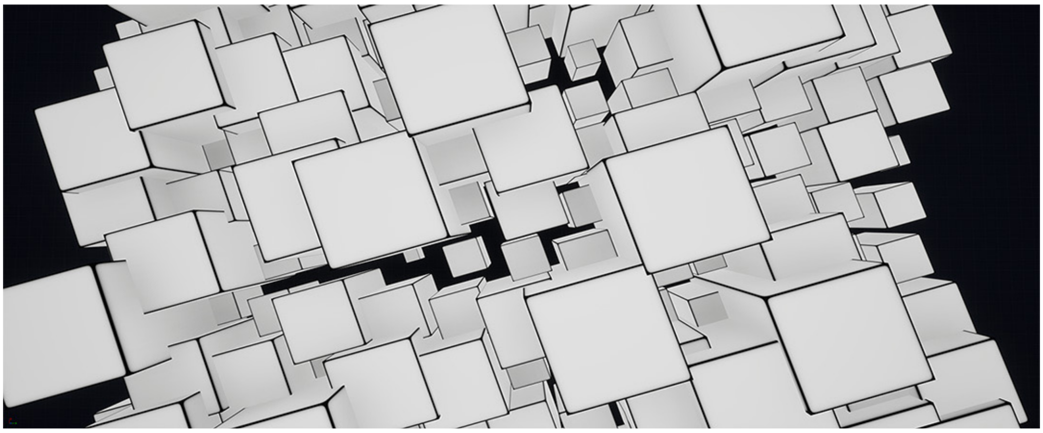
Music as an Inspiration Source for Architectural Forms through Unreal Engine

Simone Porro

Abstract

Game Engines are evolving by providing tools that allow real-time simulations which can be applied outside of their original context. Through Unreal Engine [UE4] and an integrated plugin called Sound Visualization, a system has been programmed capable of visually representing the music starting from a bidimensional matrix pattern composed of cubes that are geometrically controlled and transformed in real-time exploiting the frequency spectrum analysis of an audio source. Music has been used as an element from which dynamic forms of architectural interest are created, transforming a form of sound information into a form of visual representation that develops itself in real time as the music flows.

Keywords
architecture, music, unreal engine, game-engine, simulation.



doi.org/10.3280/oa-548.75

Introduction

One of the aspects that distinguishes the digital revolution lies in the possibilities that this context provides. Technologically there is an increase in device performance and in parallel the development and applications of software that can exploit this power.
Based on this context, the development of game engines has undergone considerable visible developments both in terms of the remarkable graphics qualities achieved by videogame products (rendering), and the functional potentialities of the same developed with increasingly advanced programming tools (scripting). The union of these two abilities is the basis of the more intimate nature of the Game Engine which is being able to simulate. In other words to visually represent, for a certain duration, calculations and processes that occur in the background based on what is coded. The use of this type of software in the architectural field at the moment has not yet been fully investigated, especially concerning new developments, therefore it can lead to results not previously considered that need to be studied in depth, as well as to investigate new forms of experimentation and interaction between space and time.

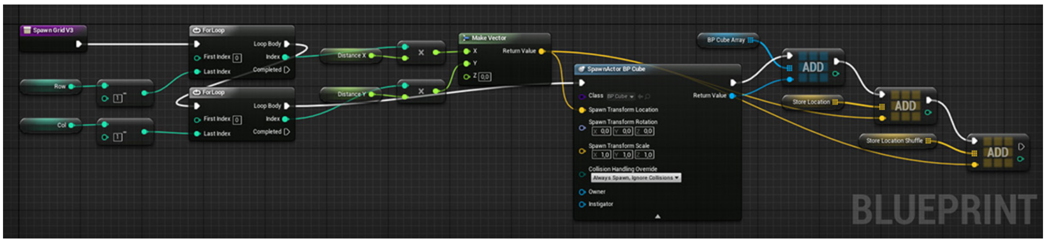
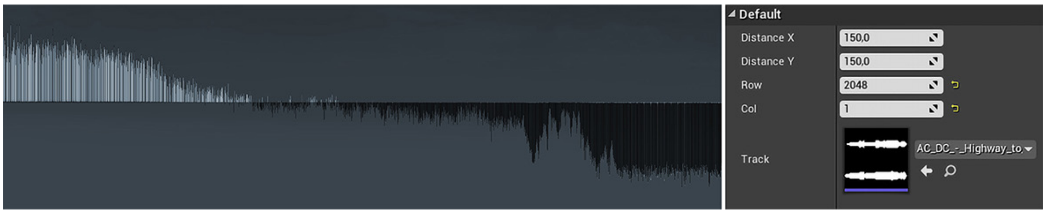


