



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE COMPUTACIÓN

VisualizAR

Por

Christian Oliveros 13-11000
Alexander Romero 13-11274

Realizado con la asesoría de:
Angela Di Serio

EP-4793 Mini Proyecto de Desarrollo de Software

Sartenejas, 18 de diciembre de 2018

ÍNDICE GENERAL

I. Introducción	1
1.1. Distribución	2
II. Uso	3
2.1. VisualizAR: Vectores	3
2.1.1. Image Render Target de Vuforia	3
2.1.2. Operaciones de vectores	5
2.1.2.1. Suma	5
2.1.2.2. Resta	6
2.1.2.3. Producto Punto	6
2.1.2.4. Producto Cruz	7
2.1.2.5. Limpiar	7
2.1.3. Componentes adicionales	7
2.1.3.1. Ejes del origen	8
2.1.3.2. Proyección del Vector	8
2.1.3.3. Planos en el Origen	9
2.1.4. Consideraciones Importantes al Momento de Realizar Operaciones	10
2.2. VisualizAR: Gráficas	11
2.2.1. Image Render Target de Vuforia	12
2.2.2. Manejo de Gráficas	13
2.2.3. Calculadora	14
2.2.4. Funciones	15
2.2.4.1. Seno	15
2.2.4.2. Seno 2D	15
2.2.4.3. Media del Seno	16
2.2.4.4. Multiseno	16
2.2.4.5. Multiseno 2D	17
2.2.4.6. Onda	17
2.2.4.7. Círculo	18
2.2.4.8. Remolino	18
2.2.4.9. Cilindro	19
2.2.4.10. Cilindro Tambaleante	19
2.2.4.11. Cilindro de Olla	20

2.2.4.12. Cilindro de Torsión	20
2.2.4.13. Esfera	21
2.2.4.14. Esfera Cardioide	21
2.2.4.15. Esfera Pulsante	22
2.2.4.16. Toroide Exterior	22
2.2.4.17. Toroide de Intersección	23
2.2.4.18. Toroide de Cuerno	23
2.2.4.19. Toroide Anidado	24
2.2.4.20. Toroide Estrella	24
2.2.4.21. Banda de Möbius	25
2.2.4.22. Banda Especial de Möbius	25
2.2.4.23. Botella de Klein Estática	26
2.2.4.24. Botella de Klein Dinámica	26

ÍNDICE DE FIGURAS

2.1. Ejemplo de la aplicación de Vectores.	3
2.2. Marca de Centro para controlar el origen de los vectores.	4
2.3. Marcas de Vector 1 y Vector 2 para controlar los destinos de los vectores.	5
2.4. Operación sumar en la aplicación.	6
2.5. Operación restar en la aplicación.	6
2.6. Operación producto punto en la aplicación.	7
2.7. Operación producto cruz en la aplicación.	7
2.8. Opción de ejes positivos y negativos activada en la aplicación.	8
2.9. Opción de proyección del Vector 1 activada.	9
2.10. Opción de plano en el eje x activada.	10
2.11. Opción de planos en los ejes x,y,z activada.	10
2.12. Ejemplo de la aplicación de Gráficas.	11
2.13. Ejemplo de la aplicación de Gráficas.	11
2.14. Ejemplo de la aplicación de Gráficas.	12
2.15. Marca de Botones para controlar la aplicación.	12
2.16. Marca de Gráficas para controlar la posición donde visualizar las funciones.	13
2.17. Gráfica del círculo donde se observan los cubos que forman la malla.	14
2.18. Calculadora como se ve en la aplicación.	14
2.19. Gráfica de la función Seno.	15
2.20. Gráfica de la función Seno 2D.	16
2.21. Gráfica de la función Media del Seno.	16
2.22. Gráfica de la función Mutiseno.	17
2.23. Gráfica de la función Mutiseno 2D.	17
2.24. Gráfica de la función Onda.	18
2.25. Gráfica de la función Círculo.	18
2.26. Gráfica de la función Remolino.	19
2.27. Gráfica de la función Cilindro.	19
2.28. Gráfica de la función Cilindro Tambaleante.	20
2.29. Gráfica de la función Cilindro de Olla.	20
2.30. Gráfica de la función Cilindro de Torsión.	21

2.31. Gráfica de la función Esfera.	21
2.32. Gráfica de la función Esfera Cardioide.	22
2.33. Gráfica de la función Esfera Pulsante.	22
2.34. Gráfica de la función Toroide Exterior.	23
2.35. Gráfica de la función Toroide de Intersección.	23
2.36. Gráfica de la función Toroide de Cuerno.	24
2.37. Gráfica de la función Toroide Anidado.	24
2.38. Gráfica de la función Toroide Estrella.	25
2.39. Gráfica de la función Banda de Möbius.	25
2.40. Gráfica de la función Banda Especial de Möbius.	26
2.41. Gráfica de la función Botella de Klein Estática.	26
2.42. Gráfica de la función Botella de Klein Dinámica.	27

LISTA DE ACRÓNIMOS

USB Universidad Simón Bolívar

DEP Decanato de Estudios Profesionales

PDF *Portable Document Format*, Documento en Formato Portable©

3D *Three Dimensions*, Tres Dimensiones

API *Application Programming Interface*, Interfaz de Programación de Aplicaciones

RGB *Red Green Blue*, Rojo Verde Azul

AR *Augmented Reality*, Realidad Aumentada

MB *Mega Byte*

cm Centímetro

GPU *Graphics Processing Unit*, Unidad de Procesamiento Gráfico

SDK *Software Development Kit*, Kit de Desarrollo de Software

AR/RA *Augmented Reality*, Realidad Aumentada

VR *Virtual Reality*, Realidad Virtual

CAPÍTULO I

INTRODUCCIÓN

La tecnología de AR (*Augmented Reality*) está apareciendo cada vez con más frecuencia en el mundo educativo. En una discusión entre miembros de la unidad educativa realizada en línea en julio de este año [3], la gran mayoría estuvo de acuerdo en dos puntos principales. Primero el uso de realidad aumentada es beneficioso tanto para el proceso de aprendizaje de los estudiantes, como para la facilidad de los profesores en el momento de la enseñanza. Segundo, solo una pequeña cantidad de estudiantes ha tenido acceso al uso de las tecnologías de AR y VR (*Virtual Reality*) en el ámbito educativo; este último punto también viene respaldado por un estudio realizado por la asociación EDUCAUSE [1].

Tomando en consideración estos dos primeros puntos, se planteó como objetivo principal realizar una aplicación que utiliza realidad aumentada para el área educativa. Por otra parte, en bachillerato, la enseñanza de las matemáticas son fundamentales para el desarrollo académico del estudiante. Uno de los temas principales, y que son la base para todas las matemáticas tanto en el ciclo educativo secundario como en los estudios superiores, son los vectores y las funciones.

