Universidad Continental

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Tesis

Comparative of Techniques: Activation by Sequence, Morph Target Animation and CG/HLSL Programming in Surgery Incision Simulation for Virtual Reality

Sebastian Mauro Espinoza Zamata

Para optar el Título Profesional de
Ingeniero de Sistemas e Informática

Huancayo, 2021

Comparative of Techniques: Activation by Sequence, Morph Target Animation and CG/HLSL Programming in Surgery Incision Simulation for Virtual Reality

Sebastian Mauro Espinoza Zamata
Universidad Continental
73592845@continental.edu.pe

Pedro Yuri Marquez Solis
Universidad Continental
ymarquez@continental.edu.pe

ABSTRACT

In this decade, the way of simulating medical scenarios has evolved considerably, for some years now, this area and virtual reality came together giving life to a much more immersive form of simulation. The great challenge in medical simulation is to achieve a considerable level of realism and performance, as it is limited by the complexity of the scenario and other factors, the hardware factor being the main limiting. Giving the user the possibility to choose between greater or less realism requires that it be defined with which technique it would be achieved, therefore, this research compares three forms of surgical incision simulation for virtual reality: Activation by Sequence, Morph Target Animation and CG/HLSL Programming, evaluating factors such as: frames per-second (fps), CPU and GPU usage, which helped to obtain the level of realism of each technique; resulting in that CG/HLSL Programming uses fewer resources, with 27% CPU usage, 5% integrated GPU, 45% dedicated GPU, 60 fps and 44.3% realism, continuing, with an intermediate level use of resources Activation by Sequence with 11% CPU usage, 18% integrated GPU, 57% dedicated GPU, 60 fps, providing 46.5% realism, finally, the technique that used the most resources and obtained the highest level of realism was Morph Target Animation with 23% CPU usage, 22% integrated GPU, 77% dedicated GPU, 53 fps and 51.3% realism; these techniques can be used depending on the objective of the project where more or less realism is required, considering the use of hardware resources.

CCS CONCEPTS

• **Computing methodologies**; • **Modeling and simulation**; • **Simulation types and techniques**; • **Interactive simulation**;

KEYWORDS

Simulation, virtual reality, animation, shader

ACM Reference Format:
Sebastian Mauro Espinoza Zamata and Pedro Yuri Marquez Solis. 2021. Comparative of Techniques: Activation by Sequence, Morph Target Animation and CG/HLSL Programming in Surgery Incision Simulation for Virtual Reality. In *2020 the 6th International Conference on Communication and Information Processing (ICCIP 2020), November 27–29, 2020, Tokyo, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442555.3442569

ICCIP 2020, November 27–29, 2020, Tokyo, Japan

ACM ISBN 978-1-4503-8809-2/20/11…$15.00
https://doi.org/10.1145/3442555.3442569

1 INTRODUCTION

Surgical simulation and virtual reality are research areas that have been giving people talk in recent years, the union of these two areas has difficulties such as: providing a considerable level of realism and performance, as long as the experience is unique for those who use it for the effect of immersion. The development of this type of applications requires a lot of attention and care regarding its performance, which is directly affected by the geometric complexity of 3D objects, lights in the scene, as well as the hardware that is available and other factors, being the hardware factor one of the fundamental limitations.

Virtual reality requires meeting two requirements; real-time and realism, the first seeks to provide a pleasant experience without loss of frames per-second, the second depends on the real-time factor and hardware capacity, if you want to express the lowest possible value of real-time, you must reduce the scenario realism level and vice versa [1].

In [2] argue that the ability to locate in virtual environments to find paths more efficiently is achieved in more realistic environments, a situation that is similar to ours, since the user must make an incision at a specific point according to the surgical activity to be carried out. In [3] is concluded that in environments that simulate everyday situations, a high grade of error is admitted, but in special environments, such as a surgical operating room, the grade of freedom to make errors is the minimum possible.

This research seeks to facilitate decision-making regarding: What technique to use given a desired level of realism, considering three techniques to simulate an incision? Which are: Activation by Sequence, Morph Target Animation and CG/HLSL Programming, where frames per-second, CPU usage, GPU and its realism are compared.

2 SURGICAL INCISION SIMULATION METHODS

The methods of simulating a surgical incision over the years have taken multiple paths, Guy Sela in his research "Real-time Incision Simulation Using Discontinuous Free Form Deformation" proposes a way to simulate an incision using Discontinuous Free Form Deformation (DFFD) technique developed by himself [4], which is a variation of Free Form Deformation (FFD) proposed by the researcher Thomas W. Sederberg [5], the DFFD technique makes use

of non-continuous basic functions which are applied in the incision contour, Guy Sela combines in his research DFFD to simulate real-time artificial surgical operations and Finite-Element Model (FEM) which simulates the movement of tissue around the scalpel during the incision as a result of pre-calculated data, this bonding provides better response time and precision to incision cutting [4], another research to make cuts on three-dimensional surfaces is proposed by George Turkiyyah in "Mesh Cutting During Real-time Physical Simulations" mentions that it is very important that virtual reality applications for simulation and interaction, have different types of responses to actions such as: separating, repositioning and reforming objects depending on the purpose to which is being developed, also highlighting that these interactions add a level of realism to the response of the actions, also proposes a more realistic incision technique based on Physically Based-Simulation and Physically-Based Finite Element, where it considers that it is not only necessary altering the geometry of the 3D model as a result of the collision with another 3D object, but also said deformation must be physically real to achieve the illusion of realism [6].

In this research, three types of incision simulation methods were compared in the same scenario and object, for its development, Unity 3D Engine version 2018.2.20 was used as a simulation medium, to alter 3D models, 3ds Max for Students and Educators was used and as virtual reality devices Oculus Rift and Oculus Touch controllers.

