International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.134.03

The use of characters in 3D environments

Jorge Madera-Valdez[1], Hernando Leo-Valenzuela[2], Francisco A. Madera[3]
[1,2]School of Architecture, Universidad Marista, Merida, Yucatan, Mexico
[3]Mathematics Faculty, Universidad Autonoma de Yucatan, Merida, Yucatan, Mexico

Abstract- In Architecture, physical and digital models of the design are essential to explore the effectiveness and possibilities of the proposal. Navigation in virtual environments offers and immersive approach to analyze projects. Traditional visualization consists of a camera view that moves around the model. However, adding characters in the virtual environment helps to make the visibility more accurate, understand the functionality and test more features such as scaling, walking flowing, traffic crowded, among others. In this work, the digital model of the proposal for a rail station in Izamal, Mexico was developed, and techniques for the creation and usage of character animation are applied to enhance the design through virtual visualization tests.

Keywords – Computational Geometry, tridimensional design, character animation

I. INTRODUCTION

During the Architectural design process, the dimensions of the spaces are a crucial aspect to test the quality and functionality of a project. The main promotional product is the virtual environment, and application to visit the building by 360 degrees view or manipulating the virtual camera. Architects use digital tools to navigate proposals in a virtual environment. This helps to stablish a visual understanding of the relation between a project and its surrounding to recognize enhancements opportunities in its context such as topography, sun, vegetation and weather patterns to name a few. All these elements should influence architect's design process and decision-making.

Navigation is one of the most popular research topics in the field of crowd simulation that attracted the interest of researchers and designers as it helps to comprehend the behavior of the characters in a scenario. The nature of visual appearance indeed affects the final appreciation and then the model design, the local and global information appearances. These findings contribute towards further understanding what information people use when interacting with others.

By inserting characters in a digital model, the appearance becomes more accurate so that designers can realize some other ideas to be considered. By adding motion to the characters will enhance our appreciation since in the real life, humans will use and move around the building. Effective visualization can reduce the misinterpretation in spatial and logical aspects. Jain et al. [1] present a time scaling algorithm for latency management and synchronization between multiple avatars, while they interact using objects in the virtual environment.

Pedestrian and crowd simulation have applications in a wide range of industries, including pedestrian facility suitability and capacity, computer graphics and gaming, the social sciences, and engineering. This broad range uses have led to extensive research into how crowds and pedestrians move around and interact with their environment. This simulation enables the effective development of solutions for planning, managing, and optimizing pedestrian flows in public buildings like airports, railway stations, shopping malls, and stadiums.

Hitachi [2] describes an optimizing simulation of movement in buildings by using people flow analysis technology that can evaluate and investigate the influence that elevator, circulations and facilities have on the paths that people take through buildings. The characters employed are rigid, they only can be translated and rotated, there is a lack of deformation (animation) to identify actions such as walking or running.

Most of the studies about crowd behavior are related to path planning, collision avoidance, autonomous agents, social interactions among groups, and personality models. When a person looks at a scene, the brain cannot perceive the whole scene and analyzed it instantly; instead, only the area within a two-degree range from the center point can be focused. There is a lack of detailed information about the creation of characters and addition to a virtual environment. Quite a lot of work has been done about the human motion employing capture devices both with markers and without them in several areas such as entertainment, sports, rehabilitation [3]. Also, with graphics hardware we can find the work of Kavan [4] who introduced the skinning with dual quaternions for an improvement of human animation.

This paper discusses the use of human characters in digital models as the build environment is meant to be used for people. By focusing on task-based simulations and evaluations, designers can improve spatial conditions for the occupant to perform their activities. Simulating people throughout the virtual environment has been of interest for architects and engineers, but it is hard to achieve such simulation. Rather than simulations are integrated in CAD (Computer Aided design) programs, CAD models are integrated into a game engine.

This article proposes a methodology to include animated characters into 3D models used for building design. Main contributions include the following: the design and construction of the 3D model, a strategy for character development, the insertion of the animated characters in the model. The model was constructed by using AutoCAD, and textures were applied with 3DSMax, characters were created and animated using MakeHuman, Blender and Unity3D, and finally, they are inserted into de model, creating a virtual environment of a building. Some scripts are implemented to adjust the character motion.

The paper is organized in the following manner: Section II presents the foundations of navigating in a virtual environment. In Section III, the Framework Section, the three modules are explained; character creation, animation definition and character insertion. Finally, we conclude with a review and analysis of the findings of the research.

