

MANUAL - MATH WARS

Universidad Nacional de Colombia

Facultad de Ingeniería - Programación Orientada a Objetos

Diciembre de 2022

Jose Luis Castro Góngora
jcastrogo@unal.edu.co

Gabriel Bustos Moreno
gbustosm@unal.edu.co

Heider Hailober Navarro Quintero
hnavarroq@unal.edu.co

Pablo Hernando Suarez vargas
psuarezv@unal.edu.co

Resumen—Math wars, es la representación de 4 jóvenes dispuestos a querer contribuir en el aprendizaje de la comunidad universitaria y su país, para darle un giro a la metodología tan ambigua de la educación. Se espera poder terminarlo a largo plazo y con la comunidad ir mejorando este gran proyecto.

Index Terms—Videojuego, Cs, Unity, Java, Firebase, Json, Metodos, Clases, Manual.

I. INTRODUCCIÓN

Mathwars es un videojuego que fue creado como herramienta de aprendizaje para poder ayudar a entender las matemáticas de una forma diferente, el prototipo de la aplicación se puede encontrar en el repositorio de github dando click [Aqui](#).

La implementación de cada uno de los temas brindados por el docente Nestor Pulgarin fueron de vital importancia para el desarrollo de Math Wars, por ello de este se explica cada una de las clases por encima como bagaje del recorrido y la metodología.

II. METODOLOGÍA

Nos dividimos una parte para cada entrega del proyecto y así poder distribuir de una mejor manera la carga de trabajo entre todos los integrantes, principalmente las escenas que se explican dentro de cada espacio. Sin embargo, el orden, el saber escuchar, las críticas constructivas y trasnochos fueron vitales para llegar acuerdos y saber trabajar en equipo.

Decidimos trabajar con el repositorio de Github el cual tuvimos que aprender a manejar, inicialmente se tuvo problemas debido a que hay archivos de Unity que hay que ignorar ya que pesan demasiado y no aportan al trabajo en grupo.

Todo el proyecto se basó en estudio autónomo de aprender a manejar tanto Unity, C chart y GitHub. Los primeros 2 para el desarrollo como tal del videojuego y el tercero para poder comunicar de forma rápida y efectiva los progresos de cada integrante.

La aplicación tiene como objetivo el aprendizaje didáctico para llamar a los jóvenes y la comunidad universitaria a que se animen a aprender de una forma más divertida. El juego

funciona de tal forma que pueda registrarse y se guarde en una base de datos, en este caso es Firebase; funcionando en la escena del menú de juego.

III. MENÚ DE INICIO

Esta escena se dedica a darle la bienvenida al usuario para así poder darle la oportunidad de interactuar con ella, haciendo uso de 8 scripts para poder funcionar de forma eficiente. De manera continua, cumple con acciones :

- Jugar: Su función principal es realizar un cambio de escena al accionar el botón dirigiendo a la escena de plano para poder iniciar el juego.
- Estudiar: Su propósito es brindarle las herramientas necesarias para entender cada uno de los conceptos esenciales para entender las funciones de primer y segundo grado. También hace uso del cambio de escena para poder entrar en acción con el consejero.
- Preferencias: En este caso, se podrá modificar el brillo y el nivel de volumen según la preferencia del usuario. Implementa el uso de una otra escena para poder realizarse.
- Iniciar sesión: Implementa el uso de la base de datos con Firebase, creando con los scripts rutas de acceso para poder acceder sin inconvenientes.
- Exit: Por último, tenemos el botón de salir en caso de él usuario desee simplemente salir del juego.

IV. GRAPH

Esta clase se encarga de calcular y graficar la función en el plano

- Graficar(): Se encarga de poner cada punto de la función seleccionada por el botón en el plano, utilizando vectores de 3 dimensiones donde se modifica su valor x y se guarda en su valor y, la forma en la que se gráfica depende de que botón se oprima.

- seno(),coseno(),tangente(),: Funciones que transforman el valor x con su equivalente de la funcion matematica con la biblioteca Math

V. FIREBASE

Esta clase se encarga de manejar el intercambio de datos entre la base de datos y la interfaz gráfica de login y registro.

- SubmitRegister(): Verifica que la contraseña en la interfaz sea la misma y procede a llamar el método addData()
- addData(): Busca la colección y documento del objeto db que es la base de datos y seguidamente por medio de un diccionario agrega el registro.
- getSizeplusone(): Se encarga de contar cuantos usuarios hay registrados y asignar el siguiente numero que se usa en addData(), también se encarga de verificar que los datos de inicio de sesión sean correctos.
-

VI. PAUSA

En todo tipo de plataforma interactiva la cual cuenta con un plano principal para la interacción del usuario se utilizan algunas acciones para volver al menú principal, se implemento mediante un botón gráfico, la opción de volver a la escena menú principal, la opción para volver a la escena del juego y por ultimo la opción de modificar el sonido mediante una barra deslizante. Para esto se utilizo la clase script sonido, la cual tiene como atributo principal la modificación de sonido, los métodos utilizados fueron modificarSlide(), musicSource(), teniendo en cuenta que este script es el encargado de la comunicación entre el slide y el volumen de la musica de fondo, su valor máximo en la escala del sonido se limito a 1.