Este proyecto, denominado VisualizAR fue el desarrollo de una aplicación con dos módulos principales:

1. Vectores: se encarga de la representación de vectores en el plano tridimensional, acentuándose en las operaciones gráficas de vectores suma, resta, producto cruz, producto punto y proyección.
2. Gráficas: se basa en el uso de ecuaciones matemáticas (principalmente con senos y cosenos) para representar varias de las funciones más representativas gráficamente.

El proyecto fue realizado utilizando el motor de videojuegos Unity 3D, que además presenta una gran facilidad y accesibilidad para realizar aplicaciones que necesiten cálculos matemáticos.

El trabajo a continuación es la documentación de las herramientas utilizadas para el desarrollo del proyecto, así como la implementación de forma detallada de cada uno de los módulos de la aplicación VisualizAR.

1.1. Distribución

Para poder distribuir el binario (*apk* para Android) y los archivos fuente, se eligió la plataforma GitHub la cual facilita la distribución de proyectos *Open Source*.

Para acceder a los archivos fuente se tiene que seguir el enlace <https://github.com/maniatic0/VisualizAR> y utilizar el manejador de versiones git.

Para acceder al binario se tiene el enlace <https://github.com/maniatic0/VisualizAR/releases>. Para instalar es necesario tener activado las opciones desarrollador. Se espera que en un futuro la aplicación pueda ser distribuida en la PlayStore de Google.

CAPÍTULO II

MANUAL DE USO

En este capítulo, se explica todo lo relacionado al uso de la aplicación.

2.1. VisualizAR: Vectores

La primera sección de la aplicación, consiste en la representación gráfica de vectores en el eje de tres dimensiones, junto con las operaciones principales de suma, resta, producto punto y producto cruz entre dos vectores. Esto permite la facilidad de aprendizaje para los estudiantes de bachillerato que están aprendiendo sobre este tema al poder observar gráficamente cómo se resuelven las operaciones de vectores. Un ejemplo de una vista de la aplicación se

Para la fácil representación de vectores, se decidió utilizar tres *Image Render Target* de Vuforia, uno denominado “Centro” que representará el origen de ambos vectores (o las coordenadas $(0, 0, 0)$), y aparte “Vector 1” y “Vector 2”, los cuales representan al destino de cada uno de los vectores correspondientes.

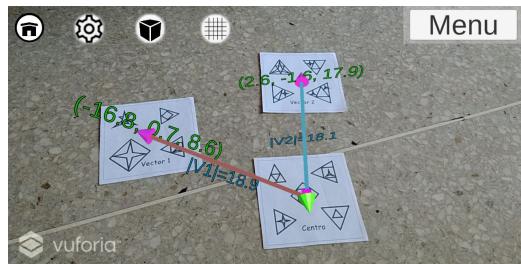


Figura 2.1: Ejemplo de la aplicación de Vectores.

2.1.1. Image Render Target de Vuforia

Para poder utilizar el motor de realidad aumentada de Vuforia, se generan tres *Image Render Target*. Estos fueron utilizados como el centro donde

comenzará cada uno de los vectores y los destinos del primer y segundo vector para controlar sus coordenadas. Para facilitar el reconocimiento por parte de Vuforia y minimizar las veces que los usuarios tengan que volver a apuntar las marcas con la cámara, se utilizaron figuras geométricas sencillas que apuntan en distintas direcciones con la finalidad de que el reconocedor de Vuforia tenga más información de la dirección de la imagen.

En las Figuras 2.2 y 2.3 se puede observar las marcas utilizadas, en estas se observa que se agregaron los nombres bajo ellas para facilitar su uso por los usuarios.

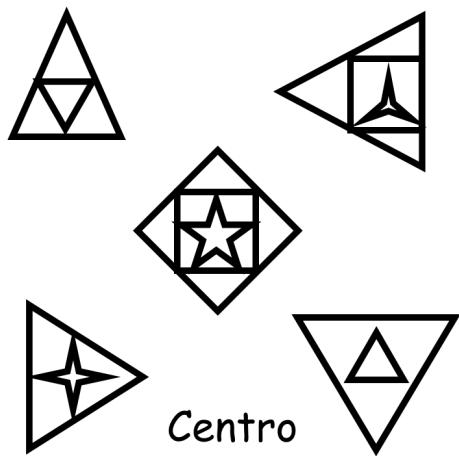


Figura 2.2: Marca de Centro para controlar el origen de los vectores.

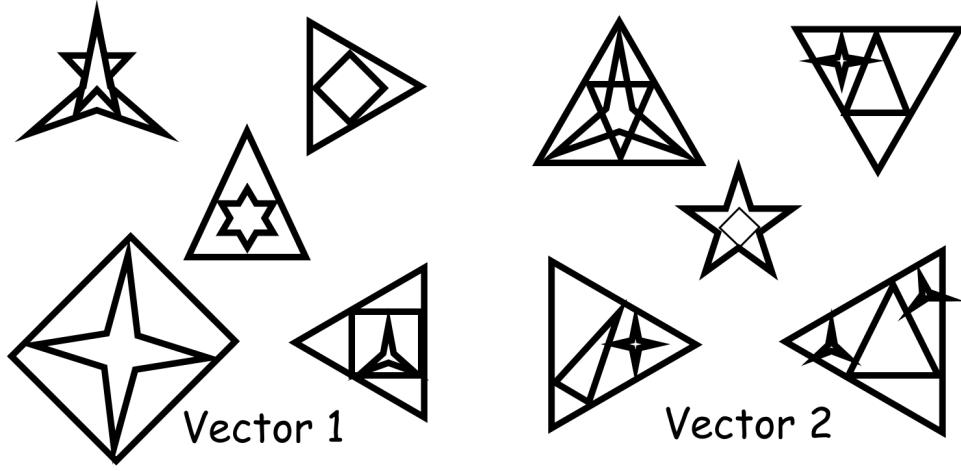


Figura 2.3: Marcas de Vector 1 y Vector 2 para controlar los destinos de los vectores.

2.1.2. Operaciones de vectores

Una vez detectada las marcas, el usuario podrá hacer uso del cálculo de las operaciones de vectores al presionar el botón “Menu”, las cuales se dividen en suma, resta, producto cruz, producto punto y limpiar. Todos los cálculos de las operaciones se realizan en el script Functions.cs el cual contiene las fórmulas necesarias para calcular cada una de ellas y mostrar gráficamente el resultado con el vector resultante (compuesto por su longitud y dirección). Por último, solo se puede mostrar una operación a la vez.