Fig. 1. Blueprint scheme for the generation of two-dimensional matrices.

Music and architecture are known to be linked together by structures, rules and proportions that are shown through the components of space and time. In this context, music and its temporal component have been used as a tool to generate dynamic spatial forms of architectural interest by exploiting the structures already written within a musical composition.



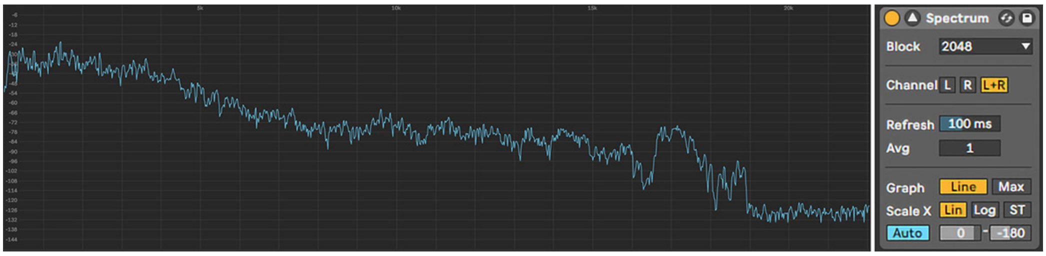
Default
Distance X 150,0
Distance Y 150,0
Row 2048
Col 1
Track AC_DC_-_Highway_to

Fig. 2. Visualization of the frequencies on the Z-axis in Unreal Engine (left), parametrized settings (right).

‘Liquid Architecture’ and ‘Frozen Music’

I find it interesting to introduce the question by quoting two characters from two different eras and contexts who shared a common vision:
"…cyberspace encodes architectural knowledge in a way that indicates that our conception of architecture is becoming increasingly musical, that architecture is spatialized music. […] In principle and with the proper architectural knowledge, any pattern can be made into a work of architecture, just as any pattern can be made into music. […] Cyberspace is liquid. Liquid cyberspace, liquid architecture, liquid cities. Liquid architecture is more than kinetic architecture, robotic architecture, an architecture of fixed parts and variable links. Liquid

architecture is an architecture that breathes, pulses, leaps as one form and lands as another. Liquid architecture is an architecture whose form is contingent on the interests of the beholder; it is an architecture that opens to welcome me and closes to defend me; it is an architecture without doors and hallways, where the next room is always where I need it to be and what I need it to be. Liquid architecture makes liquid cities, cities that change at the shift of a value, where visitors with different backgrounds see different landmarks, where neighborhoods vary with ideas held in common, and evolve as the ideas mature or dissolve" [Novak 1991, pp. 282-284].



Spectrum
Block 2048
Channel L R L+R
Refresh 100 ms
Avg 1
Graph Line Max
Scale X Lin Log ST
Auto 0 -180

Fig. 3. Visualization of the frequencies on Ableton Live (left), frequency visualization parameters used (right).

