**TAREA. HACER PROFESIONAL.**



**Christian Alejandro Pereda González**

**Octavio Valdez Fonseca**

**Profesor: José Antonio Camarena Covarrubias**

**Programación con memoria dinámica**

**07 de diciembre del 2021**

**Conclusiones Christian:**

Este trabajo estuvo lleno de aprendizajes, nos dimos cuenta que las decisiones de implementación del juego pueden llegar a afectarnos tanto que debemos cambiar por completo la estructura del programa. El primer desafío fue aprender a usar raylib, después de averiguarlo pudimos dibujar nuestro tablero con interfaz gráfica, dibujando el tablero de damas chinas y las fichas negras y blancas, luego de eso decidimos usar estructuras autorreferenciadas para identificar cada una de las fichas y en cada uno de sus atributos le introdujimos lo necesario para identificar una ficha y diferenciar los colores, también agregando un id único para identificar cada ficha individual.

Otro problema fue el turno, primero quisimos hacer una función con voids para poder cambiar el turno haciendo un swap, y luego de muchos intentos nos dimos cuenta que eso nos causaba problemas y por eso no se cambiaba el turno correctamente, entonces decidimos cambiar el turno directamente en main escribiendo el turno contrario luego de cumplir ciertas funciones.

Lo siguiente que quisimos implementar fueron los círculos rojos que aparecen en el tablero cuando tienes movimientos disponibles por hacer, y creo que eso fue un error de nuestra parte porque primero nos preocupamos por eso en lugar de los movimientos, pues el movimiento que hacíamos para probar los círculos rojos era muy genérico y las fichas no se comportaban realmente como debían, entonces cuando quisimos implementar el movimiento teníamos bugs muy raros que a veces nos tenía hasta un día entero sin avanzar (bugs de los que después supimos la razón y nos dio mucho coraje pero ya habíamos solucionado todo de otra manera). Luego de terminar todas las condiciones para los círculos rojos nos hicimos cargo del movimiento, y al terminar de implementarlo nos dimos cuenta que el turno no funcionaba al poder hacer un movimiento a la derecha con cualquiera de las fichas, fue entonces que nos dimos cuenta de el error que fue detectar las fichas con base a su posición en la interfaz gráfica usando un clic del usuario como referencia.

Luego de no poder arreglar ese bug y de tener otros muy raros, el profesor Josean nos recomendó detectar fichas con base a un arreglo, teniendo atributos también en una estructura que nos diga qué tipo de ficha hay ahí y si hay una ficha.

Cambiamos toda la estructura de nuestro código usando este arreglo en el que se guarda la posición de cada cuadro y si está disponible o no, pero el hecho de hacer todas las funciones desde cero nos ayudó a optimizar y acortar nuestro código teniendo condiciones más asertivas a lo que necesitamos y más cortas que con nuestra anterior implementación, pudimos arreglar los cambios de turno y agregar condiciones a nuestras funciones que evitan que nos dé segmentation fault el código por meterse en direcciones del tablero que no existen.

El siguiente problema a resolver fue obtener al ganador, en el que no tardamos mucho, pero sólo implementamos una solución a ganador si se acaban todas las fichas del oponente, lo que usamos fue la vida de la ficha que es un atributo de su estructura, y de esa manera se obtiene un ganador desde main.

Por último, otro de los retos importantes fueron los archivos en c para guardar y cargar una partida, realmente estábamos sin saber qué hacer, pero al final pudimos realizarlo correctamente tomando en cuenta que sólo se pueden guardar voids en el archivo. Usamos arreglos donde guardamos toda la información de todas las fichas, pero de los atributos por separado, y también guardamos el turno y si se trata de una reina, luego de guardar todo en esos arreglos, añadimos en el archivo la información que en estos hay, y para cargar una nueva partida se obtiene lo que hay en esos arreglos dentro del archivo y se va leyendo y añadiendo la información de cada ficha una por una. Lo difícil fue dar con la solución que incluye a estos grandes arreglos, pero una vez que dimos con eso fue todo mucho más fácil.

Por esto digo que estuvo lleno de aprendizajes este proyecto, pero después de todo esto me voy satisfecho porque me doy cuenta que soy capaz de programar cosas realmente buenas que no sólo son problemas de matemáticas que se resuelven en la consola. Me voy feliz y con ganas de seguir aprendiendo cada día.

Muchas gracias por su paciencia y su tiempo profesor, espero encontrármelo en otra materia y así me pueda compartir un poco más de su conocimiento.

**Conclusiones Octavio:**

Al principio del proyecto todo parecía muy abstracto ninguna idea conectaba y no sabia cómo funcionaba la librería de raylib, por eso mismo me puse a estudiar sus funciones y variables espaciales para empezar el proyecto. Lo cual me ayudo a conocer como investigar diferentes conceptos dentro del mundo de la programación y que casi todas las librerías tienen una cheat sheet en internet. Pensaba que este tipo de implementación dentro de C iba a ser complicada pero después de todo un año dentro de este lenguaje ya se hizo parte de mí.

Yo creo que lo más importante que aprendí fue a detectar errores y poder resolverlos, especialmente cuando se trata de acceder a memoria ilegal, ya que dentro de las listas de las fichas negras y blancas al elemento tenía elementos ilegales, además que como siempre la manipulábamos esta lista debería estar en la cárcel de tantas veces que accedí a memoria la cual no se debía de acceder. Claro que me ayudo a que por cada problema similar me tomaba menos tiempo resolverlo.

Los apuntadores fueron lo más importante de este proyecto por el echo de que por cada movimiento estábamos modificando el valor de una variable en especifico y aunque ya en este punto los pase por sentado por todo lo que eh aprendido en programación últimamente me van a ser extremadamente útiles en materias futuras y al mismo tiempo me ayudaran a implementar proyectos en algún futuro.

Algo que me ayudo a ver las cosas de una forma totalmente diferente fue la implementación de un arreglo para el tablero, ya había visto antes en videos que lo hacían, pero nunca entendía como se relacionaba la ventana del juego con un arreglo del programa, ahora pude relacionarlo e implementarlo correctamente, además de que esto ayudo a facilitar la lógica del proyecto, ya que, no dependíamos completamente de las coordenadas de la ventana.

Tal ves no fue el proyecto perfecto ni de cerca pero puedo decir que me voy lleno de aprendizajes y ganas de poder expandir mi panorama de la programación, esta practica hizo que mi interés hacia la programación sea más fuerte y me recuerde la razón por la cual elegí esta carrera.