2.1.2.1. Suma

La operación de suma entre los vectores $u = (u_x, u_y, u_z)$ y $v = (v_x, v_y, v_z)$ en un plano de tres dimensiones viene representada por la fórmula $u + v = (u_x + v_x, u_y + v_y, u_z + v_z)$.

En la siguiente figura (Figura 2.4) se muestra un ejemplo de la operación sumar en funcionamiento.

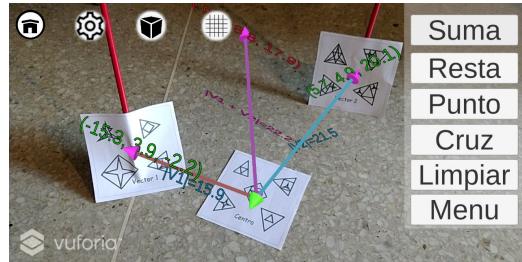


Figura 2.4: Operación sumar en la aplicación.

2.1.2.2. Resta

La operación de resta entre los vectores $u = (u_x, u_y, u_z)$ y $v = (v_x, v_y, v_z)$ en un plano de tres dimensiones viene representada por la fórmula $u - v = (u_x - v_x, u_y - v_y, u_z - v_z)$.

En la siguiente figura (Figura 2.5) se muestra un ejemplo de la operación restar en funcionamiento.

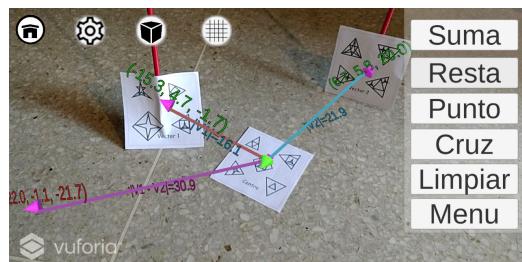


Figura 2.5: Operación restar en la aplicación.

2.1.2.3. Producto Punto

La operación de producto punto entre los vectores $u = (u_x, u_y, u_z)$ y $v = (v_x, v_y, v_z)$ en un plano de tres dimensiones viene representada por la fórmula $u \cdot v = |u||v|\cos(u, v)$, en donde $|u| = \sqrt{u_x^2 + u_y^2 + u_z^2}$.

En la siguiente figura (Figura 2.6) se muestra un ejemplo de la operación producto punto en funcionamiento.

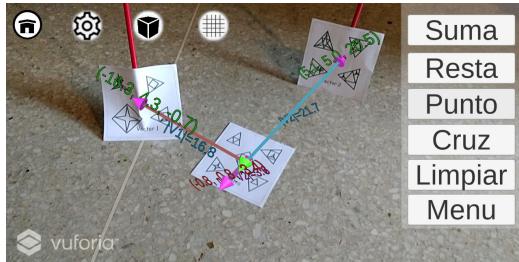


Figura 2.6: Operación producto punto en la aplicación.

2.1.2.4. Producto Cruz

La operación de producto cruz entre los vectores $u = (u_x, u_y, u_z)$ y $v = (v_x, v_y, v_z)$ en un plano de tres dimensiones viene representada por la fórmula $u \times v = (u_y * v_z - u_z * v_y, u_z * v_x - u_x * v_z, u_x * v_y - u_y * v_x)$.

En la siguiente figura (Figura 2.7) se muestra un ejemplo de la operación producto cruz en funcionamiento.

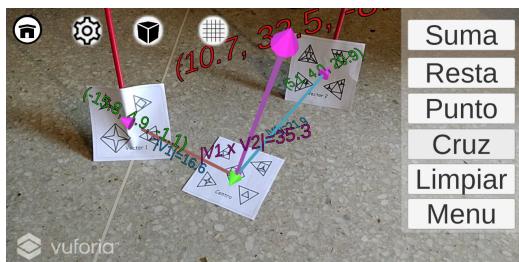


Figura 2.7: Operación producto cruz en la aplicación.

2.1.2.5. Limpiar

La operación Limpiar se encarga de desactivar la función previamente activada y mostrar únicamente la interfaz original con la representación de los dos vectores.

2.1.3. Componentes adicionales

Por último, aparte de las operaciones de vectores en la interfaz gráfica, se encuentran cuatro botones: volver al menú principal, cambiar ejes, proyección de vectores y planos en el origen. Estas opciones se pueden activar

independientes a las operaciones, es decir, no tienen restricciones entre ellas y se pueden ver todas activadas al mismo tiempo.

2.1.3.1. Ejes del origen

El botón de ejes en el origen, permite al usuario colocar distintos tipos de ejes en las coordenadas (x, y, z) para facilitar la visualización del plano cartesiano. Estas operaciones son calculadas en el *script AnimateLineAxis.cs* que se encarga únicamente de instanciar cada uno de los ejes con su dirección determinada. Las distintas opciones que se encuentran son:

1. Desactivado.
2. Ejes positivos.
3. Ejes positivos y negativos (Figura 2.8).
4. Ejes positivos extendidos.

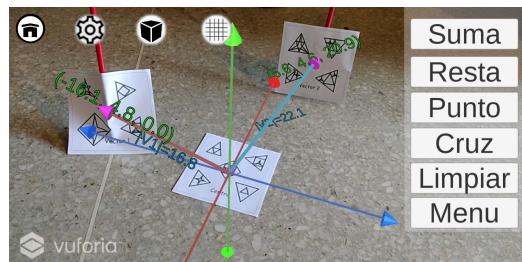


Figura 2.8: Opción de ejes positivos y negativos activada en la aplicación.

2.1.3.2. Proyección del Vector

El botón de proyección de vectores permite representar la proyección de uno de los dos vectores de acuerdo a su origen y destino. Dependiendo de cuál vector se encuentre activado, además cambia el color del botón (verde para Vector 1 y azul para Vector 2). Estas operaciones se realizan en el *script AnimateAxis.cs* asociado a cada vector, y se encarga de instanciar cada una de las líneas para finalmente proyectar el vector, representado en el plano de tres dimensiones como un cubo. Las distintas opciones que se encuentran son:

1. Desactivado.
2. Proyección del Vector 1 (Figura 2.9).
3. Proyección del Vector 2.

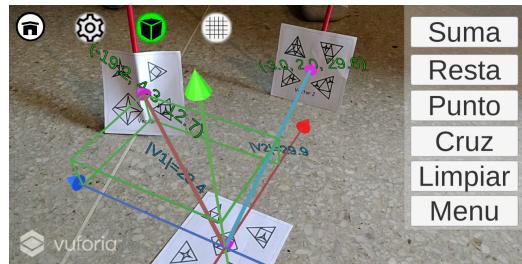


Figura 2.9: Opción de proyección del Vector 1 activada.

