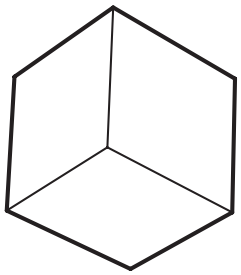
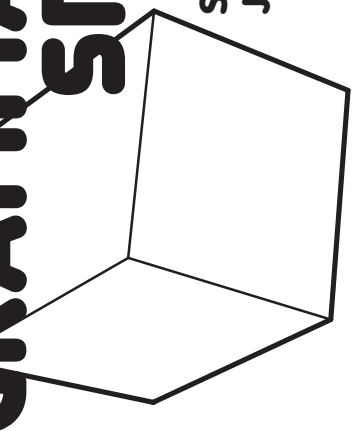
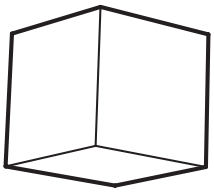
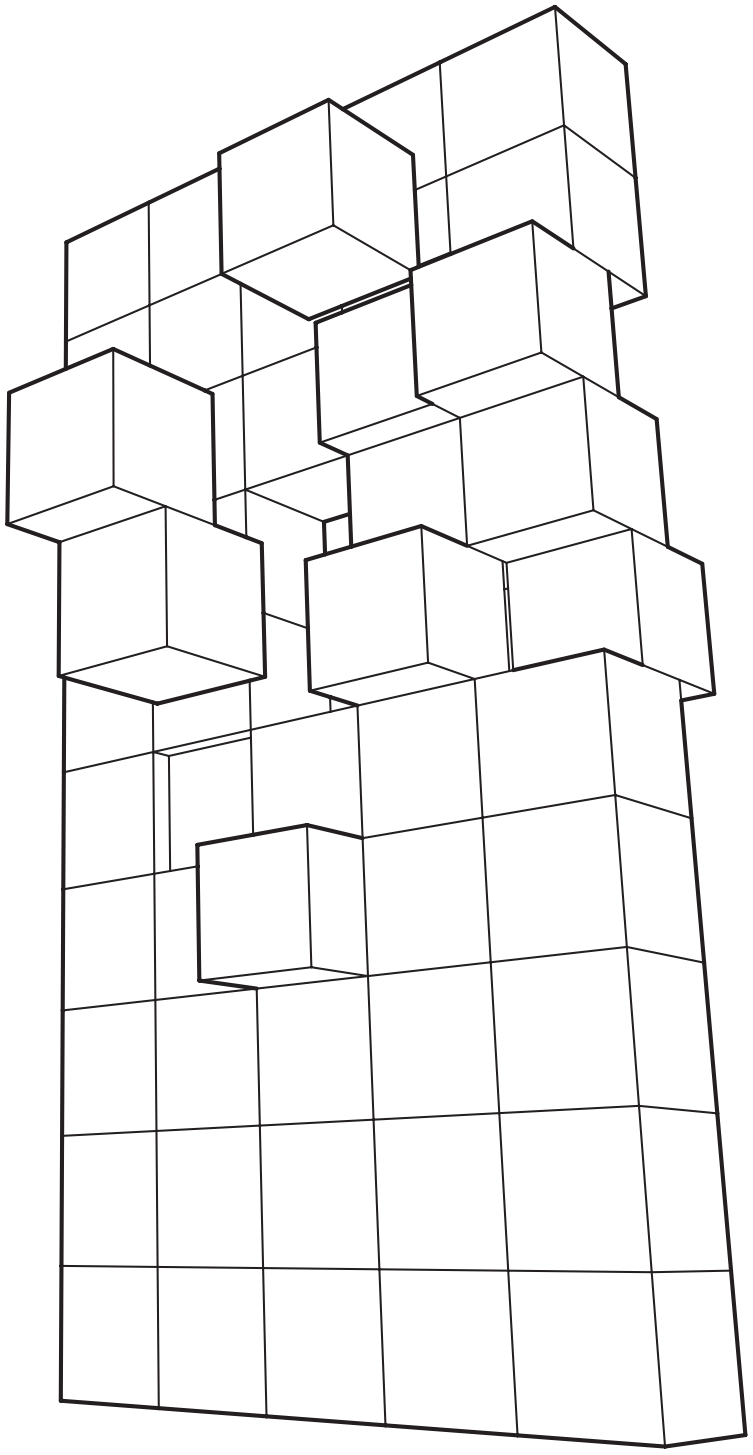
PROGRAMMABLE SPACE

