ABSTRACT
Sound Space, an interactive virtual reality tool, allows architects and designers to simulate and visualize the acoustic implications of their building designs. By providing designers with the ability to pause, rewind and fast forward a sound wave within a virtual built environment, we empower them to let acoustics influence their design decisions. With a focus on simulation accuracy as well as user experience, we let the user interact with, explore, and curate their own experience while gaining an intuitive understanding of the acoustic implications of their design. Sound Space explores the opportunities that a linked BIM connection may bring within game engine based experiences, and looks at some of the tools we used to try to make that connection.

Sound Space focuses on evaluating the acoustic performance of a space in an interactive and visual experience. For buildings such as symphony halls or theaters, acoustic engineers are a part of the design process from the beginning, but the majority of projects such as schools, hospitals, or museums might employ acoustic specialists only near the end, if at all. At this point it is often too late to make meaningful changes to account for the important acoustic characteristics that can make such spaces work better for students, patients, and visitors.

Our goal was to create an environment that was visually interesting enough to immerse and retain users in the experience, and accurate enough to give useful results to the users for them to make informed choices about their design decisions.
INTRODUCTION
The emergence of consumer VR and AR presents an opportunity for new modes of data visualization and interaction. Architecture is especially relevant to these new technologies as the spatial nature of 3D immersive rendering aligns directly with architecture's focus on creating physical space. Beyond simply creating a digital representation of a building design, VR offers opportunities early in the process for designers to analyze their decisions through meaningful visualizations and experiences (Krietemeyer 2017). VR and AR can overlay data and information that is inherently non-visual. By leveraging the possibilities of showing the unseen, Sound Space attempts to lift acoustic design to the same level of importance as light and material design.

The acoustic data presented must be accurate enough to inform decisions in a meaningful way. Black and Forwood discuss the opportunities for game engines such as Unity 3D to analyze and communicate complex building data (Black and Forwood 2017). The experience should be compelling, enjoyable, and allow for users to interact and explore in order to better understand the results of design decisions (Arnowitz 2017). Finally, it must allow for iteration and enable the designer to make changes, see the consequences of those new changes, and repeat the cycle seamlessly.

The importance of acoustic performance combined with the difficulty of presenting acoustic information in an engaging way in 2D drawings make it an ideal candidate for immersive 3D visualization. In this paper we focus on the technical details of the acoustic simulation and the proposed system for transferring BIM data from design software to the VR visualization. The algorithm for calculating sound propagation is simplified in order to enable performance at run-time, but still provides meaningful data to the user. We look at implementing a system to provide updates to the building design and acoustic data at runtime from Autodesk Revit to allow for the iterative process to work.

METHODS
The acoustic visualization system implemented in Sound Space is a simplified version of previous work—developed for Rhinoceros 3D software using the visual scripting Grasshopper plugin—that had the goal of making sound design more accessible to architectural designers. The Sound Space tool was modified and rebuilt in the C# scripting language in Unity 3D to run the acoustic simulation in real time and provides an iterative workflow for linking Revit models to the Unity game engine.

A single point source sound wave simulation
Understanding Acoustic Simulations
There exists currently a variety of software to analyze and visualize sound in architectural spaces including Odeon, CATT-Acoustic, EASE, and Pachyderm. These simulate sound at varying degrees of accuracy as either time domain waves or as geometric ray-traced particles (Savioja 2015). While these are industry standard tools, they typically live outside traditional architectural workflows and require preexisting knowledge of sound simulations. This forces designers to learn a new software and export their digital building models into the new system if they want to engage with an acoustic simulation on their own.

A room-scale digital acoustic simulation typically requires inputs which include sound frequencies, material absorption coefficients, scattering and diffusion ratios, as well as an understanding of room geometries at multiple scales and how this affects sound at different frequencies (Rindel 1997). While all of these variables should be considered for a highly accurate model, not all of them have equal impact on sound intensity based on the scale of the space, number of sound sources, and materials used. In a room with smooth rigid materials and a limited number of sound sources, the most relevant parameters can be reduced to reflection and absorption to provide and create an informed sound profile (Odeon 2018).