2.1.3.3. Planos en el Origen

El botón de planos en el origen permite al usuario instanciar en la escena un plano cartesiano que facilita las mediciones de los vectores. Cada plano tiene una medida de 30×30 cuadrículas, cada una de aproximadamente $1 \times 1\text{cm}$ reales. Estos planos se instancian en el script *AnimateLineAxis.cs*. Las distintas opciones que se encuentran disponibles son:

1. Desactivado.
2. Plano en el eje x (Figura 2.10).
3. Plano en el eje y.
4. Plano en el eje z.
5. Planos en los ejes x,y,z (Figura 2.11).

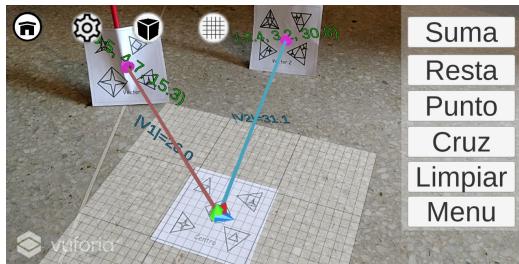


Figura 2.10: Opción de plano en el eje x activada.

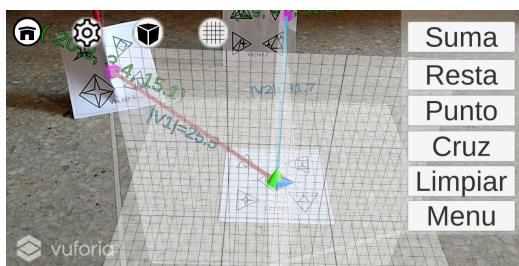


Figura 2.11: Opción de planos en los ejes x,y,z activada.

2.1.4. Consideraciones Importantes al Momento de Realizar Operaciones

En el editor de escenas de Unity, las unidades al momento de ser calculadas están definidas en Unidades de Vuforia, a diferencia de una unidad utilizada en la vida real como centímetros o pulgadas. Uno de los retos del proyecto fue encontrar cual era la relación entre unidades de Vuforia y centímetros para que los vectores tengan las mismas mediciones al ser utilizadas por la aplicación en la vida real. Luego de realizar varias pruebas, se encontró que utilizando marcas de detección de $10 \times 10\text{cm}$, se encuentra que la relación entre unidades de Vuforia y centímetros es de 1 : 1, es decir son totalmente equivalentes. Esto permite que la representación de los vectores tengan a lo sumo un margen de error de 1cm al momento de ser calculados por Vuforia y la aplicación.

2.2. VisualizAR: Gráficas

Esta sección de la aplicación se busca presentar los conceptos de gráficas a estudiantes de bachillerato, mediante la presentación interactiva de distintas funciones parametrizables con dos coordenadas, las cuales son manifestadas en 3D mediante el uso de la realidad aumentada. Además, se utiliza al tiempo como un tercer parámetro, para aumentar el llamativo de las gráficas. En las Figuras 2.12, 2.13 y 2.14 se pueden observar ejemplos de esta aplicación.

Para realizar esto se utilizaron dos *Image Render Target* de Vuforia, uno como base y otro como menú; un *Shader* para la coloración de la gráfica y cuatro *Scripts* de control.

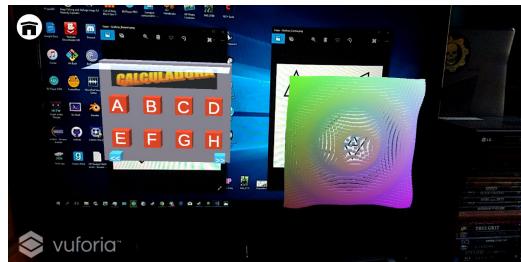


Figura 2.12: Ejemplo de la aplicación de Gráficas.

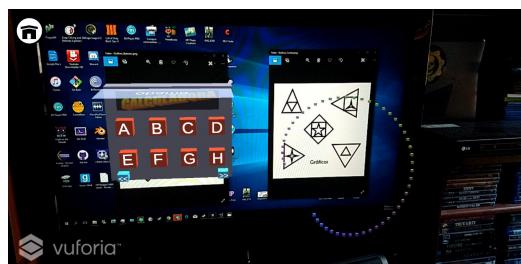


Figura 2.13: Ejemplo de la aplicación de Gráficas.

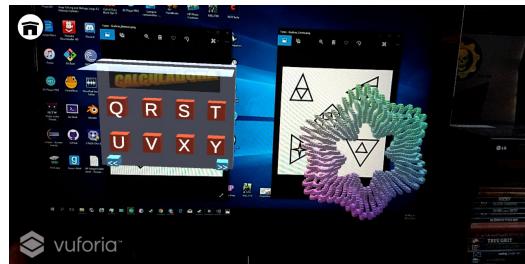


Figura 2.14: Ejemplo de la aplicación de Gráficas.

2.2.1. Image Render Target de Vuforia

Para poder utilizar el motor de realidad aumentada de Vuforia, se generaron dos *Image Render Target*. Estos fueron utilizados como la base donde poder mostrar las gráficas y los botones para controlar la selección de qué función visualizar.

En las Figuras 2.15 y 2.16 se puede observar las marcas utilizadas, en estas se observa que se agregaron los nombres bajo ellas para facilitar su uso por los usuarios.

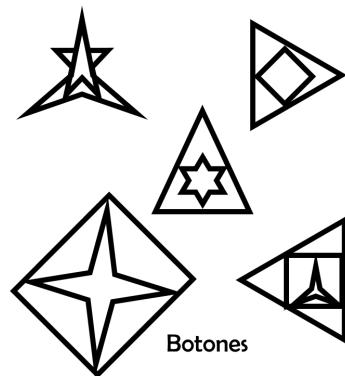


Figura 2.15: Marca de Botones para controlar la aplicación.

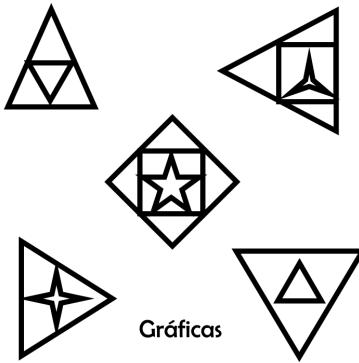


Figura 2.16: Marca de Gráficas para controlar la posición donde visualizar las funciones.

2.2.2. Manejo de Gráficas

El proceso de graficar se realiza mediante la parametrización de superficies. En el caso de la aplicación, se parametrizan superficies 3D mediante coordenadas 2D del espacio $[0, 1] \times [0, 1]$, conocido en inglés como U V Mapping, el cual consiste en proyectar un plano 2D a una superficie 3D. Además, se incluye un parámetro de tiempo para darle un aspecto dinámico a las gráficas para atraer la atención del usuario. El desarrollo del manejador de gráficas se basó en las explicaciones de Jasper Flick [2], adaptado a los requerimientos específicos de la aplicación y con más funciones.

