

<https://doi.org/10.31896/k.23.5>

Preliminary communication

Accepted 19. 11 2019.

SIMONE PORRO
LUIGI COCCHIARELLA

Use of a Game Engine Artificial Intelligence to Represent People Flows in Architectural Spaces via Geometry and Graphics

Use of a Game Engine Artificial Intelligence to Represent People Flows in Architectural Spaces via Geometry and Graphics

ABSTRACT

The digital and the related technological evolution in recent years have shifted to words such as *Virtual Reality and Artificial Intelligence*. The wider use of these and other technologies in architecture has been far limited by a lack of IT tools with which architects could interface since they have been made available in the last few years only. The evolution of the tools used by the architect can be condensed and simplified into a sequence of three stages: *Drawing Board, CAD systems, Game Engines*. The frames of this sequence, in addition to indicating instruments, are representative of the historical context in which they have been or are still being used. This study, based on a Master thesis recently discussed at the Politecnico di Milano [14], examines the role that *Game Engines* can play in the graphic representation and design processes. More specifically, it takes a closer look at the Unreal Engine as a tool for creating a real-time design environment and using Artificial Intelligence (AI) technologies to represent user flows in the space as valuable support and a relevant part of the design strategies aiming at implementing and evaluating design options. For this purpose, various simulations have been carried out both considering users' flows based on assigned spaces, and generating spaces based on the users' flows.

Key words: artificial intelligence, flows, parametric modeling, dynamic environment, game engines, simulations, unreal engine

MSC2010: 00A66, 51N05, 01A05, 97U99

Upotreba game enginea umjetne inteligencije u svrhu predstavljanja protoka ljudi u arhitektonskim prostorima pomoću geometrije i grafike

SAŽETAK

Digitalna i njoj pripadajuća tehnološka evolucija posljednjih godina pretvorila se u riječi kao *Virtualna stvarnost i umjetna inteligencija*. Šira upotreba ovih, ali i drugih tehnologija u arhitekturi dosta je ograničena zbog nedostatka IT alata s kojima su se arhitekti susretali s obzirom na to da su bili dostupni jedino u posljednjih nekoliko godina. Evolucija alata koju koriste arhitekti može biti komprimirana i pojednostavljena promatrajući je u tri faze: *crtaća ploča, CAD sustavi, game enginei*. Okviri ovih faza predstavnici su povijesnog konteksta u kojem su se koristili ili se još uvijek koriste. Ovo proučavanje koje se temelji na diplomskom radu nedavno obrađenom na Politecnico di Milano [14], ispituje ulogu koju *game enginei* mogu igrati u grafičkoj prezentaciji i procesu projektiranja. Konkretnije, bliže sagledava Unreal Engine kao alat za stvaranje okoline projektiranja u stvarnom vremenu i koristeći tehnologije Umjetne inteligencije predstavlja korisničko razmišljanje u prostoru kao korisnu podršku i važan dio strategija projektiranja s ciljem implementacije i evaluacije projektantskih opcija. Za ovu svrhu, brojne simulacije su izvedene uzimajući u obzir i protoke korisnika temeljene na konkretnom prostoru, ali i generiranje prostora temeljeno na protocima korisnika.

Ključne riječi: umjetna inteligencija, protoci, parametarsko modeliranje, dinamično okruženje, game enginei, simulacije, unreal engine

1 Play and serious game definition

To understand how real-time engines integrate in the Geometrical and Architectural context, we need to start from upstream, as they fit into the digitization process, looking

at similarities and contact points with Geometry and Architecture, as well as at what architecture can get from these digital engines. To this end it is necessary to remind meaning and definition of Game and Play in relation to the topic. The discipline behind this world is called *Game Studies* or

Ludology. This sector studies games in general, the act of playing, the player and the culture behind her/him, contextualizing the historical playground. As we could expect, the research fields falling within this sector are various and vast, involving anthropology and sociology, as well as psychology, over and above scientific and technical areas. All these aspects contribute to define the design of a game concerning both the player and the game itself and of course in relation to the user. Not to be confused *Game Studies* or *Ludology*, is about the study of video games, which is simply a branch of a much broader context dealing with *Digital Games*. In the playful context before the introduction of the digital, a publication having a discreet influence and still taken as a reference nowadays was *Homo Ludens* by John Huizinga in 1944, as well as the subsequent discussions, in which he highlighted the importance of the “Play” element inside of a culture. Huizinga argued that play is a factor of primary and fundamental importance for a society. According to this point of view, he brings the highest human achievements back as a product of the creativity of the game.

The common perspective that play is only one form of culture among many is a modern deformation, and a concept wrong in itself. On the point Huizinga expresses a simple contradiction: if the game were the product of human culture there would be no play outside of it, but this is not so since the game is much older and is already shown in nature, as shown for example in the animals plays. Therefore, since the Game exists before culture and before mankind himself, one can think, opposite, that culture is born through the Game. In the biological field, indeed, an attempt has been made to give explanations and formulate theories that try to define the game as a function of life, explaining the game as something that serves biological purposes. Even Huizinga asks: why do we play? Whatever the answer, be it biological or not, in the end there is always a relation with satisfaction, or with the pleasure of playing that escapes any logical analysis, that is, precisely, what Huizinga says is the essence of the game. In the Game there is an immaterial, conscious element of the individual, which manifests itself in the physical existence through the game itself. The Game is the door that connects material and immaterial, real and virtual. From these premises it is clear that the Game has a much higher value than one would normally attribute to it, which transcends both the biological and cultural activity of the individual who performs it. It is an act having within itself an immaterial, archetypal and creative sense.

Moreover, at the time the author had already tried to free the issue from another common but false opinion, which saw the concept of Game as the opposite of seriousness. Huizinga pointed out that this opposition of facts does not hold up since the Game can be both serious and not serious, and although there may be an oscillation between

these two opposites the Game still remains Game, showing its own identity. On this subject in 1970 Clark C. Abt introduced the phrase “serious games”, which he defined as follows: “... these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement” [2, p. 9]. Today the same idea is generally expressed as “games that aim at training, educating, persuading or communicating values and ideas” [5, p. 26]. In this context, according to the definition above, “serious games” are no longer Game, or at least they are not in their purest form, as they are exploited for a certain purpose. In a Broader sense, quoting Juul, “a rule-based formal system with a variable and quantifiable outcome” [9, p. 35]. Concerning the seriousness of a Game, in support of the thesis that the words *serious* and *game* are not necessarily two opposites of the same aspect, a historical example can be taken as a reference from Von Neumann and the Game Theory. Beyond the name, which can make you mislead, contrary to what you might think, he does not talk about Game in the common sense of the term, although it is precisely from the observation of that common context that Game is born. With the title: *Theory of Games and Economic Behavior*, he essentially put mathematical aspects at the base of studying and analysing the decisions of a subject in certain situations of conflict with other rivals. In other words, he intended to predict the behavior of individuals in situations that can lead to the division or the winning of something in monetary terms or more generally in utilitarian terms. The theory therefore applies to an infinite number of scenarios, of various nature and complexity, which can range from chess games to the financial or economic market contexts. The most interesting aspect of this story is that it shows a different definition of Game. This point of view began to be used to evaluate behaviors and choices of the “players”, no longer understood as acts aimed at a mere fun or without a completely defined purpose, but on the contrary, as actions having a plan or a strategy behind, and aiming at specific purposes.

More recently Alessandro Baricco, in the book titled *The Game* [3] describing the world in which we live in and the digital revolution that underlies it, shows how this process of *gamification* is definitely connected to our society, more profoundly than we think. Baricco starts from a simple similarity: *Table football*, *Pinball*, *Video game*. A similarity that traces some historical stages of the technological evolution of the digital. Well, starting from this path he realizes that it reveals a substantial mutation behind the use of more and more technologically evolved Games. In a *Table football* the feeling is natural, the noises are real, you have to physically do a certain movement, a certain effort and even the ball is real. It is no more than a physical game. If you switch to the *Pinball* (flipper) the situation begins to change. Many of the noises become electrical, and a screen appears that begins to take some importance.

