats: Reflections on Art Making with Autonomous Rules Systems

Assistant Prof. Chad Eby, MFA
School of Art and Visual Studies, University of Kentucky, Lexington, KY, U.S.A.
chadeby.studio
e-mail: chad.eby@uky.edu

Abstract

This paper serves as a reflection on a series of collaborative art-making relationships between me, an artist engaged in the creation of the geolocative sound walk, *Automatic Cities*, and a selection of generative rules systems, some literary (from a just pre-digital past) and others, born digital and computationally intense, with a just-escaped-from-the-lab pedigree.

This site-specific audio work engages with analog and (deceptively) simple generative systems—such as the doubled ABRACADABRA organization that underlies Italo Calvino's *Invisible Cities*— for its macro structure, and to black box machine learning systems like the *Generative Pretrained Transformer 2* (GPT2) and the *Vector Quantized Generative Adversarial Network Plus Contrastive Language-Image Pre-training* (VQGAN+CLIP) for the finer grain of its individual spoken word vignettes and their accompanying illustrations.

While much of this paper reads something like a case study, presenting (hopefully useful) details of specific platforms and approaches encountered while planning and making of *Automatic Cities*, broader questions around collaborations with machine learning in generative art practices and the potential roles of constraint- or prohibition-based rules systems in art-making contexts are also briefly considered.

Raymond Queneau once famously described the *Oulipians*—of whom Calvino is counted as a celebrated member—as "rats who construct the labyrinth from which they plan to escape." This description refers to the group's proclivity for self-imposed "writing constraints." In the process of making *Automatic Cities*, I discovered affinities not only between Oulipian methods and mazes, but also between Oulipian methods and the black boxes of contemporary machine learning systems.

A city where space is imaginary, and time

a construction project, discarded without remorse or guilt. A City to Remember. (GPT-2)

Background and Description of *Automatic Cities*

Automatic Cities is a site-specific geolocative soundwalk situated in Caligari, Sardinia. This digital, network-distributed audio piece consists of fifty-five "bubbles" of sound placed in the urban landscape through GPS coordinates and experienced through an application on mobile devices.

These pockets of sound are arranged in a spatial configuration, creating a walking path through the actual city that guides listeners through soundscapes with music and spoken word descriptions and recollections of imaginary ones. Machine learning-generated visuals shown on listeners' mobile devices complement the audio experience. The act of walking the path set out by the piece mirrors the sequence of Italo Calvino's 1972 novel, *Invisible Cities*. [1]

The overall structure of *Automatic Cities* is derived from Calvino's beautiful labyrinth of eleven thematic groups of cities arranged in precise configurations over nine chapters. Fifty of the sound bubbles describe fantastic cities as imagined by the computer-mediated hallucinations of an AI inference engine probed by Calvino's city categories ("Thin Cities," "Cities and Eyes," "Cities and the Dead," etc.). The generative aspects of the work rely both on this strict Oulipian structuring and on a hodgepodge collection of digital tools employing machine learning, artificial intelligence, and algorithmically mediated randomness.

A link to access *Automatic Cities* is available elsewhere in these proceedings.

Digital Tools

Automatic Cities derives its hallucinatory ambiance primarily through its construction from heavily curated output of machine learning tools. These tools ingest vast collections of images and text—produced originally for countless purposes from unnumbered contexts—and then may be probed to compute an endless shambling procession of exquisite corpses. By turns, these composite fragments appear uncanny, poetic, banal, Byzantine, insulting, nonsensical, promotional, or avuncular: sometimes, all at once. The texts and images that emerge from these processes possess the logic of dreams or the quality of overhead conversations, half-remembered.

Working with these machine learning tools as autonomous systems in their raw form is technically daunting. It requires both significant domain-specific knowledge and strong coding skills. Fortunately for the less technical and differently specialized among us, many of these tools are being wrapped in far more accessible interfaces, including digital breadcrumbs of varying degrees of helpfulness, to guide people interested in working with these tools through the complex processes required. Below, I will introduce some of the interfaces I encountered in making *Automatic Cities* for the benefit of others who may be interested in working with these strange new forms.