Las funciones utilizadas en la aplicación para generar las superficies visualizadas tienen la forma presentada en la Ecuación 2.1, es importante destacar que no siempre tienen que usar los tres parámetros de entrada.

$$\begin{aligned} [0, 1] \times [0, 1] \times \mathbb{R}^+ &\rightarrow \mathbb{R}^3 \\ f(u, v, t) = (x, y, z) \end{aligned} \tag{2.1}$$

Para representar los puntos de la superficie, se utilizó una malla 2D de 100×100 cubos (para darles un tamaño visible por los usuarios), la cual se instancia al comienzo de la aplicación de gráficas y se adapta dinámicamente a la función que se desea graficar. Un ejemplo para visualizar los cubos que forman la superficie, es la gráfica del círculo de la Figura 2.17.

Todo esto es manejado por el *Script Graph.cs*, el cual, además, controla las funciones para graficar. Para facilitar desarrollo y la futura posible

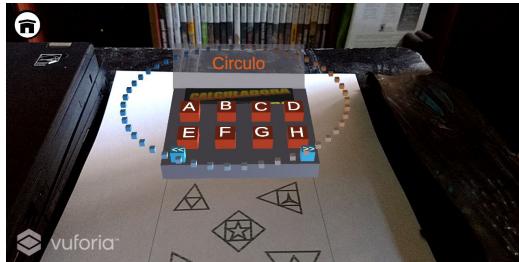


Figura 2.17: Gráfica del círculo donde se observan los cubos que forman la malla.

expansión, se agregaron: *GraphFunction.cs* el cual contiene el prototipo de las funciones que se grafican, para poder crear y utilizar más funciones; y *GraphFunctionName.cs* el cual contiene un enumerable de los nombres de las funciones, para poder visualizarlas mejor en los menús dentro del editor de Unity.

2.2.3. Calculadora

La calculadora se encarga de darle el control al usuario de decidir cuál gráfica mostrar al presionar cada uno de los botones sin necesidad de tener estas opciones en la interfaz gráfica. Por facilidad, se decidió tener tres páginas diferentes de opciones de ocho botones cada uno (identificados con una letra), las cuales pueden ser presionadas, emitiendo un sonido y animación; e inmediatamente cambiar la gráfica actual. Aparte, la calculadora muestra una pequeña pantalla transparente la cual contiene el nombre de la gráfica actual. En la Figura 2.18 se puede observar la calculadora como se ve en la aplicación.

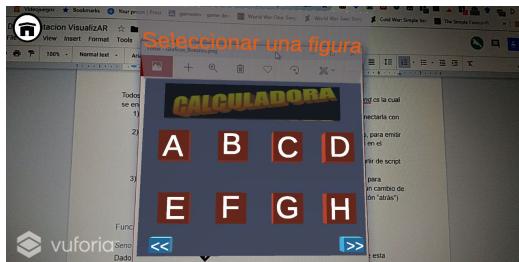


Figura 2.18: Calculadora como se ve en la aplicación.

2.2.4. Funciones

La aplicación consta de veinticuatro funciones para graficar, estas son explicadas a continuación.

Para su uso en ambientes educativos se sugiere presentar las fórmulas de forma intercalada con la presentación de las mismas en la aplicación. Un aspecto interesante es que las funciones que se presentan a continuación, es que las funciones anteriores se utilizan de una forma u otra para construir las siguientes, con lo que se pueden generar explicaciones interesantes de las conexiones de ellas.

2.2.4.1. Seno

Dado por la ecuación $f(u, v, t) = (u, \text{Sen}(\pi * (u + t)), v)$. La Figura 2.19 es la gráfica de esta función, la cual es la opción A del menú.

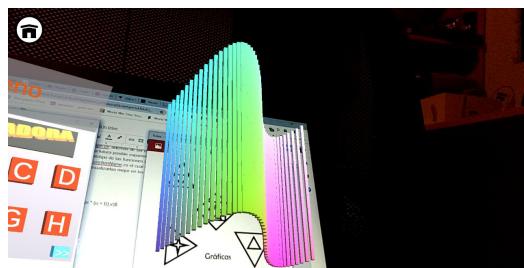


Figura 2.19: Gráfica de la función Seno.

2.2.4.2. Seno 2D

Dado por la ecuación $f(u, v, t) = (u, \text{Sen}(\pi * (u + v + t)) * 0,5, v)$. La Figura 2.20 es la gráfica de esta función, la cual es la opción B del menú.

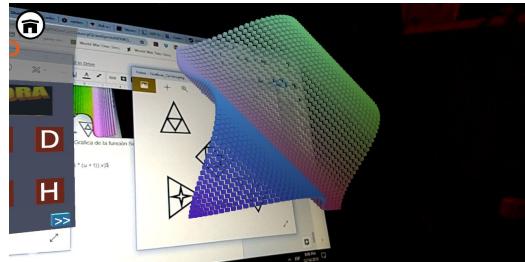


Figura 2.20: Gráfica de la función Seno 2D.

2.2.4.3. Media del Seno

Dado por la ecuación $f(u, v, t) = (u, \operatorname{Sen}(\pi * (u + t))/2 + \operatorname{Sen}(\pi * (v + t))/2, v)$. La Figura 2.21 es la gráfica de esta función, la cual es la opción C del menú.

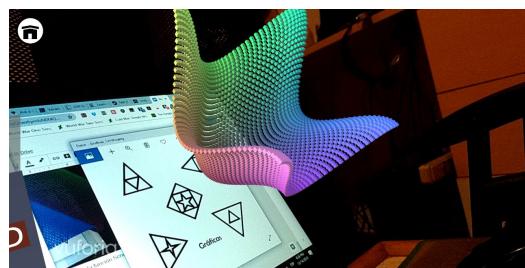


Figura 2.21: Gráfica de la función Media del Seno.

2.2.4.4. Multiseno

Dado por la ecuación $f(u, v, t) = (u, \operatorname{Sen}(\pi * (u + t)) * 2/3 + \operatorname{Sen}(2 * \pi * (v + 2 * t))/3, v)$. La Figura 2.22 es la gráfica de esta función, la cual es la opción D del menú.

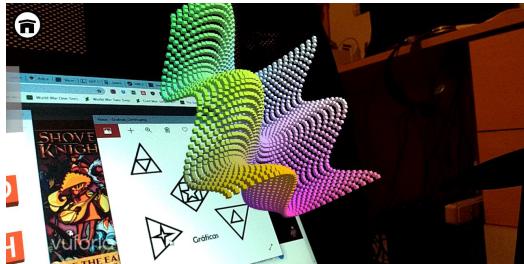


Figura 2.22: Gráfica de la función Mutiseno.

