

RITMSOUNDS

Danza según lo que oigas.

- Nombre: Sebastián Fernández
- Cuarto año, ingeniería civil.
- mail: sanslash332@yahoo.es
- Proyecto: ritmsounds
- Utilizado: python 2.7 64 bits
- Pygame 1.9.2 compilado en 64 bits
- IDE de programación: Visual Studio 2012, con python tools for visual studio
- repositorio: <https://github.com/sanslash332/ritmsounds>

El programa

Ritmsounds, el juego de baile, pretende ser un juego símil a los que uno puede encontrar en máquinas arcade tales como los "danse danse revolution", o "pump it up".

La diferencia, es que este proyecto, pretende utilizar tanto signos visuales como sonoros, para indicarle al jugador qué secuencias de pasos tiene que ir ejecutando, a medida que transcurre la canción.

El deber del jugador durante el juego, es intentar llevar el ritmo de la canción presionando las teclas indicadas por éste en el momento correcto. Si el jugador presiona la tecla cuando se le indica, ganará puntaje. Sin embargo, si falla perderá puntos y si falla consecutivamente el juego lo expulsará de la competencia no dejándolo acabar la canción en curso.

¿qué se tiene actualmente?

En esta primera versión del juego, se tiene correctamente implementados los modos de prueba de teclas, modo de grabación, y modo de juego simple.

Para poder ejecutarlo, será necesario ejecutar a través de consola el archivo "ritmsounds.py" que es el lanzador de todo el programa.

Al iniciar, aparecerá un menú en consola, indicando cada uno de los modos dichos previamente, además de una opción para cerrar el juego.

Modo de prueba:

En el modo de prueba, simplemente se desplegará una ventana bacia, en la cual el jugador podrá presionar las teclas A, S, Z y x, para oír los sonidos que corresponden con cada una de ellas.

Una vez se hayan reconocido cada uno de los sonidos, se presiona escape para cerrar la ventana y retornar al menú en consola.

Modo de grabación:

Si seleccionas el modo de grabación, el menú en consola te pedirá que se ingrese el nombre completo de una canción. La canción en cuestión, se tiene que haber colocado previamente en la carpeta songs.

El formato del archivo de audio puede ser cualquiera de los que acepta pygame, pero por preferencia y para asegurar la funcionalidad, ha de ser un archivo mp3 u OGG.

Una vez el programa acepte el archivo, te pedirá que presiones enter para iniciar con la grabación.

Una vez presiones enter, se desplegará la ventana en cuestión junto a que empezará a sonar la música que se escogió.

Mientras suena la música, se tiene que ir presionando las teclas a, s, z o x, para ir grabando los pasos correspondientes a la canción. ¡es recomendado tener ritmo y que lo que se presione tenga sentido con la música que se oye!

En caso de querer cancelar la grabación, simplemente presiona escape.

Una vez la música concluya, el juego automáticamente retornará al menú y el archivo de pasos se guardará instantáneamente con el mismo nombre de tu canción, sólo que agregando al nombre la extensión ".rtms"

Los archivos rtms son simplemente un documento de texto, en el que se le indican al juego, en qué tick corresponde presionar cada tecla. El juego luego utiliza estos archivos, para indicarle al jugador que es lo que debe presionar para ganar puntos en el modo de juego.

Si cancelaste la grabación utilizando escape, o cerrando la ventana, el juego no guardará lo que hayas grabado. Tienes que esperar que se cierre automáticamente.

Modo jugar:

Si seleccionas este modo, el juego te pedirá que ingreses el nombre de alguna canción. Para que ésta sea válida, tiene que ubicarse en la carpeta "songs", además de poseer un archivo rtms con exactamente el mismo nombre.

Una vez seleccionada la canción, y el juego verifique que tanto la canción como el archivo rtms existen, el menú indicará que presiones enter para iniciar el juego.

Aparecida la pantalla de juego, comenzará a sonar la música. Junto con la música, el juego emitirá los sonidos correspondientes a cada una de las teclas (a, s, z o x), las cuales se tendrán que ir presionando a penas se haya escuchado el aviso. El juego da aproximadamente unos 15 ticks de intervalo para presionar el botón correspondiente luego de que se haya oído el sonido característico de la tecla.

Una vez concluida la canción, el juego desplegará en pantalla la cantidad de aciertos obtenidos, y la cantidad de fallos cometidos. Tras eso, retornará al menú principal.

Se puede cancelar un juego, simplemente presionando escape para retornar al menú.

