TEC | Tecnológico de Costa Rica

PONG

INSTITUTO TECNOLÓGICO DE COSTA RICA

I SEMESTRE - 2018

SEGUNDA TAREA PROGRAMADA

TALLER DE PROGRAMACIÓN

ESTUDIANTES: FABIÁN RAMÍREZ Y TOMÁS SEGURA

PROFESOR: JEFF SCHMITH

DOMINGO 20 DE MAYO

PONG

TABLA DE CONTENIDOS

Contenido

INTRODUCCIÓN

Se va a desarrollar un juego que simula el tenis de mesa o ping Pong basado en el famoso juego Pong publicado en 1972, creado por Nolan Bushnell, para la popular consola Atari. Se logrará implementando múltiples funciones utilizando los conceptos de programación orientada a objetos, archivos de texto y estructuras de datos como matrices. Todo se programará en Python, específicamente con la librería Pygame, enfocada en el desarrollo de videojuegos de dos dimensiones.

Se profundizará sobre cómo se logró concretar el resultado final. Tomando en cuenta los desafíos que se presentaron y las soluciones aplicadas para su solución.

Se explicará cómo utilizar una herramienta como Git es de suma importancia cuando se debe desarrollar código en equipo, todos los beneficios que brinda cuando se explotan sus recursos de una manera óptima y los retos que presenta cuando se utiliza por primera vez.

DESCRIPCIÓN DEL PROBLEMA

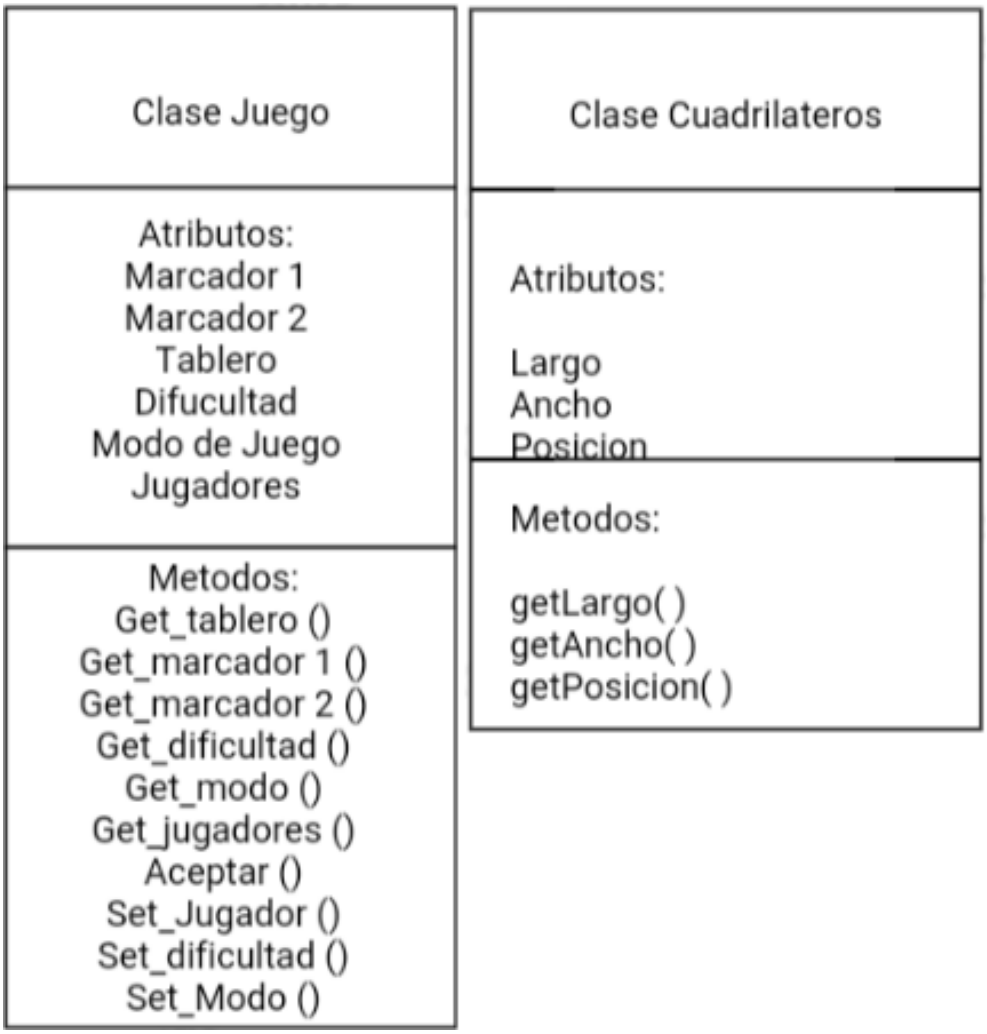
El proyecto consta de dos principales problemas. La implementación de la matriz y el manejo de clases.