2.2.4.5. Multiseno 2D

Dado por la ecuación $f(u, v, t) = (u, (4 * \text{Sen}(\pi * (u + v + t/2)) + \text{Sen}(\pi * (u + t)) + \text{Sen}(2 * \pi * (v + 2 * t))) * 0,5) * 1/5,5, v)$. La Figura 2.23 es la gráfica de esta función, la cual es la opción E del menú.

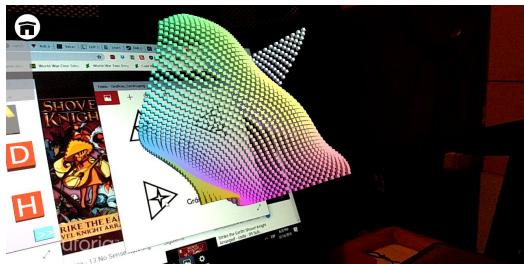


Figura 2.23: Gráfica de la función Mutiseno 2D.

2.2.4.6. Onda

Dado por la ecuación $f(u, v, t) = (u, \text{Sen}(\pi * (4 * d - t))/(1 + 10 * d), v)$ donde $d = \sqrt{u * u + v * v}$. La Figura 2.24 es la gráfica de esta función, la cual es la opción F del menú.



Figura 2.24: Gráfica de la función Onda.

2.2.4.7. Círculo

Dado por la ecuación $f(u, v, t) = (\operatorname{Sen}(\pi * u), 0, \operatorname{Cos}(\pi * u))$. La Figura 2.25 es la gráfica de esta función, la cual es la opción G del menú.

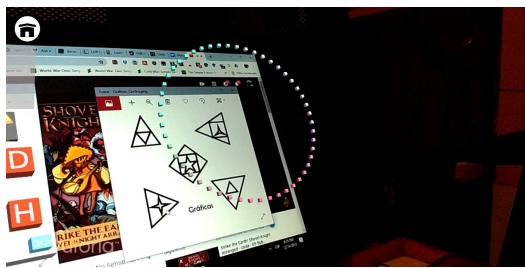


Figura 2.25: Gráfica de la función Círculo.

2.2.4.8. Remolino

Dado por la ecuación $f(u, v, t) = (\operatorname{Sen}(\pi * w), u, \operatorname{Cos}(\pi * w))$ donde $w = u + t$. La Figura 2.26 es la gráfica de esta función, la cual es la opción H del menú.

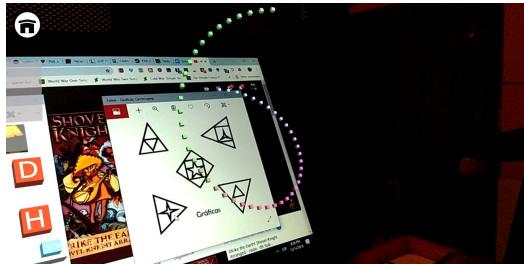


Figura 2.26: Gráfica de la función Remolino.

2.2.4.9. Cilindro

Dado por la ecuación $f(u, v, t) = (r * \text{Sen}(\pi * u), v, r * \text{Cos}(\pi * u))$ donde $r = 1$. La Figura 2.27 es la gráfica de esta función, la cual es la opción I del menú.

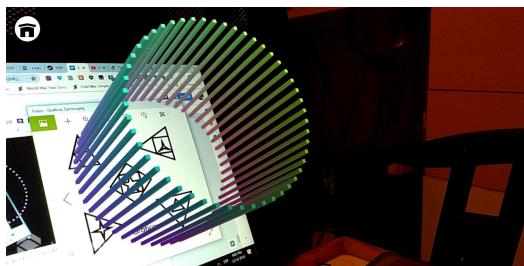


Figura 2.27: Gráfica de la función Cilindro.

2.2.4.10. Cilindro Tambaleante

Dado por la ecuación $f(u, v, t) = (r * \text{Sen}(\pi * u), v, r * \text{Cos}(\pi * u))$ donde $r = 1 + \text{Sen}(6 * \pi * u) * 0,2$. La Figura 2.28 es la gráfica de esta función, la cual es la opción J del menú.

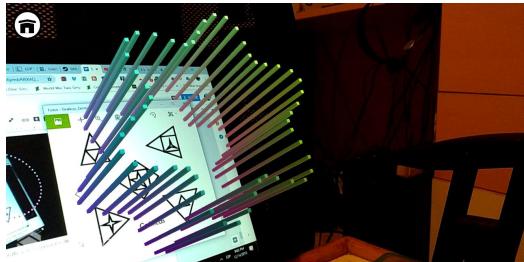


Figura 2.28: Gráfica de la función Cilindro Tambaleante.

2.2.4.11. Cilindro de Olla

Dado por la ecuación $f(u, v, t) = (r * \text{Sen}(\pi * u), v, r * \text{Cos}(\pi * u))$ donde $r = 1 + \text{Sen}(2 * \text{pi} * v) * 0,2$. La Figura 2.29 es la gráfica de esta función, la cual es la opción K del menú.

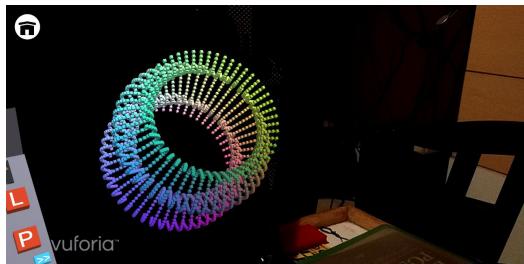


Figura 2.29: Gráfica de la función Cilindro de Olla.

2.2.4.12. Cilindro de Torsión

Dado por la ecuación $f(u, v, t) = (r * \text{Sen}(\pi * u), v, r * \text{Cos}(\pi * u))$ donde $r = 0,8 + \text{Sen}(\pi * (6 * u + 2 * v + t)) * 0,2$. La Figura 2.30 es la gráfica de esta función, la cual es la opción L del menú.

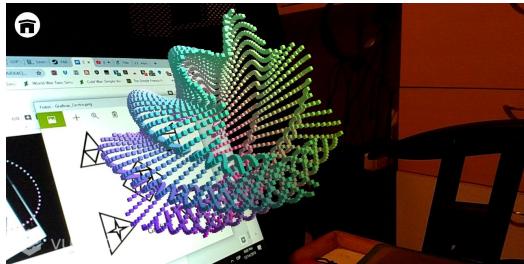


Figura 2.30: Gráfica de la función Cilindro de Torsión.