II. NAVIGATING IN A 3D VIRTUAL ENVIRONMENT

The design process in Architecture evolves constantly due to developing tools available in the field of 3D modeling and rendering. Building digital models help architects to test several aspects of the design such as solar incidence and shading in the interior spaces, general views and dimensions. Inserting 3D characters in the model improves the understanding of the whole project and allows noticing enhancement opportunities before starting the construction process.

The presence of people in spaces are highly influenced by the geometry and topology of physical space and infrastructure, the geometry and topology of humans' dynamic, and the geometry and topology of human behavior during the motion simulation. While this is not meant to replace real-world experience, it does provide possible relief from the financial costs associated with the building proposed. Our objective was to allow the designers of the current project to visualize what they could previously only imagine.

In the field of architecture and engineering, simulating projects in a virtual environment helps to simulate aspects such as suitability, functionality, and production costs.

Architectural ideas must be visually communicated so that 3D virtual environments and related technologies such as virtual reality have been considered a beneficial strategy for architectural design since the customers can appreciate in depth the building project.

The framework presented in this article can serve as a tool to evaluate routes and paths to improve pedestrians flow in the environment. Our framework can be used by designers when calculating the time for people displacement and to contribute to the adoption of performance-based building codes that better suit the nature of contemporary architecture. Serious games have been developed for a range of several scenarios [5]. These game-based scenarios serve for several purposes, including the testing of evacuation plans, the visualization of disaster scenarios, and training individuals through experimental learning environments.

The definition of the walking analyses is paramount to improving our understanding of human movement in built spaces. Modeling and simulating human movement reveal the possibilities to improve spatial quality and the role it plays in dictating human behavior within it [6].

We describe here our efforts to produce a workflow that highlights the importance of accurately capturing the geometry and topology of spaces and demonstrate how these representations impact the evacuation (offsets, displacements, visualizations, etc.) calculations, and address the challenges associated with modeling and animation.

Visualizing 3D models of buildings is an important part of the process of understanding them. We present some basic visualizations using Unity3D game engine, but there remains much to be accomplished. Game engines offer a richness of visualization potential, that allow inserting digital characters into to the scenario.

The use of characters in virtual environments has been investigated for ages. Videogaming has been one of the most demanded applications in entertainment field and most of them include characters. Serious games can also offer a variety of approaches with characters, two of the most known applications can be found in the SIMS game [7] and in the Second Life [8] game. The former allows to enjoy the possibility to create and control characters in a virtual world, where you define your parameters. Users express their creativity by personalizing the virtual people, allowing building a house, and develop relationships. In Second Life, users can create social experiences and 3D contents; share and sell them in the international market. Users always find something new to enjoy, from games to live music and much more.

In [9], authors propose mixed-reality humans (MRHs) –virtual humans that inhabit the users' physical space, which combine the dynamic visuals of virtual humans with the physicality of mannequin patient simulators. This hybrid configuration enables them to assume the roles of unavailable human team members, essentially allowing on demand team training.

The work mentioned in [10] shows a planned walk integrated with multimedia content during the visit at Leeds Museum City room. In [11], avatars are employed in an immersive environment for training on industrial emergencies. In [12], the proposed plugin set is to represent each real participant with a virtual mannequin whose pose reflects the real one. Other research has been focused on the use of a Kinect or a sensor device [13] to reproduce the character's motion in real time.

III. THE FRAMEWORK

There are many digital tools available to work on virtual environments. 3D modeling software, such as AutoCAD, 3DSMax, Rhinoceros 3D, Blender, Sketchup, offer a wide range of geometric tools to model objects, including curves and surfaces. The object can be modeled as accurate as desired since these tools provide the opportunity to make scripts for the geometric computations. Other software focuses on the rendering process of virtual models, they are used for applying textures and lighting to the surfaces in order to create specific atmospheres and communicate ideas. This is possible by using graphics cards and routines called shaders. The rendering process for achieving high quality images can take hours or days. An alternative that allows users to keep a fast workflow while navigating and editing their rendered models is the game engine.

Digital model and characters are imported in the game engine to generate a feasible and effective animation. In Figure 1, the three modules of the framework are depicted. The first module (Environment 3D) focuses on the 3D model design, the Rail Station. The second module (Characters) shows the character design process where avatars are constructed. The last module (Game Engine) joins the 3D model and the characters, applying animation to make a walking path.

