Block'hood

Developing an Architectural Simulation Video Game

Jose Sanchez
University of Southern California (USC)
www.plethora-project.com
jomasan@gmail.com

This paper outlines the conception and goals of the video game Block'hood, an interactive real-time simulation that attempts to bridge the gap between the digital and the physical. The paper presents the analysis of contemporary sand-box games such as 'Minecraft', 'Simcity', 'Factorio' and 'Dwarf Fortress' to establish a design framework. By understanding the video game medium as a real-time distributed crowdsourced simulation, these games aim to provide a divergent set of strategies and goals mainly defined by the users themselves, and do not impose an overarching narrative or bias. They also allow data collected from the user's gameplay to speak for itself, allowing us to understand the ambitions and strategies behind a larger collective crowd.

Keywords: Game design, Crowdsourcing, Combinatorics, Ecological urbanism, Interdependence

0 - INTRODUCTION

Block'hood has been conceptualized as a game of ecological urbanism, placing emphasis on the interdependence of programmatic units. The 'blocks' or tiles available for a player, are modelled after real world data and work under an input / output paradigm, requiring resources to be operated and outputting data and influence over the voxel space of units. The simulation of tiles is based on 3D cellular automata-like calculations, where blocks determine the state and data of neighbors to re-define their own internal data and states.

The ecological simulation of Block'hood has been inspired by 'The Plant' in Chicago, an urban farming facility that operates under the idea of eliminating waste, allowing the output of each productive unit within the facility to become the input for another productive unit. This paper will break down the ecological network developed by 'The Plant' and extrapolate its ideas for the use in an interactive simulation.

Based on such model, the video game Block'hood attempts to model real life objects and abstract them to tiles, allowing the player to discover technologies that are available, and play with them before deciding to purchase them, or make an investment. Energy collection and food production are some of the central themes in the initial stage of the game, connecting the project to an ongoing maker and DIY movement in search for efficient patterns for sustainable living.

The simulation of Block'hood is based in 2 symbi-
otic systems: a rectangular voxel grid that allows for the player's placement of blocks, each of which represent a simulation unit; and mobile agent units that can circulate throughout the voxel. These agents represent the simulated population of the voxel aggregation. The agents are modeled with a series of criteria as requirements allowing them to become the analysis tool for the voxel arrangement. The degree of 'happiness' or 'success' of an agent is based in conditions of accessibility, circulation and overall quality of the voxel arrangement. This paper will outline the inner workings of the 'cell', or Cellular Automata system as the agent based model is only on a prototype stage.

Moreover, the paper will go deeper into the understanding of the use of video games in architecture and design as a crowdsourced mechanism that can potentially democratize design strategies and enable novel performative strategies to be discovered by a non-expert crowd. It also outlines the use of simulation video games as an educational strategy to encourage systems thinking from an early stage of the design process.

1 - GAME ANALYSIS
Current sandbox games offer a diverse set of mechanics that allow players to design and explore generative systems that rely on resource management and ecological interdependence. A field that was pioneered by Will Wright's SimCity, today demonstrates the diversity and maturity achieved by an abundance of titles. Games like Minecraft, Dwarf Fortress, Prison Architect or Factorio, offer excellent case studies for understanding possible different ontologies in the creation of games and the role of the designer in each engine. Within USC, the 'Video Games, New Media and Simulation' seminar conducted by Jose Sanchez, studies and reverse engineers these games to understand the deeper objectives and computations that inform in real time, the role of the player.

Simcity developed a framework that is difficult to improve or break, when thinking of a city building
game (Figure 2). In fact, many games such as 'Cities in Motion', 'Cities XL' and 'City Skyline' follow that framework: a low-res version of what we have in the real world, with approximations of how traffic, labor, energy collection and zoning would actually be played out at the level of a city. The role of the player in these games is always that of a city planner: an authority that can dictate changes in zoning at will and even construct and destroy infrastructure as he or she see fits.