This premise can be compressed in the Goethe quote that follows: "Music is liquid architecture; Architecture is frozen music" [1].
Taking up what was written by Novak he describes a scenario in which architecture becomes dynamic and the fourth dimension, the temporal dimension, becomes its expression. A dimension that has the same properties as the music that he defines as 'navigable music'. Music and architecture are so intimately linked by a connection that is not directly visible. Both are generated by certain rules, orders and codes. Both share similar but hidden mathematical and geometric structures at a more primordial level, and both can be described and represented into digital worlds. Just as architecture is composed of sequences and relationships of full and empty spaces, and without one is not possible to perceive the other, so

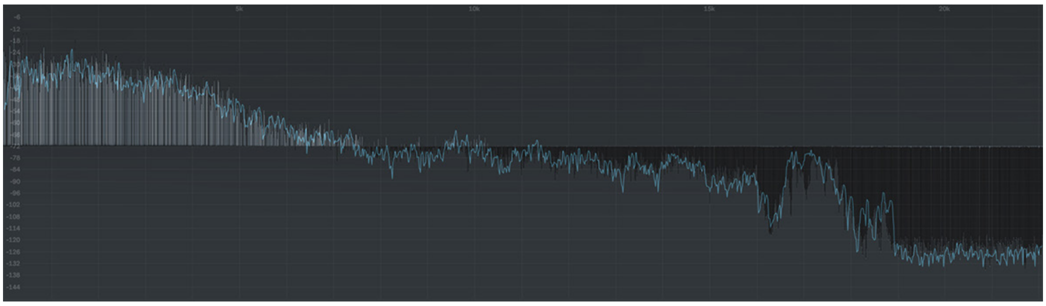


Fig. 4. Overlapping of the frequencies obtained on Unreal Engine with those obtained on Ableton Live.

music is a succession of sounds and silences, and both can be translated into the cyberspace by series of zeros and ones. Ordered information. Information tied to itself both in the single instant, 'frozen music', and considering subsequent frames in time, 'liquid architecture'. Quoting Novak: "The architect is called upon to design not the object but the principles by which the object is generated and varied in time [...] A work of liquid architecture is no longer a single edifice, but a continuum of edifices, smoothly or rhythmically evolving in both space and time" [Novak, 1991, 225].

Testing Unreal Engine

Based on these theoretical premises a system was developed for which music can be linked to architecture by creating a dynamic simulation that allows to control and manage solids in space in relation to the flow of music and therefore to time. Through Unreal Engine, a system has been programmed in Blueprint (Visual scripting editor of UE4) which allows to generate two-dimensional matrices by determining the number of elements for each axis as well as a distance between them (fig. 1).

These matrices represent the 'pattern' mentioned by Novak, a grid of spatial points where the solids on which the audio frequency is imposed will be generated. The Sound Visualization plugin provides a function that is able to calculate the frequency spectrum of any audio source and to be able to use these data in real-time during the audio flow. Then the sound is divided by the number of the pattern elements. The information is thus 'stored' in each element and then subsequently can be represented visually. A first test was carried out on a two-dimensional array by setting an array of 2048 elements, dividing the frequency by the number of these and scaling on the Z-axis each element for the calculated frequency, generating the result visible in fig. 2.