La matriz debe funcionar como tablero, de manera que la bola y las paletas se muevan a través de cada posición recorriendo la pantalla. Además, los diferentes elementos de la interfaz son ubicados por las posiciones de los índices de la matriz.

Las clases deben tener los objetos con sus propios atributos y métodos con el fin de ser implementados en todo el juego y, en general, en el programa para así ahorrar cientos de líneas de código y utilizar las mismas funciones para las diferentes instancias de una misma clase. En este caso, se utiliza la instancia “Game” de la clase Juego con parámetros como los marcadores, la dificultad, los jugadores como variables de inicio y que luego son redefinidas con los métodos del juego.

GitHub resultó complicar el avance del proyecto al inicio debido a la falta de conocimiento sobre su manejo por lo que hubo que dedicar bastante tiempo a la investigación sobre el programa.

DIAGRAMA DE CLASES



ANÁLISIS DE RESULTADOS

La matriz fue definida en base a pares ordenados pero se dejó un margen de 40 píxeles a cada lado, de manera que el punto (0,0) realmente es (40,40) hasta llegar a (820,520) . Esto se construyó así por motivos estéticos, principalmente. Aunque también para colocar objetos en el borde como los marcadores, el título del juego y el botón para regresar al menú, de modo que la matriz no es la totalidad de la pantalla. Dentro de ella solamente se encuentran las paletas, la bola y los bordes superior e inferior.

Se crearon dos grandes clases con las cuales el juego funciona. Una clase para todos los cuadriláteros, es decir, la bola, paleta y bordes, donde cada elemento es una instancia de la clase con los atributos necesarios para cumplir su función, como por ejemplo sus dimensiones o el posible movimiento que podría adquirir.

Mientras tanto, la clase Juego se encarga de todo lo relacionado para que el juego funcione, como proveer todos los datos necesarios y modificarlos para que el juego se desarrolle de la mejor manera posible.

En cuanto al diseño de la aplicación, resultó ser bastante estético y funcional para el usuario. Una vez que ingresa encontrará el menú para elegir el modo de juego que desea jugar. Una vez ingresados los datos verá la pantalla del juego que se mantendrá hasta que uno de los dos jugadores alcance 10 puntos y el programa dirija al usuario a una pantalla donde elegirá si continúa jugando, si cambias los ajustes volviendo al menú o si simplemente cierra la aplicación.

El link para el repositorio de GitHub es el siguiente:

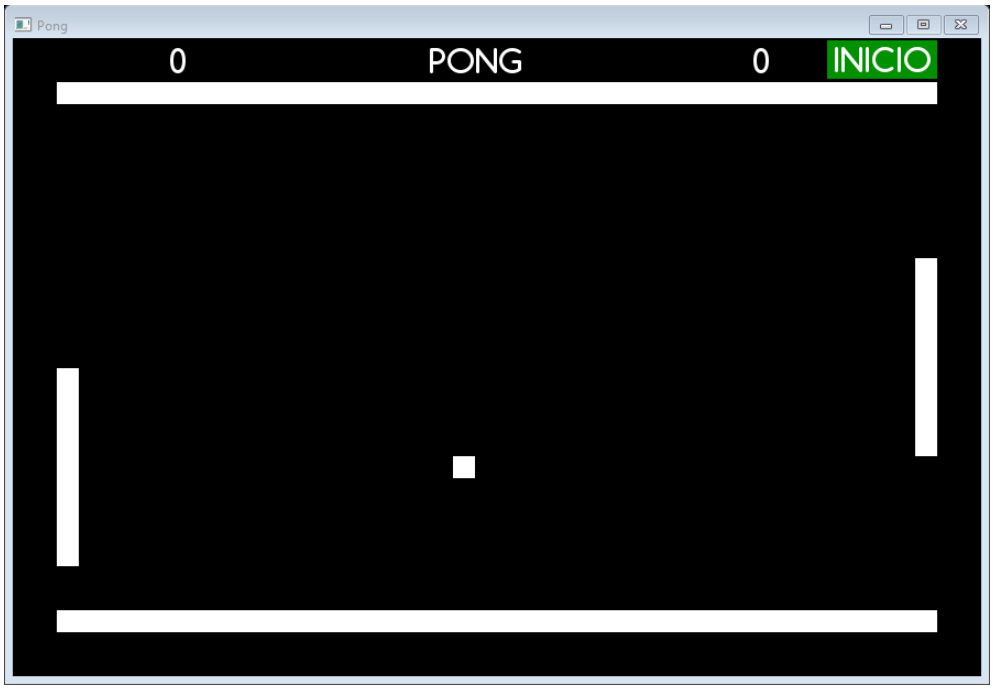
https://github.com/FabianRamirez03/Pong-2.0