In these SimCity represents the modernistic view on these matters. It shares the believe of for instance Congres Internatioanux d'Architecture Moderne (CIAM), that organic growth will only pose insurmountable problems and that only a rationalistic approach can lead to well working towns. CIAM proposed a distinction between different functions: housing, work, recreation and traffic. In their plans the leading architects of this influential organization divided up space in the most appropriate way for this functions, believing that only in this way a town could grow without chaos. In the first version of the game, SimCity stood very close to these methods and believes.' (Vermeer, 2006)

This is not the only way to think of design and construction within a video game. Minecraft opted to position the player at the same scale in relation to the low-res world (Figure 3), eliminating a 'god' perspective over the object to be designed. The player is an inhabitant of a digital world, and actions need the collection of resources to be performed. The model here is that of conservation of matter, as materials can only be created from other materials that already exist in the game. In this way, the game portrays a flat ontology in which the player is not above the world, but entangled in it. The player needs to spend time to collect and craft any desire. All this however, is within Minecraft's survival mode, where resources are scarce. Minecraft's creative mode breaks with such rules, allowing the player to create anything with infinite resources. While this mode moves closer to the 'god' model, the player still needs to place each block at a time, making larger tasks incredibly laborious and encouraging multiplayer initiatives to develop larger projects.

While Minecraft maps the actions of the player directly into the world, SimCity uses a hybrid model, where some actions are real-time constructions in the city (such as building a street) while others are filtered by the role of the simulation, requiring that the player waits to see the outcome of the simulated inhabitants (When defining building zones). This filtered agency, offers an incredible power to enable a symbiosis between the human intuition and the algorithmic intelligence of the game.

Games like Dwarf Fortress (Figure 4) take the role of this filter type of simulation to a whole new level. In Dwarf Fortress the player inputs intentions to be carried out by a set of agents in the game (the dwarfs). These agents will only perform a task requested if they have the skill, energy or even will to do so. This might seem unproductive, but the simulation engine is ensuring that an agent has received proper working conditions (food, shelter, education) before it can work. The agent might still not perform a task if there are no resources available to complete such task, al-
following the simulation, as in Minecraft, not to create something out of nothing. The engine of Dwarf Fortress trades graphics for a more accurate real time multi-layered simulation. The computation of material properties of each unit produces a high amount of emergent behavior in the player/simulation relation. It is very difficult for a player to anticipate how all systems will play out simultaneously, making the game a great tool for developing systemic thinking. The agent modelling paradigm of Dwarf Fortress allows a player to multi-task design objectives relaying that each task will only be performed if the conditions are required. This engine allows players to do more in less time than Minecraft, as you have the possibility to be able to create a colony of over 200 dwarfs working concurrently.

But the multi-tasking agency of Dwarf Fortress in not the only way to tackle larger complex productions; the game 'Factorio' (Figure 5) operates with a completely different approach. In Factorio, a player starts as a single individual, very much like Minecraft (except from having a bird’s-eye view). The game allows the player to collect resources and build units. The units of Factorio are mainly machines that allow you to collect and build faster. What you build in Factorio is a factory that will perform the operations that might take too long for a single human to accomplish. The careful connectivity and sequential placement of machines along a conveyor belt, allows a player to build a productive city without the help of others. The challenge of a game like Factorio is also that of emergence or a system’s programming, as the constructions become a circuit board that operate over time and space to achieve a particular task. The agency of the player is over a generative system that grows in complexity and develops new functionality, even gradually increasing in intelligence.

What is clear from many of these games, is that the design paradigms embedded in software are plentiful and they bias the player / designer to think and perform a particular kind of action. Many of the games described above use an ecological paradigm where matter cannot be created out of nothing, but rather by the harvest and management of resources. In the weakest examples, we see the use of 'money' as a general currency that can be converted in any other building type, but the stronger simulations, get rid of the concept of money altogether, to pursue a web of ecological interdependence of materials, inviting the player to trace their origin.