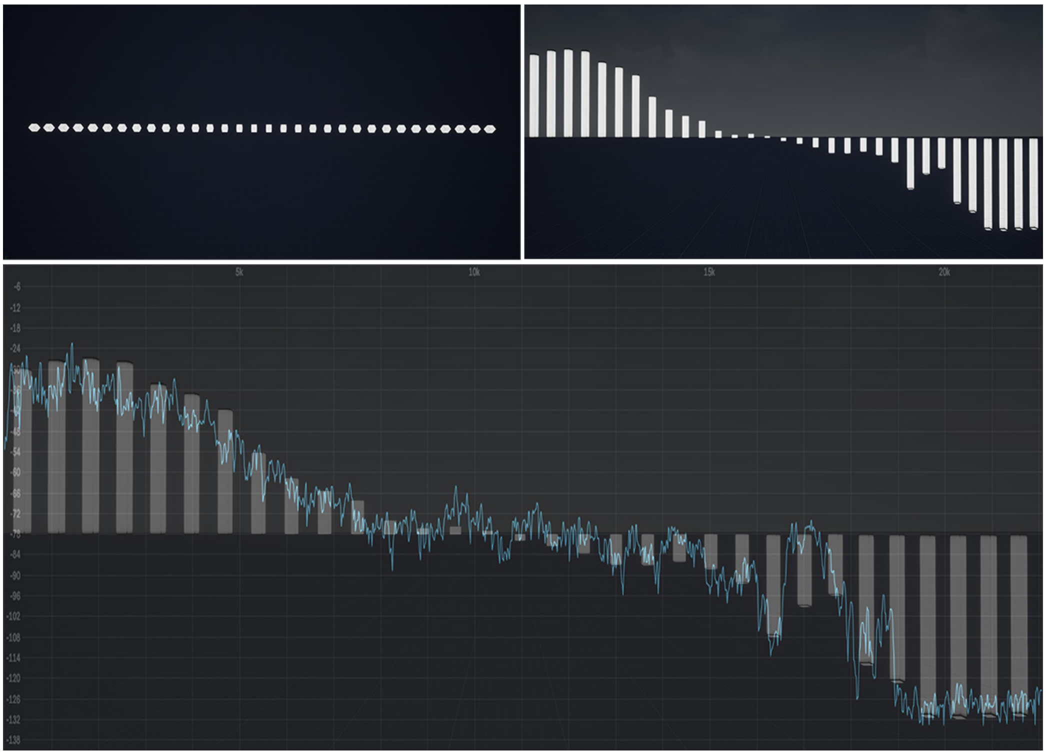


Fig. 5. Array of 32 elements (above left), array of 32 elements with frequency on the Z-axis (above right), frequency overlap obtained in Ableton Live with the oversimplified one with 32 elements (below).

This process leads to graphic results comparable to those obtained by audio software that allows you to visually represent the frequency of any audio source. The frequency spectrum of the same song was analyzed at the same time through the use of professional audio software (Ableton Live 10 Suite). Among the parameters used (fig. 3 right), we can see a sampling of the frequency of 2048 blocks and a visual representation of the linear result on the vertical scale (fig. 3 left).

The two results were subsequently superimposed to show how the charts are comparable between each other (fig. 4).

What has been shown so far, on one hand, demonstrates that the result of the programming corresponds to the real frequency of the audio taken as a reference from the other opens the doors to being able to manage the information according to other spatial criteria.

A further test has been done oversimplifying the array with 32 divisions (fig. 5 above left) representing the frequency in the same way we obtain the result visible in fig. 5 (above right). Moreover, as can be seen, the final result is in any case comparable to the graphic representation of frequency analysis obtained through professional software (Ableton Live), although there is a clear loss of information due to the reduced number of subdivisions with which the audio (fig. 5 below).

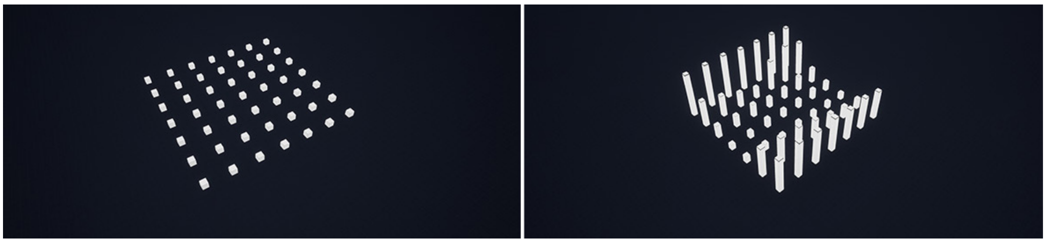


Fig. 6. Square grid (left), square grid with the application of the frequency scaling the Z-axis of the cubes (right).