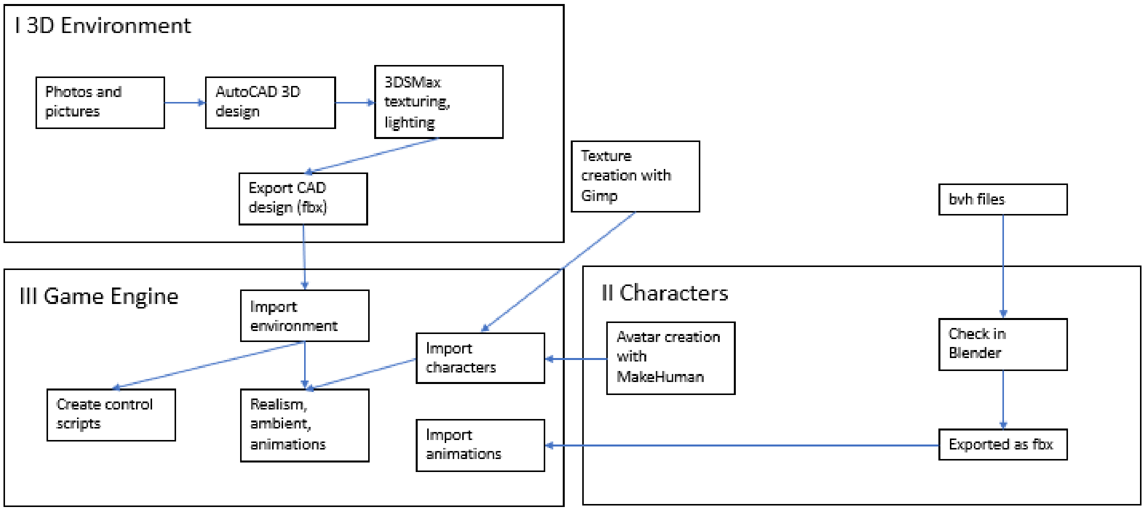


Figure 1. The three modules of the framework.

3.1. Environment 3D (module I)

Izamal is a small town in Mexico located in the southeast of Mexico. It is considered a of cultural heritage place visited by many people all the year. A rail station is proposed to connect the city with other sites of cultural value and therefore activate the economy of the touristic field in the region (Figure 2).

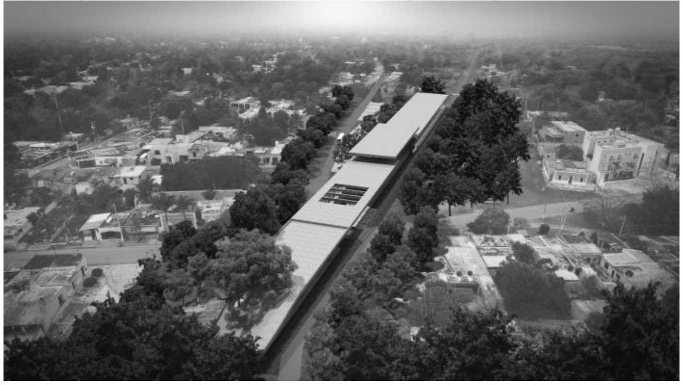


Figure 2. The Rail Station is proposed to be in Izamal city, Mexico.

Architecture students generated the 3D model as part of the design process. Authors visited Izamal city to take photos of the surrounding area, they measured the site, and collected information from local people. According to specific activities, the proposal consists of four sections. Section 1 hosts the railway station, section 2 serves as an entrance covered by existing trees, section 3 hosts art exhibitions and craft stores, and finally section 4 is a community center.

The model contains 1,400 pieces of basic geometry, such as cubes or rectangular pyramids. The model involves three main parts, the ramp, the waiting wall and the boarding area. The proposal, called Railway Portico, was shown to the community of Izamal.