2.2.4.13. Esfera

Dado por la ecuación $f(u, v, t) = (r * \text{Sen}(\pi * u), \text{Sen}(\pi * 0,5 * v), r * \text{Cos}(\pi * u))$ donde $r = \text{Cos}(\pi * 0,5 * v)$. La Figura 2.31 es la gráfica de esta función, la cual es la opción M del menú.

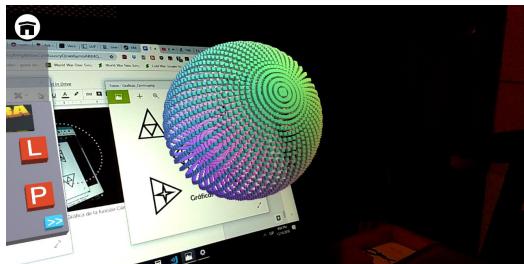


Figura 2.31: Gráfica de la función Esfera.

2.2.4.14. Esfera Cardioidé

Dado por la ecuación $f(u, v, t) = (r * \text{Sen}(\pi * u), v, r * \text{Cos}(\pi * u))$ donde $r = \text{Cos}(\pi * 0,5 * v)$. La Figura 2.32 es la gráfica de esta función, la cual es la opción N del menú.



Figura 2.32: Gráfica de la función Esfera Cardioides.

2.2.4.15. Esfera Pulsante

Dado por la ecuación $f(u, v, t) = (s * \operatorname{Sen}(\pi * u), r * \operatorname{Sen}(\pi * 0,5 * v), s * \operatorname{Cos}(\pi * u))$ donde $s = r * \operatorname{Cos}(\pi * 0,5 * v)$ y $r = 0,8 + \operatorname{Sen}(\pi * (6 * u + t)) * 0,1 + \operatorname{Sen}(\pi * (4 * v + t)) * 0,1$. La Figura 2.33 es la gráfica de esta función, la cual es la opción O del menú.

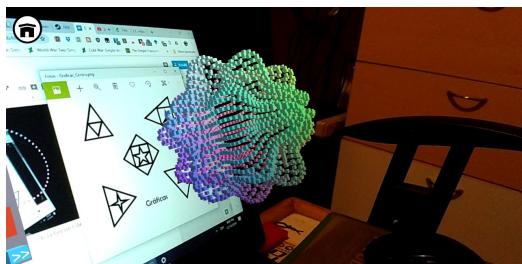


Figura 2.33: Gráfica de la función Esfera Pulsante.

2.2.4.16. Toroide Exterior

Dado por la ecuación $f(u, v, t) = (s * \operatorname{Sen}(\pi * u), \operatorname{Sen}(\pi * 0,5 * v), s * \operatorname{Cos}(\pi * u))$ donde $s = \operatorname{Cos}(\pi * 0,5 * v) + 0,5$. La Figura 2.34 es la gráfica de esta función, la cual es la opción P del menú.

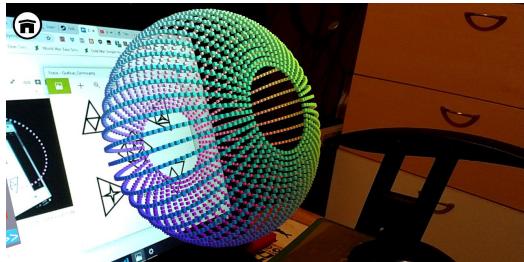


Figura 2.34: Gráfica de la función Toroide Exterior.

2.2.4.17. Toroide de Intersección

Dado por la ecuación $f(u, v, t) = (s * \text{Sen}(\pi * u), \text{Sen}(\pi * v), s * \text{Cos}(\pi * u))$ donde $s = \text{Cos}(\pi * 0,5 * v) + 0,5$. La Figura 2.35 es la gráfica de esta función, la cual es la opción Q del menú.

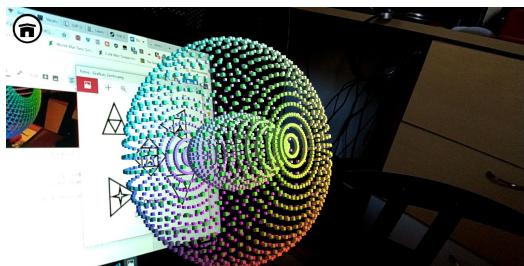


Figura 2.35: Gráfica de la función Toroide de Intersección.

2.2.4.18. Toroide de Cuerno

Dado por la ecuación $f(u, v, t) = (s * \text{Sen}(\pi * u), \text{Sen}(\pi * v), s * \text{Cos}(\pi * u))$ donde $s = \text{Cos}(\pi * v) + r_1$ y $r_1 = 1$. La Figura 2.36 es la gráfica de esta función, la cual es la opción R del menú.

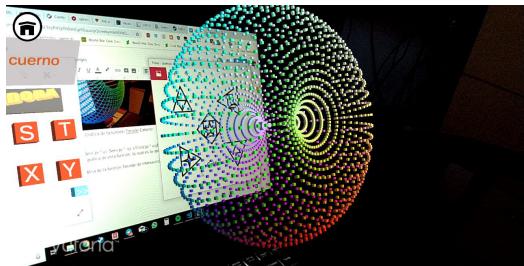


Figura 2.36: Gráfica de la función Toroide de Cuerno.

2.2.4.19. Toroide Anidado

Dado por la ecuación $f(u, v, t) = (s * \text{Sen}(\pi * u), r_2 * \text{Sen}(\pi * v), s * \text{Cos}(\pi * u))$ donde $s = \text{Cos}(\pi * v)$, $r_1 = 1$ y $r_2 = 0,5$. La Figura 2.37 es la gráfica de esta función, la cual es la opción S del menú.

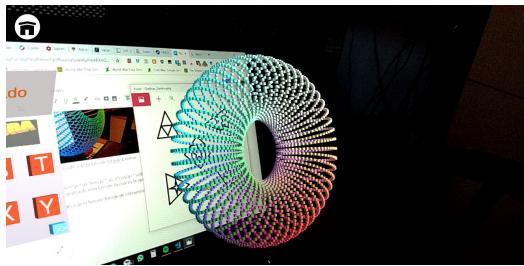


Figura 2.37: Gráfica de la función Toroide Anidado.

2.2.4.20. Toroides Estrella

Dado por la ecuación $f(u, v, t) = (s * \text{Sen}(\pi * u), r_2 * \text{Sen}(\pi * v), s * \text{Cos}(\pi * u))$ donde $s = \text{Cos}(\pi * v)$, $r_1 = 10,65 + \text{Sen}(\pi * (6 * u + t)) * 0,1$ y $r_2 = 0,2 + \text{Sen}(\pi * (4 * v + t)) * 0,05$. La Figura 2.38 es la gráfica de esta función, la cual es la opción T del menú.

