CityMatrix – An Urban Decision Support System Augmented by Artificial Intelligence

by

Yan Zhang

B. Architecture
Tongji University, 2008
M.S. Advanced Architectural Design
Columbia University in the City of New York, 2013

Submitted to the Program in Media Arts and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2017

Signature redacted

Signature of Author. . . . . .
Program in Media Arts and Sciences
August 11, 2017

Signature redacted

Certified by . . . . . .
Kent Larson
Principal Research Scientist
Program in Media Arts and Sciences
Thesis Supervisor

Signature redacted

Accepted by . . . . . .
Pattie Maes
Academic Head
Program in Media Arts and Sciences

CityMatrix – An Urban Decision Support System Augmented by Artificial Intelligence

by

Yan Zhang

Submitted to the Program in Media Arts and Sciences
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Cities are our future. Ninety percent of the world's population growth is expected to take place in cities. Not only are cities becoming bigger, they are also becoming more complex and changing even more rapidly. The decision-making process in urban design and urban planning is outdated. Currently, urban decision-making is mostly a top-down process, with community participation only in its late stages. Furthermore, many design decisions are subjective, rather than based on quantifiable performance and data. Urban simulation and artificial intelligence techniques have become more mature and accessible. However, until now these techniques have not been integrated into the urban decision-making process. Current tools for urban planning do not allow both expert and non-expert stakeholders to explore a range of complex scenarios rapidly with real-time feedback.

To address these challenges, a dynamic, evidence-based decision support system called *CityMatrix* was prototyped. The goals of *CityMatrix* were 1) Designing an intuitive Tangible User Interface (TUI) to improve the accessibility of the decision-making process for non-experts. 2) Creating real-time feedback of multi-objective urban performances to help users evaluate their decisions, thus to enable rapid, collaborative decision-making. 3) Constructing a suggestion-making system that frees stakeholders from excessive, quantitative considerations and allows them to focus on the qualitative aspects of the city, thus helping them define and achieve their goals more efficiently.

CityMatrix was augmented by Artificial Intelligence (AI) techniques including Machine Learning simulation predictions and optimization search algorithms. The hypothesis explored in this work was that the decision quality could be improved by the organic combination of both strength of human intelligence and machine intelligence.

The system was pilot-tested and evaluated by comparing the problem-solving results of volunteers, with or without AI suggestions. Both quantitative and qualitative analytic results showed that *CityMatrix* is a promising tool that helps both professional and non-professional users understand the city better to make more collaborative and better-informed decisions. *CityMatrix* was an effort towards evidence-based, democratic decision-making. Its contributions lie in the application of Machine Learning as a versatile, quick, accurate, and low-cost approach to enable real-time feedback of complex urban simulations and the implementation of the optimization searching algorithms to provide open-ended decision-making suggestions.

Thesis Supervisor: Kent Larson
Title: Principal Research Scientist
Program in Media Arts and Sciences

CityMatrix – An Urban Decision Support System Augmented by Artificial Intelligence

by

Yan Zhang

The following people served as readers for this thesis:

Signature redacted

Reader: . . . . . .

Pattie Maes
Professor of Media Arts and Sciences
Fluid Interfaces Group, MIT Media Lab

CityMatrix – An Urban Decision Support System Augmented by Artificial Intelligence

by

Yan Zhang

The following people served as readers for this thesis:

Signature redacted

Reader: . . . . . .

Cesar A. Hidalgo
Associate Professor
Program in Media Arts and Sciences
Collective Learning Group, MIT Media Lab

Acknowledgments

First, I would like to express my gratitude to my advisor Kent Larson, whose expertise, understanding, and patience have greatly added to my study and research experience. I sincerely thank him for all his guidance during the research and the writing of this thesis.

I would also like to thank my wife, 'Jamie' Ming Zhong, for her continuous, unconditional love and support for my life, dream, and exploration. Her opinions are always fresh, considerate, and delightful.

It would not be possible for me to even start this journey without my mom, Hui Zhou. I would like to thank her for fostering my curiosity since I was young and for her whole-hearted support both mentally and financially.

Next, I would like to extend my gratitude to my readers, Professor Pattie Maes and Professor Cesar Hidalgo, for their guidance, insightful comments, and feedback throughout this whole process.

I would also like to thank MAS Staff, namely Mary Heckbert, Keira Horowitz, and Linda Peterson for being patient, professional, flexible, and supportive.

Next, I have a special thank-you for my under-graduate research assistants, Alex Aubuchon and Kevin Lyons, for being knowledgeable, exploratory, and responsible. This work would not be accomplished without their dedication.

I also extend my gratitude to Arnaud Grignard for his tutorials and helps on GAMA. It is essential for the construction of the traffic simulation in this work.

It has been a wonderful experience to be a part of the Changing Places group and an excellent opportunity to work with all my amazing colleagues and labmates. A big thank-you goes to all my friends at Changing Places group and the MIT Media Lab for all the inspiring advices, stimulating discussions, and fun. My colleagues have been Michael Chia-Liang Lin, Carson Smuts, Yasushi Sakai, Luis Alonso, Ariel Noyman, Arnaud Grignard, Naichun Chen, Ira Winder, Karthik Patanjali, Markus Elkatsha, Agnis Stibe, Mario Siller, Jason Nawyn, Dalma Foldesi, Jung In Seo, Waleed Gowharji, and Mohammad Hadhrawi.

Last but not least, I would like to thank all my family and friends for their support and encouragement through my years at MIT.

Contents

List of Figures

List of Videos

All videos can be access via link: http://yanzhangworks.com/cmv/

Chapter 1

Introduction

1.1 A Global Urban Challenge

The rapid urbanization all over the world brings tremendous challenges. In China, more than 250 million people will move to the city over the next 12 to 15 years [1]. 100 years ago, each major city had a unique identity. Cities today look stunningly similar (Figure 1). They are being developed too quickly and without critical thinking. In the urban decision-making process, there is lack of community participation in developing countries [2].

Figure 1: Cities in China share a similar appearance

In the United States, on the other hand, some projects take more than ten years for community approval. Most of the time, the community engagement process happens in the late stages. It normally is inefficient, uninformed, and non-data-driven (Figure 2). The feedback from community comes late in the process, typically resulting in an expensive and time consuming redesign process. There is on average a six-year delay in starting construction on public projects nationwide because of the inefficient public engagement process. These delays cost the United States over $3.7 trillion US dollars per year [3].

Figure 2: Public engagement meeting in the U.S. Credit: NASA/Goddard/Jacob Larsen

One of the important reasons for these delays is the lack of rapid-prototyping tools for both experts and non-experts to explore multiple scenarios collaboratively. Designing and planning with traditional tools, such as sketching on a zoning map or using Computer Aid Design (CAD) software, requires years of training, making it difficult to use these traditional tools in a public event.

1.2 Urban Simulation Tools

Today, many urban simulation tools are available, including traffic, solar, wind, thermal, and energy consumption. These emerging urban simulation tools help us understand the impact of our decisions in a quantitative fashion.

However, even for professionals, some urban simulation tools are not only difficult to learn but also consume an enormous amount of time and computational resources to run for each iteration. Complex, urban scale simulations normally take hours or even days, which makes the design and collaboration process high-cost, inefficient, and non-intuitive. Most simulation tools only focus on a single aspect of the performance, whereas a city is a complex system. To understand it better, stakeholders need tools that address the internal trade-off between multiple dimensions of the city's performance. For instance, no system currently addresses and quantifies concepts such as innovation potential [4] and resource efficiency concurrently.

1.3 Rise of Artificial Intelligence

Along with the development of information technology, the growth of the computational capacity and the data our society has generated, the field of Artificial Intelligence (AI) is no longer in "AI Winter". It has made remarkable, groundbreaking achievements. Using bio-inspired techniques and deep neural networks, scientists and researchers have managed to

build and train algorithms to reach, and even exceed average human performance in many fields, including categorizing medical images [5], planning flight schedules [6], and playing the ancient Chinese board game, *Go*. Unlike Deep Blue (the machine that defeated world chess champion, Garry Kasparov, in 1997) [7], AlphaGo developed by Google Deep Mind is not composed of hand-crafted, task specific logics, but a learning algorithm that learns from a massive amount of *Go* game history data played by human, and playing against itself millions of times [8].

More broadly, Machine Learning techniques were applied in fields such as scientific data analytics, financial trend prediction [9], image recognition [10], drive-less vehicles [11], and commercial product recommendations [12]. Machine Learning is considered as one of the most promising and useful techniques for automation and productivity in our contemparary society.

In this thesis, three aspects of the strength of AI were explored to augment the ability to make better urban decisions:

1) Machine Learning for enabling real-time urban simulation prediction in multiple aspects;
2) Optimization search algorithms for enabling AI suggestions for urban decision-making process;
3) Natural-language voice interface for enhancing the public engagment process and understanding of the urban knowledge.

1.4 Goal of the Design

To address the urban challenges mentioned above and to utilize the advanced AI techniques, *CityMatrix* was designed and investigated. *CityMatrix* is an urban decision support system with a tangible user interface, real-time feedback of multiple urban simulations, optimized suggestions, and a natural-language guide. The design attempts to facilitate public engagement with multiple parties of stakeholders, both professionals and non-professionals. With *CityMatrix*, the stakeholders can explore rapidly and collaboratively the urban configurations without prior knowledge.

CityMatrix uses Machine Learning algorithms to predict multiple sophisticated urban simulations. It is a versatile, quick, accurate, and low-cost method compare to traditional options. The real-time feedbacks, including the urban performance heat-map and radar-chart, help users understand better the consequences of their decisions.

CityMatrix uses optimization search algorithms to provide AI suggestions. The suggestions help users achieve their goal more efficiently, and understand the inherent trade-offs of different aspects of the urban performance.

Collectively, the goal of the design of *CityMatrix* is to enable a more evidence-based and democratic decision-making process.

Chapter 2

Related Work

2.1 Urban Simulation with Tangible User Interface

Prior works in Tangible Media Group and Changing Places Group in the MIT Media Lab were explored using tangible user interface (TUI) to improve the accessibility and efficiency in urban planning. The "Urp" project (1996-2001) [13] from Tangible Media Group provided users digital shadow and wind simulations in an urban environment. When the users manipulated the buildings (physical objects with optical tags), the shadow and wind simulations updated accordingly (Figure 3).

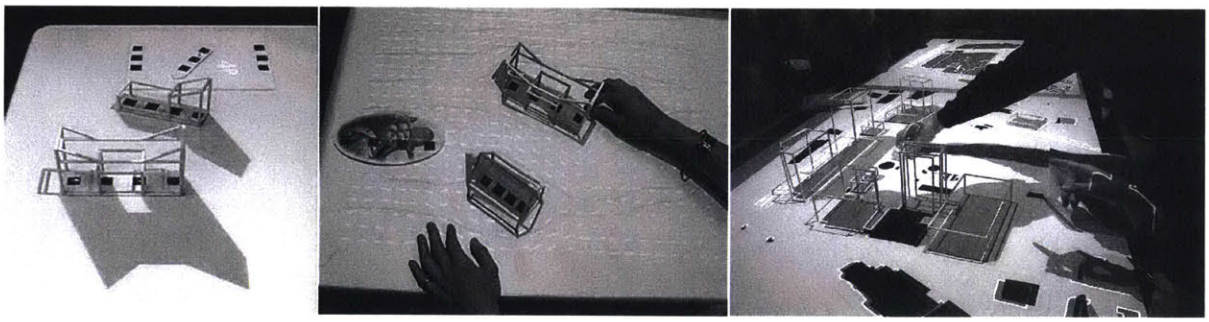


Figure 3: "Urp" project (1996-2001) is a TUI system with digital shadow (left) and wind (center) simulation in an urban environment (right).

"Unbuilt Ruins" (1999) [14] was an interactive exhibition which visualized the unbuilt architecture masterpiece designed by a famous architect, Louis I. Kahn, who died in 1974. Kahn's unexecuted design was originally modeled and visualized with computer graphic by Kent Larson with representations of geometry, material, light, and details [15]. In this exhibition, the Kahn's unbuilt spaces were accessible intuitively and interactively via a TUI. When the user put a RFID-tagged physical icon in the center of the table to select one architecture project, the table would be projected with the architectural plan of that project. Then the user would put a cylinder on the plan as a representation of desired viewpoint. The cylinder's location and orientation were tracked by a computer vision system. Then the four vertical screens around the table would display the visualization of the space accordingly (Figure 4). The TUI made the navigation of the projects and spaces much more intuitive and accessible for non-professionals.

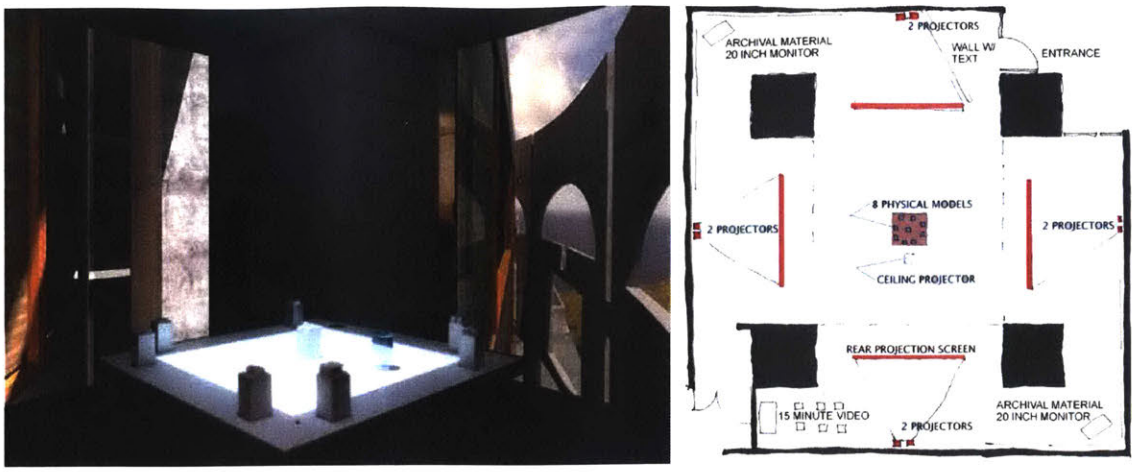


Figure 4: Unbuilt Ruins" (1999), an interactive exhibition which visualized the unbuilt architecture masterpiece designed by Louis I. Kahn. Left: setup of the TUI able and vertical display; Right: Floor plan of the exhibition.

The Riyadh Project (2014-2016) [16] from Changing Places Group visualized the walkable access intuitively on a city model made out of Lego bricks (Figure 5). Research by Tarfah Alrashed [17] has proven that compared to graphic user interface (GUI), the TUI system better promotes collaboration among a group of users.

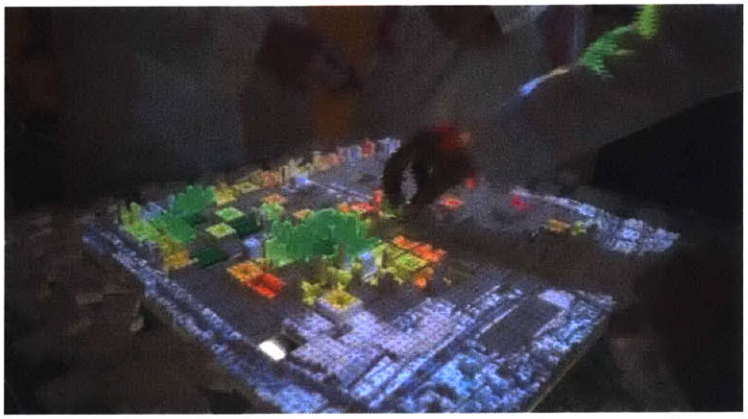


Figure 5: Riyadh Project: when users change the city configuration by moving the building components represented with Lego bricks, a heat-map visualization of walkability will be projected on the city model, where green indicates good walkability, and red indicates bad walkability.

Urban design interface with a proper level of abstraction is critical for accessibility of the urban decision-making process. Research by Tarfah Alrashed also proved that real-time feedback was crucial for collaboration [17]. However, the research mentioned here has one limitation; they can only handle one simple simulation in real-time. To achieve a complex simulation in real-time, traditional approaches includes:

1) Decreasing the simulation resolution to reduce computational requirements. This is not always a viable option because there is normally a threshold resolution, below which the simulation result is more random than meaningful. Decreasing the simulation resolution is not an adaptive approach, because each simulation

defines resolution differently, thus the impact of the computational requirement reduction.

2) Parallelization of computation. Not all the urban simulations can be parallelized. For example, in an Agent Based Simulation, if there is interaction between the agents, the simulation cannot be parallelized. Furthermore, the development of parallelization is unique for each simulation, making this method non-versatile.
3) Increasing the computational capacity. Increasing the computational capacity involves increasing investment of the hardware exponentially. There is always a physical limitation of the computational capacity using one computer.

In this thesis, a versatile, quick, accurate, and low-cost approach, which is using Machine Learning simulation prediction, was proposed and investigated to enable real-time feedback of multiple urban simulations.

2.2 Urban Performance Indexes

In *Live and Death of the Great American Cities* [18], Jane Jacobs' theory revealed the correlation between the urban form and the urban performance – like quality of life, vibrancy, and safety. The on-going work, "The math behind a high-performance, livable and entrepreneurial city", by Luis Alonso, Kent Larson, me and other researchers in Changing Place group provided a great insight to quantify the correlation between the urban form and multiple aspects the urban performance (Figure 6).

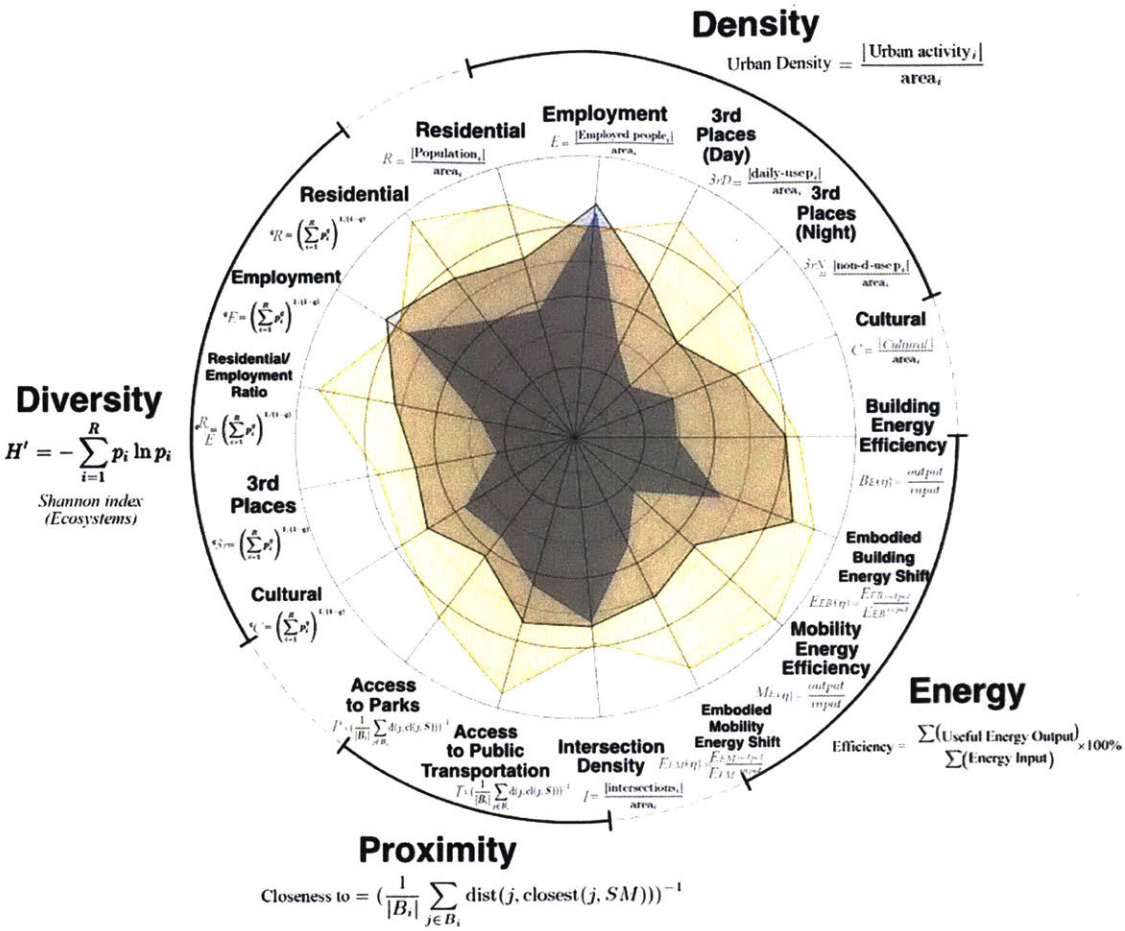


Figure 6: Urban performance radar-chat with multiple indexes represents the quantification of the correlation between urban form and urban performance.

In Figure 6, the radar-chart with 17 indexes represents 17 aspects of the urban performance of a city district. They were grouped into four high-level indexes: Density, Diversity, Proximity, and Energy. Each index was calculated separately according to the formula beneath its title. The formulas were defined by relevant research in literature. For example, Shannon index [19] was used as the formula to calculate the Diversity index. Shannon index was used to describe the diversity of an ecosystem [20], to which the diversity of a city has a high similarity. The Density, Diversity, and Energy indexes were incorporated into the *CityMatrix* system.

Chapter 3

Design of *CityMatrix*

3.1 Goal of the Design

The user interface (UI) of *CityMatrix* was designed to eliminate the necessity of prior knowledge and to provide a smooth learning curve when using the system. This chapter introduces the main functionalities and the user interface design of *CityMatrix*, followed by the hardware and software construction of the user interface.

3.2 Tangible Input for Multiple Users

CityMatrix takes user input through a Tangible User Interface (TUI), a table with optically tagged building modules (Lego bricks), and allows multiple users to build a city district with these modules collaboratively. The user could add, remove, or exchange any of the building modules to design a city district (Video 1).

The scale of this model is 1:762, meaning one Lego brick represents 26.7 x 26.7 meters. A 5 x 5 Lego brick area of *CityMatrix* represents 133.3 x 133.3 meters, which is very similar to the scale of typical urban block in Eixample district of Barcelona (Figure 7). It was decided to use this scale to achieve an appropriate level of abstraction. The benefit of abstraction was to avoid being overwhelmed with information so that the users could focus on the typological relationship of the city rather than the form and detail.

Figure 7: Typical urban block in Eixample district of Barcelona.