2.1 Activation by Sequence Method

This method consists of three stages: geometry alteration, preparation and functionality. The geometry alteration was performed in 3ds Max For Students and Educators, this stage requires a base 3D model and "n" number of altered models of it and in sequence, these are geometrically altered using the 3ds Max "Slice" modifier and moving the position of the vertices generated by the same modifier in a way that gives the feeling that the cut is growing or lengthening, which will depend on the number of altered models that you have, in this method it is not important that the altered objects have the same number of polygons, triangles, vertices, edges, etc., see Fig. 1. Once the 3D models were imported into the Unity 3D Engine, the preparation and functionality stages were carried out, the preparation consists of placing the base and altered 3D models in the same position within the scenario, the altered models are deactivated and only

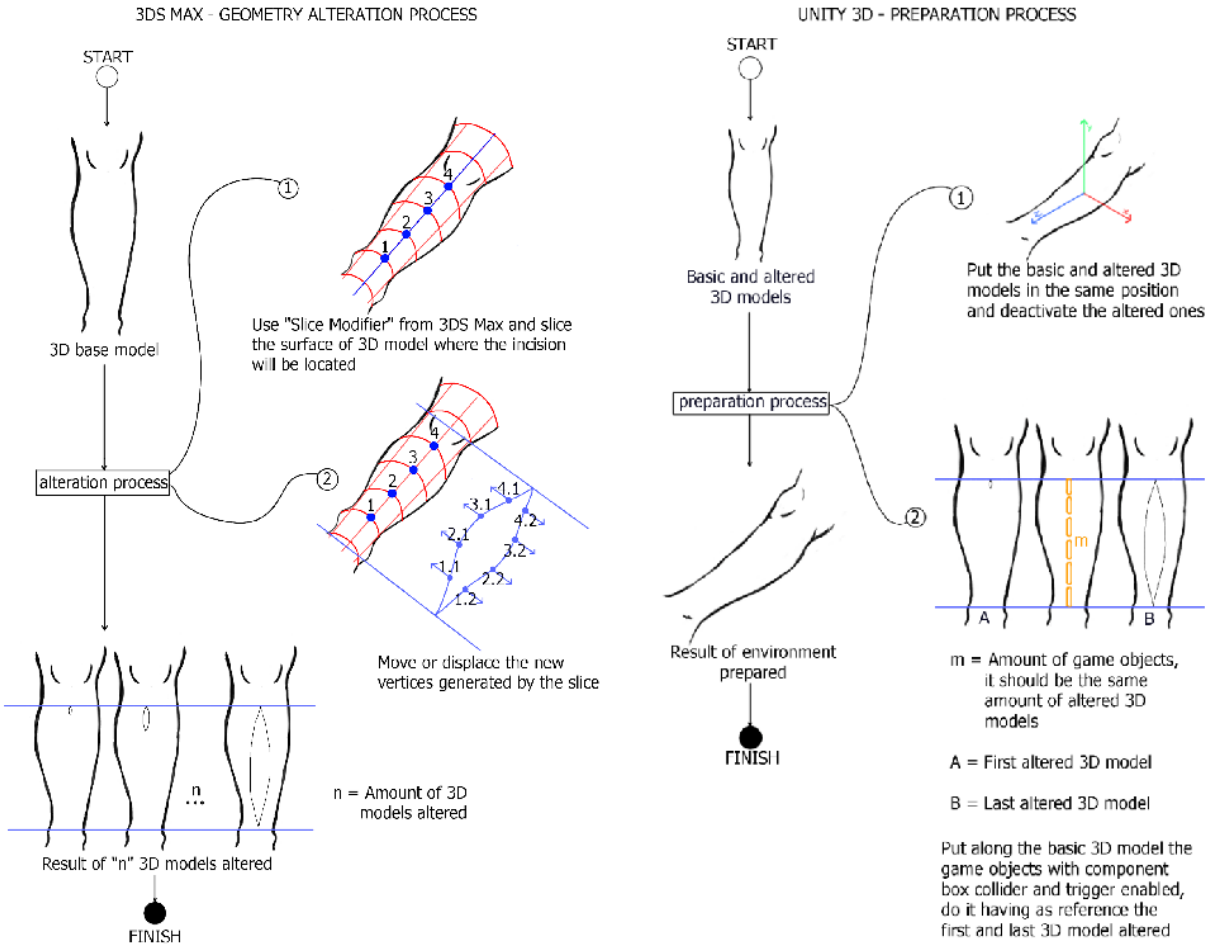

Figure 1: Geometry Alteration Process in 3ds Max and Preparation Process in Unity 3D for Activation by Sequence method.