The ball is closed under a glass and the physical sensation is reduced. There are only two keys, whose consistency becomes much softer and different from the resistance of table football. This passage is a sort of preparation for the last passage, says Baricco, a kind of “limbo” to the final step, which is the video game, of which *Space Invaders* is the key reference and prototype. Here everything has been transferred to the screen, physical remains are only the keys that act as a link between the digital world and the real world. The sounds have become completely artificial. The rhythms change, much faster, with a different concept of time, in a way more liquid. Everything has turned into sequences of bits, numbers that are translated into images, sounds, actions, etc. These games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. It slips into a dematerialized world, with a different density, in other words, as he says, in “the pure essence of the game”. In this similarity the passage is not only in terms of digitization. If on the one hand you have loss in movement or reality, on the other hand you have much in exchange. If the table football, that is, just one physical device, offered only one possibility of Game, the digital engine beyond the screen potentially offered an infinite number of simulated “realities”. What Baricco also points out is that the same mentality that led to the evolution of the Game has revolutionized and is still revolutionizing the companies. According to the author this phenomenon exploded with worldwide resonance in 2007 when Steve Jobs presented the *iPhone*. In that context, a further transition to digitization was established. The keyboard characterizing the former smartphones disappeared, and in its place a touch screen with icons to press appeared. Well, in hindsight, the iPhone itself is built like a Game device, and only the purpose for which it is used changes, including games. In addition to make phone calls (the original function still in the name), from that moment on with one single device you could send emails or write texts, take pictures or make movies, and (of course) also play in the classic sense of the term. Then on the one hand we have the Game with the aim of having fun or, as Huizinga would say, without any other necessary purpose if not that of the mere fun (let us say *funny game*). On the other hand we have the Game aimed at carrying out activities related to the everyday life (let us say *serious game*). So far, the definition of *Game* as it is intended in this discussion is not necessarily seen as a playful aspect, but much more as a tool that through the Play, intended as a productive factor, can lead into creative, and generative results, in the context in which it is applied. A metaphor of that can be seen in terms of an engine element that if inserted in a car results in allowing motion, where the Play is movement for transporting purposes, while, if applied to a children’s carousel generates fun, and the Play is for entertainment and pastime.

2 Architecture and geometrical serious games

Once you understand what is meant by Game, it is interesting to note how this could be integrated into the geometrical and architectural context and what are the common points between the two worlds. As you may have already guessed for simplicity, we will no longer use the words “serious game”, but simply: *Game*. Letting people understand that it is neither our intent to trivialize the matter in discussion, nor to diminish other related matters. It is for the pure sake of simplicity. Well, if we analyze the structure of a Game, it is generally formed by a visual, two-dimensional or three-dimensional part, managed by the *Game Designer*, and by a structural, scripting, computer-guided part precisely arranged by the *Programmer*. The same can be said for Architecture that comes from the right dialogue between an aesthetic compositional part, and a structural engineering part. And even Geometry can be seen as the result of a process and its ‘aesthetic’ representation. Therefore, Game, Architecture and Geometry, share the same conformation processes. They, indeed, share similar structural aspects concerning their ‘composition’. The similarity established by Baricco among table football, pinball, and videogame in relation to how the digital has historically changed, shows in time-lapse some key moments of the digital evolution of the Game, which in parallel could correspond to innovations in Architectural design and in Geometry. To summarize, in all these three fields we have gone on to an ever smaller physicality as well as to a translation of graphic operations for visual purposes into the form of codes, numbers, bits. Since even architectural and geometrical are involved in the digital revolution, the same time-lapse logic proposed by Baricco for Games can be applied to these contexts, with special reference to the tools used, where an interesting comparison would emerge. The similarity here proposed in relation to digital graphics for architecture is the following: *Drawing Board, Cad Systems, Game Engine*.

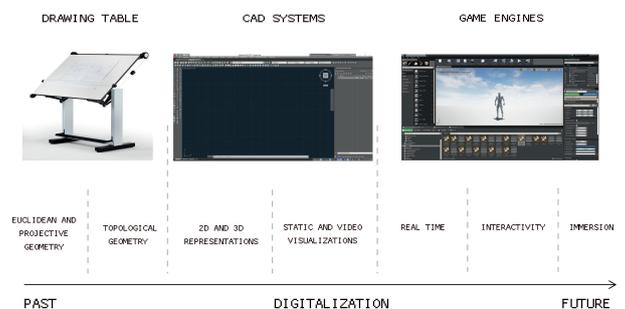


Figure 1: *Timeline of frames and functionalities of tools used by architects*

Each step of the sequence symbolically represents a frame of a wider context, in which there is a change of both the tools or the software and above all the potential that the technological evolution offers. From the analogue of the Drawing Board we have moved on to the CAD Systems in which the space is digitized in its two and three-dimensional forms, up to the last step in which beyond the digital space also the temporal component is simulated. It is interesting to note some fundamental mutations if you switch between the chain frames. When designing with pencils, ink pen, paper and tracing paper on drawing tables, everything was physical. You paid for the mistakes with ink at the time, and all the instruments had a certain consistency, certain smells and noises, and thicknesses. At the time you got your hands dirty and there was a certain slowness in all the processes, from pulling a line of which you made sure its sweetness, to its cancellation. With Cad software the situation has changed: pencils and ink no longer exist, lines are produced by a command as well as their cancellation, the tracing papers are called layers, you don't have to pour or remove them on the drawing, now they are activated and deactivated with a click. The physical drawing board no longer exists, or rather exists the digital counterpart, which has become usually black. Not only the graphic model developed is no longer limited to the two dimensions of the drawing board but it also acquires the third dimension, the extrusion along the Z axis. The only connection left with the drawing board is the paper, still used to print on what is digitally produced, while nowadays also 3D printers started to take off. Much has been lost with Cad Software, but in return it has also achieved quite a bit: production speed, graphic cleaning, management of every aspect of the design, etc. The last step in this chain is about *Game Engines*. Not so much for their current level of use but more for the potential they offer compared to the Cad frame. Game Engines allow to work with an element not well managed by Cad software: *time*. That brings two main important features: the visualization and representation of the scene in real time, with the possibility of integrating the model in Virtual Reality, and the possibility of creating, through computer codes or scripts, possible interactions between the user the element of a scene or the scene itself. In other words, all the information as well as the process that governs them will be condensed into a single three-dimensional model accessible with any screen or with a viewer, and controlled by visual interfaces. Goodbye to paper. Because the information moves towards digitalization a further aspect needs to take into account called *Gameplay*, in other words: the experience that comes from the act of playing. Since there is a more and more explicit dialogue between the spatial and the geometrical components, the rules and processes behind it (Game) and the temporal components (Play) it would be crucial consider

also this interaction between the game's response and the user, which "can generate outcomes that never could have been imagined beforehand". [12]

Game Engine choices and mapping

The choice of the *Game Engine* as a design and representation tool was based on different factors. First of all, together with Unity it is the most used software program among game engines for architectural purposes. A research conducted by CGarchitects shows that it is the most used among Real Time Engines. Responses were gathered between November 30 2017 and February 3 2018 via CGarchitect's community (social media & email newsletter) and through Facebook groups associated with real-time engines and architectural visualization. 997 Responses came from Social media and 1,066 responses came from email newsletter subscribers. Second reason for the choice is due to a series of characteristics that other competitor software programs considered do not offer yet, or they do not offer with the same quality. As you can see in the Fig. 2 [Top Left] shows that Unreal is between the most used in the architectural field as a visualization tool and also the most used for experimental purposes among general renderer software [Top Right]. What's also interesting is the use of Unreal Engine in the real time research field [Down].

Although the programming is not the architect's own discipline, the software comes in handy giving the possibility to program through a *visual scripting editor* called Blueprint. This type of interaction with the software partly recalls Grasshopper, however it should be noted that the two software programs work on different contexts, from which a different type of programming in the two cases follows. In particular, Grasshopper is largely based on geometric properties of which the architect has knowledge and s/he is more familiar with, indeed Grasshopper is a plug-in of a well-known NURBS modeler, Rhinoceros. It is not so for Unreal Engine, a much more eclectic software that must interface with software programs developed in a wide spectrum of disciplinary fields, and therefore is a much purer kind of programming software in its essence. This aspect, while on the one hand broadens the range of its programming possibilities, on the other hand clearly complicates the process, especially from those who are not accustomed to these approaches, or at least to this specific software. Another fundamental difference lies in being a software that mainly pertains to the temporal aspects, although it effectively integrates the spatial part with which it must interface. This aspect shifts the context of its use more to the simulative and experiential side of the represented space, on which the work that follows is based.

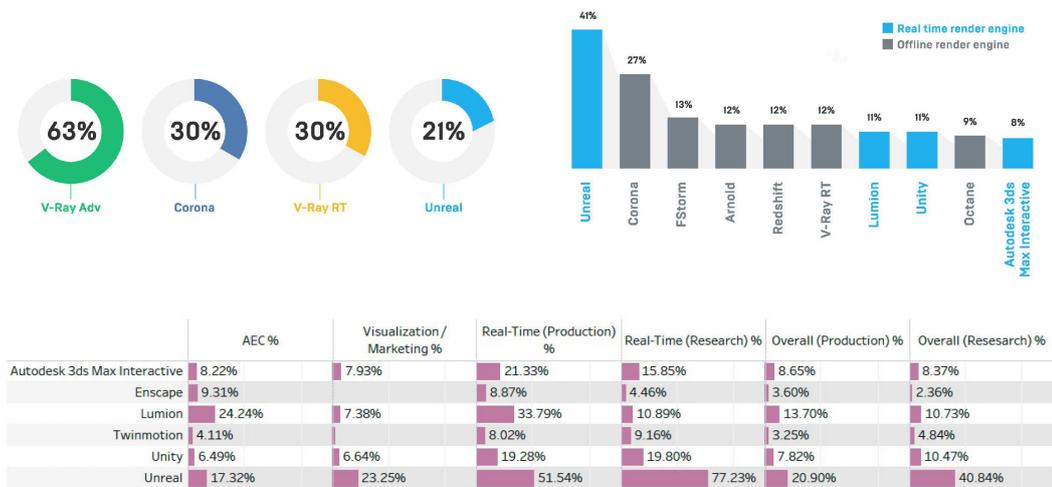


Figure 2: [Top Left] Percentage of use of rendering engines for visualization purposes. [Top Right] Percentage of experimental use of rendering engines. [Down] Percentage of use of Real Time Engines between different contexts. (Images above taken from <https://www.unrealengine.com/en-US/blog/cgarchitect-survey-shows-shift-to-real-time-rendering>) (Image below taken from <http://www.cgarchitect.com/2018/02/2018-architectural-visualization-rendering-engine-survey>)

Fig.3 shows a general mapping of the main functionalities within UE4 (Unreal Engine release 4). This map was based primarily on those features useful to architectural and geometrical practice. Several other features are available, there are more than two hundred plugins available and it is not excluded that combined together they could generate other useful tools. However, it should be noted that the true strength and effectiveness of what can be pro-

duced with this software, outside of mere visualizations or architectural walkthroughs, lies in the combination of these tools among them, thanks in particular to the functionalities of scripting. A general classification can be done within two categories: *visualization* and *parameterization* but in a way it's a simplification since there are other functions in-between them.