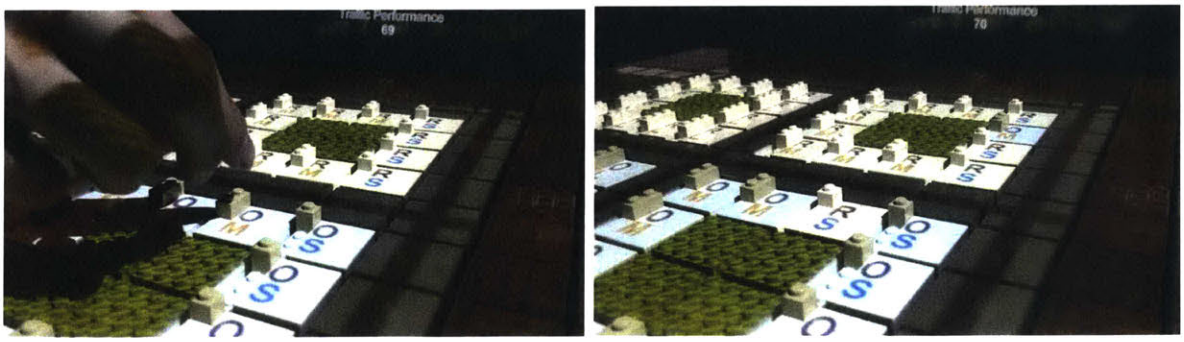
Video 1: TUI Table with Lego brick modules, projections, sliders, toggles, and a vertical display screen. (link: http://yanzhangworks.com/cmv1)

3.3 Inputs

3.3.1 Land-use

The setup of *CityMatrix* allows the users to change the layout of land-use by moving optical-tagged modules. Each cell can be changed into six types of buildings, roads, and courtyards. The users can add, remove or exchange the existing bricks, or pick from an external library of preassembled Lego brick modules (Video 2).

The six building types are Small Residential Unit (RS), Medium Residential Unit (RM), Large Residential Unit (RL), Small Office Unit (OS), Medium Office Unit (OM), and Large Office Unit (OL). These six types were decided to be the minimum necessary urban components for two reasons: 1) Living (residential) and working (office) are two main types of land-use in contemporary cities; 2) Small, medium, large unit types will attract people of different ages and income. Multiple aspects of the urban performance can be calculated according to the configurations of these six building types. Other building types might increase the difficulty in making choices for non-experts.



Video 2: Users can change the land-use layout by adding, reducing, and exchanging optically tagged Lego bricks. The system will update the projection of the land-use tags as soon as there is new user input. (link: http://yanzhangworks.com/cmv2)

3.3.2 Urban Density

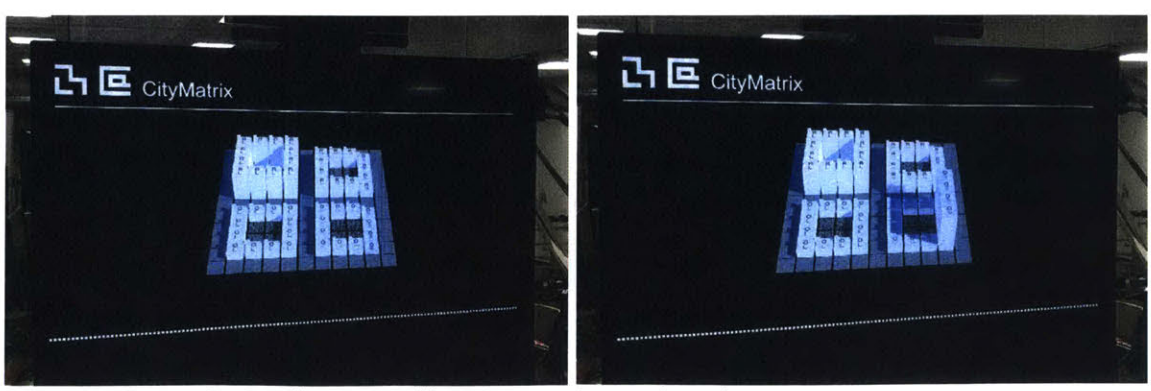
The slider and selection dock on the side allows users to change urban density. Users could put one of the six building types into the selection dock (lower right) to select all the same types of buildings in the city district. They could then slide up or down the building height slider (upper center) to manipulate the building height of these buildings, changing urban density (Video 3).

Video 3: Users can change building heights, thus, urban density by manipulating the building type selection dock (lower right) and the building height slider (upper center). (link: http://yanzhangworks.com/cmv3)

3.4 Outputs

3.4.1 3D City Display

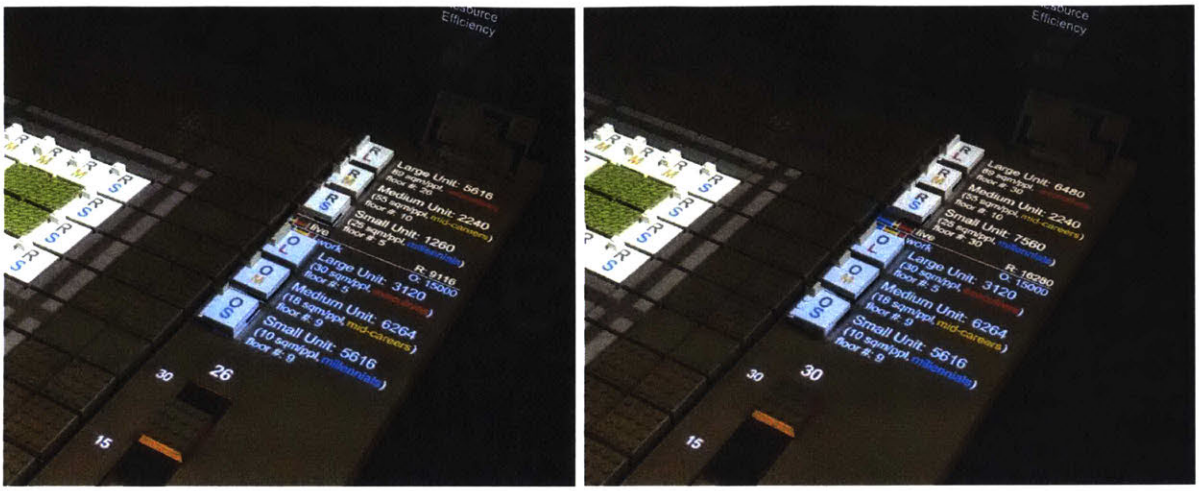
On the upper part of the vertical display screen, a three-dimensional representation of the city district they built is provided to inform the users the height of the buildings in the city district. This part of the UI complements the missing building height information of the two-dimensional projection mapping on the flat Lego bricks (Video 4).



Video 4: 3D city display (link: http://yanzhangworks.com/cmv4)

3.4.2 Population Statistics

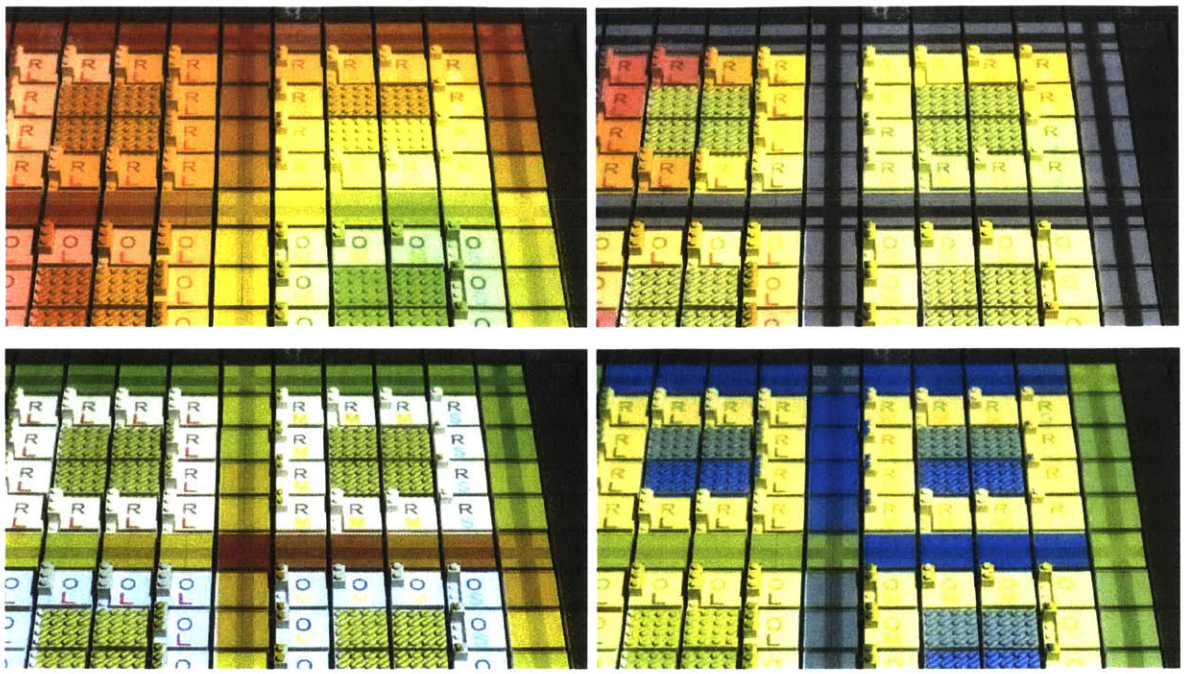
On the upper-right corner of the table, a summary of the urban density statistics is provided. It represents the individual and total population attracted by different building types in the form of numbers and a bar chart. The statistics change according to the inputs by the users (Video 5).



Video 5: Urban density statistics. (link: http://yanzhangworks.com/cmv5)

3.4.3 City Performance Heat-maps

Users can select one of the five heat-maps of a corresponding aspect of the urban performance by manipulating the heat-map slider on lower right side of the TUI table. The five heat-maps are Population Density, Experience Diversity, Energy and Cost Efficiency, Traffic Performance, and Solar Access Performance (Video 6). When the users rearrange the layout of the city district, *CityMatrix* updates the heat-map accordingly in real-time.



Video 6: Urban performance heat-maps. Upper left: Population Density. Upper right: Experience Diversity. Lower left: Traffic Performance. Lower right: Solar Access Performance.
(link: http://yanzhangworks.com/cmv6)

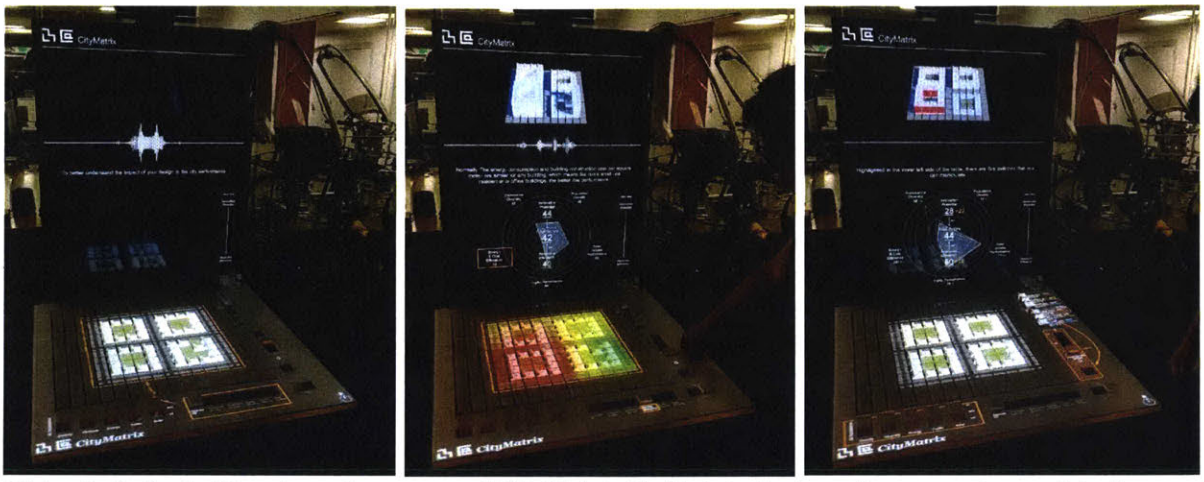
3.4.4 City Performance Radar-chart

On the lower part of the vertical display screen, a radar-chart with indexes is provided to inform the users the city district's aggregated urban performance of the same five aspects as for heat-maps. Additionally, two high-level indexes, Innovation Potential and Resource Efficiency, are provided. Population Density and Experience Diversity contribute to Innovation Potential index. Resource Efficiency index is defined by Energy and Cost Efficiency, Traffic Performance, and Solar Access Performance. All indexes are updated in real-time according to the users' inputs (Video 7).

Video 7: Radar-chart of Urban Performance Indexes. (link: http://yanzhangworks.com/cmv7)

3.4.5 *CityMatrix* Guide

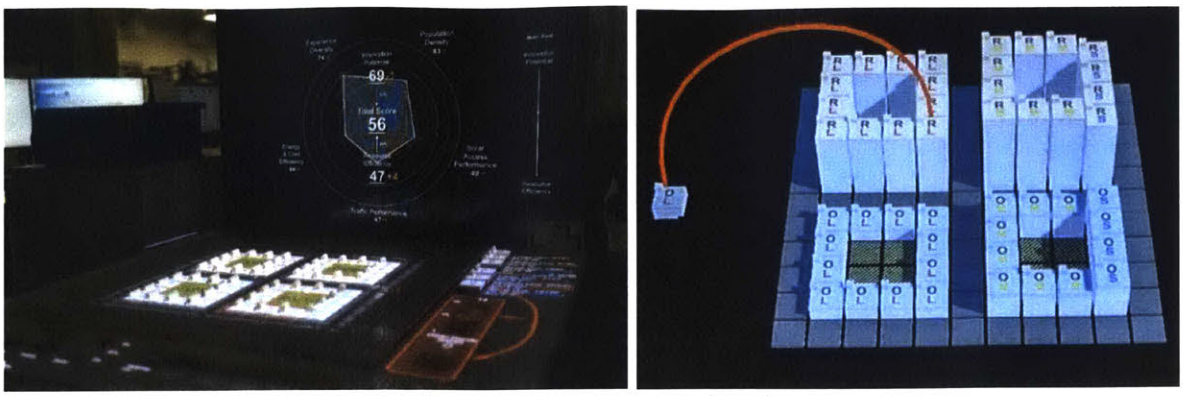
The *CityMatrix* Guide is designed as a natural-language voice interface to introduce the complete functionalities of *CityMatrix* to first time users. It speaks the explanation in natural language to help users understand how to use *CityMatrix*. In the center of the vertical display, there is a sound wave visualization and written text to indicate what the *CityMatrix* Guide is speaking. *CityMatrix* also highlights and annotates the relevant UI components with orange markers and arrows to help users focus on the correct component when it is being explained (Video 8). The video recording and text script of the complete process of *CityMatrix* Guide introduction is attached (Appendix A: *CityMatrix* Guide).



Video 8: A short video shows the process of CityMatrix Guide introducing urban concept, user interface, and AI suggestion functionalities. (link: http://yanzhangworks.com/cmv8)

3.4.6 AI Suggestion

When the AI Suggestion function is on, *CityMatrix* provides users specific suggestions of optimal moves. The orange annotation on TUI table suggests which move to take next to achieve improved the scores the most. The orange line in the radar-chart indicates the projected urban performance if its suggestion was accepted. The orange text beside each score illustrates the exact expected score changes. Users can manipulate the five switches, called AI Assistance Weights, located on the lower-left corner of the TUI table. By operating these switches, users can define the priority of the assistance from AI suggestions (Video 9).



Video 9: AI Suggestion on TUI table (left) and on 3D city view (right).
(link: http://yanzhangworks.com/cmv9)

3.5 Hardware

CityMatrix was composed of the following components:

1) A modular prefabricated table (Figure 8) developed by Karthik Patanjali in the Changing Places group at MIT Media Lab. It was used to facilitate the TUI of many urban decision support projects in the research group.
2) Pre-defined Lego bricks, of which top is flat, and bottom is an optical tag composed of four colored Lego pieces (Figure 9 left).
3) A diffused surface LED evenly lights up the optical tags.
4) A webcam at a resolution of 1080p, which capture video feed for a computer vision algorithm to decode the optical tags at the bottom of all Lego bricks (Figure 9 left).
5) A portable computer that runs all the related software.
6) A high definition, high brightness projector, which projects two-dimensional representation of the city onto the Lego bricks and the TUI table.
7) A vertical screen for displaying radar-chart, the three-dimensional representation of the city, and the sound wave and texts of the *CityMatrix* Guide.

Right side of Figure 9 shows the *CityMatrix* labeled with the main components.

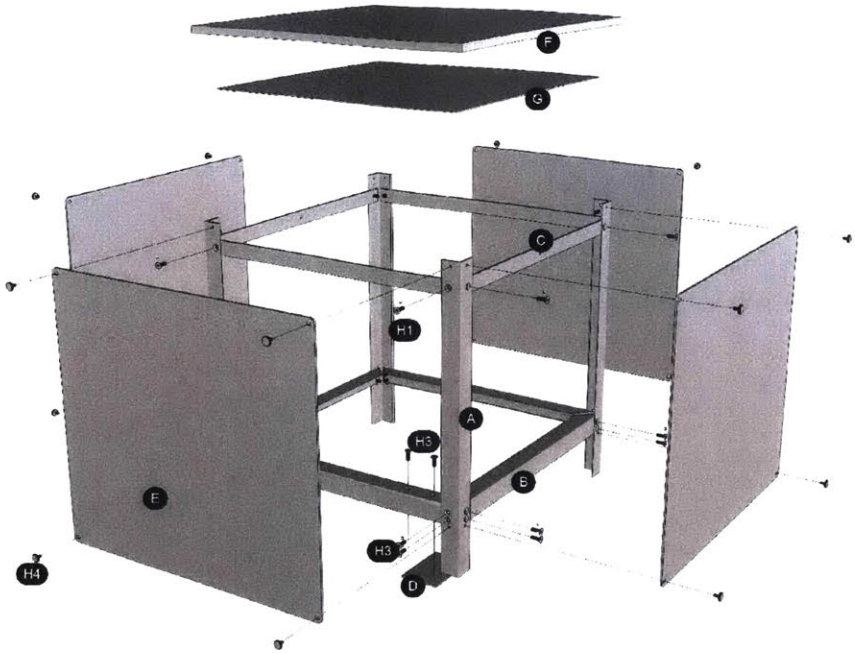


Figure 8: CityMatrix table hardware diagram.

Figure 9: Left: A view of the optical-tagged Lego bricks from beneath the table. Right: the CityMatrix *labeled with the main components.*

3.6 Software

Following is a list of software used in *CityMatrix* project:

1) **Rhinoceros 5 and Grasshopper**: for computer vision optical-tag decoding, visualization of 2D projection mapping on TUI table.
2) **Unity 5**: for visualization on the vertical display including urban performance radar-chart, 3D city display, and sound wave and text of *CityMatrix* Guide.
3) **Python 3**: for calculation of Machine Learning prediction, AI suggestions, and data logging.

Chapter 4

Architecture of Artificial Intelligence for Urban Decision-Making

4.1 System Structure

The *CityMatrix* system architecture contains four main layers (Figure 10):

1) **User Input**: The system collects the user input through reading the optically tagged Lego bricks, decoding them into indexes of building types, packaging these indexes into an "input city" object in JSON format, sending it to the next layer via UDP connection.
2) **Urban Performance Evaluation**: The Python server takes in the "input city" JSON object, and runs calculations to evaluate the five aspects of city performance, including 1) Population Density, 2) Experience Diversity, 3) Energy and Construction Efficiency, 4) Traffic Performance, and 5) Solar Access Performance. The evaluations are both seen on a local scale (heat-map) and an aggregated scale (radar chart scores). Among these five aspects of evaluation, Density, Diversity, and Energy can be calculated directly within one-half second, while the other two, Solar and Traffic, take significantly longer to simulate (approximately one minute for each). The real-time evaluation of these two aspects needs to be replaced by a much faster approach, a pre-trained Machine Learning prediction of the simulation. With this approach, evaluation of both Traffic and Solar performance could be calculated within ten milliseconds.
3) **AI suggestion**: The Python server takes the quick Machine Learning prediction as an evaluator. It searches and evaluates a large number of possible options of the next move and provides users with an optimal option, according to the weights of the five aspects of urban performance chosen by the users.
4) **Visualization Output**: The feedback and AI suggestions are visualized on both a TUI table via projection mapping and a screen display. These visual outputs will give the users key information to support their decision for the next step. When the users choose a new move on the city configuration, the system will take this new input and start the entire process again from the first layer.

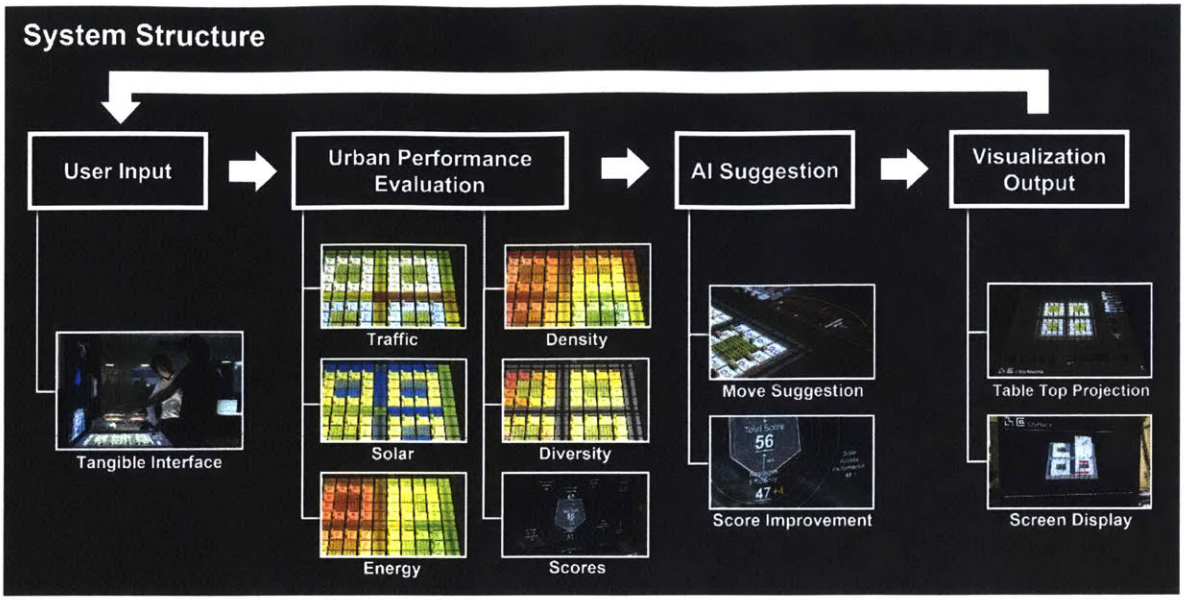


Figure 10: System structure – the workflow of CityMatrix could be understood as four main layers: User Input, Urban Performance Evaluation, AI Suggestion, and Visualization Output. Each new move made by the users triggers a full procedure loop of these four layers.

4.2 Real-time Simulation Prediction Using Machine Learning

4.2.1 Overview

Few intensive urban simulation systems can operate in real-time. The options of using traditional methods to calculate a complex urban simulation in real-time fashion are very limited. Up to now, there has not yet been a demand to enable these professional urban simulations to provide real-time feedback. However, having real-time feedback of an urban evaluation of multiple aspects is the key to the collaborative public engagement. **In this thesis, a versatile, quick, accurate, and low-cost approach to enable real-time feedback on multiple sophisticated urban simulations was proposed and tested.** The new approach is to train a Machine Learning (ML) algorithm to learn from an urban performance simulation, and predict the result in real-time.