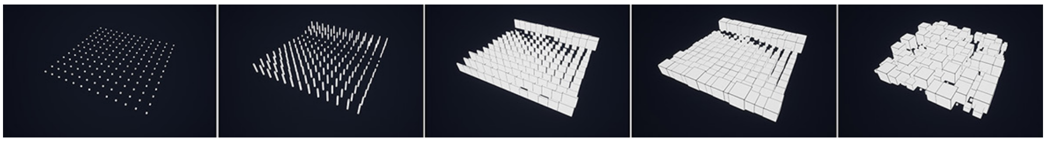


Fig. 7. Sequence generation applying the frequency on the scale parameter and then randomizing it among the elements in the last frame.

Once the correct representation of the simulation has been determined, the subsequent tests were performed by arranging the frequency on two-dimensional matrices. Fig. 6 (left) shows a two-dimensional matrix generated using the same script as in fig. 1. Since as clearly shown in fig. 5 (below) there are positive frequency values and negative values, negative values lead to having extrusions that are represented on the negative Z-axis. To avoid this type of situation it was decided to bring all negative frequency values to absolute values. Fig. 6 (right) shows the extruded frequency on the Z-axis considering all the values as absolute. Note that this choice was completely arbitrary and that as an alternative it would have been possible, for example, normalize the information on maximum and minimum positive values. In this view, an alternative to controlling the frequencies would be to control the curve through an equalization system to be able to separately manage high, medium or low frequencies.
Subsequently, several simulations were carried out exploiting the main basic geometrical parameters: translation, rotation and scale. Moreover, each of these parameters has been subdivided into the three spatial components: x, y, z. The 'archived' information can thus be expressed in one or more components simultaneously generating different possible compositions. The representation is controlled in real-time as the audio or music flows.

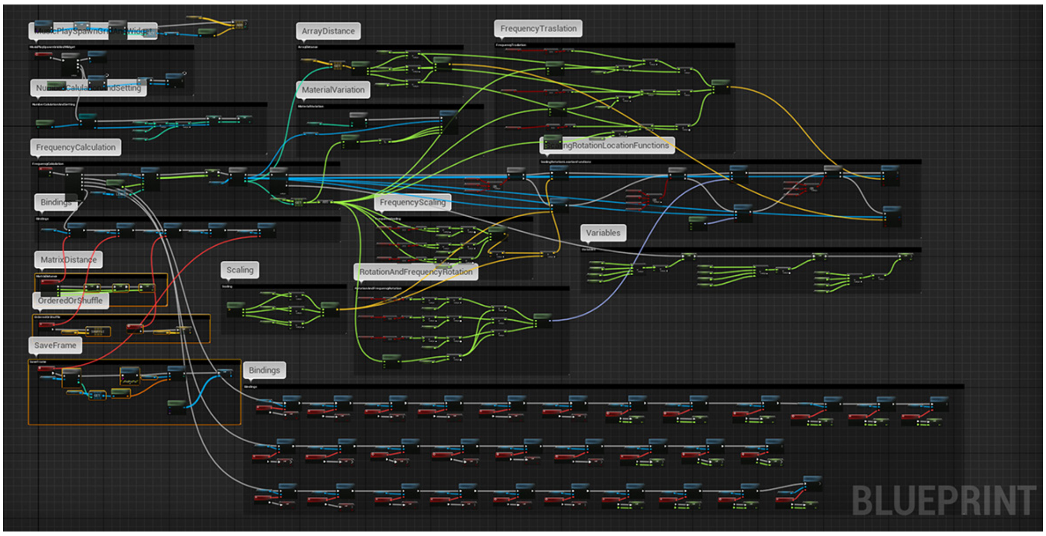


Fig. 8. Blueprint scheme of the real-time simulation.

Fig 7 shows the display of the frequency which can also be randomized among the elements of the pattern (see last frame fig. 7). Fig. 8 shows the script behind the simulation that manages the system developed.

