EXPLOITING GAME DEVELOPMENT ENVIRONMENTS FOR RESPONSIVE URBAN DESIGN BY NON-PROGRAMMERS

Melding real-time ABM pedestrian simulation and form modelling in Unity 3D

XIAORAN HUANG[1], GEOFF KIMM[2] and MARK BURRY[3]
[1]*North China University of Technology*
[1]*xiaoran.huang@ncut.edu.cn*
[2,3]*Swinburne University of Technology*
[2,3]*{gkimm|mburry}@swin.edu.au*

Abstract. Precinct-level pedestrian simulation often requires moderate to high-level modelling skills with a steep learning curve, and is usually non-flexible, time-consuming and exclusive of the broader public community. Confronting these problems, our research investigates a novel and agile workflow to test precinct pedestrian behaviours by melding agent-based simulation (ABM) and responsive real-time form modelling mechanisms within accessible visualisation of city and precinct environments in a game engine, Unity 3D. We designed an agent system prototype of configurable and interoperable nodes that may be placed in an urban modelling scenario. Realtime CSG, a fast polygon-based modelling plugin, is also introduced to our workflow where users can use the evidence observed when running a scenario to quickly adjust the street morphology and buildings in response. In this process, end users are kept in the design loop and may make critical adjustments, whereby a responsive, collective, informed design agenda for our built environments can inform more detailed outcomes of pedestrian behaviour and action and promote more efficient collaborations for both professionals and local communities.

Keywords. Agent-based pedestrian simulation; responsive modelling; computer-aided urban design; public participation.

1. Introduction

In urban design practice, there is a growing need for professional specialisations to be placed within a broader transdisciplinary concept of urban designing and, vice versa, urban design must be collaboratively conceptualised through the sum of the unique disciplinary perspectives (Burry and White 2020). Concomitantly, the digital technologies that have driven much recent professional speciation have also fostered the importance of simulation for better evaluation of built environment scenarios and creation of generative outcomes (Nguyen et al. 2014, Huang et al. 2017). In such simulations, games technology is used increasingly for its immersive user interaction, ready extensibility, visual descriptiveness, and

PROJECTIONS, Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2021, Volume 2, 689-698.

flexible kit of parts (Indraprastha and Shinozaki 2009). The confluence of those three longstanding and strengthening trends in the urban design process supports investigation of exploiting game development environments for responsive urban design by non-programmers.

This paper presents a workflow for real-time urban simulation and responsive design within game engine development environments. The approach makes novel use of a game development environment to, rather than build and export a game for external or standalone use, embed the game and user experience itself directly within it by running in effect in "debug" mode. The process enables an integrated conjecture-test-refine workflow, accessible to non-experts from diverse backgrounds, of responsive, flexible urban simulation and design for advocating sustainable and liveable urban design schemes. The paper is structured as follows. Section 2 presents the rationale for the use of walkability and agent-based modelling (ABM), and the Unity 3D (Unity) game engine environment in its development and testing. Section 3 outlines the implementation of a prototype design computing tool ("the prototype") based on this workflow in Unity with emphasis on its design for accessibility by the non-programmer. Section 4 discusses the testing of the prototype in a one-day international workshop in which participants engaged in a series of design explorations using rapid, iterative visualisation with agent-based pedestrian modelling and related simulation techniques. Finally, a conclusion presents the key findings.

2. Exploiting Game Development Environments

2.1. UNITY 3D

Unity is used in this research for reasons of relative accessibility to the novice user, integrated manipulation of a game's 3D environment and running of that game directly within the development interface, and allowance for scripting custom logic in the C-SHARP programming language.

The flexibility and extensibility of the development environment of Unity have been exploited extensively by expert users in a broad gamut of built environment simulation and analysis research applications. In computer science, it has been used to generate training data for deep learning models to recognise components of urban scenes (Ros et al. 2016). In governance, it has been used in developing citizen-led interactive modelling of smart villages to identify and prioritise needs and novel solutions (Kimm and Bury 2020). In engineering, it has been used at building, precinct, and city scales to integrate external and embedded modelling tools for applications such as multi-modal traffic simulation (Olaverri-Monreal et al. 2018). In studies of pedestrian movement, Unity has been used for diverse purposes including VR-enabled modelling of pedestrian interaction with traffic, and the dynamics of social interactions in interior spaces (Orlosky et al. 2015, Pedica and Vilhjálmsson 2018).