There are many advantages of this new approach:

1) **Versatility**: Cities are complex, but not random, at an aggregated level. Theoretically, as long as there is any pattern in the simulation, an appropriate ML algorithm should be able to learn from it. In this work, ML algorithms were trained to predict both solar radiation and traffic simulations with great precision. This result indicated that different types of simulation could be learned in the future.

2) **Speed**: On the average, it takes the ML algorithm less than ten milliseconds to predict both solar and traffic simulation result, whereas the original simulation took about two minutes. Depending on the complexity of the simulation, it took some time (less than one day) to generate training data and to train ML algorithm. After that, users can use the trained the ML predictor to evaluate as many city configurations as they want with virtually no costs for time.
3) **Accuracy**: Achievement of the coefficient of determination (R^2) of above 0.82 (0.57 for LR) for both solar and traffic simulation was achieved. Because the feedback was provided in the form of heat-maps and aggregated scores on radar-chart to support decision-making, it was more important to control the visual differences of the heat-maps between real simulation results and the ML prediction ones. These differences were proved small enough so that a user's decision was not affected. The average percentage-residuals of the aggregated score of the solar and traffic simulations are both below 7.1%.
4) **Development Cost**: The procedure of generating simulation data and training ML algorithm could be automated. Whereas, to speed up the feedback with traditional methods, different simulation types required different strategies, therefore, increased the costs because additional development was needed.

4.2.2 Procedure

The procedure of implementing the Machine Learning real-time simulation prediction is as follows: 1) Define or construct an urban simulation engine. 2) Generate a significant number (10,000) of city configurations and simulate each of them. 3) Take the simulation results as training data to train an appropriate Machine Learning algorithm. 4) Use trained ML algorithm to predict the simulation results according to the new user input (Figure 11).

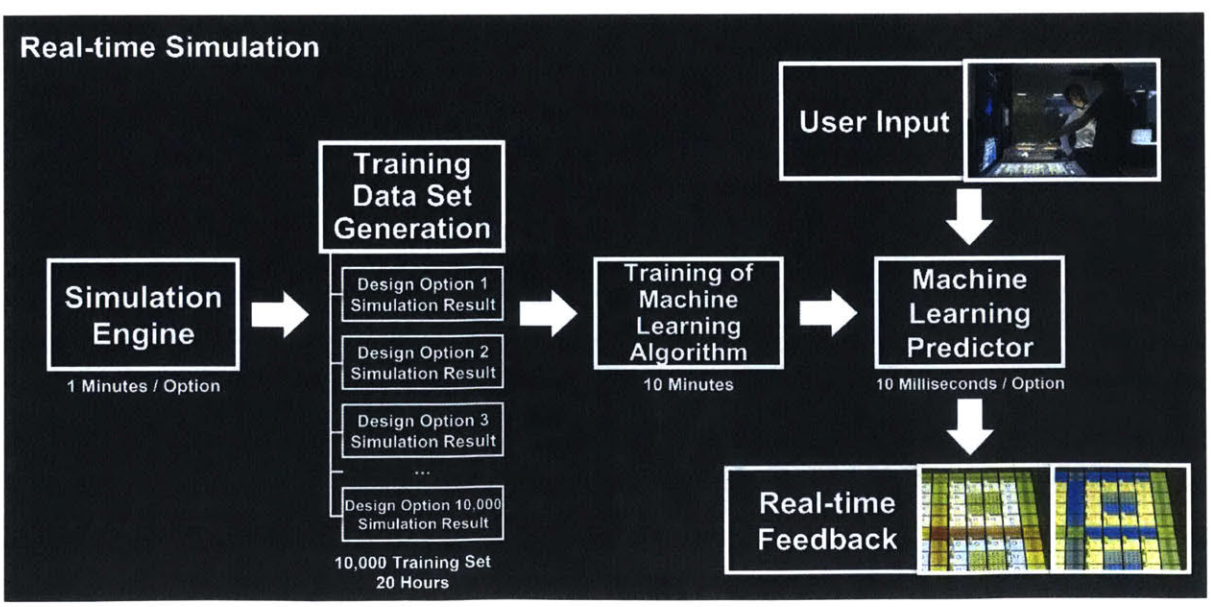


Figure 11: The four-step procedure of real-time simulation prediction using Machine Learning with the time needed for each step. Label lower right two pics.

As shown in Figure 11, the entire procedure to get the Machine Learning predictor was relatively long (20 hours and 10 minutes). However, when the predictor was trained, it was utilized on a large number of new city configurations with little time costs (ten milliseconds per city configuration). The purpose of this procedure was to shift the time for calculations before the use of the platform.

The main goal of this work was to explore and examine the feasibility and the effectiveness of the methodological approach, rather than perfect any aspect of urban simulations. The simulations conducted were rather abstract and limited in scope. Since this method was meant to be an open, adaptive approach, in future works, any of the simulations could be replaced or enhanced by experts' works that are more sophisticated.

4.2.3 Solar Access Simulation & Training Data Generation

The goal of the solar access simulation was to evaluate how much annual sun light could be obtained in the city. Assuming the rooftop of each building in the city would fully consist of solar panels, the simulation would be able to calculate the solar access of all the rooftops, plaza, or courtyard spaces.

A commercially available solar radiation simulation software, DIVA [21], as the solar access simulation engine was used. The original plan was to generate 10,000 city configurations (16 x 16 city-grid), run DIVA simulation on each, and use them as training data set. Through testing and observation, in the geolocation of Boston, a building with a height less than 80 meters had little effect on solar radiation on another location 120 meters (2 grid cells long) away. This meant the shadows did not reach that far during the daytime of a year. Instead of simulating the whole table (16 x 16 grid), the simulation of the impact of one building on a 5 x 5 grid was conducted twice, either with or without the center building. The two results were subtracted from each other. Instead of training the Machine Learning algorithm to predict the entire table of a 16 x 16 grid, predictions of the impact could be made on a 5 x 5 grid (Figure 12). The predicted impact was then added the existing condition of the entire table. Both the complexity of the prediction problem and the amount of simulation training data were drastically reduced.

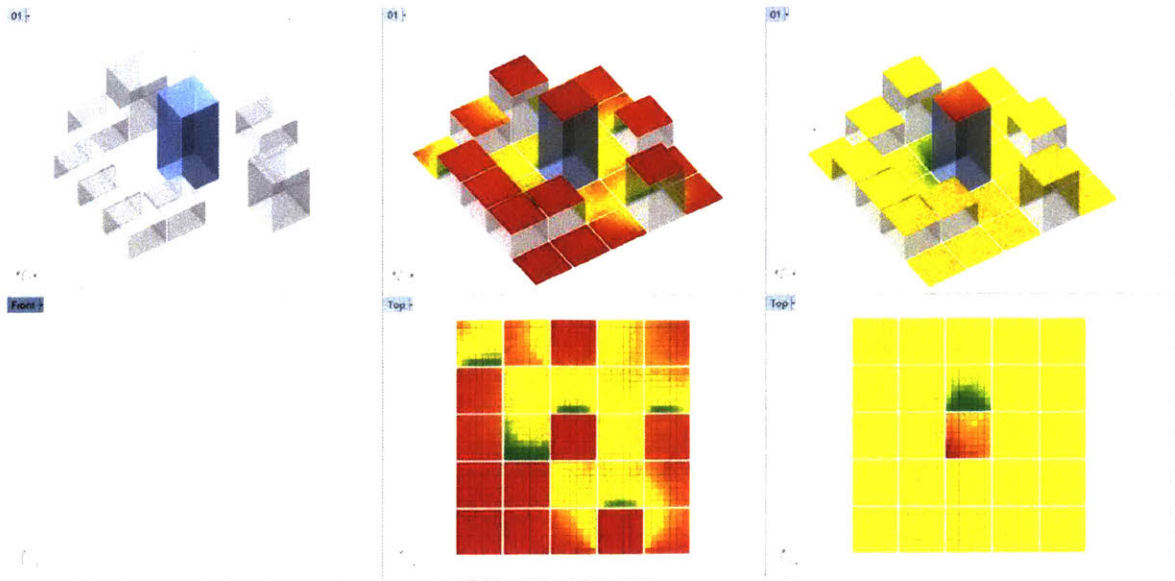


Figure 12: Solar radiation simulation with DIVA. Upper: 3D views; Lower: 2D top views. Left, a random configuration of the 5 x 5 city grid; Center: solar radiation simulation result with center building (Red: more solar radiation; Green: less solar radiation); Right: difference of two simulation results - with and without the center building (the solar radiation impact of adding the center building).

An automation script generated 10,000 simulated results of the solar radiation impact. Results of each of the simulations took about one minute and 20 hours to generate all the simulation results.

4.2.4 Solar Access Simulation Machine Learning Prediction

The 10,000 solar radiation simulation results were then used as the training data set to train a Machine Learning algorithm. Using Python Machine Learning library, Scikit-learn [22], three ML algorithms were tested: Linear Regression [23], K Nearest Neighbor [24], and Random Forest [25]. All worked to produce promising visual results (Figure 13).

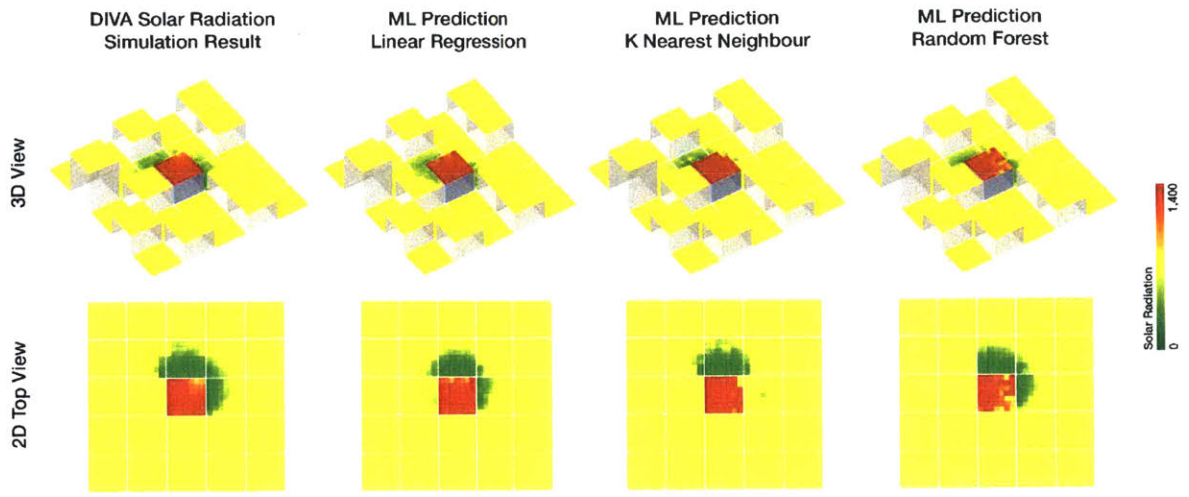


Figure 13: Comparison of the solar radiation simulation result (a) and three ML prediction results (b, c, d) in both the 3D view (upper) and the 2D top view (lower).

Each ML algorithm's prediction had unique characteristics. As shown in Figure 13, the prediction result of the Linear Regression algorithm (b) was smooth, similar to the original simulation result (d). However, there was a minor artifact that the center building casted shadows on the rooftop of the building on its left side (b). Whereas, in the results of the real simulation, there was no shadow on that building because the adjacent building was the same height. Linear Regression in principle works as an average out of all training data. It was not capable to discriminate the conditional changes. The prediction results of K Nearest Neighbor (c) and Random Forest (d) were less smooth visually than the Linear Regression (b). They did predict correctly the conditional changes such as the impact on the adjacent building with the same height (c and d). In future research, there is a need explore a way to combine the strengths of both.

Finally, a script was written to apply the building change impact (5 x 5 grid) to existing solar radiation result (16 x 16 grid) and to keep accumulating the new changes. The final R^2 for the whole table was 0.825 using the Linear Regression.

4.2.5 Traffic Simulation & Training Data Generation

Similar to the solar access simulation prediction, in order to train a real-time Machine Learning prediction model for traffic simulation, a training data set that would contain about 10,000 instances of traffic simulation results was necessary to reach enough prediction precision. However, not like the solar access simulation, the traffic simulation contained much more rules and parameters. Therefore, there was a need to construct the traffic simulation in a flexible programming environment rather than using an existing simulation engine.

This session briefly describes the construction process of the traffic simulation and generation of training data.

Inspired by the Persuasive Electric Vehicle (PEV) project (2013-present) by Michael Lin from Changing Places group at the MIT Media Lab, the goal of the traffic simulation was to model the traffic volume and user wait-time in a city district, assuming all trips were performed by a highly efficient autonomous shared-mobility system composed of a fleet of PEVs. The input of the simulation was a 16 x 16 matrix of the city grid. Each cell in city grid could be either a building with a specific number of people living or working in it (building population), an empty lot, or a segment of a road. The building population drove the generation of trip requests during a one-day cycle in the simulation. Similar to a shared-mobility system such as Uber, the nearest available PEV would pick up the person requesting transportation, travel, and drop him or her off at the required destination. The system logged the accumulated traffic volume and user wait-time on each road cell for the period of a one-day cycle in the simulation. The output of this traffic simulation again was a 16 x 16 matrix of which each road cell contained both value of the traffic volume and wait-time.

The earliest traffic simulation model was implemented in Processing 3.0 [26], an object-oriented programming language built for visual artists and designers. Processing provided custom control over the visual representation of the map and the agents (PEVs in this case) (Figure 14). However, because there were no pre-defined libraries for Agent Based Simulation (ABS), the entire simulated architecture including the road graph, agent object, agent behavior, scheduling, and path-finding algorithm had to be built from the ground up. It became more challenging as complexity was added to the simulation. For instance, either enabling one-way road segments or introducing different vehicle speeds required reprogramming the road graph structure.

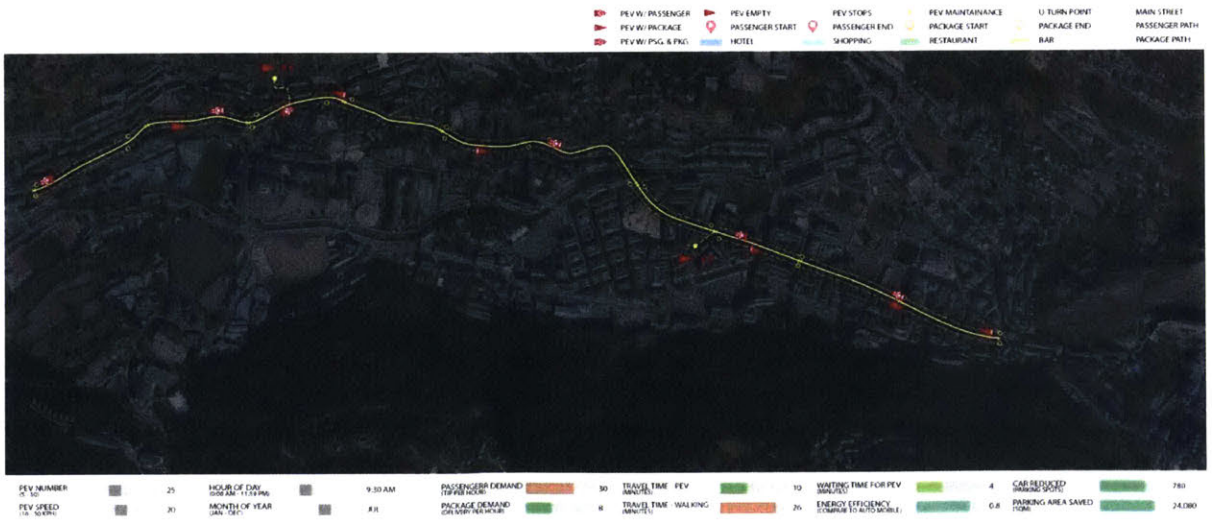


Figure 14: Traffic simulation implemented in Processing 3.0 with the benefit of great custom control over the visual representation of the map and the PEVs. The red triangles represent PEVs in different states: Empty, In-Route, and Delivering.

After ten months of simulation development, it was determined that Processing was not suitable for the ABS for this work. After considering several other platform options, the

CityMatrix team decided to move to GAMA platform [27] [28]. GAMA is a modeling and simulation-development environment for building spatially explicit agent-based simulations. It is a multiple-application domain platform using a high-level and intuitive agent-based language. With GAMA, users can undertake most of the activities related to modeling, visualizing, and exploring of the simulations using dedicated tools [29]. Taking advantage of many of the features offered by GAMA, the task of rewriting the existing simulation, written in Processing, to GAMA platform took less than three weeks. GAMA came preloaded with uniform cost search path-finding algorithms, allowing abstraction of custom path-finding logic that created problems in Processing. With the new GAMA model, a single traffic simulation for a day's cycle could be completed as quickly as 40 to 60 seconds, depending on the population and the CPU configuration.

The next step was to randomly generate 10,000 different city configurations. A script, *CityGenerator*, was developed. It generated cities with a fully-connected road network in a 16 x 16 city-grid, with six basic types of buildings. *CityGenerator* allowed the control of several tunable parameters, such as road density and maximum population. *CityGenerator* is fast and flexible. It can generate 10,000 city configurations in under 10 seconds (Figure 15).

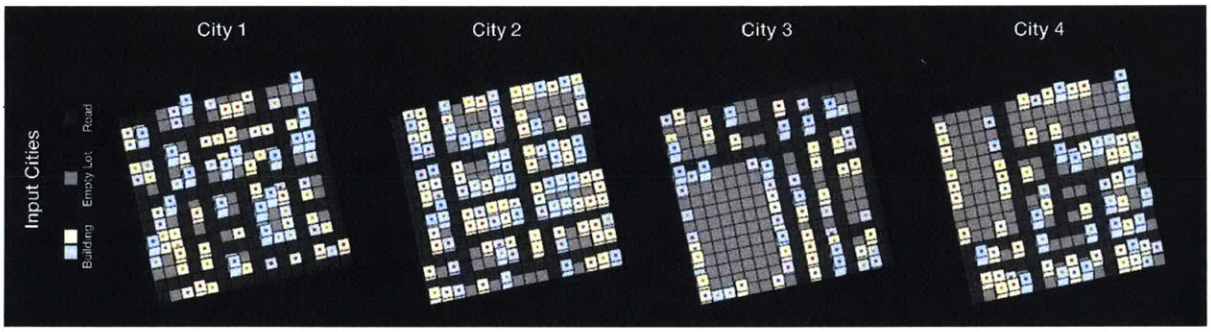


Figure 15: Four out of ten thousand city configurations generated by CityGenerator script. 10,000 different city configurations were generated as the input of our GAMA traffic simulation model. The height of a building represents the population of people living or working in the building. These four example city configuration demonstrate the variety in building height, road density and road distribution.

A batch process in GAMA was developed. This took multiple input city configuration JSON files, ran a traffic simulation model one by one, and provided JSON city files for each road cell. The JSON object contained simulation results of both traffic volume and wait time. As mentioned earlier, the simulation of one city configuration took about less than one minutes. All simulation tasks were distributed to eight computers, so that completion the 10,000 simulations took 18 hours (Figure 16).

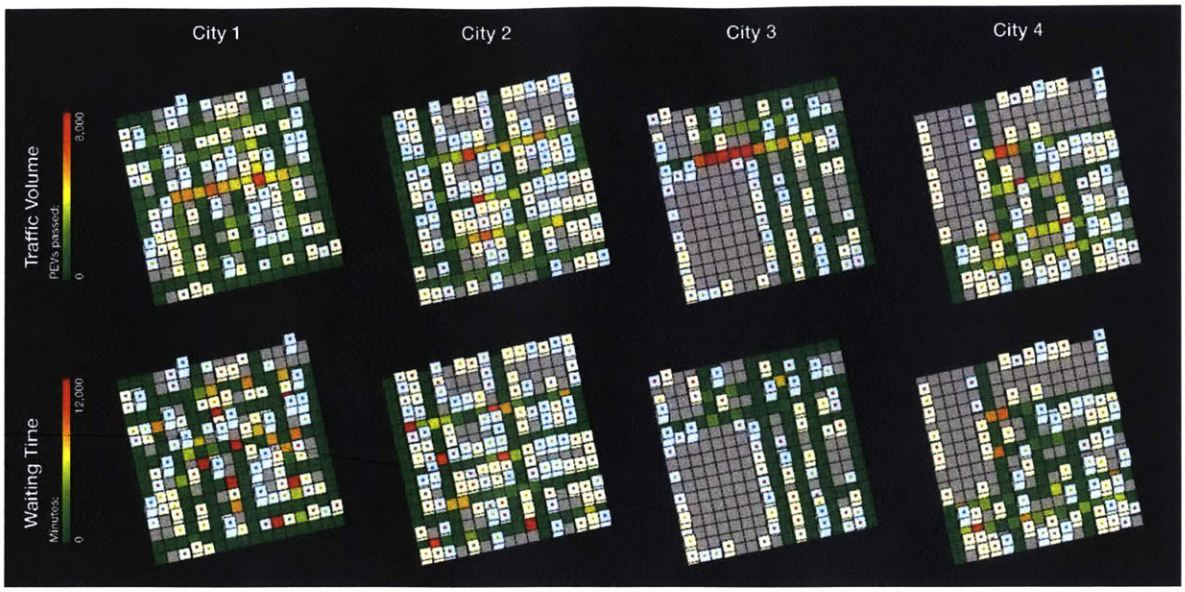


Figure 16: Traffic simulation results of four out of ten thousand city configurations. Upper: city configurations with traffic volume results on the road, represented as heat-map (red means more traffic volume); Lower: city configurations with wait-time results on the road, represented as heat-map (red means more accumulated wait-time).

4.2.6 Traffic Simulation Machine Learning Prediction

Once the 10,000 city configurations were simulated, 7,000 were used as a training set and 3,000 used as a test set. Multiple ML algorithms were compared in the process, including Linear Regression (implemented with Scikit-learn) and Convolutional Neural Network (CNN) [30] (implemented with Tensorflow [31] and Keras [32]). CNN provided the most significant R^2 and visual result. Different parameters and input feature combinations were tested for each ML algorithm to achieve the best R^2 and without overfitting.

With Linear Regression, the best R^2 achieved was 0.571. With CNN, the best R^2 achieved was 0.861. The architecture of the CNN contained three convolution layers (Figure 17). CNN was effective at this task because a heat-map by nature was an image. CNN was designed to learn the feature of images on multiple scales.

Figure 18 and Figure 19 are comparisons of the GAMA traffic simulation results (upper) and the Machine Learning predictions (lower) of both the traffic volume heat-map (Figure 18) and the wait-time heat-map (Figure 19) in four different cities. The prediction successfully kept the features of the simulation.

