**PRIMARY RESEARCH**                                             **OPEN ACCESS**
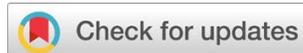
# Space Syntax in Mixed Reality Gaming Applications

Shivam Mahajan, BCS [1]*, Jake Moore, BCS [1], Isaac Fresia, MArch, BEDS, BFA [2],
Juliano Franz, BEng [1], Joseph Malloch, PhD [1], Derek F. Reilly, PhD [1]

[1] Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada B3H4R2
[2] Faculty of Architecture, Dalhousie University, Halifax, Nova Scotia, Canada B3H4R2

*Corresponding Author: shivam.mahajan@dal.ca

**Abstract**
With the rapidly growing interest in AR, grows the motivation to overcome some of the problems facing the implementations of the technology. The main challenge encountered in the building of large-scale mixed reality AR games is the uniqueness of the spatial settings in which the game will be experienced by the user. Game designers will require data of the spatial settings to determine game object placement, events and narrative flow. The problem arises because the designers are not aware of the physical environment in which the game will be played. In our research, we address this problem and take an approach to solving it by using Space Syntax techniques. We demonstrate the use of this technique, using a proof-of-concept game called Adventure AR.

**Keywords:** space syntax; mixed reality games; augmented reality; spatially adaptive systems

**Introduction**
With no prior knowledge of the physical spaces in which the game will be experienced, a mechanism is needed, to allow level design to adapt to different physical environments. To do this, our research seeks to use Space Syntax analysis to define the spatial structure of the area in which the game will be played and use the results to place game assets according to the design of the environment. In this paper we will focus on acquiring and encoding the Space Syntax data that includes the connectivity, visual complexity, and openness graphs. Using this data and a 2D/3D model of the floor or region the game will be played in, we will develop a RGBA bitmap representing Space Syntax values. A calibration step is performed for scaling the real world to the virtual space. Finally, the retrieved data for the spatial attributes (Visual Complexity, Openness, connectivity) can be displayed on the Heads-up-display (HUD).

Furthermore, we developed a proof-of-concept game called Adventure AR to explain how the Space Syntax data is implemented to determine the placement of spawning game objects. This game is developed in Unity for the Microsoft Holo-Lens. This paper discusses the methodologies to retrieve the Space Syntax data for a given space, and then apply that to dynamically place assets in that region.

Space syntax [1] involves the study of the relationship between the geometric and relational structure of built environments (buildings, cities) and experience of (and behaviour within) the environment. Actively developed over the past decades, Space syntax is not a single method, but a family of graph-theoretic approaches that describe the structure of a space from the perspective of experience [2].

**Methods**
We are using attributes of Space Syntax such as connectivity, visual complexity, and openness, where connectivity represents how well a region in space is connected to another region [3], visual complexity represents how much area can be seen from one point in space and openness is defined by how open the area is [4]. Values of these attributes are then calculated for a given space and are scaled to the ratio of 255. These individual attribute values are then mapped to an RGBA data structure where blue represents connectivity, green represents visual complexity, and red represents openness (as shown in Figure 1). At the start of the gameplay, the spaces are calibrated by scanning a static object. We use Vuforia, an object recognition API to retrieve coordinates of this static object. The algorithm will then drift the virtual space and map it with the real environment. Finally, the Space Syntax data of any given point in the calibrated space can be retrieved through this RGBA data structure (as shown in Figure 2).
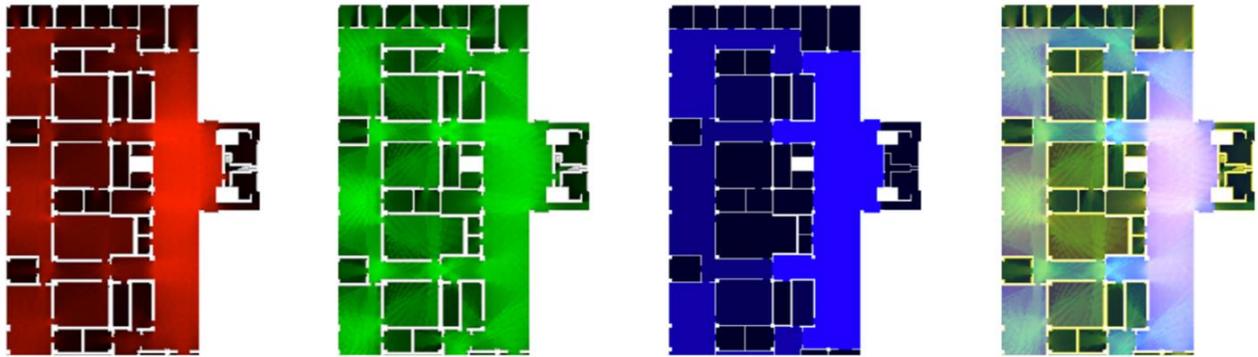
**Figure 1:** Space Syntax attributes laid on top of a floor plan. The rightmost figure is a composite encoding of all three attributes.