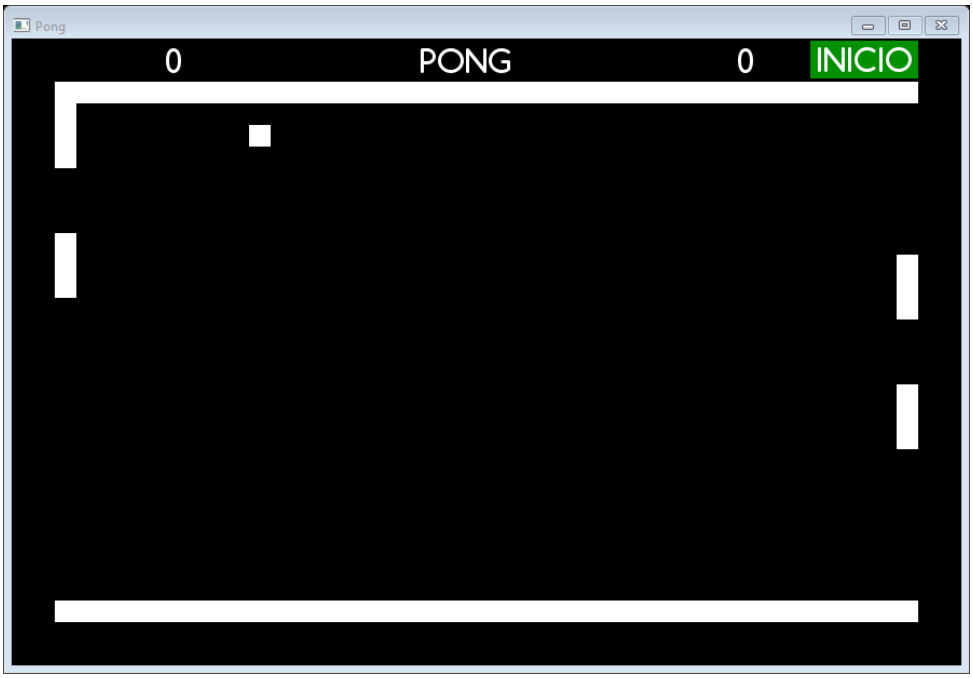
A continuación, diferentes capturas de pantalla del programa.

Pantalla de inicio:

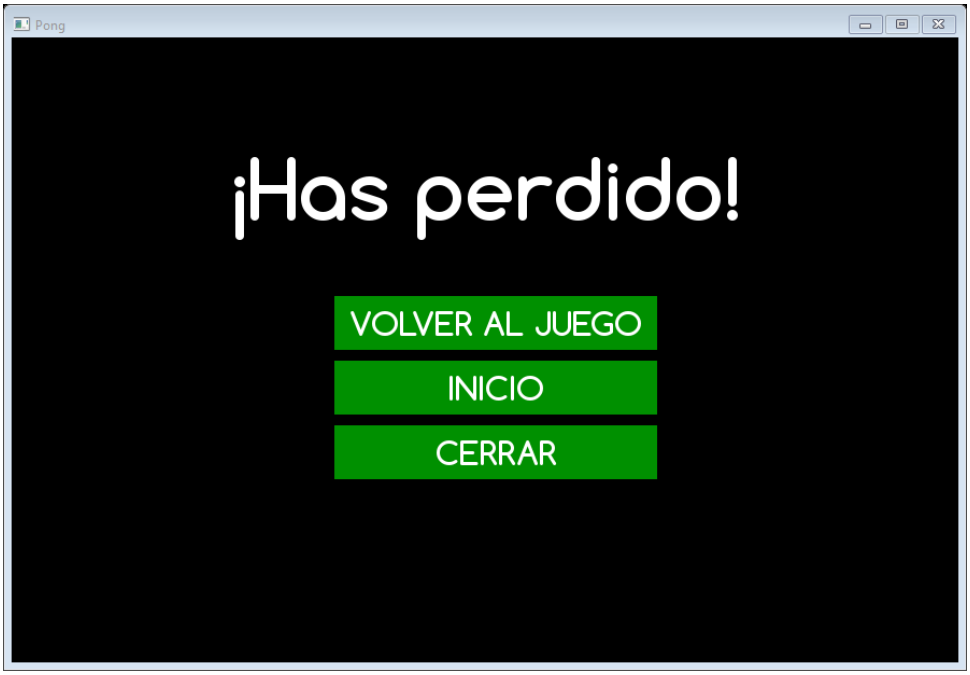
Menú de inicio:

Pantalla de juego:





Pantalla de jugar de nuevo:



DIFICULTADES EN EL PROCESO

La implementación de las clases resultó complicada al inicio, en lo que se refiere a la interacción entre ellas y con el juego. Para solucionarlo se hizo uso de métodos set y get, que permiten tomar un dato del juego (get) y asignarlo (set). Mediante el uso de estos métodos y algunas funciones adicionales los ciclos del juego son capaces de interactuar entre y dentro de ellos, y modificar las variables que lo definen, constantemente.

Debido a que se decidió implementar Pygame como única interfaz gráfica, la ausencia de una función para los botones resultó complicar el proceso debido a que se debieron asignar todas las funciones para los diferentes botones, además de escribir las diferentes condiciones para que botones cumplieran con su asignación. Por ejemplo, los botones del menú de inicio para seleccionar los modos de juego, requirieron de una función aparte porque cada vez que se seleccionar una opción, todos los cuadrados de las opciones vuelven a ser dibujados. Adicional, también se escribió una función para botones con texto, principalmente con las funciones: cerrar, volver al menú de inicio y aceptar.

Al inicio del proyecto, GitHub resultó ser una herramienta complicada de utilizar. La sincronización de los archivos y todo lo relacionado con la herramienta demostraron ser difícil de manejar, tras varios intentos y muchos errores fuimos capaces de entender mejor el programa.

Por último, el trabajo en equipo y la comunicación. A pesar de que ambos hemos trabajado en distintos grupos, nunca habíamos trabajado juntos ni nos conocíamos por lo que debimos entender cómo funciona y trabaja cada uno para solucionar las diferentes situaciones que se presentaron en el trayecto.

BITÁCORA

Día 1: 4/05/18: Montamos el proyecto en GitHub e investigamos sobre la aplicación. Dejamos las clases definidas.

Día 3: 5/05/18: Se programaron las clases con los respectivos argumentos, además de dar inicio con la creación de la interfaz gráfica con la librería Pygame. Se Crea Pong 2.0 ya que se dieron múltiples problemas con los archivos. Además de que las ramas locales estaban por encima de la rama master por varios comentarios. También se dieron varios conflictos que no se pudieron resolver.

Día 4: 6/05/18: Se crean los archivos de pruebas para poder editar el código allí y luego implementarlo al código principal cuando estemos seguros. Luego de trabajar en estos archivos auxiliares se agregaron los cambios realizados a Pong.py. Se crea la función generadora de la matriz a la cual corresponde el tablero del juego, además de una función que sólo se utilizará a nivel de consola para conseguir el índice correcto dentro de la matriz de mil elementos. Se optimiza la clase juego. Se arreglan errores varios que afectaban el funcionamiento del juego.

Día 3 7/05/18: Montaje del menú de inicio, interfaz. Se termina la sección de variables necesarias para el correcto funcionamiento del juego. Se inicia el montaje de los ciclos de los diferentes modos de juego. Además de la colocación de todos los elementos dentro de la pantalla del juego.

Día 4 8/05/18: Se agregaron los rebotes de la bola en las paletas, incluyendo las secciones donde debe rebotar según la posición de la paleta donde colisiona. Además del movimiento de la paleta del jugador 2 con sus respectivos bordes. Por último, se crea la detección de puntos por cada jugador dependiendo de la zona que traspase la pelota.

Día 5 9/05/18: Se crea el sistema de detección de puntos, donde luego de sobrepasar los límites, la pelota aparece en el centro de la pantalla, espera un segundo y luego comienza de nuevo. También se optimiza la funcionalidad y el aspecto del menú del juego.

Dia 6 13/05/18: Se definió un segundo ciclo para el menú de inicio que define las variables para el modo de juego, la dificultad y los jugadores. Además, se trabaja la interfaz del menú. Se crea el método getModo en la clase juego para solicitar si serán una o dos paletas. Además de agregado el modo dual con sus respectivas colisiones. También se agrega la tipografía que utilizarán los textos dentro del juego. Con esto se agrega el título y los marcadores dentro del juego. Se crea la función botón para la selección de modalidades dentro del menú. Se arreglan diversos bugs relacionados con los bordes y sus respectivas colisiones. Se adelanta la documentación interna del código.