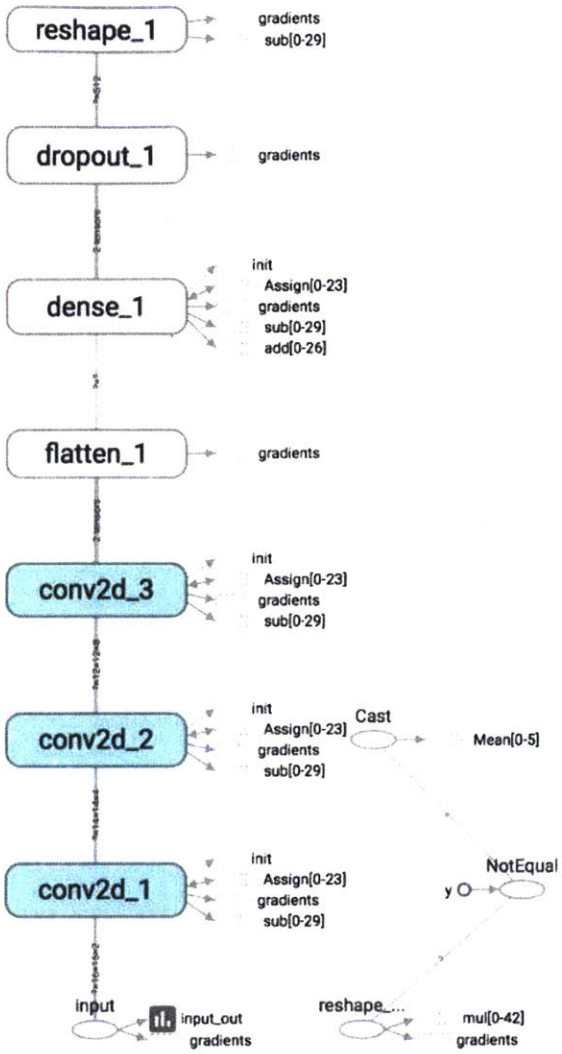


Figure 17: Architecture of the CNN predictor.

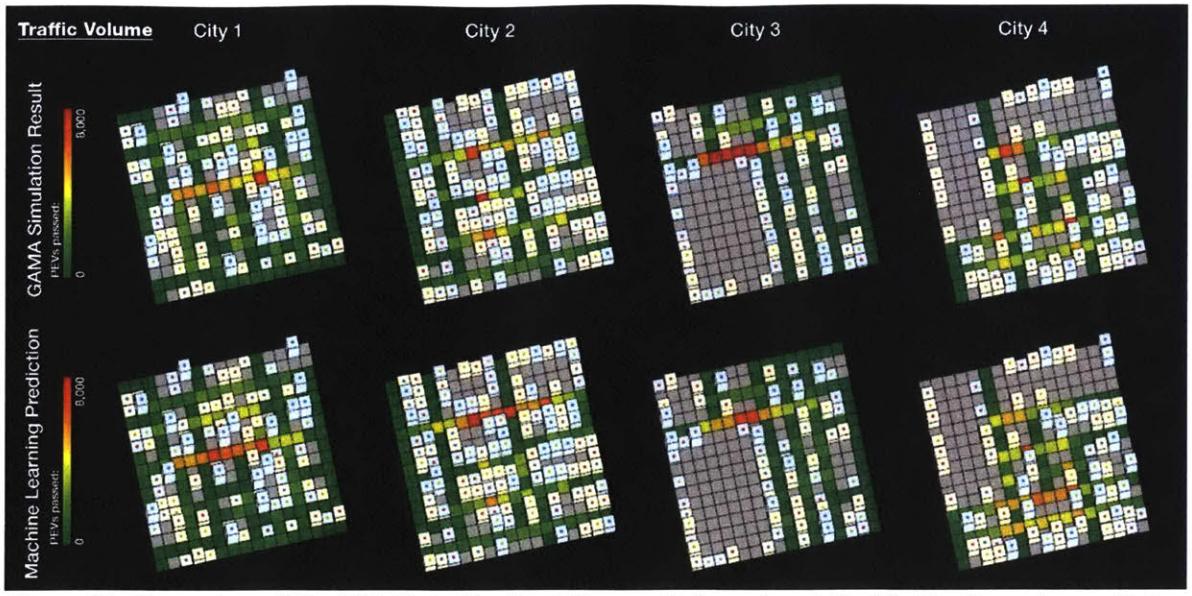


Figure 18: Comparison of the GAMA traffic simulation result (upper) and the Machine Learning prediction (lower) of traffic volume heat-map in four different cities. The prediction successfully kept the features of the simulation.

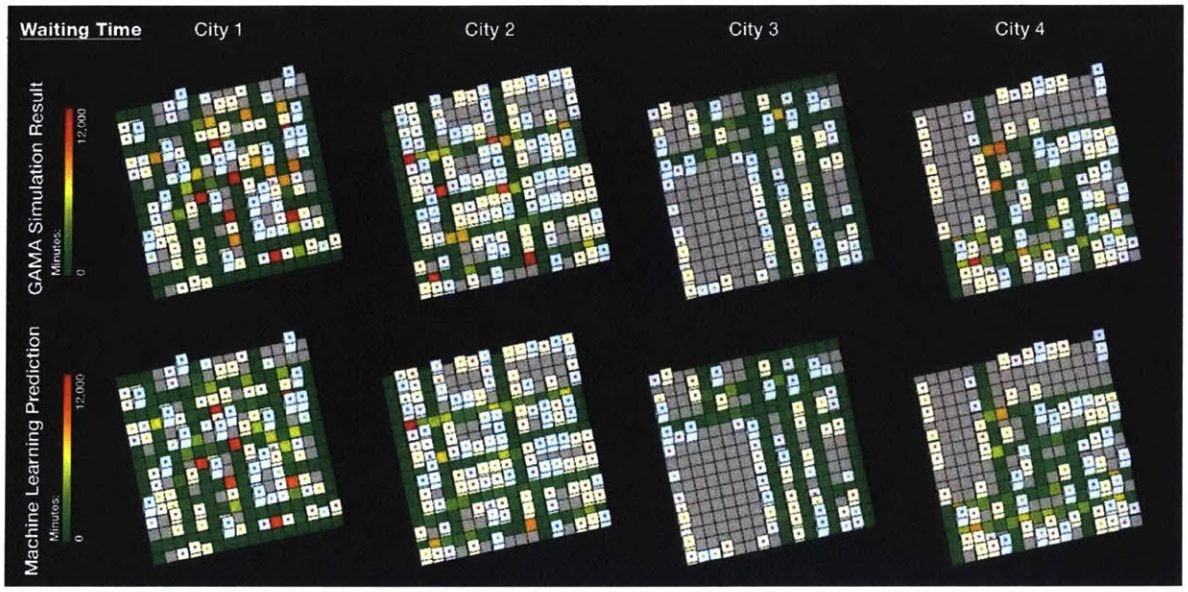


Figure 19: Comparison of the GAMA traffic simulation result (upper) and the Machine Learning prediction (lower) of wait-time heat-map in four different cities. The prediction successfully kept the features of the simulation.

4.3 Artificial Intelligence Suggestion

4.3.1 Overview

The purpose of creating the AI suggestion functionality for *CityMatrix* was to combine the strength of both human intelligence and machine intelligence in a decision-making process. Humans are skilled at subjective value judgments such as culture, aesthetics, and emotion, whereas machines are not. Machines are capable of handling complex numeric calculations with multiple objectives simultaneously with efficiency and precision where humans cannot (Figure 20). The AI system optimized the quantitative criteria chosen by the users, while the users focused more on values and qualitative issues. Bringing these two kinds of strengths together organically was the key to better decision-making. For this reason, instead of giving an optimal final solution of a city design, the AI should have given step-by-step open suggestions that the users could choose to accept or reject.

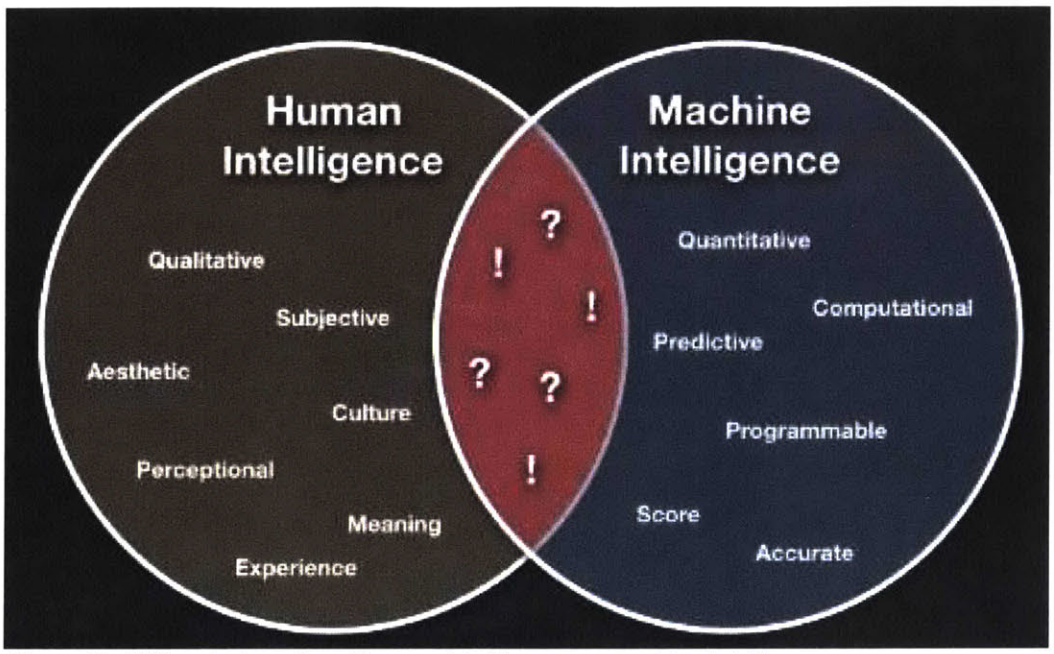


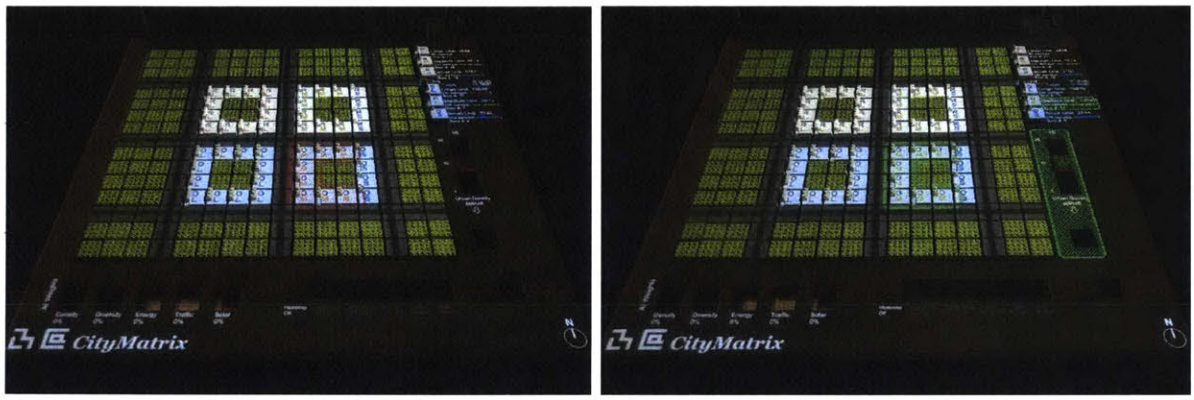
Figure 20: Different strengths of human intelligence vs. machine intelligence.

In this work, *CityMatrix* provided users a specific suggestion of the most efficient move based on the current city configuration at that moment. The users could have chosen to use these suggestions as a guide, accepting or rejecting them, considering the context or aspects of the city that was not addressed by the system.

4.3.3 Design of AI Suggestion Representation

From the perspective of user experience design, it was challenging to communicate to the users the AI suggestions because this added more information to the existing information that was already complicated. A problem occurred in showing the users both existing conditions and suggested conditions, while simultaneously allowing the user to distinguish the suggestions from the existing conditions. This was the key of the design.

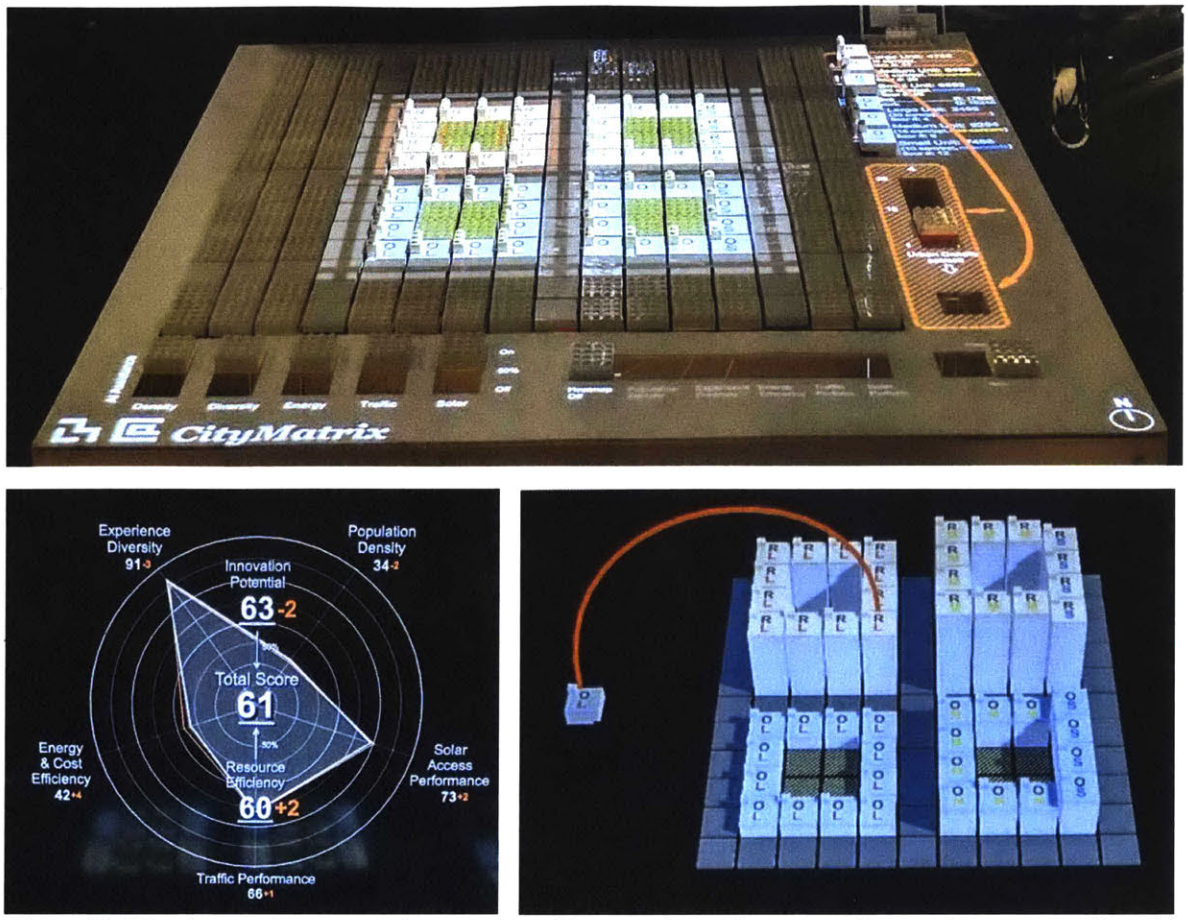
The early versions of the AI suggestion design adopted the concept of Blink, which is the information of the existing and suggested alternates every second. For example, for all odd seconds, the users saw the **existing** conditions of a Lego city grid, heat-map, and radar-chart scores. For all even seconds, the users saw the **suggested** conditions. For users to distinguish which was existing and which was suggested, the area with a difference was highlighted by red or green markers. The markers blinked red for all odd seconds and green for all even seconds (Video 10).



Video 10: Blink approach to represent AI suggestions: Red markers show the existing condition for all odd seconds (left); Green markers show the suggested condition for all even seconds (right).
(link: http://yanzhangworks.com/cmv10)

During the first round of user tests, it became apparent that it was difficult for the users to keep track of what had been changed while multiple parts were blinking visually. A learning curve for the users occurred preventing them from memorizing that red was the existing condition while green was a suggestion. Because of this, the AI suggestion representation design was iterated to a new approach, the concept of annotation.

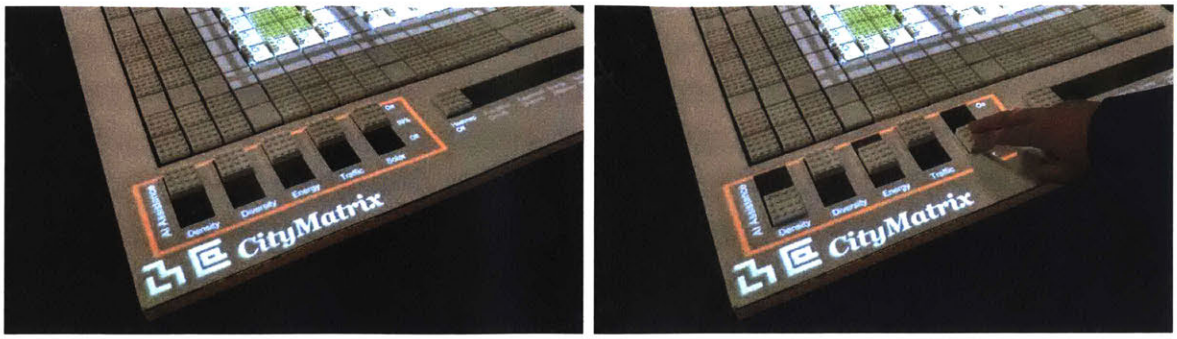
The Annotation approach meant that all the information of the existing condition remained untouched, and all the suggested information was represented as an annotation in orange. For example, as shown on the table, the orange markers and arrows indicated what suggested move was to be made next. The orange polyline in the radar-chart and the texts beside the scores indicated the projected score changes if the users accepted the suggestion. The orange arrow in the 3D city view also indicated the suggested move (Video 11).



Video 11: The Annotation approach represents AI suggestions: Orange markers and arrows on the table indicate the suggested next move (upper); Orange polyline and texts beside the scores indicate the projected score changes if accepting the AI suggestion (lower left); Orange arrow in the 3D city view indicate the suggested next move (lower right). (link: http://yanzhangworks.com/cmv11)

4.3.4 AI Assistance Weights

To introduce a dialog the users and AI, *CityMatrix* provided an important feature, AI Assistance Weights, shown on the lower left side of the table. The five 3-level switches corresponded to the five aspects of urban performance. Users could switch them separately to ON, OFF, or 50% representing how much help they wanted from the AI suggestions. The AI then updated its suggestions according to those weights (Video 12).



Video 12: Users can adjust AI assistance weights through five 3-level switches. When all switches are turned to ON state (left), AI will give an equal weighted suggestion. When only one switch is ON while the rest are OFF (right), AI will give suggestions that only focus on improving the score of that aspect.
(link: http://yanzhangworks.com/cmv12)

Because the user could adjust the AI assistance weights, anytime when using the system's suggested AI moves, the users could focus more on defining their goal of the design, instead of calculating and testing to reach the goal. This effect was a meaningful phenomenon observed in the user test about the usefulness of the AI assistance.

4.3.2 Construction

The AI suggestion was a part of the *CityMatrix* Python server. The main body of the AI suggestion program consisted of a strategy component and an evaluation component. (Figure 21) When new city configuration input was given, the strategy component first planned the possible moves to be made next and let the evaluation component evaluate each of the city configurations resulting from the possible moves. Then it found the best-scored move to be made next and sent it back as a suggestion to the visualization output component. The evaluation component evaluated the score of one city configuration as the sum of the aggregated urban performance metrics weighted by AI assistance weights. Traffic and solar performance metrics were predicted by the ML simulation predictor.

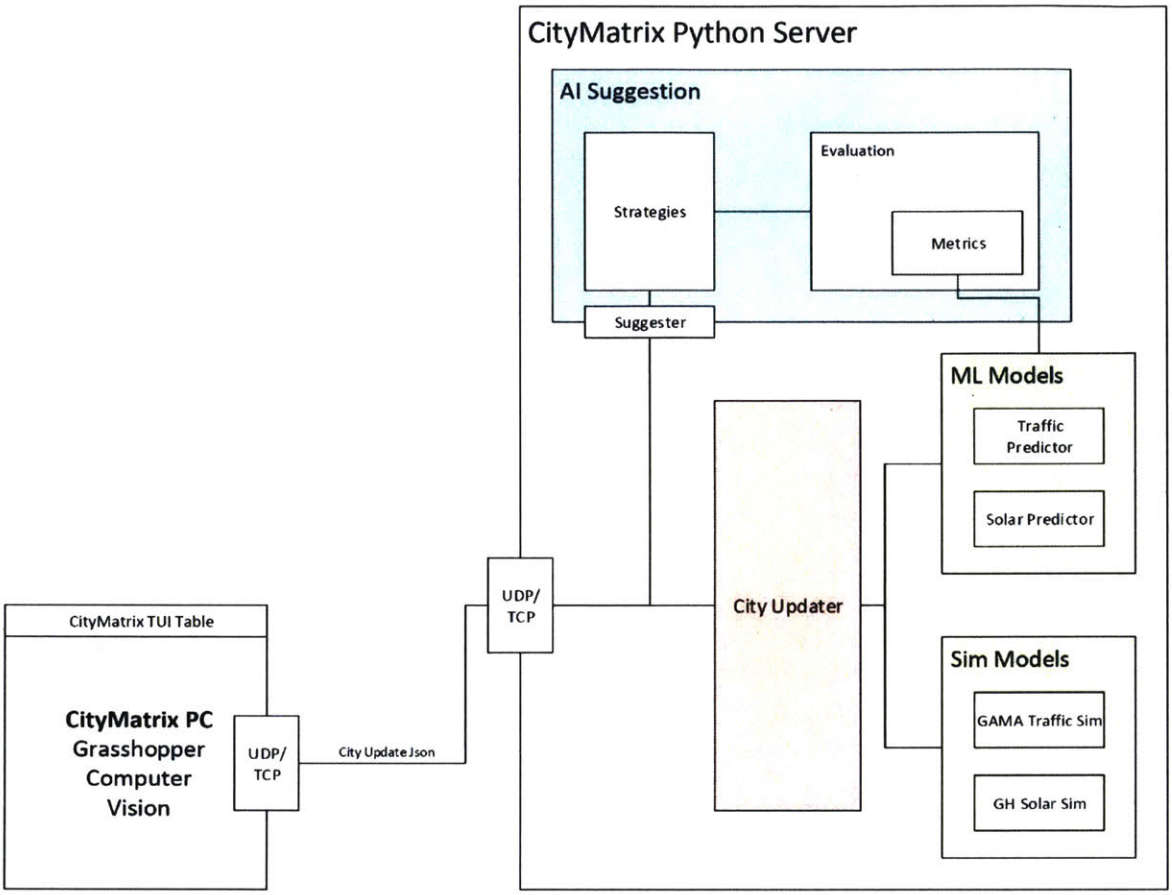


Figure 21: System diagram of CityMatrix Python server and its connection to the TUI table.

4.3.5 Search Algorithm

The search algorithm implemented in the strategy component was uncomplicated. It randomly determined a subset of the total possible moves that could be made next. The total number of the possible moves (search space) was 1966 (16*16*7 for changing a building cell + 6 * 29 for changing the building height of one type). For such a small search space, 150 to 500 possible moves were chosen and evaluated. It took one CPU thread less than one second to evaluate 150 possible moves.