SAM KRONICK
JUNE 2011









THE PROGRAM METAPHOR

In the field of architecture, the term "program" refers to the types of spaces to be included in the building or set of buildings– a university's program would call for classrooms, auditoria, dormitories, food service, etc. The architect's job is, in part, to shuffle around the building masses necessary to house these functions into an effective form. Whereas most architecture renders the program static, a distinct movement of mid-20th century avant-garde architects focused on the design of frameworks or superstructures into which various programatic elements could be inserted, removed, and replaced with relative ease. Such dynamic architecture is exemplified by Cedric Price's *Fun Palace* (1961-1964), Archigram's *Plug-In City* (1964), Yona Friedman's *Spatial City* (1959-1960) and Constant Nieuwenhuys' *New Babylon* (1959-1974). In all these projects, the users of each structure are given agency to reconfigure their environment by moving about walls, staircases, or even entire rooms within a grid of possible locations. In this way, these spaces move from being programmed to become *(re)programmable*. An analogous shift occurred at about the same time as single-function computers (such as World War II-era ballistics trajectory computers, or desktop calculators) transitioned to the von Neumann architecture of contemporary microprocessors– an architecture that supports interchangeable programs.

Suggesting that space itself can become programmable is not an unprecedented metaphor; software architects have been borrowing language and concepts from their brick-and-mortar counterparts for decades. Examples include Eric S. Raymond's seminal paper *The Cathedral and the Bazaar*, Mac OSX's "Core Foundation" API, Ruby on Rails' "scaffolding," and the general focus on "software patterns" (including the "factory method" pattern) which implement the style of thinking suggested in Christopher Alexander's *A Pattern Language* far more extensively than the architects for whom it was originally written. Curiously, this conceptual crossover has been largely one-directional. Contemporary architectural discourse is rife with discussion of how best to use software tools like CAD/CAM and Building Information Modeling systems, but, except perhaps for a few loose interpretations of the term "Open Source," the philosophy of software design has not had a great impact on the art of making space.

CHALLENGES OF PROGRAMMING SPACE

A possible explanation for this asymmetry is that making physical space programmable isn't as easy or effective as it looks on paper. Take, for example, Japanese Metabolist Kisho Kurokawa's *Nakagin Capsule Tower* (1972)– a rare example of built architecture that embodies principles of modular, interchangeable components. Its 140 micro-dwellings are designed to be removed from its two central towers and replaced every 25 years. Thirty-nine years later, the capsules are hopelessly outdated (with their built-in reel-to-reel tape decks and corded telephones) but replacing them with new modules has proven difficult to organize and finance, so the building as a whole is at risk of demolition. It is likely a similar fate would have met the other avant-garde designs would they have been built, too.

Nakagin's capsules seem to be more inspired by the industrial revolution's notion of interchangeable parts than the computational revolution's interchangeable programs. Thus, Nakagin and other blueprint-based speculative architecture can be read as analogous to the mechanical analytical engine of Charles Babbage, an early computer that aspired to do with gears and levers that which was ultimately only made efficient through the transmission of electrons. In an era of cheap digital computing, these architectural visions from the past century deserve a fresh look. Not only are computational tools unthinkably more powerful than they were when Archigram and the rest were designing, they also embody the notions of portability, ubiquity, and pop-culture integration that fueled these designers' visions. With a mobile phone in the pocket of an ever-expanding percentage of the world's population, the low costs of experimenting with software provide a viable alternative to periodically retooling factories as a means of reprogramming space.