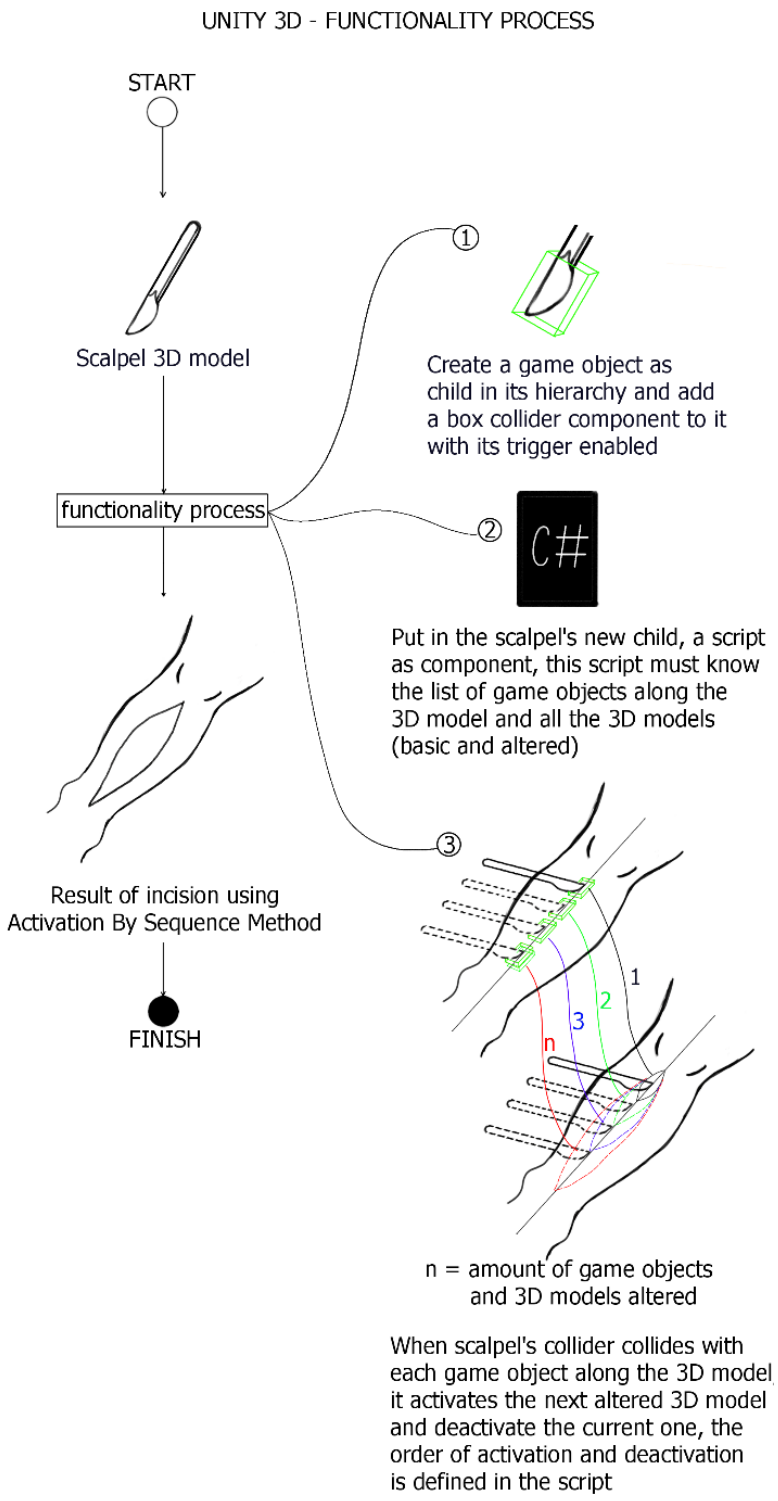

Figure 2: Functionality Process in Unity 3D for Activation by Sequence method.

left activated the base model, then empty objects or game objects are added along the cut, the amount of these must be the same as the number of altered 3D models, to locate them it is recommended to have as reference the first and last altered model, the game objects, in addition to be ordered, must also contain a box collider with trigger option enabled, see Fig. 1. For the last stage the 3D model of a scalpel is required, which will contain as its child an empty object with a box collider with trigger option enabled and a C# script which allows to activate and deactivate the 3D models in sequence, its logic is to verify if the scalpel is colliding with the object in turn; according to the data of an array, if so, then the object is deactivated, increasing the array at the same time to prepare the next collision check, thus simulating a basic-level surgical incision, see Fig. 2

2.2 Morph Target Animation Method

This method, similar to the first one, follows three stages: geometry alteration, preparation and functionality, in this method the geometric alteration has a different workflow, a base 3D model is needed and "n" number of altered models of the same and in sequence. For the Morph Target Animation method, to work the altered objects must have the same geometry, the alterations to the 3D models must give the feeling that the cut is growing or lengthening, once the altered 3D models are completed, it is added to the base 3D model the "Morpher" modifier, this allows the base 3D object to alter its mesh according to a list where the altered 3D models are located, this list modifies the mesh of the base 3D model, which is controlled with an animation that manages the amount or weight by which the mesh varies or takes the values from the "Morpher" modifier list, see Fig. 3. Once the alteration and animation are finished, the models are imported into the Unity 3D Engine where they have a preparation flow and similar functionality to the previous method, we place in a single position both the base model and the altered models leaving the base model activated and deactivating the altered, empty objects are created with a box collider component with trigger option enabled, these objects are placed along the cut, for which the first and last altered model are taken as guides, the base model must have an animator as a component; which contain an animator controller with the animation that was imported from 3ds Max, the incision animation contains within it animation events which are executed when a certain point in the animation playback is reached, see Fig. 4, the events are executed on contact of the 3D model of the scalpel with the collider of the empty