While Unity is being increasingly used in research by adept users, a literature review indicates its potential as a non-expert or pedagogical built environment modelling tool is as yet understudied. Although Unity Technologies' 2020 release of the Bolt visual scripting add-on for Unity brings closer the accessibility seen in tools such as Grasshopper for the Rhinoceros3D application, it and Unity itself are not targeted specifically to design or built environment issues. Furthermore, the development environment of Unity is designed to be simple to learn but nonetheless is not readily usable by the neophyte user.

The objectives of this research therefore require the development of a user experience that utilises the components of, and sits within, the existing development environment to create a simple, intuitive, and design-focused interface. Creating the prototype within Unity's development environment, rather than developing it as a standalone app, exploits the existing user interface and reduces the resources required for its design and implementation.

2.2. WALKABILITY / ABM

Over the past decade, walkability has been widely considered a critical component of urban liveability and sustainability. As a result, facilitating pedestrian-friendly environments has become urgent for many urban design and planning agendas which require full engagement from different stakeholders (Aschwanden 2014).

To understand the walking experience with more profundity, many precinct-level pedestrian simulations have investigated the relationship between crowd patterns and urban typologies in recent studies (Asriana and Indraprastha 2016). However, these approaches are often less accessible for non-programmers due to the technical nature of ABM modelling and scripting. Therefore, walkability, and its simulation via ABM, was selected as the focus of a design computing tool for its ready, integrated support and accessibility in Unity and importance to effective urban responses to urgent environmental emergencies.

3. Prototype Development

A walkability ABM prototype was developed in the Unity development environment. Four key components were built on: the *hierarchy window*, the *scene view*, the *inspector window*, and the *game view* (Figure 1).

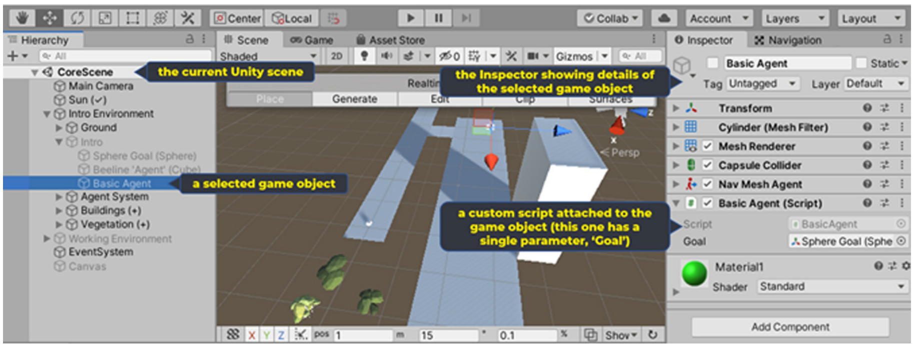

Figure 1. 1) Unity's hierarchy window, 2) the scene view with a game in progress showing spawn points (black cubes), agents, and sensors (orange columns), 3) the inspector window with details displayed of the Agent Spawner game object, and 4) the tab of the game view.

1. The hierarchy window presents a tree structure of all items in a Unity game or scene such as lights, geometry, or pedestrian agents. Each item is a *game object*. A game object may be empty, in which case it contains basic attributes including a name and a 3D position, and may have attached to it additional components and scripts. Scripts can add logic or actions to a game object; for example, a game object may be told to move in a certain direction when another game object is nearby. All game objects are a child of the scene itself or of another game object.
2. The scene view displays the game objects of the hierarchy window in 3D space. A user may directly select and reposition game objects in this view.
3. The inspector window displays the attributes of game objects selected in the hierarchy window or scene view. Those attributes, such as a parameter of a custom script, may be manually adjusted by the user within the inspector's UI.
4. The game view displays the game when it is run. The game is the sum of all game object presences and actions in the scene and the interactions of a user.

3.1. IMPLEMENTATION IN UNITY 3D

A framework for the proposed essential simulation workflow is encapsulated in the Unity scene within a single empty object named as the *Agent System* (Figure 1). Under this Agent System is an array of modules of game object and attached custom script pairs ("simulation modules") that each provides one aspect of the essential functionality and user interaction.

To facilitate non-expert participation, a design objective for the prototype was to simplify all user interaction. Unity is a professional tool: inherent in using it directly for participatory modelling is a tension between its sophistication and the potential inexperience of participants. We applied three principles to address this.