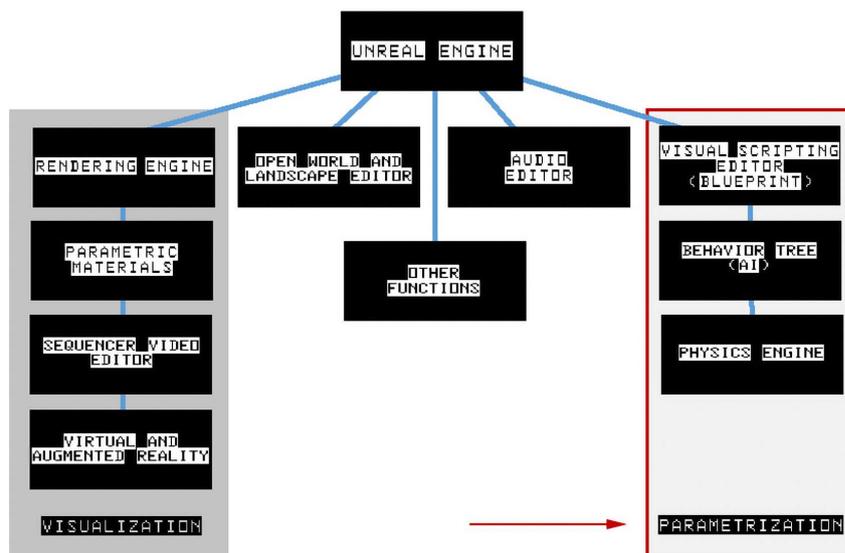


Figure 3: General mapping of Unreal Engine: right block shows the set of functions we used

Playing Architecture

For the Master dissertation, as well as for this paper, a closer look at the *Unreal Engine* release 4 (UE4) has been taken, as a tool for creating a real-time design environment and for using Artificial Intelligence (AI) technologies to represent users' flows in the space, which can be adopted to carry out design strategies and to evaluate design options. For this purpose, several simulations have been developed, either considering the pedestrian planar flows interlinked with the form of space, either parametrically generating spaces on the bases of the number of users. The first series of tests is based on pre-assigned spatial contexts. In order to test the AI programmed sets, different situations were figured out and modeled in advance. Given the assigned space, a series of points has been subsequently assigned, working as 'attractors' according to possible users' interest locations, and a virtual robot (silhouette) has been placed to explore the various possible paths, based on a random sequences of movements towards the assigned attractor-points. In order to graphically represent the visual simulations, the silhouette has been equipped with a tracing video-camera system showing at the same time its movements and the scene from the camera point of view, and allowing to reproduce in real-time the flows as graphic diagrams in the space. The AI system was also tested in a 3D spatial context characterized by differences in heights, such as inclined corridors, vertical lifts, and so on, connecting the floors. A second series of tests has been carried out considering the inverse process, that is, implementing a generative system able to create new spaces, such as rooms and paths, according to the needs emerged from the real-time analysis of the parametrically assigned users' flows. Therefore, a generative algorithm was set, able to update the geometry of space according to the number of people supposed to 'need space': in other words, space expanded according to the number of users. This generative process is based on a preliminary evaluation of the entire scenario, which essentially controls the generation itself in order to match 'rooms and paths' with the number of users. What we developed here with *Unreal Engine* would only show the power and the potentiality of this typology of software, which is still to be fully discovered, since it has been available only few years ago, and the software houses are only recently getting increasingly interested in the architectural field. However, considering the present state of art, we tried to propose some tests on if and how it is possible to use the system outside of its native target environment, adapting it to an architectural design spatial context. In our case the Master thesis, which was at the origin of this work, was confined to focusing on the use of

UE4 to realize an AI system helpful to represent and control - visually and parametrically - pedestrian flows in a three-dimensional environment, either pre-existing or generated according to specific inputs. More generally, linking analysis and project, especially in more complex scenarios, it can serve as a tool for mapping and analyzing architectural contexts, as well as for implementing, verifying, and comparing design choices, that is, efficiently sustaining the whole chain of the architectural design process.

Artificial Intelligence: NPC and Behaviour Tree

In order to develop the simulation mentioned above an AI process has been set and programmed. The simulation is mainly composed by three parts: the logic of how it works (Behaviour Tree or BT), a virtual actor (Non-Player Character or NPC) that can represent visually the result of the Ai logic process, and the basic physics (Collision Physics) of the digital environment of the scene (Map or Level). The First step has been about setting in the scene a NPC, shown in Fig. 4 and formed by the components appearing in the drop down menu placed on the top in the same figure.

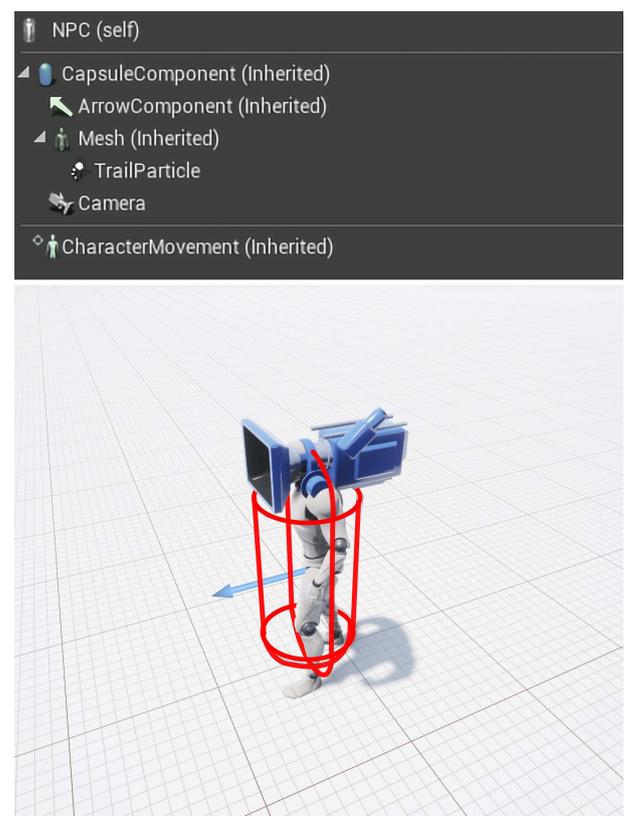


Figure 4: [Above] NPC components list, [Below] NPC visual representation (Camera remain hidden during simulations)

The mesh representing the body of the silhouette, embeds the physical information of the body including human joints and articulations and their possible movements in space. (Fig. 5)

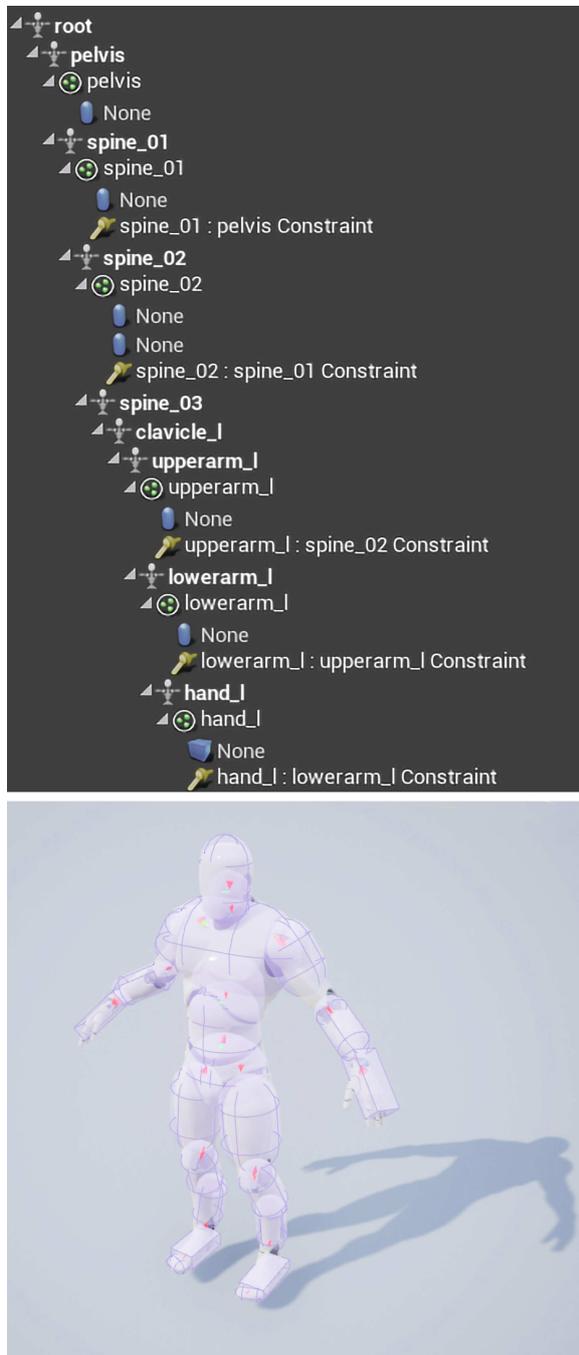


Figure 5: [Above] NPC partial skeleton composition list, [Below] NPC with hidden overlapped collision physics