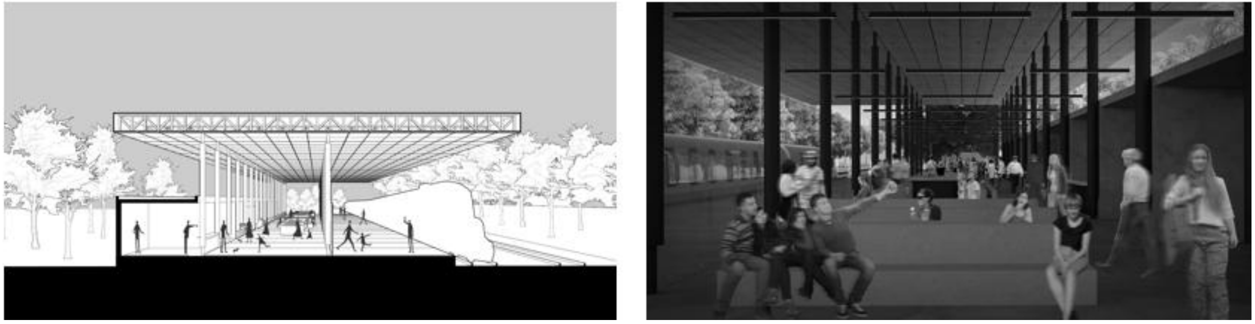


Figure 3. Arquitechture diagrams express how people interact in the railway station

Figure 3 depicts people in the Rail Station. (a) 2D people walking and idling, (b) 3D people walking. The goal is to add animation to characters and have a better perspective of the Railway station.

3.2. Characters (module II)

To construct the characters, MakeHuman software was employed. Four female characters were created with armatures included. Armature refers to the skeleton of the characters. The human motion can be divided in two main parts: animation (deformation) and geometric transformation (displacement). Animation consists of the deformation or movement of some parts of the character; for instance, the upper limbs, the head, the fingers, one hand, etc. Displacement consists of the motion of the whole character, keeping in the same pose. This could be translation, rotation, scaling, shearing, etc. Examples of deformable objects are characters, ropes, hair, elastic bodies, etc. Examples of rigid bodies are airplanes, monitor, screwdriver.

3.2.1 Animation

Skeletal animation is a technique in which a 3D model is going to be animated; the human character is composed by two main elements: a mesh that models the shape of the avatar and a skeleton that is a hierarchy of connected nodes, or bones that controls the mesh pose. The skeleton is a hierarchy of bones; a bone is described by a 4x4 transformation matrix that defines a reference frame for his children. This hierarchical approach of the reference systems, defined by each bone, permits to model any movement of a limb by a rotation matrix applied to the root bone of that limb.

To animate a character, we require a virtual skeleton to guide the vertex movements. The mesh M of the avatar is defined by a set of n polygons and m vertices M=(△, V), where the set of polygons (normally triangles) and the set of vertices are △={△0, △1, . . ., △n-1,}, and V={v0, v1, . . ., vm-1}, respectively. The four characters constructed are illustrated in Figure 4, they are labelled as C1, C2, C3, C4.

We join the animation file (BVH) to the character in the third module. To choose the animation file we proceeded to test several files, the animation, translation, rotation, frames, etc. This test was performed using Blender software, where the BVH files were converted in FBX files.

We obtain BVH files from [14], [15]. The skeleton can be defined by a BVH file [16] (Bio Vision Hierarchy), the motion data capture format. This file consists of two sections, skeleton definition and motion. The former defines the position and orientation of the bones while the latter section defines the motion per frame of each bone. The start of the first section begins with the keyword HIERARCHY. The following line starts with the keyword ROOT followed by the name of the root segment. The hierarchy is defined by curly braces. The offset is specified by the keyword OFFSET followed by the X,Y and Z offset of the segment from its parent. Note that the order of the rotation channels appears a bit odd; it goes Z rotation, followed by the X rotation and finally the Y rotation. The second section begins with the keyword MOTION followed by a line indicating the number of frames (Frames:

keyword) and frame rate. The rest of the file contains the motion data. Each line contains one frame and the tabulated data contains data for channels defined in the first section.

The BVH file is imported from Blender, a skeleton is displayed in the 3D scene. This skeleton contains an animation such as running, walking, jumping, etc. Several BVH files were tried, and finally four of them were chosen.

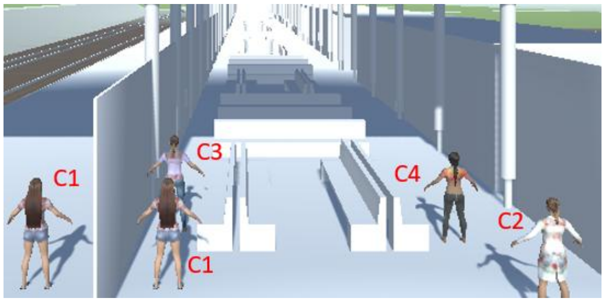


Figure 4. The four characters constructed C1, C2, C3, C4.

Walking animation requires a displacement by using translation even some animations include their own displacements. This occurs with character C3. Other walking animations rotate the character, so that we should rotate again in order to preserve the direction required. This is the case of C1 and C2, where a rotation is applied during the walking action. All the walking animations change the vertical position, so that we need to adjust it by applying a pos (position) operation.