1. First, only a minimum necessary set of custom script functionality should be exposed in the inspector window so to not overwhelm the user.
2. Second, full interaction should be supported by as few UI elementary operations as possible. A user can fully interact with the simulation modules with only two operations: the creation or deletion of empty game objects within the scene, whose location within the scene's tree structure gives their meaning; and the adjustment of the behavioural control of custom scripts via the inspector window.
3. Third, what a participant should interact with should be explicitly clear. A schema of symbols was used to indicate what UI elementary operations may be taken on a simulation module (Figure 1). A plus symbol on a game object means child game objects may be added or removed. A tick on a game object means there's an adjustable custom script attached and on a custom script parameter means it's safe to adjust. A cross on any element warns it should not be touched.

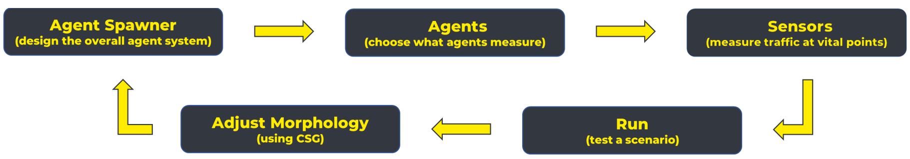

Figure 2. Proposed ABM-Responsive Modelling workflow in Unity 3D.

Three simulation modules are provided to the participant and control an agent's lifecycle and interaction with its environment: *Agent Spawner*, *Agents*, and *Sensors* (Figure 2). Together, the three modules, along with Realtime CSG, form the reification in Unity of the proposed essential workflow, as seen in the scene view of (Figure 1). Each workflow step is discussed below.

1. The Agent Spawner simulation module controls the spawning rate and maximum number of agents as well as essential characteristics of each newly spawned agent (Figure 1). New agent spawning nodes are assigned by a UI elementary operation: the 3D location of any game object placed under the Agent Spawner is automatically used to place agent spawn nodes when the scene is run. A spawn node game object is both a start and end point by default; a user may attach an optional *Agent Spawn Node* script to specify other behaviour.
2. The *Agent Controller* script of the Agents simulation module sets what illustrative visual guide an agent will display of its 'satisfaction' with the environment as it navigates between its birth and death spawn nodes. Four experience metrics are provided that are intended to be intuitive to understand and to cover the needs of common scenarios: *distance*, *speed*, *directness*, and *proximity*. An agent's satisfaction is indicated by its colour graduated between user-configurable 'good' and 'bad' colours. The distance metric considers an agent's maximum ideal journey distance; an agent exceeding this distance will become increasingly dissatisfied. The speed metric considers an agent's acceptable decrease in speed before its satisfaction starts degrading; an agent may be slowed if the street morphology contains chokepoints. The directness metric considers an agent's acceptable percentage increase in travel distance over the beeline distance; a street morphology that is circuitous or closed may rate poorly on this metric. The proximity metric considers an agent's personal space; an agent will be less satisfied the closer other agents are within this radius, and a street morphology that does not properly accommodate pedestrian volumes may also rate poorly.
3. Sensors added via the Sensors simulation module track pedestrian street use and are placed within the simulation in the same way as spawn nodes. A sensor in-game displays a dynamically updating 3D column whose height matches a count of passing agents. Two key parameters are exposed in the *Sensor Controller* script to facilitate users' particular hypothesis tests. *Range* determines the physical extent of the sensors' detection fields and allows fine-tuning of the granularity of a web of sensors in scene. *Window* sets the number of seconds for which to smooth data as a moving average: a user may hence see instantaneous feedback or results smoothed over changing simulation conditions. Sensors placed strategically within the scene may provide an overview of how a street morphology is functioning, and their dynamic response was developed as a precursor to the implementation of generative urban design functionality.
4. Initiation and scenario test: a participant may run the simulation after setting up their hypothesis. Once running, they may engage in on-the-fly editing of the simulation and dynamic moderation of their hypothesis testing. Switching to the scene view allows real-time repositioning of simulation elements such as spawn nodes. Similarly, the participant may change parameters of the custom scripts; within a single simulation session a scenario to test a particular street morphology may, for example, be 'stress tested' by greatly increasing the agent spawning rate.
5. Morphology adjustment: participants can exploit the evidence observed when

running a scenario to adjust city block and building envelope geometry. Constructive solid geometry (CSG) is used for this as it enables intuitive modelling and human-friendly interaction through separation of the creation of geometric primitives from the logic of how they combine into complex solids (Rossignac 1987). In CSG, geometric primitives are created relative to each other by the user who determines their interactions of boolean operations of union, intersection, and difference. This research uses Realtime CSG (realtimecsg.com), a third-party Unity tool, that integrates directly into the scene view (Figure 3).