In addition to these deep machine learning tools, I used other algorithmic tools of varying autonomy to produce

Automatic Cities. “Neural” synthetic text-to-speech voice actors perform the spoken word recordings of the city texts. Virtual modular synthesizers with both native algorithmic sequencer modules and external rules-based sequencers produce both the ambient sound sculptures that punctuate the piece and the background music beneath the spoken word parts. I will briefly touch on each of these systems.

Finally, it would be a strange not to mention the delivery platform (and systems upstream) that provides a way for listeners to experience the work at all.

The Text: GPT-2 via the InferKit Interface

The descriptions of the imaginary cities are the output of a Generative Pre-trained Transformer 2 (GPT-2) open-source artificial intelligence [2]. Specifically, I generated these texts in collaboration with the web based InferKit service [3]. InferKit, a commercial service developed by Adam King, employs the Megatron_11b [4] language model and makes an improved version of the GPT-2 AI more readily accessible. InferKit achieves this both by reducing the complexity of working with the AI by wrapping the interaction in a simple browser-based interface, and by performing the inference computation server-side—so that the user’s computer doesn’t require any software or hardware beyond what is required to access the web.

To work with InferKit, one enters a string of text into a web form to generate a response. There are a few parameters to roughly guide the output, including the number of characters to generate, specific words to try to include, and categories of text genres (“any,” “fiction,” “news,” “blogs,” “other”). There are also two sliders: one which influences how often the system will discard unlikely text combinations, and one that influences how often the system will sample low probability text. The first slider, *Nucleus sampling top p* amounts to shaping the coherence vs. diversity of the text, and the second slider, *Sampling temperature*, modulates the degree of repetition vs. sensibility of text choices (this latter slider is presented in the interface as “creativity.”)

To generate the city texts, I entered a prompt for the AI that included the thematic word or phrase for the class of city required, such as “In this thin city” or “I remember a city.” In terms of settings, I chose “Fiction” or “Other” as the text category, selected a *nucleus sampling p* setting approaching one, a *sampling temperature* setting of just over one, and asked for 250 characters at a time. Sometimes, I also added the thematic adjective to the “Try to include…” parameter. After entry, the AI would then respond with between 200-250 characters which I would evaluate and choose to accept or reject. By a wide margin, I rejected most output. I was seeking texts that embodied a machine poetics and leaned towards Calvino’s combination of otherworldly and familiar.

Once a promising first set of characters was established, I would continue generating new blocks of text, either accepting or rejecting the blocks as they appeared; these subsequent blocks were generally more likely to survive my edits since the model looks both back as well as forward during inference and when looking back included text I had already selected. Trying to stay within the spirit of constraint-based writing, this was the

only level of editing I allowed myself for the text generation.

Sometimes this method yielded results that nodded toward the *content* of Calvino:

"In that city and port, merchants cater to a varied clientele with the sale of daily necessities such as wine, bread, candies, tobacco, tea, spices, dried fruit and meat, and fruits in season such as lemons, oranges, pears, figs, plums, strawberries, apples, peaches, pomegranates, apricots, figs, sour cherries, currants, hazelnuts, walnuts and chestnuts." (GPT-2)

"I should now list the wares that can profitably be bought here: agate, onyx, chrysoprase, and other varieties of chalcedony: I should praise the flesh of the golden pheasant cooked here over fires of seasoned cherry wood and sprinkled with much sweet marjoram…" [5]

At others, toward the *spirit*:

"In this city, it is said that in the worst months of winter, it is possible to walk on the ice—thin and flexible—between houses. Many of us have walked there, never noticing the houses there, our minds entirely engrossed in wandering the city, relishing the experience of walking without shoes on, or the delight of finding an out-of-the-way place to sit and read, from a book we haven't heard of, read by a writer we haven't met." (GPT-2)

And sometimes, a whole, nearly coherent narrative would emerge, complete with uncanny call-backs and the rhythms of a folktale:

"In this city, it is said that the wind is a skillful, mischievous creature and that the law favors revenge over forgiveness. It is said that in this city, few can say no to the mystery of love and the complicated art of forgiveness.

It was the wind that had called to the old innkeeper, calling from the same room where the wind had offered its tale to me many years ago. It was the wind that had led him from the room, calling from the same room where my mother had screamed to see me cry in my bedroom. It was the wind that had led him to the bell of my doorstep, only for the wind to tell him of what he already knew, the scent of her flesh and the mark of her heart-shaped hand upon his cheek.