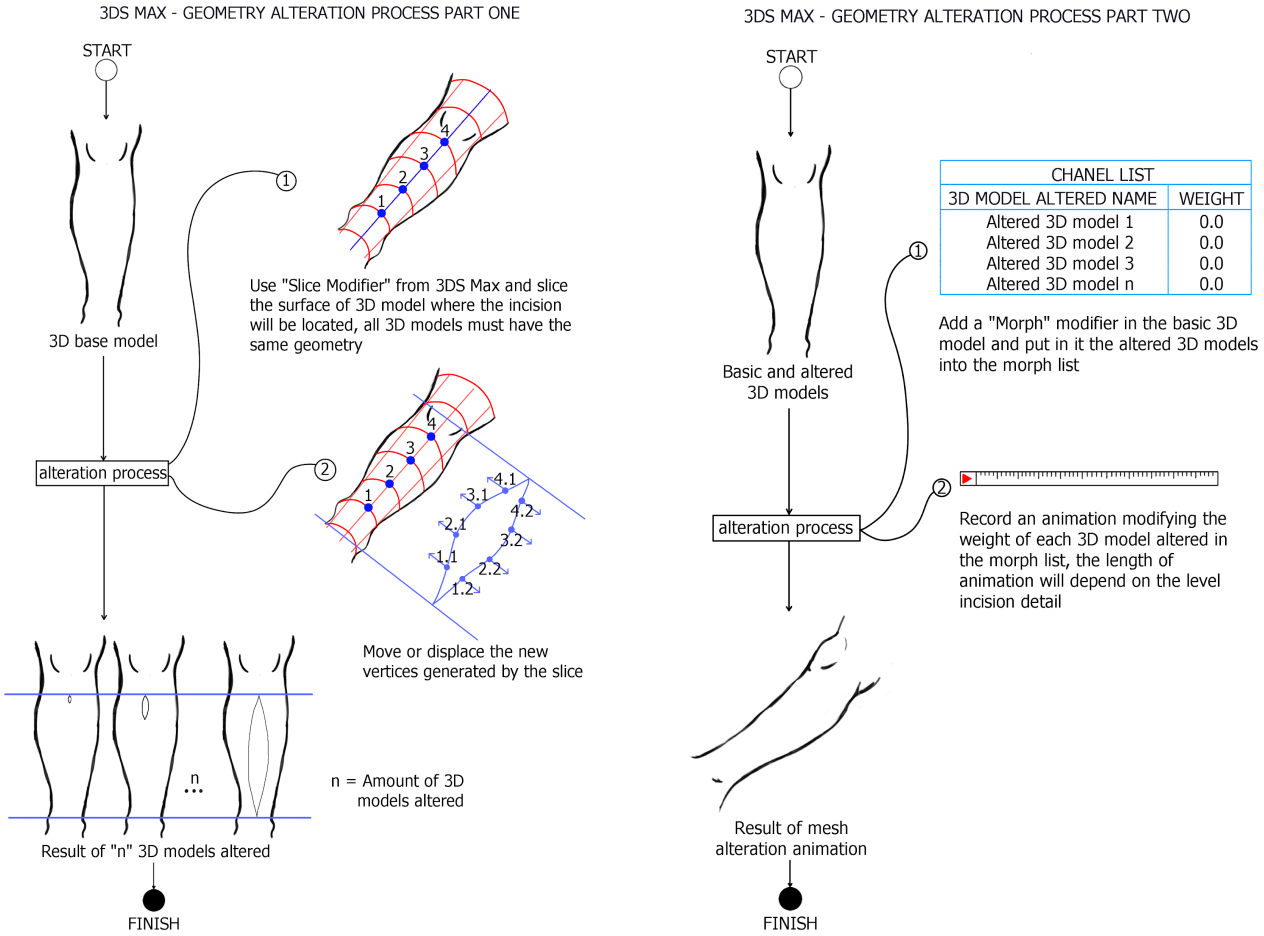

Figure 3: Geometry Alteration Process in 3ds Max for Morph Target Animation method, part one and two.