Of course, the challenges of making a physical space programmable are not just technical or economical in nature; there are social barriers to achieving a critical mass of spatial dynamism as well. Architects like to think that everyone is as interested as they are in contemplating the buildings around them as a part of everyday life. The mid-century avant-garde was no exception; Price's *Fun Palace* was pitched as a viable business model for a citizenry that would crave spatial reconfiguration as a weekend leisure activity, and the imaginary species *Homo ludens* set to inhabit Constant's *New Babylon* would not only be allowed but expected to build diverse spaces inside a maze of homogeneous, standardized parts as an act of play. That people would jump at the chance to engage with such systems seems a misguided assumption in hindsight. In revisiting these visions, it might be best to look at a growing contemporary subculture that already has practice in the sort of spatial literacy needed to take advantage of reprogrammable space to its fullest. Here, the popular model of video gaming can provide guidance.

VIRTUAL GAMES TO AUGMENTED REALITIES

Some of the most successful video games explicitly use space as a core gameplay mechanism. Maxis' *SimCity* outright casts the player as an urban planner with no other reward than those that come from implementing a satisfying plan, while newer games like Valve's *Portal* series require a player to keenly read the walls, voids, and walkways, of the surrounding space and precisely alter its topology in order to reach the next level. There are plenty more examples of interesting game spaces, but perhaps the most relevant to the mid-century avant-garde is Mojang's *Minecraft*. This low-budget indie hit takes place in a world made entirely of one-meter cubes that bears a striking resemblance to the sprawling overhead grid of Friedman's *Spatial City*. The gameplay stems from the fact that this cubic world is entirely mutable. Every cube can be "mined," carving out voids into the terrain, and subsequently replaced to build up structures with the radical variety of LEGO bricks. Its two-button interface (left click destroys cubes, right click places a new one into the grid) perfectly captures the duality of literacy that is an ability to both read as well as to write within the grammar of a given system. A careful balance of gameplay combined with this novel building mechanic creates a highly addictive experience. The result is a user base of millions; Minecraft's multiplayer worlds are teeming with activity that would make any participation-hungry architect jealous. Minecraft demonstrates that gaming is a format full of deep incentives for participation, capable of training actors to effectively transform their world. What Minecraft lacks (along with myriad other "virtual worlds"), however, is the connection to the physical cities that interested the avant-garde as spaces ripe for critical intervention– spaces that might be the foundation for widespread social reform or political revolution. This is where we must depart from purely synthetic spaces and move into the hybridized space that augmented reality can provide.

Augmented Reality (AR) is a general term that suggests a perceptual blending of the physical with the virtual. At this point in time, it primarily refers to the computer-aided superimposition of textual/graphical annotations or 3D geometry over a live video feed. The format most relevant to the discussion at hand sites these functions inside a mobile phone or tablet computer– computational machinery that contains a critical combination of portability, ubiquity, processing power, location-awareness, and wireless networking which could facilitate urban-scale exploration by a multitude of users. Not only does AR hold potential for its ability to fuse engaging synthetic game spaces with socially meaningful physical places, it works on the level of vision, augmenting the image of the city and thus connecting to a primal sense of how we perceive ourselves as inhabitants of the surrounding environment.

As masters of visual presentation, architects have long been working with a sort of proto-AR to communicate their ideas. Yona Friedman's sketches for Spatial City were often drawn over real photographs of the existing city, using what currently is as a starting point to show what could be. The overhead maps of Constant's New Babylon were made from cut up pieces of other maps, pasted as an overlay onto the intended site. Archigram was famous for their use of collage to illustrate how their radical visions might look in the context of a familiar space. While these images were static and captured only a moment in time, they represented plans for vibrant, active spaces. The real-time, on-site nature of AR has potential to extend these composited, active urban plans of the 20th century with the socially-networked, interactive technology of the 21st.