The Arrow vector gives indication of the primary movement direction, and a capsule component is also used to trigger collision events or even understand where other NPCs are located in the scene. This function has not been used at the moment for the following tests, due to the particular induced behaviours when two or more NPC's path direction collide, and NPC simultaneously take the same decision to avoid the other(s). A video camera allows to record the AI movements, and finally a *trail particle* is set, which allows to trace the movements in the space basically linking the coding part with the representative graphic part. A *collision physics* useful to avoid overlapping geometries has been inserted in the scene by superimposing to all the solids a hidden simplified mesh of their geometry itself (see Fig. 5 in purple the simplified body mesh of the NPC)

Then, in order to represent flows in space, the first necessary step was to program an AI system which simulates a hypothetical logical process for which an individual tries to reach a point or an array of interesting locations by traveling in the fastest way and considering obstacles and areas where the passage is prevented for various reasons. In UE4 it is possible to obtain this result by using two integrated functions, the Blueprint Editor, that is, a visual scripting tool used to code functions, and the Behavior Tree (BT), which manages the programmed functions with a system of structural nodes. The Fig. 6 [Top] shows the BT system that simulates the user behavior, while in the Fig. 6 [Center and Down] some blueprints that manage specific behaviours.

In simplified terms the AI operation is based on the following logic: recalling by function the NPC; loading in memory the NPC location point and his destination point, previously set; imposing the creation of one or more vectors that connects the two points; making the NPC move towards the first point previously set, along the generated vector. Once at destination, a certain waiting time has been set (it can be avoided) before the following command is activated. At this point the AI must be activated to understand which is the next destination point, in order to repeat the process a number of times up to the end point of the array. Once it gets the last point, it would be possible to decide whether (or not) to reverse the preset points or make them repeat in loop. Another alternative is to randomize the system of the preset points, so that there is no predefined path by increasing the number of possible displacements, then without binding the movement to a predetermined sequence. This last option has been chosen in order to replicate a hypothetical not predetermined behaviour.

About the physics, it has been set what is called *collision physics* that gives to the elements in the scene the property

of “solidity” as in the real world, in order avoid geometries and element in the scene to interpenetrate, passing through each other. Also the *gravity force* has been set. Finally, closely linked to the *collisions*, and in order to be able to make the AI understand if there are obstacles along the path that the NPC will have to travel, a volume called “Nav Mesh Bounds Volume” has been added to the scene. It allows to create a mesh calculated on the basis of the objects to which a collisions physic was applied and which reside

within this volume. The generated mesh will be parallel to the surfaces of the scene, considering also the angles of the slopes. If they are slower than the established one it will be walkable. If higher it will be interpreted as obstacles. Therefore, this process will determine the space within which the AI can move, after eliminating the obstacles, and on which the AI logic will calculate the route towards the preset points.

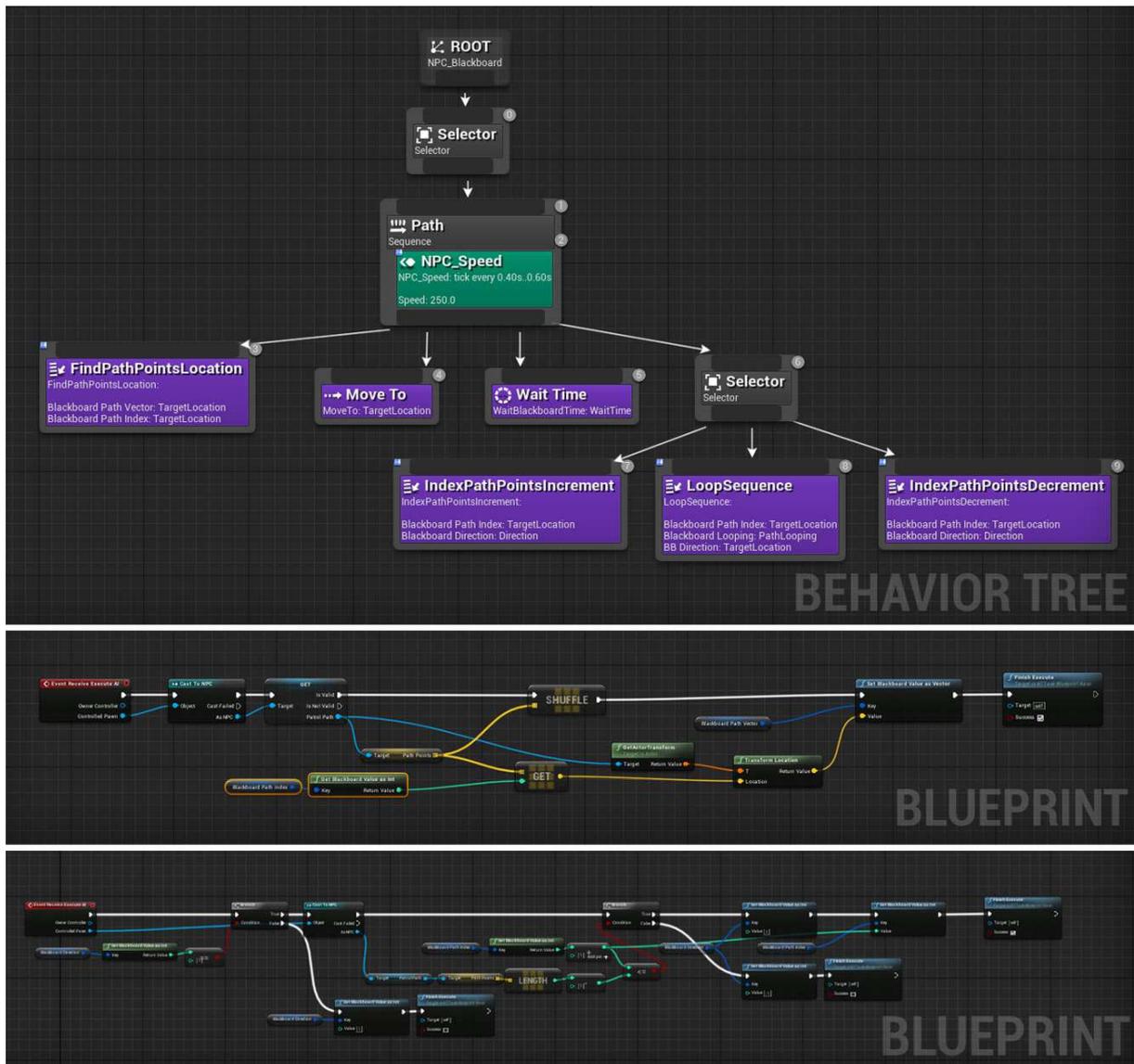


Figure 6: [Above] Behavior Tree programmed to simulate user flows. [Center and Below] Examples of blueprints that manage some nodes of the BT.