Some more elaborate results are visible in fig. 9 and by applying the frequency, appropriately architectural forms of possible interest can be obtained. The simulations have been made considering square matrices. It should be noted that it is possible to start from different starting conditions by varying not only the structural 'pattern' but also the type of element treated with spheres, cones, cylindrical elements and so on. It can be even foreseen situations with mixed or non-basic geometrical elements. The results highlight how different pieces but with the same application of the frequencies lead to similar results visually but never the same, especially in time sequences. It is also planned to investigate the possibility of using more than one pattern at the same time, thus creating more complex articulations. Or to generate more geometric shapes from the same pattern but managing each series independently from the other. Other compositions could be generated using more specific architectural elements such as planes, columns, arcs and so on. Other possible fields of

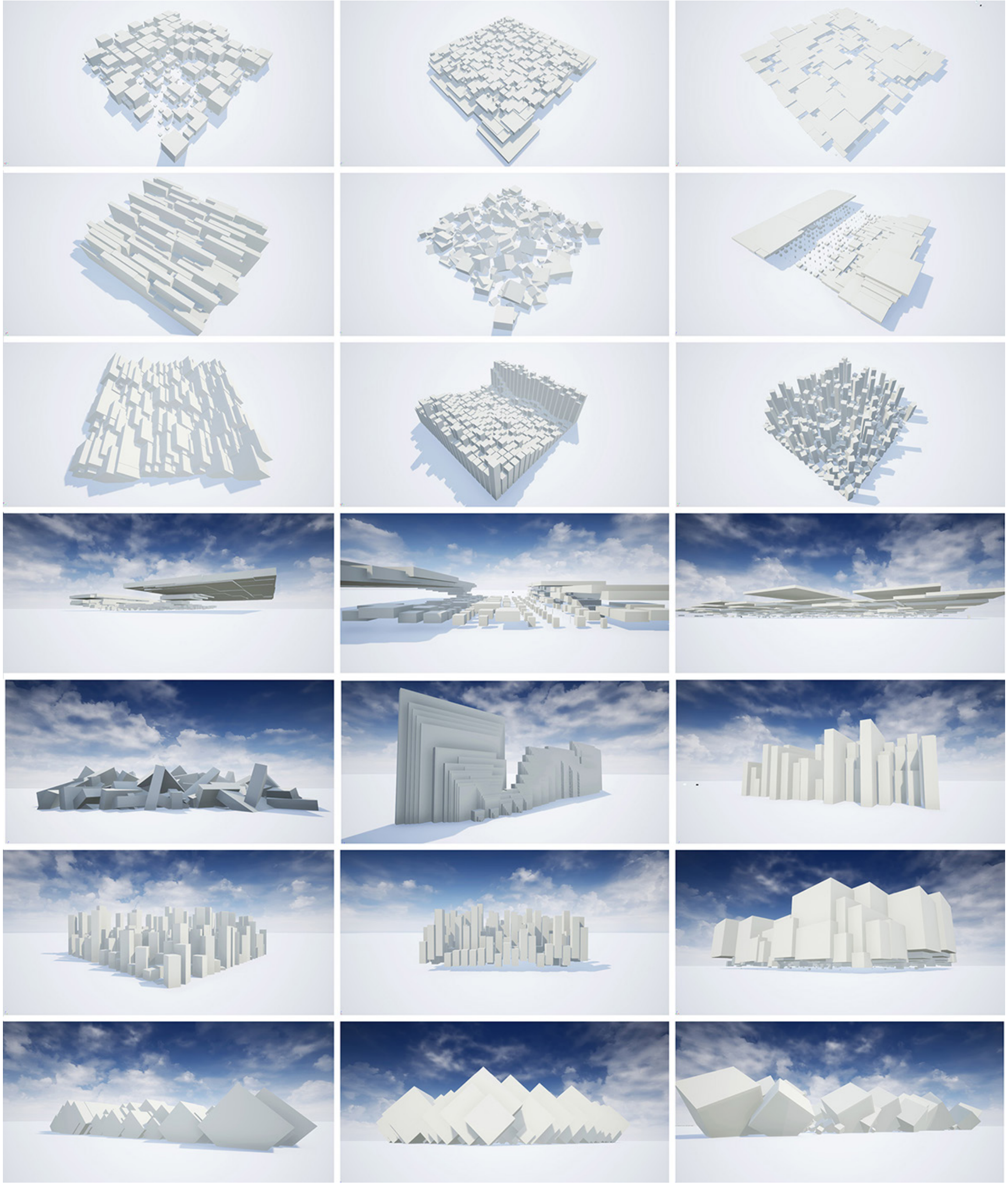


Fig. 9. Results based on a square matrix applying the frequency through geometrical parameters of rotation, scale and translation considering their components x, y, z.

experimentation concern the use of other patterns different from matrices. It is noted, as it has already been understood that the potential development of this modality of musical, architectural or design key representation extends to a large number of possible experiments.

Conclusions

This study has different ways to see this research. On the one hand, it shows that the use of the scripting component of video game software can be used for other non-recreational purposes (serious game). On the other hand, to show the potential that this software can bring to and which at present the current architectural tools are not able to do, as they exploit the temporal aspect, the simulative one.
Finally, since the technological evolution moves more and more towards the cancellation of the calculation and updating times of information in favor of real-time, in an innovative vision of the future it is necessary to integrate this type of instruments by the architect especially concerning the simulation aspect. The use of this type of software in the architectural field at the moment has not yet been fully investigated especially in relation to new software developments, therefore it can open the way to further experimentations.