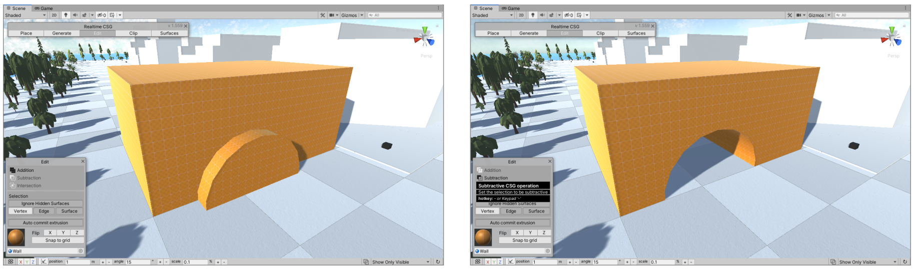

Figure 3. Built environment form adjustment using Realtime CSG in Unity.

4. Workshop Case Study

The proposed framework and ABM prototype were introduced and tested at an August 2020 one-day workshop held at an international conference. Participants were asked to engage with a series of design explorations using rapid, iterative modelling and simulation. The workshop objectives were twofold. The proximate aim was to investigate the capacity to design walkable communities to encourage collective responsibility for our planet and effective responses to environmental emergencies. The distal aim was to test the subject framework of this research through participants collaboratively developing their understanding of the use of responsive, custom modelling in the design of better-built environments.

4.1. WORKSHOP STRUCTURE

Due to COVID-19, the workshop was conducted fully online via Zoom. Fifteen participants were registered as active candidates and the group was diversified geographically and demographically (one-third female; participation numbers of 6 from Asia, and 3 each from Australasia, Europe, and elsewhere). No expertise with Unity was required. However, participants were asked to preinstall it, download an implementation project file, and review an instructional "cheat sheet".

In accordance with the objective of accessibility of the computational framework, the workshop began with a morning two-hour skilling-up session as a complete introduction. Essential concepts of ABM and walkability were covered, along with a live "how-to" demonstration of the Unity environment and proposed framework. One virtual site of investigation, pre-packaged within the Unity implementation project, was demonstrated and offered as an exemplar based on the Barcelona superblocks urban scenario.

Preceding a midday break, active participants were divided into three groups based on their common interests in site selection or design proposition. Within those groups, all members were encouraged to explore different design directions and urban morphologies, and each member could have alternative or branching revised urban form propositions based on their personal investigation.

In the afternoon session, groups were assigned into breakout rooms (virtual sub-meetings within Zoom) to discuss and collaborate on their projects closely, while tutors jumped between different rooms providing hands-on assistance. Within their groups, candidates were asked to confirm the site and the type of urban morphology they would explore, and then probe possible design investigations through the proposed ABM-responsive modelling framework in an iterative design process. Urban morphology iterations were saved directly in the Unity projects themselves by simply duplicating their working environment root game object and deactivating the old copy. Hence, histories of essential geometries and agent system configurations were captured.

5. Results

5.1. CASE 1: FROM BARCELONA SUPERBLOCK TO ADELAIDE CBD

This group had relatively low experience with Unity but built essential skills quickly with tutor assistance. Some technical challenges delayed Unity project environments set up due to script incompatibilities arising from installation of different versions and a corrupted central project repository. Subsequently, they properly set up the simulation framework using the given Barcelona example project, and then tested alternative design possibilities by ‘injecting’ internal pedestrian paths using the framework introduced to them in the morning session.

The difficulties of the initial stage constrained the time available for their own design proposals; the group hence selected the CBD of Adelaide, Australia, as their site of investigation. Adelaide is renowned to designers for its unique grid pattern and hierarchical street layouts. Group members exploited this typological similarity to the Barcelona superblocks example and repurposed some of the critical agent settings from their Barcelona studies through a modification process. Realtime CSG integration also allowed them to expeditiously remodel urban massing and therefore novel design investigations were possible within the restrictive timeframe.

During iterations, they proposed and tested many pedestrian-oriented design strategies, including but not limited to: widening footpaths, proposing ground-floor pedestrian connections through podia, and suggesting 3D circulation with ramps and sloped landscapes (Figure 4). This case study demonstrated how flexible and accessible the proposed design framework could be for supporting fast iterative design within an intense schedule in pragmatic design practices.