It is the wind that has led him to my front door, calling to him from the walls of this city, calling him to the place of my bed, whispering into his ear about the fallen woman he had once loved and lost, the woman who had broken his heart and left him empty.

It is the wind that has led him to my door, calling to him from the walls of this city, calling him to my bed, whispering into his ear that the scent of his woman's skin and her heart-shaped hand would once again fill his thoughts, whispering to him of the love he once held for her and of the love he now held for me.

And it is the wind that has summoned him into my room, whispering into his ear that she is waiting for him in my room, calling to him from the walls of this city, calling him to his love, calling him to his woman, calling him to his dark and sinister love.

It is the wind that has called him to me, calling from the walls of this city, calling

to me from the shadows of my room, calling me to my dark and sinister love. It is the wind that has told me that I am meant for him and that I will not return his love, calling me to a past that cannot be recalled, calling me to a future that cannot be measured, calling me to a secret that has been kept from me, calling me to a dark and haunted love that no one else knows.

I will not return his love, I will not return his love, I will not return his love. I will not return his…" (GPT-2)

In this latter run, the algorithm got caught in a loop at the end and would only repeat the last sentence (with small variations in punctuation) over and over, no matter how many times I generated more text.

The Images: GANs

I next worked with a set of Generative Adversarial Networks (GANs) to create the images that illustrate *Automatic Cities*, most frequently collaborating with a so-called VQGAN + CLIP system [6]. This method refers to a Vector Quantized Generative Adversarial Network paired with a Contrastive Image-Language Pretraining network working together to create synthetic images from text prompts. A Real-ESRGAN (Enhanced Super Resolution Generative Adversarial Network) [7] inference system then enlarges and enhances the raw images.

Figure 1: GAN image "In that city, the dead only sing in their sleep and hear nothing…"

I do not begin to understand the inner workings of these densely acronymed approaches. Of course, there are technical papers available for the curious [8]. At a high level, though, the VQGAN portion of the system iteratively generates images that may relate to the prompt text, and the CLIP portion acts as a judge to decide how closely the images relate to the text. In this way, a single process serves as both a *Generate* and *Curate* verb as laid out in "The Machines Wave Back." [9] After initial image generation, Real-ESRGAN upscales the image with a second Generative Adversarial Network. Fortunately, just as in the case of the text transformers, people have wrapped these complex processes into (relatively) easy-to-use interfaces.

For both VQGAN + CLIP and Real-ESRGAN, these simplified interfaces take the form of Google Colab Notebooks that abstract the complex processes into bite-sized, usually annotated, steps that are run sequentially in a web browser. Katherine Crowson [10] pioneered this approach and is largely responsible for popularizing the use of GANs in generative art contexts. The Colab Notebooks featuring GANs require Graphic Processing Units (GPUs) on

Google's servers. This means that unless you are a paying subscriber to the service, access to the GPUs on the servers may be sporadic, and disk space and memory availability will be rationed.

My process for AI image generation for *Automatic Cities* was to prompt the GAN with evocative words and phrases from the GPT-2 city text, choose a seed number (to set random number generation to a repeatable sequence in case of interruption), set the iterations around 400, and set the image interval to 20 or 30. An image interval value lower than the default makes it easier to monitor early developments. Just as in the GPT-2 text generation, I canceled a majority of GAN runs early in the process because they were developing uninteresting visual compositions or had gone off in an inappropriate or unrecognizable direction.

One strange quirk of GAN processes is that since the training images somehow include their original context, the introduction of adjectives and descriptive phrases to the input may radically alter the graphic presentation of the output. Adding text to the prompt like "bokeh," "8K," "Unreal Engine," or "by Thomas Kincaid" will usually nudge (and sometimes shove!) the output image toward a look that respectively includes depth of field, intricate details, global illumination, or bright pastel colors and brushstrokes.

The Voices: AWS Polly

Synthetic voice actors from Amazon Web Services text-to-speech service, Polly [11], perform the spoken word component of *Automatic Cities*. Polly is yet another browser-based service with a simplified interface, but there is also a command-line interface for advanced users. Polly is a commercial service but can be cost-free for low-volume use in the first year.

Although AWS is a multinational corporation, its US roots are evident in the representation of the available voices' languages and regions. In the higher quality "neural" voice category, US English has nine voices (five female and four male) and UK English has three (two female and one male). Most other languages and regions have only one neural voice or none at all.

