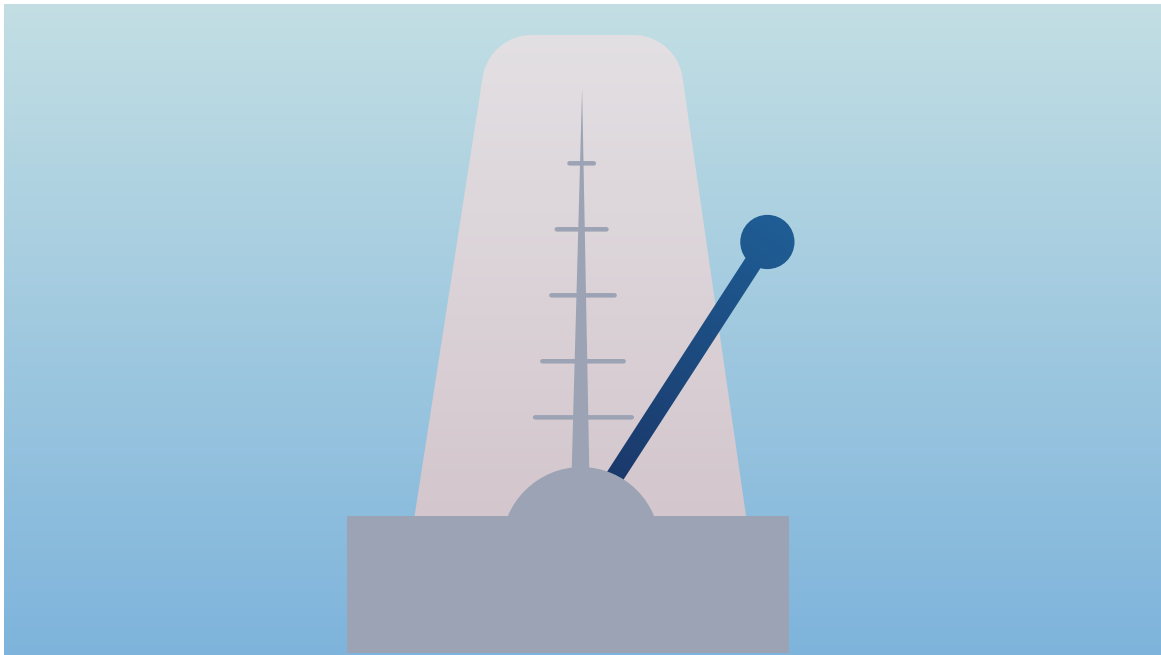




El Metrónomo de DJ

Combina sensores, matemáticas y una lógica de tiempo avanzada para construir una herramienta de DJ profesional: un metrónomo cuyo tempo controlas con la inclinación.

Cursos

- Grados 9-12

Materiales

- Celular, tableta o computador
- Conexión a Internet
- Cartón, tijeras y un par de elásticos para sujetar el celular al cartón

Descripción

Este es un proyecto de síntesis avanzado que desafía a los estudiantes a integrar múltiples conceptos de alto nivel: mapeo de sensores, aplicación de fórmulas matemáticas, gestión de estado compleja (rastrear el tiempo del último evento) y, lo más importante, la implementación de un temporizador "no bloqueante" utilizando variables y condicionales.

Objetivos Educativos

- Aplicar fórmulas matemáticas para convertir un tipo de dato en otro (BPM a milisegundos).
- Implementar una lógica de temporización no bloqueante, un pilar de la programación interactiva.
- Sintetizar los conceptos de sensores, variables, bucles y condicionales para crear una herramienta funcional.
- Entender cómo se gestiona el tiempo en aplicaciones que requieren una respuesta continua.

Inicio (10 minutos) - El Ritmo Preciso

1. Da la bienvenida a la clase: **"Hoy vamos a construir nuestra herramienta más profesional hasta la fecha. Pero para ello, necesitamos resolver un gran problema de la programación: el tiempo."**

2. Plantea el problema: "Hemos usado el bloque `esperar`, pero eso 'congela' todo el programa. Mientras espera, no puede leer sensores ni hacer nada más. ¿Cómo podría un videojuego mover a los enemigos *mientras* espera a que el jugador actúe? Usa una lógica de tiempo más inteligente."
3. Introduce la solución: **"En lugar de detenerse, los programas profesionales constantemente se preguntan: '¿Ya ha pasado suficiente tiempo para la siguiente acción?'. Hoy aplicaremos esta técnica para construir un metrónomo de DJ, una herramienta que necesita una precisión rítmica perfecta y una respuesta instantánea."**

¿Qué es un Metrónomo?