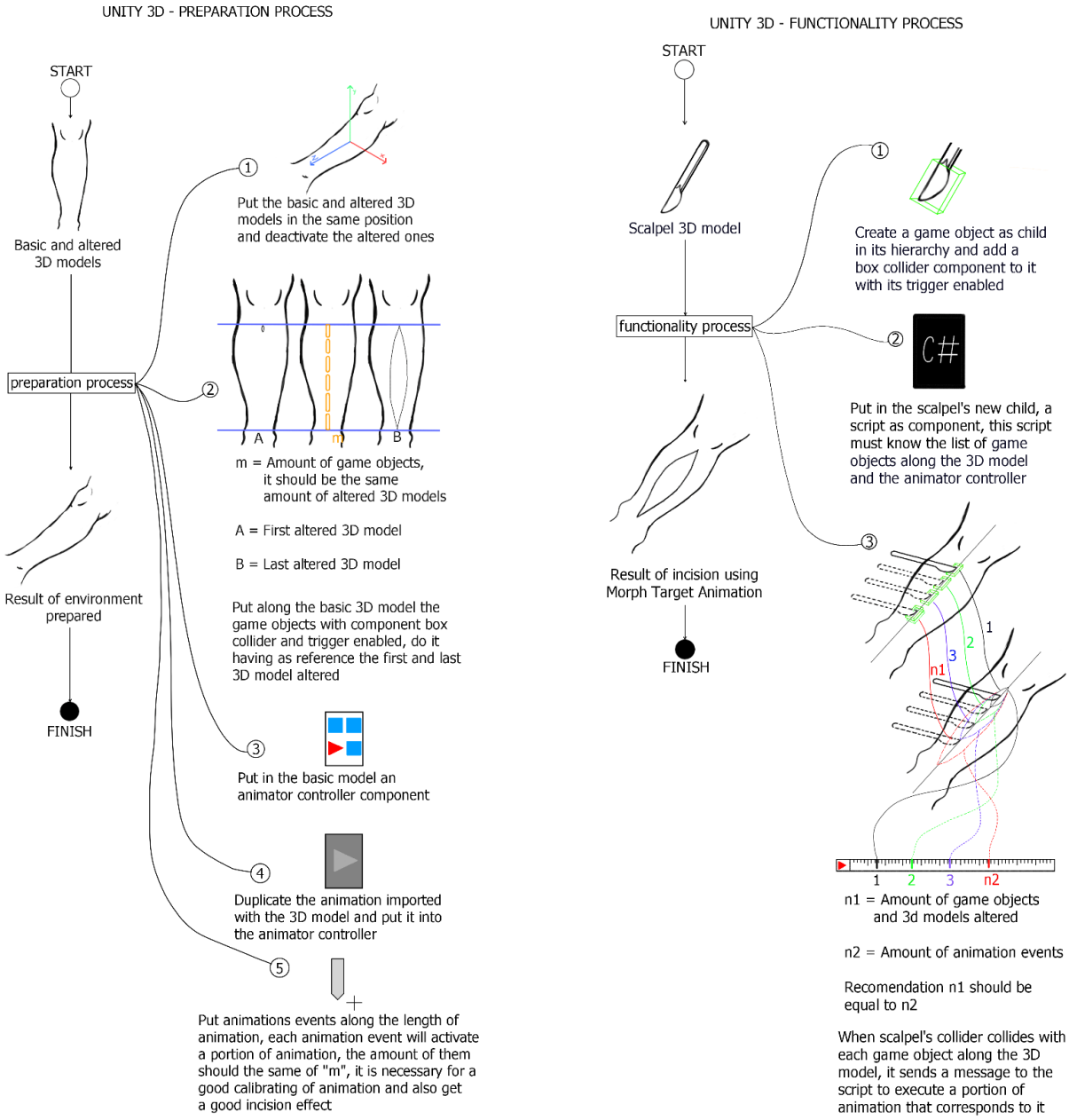

Figure 4: Preparation Process and Functionality Process in Unity 3D for Morph Target Animation method.

objects, to achieve this, the 3D model of the scalpel must have an empty object with a box collider component with trigger option enabled as a child and the C# script which controls the incision simulation, its logic consists in to obtain the animator controller of the base 3D model and execute parts of the incision animation, each part is played depending on the collision order controlled by an array which defines how it should be executed, each collision sends information to the script that controls the functionality of the incision and this executes the animation if it is being performed in the correct order, otherwise the animation will stay stopped at the point of the current animation event, see Fig. 4

2.3 CG/HLSL Programming Method

This method is based on the use of graphic computing, unlike the previous two methods it only has two stages: preparation and functionality, the preparation process consists of adding a material to the 3D model with a new shader which controls part of the functionality, in order for the flow of said functionality to be carried out correctly, an identifier or tag needs to be added to the object, as well as a mesh collider component, see Fig. 5. For the functionality, a C# script and a CG/HLSL-based shader are necessary; Unity

3D Engine's own shading programming language, this shader contains as an implemented functionality the "Tessellation" technique, which consists of adding geometry to a 3D object with a pattern, in this method we use this technique to give an incision cut feeling smoothing, to this was added the functionality so that the areas of the 3D model collided by an object or scalpel are painted with configurable transparency and gradient color, the C# script provides the information of the collision position between the scalpel and the 3D model to the shader, giving in response the reduction in height of the contact surface as well as adding a color that gives the feeling of cut skin, see Fig. 6

3 TESTS OF SURGICAL INCISION SIMULATION METHODS

The tests were done with the same scenario, 3D object and hardware so that there are no disadvantages in the comparison. As mentioned above the comparison features were frames per-second, GPU, CPU usage and level of realism.

Chris Dickinson in his book "Unity 2017 Game Optimization" mentions that the frames per-second are the result of the sum of metrics such as the use of the CPU, GPU, Physics Engine, among others, which are commonly used to perform performance analysis [7]. In [8] is sustained that normally most systems are optimized to have a speed of 60 frames per-second which is a standard, the reasonable minimum speed is 25 frames per-second for the level of human perception or hardware capacity.

The mention of [9] is very similar, indicating that systems in general share the same properties or characteristics and despite the fact that these have the corresponding optimizations, it is not easy to predict their performance, having as knowledge the characteristics of the hardware that is owned, the frame-rate is a value, which gives, a measurement of the performance of the application, it is used to make improvements in the development and testing process of the application. To measure frames per-second, Graphy Ultimate FPS Counter Stats Monitor & Debugger asset was used, free in the asset store of Unity3D itself, this resource is commonly used to measure frames per-second, memory usage, among others.

David A. Petterson mentions that the Graphics Processing Unit or GPU is an improved processor for 2D graphics, 3D, videos, etc [10]. On the other hand, [11] describes the GPU as a unit optimized for fast rendering, image and video processing.

To measure GPU and CPU usage, the task manager of the Windows 10 operating system was used.