The synthesis quality of the neural voices is dramatically superior to the standard ones, but the gain in "naturalness" comes at the expense of control. While standard voices have full support for SSML tags (Speech Synthesis Markup Language)—which are used to shape emphasis, breath, intonation, and other parameters of speech—the neural voices support only a small subset of SSML. In *Automatic Cities*, I chose the neural voices for their superior quality and accepted the narrowing of options for control. For consistency, each category of city has its own synthetic voice actor, which exhausts all but one (sorry, Kendra!) of the US and UK English neural voices.

One SSML parameter that is supported by the neural voices is dynamic range compression. By employing the syntax `<amazon:effect name="drc">` in the console, the middle frequencies of the spoken output are boosted. This type of audio compression potentially makes the voice more intelligible in noisy environments. [12] Owing to the urban setting of *Automatic Cities* I employed this parameter extensively in production.

The Music: VCV Rack + Monome Norns

I composed the ambient audio interludes that occur between "chapters" of *Automatic Cities* and the background music for the city descriptions with a collection of generative sequencers, some native to VCV Rack (a virtual modular synthesis system) [13] and some situated outside VCV Rack within monome's hardware and software ecosystem known as NORNS [14].

The Eurorack modular synthesis standard is an amazingly flexible sound and music generation standard pioneered by Doepfer Musikelektronic in 1996 [15]. This new *de facto* standard atomized the monolithic analog synthesizers of Moog and Buchla into discrete physical units such as oscillators, amplifiers, filters, and sequencers, all sharing a uniform vertical form factor, a standard signal type, and uniform connections via 3.5mm patch cords.

In the late 90s, many companies adopted the standard and began manufacturing Eurorack-compatible modules resulting in a Cambrian-like explosion of diverse forms that, after a brief decline, once again accelerates.

VCV-Rack is a free and open-source virtual simulation (and in the case of some modules, emulation) [16] of a complete Eurorack system with nearly 3000 modules—most open-source—currently available.

This software clears one of the highest hurdles for academic and artistic access to modular synthesis: cost. The platform itself is free, most modules are free or low-cost, and having multiple instances of a module in a single rack incurs no additional costs.

Within VCV Rack, there are many generative sequencing and routing modules with varying levels of autonomy: some are powered by Euclidean division [17], Markov chains [18], physical modeling [19], and other relatively sophisticated algorithms, while others use low-level logic such as comparators [20], or simple probabilistic methods like Bernoulli gates [21]. I "wired" these types of modules together with virtual voltage-controlled oscillators and effects within VCV Rack to compose generative, self-playing "patches" from which I recorded excerpts for the ambient audio of *Automatic Cities*.

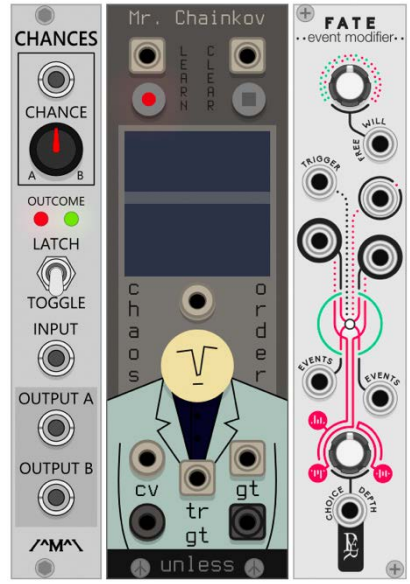

Figure 2: 3 VCV Rack Modules

In addition to the internal sequencers in VCV Rack, I also employed various scripts running on a NORNS shield externally, especially 100 Rabbits' "esoteric programming language," ORCA [22]. I find that the spatial, numerical and

conceptual constraints this tool offers are a close match for Oulipian maze-building.

The audio pieces increase in complexity from the beginning to the middle of the walk—as the diversity of city types per chapter waxes—and then become less complex from the middle to the end as the number of different city types per chapter wanes.

The Platform: ECHOES.xyz

ECHOES [23], located in the City of Bristol in the UK, hosts the sound and image files for *Automatic Cities*. The web- and mobile app-based platform provides a browser-based interface to author geolocated sound work and a mobile app for listeners to experience the work. This is the "distribution" arm of the starfish [24].