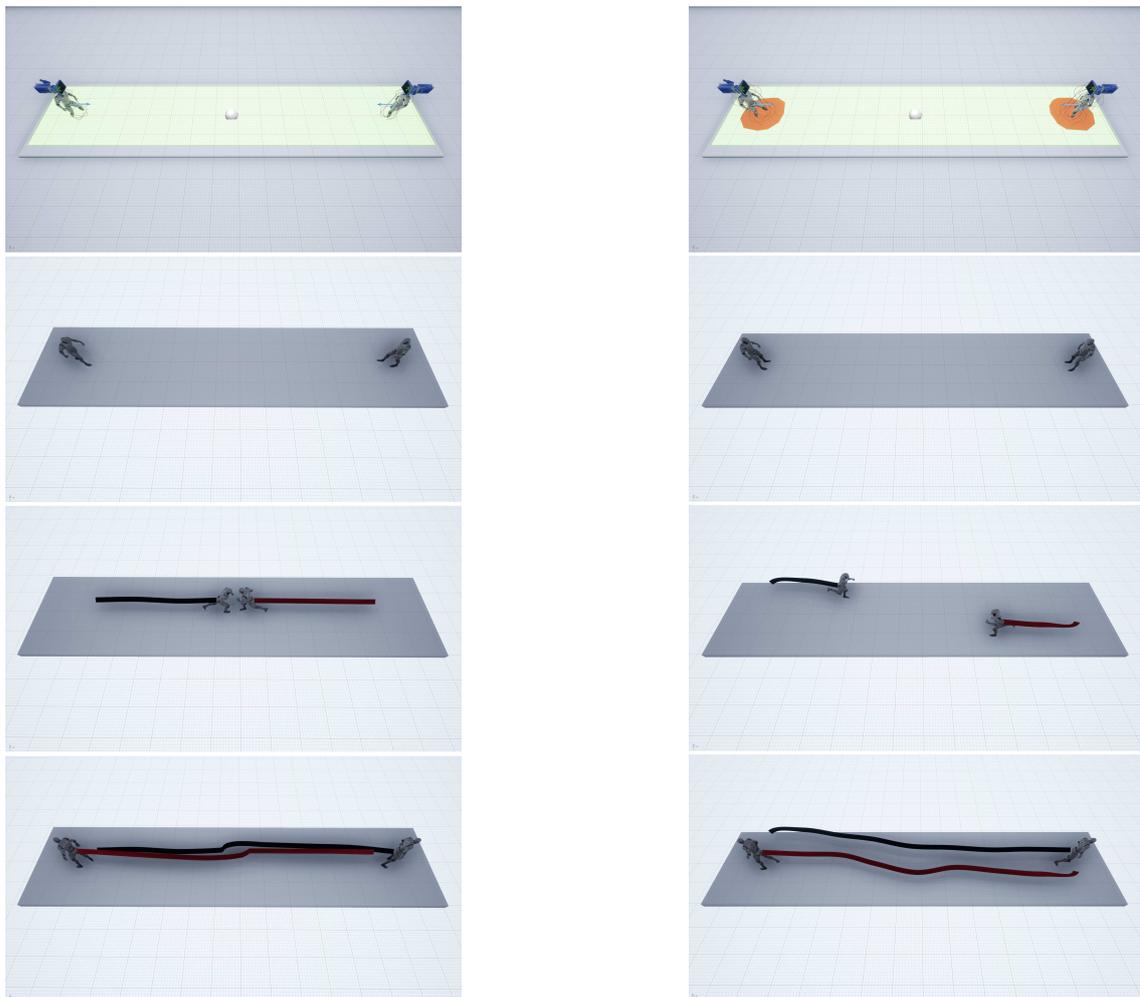


Figure 7: [Sequence Left] Simulation with collision physics applied only to the body of the NPCs. If they collide they change their trajectory due only to physics. [Sequence Right] Collision physics applied to a larger area and calculated dynamically by the Nav Mesh Bounds Volume frame by frame. The AI that manages the NPC trying to find a way not to collide, since they see each other as obstacles.

Form to flows

In order to test the AI programmed, a hypothetical residential interior space has been modeled, to which the volumes of some furnishings have been added. A series of points were subsequently assigned, within the housing hypothesis, in places of possible interesting areas. (Fig. 8)

The sequence of movement of the silhouette towards these points has been randomized so that it does not follow a pre-determined sequence, but once the simulation has started it can move freely and randomly to any of the preset points inside the Nav Mesh Bound Volume. (Fig. 9)



Figure 8: Hypothetical residential plan with the location of the 'PathPoints'



Figure 9: *Model with indicated in green the 'Nav Mesh Bounds Volume'*

Once the first point is reached, the sequence will be shuffled again so that from any point the silhouette can reach all the others, even the ones visited before. This prevents a previously reached point from being ignored at a later stage. It also prevents the same route from being traveled over time, and consequently allows to increase the number of possible routes that can be considered. Moreover, the choice to randomly determine the path of the NPC allows to evaluate the simulation from other points of view as a privileged point of view it is not predetermined. This choice, although it may seem simplistic, gives the AI a certain unpredictability that partly mimics human behavior. In order to be able to graphically represent the simulation, the silhouette has been equipped with a system that traces its movement during the displacement, and allows to visualize the flows in space for the time of the simulation. Since the simulation would go on to infinity, it was programmed to suspend it as soon as all the points in the scene were reached from each position. It should be noted that if the AI is set on the random proceeding of the points, each simulation is different from the other even though sharing similar patterns, while the most traveled areas, as well as those that are not, are graphically highlighted. In the residential housing hypothesized it can be seen that both in relation to the fastest routes and in relation to obstacles such as walls and furnishings, the corridor is one of the most crossed areas by the silhouette, as it is supposed to be in a real similar spatial situation. Fig. 10-13

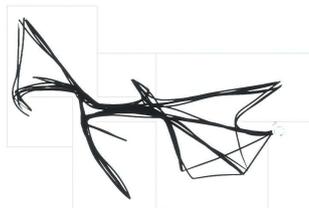


Figure 10: *AI calculated flows without considering furniture*

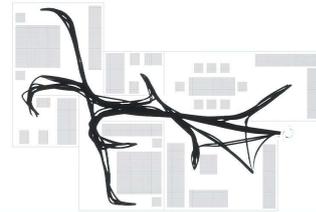


Figure 11: *AI calculated flows considering furniture*



Figure 12: *Perspective view of the AI calculated flows without considering furniture*

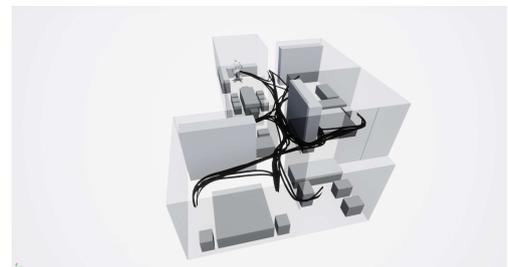


Figure 13: *Perspective view of the AI calculated flows considering furniture*

In another simulation, the AI was tested in a context with differences in heights reached through inclined corridors, or vertical lifts. Even in this case the movements' sequence was randomized for the same reasons as in the previous case. The representation of the flows appears here in three dimensions, according to the differences in high, then we have a properly said three-dimensional dynamic spatial scenario. The AI is able to calculate the hypothetical flows given any form of space. The (Fig. 14) show the model on which the test was carried out, (Fig. 15) the set of points that allowed the AI to address the Silhouette on the basis of the Nav Mesh Bounds Volume (Fig. 16) and finally the graphic representation of the resulting flows, represented both in perspective (Fig. 17) and in top view (Fig. 18).

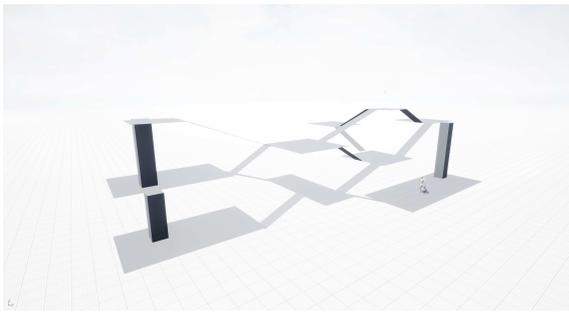


Figure 14: Hypothetical model designed with different levels

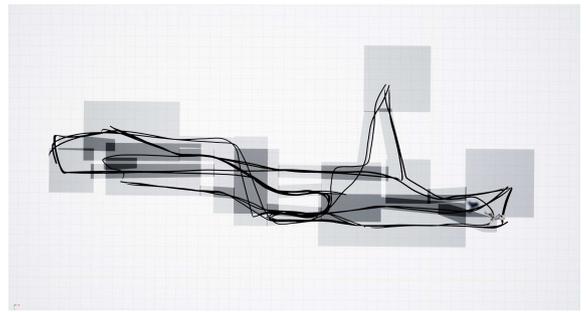


Figure 18: Zenith view of the model with the AI calculated flows represented

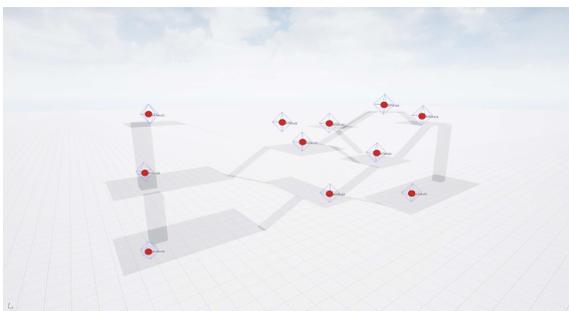


Figure 15: Model set with 'Path Points'

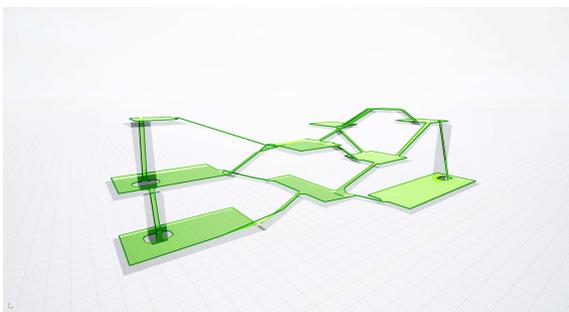


Figure 16: Model with indicated in green the 'Nav Mesh Bounds Volume'

Flows to form

Considering the inverse process that can be described as the determination of form given the flows, a generative system has been programmed to create, on the basis of various parameters (Fig. 19), a spatial system based on several floor levels, where some spaces are connected by ramps. The generation of spaces is based on a system of nodes (Fig. 20) which corresponds to hypothetical spaces of interest and which subsequently correspond to the points that the AI identifies as areas to be reached.