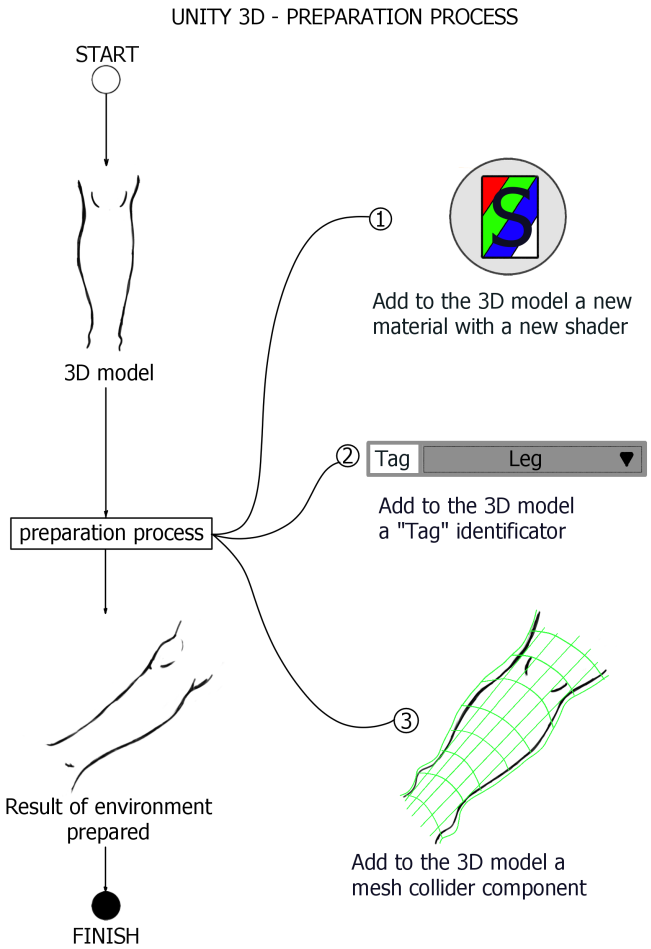

Figure 5: Preparation Process in Unity 3D for CG/HLSL Programming method.

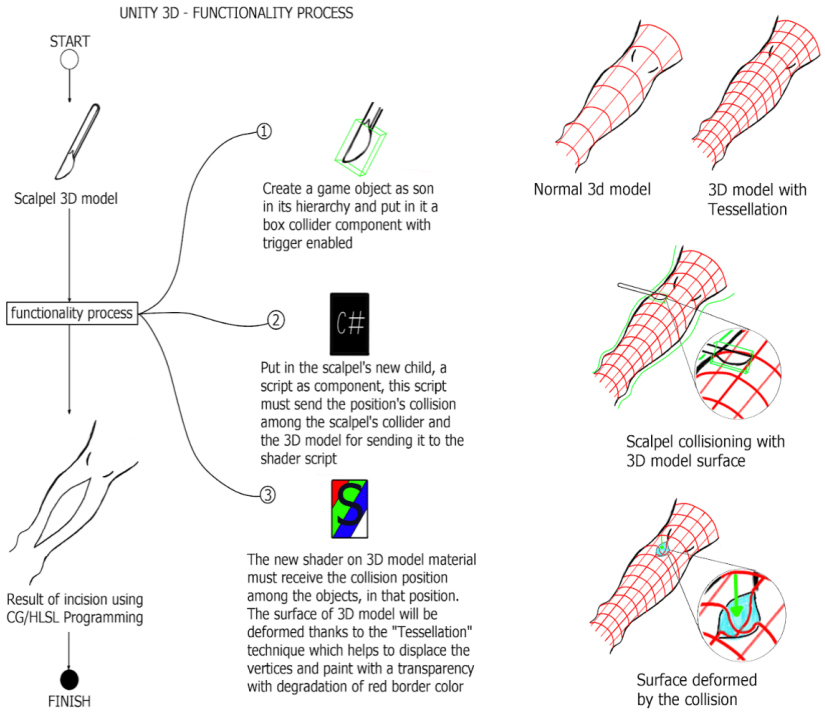

Figure 6: Functionality Process in Unity 3D for CG/HLSL Programming method.

3.1 Activation by Sequence Method Test

Table 1: Activation by Sequence method results

Variable	Value Obtained
Frames per-second	60 fps (100%)
CPU usage	11%
Integrated Intel (R) UHD Graphics 630 GPU usage	18%
Dedicated GTX 1060 TI 6GB GPU usage	57%

3.2 Morph Target Animation Method Test

Table 2: Morph Target Animation method results

Variable	Value Obtained
Frames per-second	53 fps (88.33%)
CPU usage	23%
Integrated Intel (R) UHD Graphics 630 GPU usage	22%
Dedicated GTX 1060 TI 6GB GPU usage	77%

3.3 CG/HLSL Programming Method Test

Table 3: CG/HLSL Programming method results

Variable	Value Obtained
Frames per-second	60 fps (100%)
CPU usage	27%
Integrated Intel (R) UHD Graphics 630 GPU usage	5%
Dedicated GTX 1060 TI 6GB GPU usage	45%

4 COMPARATIVE OF RESULTS

Table 4: Comparative of results of incision simulation methods

Variable	Activation by Sequence	Morph Target Animation	CG/HLSL Programming
Frames per-second	60 fps (100%)	53 fps (88.33%)	60 fps (100%)
CPU usage	11%	23%	27%
Integrated Intel (R) UHD Graphics 630 GPU usage	18%	22%	5%
Dedicated GTX 1060 TI 6GB GPU usage	57%	77%	45%