The Black Box

The most sophisticated of the generative systems surveyed here, such as the GANs and GPTs, posses many of the benefits and pitfalls endemic to a relationship with highly autonomous art-making systems. On the positive side, these rules systems provide a tremendous extension in pace and scale for exploration, and they excel at presenting surprising outcomes.

I feel there may be some affinity between these machine learning algorithms and the pre-digital Burroughs and Gysin cutups—the ability, with algorithmic temperatures set high enough, to throw off the word-locks and the word-image-locks that Burroughs railed against as instruments of Control [25]. Although, paradoxically, these methods are only possible due to a massive digital surveillance program and some of the tools are provided by the agents of Control themselves.

Because of their relative opacity, these machine learning systems are not especially useful as parameter-driven sketches for testing alternatives through iteration. Small adjustments to inputs can cause *wild* swings in output, and, especially in the case of GANs, the processing time is on a scale that does not permit a flow state to develop between system and artist.

On an even less positive note, working with these highly autonomous generative systems can be a serious distraction, and unless an artist has a strong vision and is willing to curate ruthlessly toward it, she runs the risk of being side-tracked by the novel—and often beguiling—output.

Finally, there is a particularly insidious aspect to these ever-larger machine learning inference engines. The training of a model for their use is an enormously energy-hungry process. Rob Toews, writing in *Forbes* magazine, cites a 2019 study from the University of Massachusetts, Amherst by Emma Strubell that estimates the computation required to train a GPT-2 class model could potentially generate up to 626,155 pounds (about 284,000 kg) of CO^2 emissions—roughly equal to the total lifetime carbon footprint of five cars [26]. Granted, there is a much lower energy expenditure implicated in *using* (as opposed to *training*) these systems, but as Toews points out, the size of the training models grows exponentially with each new generation of the system making future energy consumption an issue worthy of consideration.

The Labyrinth

If relationships with complex machine

learning tools represent an avenue of option-expanding collaboration, what of taking on a set of self-imposed constraints?

As mentioned previously, Italo Calvino was a member of the *Oulipo* (*Ouvroir de littérature potentielle*), a primarily Francophone group of writers and mathematicians. The workshop was (and very much still is!) interested in self-imposed "constrained writing" as method to spur creativity. As is often observed, this impulse seems in keeping with Igor Stravinsky when he wrote: "The more constraints one imposes, the more one frees oneself of the chains that shackle the spirit…the arbitrariness of the constraint only serves to obtain precision of execution" [27]

Some of these Oulipian writing constraints are strongly algorithmic in nature—even if they do not involve digital computers—and appear more like post-processing filters than *a priori* constraints. The S+7 method [28], for example, involves the replacement of each noun in a text with the noun appearing seven places after it in a specific dictionary. This process seems less a *labyrinth* [29] and closer to the previously discussed black box.

Other of the group's constraints are simple prohibitions, perhaps the most notorious of which is disallowing the writer the use of the letter "e." This style of constraint as a rule system is arguably not generative but it is the very image of the labyrinth—which must be overcome through application of skill and effort.

Situated somewhere in the middle, between black box and maze, is, I believe, Calvino's structural constraint from *Invisible Cities*, regulating the categories of cities per chapter, like a doubled amulet of ABRACADABRA:

A
ABR
ABRA
ABRAC
ABRACA
ABRACAD
ABRACADA
ABRACADAB
ABRACADABR
ABRACADABRA
ABRACADABRA
ABRACADABR
ABRACADAB
ABRACADA
ABRACAD
ABRACA
ABRAC
ABRA
ABR
A

This style of constraint is something like a sine wave or lunar cycle. Though neither particularly generative nor prohibitive, it points in two directions and provides a kind of scaffolding (or playfield) to act as a container for the ongoing game.

References

[1] Calvino, Italo, William Weaver, and Italo Calvino. *Invisible Cities*. New York, NY: Harcourt Brace Jovanovich, 1974.

[2] Radford, Alec. "Better Language Models and Their Implications." OpenAI. OpenAI, June 21, 2021. https://openai.com/blog/better-language-models/.

[3] "Inferkit home page." InferKit. Accessed November 7, 2021. https://inferkit.com/.