**Figure 2:** Values of SS attributes on HoloLens HUD. The color of floor corresponds to composite RGB representation of SS attributes. These values are updated on HUD when wearer navigates the environment.

## Results

Our research is focused on the fundamental issue facing the development of large-scale mixed reality, immersive AR games. In this paper, so far, we have discussed about the methods we have followed to retrieve the Space Syntax data. But how is this data used to distribute spawning game objects around the gaming area?

A proof- of concept game Adventure AR, based on the classic *Atari* console game *Adventure*, is used to test the results of the research that we have carried out. The rules of the game are simple and concise. It requires the player to navigate around the space looking for keys which can open chests that are distributed among the region, while avoiding contact with roaming Artificial Intelligence Monsters. The game is over when the player collides with a monster or when all the chests are opened. Now, the placement of these game assets (Keys, Chests and monsters) will vary according to the spatial settings of the gaming environment. This is the part where the attributes of SS (openness, visual complexity and connectivity) are used to determine the placement of the spawning game objects.

As soon as the game loads and the spaces get calibrated, the RGBA bitmap is loaded. Alongside this, a configuration file is loaded that identifies which room should hold the game objects, and rules for placing the chest and key assets. Once the game ends, the configuration is randomized to generate a new set of location for the game assets keeping the rules of placement constant. For instance, the keys are likely to be placed in areas with low values of openness, visual complexity and connectivity. Similarly, the chests will be placed in spaces with a higher value of these attributes. The game predefines ranges for "low", "medium" and "high" for the spatial attributes. If the location has enough spaces for game object placement, the objects are randomly placed within those spaces. In case, the candidate locations are not enough according to the placement rules, the game won't play, and the rules will need to be changed, or a new location found.

## Conclusions

This paper defines a way to derive and encode Space Syntax data to dynamically place game objects in a given space. In the proof- of concept game Adventure AR we have

demonstrated the use of these Space Syntax values to place objects in different locations. For future work, we are working on building a robust toolkit to support more complex constraints of immersive AR games. We believe that Space Syntax is the key to looking at Mixed- reality games beyond the dynamic adaptation. And in future work, we hope to achieve results that can significantly change the experience for these mixed reality AR games.

**List of Abbreviations**
API: Application Program Interface
AR: Augmented Reality
HUD: Heads-up-display
RGB: Red Green Blue
RGBA: Red Green Blue Alpha
SS: Space Syntax

**Conflicts of Interest**
The authors declare that they have no conflicts of interest.

**Ethics Approval and/or Participant Consent**
No human subjects were experimented in this research.

**Authors' Contributions**
SM: Developed the proof-of-concept game by integrating group resources. Wrote the paper.
JM: Designed rules for placement of assets in the proof-of-concept game.
IF: Analysed the Space Syntax values for the model.
JF: Provided support for the development challenges.
JM: Supervised the development process.
DFR: Supervised the project. Collected day-to-day reports on the project.

**References**
[1] Al_Sayed K, Hillier B, Iida S, Griffiths S, Karimi K, Dalton N et al. Space Syntax Methodology. 4th ed. London: Bartlett School of Architecture, UCL, London; 2014.
[2] Overview Space Syntax_Online Training Platform [Internet]. Otp.spacesyntax.net. 2019 [cited 2 Feb 2019]. Available from: http://otp.spacesyntax.net/overview-2
[3] Software | Space Syntax Network [Internet]. Spacesyntax.net. 2019 [cited 2 Feb 2019]. Available from: http://www.spacesyntax.net/software/
[4] Nagi D. View analysis with Isovist [Internet]. Medium. 2018 [cited 27 Feb 2019]. Available from: https://medium.com/generative-design/view-analysis-with-isovist-587fce149956

**Do you research in earnest? Submit your next undergraduate research article to the URNCST Journal!**
| Open Access | Peer-Reviewed | Rapid Turnaround Time | International |
| Broad and Multidisciplinary | Indexed | Innovative | Social Media Promoted |
Pre-submission inquiries? Send us an email at info@urncst.com | Facebook, Twitter and LinkedIn: @URNCST
**Submit YOUR manuscript today at https://www.urncst.com!**