



Laboratorio – Simulación del Sistema Oscilatorio

MANUAL DEL SISTEMA

Fuentes Licero Luis Daniel
Código: 200149098

Mejía Caballero Darío José
Código: 200151122

Asignatura: Algoritmia y Programación II

Profesora: Rocío Ramos Rodríguez

Monitor: Omar Mejía Suarez

Semestre: 2020_30

INTRODUCCION

El programa se trabajó en el lenguaje de programación Java – Processing, este cuenta con más de 1400 líneas de código, uso de arreglos de datos, de funciones y subrutinas y de archivos. El programa hace uso de 2 librerías: la primera “minim” para la reproducción de audio y, la segunda, “gráfica” para el dibujo de las gráficas de posición, velocidad y tiempo.

DESARROLLO

Se comienza el programa definiendo e iniciando todas las variables que se van a usar así como los debidos arreglos de datos. Luego, en la función “setup” de Processing se cargan todas las imágenes, los audios, se crean las tablas que se usarán para las gráficas y los archivos de texto que se mostrarán en la carpeta del proyecto.

Continuamos con la función “draw” de Processing, que es la encargada de mostrar todo en pantalla.

ETAPAS

El **programa está dividido por etapas**, donde cada etapa es una pantalla que se mostrará comenzando por la etapa 0 hasta la etapa 5.

ETAPA 0: aquí se muestra la primera pantalla con el nombre de nuestro proyecto OSCILAB, y mostramos un botón para reproducir o pausar la música usando la librería “minim”.

Si presionamos el botón de “Continuar” se nos llevará a la etapa 1. Si presionamos el botón de “Salida” se deja de ejecutar el programa, y si presionamos el botón de “Instrucciones” se mostrará las instrucciones para el usuario, en estas, con los debidos botones pueden ver las diferentes instrucciones y al finalizar se le mostrará un botón que lo lleve a la Etapa 1.

ETAPA 1: aquí se le pide al usuario el sistema que desea trabajar, sea sistema no amortiguado, amortiguado u oscilaciones forzadas o resonancia., y, a través del uso de condicionales para el cambio de sistema para trabajar, al presionar “Continuar” se dirige a la Etapa 2.

ETAPA 2: aquí se le pide al usuario dependiendo del sistema que previamente eligió, los respectivos valores que debe digitar para poder calcular la posición, velocidad y aceleración. Estos menús fueron implementados con una función para optimizar el código, dicha función hace uso de las figuras geométricas de processing para mostrar cada recuadro, así como el uso de cadena de caracteres para digitar los valores. los botones de continuar y borrar son programados en la función de processing de MouseClicked y KeyPressed que se explicará más adelante, después de digitar el último valor el programa automáticamente lo llevará a la Etapa 3.

ETAPA 3: esta etapa para los casos de sistema no amortiguado y oscilaciones forzadas no hace prácticamente nada, pero, para el caso de sistema amortiguado se hace cálculo de la condición del oscilador y después lo muestra en pantalla, si pulsa en el botón de “Continuar” lo lleva a la Etapa 4.

ETAPA 4: esta etapa es la de la animación, para esta etapa es muy importante saber que la función de Processing “void draw” se ejecuta a 60 FPS, es decir, 60 imágenes por segundo. Las gráficas del oscilador se hacen teniendo en cuenta esto, donde una variable llamada “y” hace la animación mediante la opción que cuenta las imágenes de Processing para poder ajustar su tamaño horizontal y vertical, por lo que la “y” que representaría su tamaño vertical, cambia cada segundo estirando la imagen o comprimiéndola, haciendo el efecto de oscilar. Con el uso de condicionales se controla el cambio de esta, haciendo o que se comprima o se estire dependiendo si alcanza cierto valor.

Esta etapa va a durar el tiempo que el usuario haya registrado y mediante el uso de 2 contadores se puede “crear” el tiempo pues al ejecutarse a 60 FPS, un contador se suma a sí mismo y mediante un condicional, donde si el contador es igual a 60, el contador número 2 se suma a sí mismo y se resetea el contador 1, se logra controlar el tiempo de ejecución, donde si el contador 2 es igual al tiempo registrado se nos llevará a la etapa número 5.

En este mismo condicional, donde se calcula el tiempo, se hace el cálculo de los valores en cada segundo para la posición, velocidad y aceleración y mediante una función se guarda en una arreglo los valores que se usará para las gráficas y para descargar el archivo Excel, así como para la escritura del archivo de texto, el cual se hace mediante el uso de la referencia “createWriter” de Processing. Y, se muestra a tiempo real los cambios de los valores en las gráficas usando las referencias que la librería “gráfica” nos permite usar.

ETAPA 5: en esta etapa se muestran las gráficas en un tamaño más grande para poderlas ver mejor y con el uso de las herramientas que nos proporciona la librería “gráfica”, podemos mover y hacer zoom a las gráficas.

También está el botón de “Descargar”, donde al pulsarlo se guardará en la carpeta del proyecto una imagen en formato “jpg” y un archivo Excel de la debida gráfica, así como un archivo “txt”, aunque este se realiza automáticamente.

Esto se logra con una función de processing llamada “save()” donde toma una captura de la pantalla completa, por lo que para que solo salga la gráfica esta debe ponerse en tamaño completo, tomar la captura de pantalla y volver al tamaño original, aunque al ejecutarse a 60 FPS, el cambio del tamaño de la gráfica es casi imperceptible por lo que no molesta la experiencia del usuario, y la tabla que en la etapa 4 se llenó, se guarda igualmente con la referencia “saveTable()”, todo esto fue optimizado usando subrutinas, siendo esta la última etapa del proyecto.

Sub proceso “KeyPressed()”: en este subproceso de Processing se lee del teclado la tecla que se presione y, con el uso de condicionales, se restringe a solo poder usar teclas numéricas, el punto y signo negativo, estas se guardan en una variable de tipo carácter por lo que hay que posteriormente convertirla en tipo real, y, tienen un tope, mediante la longitud de la cadena, a solo poder digitar 10 caracteres, pues si hay más de eso las operaciones puede que arrojen resultados más grandes de lo que una variable de tipo float puede almacenar. No se usa variables de tipo double pues estas no se pueden digitar en las funciones de Processing de seno, coseno, raíz y exponente que se usarán para calcular posición, velocidad y aceleración.

Sub proceso “MouseClicked()”: en este subproceso de Processing se programan todos los botones que el programa utiliza mediante el uso de la herramienta de Processing, Mousex y Mousey, para saber la posición del ratón o mouse en la pantalla y saber si se encuentra en la posición del botón.

En las últimas líneas de código se pueden apreciar las funciones creadas por nosotros mismos para optimizar el programa.