Idling animations performs a deformation with no displacement. There are four types of idling animations. During this animation, we need to control the time in order to preserve the static behavior. We can code this in two ways, by using a timing operation or by using a counting variable. To start the idle animation, a condition must be tested with a coordinate value. In Figure 5, the two types of animations are shown.

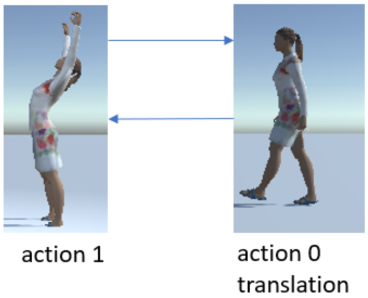


Figure 5. Idling and walking animations of character C2.

3.2.2 Geometric Transformation

Geometric transformation maps points in one space to points in another space: f(x, y, z)=(x’, y’, z’). These transformations can be very simple, such as scaling, blending, translation, rotation. We focus on transformations that can be represented with matrix operations. We can represent a 2D transformation M by a matrix M=$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$. If p is a column vector, M goes on the left p’=MP = $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$.

Here is all you get with a 2x2 transformation matrix M: x’= ax + by, y’=cx + dy.

We use the translation and the rotation operations. Humans are deformed objects so that their animation are actions such as walking, running, sitting, etc. The actions employed are shown in Table 1.

Character	Action 0	Action 1
C1	Female1_C05_WalkToRun, 13 frames	Female1_D2_Wait, 361 frames
C2	Female_B01_StandToWalk, 187 frames	Female1_A02_Sway, 1156 frames
C3	Female1_B03_Walk1, 187 frames	Female1_B03_Walk1, 187 frames
C4	Avatar_walk, 13 frames	Female_A03_Swing, 224 frames

Table 1. Animations of the characters: action 0 (walking) and action 1 (idling).

Having the paths illustrated in Figure 6, we could instantiate the characters, choosing specific start positions. The environment is divided in three sections. The path of the characters can be defined in the pre-processing stage. The camera can be handled as required to visualize specific parts of the model or setting as a first-person or a third-person view, as games usually work. On the other hand, the camera could be prepared to be manipulated by the user.

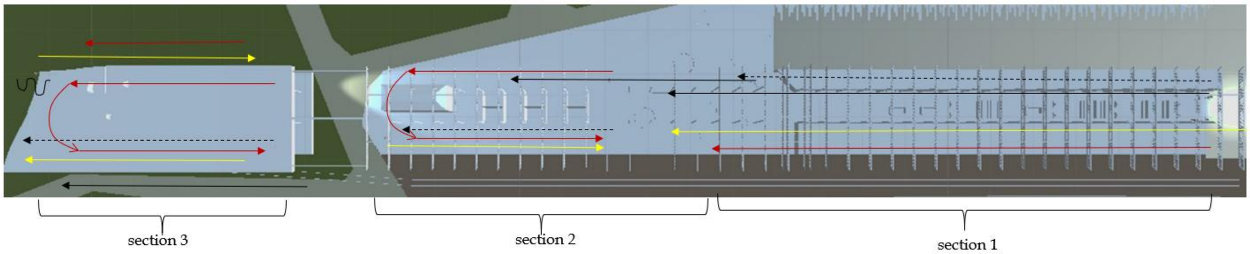


Figure 6. The Rail Station Model with the paths of the characters: ←--- C1, ← C2, ← C3, ← C4. This environment is divided in three main sections where character can be walking around.

Figure 7. The elevation perspective of the rail station is illustrated as a top view.

3.3 Game Engine (module III)

In Unity3D, the imported characters are converted into a game object to start moving. When creating a game object in unity, we can add a type of Bounding Volume (BV). The basic BVs are sphere, box, or mesh collider. Collisions make objects behave like solid objects and cannot cross each other.

The displacement of the characters is basically performed with a translation or rotation operations. However, we could add rigid body component in order to have the object influenced by gravity and external forces. The advantage of the rigid body is that collisions are detected automatically, and a repulsion force is applied to the character to maintain the movement. However, when trying this option with many characters, they collide each other and the displacement become locked, having characters in the same place, causing unreal animations. When the character is imported from Unity3D, a configuration must be performed in order to have the skeleton ready for animation.