To search the optimal move was the same as playing a board game, such as chess or Go. The AI suggestion search algorithms could be developed similar to the AI for chess or Go. AlphaGo provided a good model because of its generic learning strategy.

A particular algorithm was to be implemented in *CityMatrix* AI in the future work will be the *Monte Carlo* tree-search algorithm [33]. At the present, the AI suggestion searches for the next move only. There could be more steps while balancing the depth and the width of the search (Figure 22). This might address the issue as to whether or not the best move in the next step could be the best move ten steps later.

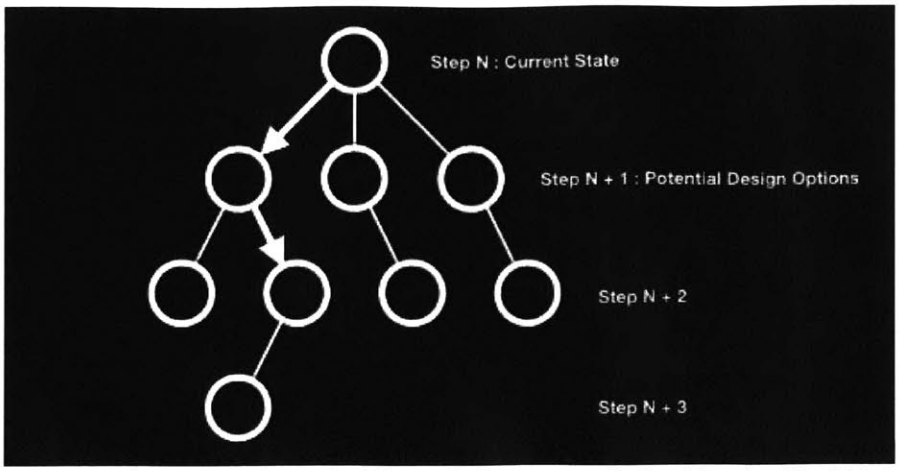


Figure 22: A diagram shows the Monte Carlo tree-search algorithm for CityMatrix AI suggestions.

4.4 Natural-Language Voice Interface

An ever-increasing number of digital products were then equipped with a natural-language voice interface, such as Alexa from Amazon, Siri from Apple, Google Assistant, Cortana from Microsoft, and Watson from IBM. One of the reasons for this is that for each person, other than visual input, listening, and speaking in a natural-language cadence would be the most accessible and effective method of communication, especially when having more abstract concepts. In this work, the *CityMatrix* Guide was designed to leverage the benefit of a natural-language voice interface to improve the usability and effectiveness of *CityMatrix* (Video 8 in 3.4.5 *CityMatrix* Guide).

The technologies used to realize *CityMatrix*'s natural-language voice interface were

1) Synthetic voice speaking, RT Voice library in Unity
2) User voice recognition, Windows 10 voice recognition implementation in Unity

Currently, for voice output the *CityMatrix* voice interface only read what was preprogrammed, such as introduction spoken by the AI guide. For voice output, the *CityMatrix* only listened to a limited number of preprogrammed voice commands. It did not respond to any questions for the user that were not predefined in the system. Future work could enhance *CityMatrix's* ability to understand flexible questions and provide more context-aware answers or conversations. Collaboration with companies with experience in technology of related field is preferable.

Chapter 5

Evaluation

To understand and evaluate the user-interface and the AI suggestions functionality of *CityMatrix*, five pilot-user tests were conducted. Each group had four users from both professional and non-professional backgrounds related to urban decision-making (Figure 23).

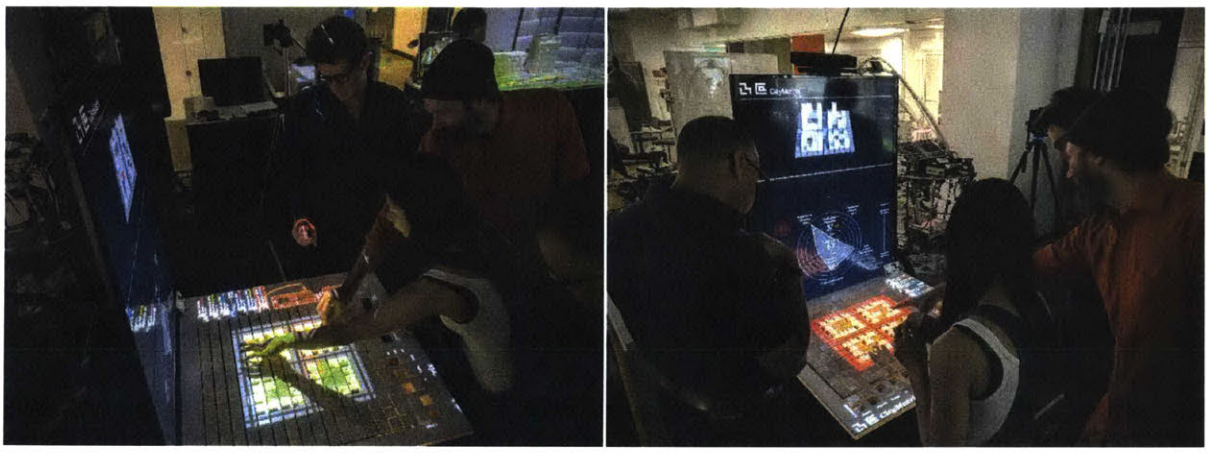


Figure 23: Pilot-user test of CityMatrix with four users from both professional and non-professional backgrounds related to urban decision-making.

5.1 User Test Procedures

The duration of each experiment was 60 minutes

1) Before the on-platform experiment, a short session was held to briefly introduce the research and survey the user's knowledge or experience about urban decision-making to determine the priority of their concerns about urban issues. Refer to Appendix **Error! Reference source not found.**) for the actual questions asked. Then the subjects were required to reach a consensus on the answer. This became their overall goal in the experiment.
2) During the on-platform experiment, the subjects were assigned to a group of four and stood around the *CityMatrix* platform. The voice-interface and the visual-interface then conducted an automatic, step-by-step introduction on the functionalities. The introduction lasted about five minutes. After the introduction, the subjects were asked to collaboratively operate the tangible interface, the Lego

city model, to achieve a city design with as high as possible performance score and their overall goal. The subjects were able to see the real-time performance scores during the experiment via the visual interface. The Round 1 experiment had a time limitation of 15 minutes.

3) Next, the subjects were asked to repeat the collaborative operation with the AI suggestion functionality. The Round 2 experiment had a time limitation of 15 minutes.
4) After the on-platform experiment, the subjects were required to answer a questionnaire about the experience of using and the usefulness of the platform and the AI suggestions (Appendix **Error! Reference source not found.**, After-Test Questionnaire Form).
5) End of the session.

In addition to the questionnaire, log data generated by the Python server was collected. The whole process of the experiment was recorded in video and audio formats. Combining the questionnaire, server log data, and tape-recorded data provided an insight for a better understanding of the *CityMatrix* platform. These results were evaluated both quantitatively and qualitatively.

5.2 Server Log Data Analysis

The server logged data points constantly when there was a new user input. Each data point included the current state of the building type ID on each grid cell, the building height (building population) of each building type, the heat-map value of the five aspects of each grid cell, the urban performance metrics of the five aspects on radar-chart, the move to be made next suggested by AI, and the five weights of AI assistance.

With the collected server-logged data, the urban performance metrics of five aspects and the total score evolving with time in each round was reconstructed. Figure 24 shows the data of one typical user test. The x-axis represents the scale of time in minutes, while the y-axis represents the normalized score. In Round 1, the experiment was conducted without the AI suggestion function. In Round 2, the users were provided with the suggestions from AI.

Insights about the process were revealed by the chart (Figure 24) and the tape recordings. During the first five minutes (at the 8-13 minute-milestone) of Round 1, the scores and the metrics did not change significantly. The group was not yet familiar with the tools and each other because they were discussing where to start. After that (at the 14-15 minute-milestone), one user decided to try some moves out, causing the scores and metrics to rapidly change. There was a period (at the 15-22 minute-milestone) where the users became more proficient in improving the total score while balancing the metrics. However, the total score dropped during the last part of Round 1 (at the 22-25 minute-milestone) because the

group decided they wanted an overall better goal, which was to make the resource efficiency score higher. When they started testing this new strategy, the metrics shifted. The traffic (blue) obtained a higher score while the diversity (yellow) score decreased. The total score decreased as well. Until the end of Round 1, the group did not have enough time to bring the total score higher. In Round 2, the team began with some testing (at the 29-35 minute-milestone). After the 35 minute milestone, they better understood the inherent tradeoffs in the system. They decided on their final strategy of the score combination. From then to the end of Round 2 (at the 36-44 minute-milestone), they kept refining the city configuration to achieve a better total score.

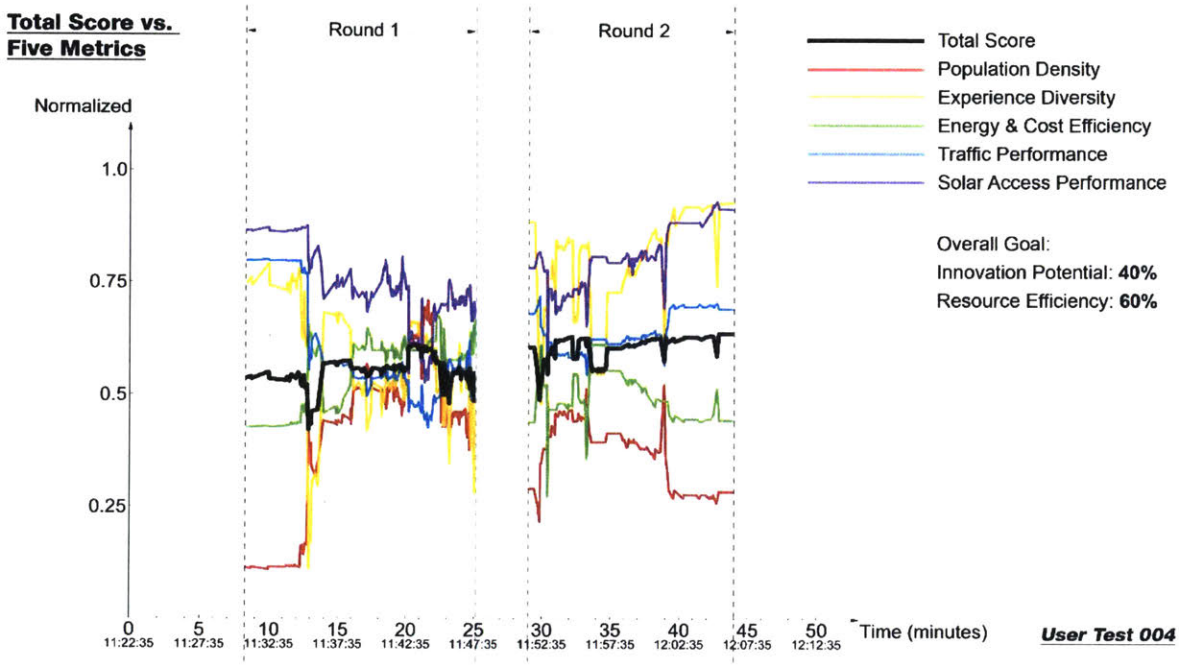


Figure 24: Total score vs. five metrics.

Figure 25 gives details about the AI suggestion accepted by the users and its correlation with the improvements of the total score during the same user test. Most of the groups had a higher AI suggestion acceptance ratios at the beginning of Round 2 (at the 29-35 minute-milestone), lower ones in the center (at the 29-43 minute-milestone), and higher ones again at the end (at the 44 minute-milestone). This cycle of the change in AI suggestion acceptance ratio indicated that the team was becoming familiar with the AI suggestions. The team became skeptical and started rethink about their overall goal. They finally used the suggestions made by AI to increase their final total score.

In the pattern found on the chart (Figure 25), AI suggestions helped stabilize and improve the total score in Round 2 compared to Round 1.

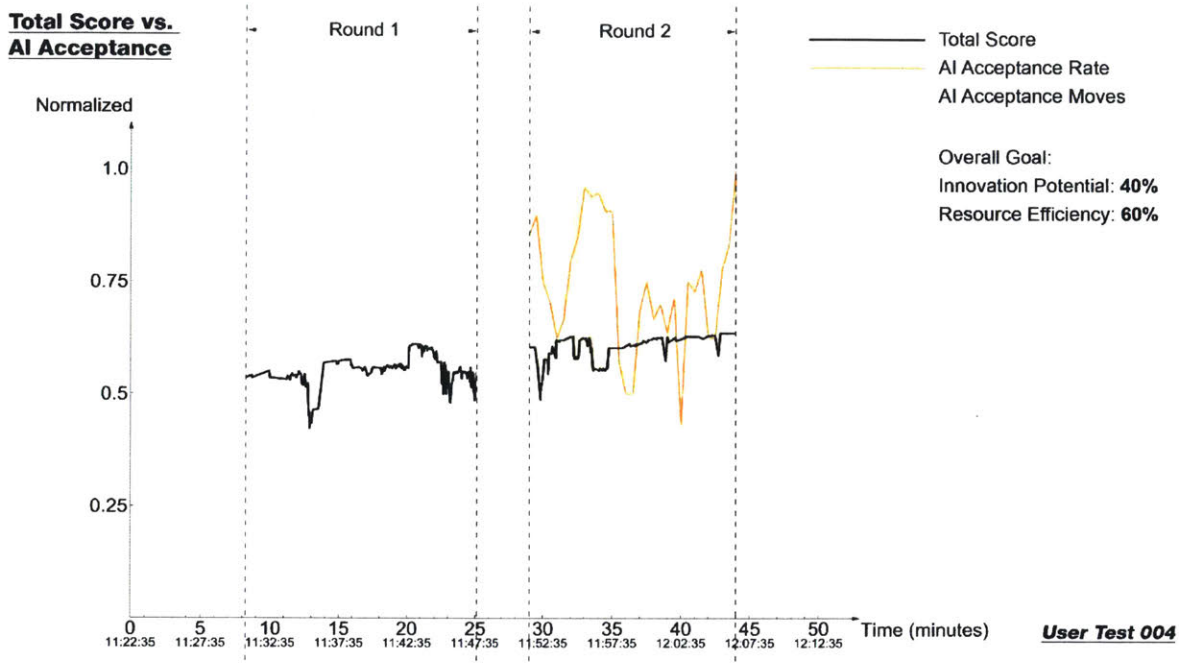


Figure 25: Total score vs. AI acceptance.

Figure 26 shows how five metrics evolved through time against the AI Weights chosen by the users and the accepted AI suggestions. It was a visual tool to examine when the user used AI suggestions for which aspects of the urban performance and the impact of accepting an AI suggestion. One phenomenon found in all the tests was that the users made significantly more moves in Round 1 than in Round 2. The scores of individual metrics fluctuated more in Round 1.

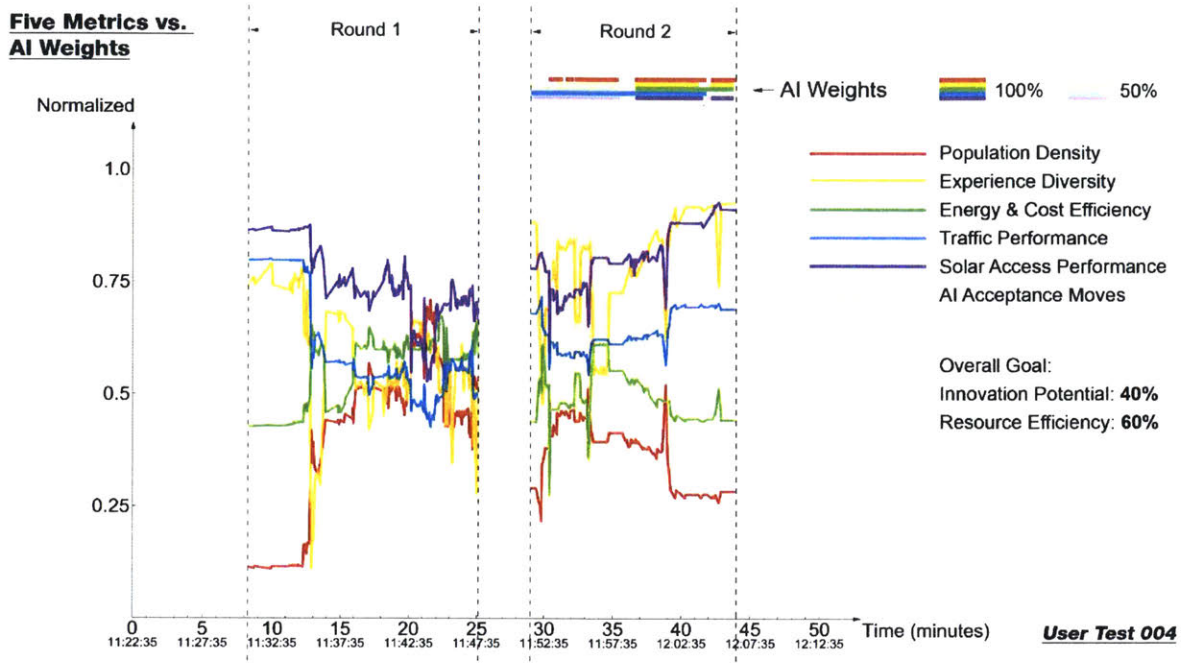


Figure 26: Five metrics vs. AI weights.

From the collected data, a better understanding of how AI statistically provided suggestions can be seen. For example, in all five user tests, 74% of the suggested moves were cell changes and 26% were density changes. (Figure 27, left) AI suggested changing the height of Large Office Unit (building type index 3) to lower building heights more frequently. (Figure 27, right) This could be due to these types of changes increasing the diversity metrics most efficiently.

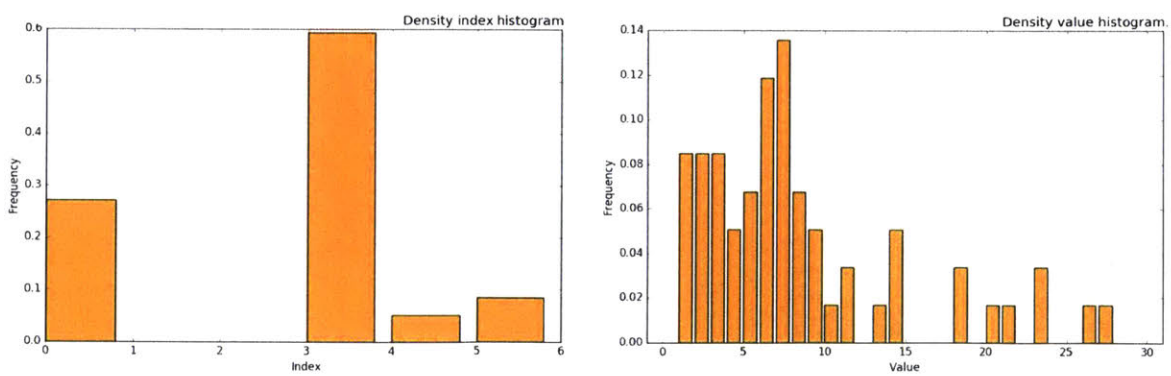


Figure 27: Frequency of the target building type index suggested by AI (left) and frequency of the target building height values suggested by AI (right).

(For more data analytics, please refer to Appendix E.)

5.3 Questionnaire Data Analysis

Questionnaire data provided a perspective to some of the topics about the usefulness of the system. Below is the statistical data of the answers to the selected questions. (For the complete question list and the data analysis, please refer to Appendix F.)

1) How much do you agree: the user interface is overall easy to understand?
(On a scale of 1-10 with 10 being the highest indicate how much you agree with)

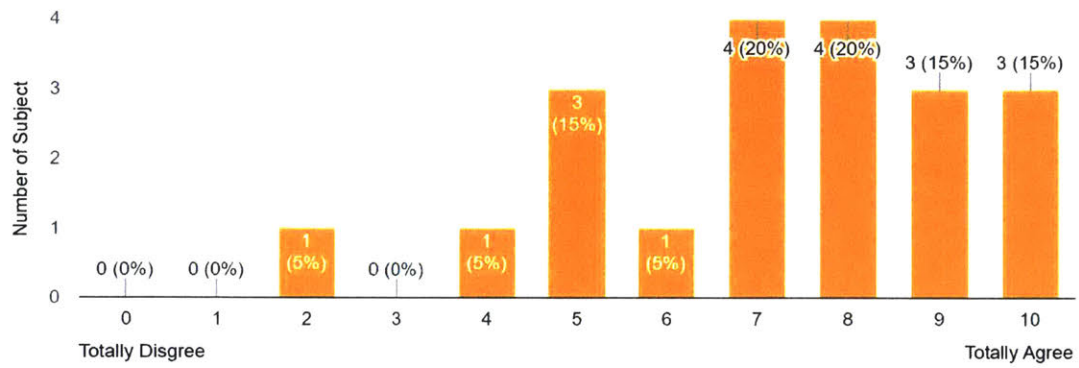


Figure 28

2) How much does the system help you understand better how the city works?
(On a scale of 1-10 with 10 being the highest indicate how helpful it is)

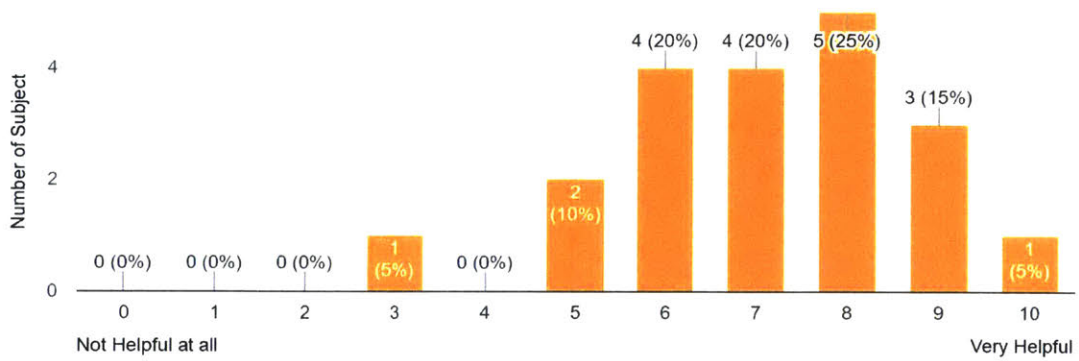


Figure 29

3) How helpful do you think are the suggestions made by the AI?
(On a scale of 1-10 with 10 being the highest indicate how helpful it is)

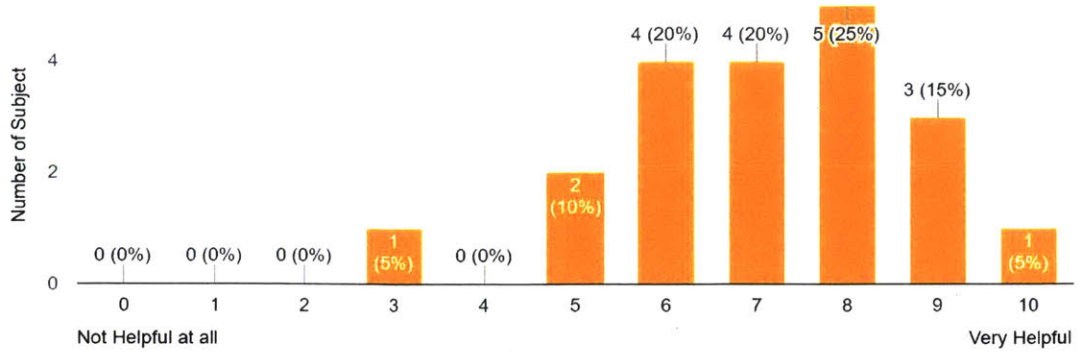


Figure 30

4) How are the suggestions made by AI helpful?
(Selected quotes of subjects' answers)
   a. *"They clearly help optimize your overall score, but sometimes it's difficult to understand what the conceptual / design implications of those suggestions are."*
   b. *"The connection between the intended goal and recommended action was clear."*
   c. *"It helped me map/evaluate the general effect of the single decision. I think the AI helped shift attention to the total scores."*
   d. *"Easy to see if things would improve."*
   e. *"When well configured, the AI tells us the right thing to do."*
   f. *"It helps convey the logic."*
   g. *"It helps on getting more points but not to understand why."*
   h. *"It's easy to get lost in blindly following its steps."*
   i. *"Once you identify the weakness you are trying to address, the AI was able to specify changes that would help reduce the weakness."*
   j. *"There are many variables to consider. The AI helps focus attention on one relevant variable at a time."*

5) How much do you agree: I have learned something from the suggestions made by the AI? (On a scale of 1-10 with 10 being the highest indicate how much you agree with)

6) What did you learn?
(Selected quotes of subjects' answers)
   a. *"Sometimes the best move is replacing building types."*
   b. *"density and environment are hard to achieve."*

c. *"It showed new combinations of types of buildings."*
d. *"The complex system that the city is."*

7) I am more satisfied with the design of the city district in the second round because:
   a. as we spent more time working together, we formed a better collaboration;
   b. as I spent more time working on the system, I got more familiar with it;
   c. the suggestions made by AI was helpful;
   d. other reason; (please specify)
   e. Compared to the first round, I am NOT more satisfied with the design of the city district in the second round.