FORM FOLLOWS FRAMEWORK

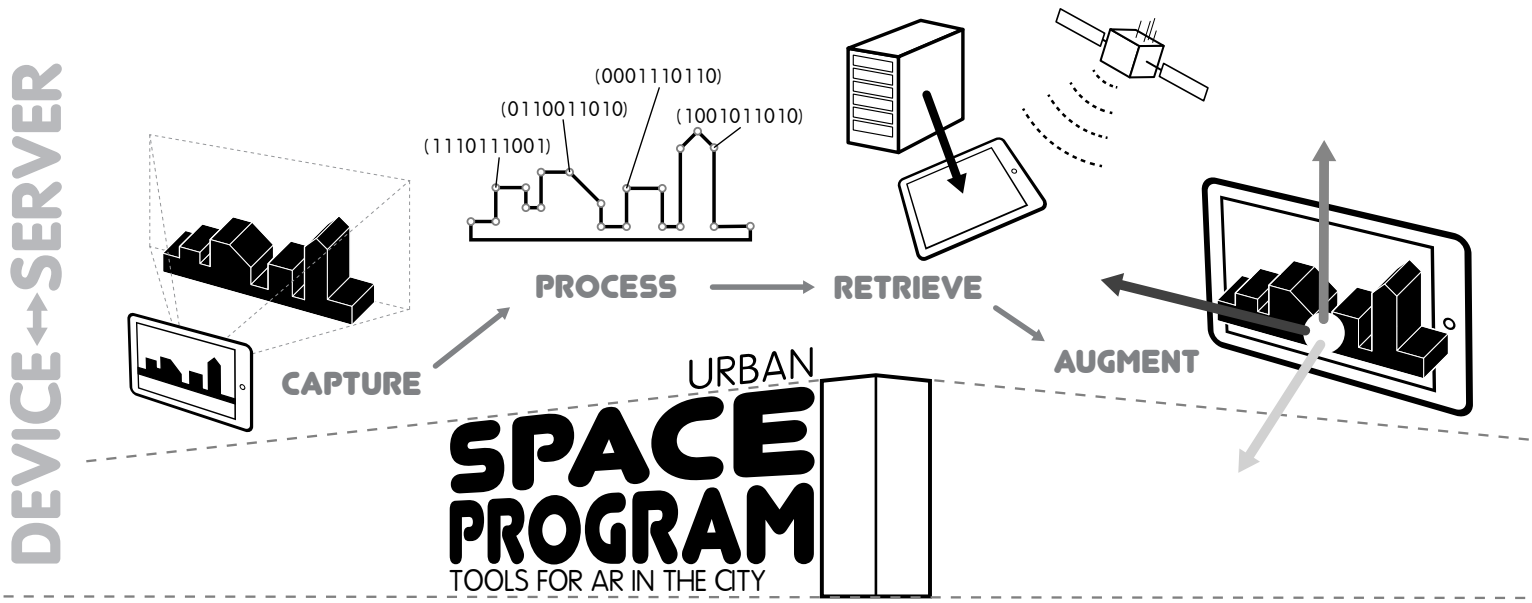
Bringing all this together seems too great a task to be accomplished by a single piece of software. One great AR app will not change the world, just as one Metabolist capsule tower could not transform millennia of architectural wisdom about the importance of permanence. Fortunately, an alternative path can be discerned by hybridizing the historical precedents with the practices of contemporary software design. The mid-20th century avant-garde did not aspire to design singular monuments; instead they focused on the production of systems which would facilitate a certain kind of desired activity. Form followed framework– the latter manifest as structural steel beams, service hookups, communications equipment, and machinery to manipulate modular substructures. Even after years of development, the physical design for the *Fun Palace* consisted of little more than a set of towers and gantry cranes that would move along suspended rails to construct the interior under the command of the structure's spatial programmers. Likewise, the notion of a framework in software design consists of a set of algorithms and data structures which are targeted to approach a certain body of problems without solving any one in particular (frameworks exist for robotics, graphics programming, user interfaces, etc). The constraints and structure of the framework define the configuration space of derivative works– the space of all possible programs that could be built on top of what is put forth as merely a starting point. Therefore the framework creates not a single application, but embedded seeds which self-replicate as they aid and influence how others approach the design process. In this way, a multitude of programs may arise that carry on in the spirit of the original framework but also, as products of various programmers' desires, mitigate the shortcomings or authoritarian impositions of the original structure.

DESIGN CONSIDERATIONS FOR AN URBAN SPACE PROGRAM

Existing augmented reality frameworks neither usefully constrain the problem space by targeting a clear set of applications nor do they provide relevant algorithmic tools that are specific to the task of reprogramming the city. What is needed is an AR framework for spatial programming, targeted at the creative coders who use software as a primary means of aesthetic and conceptual communication. Such a framework is currently under development under the working title Urban Space Program. In developing this framework, several design choices became clear.