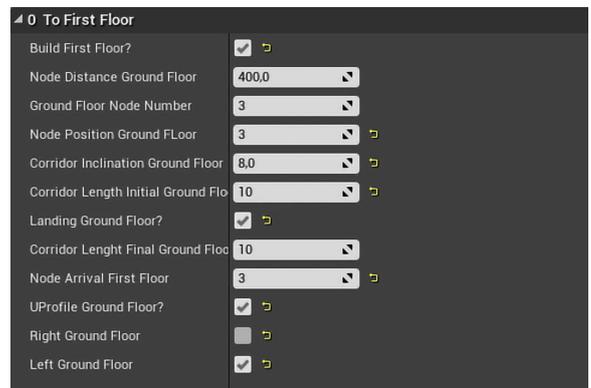


Figure 19: Parameters: generation settings

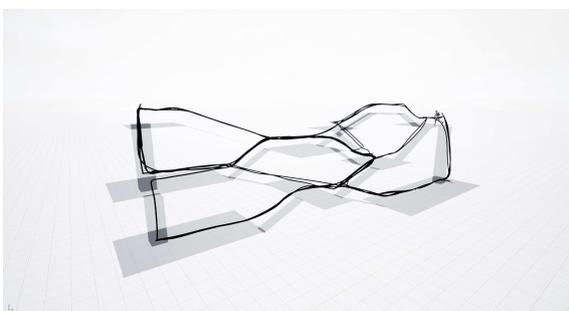


Figure 17: Model with the AI calculated flows represented

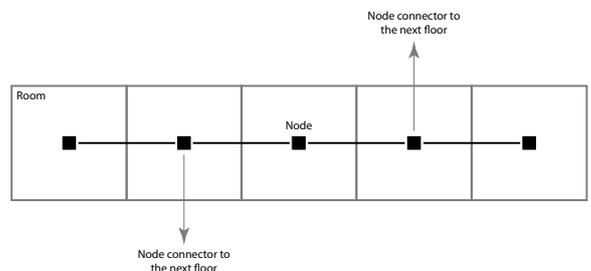


Figure 20: Node based generation system

The number of nodes also corresponds to the number of people who are supposed to reach a certain level or the probability of this happening. Knowing the points from where people start, up to where and in what number they have to arrive, is the information based on which the form of rooms and connections is generated and subsequently the AI is asked to determine the generation of the flows. Once the inputs have been set, we have to make sure that the generation will take place within the Nav Mesh Bounds Volume previously placed in the scene. Then, after starting the simulation, based on the parameters previously set the system calculates the forms, which automatically and dynamically will be read by the Nav Mesh Bounds Volume, that will adapt to the geometry of the spaces just created. This system, however abstract and partly limited, allows for the creation of forms and spaces in relation to the needs. The code that allowed the generation was programmed to create the various elements in sequence and then repeat the process on each floor but varying the generation points depending on what was previously created. Starting by the planes and volumes visible in Fig. 21 and on the basis of the number of nodes set the forms could be generated. (Fig. 22) The code (Fig. 23) that forms the basis for creating a floor is essentially the same as the one that generates the next floor, except for the variation of the generation coordinates, since also the elevation is considered.

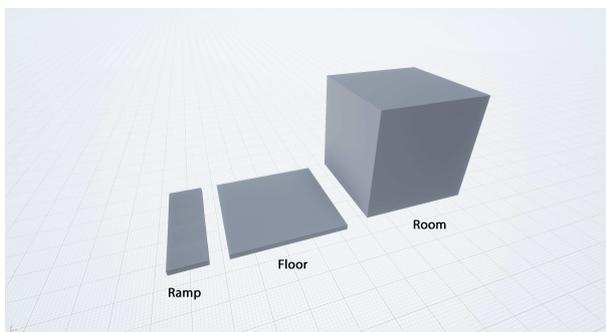


Figure 21: *Basic geometric settings at the starting point*

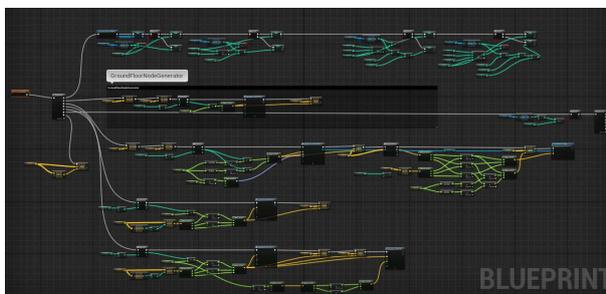


Figure 23: *Code scheme for the generation process*

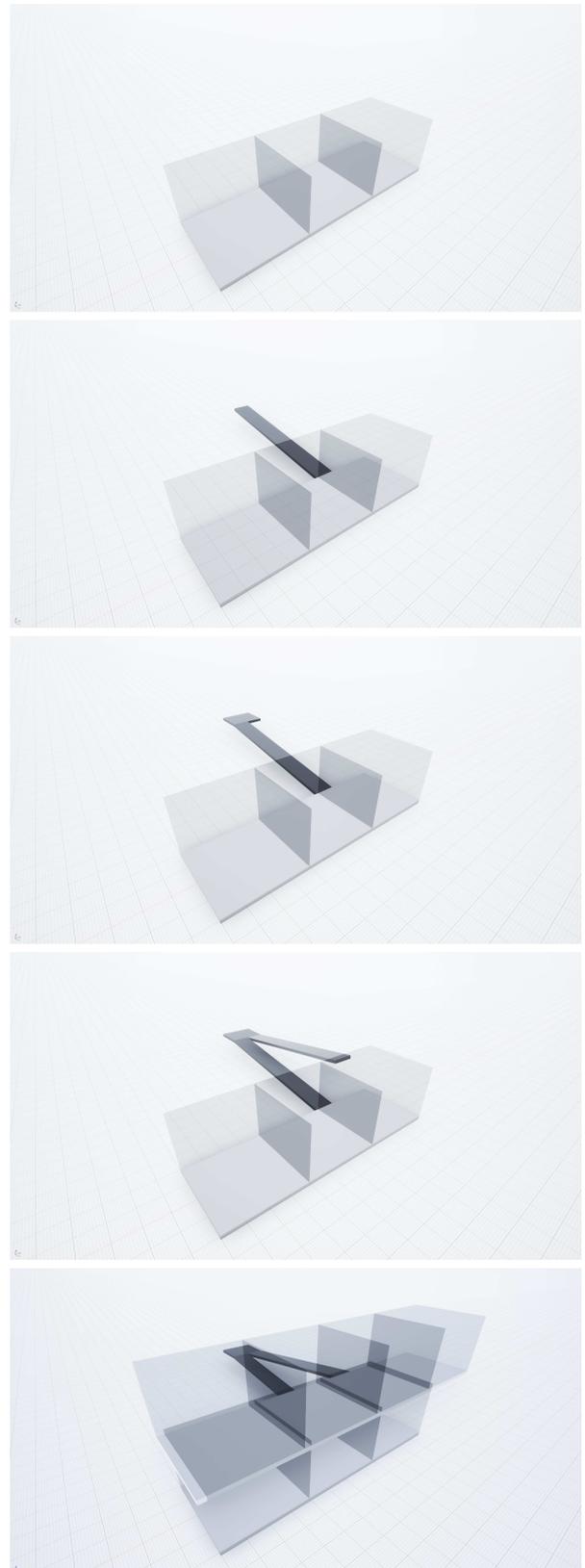


Figure 22: *Generation sequence*

So, each subsequent floors can be repeated and extended to the number of floors desired, which are placed as a series by adding the appropriate variables for each new floor. Fig. 24-25 show some generative examples obtained by varying the settings programmed, in synergy with the AI which automatically identifies the destination points generated.