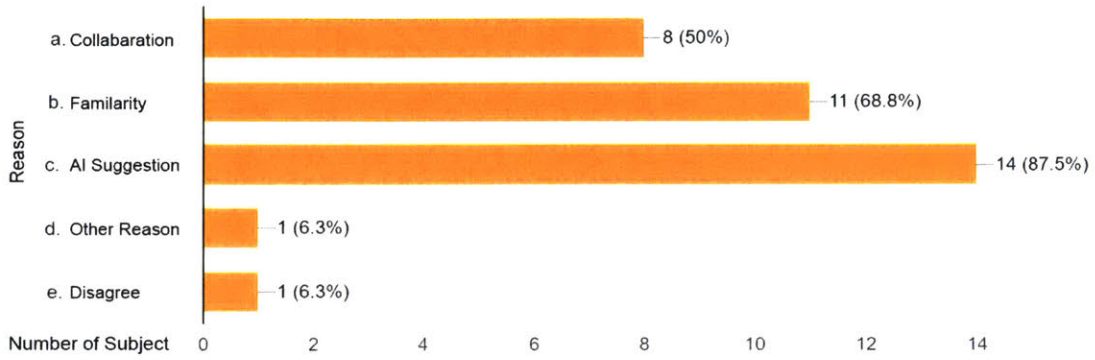


Figure 31

8) How much do you agree: sometimes I tend to follow its instructions blindly? (On a scale of 1-10 with 10 being the highest indicate how much you agree with)

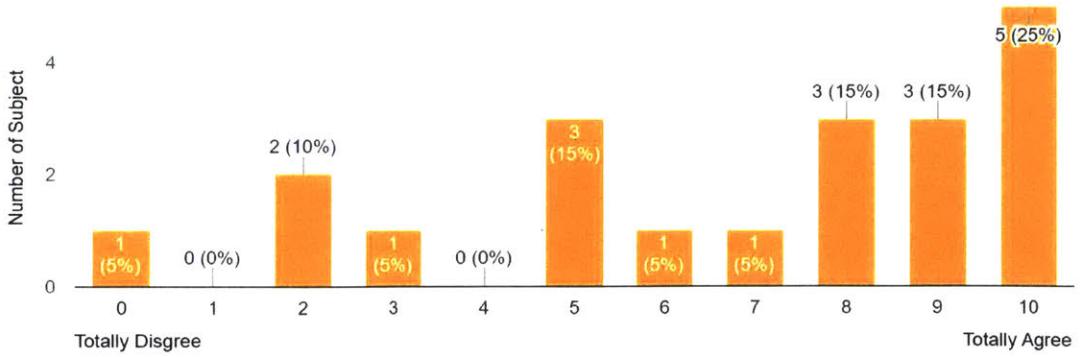


Figure 32

9) Do you feel that the AI suggestions make you think less or more about the city? (On a scale of 1-10 with 10 being the highest indicate how much you think)

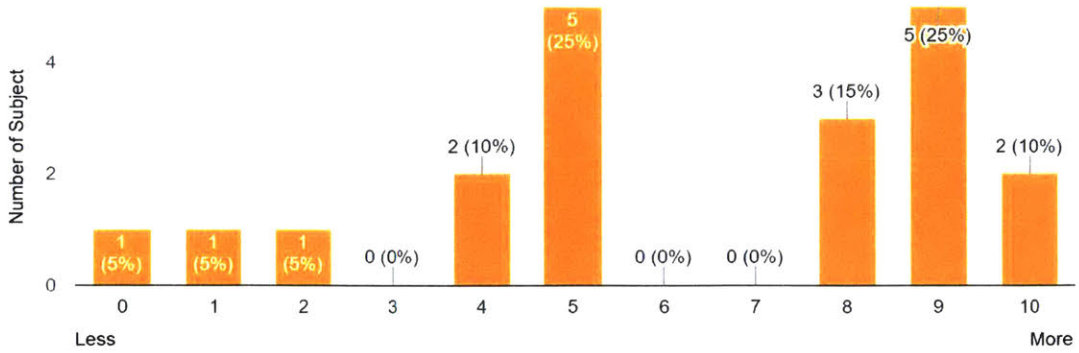


Figure 33

10) How much do you agree: *CityMatrix* system promotes collaboration in urban decision-making?
(On a scale of 1-10 with 10 being the highest indicate how much you agree with)

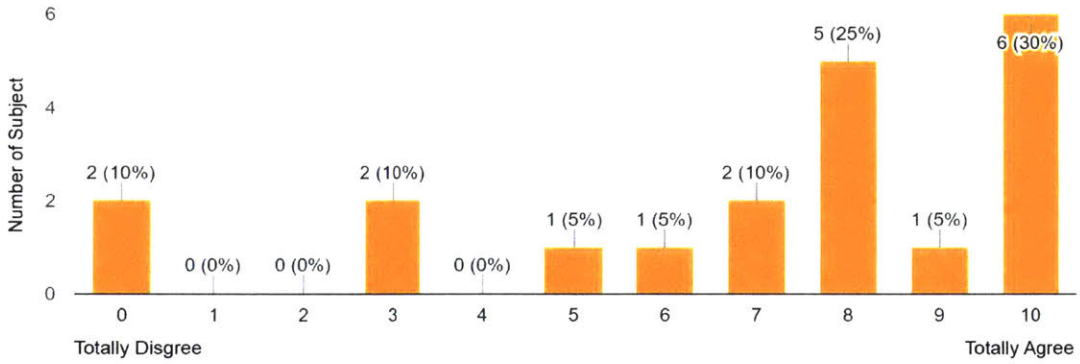


Figure 34

11) Do you have any comments or suggestions for improving the system?
(Selected quotes of subjects' answers)

a. *"The model presents really good fundamental considerations for trade-offs in urban design. The interface is very cool and engaging. The system seems too simplistic to be a useful planning tool. Other types of land uses should be introduced (commercial, institutional), and issues in equitable development could be included (factors like transit and affordable housing)."*
b. *"After we realized the game of the suggestions would not help, we get more critical of the AI - before we were pretty much just following its instructions."*
c. *"Most of the time I follow the instruction without thinking."*
d. *"I really appreciate your efforts making us more aware of the city we live in."*

e. *"AI suggestions didn't make me follow its instructions blindly, actually make me think more!"*
f. *"The whole system looks very cool!"*
g. *"In the first game we were collaborating and working as a team, but the score was lower. In the second try, our score was better, but we stop the collaboration."*
h. *"It would be helpful for the community activists who are opposed to height and density to be able to benefit from the project and what it has to teach about development in urban areas."*
i. *"Good to share opinions with other people!!!"*
j. *The blocks are very abstract, making it hard to visualize exactly what each change meant.*
k. *"Improving the latency of system response would help prevent some errors where it was not clear exactly which change had an impact."*
l. *"The interface is great as is the AI instruction. One improvement could be to explain how the AI makes suggestions."*

Chapter 6

Conclusion

6.1 Discussion

"Most of the time I follow the instruction without thinking."

"I didn't pay attention to what it told me to do."

"The suggestions are helpful to improve your score, but the system does not tell you why it's making the recommendations."

As quoted above from the subjects' answers, one of the most common notes from the users was that some users tended to blindly follow the suggestions made by AI. Two reasons were identified:

1) The lack of city context resulted in the lack of the understanding of the goal. Because the user test was an abstract, hypothetical city, it was difficult for the user to imagine the meaning and goal of the city other than the five given aspects. This directly led to the sole purpose of achieving a higher score. This could be improved if the system will be applied on a real project.
2) The lack of understanding of why and how a suggestion was made also contributed to blindly following the instructions. This could be improved by developing a UI that communicates reasons for the suggestions. It can be displayed using some text labels of categorized reasons or in a natural-language voice interface.

Another comment was that the suggestions made by AI were too specific. Some users wanted to have high-level suggestions. For example, instead of giving a suggestion such as "increasing the density of OS to 25 floors," some users preferred the AI suggestion such as "arranging taller buildings on the north and lower buildings on the south will help the solar access performance." These high-level suggestions required the AI to learn from its experience and summarize the pattern of the solutions, which could become a challenge in future work.

CityMatrix raised an interesting question: What will be AI's roll in the future of urban decision-making process. The AI's roll in *CityMatrix* was **to augment but not to replace human**. For this reason, it provided the users step-by-step suggestions but not final

solutions. The *CityMatrix* Guide also reminded the users that they could either accept or ignore a suggestion made by AI and depend on more aspects, which could not be quantified by the system.

From the questionnaire results, the educational benefits of *CityMatrix* became apparent. Using *CityMatrix* helped users increase their awareness of a city. Some users learned from *CityMatrix* that a city is complex, and there are inherent tradeoffs among the different aspects of urban performance.

Another lesson learned in the pilot user tests was: Small details in system design could make a big difference. The tool might be misleading when it was too simple and was not representative enough. For instance, the density score on the radar-chart was not changing when exchanging buildings. The reason was that proximity was not taken into account in the aggregated scores.

6.2 Future Work

The current version of *CityMatrix* and AI suggestions has definitely raised more questions than answers, which inspired further thinking, refinement, and improvement of the project. Listed are some potential directions for the future work:

1) It would be more meaningful to apply the system to a real context, useing real data, and solving a real community issues. Either The Volpe project in Cambridge, Massachusetts, or the Living-Line project near Tongji University, Shanghai, would be an effective place to begin.
2) When AI gives a suggestion, it could also tell users why it is giving such suggestion.
3) Investigating and adapting a variety of depth search algorithm, such as a *Monte Carlo* tree-search, would improve the quality of the suggestions made by AI.
4) AI could give multiple-step suggestions instead of one-step suggestions. However, it needs to be kept in mind that a suggestion of too many steps might later hurt the openness of the suggestion system.
5) *CityMatrix* could provide less specific suggestions, such as a heat-map indicating where the best ten moves would be.
6) *CityMatrix* could incorporate more aspects of urban simulations.
7) The conversation with the AI could be more flexible. For greater accessibility and usability, the AI could understand and respond to users' inquiries in wider range, instead of only predefined ones. Collaboration could be made with the development team of Watson, Alexa, or Google Assistant.
8) A parallel process would enable the original urban simulations to be run simultaneously, supporting both the preliminary and final stage of the decision-making process. The real-time, rapid prediction of the simulation provides enough precision for an effective decision-making process. Meanwhile, the system would continue calculating the accurate, high-resolution simulation solutions to confirm the final decision.

9) A renewable training data set for Machine Learning (ML) could be explored. The parallel simulation results could be fed into the ML training data set to improve the precision of the ML prediction. These new data entries could increase the precision of the ML prediction, because they are more similar to the future user input compared to randomly generated city configurations.
10) AI could learn from the preferences of the users. From the learned patterns, more insights or conclusions could be made to infer users' intentions.
11) *CityMatrix* could provide benchmarks on the radar-chart to help compare the performance with other city configurations.
12) To promote the usage of CityMatrix at a larger scale, more open-sourcing works need to be done, including documentation, tutorial, and community development.

6.3 Contribution

6.3.1 Evidence-Based Democratic Decision Making

CityMatrix was designed as a first step towards evidence-based and democratic decision-making. It promoted collabrations among a wide range of stakeholders and a potential solution to enhance the accessability and efficiency of public engagement events. This would reduce the delay costs and improve the design quality.

6.3.2 Real-time Simulation Prediction

Real-time simulation feedback was not in demand because the proper collaboration and public engagement has not yet been emphasized and addressed. The research in Changing Places group such as the CityScope project opened up new opportunities to innovatively address these issues. To facilitate a more informative urban public-engagement tool, real-time feedback is necessary.

The real-time feedback is not only important for public accessibility, but also for generally enhancing the quality of decision-making and design processes. With real-time feedback, examining the whole spectrum of the design variation in a short amount of time, or even simultaneously could be accomplished. This transforms a quantitative calculation into a design intuition.

In this work, a versatile, quick, accurate, and low-cost method to enable real-time feedback of multiple complex simulations for stakeholders was explored and deployed. Instead of running these complex simulations in real-time, which would have been impossible even given the massive required resources, the system predicts a simulated result with a pre-trained Machine Learning algorithm in real-time.

6.3.3 AI Suggestion System

Cities are extraordinarily complex. To enable people to navigate, understand, and interact with complex information and possible solutions, the advanced optimization search algorithm was developed to provide suggestions on a step-by-step basis. The AI suggestions freed stake-holders from excessive quantitative considerations and enabled them to focus on the qualitative ones. As a result, the quality of the decision was improved by taking advantage of the organic combination of both human intelligence and machine intelligence.

6.3.4 Global Impact

CityMatrix is an open-source research project that could be used and redeveloped by professionals and non-professionals globally. Earlier versions of *CityMatrix* have already been used in Dubai, Hamburg, and Andorra. It was also used as primary research platform in workshops collaborating with Tongji University in Shanghai, Aalto University in Helsinki, and a City Sciences course in MIT Media Lab in Cambridge. During these workshops, the students were not only able to use *CityMatrix* but also redevelop their decision-making support platforms assisting in their research topics. *CityMatrix* enables students and designers to plan and design through a more intuitive, dynamic, and evident-based methodology (Figure 35).

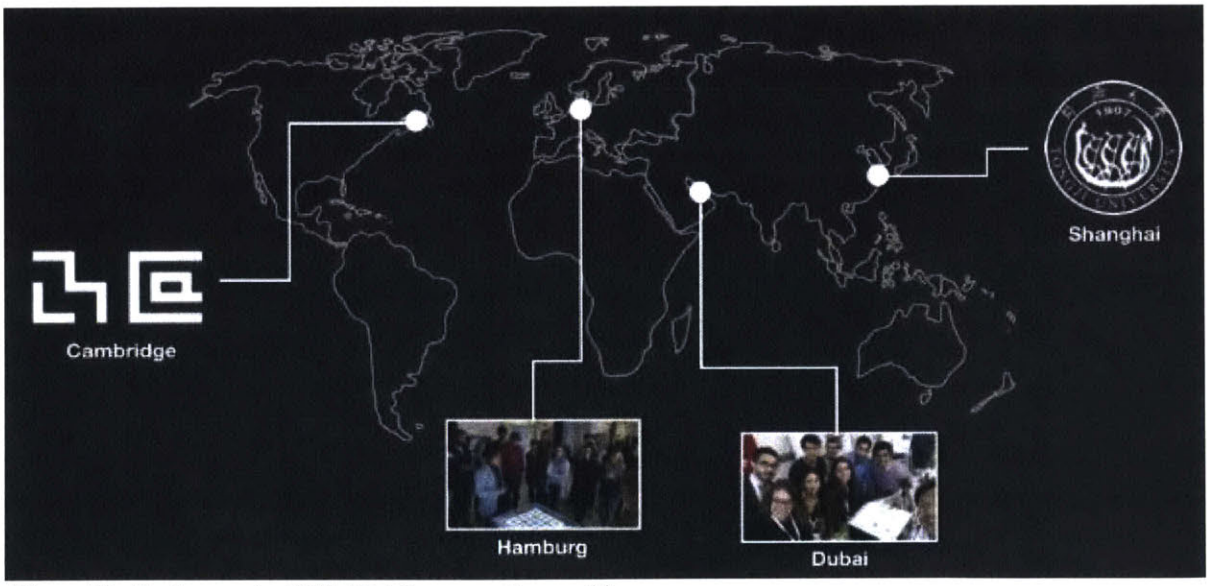


Figure 35: Deployment of CityMatrix across the world

Appendix A

CityMatrix Guide

The following text is the voice script of the AI Guide. Text in the parentheses at the beginning of each session indicates the state of the self-introduction progress or which component of the interface will be highlighted when AI Guide says the script of the correspondent session. A video record of the complete AI Guide process is attached (Video 13).

Video 13: The complete process of CityMatrix AI Guide introducing urban concept, user interface, and AI suggestion functionalities. (link: http://yanzhangworks.com/cmv13)

*(***START***)*

(mask: all; highlight: none)

Welcome to *CityMatrix* - an urban decision support system augmented by artificial intelligence. I am your AI assistant, RZ-14.

(highlight: Lego grid)

Please look at what is highlighted below on the table. This is the hypothetical city district we are going to work on. Yes! We are going to use Lego bricks to build the city district.

(highlight: building types)

Now the highlighted are the six different types of buildings, including three residential and three office. And each of them has small, medium, and large unit types, which will attract people of different age and income.

(highlight: building types, Lego grid, and arrow 1)

You can add, remove or re-arrange the Lego blocks to create your city district. Once you make a change, the system will update according to your input.

(highlight: building types, selection dock, and arrow 2)

(highlight: selection dock, density slider, and arrow 3)

(highlight: 3D city model)

Also, you can put one of the Lego bricks into the dock highlighted here to select all the buildings which are the same type, and then operate the slider to change the building height, thus the population density of these buildings.

Notice how the building height changes accordingly in this three-dimensional representation of the city district on the vertical screen.

(highlight: people statistics)

The system will calculate how many and what type of people your city district will attract, highlighted here.

(highlight: radar-chart)

As well as the performance of the city district in five aspects, represented by this radar-chart: population density, experience diversity, energy and construction cost efficiency, traffic performance, and solar access performance.

(highlight: radar-chart upper part)

According to previous research, population density and diversity are the key factors contributing to the innovation potential of the city. So we want to maximize both the population density and diversity to achieve a high Innovation Potential score.

(highlight: radar-chart lower part)

Meanwhile, higher population density tends to generate environmental issues. That's what we would like to avoid by increasing the energy and construction cost efficiency, traffic performance, and solar access performance. These three aspects define the Resource Efficiency score.

(highlight: radar-chart)

Both Innovation Potential score and Resource Efficiency score contribute to the Total Score according to the percentage importance you decided together earlier. So your goal is to

create a city district which maximizes the performance of all five aspects, thus the Total Score.

(highlight: Visualization slider, Lego grid, and arrow 4)

To better understand the impact of your design to the city performance, you can inspect each of the aspects by changing the visualization slider highlighted here. When you slide the Lego brick on the slider to one of the items, you will see the correspondent heat-map of the performance of that aspect on the Lego grid. Green is good. Red is bad. As simple as that!

(highlight: radar-chart "Population Density" part, slider "Population Density" part)

First, let's look at the "Population Density". The heatmap reflects the number of the people in the building. The greener, the more people in that building.

For example, if you select the "Office Large" type and slide up the density, that area will get greener. You will also get a higher score on the "Population Density" axis on the radar-chart, since it reflects the total number of the people in the district.

(highlight: radar-chart "Experience Diversity" part, slider "Experience Diversity" part)

Then, let's look at the "Experience Diversity". The heatmap reflects how mixed are the residential and office types, as well as the balance between the working and living. The greener, the more diverse.

For example, if you select the "Office Small" type and slide up the density, some area will get greener, meaning a better diversity in that area. You will also get a higher score on the "Experience Diversity" axis on the radar-chart, since it reflects the overall diversity throughout the whole district you built.

(highlight: radar-chart "Energy/Cost Efficiency" part, slider "Energy/Cost Efficiency" part)

Move to "Energy and Cost Efficiency". The heat-map shows the building energy consumption and building construction cost per person. Normally The energy consumption and building construction cost per square meter are similar for any building, which means the more small unit residential or office buildings, the better this performance.

For example, if you add an "Office Small" in the lower-left courtyard, that area will get greener. You will also get a slightly higher score on the "Energy and Cost Efficiency" axis on the radar-chart, since it reflects how efficient the whole city district is.

(highlight: radar-chart "Traffic Performance" part, slider "Traffic Performance" part)

Switch to "Traffic Performance". Assuming we are using a highly efficient autonomous sharing mobility system, the heat-map here shows how much traffic volume is in the city district in one day.

For example, if you decrease the building height of "Office Small", the traffic them around will improve. You will also get a higher score on the "Traffic Performance" axis on the radar-chart.

(highlight: radar-chart "Solar Access Performance" part, slider "Solar Access Performance" part)

Last, let's look at "Solar Access Performance". Assuming we want to fully utilize the solar energy on the roof tops of all the buildings and have good sunlight in the courtyard, the heat-map shows how much solar radiation you can get in each area in the city district annually. The more yellow, the better. The more blue, the worse. The symbol in the lower right corner shows the orientation of north.

For example, if you decrease the building height of "Office Large", the courtyard will get more yellow because it now has less shadow. You will also get a higher score on the "Solar Access Performance" axis on the radar-chart.

(mask: none; highlight: none)

Now, you've been through the introduction of *CityMatrix* functionalities.

Again, your goal is work collaboratively to build a city district with these Lego bricks that maximizes all five aspects of the city performance and the total score.

You have 15 minutes for this round. It's Lego time! Good Luck!

(reset the city physically)

(show timer of 15 minutes)

(USER TEST STAGE ONE: WITHOUT AI ASSISTANT)

(Beginning of the AI introduction)

Hope you have had fun with the first round!

From now on, I will try to give some help. Note that now on the table, you can see an annotation in orange. That is my suggestion of the most efficient next move to improve the score, given the current condition.

The orange line in the radar-chart on the screen is the city performance you could get if you follow my instruction. The orange texts on the right side of each of the scores illustrate the exact expected change on the score according to my suggestion.