Implementation in Rhino/Grasshopper
The driving concept of the acoustic simulation script in the visual scripting tool Grasshopper was to reproduce sound waves as a triangulated mesh sphere with vertices moving incrementally along a ray-traced vector system. To increase the computation speed and require fewer inputs from designers, the system only requires the room geometry, the material NRC’s of the room surfaces, and a sound source. Another simplification is that it assumes all collisions with room geometry regardless of its material properties, resulting in specular reflections rather than diffuse reflections.

To calculate the resulting decibel level in the room, an impulse response is generated by storing the sound intensity value of each vertex as it moves through the space with a decibel decay of -6 dB per distance traveled doubled. The sound intensity is also impacted by the geometry’s Noise Reduction Coefficient (NRC) which is the resulting sound decay caused once one of the vertices collides with a room surface geometry (Everest 2015). This simplified equation has been compared against real world results, showing that it can approximate sound for architectural purposes, while still existing in standard architectural 3D modeling software (Figures 3 and 4). This process can be highly computationally intensive when calculating decibel values, and can take over ten hours per simulation. Additionally, as Rhino is not a BIM software, each material NRC is entered manually, increasing the risk of an error in the system.

Implementation in Unity 3D
Using the raycasting function that is built into Unity, the simulation process is able to run significantly faster in far fewer steps than the previous implementation in Grasshopper. Building on the same process of creating a mesh from a raycast simulation, sound was again represented as an expanding sphere reflecting off of the geometric surfaces in the room. To further increase the speed of the simulation, decibel levels were not calculated and the sound visualization was designed to decay more quickly. With this method, the simulation could be generated instantaneously, creating an interactive visualization where a user could pause, play, or rewind the results in real time.
The initial Grasshopper script simulated and calculated a decibel sound profile based on early and late reflections, while the Unity implementation only simulates the early reflections and dissipates the sound before the computationally intensive late reflections are displayed. This is one of the main factors that allowed the game engine to give a real time representation of the sound, but kept it from being able to calculate a decibel level of the sound in the space.

Data Interoperability
A common Building Information Modeling (BIM) software, Revit, is an industry standard for modeling and documenting a building through the design process. “BIM models are comprised of both 3D geometry and connected meta-data, both of which can be leveraged inside an immerse experience to present more robust incites to the user” (Nandavar 2018). Having the ability to move this data-rich geometry into other external tools quickly and efficiently, while keeping the embedded BIM data, is a valuable and non-trivial task. Systems that facilitate live interoperability into other software, in this case Unity, empower future development and open up the possibility of new industrial use cases (Boeykens 2011).

Sound Space is comprised of two applications, a desktop based VR application (Figure 2), and a NodeJS web application (Figure 6). The web application allows access to and processes the BIM data, using the Autodesk Forge APIs. We use Autodesk’s user authentication to connect to a specific user’s BIM360 projects. The user verifies and manages the model in the web viewer, which then processes/converts the model. This conversion, using Forge’s Model Derivative API, allows the model and data to be more consumable within game engines. Once processed, Forge outputs a Unique Registration Number (URN) and authentication token that can be used within the VR application to bring in the processed model and data.

From within Unity we attempted to leverage the beta version of Forge’s AR/VR Toolkit SDK to download and instantiate the most current BIM360 hosted model with its relevant data at runtime. Due to time constraints and the experimental nature of the toolkit, we were not able to successfully complete this live data connection with the beta AR/VR Toolkit SDK, but as new ways of connecting BIM data and game engines emerge, there will inevitably be the same implications.

Once the model and data are in Unity, the data pipeline is complete. The developer now has the ability to use this data in creative ways within the game engine and deploy to a broader array of both users and devices (Figure 5). Taking this pipeline further we can create custom parameters within the authoring software in order to use it on the other end for preconceived use-cases.