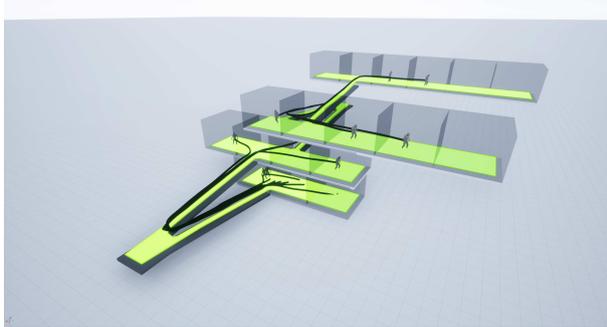


Figure 24: *Forms generation with indicated the dynamic Nav Mesh Bounds Volume and AI flows calculation*

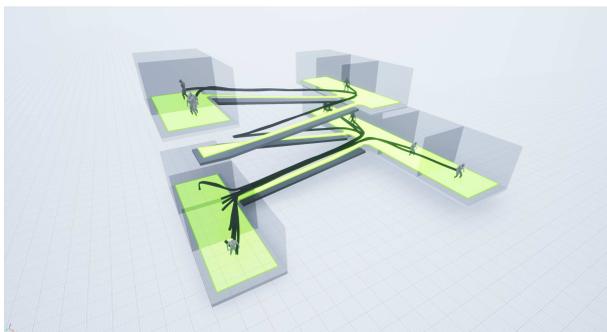


Figure 25: *Forms generation with indicated the dynamic Nav Mesh Bounds Volume and AI flows calculation*

A further development on the generative theme, based on linking users and flows, was in programming a system that is generated according to the number of people who are supposed to need space. In other words, space expands according to the number of users. The generative process that led to the realization of the architectural elements is based on the same script as in Fig. 23 placed in series but adding a piece of script that allows to indicate to the system if and when there is a need to add a new floor. In other words it activates (or not) the generation of new floors on the basis of an evaluation. This evaluation is managed by a function placed at the beginning of the entire generative process and which essentially sets the number of floors and the number of nodes desired for each floor based on the number of users. The generative sequence acts to always match the total number of nodes with a certain limit number of users preset. Fig. 26 shows in a graphic sequence the relation between users and the expansion of the space.

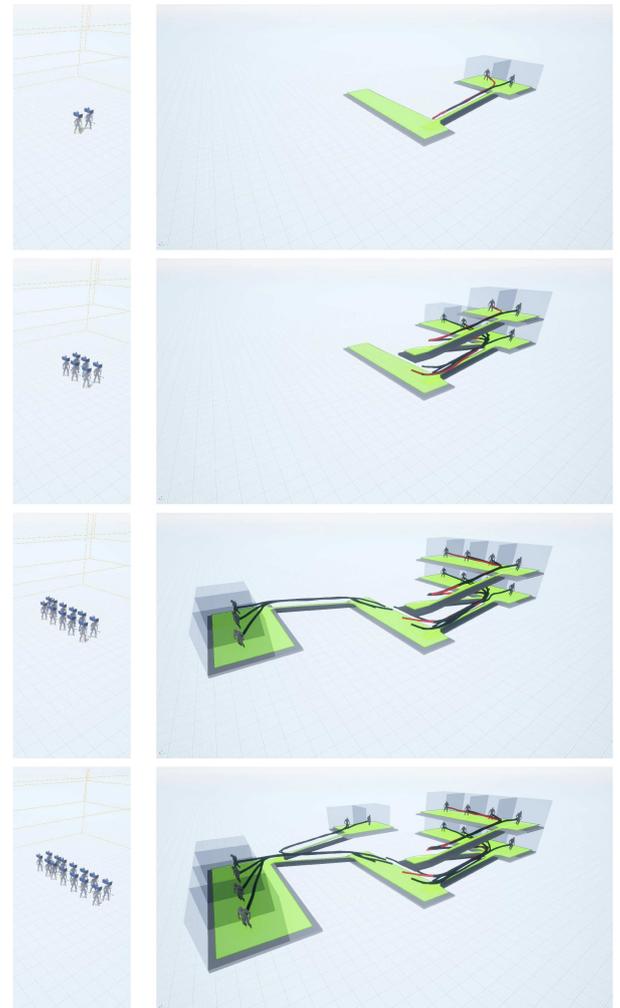


Figure 26: *Simulation of the sequences generating spaces in relation to the number of users added to the scene with subsequent simulation of flows through AI calculation*

Campus Leonardo's main square at the Politecnico di Milano: flows simulation

The experimentations presented until now have been carried out either starting from pre-existing spatial situations or moving toward the hypothetical generation of new spaces, on a reduced scale. We present now a test of the AI system carried out on a real case and on a wider scale. Piazza Leonardo da Vinci, that is, the main square in front of the Campus Leonardo of the Politecnico di Milano (head-quarter) is taken as a test area. Several simulations have been made to test the AI in this context. As in the previous cases, a series of points have been set which are visible in Fig. 27 which corresponds to the main access points of the Politecnico and to all the possible roads that can be followed to exit or enter the square. The digital model of the

space has been developed integrating data from the municipal vector maps, Google Maps and with a real life survey by photo modelling. The first simulation was done by controlling the flows, and directing them to predetermined points of interest without moving the silhouettes towards other points. Then a further simulation was done making people move from one point to another in Piazza Leonardo. In this case the NPC could reach a point and then randomly choose another one to go to, in this way the same NPC

could simulate a variety of user behaviours. Several situations were considered: first silhouettes moving from some designated points to other designated points, and finally towards any points at random. All the simulations were then repeated increasing the level of restrictions in the paths: considering green areas, benches, and driveways as obstacles. A summary of some results of these simulations can be seen in Fig. 28.

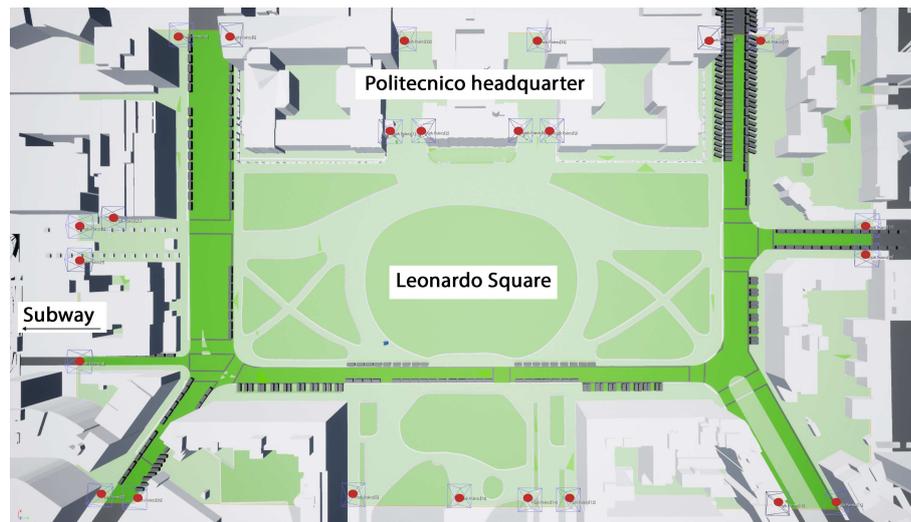


Figure 27: *Leonardo Square model with the set of 'Path Points' for the simulation flows through the AI*

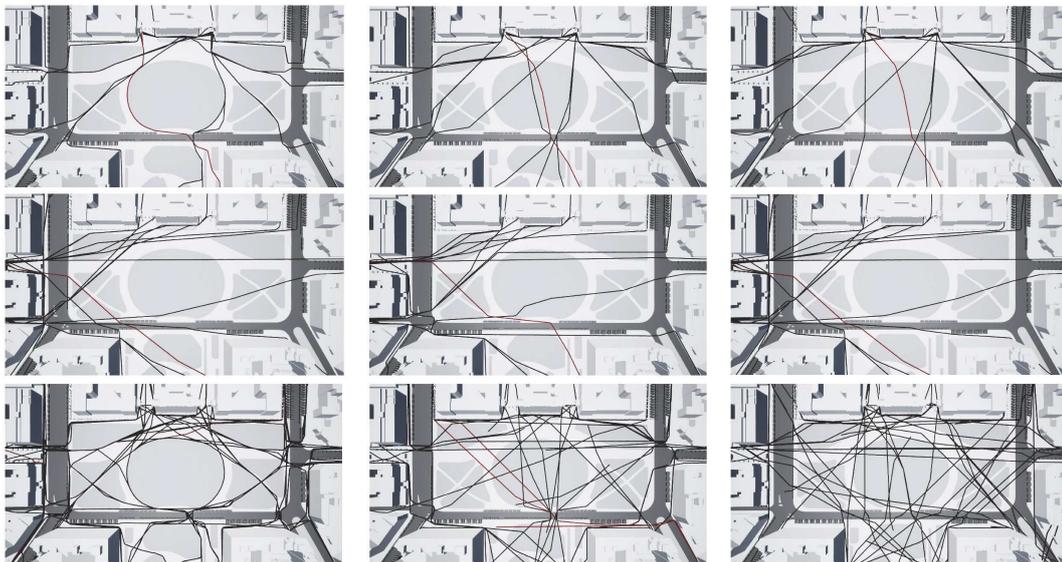


Figure 28: *[From Top to Down] Flows simulation on Leonardo Square: from the headquarter accesses to the access/exit points of the square, from the subway to the access/exit points and randomly. [From Left To Right] Considering both the flower beds and the streets as not viable, considering only the roads as viable, completely free.*

Finally, a hypothetical movement of a NPC was reproduced by superimposing the virtual animated walk of the NPC and the corresponding graphic track of the path inside a video, previously recorded in the Piazza Leonardo da Vinci. To do this, additional software programs have been used, together with Unreal Engine. Once the video in the real place was made, the camera movement was traced through a software called Boujou (Fig. 29), which allows to track the movement performed by the operator and representing it three-dimensionally in a virtual space. Subsequently, the track made by the camera at the time of recording was exported in 3ds Max in which the viewpoint was positioned at the desired height and orientation (Fig 30). Finally the file was exported in .fbx format and imported into the UE4. In the UE4 a camera was set and it was given the same movement in the space traced by the Boujou software. Moreover, the parameters of light and sunshine were set on the basis of those detectable in the real space at the time of video recording. Subsequently, the desired movement of the silhouette was recorded on a green background (Fig. 31), and finally, the final sequence was composed using the software program After Effect, by combining the original video with the one created in the UE4, appropriately cropped to eliminate the green background (Fig. 32). The result can be seen in Fig. 33.