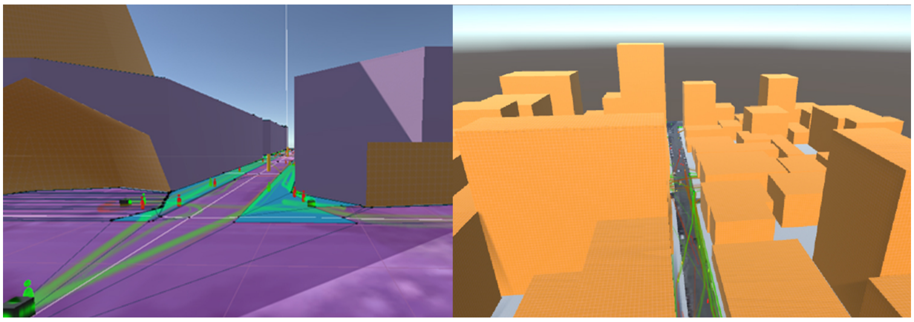

Figure 4. Left: Barcelona Superblocks test; Right: New proposed design scenario in Adelaide.

5.2. CASE 2: A RESIDENTIAL COMMUNITY NEAR THAMES RIVER, LONDON

This group was exceptionally experienced in computer-aided design and had a clear objective of seeking possible integrations between agent-based pedestrian modelling and their main research interests and expertise. Their workshop case study originated in one member's existing pedestrian density study of the Munich city. In it, a workflow of data collection, visualisation, and analysis was undertaken in which Rhino modelled city infrastructure and Python collected core data including Twitter geospatial usage. That workflow was applied in their workshop case study of London's East India Dock residential precinct. Subsequently, walkability costs of street spaces to differentiate more and less walkable spaces were defined according to the analysis results. Twitter geospatial data was used to determine the active population density in the selected site and adjust the Agent Spawner module for agent spawning rate and overall count.

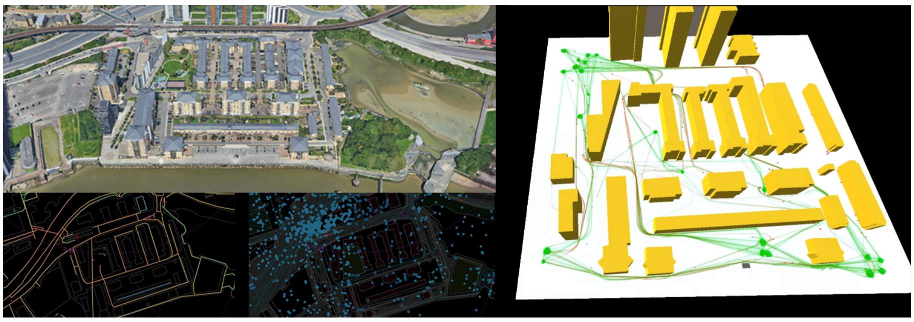

Figure 5. Left: The geographical information, population density, and Twitter signalling data were collected and analysed. Right: The running simulation scene in Unity 3D.

During simulations, the group explored precinct redesign to enhance local walkability. For instance, their prior big data collection revealed people prioritise a more walkable environment in the precinct's Virginia Quay Park area and this was verified with the agent experience metrics visualisation of the Agents simulation module. A chokepoint in the west of the precinct, and hence less walkable, was still traversed by many agents due to its proximity (Figure 5), and this is a piece of critical evidence that could support an alternative design blueprint. Similarly, other pedestrian-related issues, such as travelling time, were also identified through the iterative investigation process. Through this case study, the proposed design framework has proven a high extensibility and configurability when melding with other digital techniques and workflow and can be mutually verified with big data analysis and other research methods.

5.3. CASE 3: SHANGHAI NORTH BUND

This group engaged in a highly collaborative process and at the outset drafted a schedule with a distributed workload. Our proposed framework allowed them to split a complete procedure into individually executable tasks: site investigation and data collection, built environment modelling in Unity, ABM system set up, any script calibration, visualisation, and documentation. In design iterations, each member could contribute ideas by editing urban morphologies through Realtime CSG within minutes to be saved as a unique scene under the same project.