To create this connection within the Sound Space VR experience, we added a custom “Noise Reduction Coefficient” parameter to specific elements within Revit. Once brought into Unity, our interaction and experience differs depending on the values of these parameters. With this live connection, a designer would be able to change this parameter value in Revit and Sync back to BIM360 in order to explore a different option. The user in VR will then see the difference in the affected sound wave.

RESULTS
The result of this project is a proof-of-concept experience that allows users to explore the behavior of sound reflections in an architectural space. One of the main challenges of this project was creating a meaningful visualization of sound, something that is not inherently visual. In user testing, we found that many users were expecting to be able to hear the sound when the sound wave passed them. Such a feature, however, would not make sense in the context that we are presenting. As sound travels very quickly through the air, it would not be possible to simulate it both visually and aurally at the same time. We focused on interactivity as an important feature of the experience to engage designers. By allowing users
to place point sources at runtime, we had to simplify the calculations while producing results that we feel remain meaningful, despite the fact that they are inherently not as accurate as running the full calculations.

Another challenge was creating the live connection to the BIM model. Removing the manual import process and allowing design teams to quickly iterate through simulations of design options will inevitably make this tool more useful. Our exploration of Forge's APIs proved valuable in building a NodeJS web application, regardless of our failure with Forge's AR/VR Toolkit.

CONCLUSION
This implementation was created at a hackathon, and is a conceptual demonstration of the potential for VR to aid the design process in architecture. There are a number of ways in which the experience can be enhanced to be more effective.

Future versions would allow designers to place point sources for robust calculations. Such calculations would need several hours to complete, but once done, would allow users to engage with the sound propagation in a similar manner, with the exception of moving the point source.

Another future step would be to attempt to add auralization, the ability to listen to the acoustics of the simulations or measurements. This task requires an anechoic recording and pairs it with the calculated acoustic profile of a room, creating a live sound which would let a user hear the intensity and clarity of that sound as if it had been generated in that room (Vorländer 2015). We also wanted to present the information in a spatial and engaging way to designers who are used to working visually. Developing an accurate aural simulation of the space could be an effective complement to run in addition to (not at the same time as) the visual simulation.

Finally, this project was a demonstration of using building information direct from modeling software. We chose to focus on acoustic information, but this same process could be used for other types of building data as well. This project is in use by a group of undergraduate architecture students to understand how different room geometry can impact sound. They are using Sound Space in conjunction with the companion Grasshopper script to experience and then calculate the resultant decibel level of their designs.

REFERENCES


Luke Gehron is a 3D Visualization Designer at Payette Architects in Boston. He previously worked at DFA and KushnerSTUDIOS, and received his degree in architecture from Auburn University.

Adam Chernick is Associate of Interactive Visualization at SHoP Architects. He focuses on research and development for Augmented Reality and adjacent emerging technologies. Adam’s work encompasses a range of interactive applications including web, mobile, and headset-based experiences for large projects around the world. Adam earned his Masters of Architecture from Pratt Institute and a full-stack web development certificate from Columbia University.

Christopher Morse is Associate of Interactive Visualization at SHoP Architects. His research focuses on the integration of interactive visualization technologies such as VR and AR into all phases of the design process. He is collaborating with the Department of Architecture at Cornell University to use immersive VR tools as an integral part of an urban design studio. He received a master's degree in architecture from Cornell University and holds additional degrees in mathematics education and physics.

Sabrina Naumovski is a Design Researcher at KieranTimberlake. She received her BArch from New Jersey Institute of Technology in 2015 and her Masters in Advanced Architecture from the Institute for Advanced Architecture of Catalonia in 2018.

Zeyu Ren is a designer and animator who uses emerging technologies to tell stories. He earned his BFA from UCLA and MFA from RISD. Zeyu uses Virtual Reality technology to create animation. His short film "Wired" is the first multi-scene narrative animation made in VR. He is also a published author of "Planet Diary," which tells a story of a unique world every day.