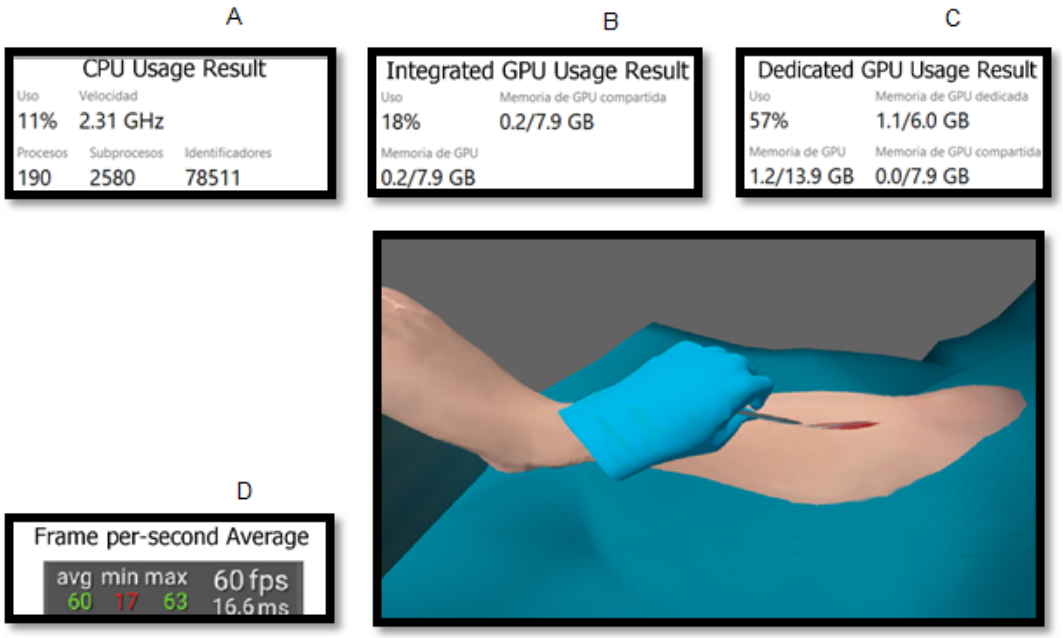

Figure 7: A) CPU usage B) Integrated GPU usage C) Dedicated GPU usage D) Frames per-seconds, and test of Activation by Sequence method.

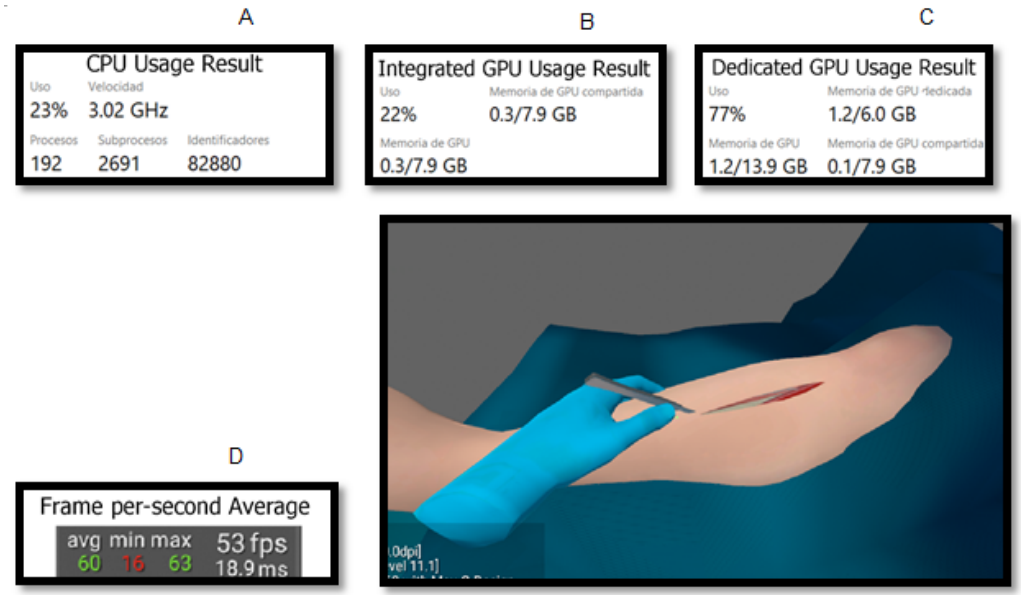

Figure 8: A) CPU usage B) Integrated GPU usage C) Dedicated GPU usage D) Frames per-second, and test of Morph Target Animation method.

According to the data obtained from each of the incision simulation methods, none resembles the other and each one has a certain advantage over the other in a certain variable.

In the use of CPU, the Activation by Sequence method is the one that makes less use of it, see Fig. 7 and Table 1, because this method only consists of activating and deactivating objects according to a Boolean instruction which is generated by a collision, compared to the other methods like Morph Target Animation, see Fig 8 and Table 2, that works by sending collision information to the C# script, then that information is passed to the animator component

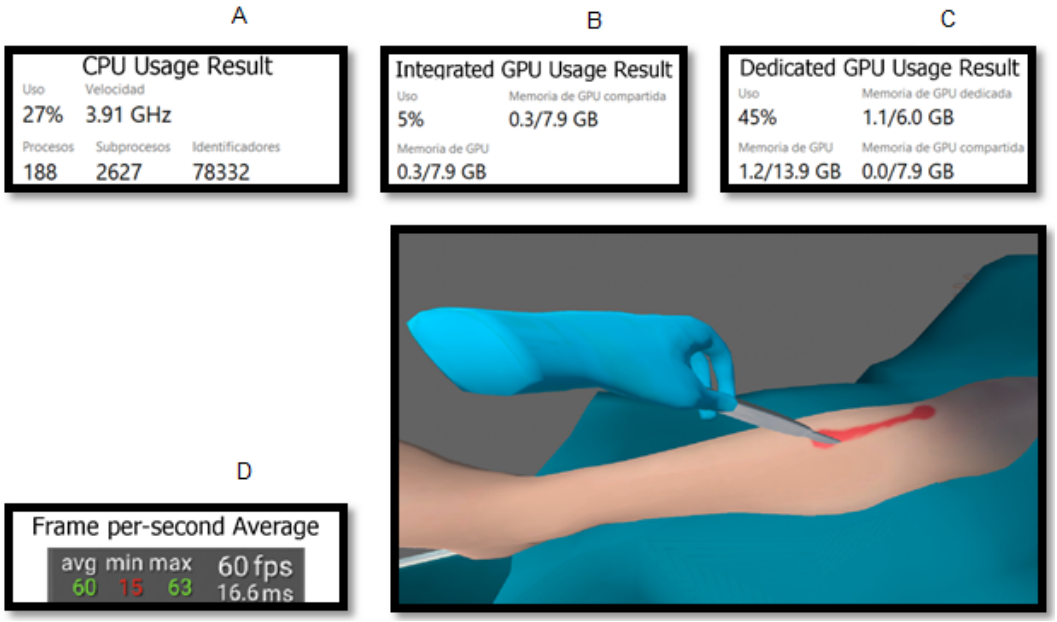

Figure 9: A) CPU usage B) Integrated GPU usage C) Dedicated GPU usage D) Frames per-second, and test of CG/HLSL Programming method.