---

2 - ECOLOGICAL MODEL - THE PLANT CHICAGO

'The Plant' (Figure 6) is an urban farm and food incubator located in Chicago. The initiative was led by John Edel, who used Kickstarter to partially fund the project. The main idea behind 'The Plant' is to generate a chain of interdependent production, were the waste of one process can be used as the input of another. One good example is the case of aquaponics; the practice of hybridizing hydroponics with aquaculture, generating a symbiotic system. The by-products from the aquaculture production are broken down to be used as nutrients in the hydro-
ponic system, which in turn, feed back into the aquaculture system.

The Plant has taken this approach as an ecological manifesto for the whole facility, looking for creative ways in which different products can feed into each other but also support the ecological growth of the whole facility. In the final steps of the project, an anaerobic digestor utilizes the waste from different production chains to power the electricity and heating of the building. This is the way in which John and his company Bubbly Dynamics, have approached the conception of sustainable development.

"The renovation and re-use of existing buildings is vital to the health of our city. The goal of The Chicago Sustainable Manufacturing Center (CSMC) is to provide an applicable model to the manufacturing sector for ecologically responsible and sustainable urban industrial development. (...)To provide a test bed for the application of energy efficient technologies and methods, using the highest possible amount of recycled material. In this fashion, CSMC will help to revitalize an underdeveloped/urban Chicago area, providing skilled jobs through innovative partnerships and encourage neighboring industry to move toward sustainabilityBubbly Dynamics Mission Statement." (John Edel, n.d.)

This model is perhaps simple and obvious but it's not hardwired in our architectural education. We work with software that allows us to be incredibly creative, but forms are often far from what is possible. Most software assume a condition of abundance and do little work on presenting relevant data to a designer to take informed decisions from the early stages of a project. Current software use a model of post-rationalization, one in which a creation can be evaluated by its performance, but only as an afterthought, creating an iterative loop between design and analysis.

The video games presented above, present a very different picture, making a case for the use of information and constraints in the designs of a player in order to encourage innovation.

Block'hood is a community simulation game designed with these precedents in mind. It does not attempt to design or model a whole city, but rather look at the networks of interdependence that we can find at the scale of an urban block. Using the ecological paradigm of 'The Plant,' the game makes the player responsible for both inputs and outputs required for a programmatic unit to exist, and places emphasis on notions of entropy and management, helping the player to take informed decisions in real time.

3 - BLOCK'HOOD ENGINE INPUT OUTPUT ECOLOGY

Block'hood (Figure 7) can be understood as a multi-state cellular automata. The 3 dimensional grid of units (voxel) allows for neighbor calculations for the transfer of data. Each block represents a cell state, where each state is responsible to perform unique computations. This models have been demonstrated by Micheal Batty;

‘Our simplest models of the city compare locations containing activity, which are cells where change takes place based on actions that influence the cell’s nearest neighbors. These make up cellular automata (CA) whose local action generates spatial order at more global scale.’ (Michael Batty, 2007)

In Block'hood, the cell calculations can be broken down into 2 main categories: input / output (I/O) calculations and adjacency calculations.

The I/O calculations allow each tile to determine the inputs required to generate an output. This inputs are collected from a global pool of resources.
that belong to the player, or in the metaphor of the game, to the community. A block will attempt to produce an output at a specified rate of production. This determines how fast a block can consume inputs to generate outputs. If a block does not have enough resources to generate an output, it will decay, increasing its decay amount by 1 unit. Each unit is able to endure a certain amount of decay, based on its resilience level. Once the resilience of a block is depleted, a block will die or lose its capacity to operate. This engine can be likened to the performance of a tree; a tree requires water (input) to generate oxygen (output). If there is no water available, the tree will start to decay, to the point in which it will die, and will no longer be capable to produce fresh air.

This is the calculation that each one of the current 60 different block types are performing, each with a specific list of required inputs. A block can require a maximum of four different inputs to compute and generate up to four different outputs.