Whenever you make a move, I will give you a new suggestion based on the latest situation. Please try following my suggestions once or twice to get familiar.

(highlight "AI Assistance")

Highlighted in the lower left side of the table, there are five switches that you can manipulate. Each of them indicates how much help you want from me in the corresponding aspect of the city performance.

Any time in the next round, you can move the slider up and down to ON, OFF, or 50%, according to your priority at the moment. I will update my suggestion every time you change them. Please give it a try.

(no highlight)

Lastly, I would like to remind you that you have the choice to accept or ignore my suggestion at each move.

My suggestions are solely based on the calculations of these five aspects of city performances, whereas the city is a complex dynamic system with much more to consider.

(no highlight)

Now, let's work together to make our city district even better! I hope my assistance helps.

You have 15 minutes for this round. Let's do it!

(reset the city physically)

(show timer of 15 minutes)

(USER TEST STAGE TWO: WITH AI ASSISTANT)

*(***END***)*

Appendix B

Server Log Data Analysis

Attached are the data analysis charts for the server log of all five user tests.

B.1 User Test 001

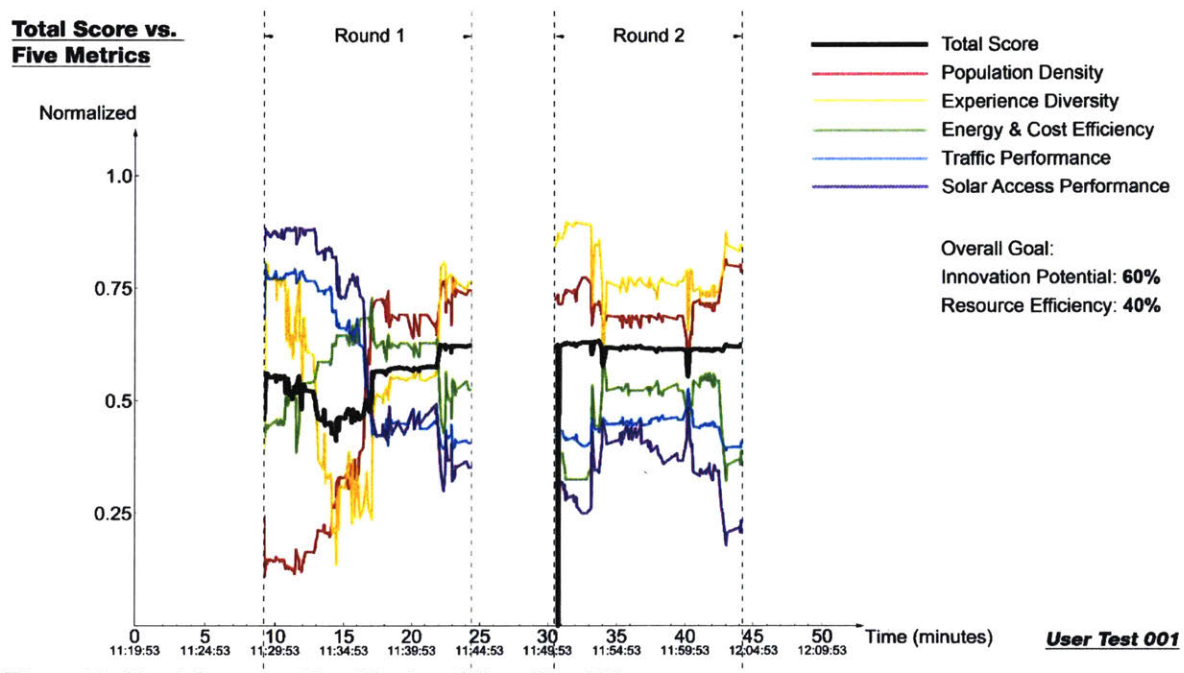


Figure 36: Total Score vs. Five Metrics of User Test 001

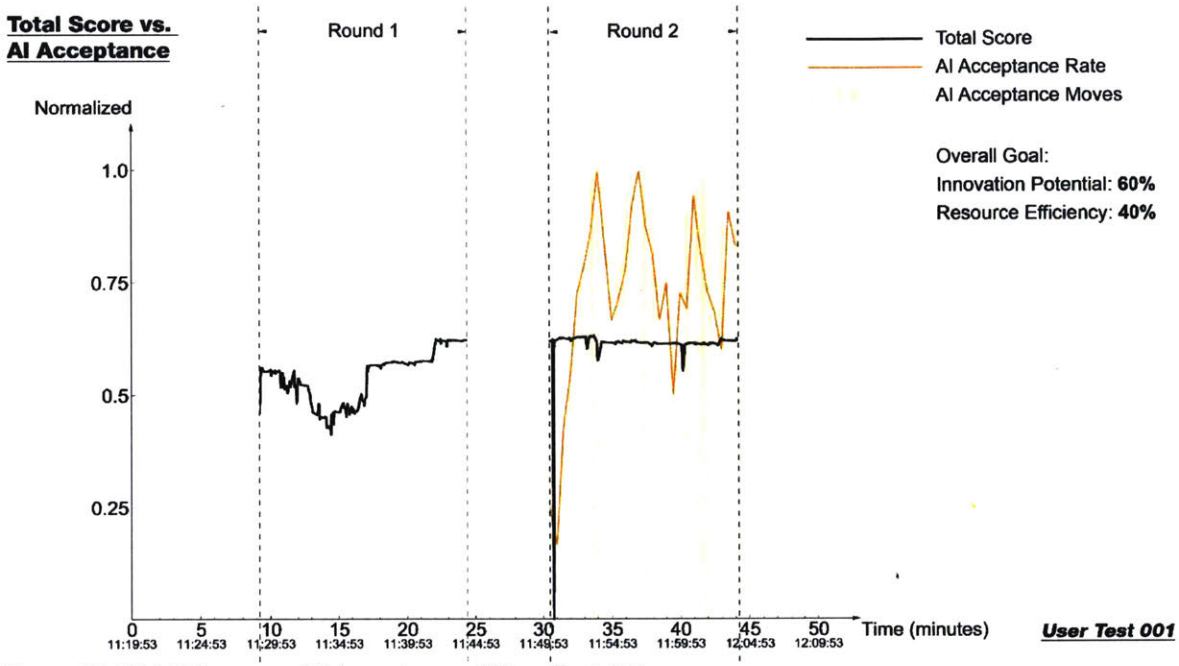


Figure 37: Total Score vs. AI Acceptance of User Test 001

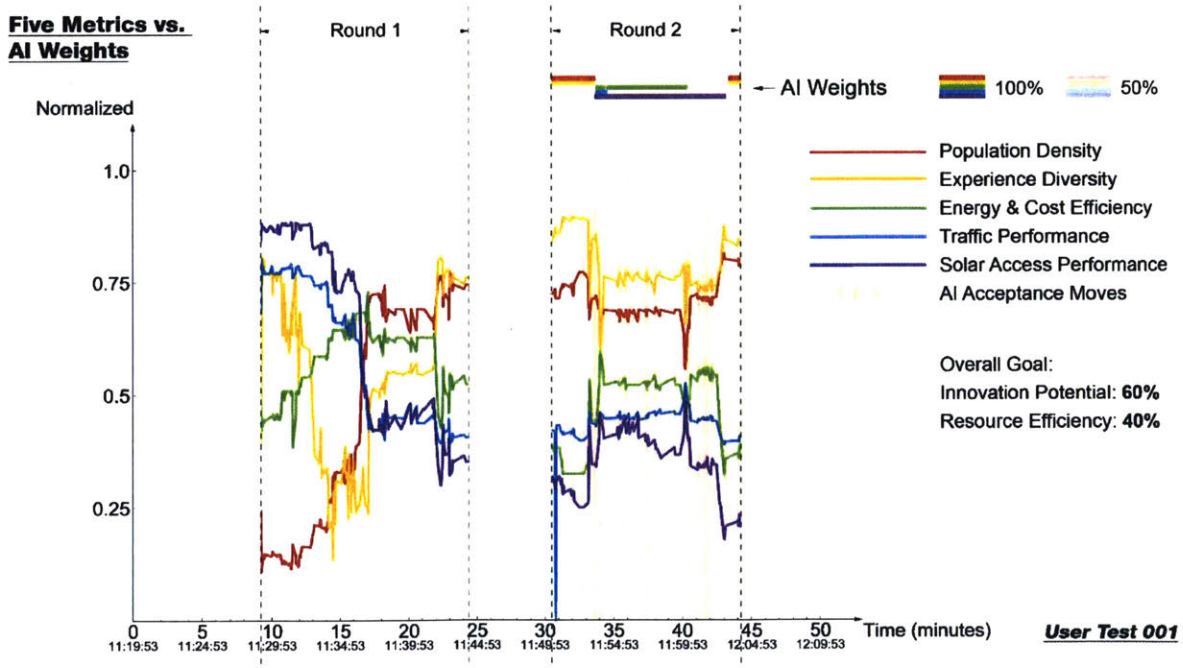


Figure 38: Five Metrics vs. AI Weights of User Test 001

B.2 User Test 002

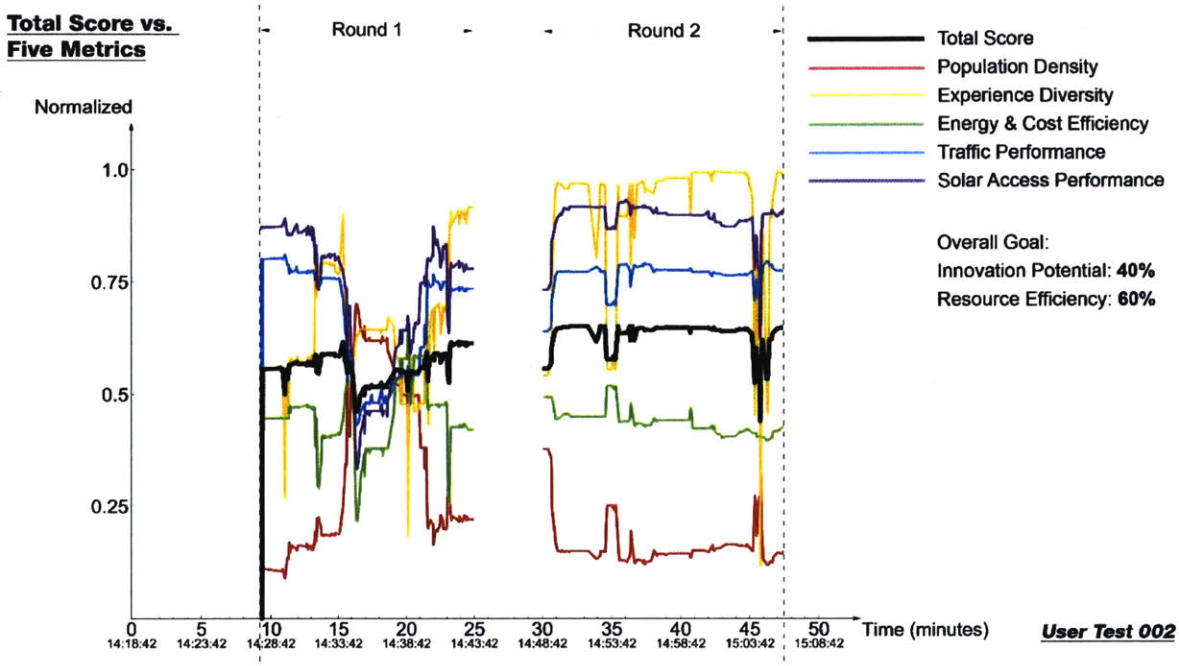


Figure 39: Total Score vs. Five Metrics of User Test 002

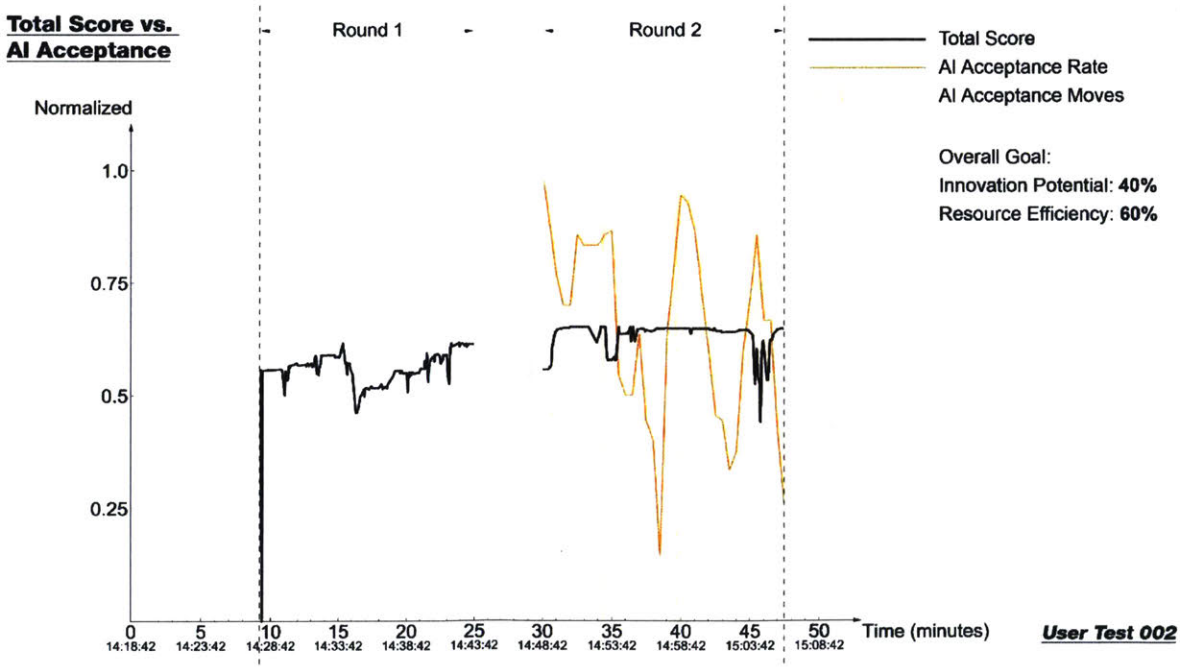


Figure 40: Total Score vs.AI Acceptance of User Test 002

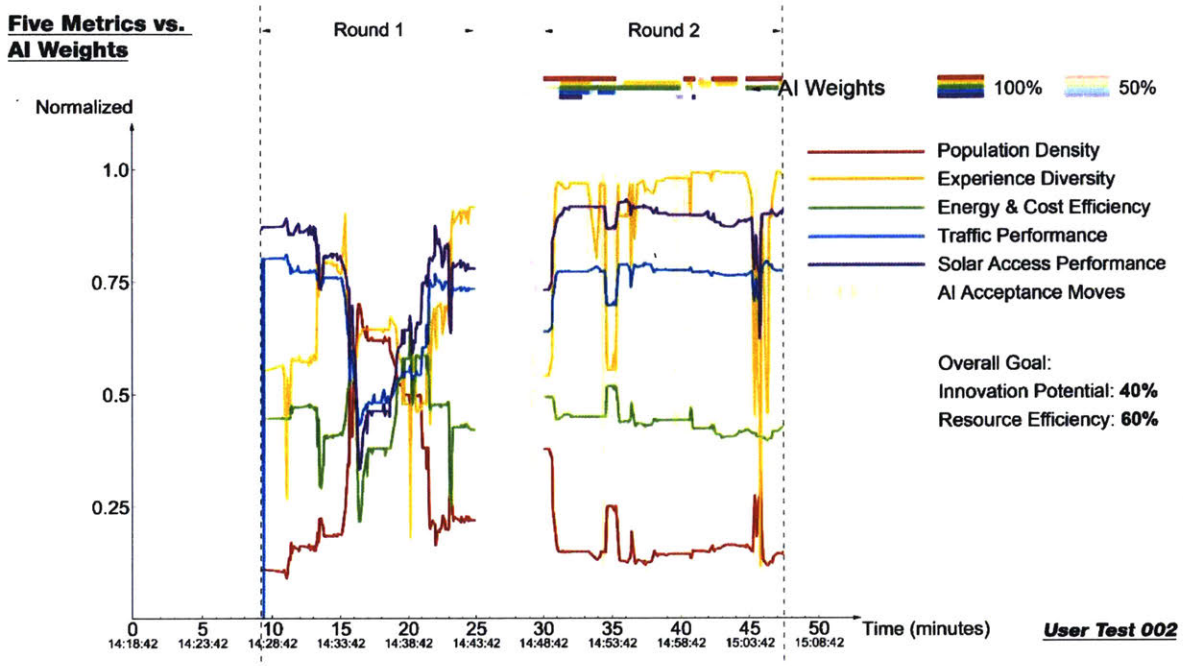


Figure 41: Five Metrics vs. AI Weights of User Test 002

B.3 User Test 003

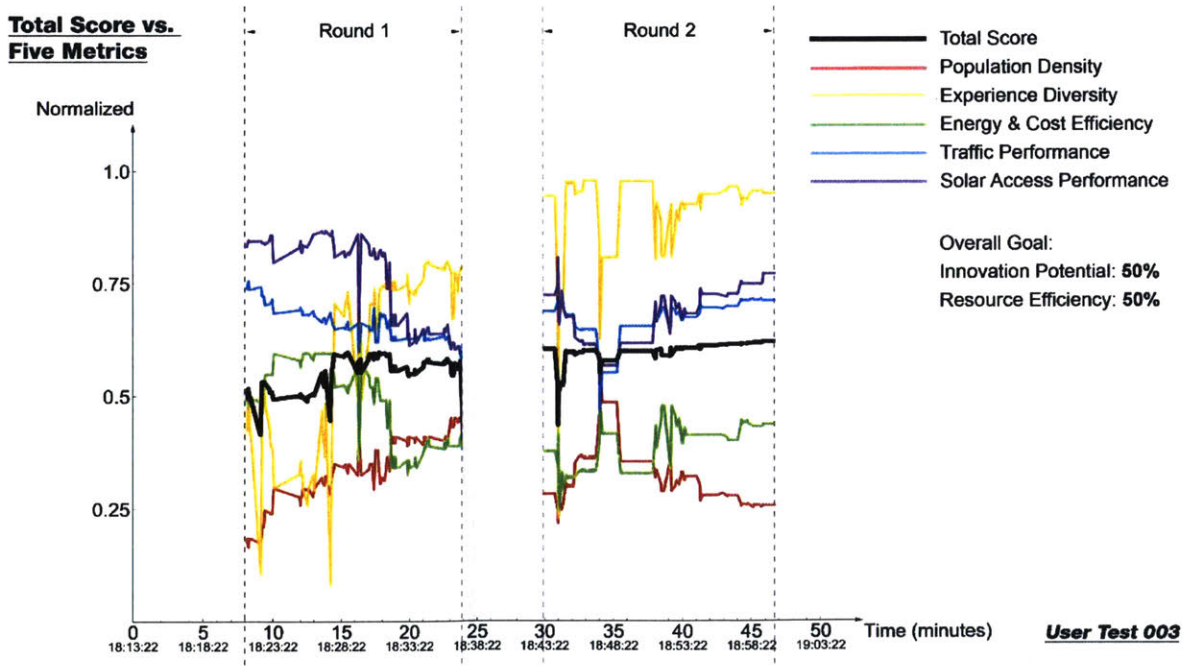


Figure 42: Total Score vs. Five Metrics of User Test 003

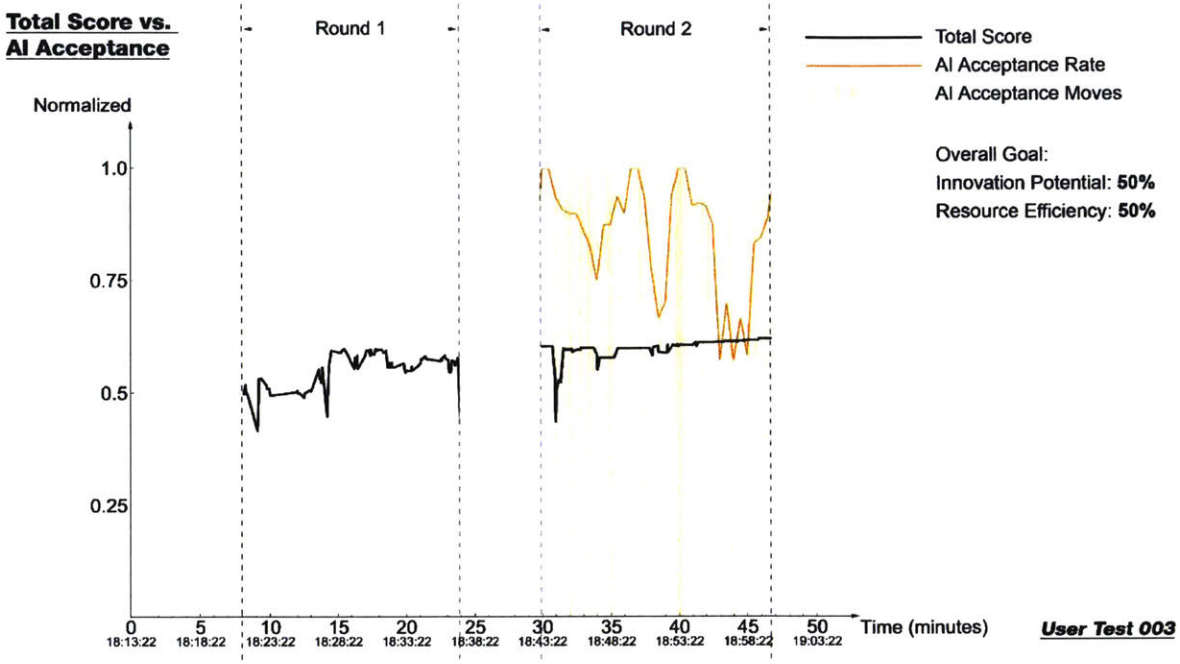


Figure 43: Total Score vs. AI Acceptance of User Test 003

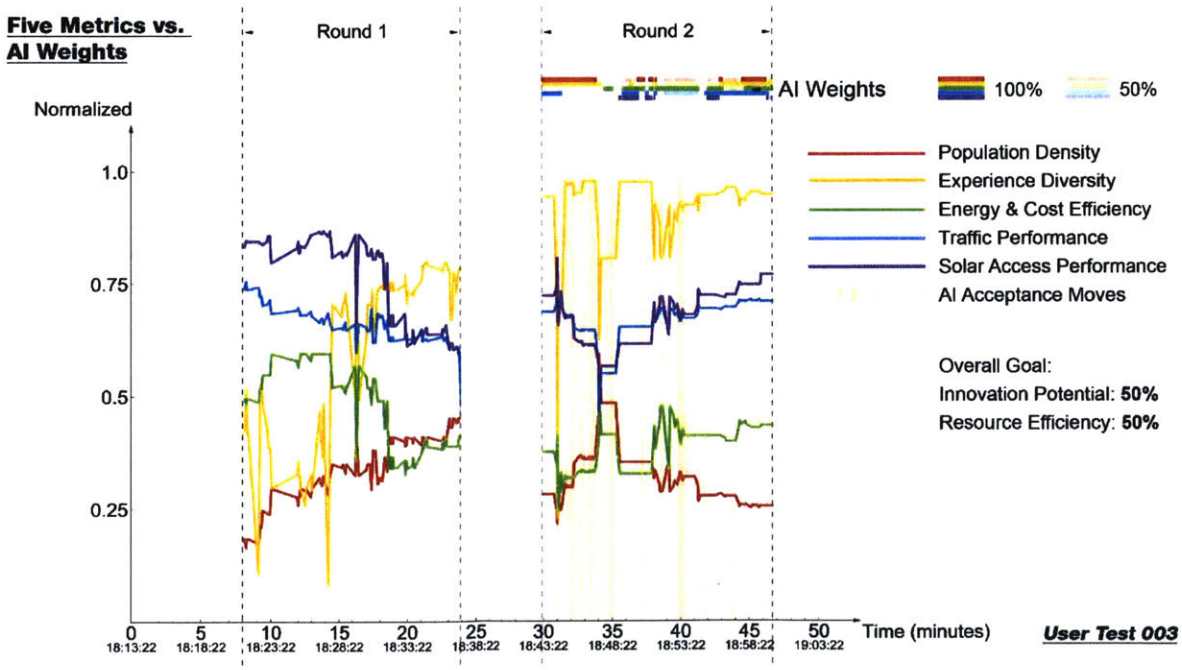


Figure 44: Five Metrics vs. AI Weights of User Test 003

B.4 User Test 004

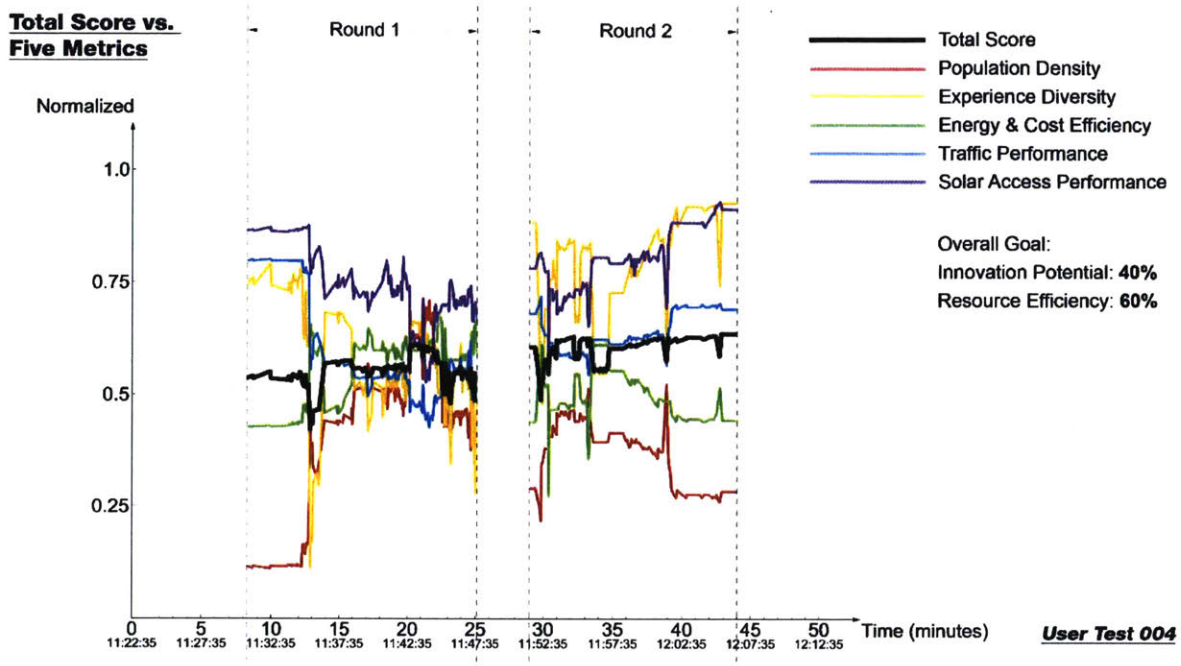


Figure 45: Total Score vs. Five Metrics of User Test 004

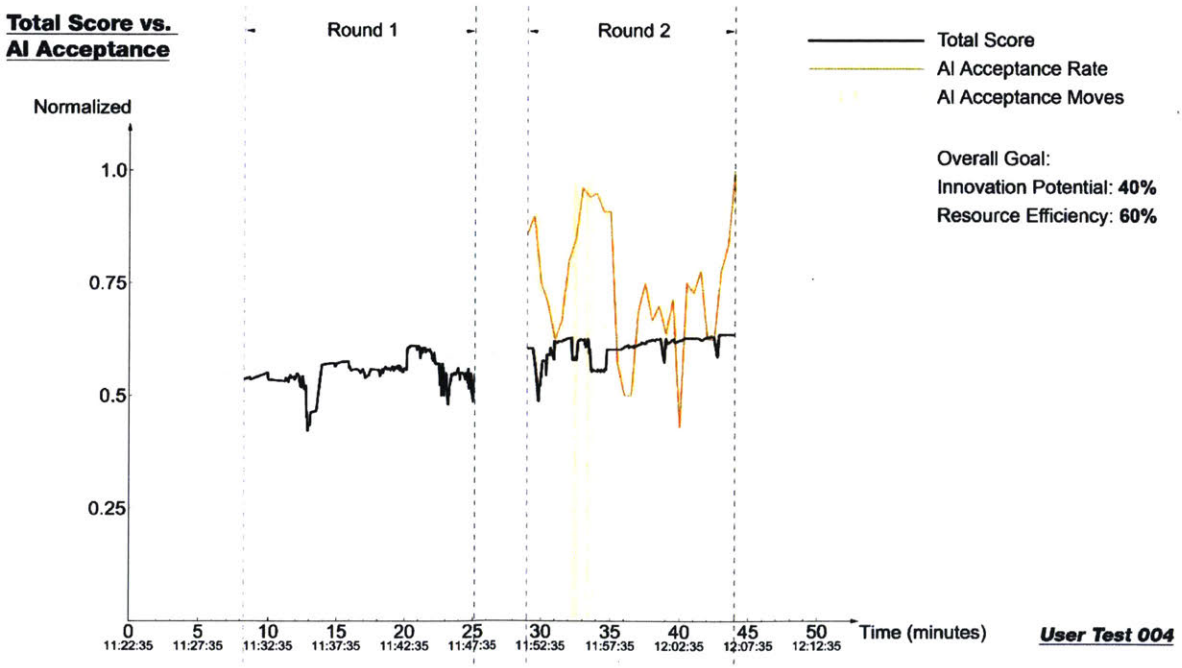


Figure 46: Total Score vs. AI Acceptance of User Test 004

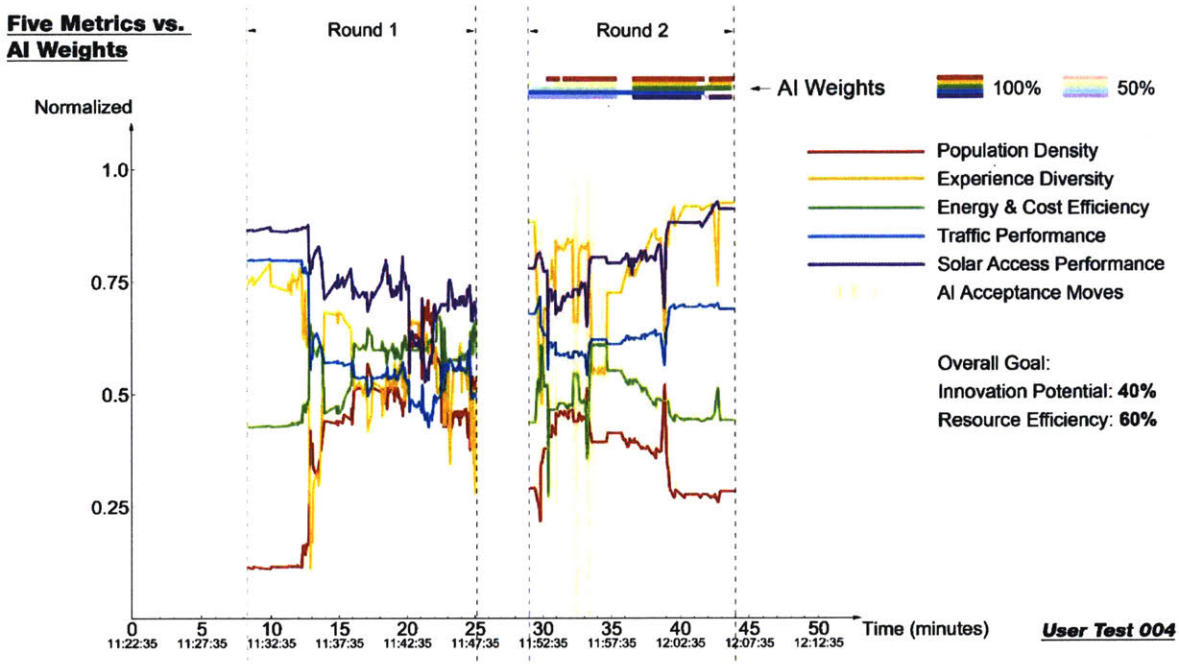


Figure 47: Five Metrics vs. AI Weights of User Test 004

B.5 User Test 005

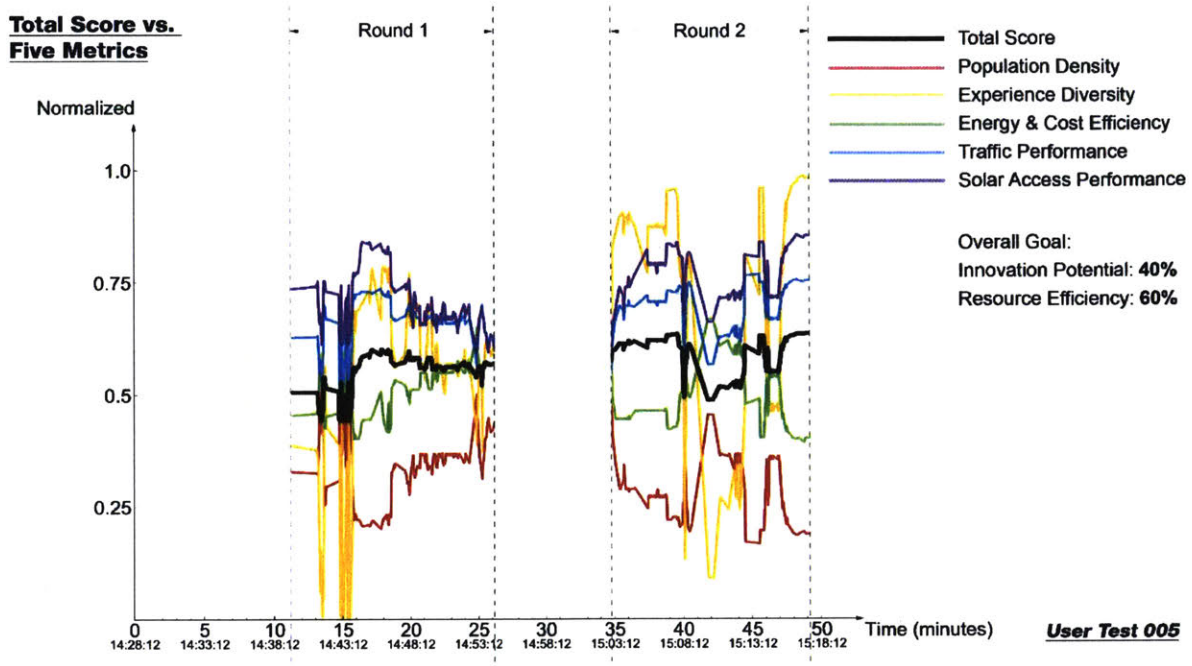


Figure 48: Total Score vs. Five Metrics of User Test 005

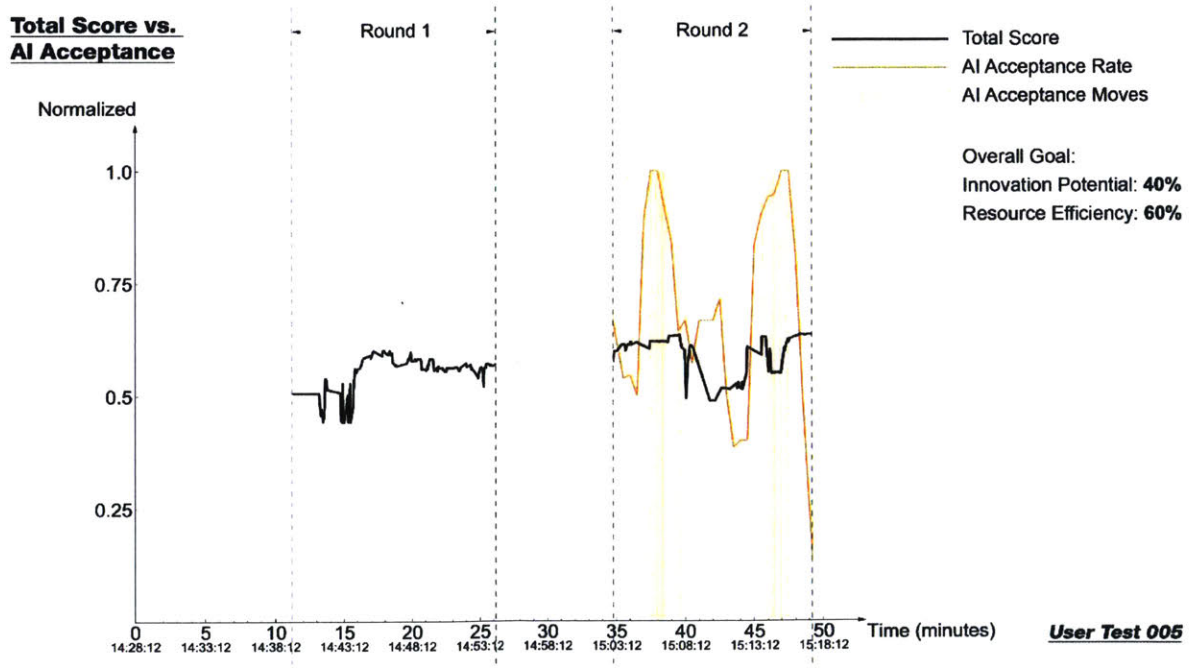


Figure 49: Total Score vs. AI Acceptance of User Test 005

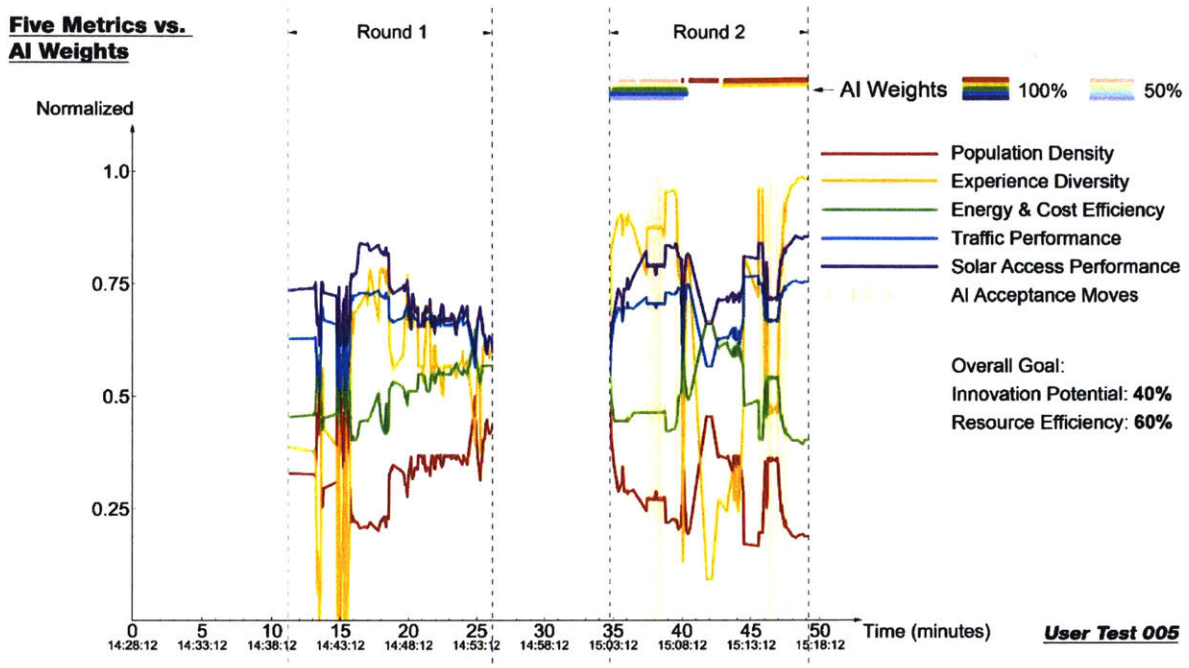


Figure 50: Five Metrics vs. AI Weights of User Test 005

Appendix C

Questionnaires

C.1 Before-Test Questionnaire Form

Q-1 Form

CityMatrix Before Study Questionar

You name/nick name:

Imagine you are taking part in a public engagement event for urban development of Kendall neibourhood. Which do you think is more important? To increase the Innovation Potential (creating an environment favor for tech, social, or business invention, enterpreneurship, etc) or to increase the Resource Efficiency (reducing engergy consumption, traffic conguestion, promoting sustainable energy, etc)? Please weight the priority with percentage numbers (total should be 100%):

Innovation Potential: _____%

Resource Efficiency: _____%

C.2 After-Test Questionnaire Form

Q-2 Form

CityMatrix After Study Questionar

You name/nick name:

How much do you agree: the user interface is overall easy to understand? (0 to 10)

How much do you agree: the heat-map is helpful to understand the impact of your move? (0 to 10)

How much do you agree: the radar-chart is helpful to understand the impact of your move? (0 to 10)

How much does the system help you understand better how city works? (0 to 10)

Have you had enough time to finish building your city in the first round? (Yes/No)

Have you had enough time to finish building your city in second round? (Yes/No)

Are you satisfied with your design/have you achieved your goal about city in the first round? (0 to 10)

Are you satisfied with your design/have you achieved your goal about city in the second round?(0 to 10)

How helpful do you think are the suggestions made by the AI? (0 to 10) If >=5 , how are they helpful?

Have you learn anything from the suggestions made by the AI? (0 to 10) If >=5, what did you learn?

1) How much do you agree: The suggestions made by AI is helpful for achieving our goal about the city district faster? (0 = totally disagree, 10 = totally agree)

2) I am more satisfied with the design of the city district in the second round because _____. (Please select all the applicable items, you can select multiple items):