An animation controller is required to manage the skeleton movements. The controller can be seen as an FSM (Finite State Machine). Now we can add the animation files (BVH) that were tested in Blender, these files are now FBX files and are imported into Unity3D. Animations can be adjusted by specifying the start frame, the end frame, the loop cycle over the animation, and the speed animation (fps).

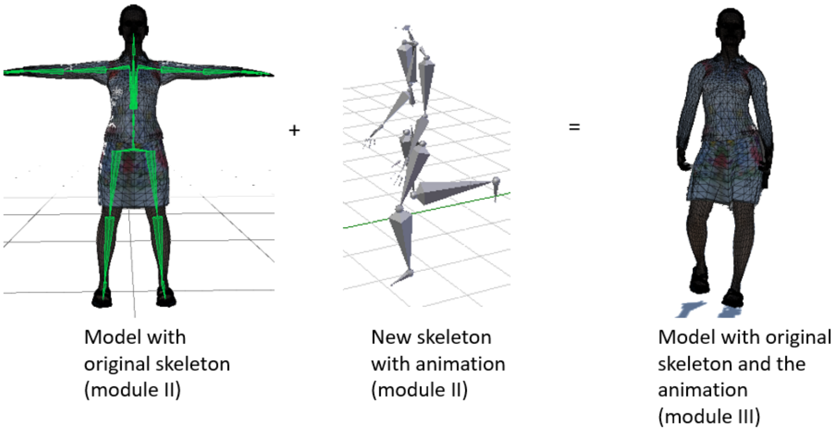

+
=
Model with original skeleton (module II)
New skeleton with animation (module II)
Model with original skeleton and the animation (module III)

Figure 8. The skeleton of character is attached to the animation bvh file to achieve an animated character.

We define the locations of several characters around the environment. The characters are called C1, C2, C3, C4 and they have several instances. The main idea is that the character walks along its correspondent path and make some stops. A path is represented by a direction in x-axis due to the ground is taken as the plane XZ, being the Y-axis the vertical axis. The character can be walked from left to right, with a vector direction <-1,0,0>, or from right to left with a vector direction <1, 0, 0>.

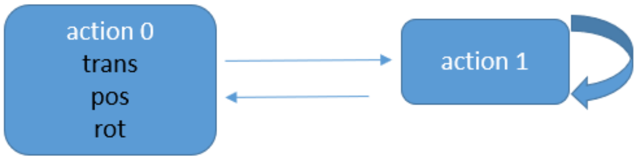
Each path includes two types of animations: action 0 and action 1, several times. The route contains a start and end positions. During this path, the character starts moving with action 0, then stops in some position and performs animation action 1. The character continues the route by applying action 0 again with motion and a stop with action 1. This means that a route contains several times action 0 and action 1. The routes are defined in Figure 6.

In the character C1, there are four paths, two with direction <-1,0,0> and the others with opposite direction <-1,0,0>. Geometric transformations are translation, position and rotation. However, a rotation must be performed in order to keep the character well balanced.

```
Move(Trans, Rot, Pos, stops[ ])

action=0
Apply Translation, Position, Rotation

if stop  or EndPos
    action=1
```



(a) The algorithm to move the character (b) a Diagram of the algorithm

Figure 9. Algorithm to move and animate characters.

The algorithm of Figure 9 starts with action 0, the character walks and displaces in the specified direction. Adjustments to keep the balance of the character are performed: position and rotation. The input of the algorithm contains data to determine if the character need a geometric adjustment. This process was computed by performing some trials of the character animation and displacement. The input also contains information about the stops required, where the character changes animation to action 1, an idle animation with no displacement. The array stops contain the coordinates where the character will be stopped. The algorithm can be represented in the diagram of Figure 9(b), and C1 calls it as Move(Trans=T,Rot=T,Pos=T, {162, 241, 223, 171}).

At the end of the path, the character remains in action 1. The FSM of the animation is depicted in Figure 10. Action 0 takes 13 frames, making a loop cycle to keep the animation continued. Action 1 contains 361 frames, taking 15 seconds with 24 fps.

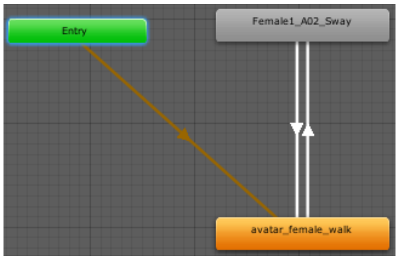


Figure 10. C1 character animations, action 0 is the walking animation in orange, and idle animation is the action 1 in grey.