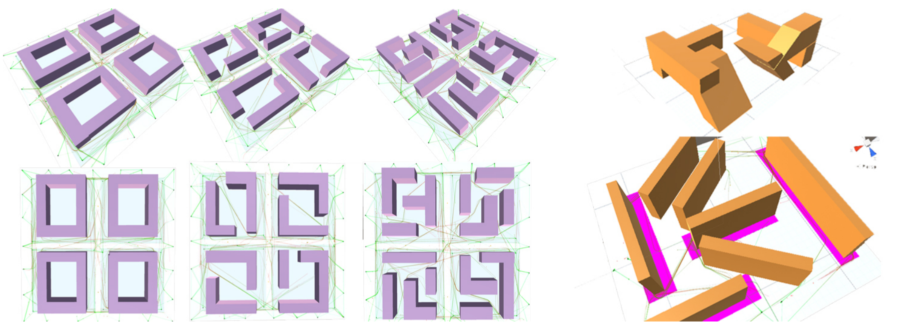

Figure 6. Left: Closed, through, and spread typical block morphologies in Shanghai North Bund district with the simulation results; Right: Morphologies emerged from the rapid-fire ideation process.

In their design investigation, *closed*, *through*, and *spread* typical residential block morphologies were selected from the Shanghai North Bund district to probe how urban form variations could influence pedestrian behaviours in a grid-based urban layout. In all simulations, the agent numbers and spawning positions were intentionally kept constant, and the average travel time and distance were reduced with the increment of the block permeability. This was not a novel approach per se, yet with the rapid modelling tool and configurable ABM system, they were able to recalibrate the urban massing by rotation, scaling, trimming or duplication and investigate the potential impact to pedestrians accordingly. More creative morphologies also emerged during the participative design process. Slopes, curvilinear compositions, and different visualisation options were enabled in a rapid-fire ideation process due to the simplified modelling and simulation procedure powered by the Unity platform (Figure 6).

6. Conclusion

In this research, we augment and incorporate concepts of digital simulation and pedestrian-centric design in the built environment by exploiting game development environments with agent-based modelling. A flexible and real-time ABM pedestrian simulation system is proposed and developed in Unity 3D, which has become readily available for the participation and contribution of non-programmers in the process, consequently increasing the design transparency. We also demonstrate how conventional urban form modelling and walking simulation can be integrated and reshaped by adopting our rapid ABM-responsive modelling workflow, addressing a range of spatial and temporal urban walkability concerns. The flexibility of our approach and its successful demonstration in the case studies suggest significant potential for exploiting game development environments in making highly configurable digital urban models accessible for public or non-expert participation and promoting responsive design and impact projection in global urban scenarios.

References

Aschwanden, G.: 2014, *Health and Place: an analysis of the built environment's impact on walking behavior and health*, Ph.D. Thesis, ETH Zurich.

Asriana, N. and Indraprastha, A.: 2016, Making sense of agent-based simulation: Developing design strategy for pedestrian-centric urban space, *Proc. of the 21st CAADRIA*.

Burry, J. and White, M.: 2020, From 'T'to 'pi'-shaped people: Better Urban Practice through Dual Depth in Architecture and Planning Education, *Architectural Design*, **90**(3), 114-121.

Huang, X., White, M. and Burry, M.: 2017, A pedestrian-centric design strategy: melding reactive scripting with multi-agent simulation, *in Proc. of SIMAUD2017*, 1-8.

Indraprastha, A. and Shinozaki, M.: 2009, The investigation on using Unity3D game engine in urban design study, *Journal of ICT Research and Applications*, **3**(1), 1-18.

Kimm, G. and Burry, M.: 2020, Encouraging Community Participation in Design Decision-making through Reactive Scripting, *Proc. of the 25th CAADRIA*, 51-60.

Nguyen, A.T., Reiter, S. and Rigo, P.: 2014, A review on simulation-based optimization methods applied to building performance analysis, *Applied Energy*, **113**, 1043-1058.

Olaverri-Monreal, C., Errea-Moreno, J., Diaz, A., Biurrun-Quel, C., Serrano-Arriezu, L. and Kuba, M.: 2018, Connection of the SUMO microscopic traffic simulator and the Unity 3D game engine to evaluate V2X communication-based systems, *Sensors*, **18**(12), 4399.

Orlosky, J., Weber, M., Gu, Y., Sonntag, D. and Sosnovsky, S.: 2015, An interactive pedestrian environment simulator for cognitive monitoring and evaluation, *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, 57-60.

Pedica, C. and Vilhjalmsson, H.: 2018, Study of nine people in a hallway: Some simulation challenges, *Proc. of the 18th ICIVA*, 185-190.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D. and Lopez, A.M.: 2016, The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3234-3243.

Rossignac, J.R.: 1987, Constraints in constructive solid geometry, *Proceedings of the 1986 workshop on Interactive 3D graphics*, 93-110.