Cosas que faltan por implementar:

El sistema de puntos que tiene el juego es demasiado sencillo, sin considerar ni barras de vida, o combos de aciertos que den bonificaciones de puntajes.

Un aspecto más importante, es todo el entorno gráfico. El juego mecánicamente funciona bien, sin embargo las pantallas deberían incluir un menú completamente visual, indicaciones con luces y movimientos además de los sonidos para qué tecla presionar, desplegar en pantalla los puntajes del jugador en cada momento de la canción, tiempo restante de la susodicha y su nombre.

Pygame y python

Para mi suerte, el semestre anterior justo realicé el curso de sistemas operativos y redes, donde tuvimos que realizar un proyecto totalmente programado en python, por lo que el lenguaje de programación lo tenía bastante fresco en mi memoria.

Tuve un par de dudas con las sintaxis de ciertos comandos, pero nada que no se pudiese resolver con un par de consultas a google, o revisando el autocompletador del visual studio.

Pygame para mí fue una librería totalmente nueva, pero cómoda de usar. Leyendo dos tutoriales rápidos de la librería, que incluían dos buenos ejemplos sencillos, pero completos (uno era una recreación del clásico Pong) me fueron suficientes para conocer el estilo de programación que empleaba la librería. El resto, fue tan sólo revisar la documentación de pygame para encontrar módulos o clases que realizaran lo que yo buscaba.

Comencé a revisar la librería y lo que ofrecía más o menos por la segunda semana de Septiembre, pero nada intenso. La verdadera dedicación fue en la última semana de plazo que se nos dio para realizar esta entrega, donde pensé bien el problema, lo modelé (siguiendo el estilo de programación orientada a objetos, que es la que más cómoda me queda) para luego concretarlo en estos días.

Con respecto a GIT y Github, ya estaba familiarizado con lo básico de la plataforma, que tuve que utilizarla en el proyecto de ingeniería de software el año pasado. Junto con que si se utiliza con ayuda de software que proveen comandos rápidos, como realizar comit sólo haciendo clic derecho sobre la carpeta del proyecto y luego la opción correspondiente, no es mucho realmente lo que hay que aprender. Si hay algo que me puede complicar, es un tanto el tema de dividir un proyecto en diferentes branch, y luego tener que hacer merge de ellos.

las dificultades que tuve durante la creación del proyecto, fueron básicamente que no pude realizar nada gráfico, por no poder ir revisando al momento como iba quedando. Tuve que limitarme a utilizar lo que quedaba a mi alcance de la librería, junto que por no acomodarme el sistema de eventos, tuve que buscar uno que se asemejase al empleado por .net.

Otras complicaciones que surgieron durante el proyecto, no fue nada de real importancia. Más que nada lo que me toma siempre mucho tiempo, es la revisión de errores, la cual me obliga a

recurrir a la creación de un archivo de log, donde ir reportando prácticamente que cada acción del programa para poder encontrar en cuál de ellas ocurren los conflictos.

Cosas que faltan:

Personalmente lo que necesito, es bastante apoyo a la hora de poder desarrollar un software completo; en especial en todo lo que se refiere al área de diseño gráfico, y en muchas ocasiones algoritmos más completos.

Me falta mucho por aprender en lo que se refiere a la modelación del problema, y también poder estructurar, y dividir en diferentes fases el proyecto completo, como para que un equipo pudiese encargarse de él.

En esta entrega, intenté dividir lo más posible el backend del juego de su frontend, para que el despliegue de lo gráfico no se mezclase con la lógica del juego. Creo que fue un resultado satisfactorio, pero no perfecto.

Además de ello, lo más importante que me falta, es conseguir experiencia. Este sería mi primer juego que puedo decir que sí, está completo y que funciona bien. Podría pensar en los que realicé en programación avanzada, pero ninguno me quedó funcional como este.

Y ¿qué espero con esto? Aprender más sobre el tema de cómo diseñar un videojuego correctamente, poder inmiscuirme en el tema de su producción, y de paso, si es que se puede, que sea un juego que puedan jugar tanto personas que ven, como las que no.

Bibliografía:

Zona de descarga de Python, python 2.7 64 bits: <http://www.python.org/>

Python extension packages for windows, pygame 1.9.2 compilado con python 2.7 AMD 64:
<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

Python tools for visual studio: <http://pytools.codeplex.com/releases>

Tutorial de pygame: <http://pythonmania.wordpress.com/2010/04/07/tutorial-pygame-3-un-videojuego/>