The second kind of computation is the 'adjacency' calculation. While the I/O calculations are performed in parallel with a holistic model, the adjacency calculations, or cellular neighbor computations are performed locally between two units. Each unit will look at its neighbors and determine if there is any relevant data to communicate. For example, a corridor block will first determine if it connects to the exterior of the voxel (border condition), if so that block will be considered accessible. Once a block is accessible, it can pass its accessibility (an internal block data) to any adjacent tile. This allows the system to determine if a particular unit can be reached by people. This kind of computation is performed in different channels allowing for the calculation of cantilevers, water communication and access, among others. Finally, the adjacency calculations allow for each block to have synergies with other blocks allowing for a production bonus if such criteria are met, or a penalty in the case of a negative adjacency. This allows the system to encourage certain combinations of blocks, and challenges players to discover
and share patterns that perform successfully.

Block’hood has been developed as a scalable engine; by developing a custom Unity3D interface, a developer can easily create a new Block by following a wizard, inputting name, inputs, outputs, resilience, etc, of a new block, and link the geometry, icon and sound necessary for the game to allow that new block to be available to the player. This allows for a fast iteration over the interdependence of the system and the creation of new blocks that allow the system to close the ecological cycle.

Blocks are currently defined in four categories; Circulation, Production, Buildings, and Organics. We expect to continue expanding the library of blocks available to players over time, introducing new alternatives and new currencies for trading. The data that each block contains is currently a placeholder for testing gameplay and real time feedback, but the research side of the game ultimately aims for units to contain real world data, allowing the player to perform simulations that can represent real world conditions.

The data that each block collects or outputs, is what we call a resource. Resources in the game are a form of currency and score-keeping. Things such as money, water, electricity, food, etc, are represented in game data, and a player can see the amount they own for each one of them. Again, a scalable approach was implemented, by developing a parallel module that allows a developer to create a new resource, we can define with incredible granularity what kind of input and output each block has. Here we opted to be specific and not use general labels such as food but rather bread, apples or carrots, allowing a deeper specificity for each block, and an exploration experience for a player that is trying to understand how to build a particular network. As previously mentioned, scalability and specificity are the key ideas behind the development.

4 - THE FUTURE OF GAMES IN DESIGN

Block’hood is an initiative attempting to research the future of crowdsourced design and player/user participation in contemporary urbanism. The current methods of participation position the architect as a filter of what communities might express as needs, but often fail to educate a community of the challenges involved in implementing a new idea. By allowing users to play with the variables of a system, they can understand the intricate interdependence of factors involved in city planning.

Education can be an initial goal for the game, but this research aims to go beyond that; Games like Minecraft have demonstrated that an engaged community of players can be a great resource for creativity and innovation. We no longer believe that the solution for an architectural challenge can be found necessarily by one architect, even less so, the problem of the design of the city. This is why game design has been selected as a medium of mass reach and user participation on a grand scale.

Initiatives like ‘Block by Block’ (Figure 8), an educational design platform developed from the collaboration of Mojang and the UN, allows communities to use Minecraft to envision public space and their community. This is a great example of how games are starting to influence decision making and allow user participation in the urban design process.

Block’hood hopes to develop this trajectory by adding to the conversation a deeper intuition of ecological interdependencies; Minecraft is a great engine to design form like playing with lego blocks, but lacks the data implications or contributions of each building unit. The challenge here is to look beyond form, and understand the processes at play, and the challenges that the access to resources and knowledge entail.

The value of things is something that vary locally, in relation to the context. This requires the engine to be flexible enough to model the available resources and their energetic contributions in a particular climate.
Figure 8
Block by Block. UN using Minecraft for citizen participation.

Figure 9
Screenshot of Block’hood by Jose Sanchez
For such specificity, the software design of Block’hood (Figure 9) considers that players and local communities are the most suited candidates to modify the game, as opposed to attempt to create a system that can anticipate every possible variation or technology available.

We are entering an age where exponential design strategies will be necessary to address global problems, and no singular architectural initiative will be able to perform or impact the urban environment in a meaningful way. The ‘Gamescapes’ agenda that has conceived of Block’hood seeks to allow for exponential strategies, models of mass reach and mass customization involving citizens with architects in the design of the city and define what 21st century urbanism is.

REFERENCES
Batty, M. 2007, Cities and Complexity, Understanding Cities with Cellular Automata, Agent Based Models and Fractals, MIT Press