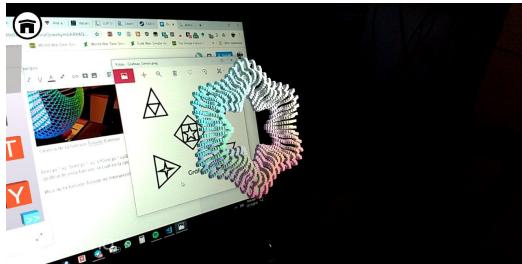


Figura 2.38: Gráfica de la función Toroide Estrella.

2.2.4.21. Banda de Möbius

Dado por la ecuación $f(u, v, t) = (te * \text{Cos}(nu), te * \text{Sen}(nu), halfnv * \text{Sen}(halfnu))$ donde $te = 1 + halfnv * \text{Cos}(halfnu)$, $nu = 2 * halfnu$, $halfnv = v - 0,5$ y $halfnu = (u + t * 0,05) * \pi$. La Figura 2.39 es la gráfica de esta función, la cual es la opción U del menú.

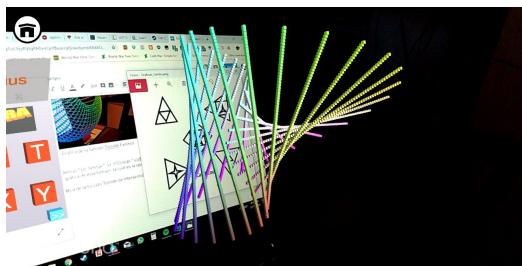


Figura 2.39: Gráfica de la función Banda de Möbius.

2.2.4.22. Banda Especial de Möbius

Dado por la ecuación $f(u, v, t) = (te * \text{Cos}(nu), te * \text{Sen}(nu), halfnv * \text{Sen}(halfnu))$ donde $te = 1 + halfnv * \text{Cos}(halfnu)$, $nu = 2 * halfnu$, $halfnv = (v - 0,5) * \text{Cos}(\pi * (t * 0,05 + v)) * \text{Cos}(\pi * (t * 0,1f + u)) * \text{Cos}(\pi * (t * 0,2 + u + v))$ y $halfnu = (u + t * 0,05) * \pi$. La Figura 2.40 es la gráfica de esta función, la cual es la opción V del menú.

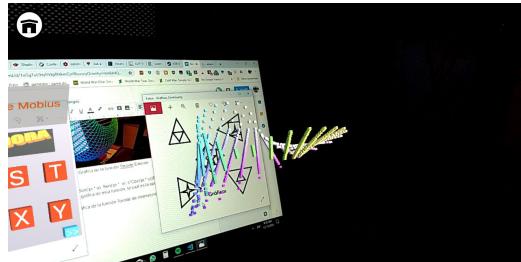


Figura 2.40: Gráfica de la función Banda Especial de Möbius.

2.2.4.23. Botella de Klein Estática

Dado por la ecuación $f(u, v, t) = (-2/15 * cu * (3 * cv - 30 * su + 90 * cu^4 * su - 60 * cu^6 * su + 5 * cu * cv * su), -1/15 * su * (3 * cv - 3 * cu^2 * cv - 48 * cu^4 * cv + 48 * cu^6 * cv - 60 * su + 5 * cu * cv * su - 5 * cu^3 * cv * su - 80 * cu^5 * cv * su + 80 * cu^7 * cv * su) - 2, 2/15 * (3 + 5 * cu * su) * sv)$ donde $sv = \text{Sen}(nv)$, $cv = \text{Cos}(nv)$, $su = \text{Sen}(nu)$, $cu = \text{Cos}(nu)$, $nv = v * \pi * 2$ y $nu = u * \pi$. La Figura 2.41 es la gráfica de esta función, la cual es la opción W del menú.

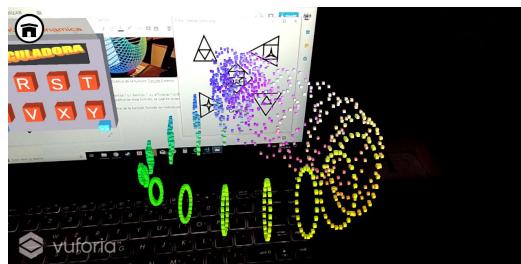


Figura 2.41: Gráfica de la función Botella de Klein Estática.

2.2.4.24. Botella de Klein Dinámica

Dado por la ecuación $f(u, v, t) = (-2/15 * cu * (3 * cv - 30 * su + 90 * cu^4 * su - 60 * cu^6 * su + 5 * cu * cv * su), -1/15 * su * (3 * cv - 3 * cu^2 * cv - 48 * cu^4 * cv + 48 * cu^6 * cv - 60 * su + 5 * cu * cv * su - 5 * cu^3 * cv * su - 80 * cu^5 * cv * su + 80 * cu^7 * cv * su) - 2, 2/15 * (3 + 5 * cu * su) * sv)$ donde $sv = \text{Sen}(nv)$, $cv = \text{Cos}(nv)$, $su = \text{Sen}(nu)$, $cu = \text{Cos}(nu)$, $nv = (v + t * 0,2) * \pi * 2$ y $nu = (u + t * 0,05) * \pi$. La Figura 2.42 es la gráfica de esta función, la cual es la opción X del menú.

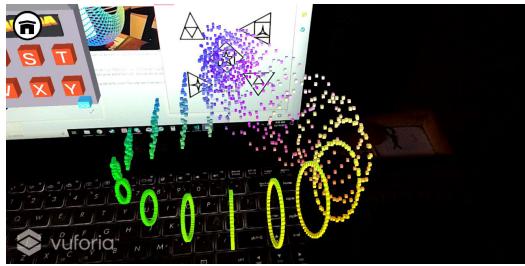


Figura 2.42: Gráfica de la función Botella de Klein Dinámica.

REFERENCIAS

- [1] Emory Craig and Maya Georgieva. From VR and AR to Our XR Future: Transforming Higher Education, Agosto 2018. Último Acceso: 2018-12-12, <https://er.educause.edu/blogs/2018/8/from-vr-and-ar-to-our-xr-future-transforming-higher-education>.
- [2] Jasper Flick. Mathematical Surfaces Sculpting with Numbers, 2017. Último Acceso: 2018-11-12, <https://catlikecoding.com/unity/tutorials/basics/mathematical-surfaces/>.
- [3] Michael Sano. Can AR/VR Improve Learning? Integrating Extended Reality Into Academic Programs #DLNchat, Julio 2018. Último Acceso: 2018-12-12, <https://www.edsurge.com/news/2018-07-19-can-ar-vr-improve-learning-integrating-extended-reality-into-academ>