Character C2 has three paths with the same direction, from right to left (←, <-1,0,0>). There are 3 or 5 stops in the three paths. Action 0 includes a translation so that the character is changing her position automatically. The algorithm is called with Move(Trans=F, Rot=F, Pos=T, {240, 220, 200, 180, 160, 140}). Path 1 has five stops, while the other two paths have three stops. At the end of the path, the character remains in action 1. Action 0 takes 187 frames, and all the frames were employed, making a loop cycle to keep the animation continue. Action 1 takes 1,156 frames.

The two paths of character C3 are coming from right to left (←), where action 0 includes the displacement, so that we only need to control the direction and rotation. There is one stop, in the end, where action 1 is performed. This animation is called with Move(Trans=F, Rot=T, Pos=T, {160}). Rotation is needed after applying action 1 due to the character takes another direction. The character remains walking in a straight direction and stays in the floor. At the end of the path, the character remains in action 1. Action 0 takes 187 frames, and all the frames were employed, making a loop cycle to keep the animation running. Action 1 takes 1,156 frames.

The other two paths include a rotation; the graph depicted in Figure 11 helped to make the script code due to it specifies in detail the positions in the plane and the rotations required. As shown in such Figure, four stops are utilized.

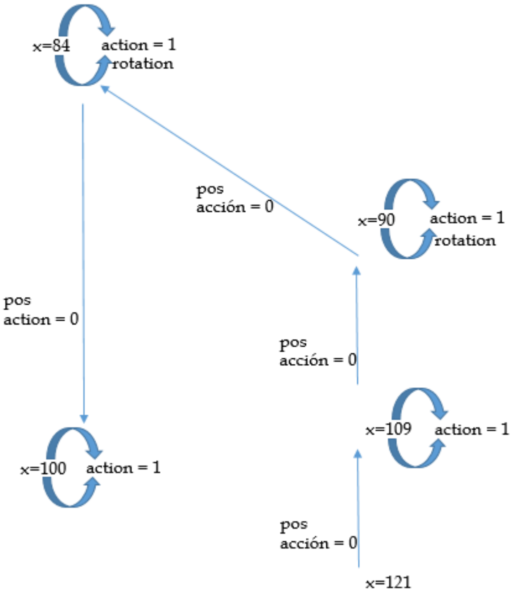


Figure 11. The path of character C3 requires rotations while idling animation action 1 is performed.

Three paths are employed in C4, which have left direction from (←, <-1, 0, 0>). Action 0 includes the displacement, so that we only need to control the position and rotation. There are four stops, where action 1 is performed, and the algorithm is called with Move(Trans=T, Rot=T, Pos=T, {210, 190, 160, 130, 100, 70}). This animation requires adjusting position and rotation during the animation of action 0 to keep a properly walk. The character remains walking in a straight-line direction and stays in the floor. At the end of the path, the character remains in action 1. Action 0 takes 13 frames, like C3, making a loop cycle to keep the animation running. Action 1 takes 244 frames. The other path serves to make the animation of the character upstairs; and when the character achieves the top floor, she starts waking.

IV. TEXTURING

This is the last process; the basic clothes or garments were chosen to each character. The texturing process can be done after creating the character. MakeHuman offers basic garments, but it also provides a URL [17] with many garments which can be download as required. Texture mapping is a method for defining high frequency detail, surface texture, or color information on a computer-generated graphic or 3D model. A texture map is an image applied to the surface of a shape or polygon. Image may be stored in common image file formats.

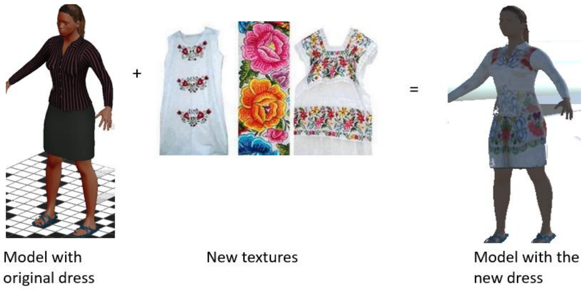


Figure 12. Images utilized for ter cloth texturing

V. CONCLUSION

In this paper, we have presented a framework to simulate people motion in a virtual environment. We detailed a method described in the diagram of Figure 1, where three modules were considered. Our method is flexible in the sense that designers can extend to other virtual environments and other characters. It should be noted that the paths can be changed, and more characters can be inserted around the 3D environment. The animations and displacements can also be changed or added.