Notes

[1] The quote is often attributed to Johann Wolfgang Goethe.

Acknowledgment

This paper is based on a Master thesis discussed at the Politecnico di Milano, together with Prof. Luigi Cocchiarella, we examined the role that Game Engines can play in the graphic representation and design processes.

References

Porro Simone (2019). *Playing Architecture. Using Real Time Engines as Operational Tools for Architectural Design*. Master dissertation, April 2019, tutor prof. Luigi Cocchiarella. Politecnico di Milano.

Hemmerling Marco, Cocchiarella Luigi (2018). I*nformed Architecture. Computational Strategies in Architectural Design*. Cham: Springer.

MVRDV/DSD (2007). *Spacefighter. The evolutionary city (game)*. Barcelona: Actar-D.

Novak Marcos (1991). Liquid Architectures in Cyberspace. In Benedikt Michael (ed.). *Cyberspace: First Steps.* Cambridge: MIT Press, pp. 225-254.

Abt Clark C. (1987). *Serious games*. Lanham: University Press of America.

Von Borries, Friedrich Walz P. Steffen, Böttger Matthias (2007). *Space time play, computer games, architecture and urbanism: The next level.* Basel: Birkhauser.

Baricco Alessandro (2007). *The Game*. Milano: Giulio Einaudi Editore.

Frasca Gonzalo (2007). *Play the message: Play, game and videogame rhetoric*. Copenhagen: IT University of Copenhagen, Ph.D. Dissertation.

Huizinga Johan (1949). *Homo Ludens. A study of the play element in culture*. London: Routledge & Kegan Paul.

Author
Simone Porro, Politecnico di Milano, porrosimone@gmail.com

To cite this chapter: Porro Simone (2020). Music as an inspiration source for architectural forms through unreal engine. In Arena A., Arena M., Brandolino R.G., Colistra D., Ginex G., Mediati D., Nucifora S., Raffa P. (a cura di). *Connettere. Un disegno per annodare e tessere. Atti del 42° Convegno Internazionale dei Docenti delle Discipline della Rappresentazione/Connecting. Drawing for weaving relationships. Proceedings of the 42th International Conference of Representation Disciplines Teachers*. Milano: FrancoAngeli, pp. 1356-1362.

Isbn 9788835104490