so that it finally reaches the animation and it executes a part of itself, the CG/HLSL Programming method does high CPU usage, see Fig. 9 and Table 3, since the collision information and graphical characteristics that are given to the incision result are transferred between the C# script and the shader script, which happens in a different way with the other methods. In the case of GPU, the test was carried out on a computer with two graphics cards, for which two measurements were obtained, the first with an integrated Intel (R) UHD Graphics 630 graphic card and the second with a dedicated GTX 1060 TI 6GB graphic card; it was obtained that the method that makes efficient use of the GPU is CG/HLSL Programming, see Table 4, because it only paints the pixels of the object at the point that is collided by the 3D model of the scalpel, unlike the first method that needs to render the 3D models again every time they are activated and deactivated and the second method alters the shape of the 3D model, which is a considerable effort for the GPU. By the side of frames per-second, the Activation by Sequence and CG/HLSL Programming methods achieve 60 fps, since these were maintained between 58 fps and 60 fps, compared to the Morph Target Animation method that varied between 50 fps and 53 fps, see Table 4, because the frames per-second are related to the values generated by the CPU and GPU, this method, having a low number of frames per-second, causes the result of the actions and the projection of the images to take time in their process when being executed and projected.

It is crucial to highlight the grade of realism that each technique generates; where real-time and realism strongly influence the frames per-second and the latter in the use of GPU and CPU [1]; the result of the level of realism was obtained from the average of the variables for each method, see Table 5

Table 5: Result of the level of realism of the incision simulation methods

Variable	Activation by Sequence	Morph Target Animation	CG/HLSL Programming
Realism	46.5%	52.6%	44.3%

5 CONCLUSIONS

After comparing the aforementioned methods to simulate an incision, it is concluded that CG/HLSL Programming is the method that makes a minimum use of resources, having a number of frames per-second varying between 58 fps and 60 fps, 27% of use of CPU, 5% use of the integrated Intel (R) UHD Graphics 630 GPU, 45% use of the dedicated GTX 1060 TI 6GB GPU and a realism level of 44.3%. Therefore, the decision to opt for a technique will depend on the desired level of realism, considering that: the CG/HLSL Programming method requires less use of resources, shows lower realism and offers greater freedom to make an incision compared to Activation by Sequence and Morph Target Animation this because these two methods are based on altered 3D models, which makes the incision is predefined or has a single shape, but with a higher level of realism. However, it is suggested not to take this as a single solution, since it depends on your application and the level of realism that you want to present to the user.

The Activation by Sequence and CG/HLSL Programming methods have an adequate performance to be used in virtual reality because their amounts of frames per-second are recommended for that environment, otherwise the Morph Target Animation method

since it uses 77% of the GPU by altering the surface of the 3D model with the "Morpher" modifier, however, this technique can be improved by making use of graphics cards higher than GTX 1060 TI 6GB and optimizing the 3D models.

REFERENCES

[1] Chalmers, A., and Ferko, A. 2008. Levels of Realism: From Virtual Reality to Real Virtuality. Proceedings of the 24th Spring Conference on Computer Graphics. (April. 2008), 19-25. DOI= https://doi.org/10.1145/1921264.1921272

[2] Stachoň, Z., Kubíček, P., Málek, F., Krejčí, M., and Herman, L. 2018. The Role of Hue and Realism in Virtual Reality. 7th International Conference on Cartography and GIS. (Jun. 2018), 932-941.

[3] Van Gisbergen, M., Kovacs, M., Campos, F., Van der Heeft, M., and Vugts, V. 2019. What we don't know. The effect of Realism in Virtual Reality on Experience and Behaviour. Augmented Reality and Virtual Reality: The Power of AR and VR for Business. (Feb. 2019), 45-49.

[4] Sela, G., Schein, S., and Elber, G. 2004. Real-Time Incision Simulation Using Discontinuous Free Form Deformation. Medical Simulation: International Symposium. (Jan. 2004), 114-123. DOI= https://doi.org/10.1007/978-3-540-25968-8_13

[5] Sederberg, T. W., and Parry, S. R. 1986. Free-Form Deformation of Solid Geometric Models. ACM SIGGRAPH Computer Graphics. (Aug. 1986), 151-160. DOI= https://doi.org/10.1145/15886.15903

[6] Turkiyyah, G., Karam, W. B., Ajami, Z., and Nasri, A. 2009. Mesh Cutting During Real-time Physical Simulation. 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling. (Oct. 2009), 159-168. DOI= https://doi.org/10.1145/1629255.1629275

[7] Dickinson, C. 2017. Unity 2017 Game Optimization. Packt Publishing Ltd., Birmingham, UK.

[8] Garney, B., and Preisz, E. 2011. Video Game Optimization. Course Technology PTR, Boston, USA.

[9] Grisoni, L., Dequidt, J., and Chaillou, C. 2004. VIRTUAL REALITY APPLICATIONS AND FRAME-RATE CONTROL. 7th International Conference on DEVELOPMENT AND APPLICATIONS SYSTEM. (May. 2004).

[10] Petterson, D. D., and Hennesy, L. J. 2018. COMPUTER ORGANIZATION AND DESIGN The Hardware/Software Interface. Elsevier, Cambridge, USA.

[11] Stallings, W. 2015. COMPUTER ORGANIZATION AND ARCHITECTURE Designing for Performance. Pearson Education Ltd., USA.