Día 7 17/05/18: Arreglada la bitácora gracias a los comentarios dentro de Git.

Dia 8 19/05/18: Se arreglan errores varios. Además de crear la pantalla que emerge una vez que alguno de ambos jugadores alcancen la máxima puntuación que en este caso se fijó en 10.

Dia 7 14/05/18: Se continúa el menú de inicio. Los botones aun no quedan de color diferente al ser seleccionados, pero ya definen las funciones y llaman a la clase de juego indicada. Se junta el código del menú con el oficial. Se crean las funciones para los diferentes niveles de dificultad, incluyendo las colisiones dependiendo del tamaño de la paleta y su respectiva sección. Se hacen mejoras a las colisiones, al menú y arreglos de bugs varios. Arreglado error para que la distancia entre paletas del modo dual sea proporcional al tamaño de cada paleta.

Día 8 15/05/18: Hechas las colisiones de las paletas con los bordes en el modo dual, además de los rebotes de la bola con sus respectivas direcciones. Arreglo de algunos errores en el menú y diversas mejoras. Además de avanzar con la documentación interna del código. Se fusiona el código del juego junto con el código del menú.

Día 9 16/05/18: Se agrega la verificación en caso de que el usuario quiera jugar contra la computadora o contra otro jugador. Se agregan sonidos a los rebotes, tanto para cuando la pelota pega contra una paleta como cuando sucede con uno de los bordes. Se comenta gran parte del código. Además de notables mejoras en la función botón.

Día 10 18/05/18: El menú quedó listo, los botones funcionan bien. Cada pantalla tiene su propio botón que redirige al inicio de ser necesario.

Día 11 3h 19/05/18: Se definió una pantalla de inicio de juego. Se trabajó más la interfaz y se agregaron los botones para las pantallas de jugar de nuevo.

Día 12 20/05/18: Se le agregó música al juego, además de añadir el archivo .mp3, se finaliza con la documentación externa e interna.

TABLA DE HORAS

FUNCIÓN	Fabián Ramírez	Tomás Segura	TOTAL
Análisis de Requerimientos	1:00 Hora	1:00 hora	2:00 horas
Diseño de la aplicación	3:00 horas	6:00 horas	9:00 horas
Investigación de funciones	6:00 horas	4:00 horas	10:00 horas
Programación	11:00 horas	8:00 horas	19:00 horas
Documentación interna	3:00 horas	2:00 horas	5:00 horas
Pruebas	3:30 horas	2:00 horas	5:30 horas
Elaboración del documento	4:00 horas	4:00 horas	8:00 horas
TOTAL	31:30 horas	29:00 horas	58:30 horas

CONCLUSIONES

La programación orientada a objetos es sumamente útil para la simplificación de código. Los métodos get y set permiten manejar las diferentes variables sin la necesidad múltiples líneas de código que representen lo mismo. En nuestro caso, nos permitieron dibujar diferentes objetos con características similares entre sí sin la necesidad de escribir lo mismo muchas veces. La herramienta GitHub es muy útil para compartir código entre sí y mantener un trabajo en equipo a distancia.

Pygame es una interfaz que debe ser usada, preferiblemente, para el diseño de programas con transiciones y animaciones, más que para funciones básicas como la definición de textos y botones debido a que es más complicado que en otras como Tkinter (anteriormente usada). Fuera de eso, facilitó mucho la escritura del juego.

Finalmente, a partir de este proyecto se pueden generar bastantes conclusiones, pero hay unas en específico a las cuales se les puede hacer énfasis. Por ejemplo, lo útil que puede llegar a ser el paradigma de la programación orientada a objetos en el desarrollo de un software eficiente ya que implementándolo de una manera correcta se puede sacar el máximo provecho del código realizado. Por otro lado, fue inesperado la gran ayuda que una herramienta como Git puede brindar. Necesita esfuerzo para entenderla por completo pero una vez se comprende su funcionamiento resulta ser una ayuda indispensable para el desarrollo de software colaborativo, aunque también resulta beneficioso para proyectos personales por el respaldo en el servidor que se genera.

Por último, resaltar la importancia de trabajar en equipo y de la comunicación que se necesita para realizar un proyecto en conjunto. No es fácil porque al programar cada persona tiene su modo de pensar, su forma de crear soluciones y para llegar a la meta se deben unir ambas perspectivas.