Un metrónomo es una herramienta esencial para los músicos. Produce un pulso o clic a una velocidad constante y ajustable para ayudarles a mantener un tempo estable. La velocidad se mide en **BPM (Beats Per Minute)**, o "Pulsos Por Minuto". Un tempo de 60 BPM significa exactamente un pulso cada segundo.

El Problema con 'Esperar'

Nuestro primer instinto podría ser usar un bucle con un bloque `esperar`. Sin embargo, `esperar` **bloquea** el programa. Mientras está en pausa, no puede hacer nada más: no puede leer el sensor de inclinación para ajustar el tempo, no puede actualizar la pantalla, no puede responder a botones. Para una herramienta que necesita ser fluida y responsiva en tiempo real, este método no funciona.

Lógica de Tiempo Avanzada (No Bloqueante)

Los profesionales usan una lógica que nunca "congela" el programa. Funciona como un cronómetro manual: 1. Guardamos el momento exacto en que ocurrió el último pulso en una variable (`ultimoPulso`). 2. En un bucle que se ejecuta muy, muy rápido, constantemente calculamos el tiempo que ha pasado y lo guardamos con nombre propio: `tiempoTranscurrido = milisegundos - ultimoPulso`. 3. **SI** ese `tiempoTranscurrido` es **mayor** que nuestro

`intervaloPulso` deseado — si `(tiempoTranscurrido > intervaloPulso)` —, ¡sabemos que es hora de un nuevo pulso! 4. Cuando ocurre el pulso, el paso más importante es **actualizar** `ultimoPulso` **al valor de milisegundos**, lo que efectivamente "reinicia" el cronómetro para el siguiente pulso.

Las Matemáticas del Ritmo

Para que nuestra lógica de tiempo funcione, necesitamos calcular el intervalo correcto a partir de los BPM. La fórmula es simple. Sabiendo que hay 60 segundos en un minuto y 1000 milisegundos en un segundo, hay 60.000 milisegundos en un minuto. **Intervalo (en milisegundos) = 60000 / BPM** Usaremos esta fórmula para traducir el tempo que leemos del sensor a la pausa precisa que nuestro programa necesita.

Desarrollo (20-30 minutos) - Programando el Tempo

1. Este proyecto es conceptualmente denso. Guía a los estudiantes con calma a través de la lógica.
2. Ayúdalos a **construir el metrónomo, prestando especial atención a la fórmula matemática y a los dos pasos:** establecer `tiempoTranscurrido` a `(milisegundos - ultimoPulso)` **y si** `(tiempoTranscurrido > intervaloPulso)`, como se detalla a continuación. Este es el "corazón" del programa.

Cierre (5-10 minutos) - La Máquina del Tiempo

1. Una vez que los metrónomos estén marcando el ritmo, es hora de analizar la sofisticada máquina que han construido.
2. Pregunta a la clase: **"Describan el flujo de datos y lógica. ¿Cómo se convierte un movimiento en un ritmo preciso? ¿Por qué es absolutamente crucial actualizar la variable `ultimoPulso` cada vez que suena un golpe? ¿Qué pasaría si nos olvidáramos de ese paso?"** (La respuesta: el pulso sonaría sin parar después del primer intervalo, ya que la condición siempre sería verdadera).

Reflexiona

Explica con tus palabras por qué usamos esta lógica de tiempo con variables y milisegundos en lugar de un simple bloque `esperar`.

La fórmula para el intervalo es $60000 / \text{BPM}$. Si los BPM son 60, ¿cuál es el intervalo? ¿Y si los BPM son 120? ¿El resultado matemático tiene sentido en el mundo real?

¿Qué crees que pasaría si te olvidaras de actualizar la variable `ultimoPulso` dentro del bloque `si`?