First and most importantly, the framework needs to be able to recognize where in the city a user is located and what they are looking at. This can be accomplished by combining location-aware sensors such as GPS, compass, accelerometers, and gyroscopes with computer vision data gathered through a mobile device's camera. All existing AR frameworks at this point in time use one of these methods or the other but do not fuse data from both sources. Frameworks that are location-aware without using vision (such as Layar and Wikitude) tend to augment with textual annotations rather than alternative geometries because of the limited accuracy of GPS; pixel-perfect image alignment is only possible through the use of computer vision. In Urban Space Program, building facades become fiducial markers; the distinct face of a storefront or tower replaces the blocky black-and-white barcode as a machine-readable identifier. The other on-board sensors help limit the search space of which facades might possibly be in view and, using a set of algorithms called "sensor fusion," fill in the gaps when no known facades are visible.

Second, the framework needs to be built with a client-server architecture to enable simultaneous augmentation of a scene by multiple users. If all data were stored on each user's mobile device, changes made by one user would not immediately be available to another user in the same area– this would create a fundamentally single-player experience that runs contrary both to the social act of building promoted by the historical avant-garde and the multiplayer nature of successful contemporary video games. Thus, storage and processing of augmented data and recognizable facades happens on a server and is streamed to the users' devices as they enter a new area and reprogram the space. Rather than centralizing this on one server, the software tools are openly available for installation on any number of servers from which the mobile device can choose to connect to, allowing parallel systems to operate in the same locale.



Third, the database of facades should be built in part with user content. This is suggested as a critique of the historical architectural projects; while designed *for* radical user participation, they were not designed *with* users. In each case the architects unilaterally chose the sites and situations in which activity would take place. In the case of AR, it is easy to imagine that if only the system designer can define which buildings are recognized (and thus which can be easily augmented), this encourages applications of the framework which perpetuate traditional power structures. Recognition of a fixed set of popular landmarks, buildings concentrated in hip districts, or storefronts of sponsoring companies correspond with notions of exploration-as-tourism, advertising, and brand recognition rather than opportunities to truly reimagine and reprogram how a city could be. The Urban Space Program framework includes user interface elements that simplify the process of capturing imagery from a mobile device, processing it for later recognition, and integrating user-captured facades into the database for later retrieval. Of course, the system designer could always choose to exclude these interfaces from their application, but their presence in the framework and presentation as end-user tools rather than developers' utilities suggests that they be included as a sort of meta-game that expands the frontiers of where a particular program can be used.

Finally, if this framework is to connect with a relevant audience of developers, it should tie into existing tools that creative coders are familiar with. Urban Space Program fulfills the role of figuring out where to draw an augmented overlay, but not how or what to draw on top of the live video feed. This latter task is handled by a rendering engine. Conveniently, the creative code community is in many ways defined by the rendering engines it collectively tends to use. Community websites like CreativeApplications.net tag projects according to the rendering framework they use, similar to the way traditional artwork is categorized by material. Tools like Processing, OpenFrameworks, vvvv, and Cinder are common choices for a growing number of artists and designers. Whereas many rendering engines are set up to produce mainstream video game graphics, the tools for creative coders encourage interactive generative design and abstract visualization of data. Very different results are obtained when images are programmed rather than modeled. Integrating Urban Space Program with these other frameworks will hopefully lead to fresh perspectives on how our cities might function and change beyond the well established modes of representing spaces in photorealistic 3D.

FUTURE SPACES

The Urban Space Program framework represents just one possible way to explore the programmability of space. Augmented reality clearly captures the notion that the image of a place could be altered, though it does little to immediately effect the tangible components of a place. Questions of materiality require other means of investigation. As a software-based media, AR inherently engages with processes and algorithms as a method of producing images, though actually reprogramming the dynamics of an urban environment may call for other interventions. An application that is at its core algorithmic could hide nonlinear processes and data behind a linear presentation method, playing back fixed narratives rather than the branching paths of participation proposed by the mid-century avant-garde. Gaming is a broad category covering numerous ludic activities and discerning exactly which mechanics can best transform contemporary urban spaces is another field of inquiry in itself. Urban Space Program does not attempt to resolve all these issues. Rather it is an attempt to reconcile present technologies with past philosophies of space to not only speculate, but actively create possibilities for the near future.