[4] Pytorch. "Fairseq / Examples / megatron_11b at Main · Pytorch / Fairseq." GitHub. Accessed November 7, 2021. https://github.com/pytorch/fairseq/tree/main/examples/megatron_11b.

[5] Calvino, Italo, William Weaver, and Italo Calvino. *Invisible Cities*. p. 14. New York, NY: Harcourt Brace Jovanovich, 1974.

[6] “Generate Images from Text Prompts with VQGAN and CLIP (z+Quantize Method).” Google Colab. Google. Accessed November 7, 2021. https://colab.research.google.com/drive/1ZAus_gn2RhTZWzOWUpPERNC0Q8OhZRTZ?usp=sharing.

[7] xinntao. “Real-ESRGAN Inference Demo.” Google Colab. Google. Accessed November 8, 2021. https://colab.research.google.com/drive/1k2Zod6kSHEvraybHl50Lys0LerhyTMCo?usp=sharing.

[8] Esser, Patrick, Robin Rombach, and Bjorn Ommer. "Taming transformers for high-resolution image synthesis." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12873-12883. 2021.

[9] Eby, Chad Michael. "The Machines Wave Back." Media-N 15, no. 1 (2019): 69-81.

[10] Crowson, Katherine. “Crowsonkb - Overview.” GitHub. Accessed November 7, 2021. https://github.com/crowsonkb.

[11] North, Freya. “Polly.” Amazon. AWS, 2021. https://aws.amazon.com/polly/.

[12] North, Freya. “Polly.” Amazon. AWS, 2021. https://docs.aws.amazon.com/polly/latest/dg/supportedtags.html#drc-tag.

[13] Belt, Andrew. “Rack.” VCV, 2021. https://vcvrack.com/.

[14] Crabtree, Brian, and Kelli Cain. “Sound Machines for the Exploration of Time and Space.” monome. Accessed November 8, 2021. https://monome.org/norns/.

[15] Hyde, Joseph. "The new analogue: Media archaeology as creative practice in 21st-century audiovisual art." In *Sound and Image*, p. 194. Focal Press, 2020.

[16] Some virtual modules are software emulations of open-source hardware modules; see *Audible Instruments* in VCV Rack which are ports of *Mutable Instruments*

[17] “Euclidean Modules.” VCV Library. Accessed November 7, 2021. https://library.vcvrack.com/?query=euclidean&brand=&tag=&license=.

[18] “Markov Modules.” VCV Library. Accessed November 7, 2021. https://library.vcvrack.com/?query=markov&brand=&tag=&license=.

[19] “Physical Modeling.” VCV Library. Accessed November 7, 2021. https://library.vcvrack.com/?query=&brand=&tag=Physical+modeling&license=

[20] “Logic Modules.” VCV Library. Accessed November 7, 2021. https://library.vcvrack.com/?query=&brand=&tag=Logic&license=

[21] “Bernoulli Gates.” VCV Library.

Accessed November 7, 2021. https://library.vcvrack.com/?query=bernoulli&brand=&tag=&license=

[22] Rek, and Devine. “Orca.” 100R. Accessed November 7, 2021. https://100r.co/site/orca.html.

[23] Kopeček, Josh. “Geolocated Audio Tours & Experiences.” ECHOES, May 31, 2021. https://echoes.xyz/.

[24] Eby, Chad Michael. “What Starfish Know.” Generative Art 2020 Proceedings of XXIII GA Conference (November 15, 2020): 79–85.

[25] Burroughs, William. Interviews. In: Odier, Daniel. *The Job: Interviews with William S. Burroughs*. p. 28. New York: Penguin Books, 1989.

[26] Toews, Rob. “Deep Learning's Carbon Emissions Problem.” Forbes. *Forbes Magazine*, July 21, 2020. https://www.forbes.com/sites/robtoews/2020/06/17/deep-learnings-climate-change-problem/.

[27] Stravinsky, Igor. *Poetics of music in the form of six lessons*. p. 65. Harvard University Press, 1970.

[28] Beaudouin, Valérie. “S+7.” *Ouvroir de Littérature Potentielle*, November 26, 2005. https://www.oulipo.net/fr/contraintes/s7.

[29] Bens, Jacques. *Genèse de l'Oulipo: 1960-1963*. p. 49. Le Castor Astral, 2005.