



Proyecto de Programación #2 – Documentación

Tecnológico de Costa Rica, Cartago
Talle de Programación – Prof. Diego Mora
Escuela de Ingeniería en Computación

Melania Morales Serrano
2019037986
Primer semestre del 2019

27 de mayo del 2019

Tabla de contenido

Introducción.....	3
Descripción del problema.....	4
Análisis de resultados	5
Estadísticas de tiempos (Tabla).....	6
Conclusión personal	7

Introducción:

En el siguiente trabajo hablaremos de la creación de una versión del juego “4 en línea”.

El juego “4 en línea” original consta de una matriz de 6 filas y 7 columnas y 42 fichas de dos colores distintos, las cuales se reparte en igual cantidad a cada uno de los dos jugadores posibles. Además, cada jugador coloca una ficha a la vez en turnos intercalados. Es posible que el juego quede empatado.

Para este segundo proyecto de programación de los estudiantes de ingeniería en computación en el curso “Taller de programación” con el profesor Diego Mora, el juego del “4 en línea” es un poco diferente. Entre las instrucciones distintas se encuentra el hecho de que la matriz ya no tiene una medida de 6x7 únicamente, sino que ahora será infinita y se podrá colocar la ficha en la matriz que se quiera. Ojo, que existe una restricción, el jugador solo podrá colocar la ficha en columnas que estén a una distancia máxima de 7 columnas de una ficha.

Aparte de esto las otras reglas se mantienen.

Descripción del problema

Siguiendo con el hilo de la introducción, el problema era crear una versión del juego 4 en línea con matriz infinita, esto además de otro plus, debería tener un modo de juego VS la computadora, más conocido como IA.

Para comenzar tomamos nuestro primer componente, la representación gráfica:

1. En pantalla siempre debe verse 6X7 fichas y tendrá que implementar un "scroll" para ver la matriz si ha crecido, sin embargo, para más facilidad, solo se verán 6x7 en todo momento.
2. Importante es que las columnas y filas deben estar siempre numeradas, para saber en todo momento en cuál posición de la matriz se está.
3. El programa debe ser gráfico, mostrar colores de fichas según el estado de la matriz.
4. También debe mostrar cuál es el jugador que le corresponde el turno.

Luego tomamos nuestro segundo componente, los jugadores:

1. Solamente pueden participar dos jugadores, donde cada uno tendrá un color de ficha.
2. Los turnos son intercalados.
3. En la pantalla debe mostrarse el turno.
4. Antes de iniciar el juego, debe solicitarse el nombre de ambos jugadores.

Luego tomamos nuestro tercer componente, el juego contra la computadora (IA):

1. Puede jugar un solo jugador contra la computadora.
2. Este modo de juego tendrá que implementar un algoritmo que haga el segundo turno.
3. Debe ser un algoritmo que se base en colocar la ficha donde no se pierda, es decir, donde el otro jugador no pueda formar cuatro en fila. [Cuidado porque habrá ocasiones en que no hay alternativa y se le podría "enciclar" el algoritmo].

Luego tomamos nuestro cuarto componente, los puntajes:

1. Debe tener un archivo donde guarde los puntajes (cantidad de juegos ganados) de todos los jugadores del juego.
2. Además, una pantalla donde se puedan consultar los puntajes, ordenados del mayor a menor.

Para finalizar tomamos nuestro quinto componente, guardar partida:

1. Debe permitir guardar una partida no finalizada y luego poder jugarla nuevamente en el estado en el cual quedó. NOTA: Para esto utilice archivos para almacenar en disco las partidas.

Análisis de resultados:

Al probar el proyecto me di cuenta de que todo lo que tenía funcionaba correctamente, pero hubo una función que me encantaría haber tenido tiempo de implementar y es la de validación de posición, ya que no la terminé de desarrollar.

Para el desarrollo del juego con su interfaz utilice Pygame, debido a que me parecía un reto nuevo para afrontar y que la interfaz resultante con tkinter no era mucho de mi agrado.

Entre los análisis más sobresalientes están:

1. La validación de los 7 espacios la hice aproximadamente 4 veces en ninguna ocasión me funcionó de manera correcta.
2. El IA (juego contra la computadora) no tiene un opción de guardar partida y no sabía si esta era necesaria.
3. El IA (juego contra la computadora) funciona y me parece que bastante bien para todo el esfuerzo que conllevo, pero, me hubiera gustado poder implementarle más funciones para que fuera un poco más inteligente y también fuera a ganar, dándole más dificultad al juego.
4. El guardar partida solo guarda la partida más reciente, no guarda una lista de partidas. Esto me generó desconfianza porque no sabía qué era lo necesario, pero no tuve mucho tiempo para seguir pensando en eso.
5. El uso del "Pygame.update" me generó mucha confusión porque en ocasiones olvidaba ponerlo y eso generaba que no me trabajaran las cosas de la manera correcta. Con los días fue aprendiendo de mis errores, pero en principio es muy tedioso.

Una de las funciones más complicadas fue definitivamente el IA, el trabajo que conllevó y la gran cantidad de condiciones que tenía fue muy agotador. A pesar de esto estoy satisfecha con el resultado y esperarí que funciones bastante bien.

También cabe rescatar que como dije anteriormente la función de validar los 7 espacios se me complicó y no me funciona como debería, por lo tanto, no la implementé.

Entonces se podría decir que un 90% de mi juego funciona en este momento al realizarle pruebas.

Estadísticas de resultados (Tabla)

Análisis de requerimientos	12 horas
Diseño de la aplicación	60 horas
Investigación de funciones	20 horas
Programación	80 horas
Pruebas	30 horas
Elaboración documento	1 hora
TOTAL	193 horas

Conclusiones:

Al concluir este trabajo tengo un pensamiento de satisfacción debido a que el esfuerzo tuvo sus frutos. Esta es mi segunda progra en toda mi vida y la verdad es que he aprendido mucho.

Gracias al trabajo logré interesarme más en aprender nuevas funciones que contiene Python y aprendí la gran cantidad de cosas que te deja hacer Pygame.

El uso de Pygame fue completamente un reto, la dificultad de la progra aumentó al tener que aprender en el camino, pero obviamente era una gran práctica.

Además, volví a poner en práctica el “divide y vencerás”, muchas de mis funciones presentaban problemas al estar tan cargadas y el dividir las me ayudó mucho a la limpieza de mi código y su funcionalidad.

También aprendí a pensar “¿Qué me gustaría a mí ver si la tuviera que usar?”, pensar en las necesidades, gustos y comodidades de la persona al estar jugando, siendo crítica del juego como algún día yo fui con el mío.

Otro punto para tomar en cuenta es que al igual que en la progra pasada hice el uso de los .txt que me ayudaban a guardar en disco información y a poder cargarla cuando quisiera. Esta herramienta me parece fácil y muy cómoda para trabajar.

Además el uso del mouse y los clics para operar la matriz me generó algunas fallas e inconvenientes, por ejemplo, los clics son muy sensibles en mi laptop entonces a veces tiene a poner dos clics en lugar de uno, también a un rato de usarlo te das cuenta de que se vuelve un poco loco y los clics pasan a valer más de la cuenta, cuyo funcionamiento no me logro explicar bien del todo y por último el uso de los botones que me generó conflicto en algunas ocasiones debido a que al darles clic a acción se implementaba en más de una ocasión.

A pesar del gran reto que fue este trabajo sé que me falta aprender mucho y que vienen cosas más fuertes, esta experiencia brinda mucho conocimiento y sé que valió la pena todo el esfuerzo dado.