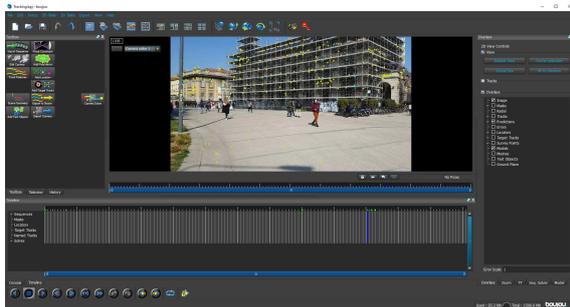


Figure 29: *Boujou interface, used to track camera the movement of a video*

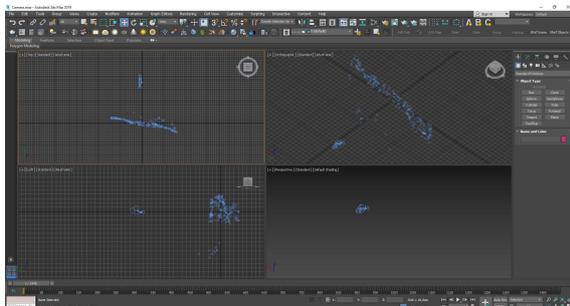


Figure 30: *Tracing camera movement exported in 3ds Max in order to set the right camera height, and for exporting the .fbx format loadable into Unreal Engine*

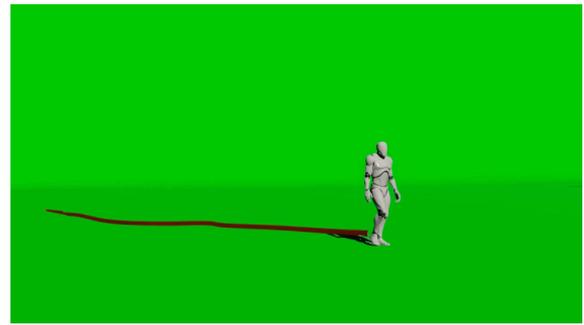


Figure 31: *Video screenshot with a green background rendered in Unreal Engine using the traced movement of the camera*

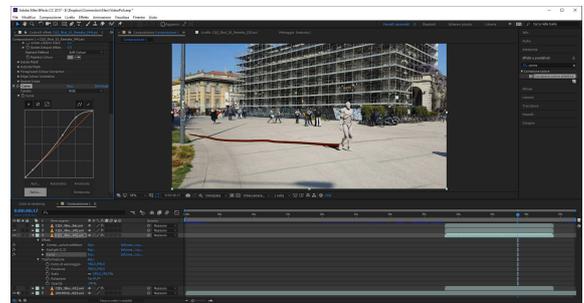


Figure 32: *Overlay of the original video and green screen video using After Effect*

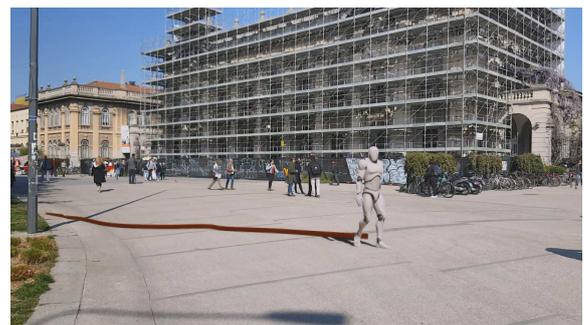


Figure 33: *Overlapping level and final result*

Possible future developments

In the area of AI research, further developments can be expected in different directions, either focusing on the simulation of contexts and their uses, either on types and behaviour of the users, as well as on the complexity of their interactions. Many of these are already at work in the world of Video Games. Concerning the path point they

could be set in a probability reach state, in other words each point would have a probability chance to be reached by the NPC, which could even change in relation to certain external factors. In the architectural field, a welcome development can of course deal with more advanced aspects, like patterns referring to perceptual senses (sight, hearing, or tactile, also including external events, and so on), and on their translation into the virtual environment, based on similar parametric operations. The integration of senses could suggest to the based subjects (NPC) changes of direction, path or any other behavioral reaction according to the context topic. Additional forms of AI can arise from the combination of more senses, as well as from other external factors generally attributable to virtual atmospheric events or other AI behaviours which could integrate the possibility for the AI to predict hypothetical future events and scenarios and take decision based on that. Strictly linked with this last topic is the psychological behavioral factor both related to the movement of the individuals in relation to the masses and vice versa. This point introduces a relevant and very actual subject linked to the “realism” of the context of the AI environment. In this case, given the difficulty of translating behavioral psychological aspects into

appropriate descriptive codes because of their probabilistic nature, it would be appropriate to introduce neural network technologies based on machine learning and deep learning systems, which we are aiming to do in the future.

Conclusions

What this study has brought to light is the possibility of using tools currently used in the video game world for architectural analysis, design or teaching purposes, bringing a series of potentials that are still almost rarely found among the tools currently used in the architectural design contexts. It has been shown how through the use of UE4 it was possible to realize an artificial intelligence-based process able to represent three-dimensional flows both in pre-existing and in generative models according to design needs, as well as to expand a space in relation to an increasing number of users. More extensively this system can be used for analyzing, comparing and even introducing new design choices, resulting very useful in contexts characterized by high levels of complexity and in the advanced stages of the design development.

References

- [1] E. AARSETH, Allegories of space. The Question of Spatiality in Computer Games. In: Von Borries, F. Walz P. S., Böttger, M., *Space time play, computer games, architecture and urbanism: The next level*, Basel: Birkhauser. p. 44, 2007.
- [2] C. C. ABT, *Serious games*, University Press of America, 1987.
- [3] A. BARICCO, *The Game*, Giulio Einaudi Editore, 2018.
- [4] E. CHAMPION, *Critical Gaming: Interactive History and Virtual Heritage*, Routledge, 2015.
- [5] G. FRASCA, *Play the message: Play, game and videogame rhetoric*, IT University of Copenhagen, Ph.D. Dissertation, 2007.
- [6] M. GEROSA, *Second Life*, Meltemi Editore, 2007.
- [7] M. HEMMERLING, L. COCCHIARELLA, *Informed Architecture. Computational Strategies in Architectural Design*, Springer, Cham, 2018.
- [8] J. HUIZINGA, *A study of the play element in culture*, Switzerland, Routledge & Kegan Paul Lond, Boston and Henely, 1944.
- [9] J. JUUL, The Game, the Player, the World: Looking for a Heart of Gameness. In Level Up: *Digital Games Research Conference Proceedings*, edited by Marinka Copier and Joost Raessens, 30–45. Utrecht, Utrecht University, 2003.
- [10] D. G. LOBO, Playing with Urban Life, How SimCity Influences Planning Culture, In: Von Borries, F. Walz P. S., Böttger, M., *Space time play, computer games, architecture and urbanism: The next level*, Basel, Birkhauser, p. 206, 2007.
- [11] W. MAAS, Spacefighter: A game for the evolutionary city. In: Von Borries, F. Walz P. S., Böttger, M., *Space time, play, computer games, architecture and urbanism: The next level*, Basel, Birkhauser, p. 362, 2007.
- [12] MVRDV/DSD, *Spacefighter. The evolutionary city (game:)*, Actar-D, Barcelona, 2007.
- [13] M. NOVAK, Liquid Architectures in Cyberspace, in Benedikt M. (ed.), *Cyberspace: First Steps*, MIT Press, Cambridge MA, 1991, 225–254.

- [14] S. PORRO, *Playing Architecture. Using Real Time Engines as Operational Tools for Architectural Design*, (Master dissertation, tutor Prof. L. Cocchiarella), Politecnico di Milano, School of Architecture Urban Studies Construction Engineering (AUC), 2019.
- [15] J. SANCHEZ, *Block'Hood. Developing an architectural simulation video game*, 2015.
- [16] J. SANCHEZ, *Gamescapes*, Studio Course Master, March Graduate Architectural Design, The Bartlett, UCL, London, 2013.

Simone Porro

e-mail: simone1.porro@mail.polimi.it

Politecnico di Milano, School of Architecture AUC,
Via Golgi 42, 20133 Milano, Italy

Luigi Cocchiarella

orcid.org/0000-0002-3201-4189

email: luigi.cocchiarella@polimi.it

Politecnico di Milano, Department DASTU,
Via Bonardi 3, 20133 Milano, Italy