The program enables the viewer to retain an optimal vantage point for simultaneous viewing the exhibit and learning more about it. It also enhances the viewer's understanding of the object's wider context by engaging them in a process of detailed interpretation. Unlike other approaches where the focus lies on people crowded, path planning or the use of agents, our proposal is focused on the animation and displacements issues of the characters where we can find many problems.

To test the framework, we run simulations containing several characters on a PC with Intel Core i5-4590 3.30 GHz, 8GB RAM, and Nvidia GeForce GTX 970 Graphics Card. After running the program, architecture students realize that some changes should be done

After running the program, architects realize that some changes in the design should be done:

The height of the stairs should be scaled down.

The height of the roofs should be lowered.

As further work, we can add a camera on third or (first) person view to navigate around the building, add trees, grass, sounds. Another important future effort would be to extend the framework to virtual reality and develop motion planning algorithms for transitions between the poses generated by the proposed method.

VI. REFERENCE

[1] Nisha Jain, Andrzej Wydra, Wen Hai, Nadia Magnenat Thalmann, Daniel Thalmann. Time-scaled interactive object driven multi-party VR. The Visual Computer (2018) 34: 887 – 897.

[2] T. Hoshino, M. Fujiwara, T. Hatori, A. Omachi, Y. Kondou. Optimizing Simulation of Movement in Buildings by Using People Flow Analysis Technology. Hitachi Review, 2018; vol. 67 , No 4, pp. 7117 – 7131.

[3] J. D. Lee, C. H. Hsieh, and T. Y. Lin. 2014. A Kinect-based Tai Chi exercises evaluation system for physical rehabilitation. In International Conference on Consumer Electronics. 177–178.

[4] L. Kavan, S. Collins, Jiří Žára, Carol O'Sullivan. Skinning with dual Quaternions. I3D 07 Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp 39-46. Seattle, Wshington, April 30 – May 02 2007.

[5] F. Meng and W. Zhang. Way-finding during a fire emergency: an experimental study in a vrtual environment. Ergonomics, 2014; (6) pp. 816 – 827.

[6] Paul M. Torrens. Intertwining agents and environments. Environmental Earth Sciences, 2015; vol. 74, issue 6, pp. 7117 – 7131.

[7] The SIMS game https://www.ea.com/es-es/the-sims (Accessed on June 10, 2019)

[8] Second Life Game. https://secondlife.com. (Accessed on June 10, 2019)

[9] A. Cordar; A. Wendling, C. White, S. Lampotang, B. Lok. Using mixed reality humans to influence best communication practices. 2017 IEEE Virtual Reality (VR), 18 – 22 March, Los angeles, CA, USA.

[10] User Experience of Markeless Augmented Reality Applications in Cultural Heritage Museums: "MuseumEye" as a Case Study. Ramy Hammady, Miinhua Ma, Anna Powell. Lecture Notes in Computer Science. Augmented Reality, Virtual Reality, and Computer Graphics. 5th International Conference, AVR 2018. Otranto, Italy, June 24-27, 2018. Proceedings part II.

[11] Mauricio Rosero, Rai Pogo, Edwin Pruna, Víctor H. Andaluz, and Ivón Escobar. Immersive Environment for Training on Industrial Emergencies. Intrnational Conference on Augmented Reality, Virtual Reality and Computer Graphics AVR 2018, pp 451 – 466. Otranto, italy.

[12] An UE4 plugin to develop CVE applications Leveraging Participant's Full Body Tracking Data. Carlo Luongo, Paolo Leoncini. Intrnational Conference on Augmented Reality, Virtual Reality and Computer Graphics AVR 2018, pp 451 – 466. Otranto, italy.

[13] X. Wang, Q. Ma, Q. Wang. Kinect driven 3D character animation using semantical skeleton. 2012 IEEE 2nd International Conference on Cloud Computing and Intelligent Systems. 30 Oct – 1 Nov, Hangzhou, China.

[14] Motion Capture Lab, Advanced Computing Center for the Arts and Design. (accessed june 10, 2019). https://accad.osu.edu/research/motion-lab/system-data

[15] CMU Graphics Lab Motion Capture Database. (accessed June 10, 2019). http://mocap.cs.cmu.edu/

[16] Central Source: The BVH file format. http://www.centralsource.com/blender/bvh/what.htm (accessed June 10, 2019).

[17] Clothes for MakeHuman. http://www.makehumancommunity.org/clothes.html?page=4 (accessed June 10, 2019).