a. as we spent more time working together, we formed a better collaboration;

b. as I spent more time working on the system, I got more familiar with it;

c. the suggestions made by AI was helpful;

d. other reason; (please specify)

e. Compare to the first round, I am NOT more satisfied with the design of the city district in the second round.

How much do you agree: sometimes I tend to follow its instructions blindly? (0 to 10)

Do you feel that the AI suggestions make you think less or more about the city? (0: less, 10: more)

How much do you agree: CityMatrix system promotes collaboration in urban decision making? (0 to 10)

Any comments or suggestions for improving the system?

Bibliography

[1] Grimm, Nancy B., et al. "Global change and the ecology of cities." science 319.5864 (2008): 756-760.

[2] Midgley, James, et al. Community participation, social development and the state. Methuen, 1986.

[3] Howard, Philip. "Two years not ten years, redesigning infrastructure approvals." Common Good (2015).

[4] Katz, Bruce, et al. "Rise of Innovation District." Brookings (2014).

[5] Kononenko, Igor. "Machine learning for medical diagnosis: history, state of the art and perspective." Artificial Intelligence in medicine 23.1 (2001): 89-109.

[6] Hansen, James V. "Genetic search methods in air traffic control." Computers & Operations Research 31.3 (2004): 445-459.

[7] Campbell, Murray, A. Joseph Hoane, and Feng-hsiung Hsu. "Deep blue." Artificial intelligence 134.1-2 (2002): 57-83.

[8] Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

[9] Schumaker, Robert P., and Hsinchun Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system." ACM Transactions on Information Systems (TOIS) 27.2 (2009): 12.

[10] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[11] Pomerleau, Dean A. "Efficient training of artificial neural networks for autonomous navigation." Neural Computation 3.1 (1991): 88-97.

[12] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.

[13] Underkoffler, John, and Hiroshi Ishii. "Urp: a luminous-tangible workbench for urban planning and design." Proceedings of the SIGCHI conference on Human Factors in Computing Systems. ACM, 1999.

[14] Sparacino, Flavia, et al. "Technologies and Methods for Interactive Exhibit Design: From Wireless Object & Body Tracking to Wearable Computers." ICHIM. 1999.

[15] Larson, Kent. "Louis I. Kahn: unbuilt masterworks." The Monacelli Press, 2000.

[16] Winder, I. J. "System for real-time digital reconstruction and 3D projection-mapping of arbitrarily many tagged physical objects." US Provisional Patent (2015).

[17] Alrashed, Tarfah, et al. "An observational study of usability in collaborative tangible interfaces for complex planning systems." Procedia Manufacturing 3 (2015): 1974-1980.

[18] Jacobs, Jane. The death and life of great American cities. Vintage, 2016.

[19] Shannon, Claude E., Warren Weaver, and Arthur W. Burks. "The mathematical theory of communication." (1951).

[20] Peet, Robert K. "The measurement of species diversity." Annual review of ecology and systematics 5.1 (1974): 285-307.

[21] Kröner, Andreas, et al. "DIVA—an open architecture for dynamic simulation." Computers & chemical engineering 14.11 (1990): 1289-1295.

[22] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12.Oct (2011): 2825-2830.

[23] Neter, John, et al. Applied linear statistical models. Vol. 4. Chicago: Irwin, 1996.

[24] Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." The American Statistician 46.3 (1992): 175-185.

[25] Liaw, Andy, and Matthew Wiener. "Classification and regression by RandomForest." R news 2.3 (2002): 18-22.

[26] Reas, Casey, and Benjamin Fry. "Processing.org: programming for artists and designers." ACM SIGGRAPH 2004 Web graphics. ACM, 2004.

[27] Grignard, Arnaud, et al. "GAMA 1.6: Advancing the art of complex agent-based modeling and simulation." International Conference on Principles and Practice of Multi-Agent Systems. Springer, Berlin, Heidelberg, 2013.

[28] GAMA website: http://gama-platform.org (2016), <http://gama-platform.org>.

[29] Arnaud Grignard, Alexis Drogoul. "Agent-Based Visualization: A Real-Time Visualization Tool Applied Both to Data and Simulation Outputs." The AAAI-17 Workshop on Human-Machine Collaborative Learning. AAAI Technical Report. 2017.

[30] Niepert, Mathias, Mohamed Ahmed, and Konstantin Kutzkov. "Learning convolutional neural networks for graphs." International Conference on Machine Learning. 2016.

[31] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).

[32] Chollet, François. "Keras (2015)." URL http://keras.io (2017).

[33] Browne, Cameron B., et al. "A survey of monte carlo tree search methods." IEEE Transactions on Computational Intelligence and AI in games 4.1 (2012): 1-43.