- ChangeSlider(): Este método fue utilizado para guardar los cambios generados por el usuario.
- RevisarEstoyEnMute(): este método es el encargado de generar una imagen al lado del slider cuando el valor de este sea 0, por así decirlo en estado de mute.
- Music source(): este en si se podría catalogar como una clase por si sola, en el unity es utilizado como un GameObject, este como su nombre lo indica es el encargado de ambientar el plano de juego con musica, este a diferencia de otros tiene un uso simple lo que lo hace mas fácil de conectar con los otros métodos.

En el apartado de brillo no cambiaron muchas cosas en este caso también se utilizo un slide para la fácil manipulación del usuario.

La clase brillo se utilizaron los siguientes métodos:

- El metodo slideValue() se utilizo para que al principio del juego este se situara en la mitad , teniendo en cuenta que el rango se genero de 0 a 1 este se le puso un valor de 0.5.
- Se utilizó el método start() y el método uptade() respectivamente para iniciar el slide cuando este panel se activa y para guardar su valor así este se salga de el panel.

VII. AYUDANTE

Tiene como meta principal informar al jugador y darle una introducción llamativa al mundo del calculo. Se realizó varias pruebas con el objetivo de obtener resultados eficaces. Los gifs utilizados fueron seleccionados para llamar la atención de los usuarios en cuestión. Por eso también se implementa la interacción entre player y consejero es utilizado para una distribución de temas. Finn y Newton, son personajes que suelen ser llamativos por como marcan nuestras vidas, tienen su contenido de texto e imágenes cambiantes, este apartado cuenta con su escena independiente, en la cual el usuario accede mediante el menú principal mediante el botón "Consejero".

- Start () : Este método si incluye para poder establecer las componentes como el audio y poder llamar al método OnTrigger para iniciar el dialogo al detectar cerca al player
- StartDialogue () : Declaramos un contador, una escala de tiempo para poder tipear el texto y detonamos el panel para poder activarlo y puedan reproducirse los textos.
- NextDialogueLine () : Su función principal es pasar a las siguientes lineas de textos incluyendo un condicional y una escala de tiempo para que sea adecuado el tipeo.
- SelectAudioClip() : En este caso implementa un operador ternario, es otra forma de alternar audio, es equivalente a un condicional.
- IEnumerator ShowLine() : Con esto lo que haremos es mostrar la primera linea, la haremos tipeada. Se le implementa un foreach para que el pueda reproducirse el sonido cada cierto tipo de caracteres.
- OnTriggerEnter2D(Collider2D collision) : Este método se implementa con el objetivo que detecte al sprite (conjunto de imágenes animado) que esté cerca y así poder iniciar todo el dialogo, de lo contrario no y expulsa el mensaje que no está cerca.
- OnTriggerExit2D(Collider2D collision) : Tiene como objetivo detectar que el sprite del player haya salido del espacio después de terminar con el dialogo, de lo contrario no termina el dialogo.
-

VIII. RESULTADOS OBTENIDOS

Los resultados después de varios meses de trabajo fueron obtenidos, se mejoro el plano principal del juego, su apariencia y su uso, se trabajaron muchos errores del principio, la graficación de funciones fue un reto bastante grande las posibilidades errores fueron abismales, pero se lograron corregir.

IX. ANÁLISIS DE RESULTADOS

Dentro del proceso de cada uno de las escenas del proyecto, pudimos observar qué:

- El prototipo crea una forma interesante de manejar e interactuar con funciones, haciendo que el jugador se interese por el tema desde otra perspectiva.
- Dos personas no pueden trabajar a la vez en una misma escena debido que cuando se fusionen pueden se crean

problemas de fusión y para poder resolverlas hay que mirar que cosas cambiaron en el código de como Unity está programado.

- Los diagramas o las rutas de diseño, son de vital importancia para no programar de forma inconclusa.
- Cada vez que serializa una variable pasa a ser una opción en Unity que puede ser modificada arbitrariamente.

X. BIBLIOGRAFIA

- <https://github.com/Pab10Suarez/Mathwars/releases/tag/v0.3.0-alpha>
- <https://learn.unity.com>
- <https://assetstore.unity.com>
- <https://www.academia.edu>
- <https://adamwreed93.medium.com>
- <https://www.youtube.com/watch?v=I2Dnx7MbbQ>
- <https://www.youtube.com/watch?v=o0hpnwxqe6g>
- <https://docs.unity3d.com/Packages/>
[://www.loom.com/share/3e94ccff12d4499b9708b26c68c587cc](https://www.loom.com/share/3e94ccff12d4499b9708b26c68c587cc)
- <https://www.youtube.com/watch?v=r1kEIOoMNdM>
- <https://assetstore.unity.com/>
- <https://docs.unity3d.com/es/530/Manual>
- <https://docs.unity3d.com/es/530/Manual/class-BoxCollider2D.html>
- <https://unity3d.com/unity/qa/lts-releases>