

Instituto Tecnológico de Costa Rica
Centro Académico de Alajuela
IC4700. Lenguajes de Programación



Tarea Programada #4:
Programación imperativa con C

Estudiantes:

Valery Carvajal Oreamuno – Carné: 2022314299
Anthony Josué Rojas Fuentes – Carné: 2018027141

Profesor:

Maria Auxiliadora Mora Cross

Fecha de entrega

13/11/2023

II Semestre, 2023

Tabla de Contenidos

Descripción del Problema.....	3
Diseño de Componentes:.....	4
Instrucciones del Juego.....	5

Descripción del Problema

Objetivo del proyecto:

El objetivo del presente trabajo es poner en práctica el conocimiento adquirido durante el curso sobre programación imperativa y diseñar e implementar funcionalidad utilizando el lenguaje de programación C.

Historia del Juego:

Len Choa es un juego tradicional tailandés que ha existido durante muchos años. Es un juego de estrategia que combina elementos de caza y persecución. La historia y origen exactos del juego no están bien documentados, pero se sabe que ha sido parte de la cultura tailandesa y que se juega principalmente en esa región.

Como ocurre con muchos juegos tradicionales, Len Choa posiblemente tiene raíces históricas y culturales, vinculadas a la mitología o a las actividades de caza y supervivencia. A lo largo del tiempo, los juegos tradicionales han servido como formas de entretenimiento, transmisión de habilidades y valores culturales.

La falta de documentación detallada puede deberse a que muchos de estos juegos se han transmitido oralmente de generación en generación. Además, el énfasis suele estar en la experiencia del juego en sí, más que en su historia escrita.

Descripción del Juego:

El programa de la tarea está diseñado para implementar el juego "Len Choa: Leopards and Tigers" en el lenguaje de programación C. Utiliza listas doblemente enlazadas con punteros para representar el tablero del juego, asignando a cada nodo información sobre la posición y el tipo de ficha (tigre o leopardo).

En términos de reglas, el programa define estructuras y funciones que aplican las reglas específicas de Len Choa. Esto incluye verificar la validez de los movimientos, manejar los turnos de los jugadores y controlar el flujo del juego. Los jugadores interactúan ingresando sus movimientos desde el teclado, y el programa actualiza las estructuras de datos según estas decisiones, asegurándose de que los movimientos sean legales. La visualización del estado del juego se realiza de manera amigable, mostrando el tablero después de cada movimiento con caracteres especiales para representar al tigre, a los leopardos y el tablero en sí. El programa valida los movimientos para prevenir jugadas ilegales y garantiza el cumplimiento de las normas del juego.

Además, se implementan funciones que verifican si un jugador ha ganado, considerando condiciones como la inmovilización del tigre o el número de leopardos restantes. La interacción con el usuario es intuitiva, permitiéndoles ingresar movimientos y proporcionando mensajes informativos sobre el estado del juego.

En última instancia, el programa proporciona una experiencia completa del juego Len Choa en C, donde dos jugadores compiten por la victoria siguiendo las reglas establecidas, y el programa determina el resultado del juego.

Diseño de Componentes:

Estructuras de Datos:

Nodo:

Atributos: valor, siguiente, anterior, izquierda, derecha.

Métodos: crearNodo(int valor).

Lista:

Atributos: actual, raiz, final, longitud.

Métodos: crearLista(), agregarAlFinal(Lista* l, Nodo* n), agregarAlInicio(Lista* l, Nodo* n), len(Lista* l), recorrerAdelante(Lista* l), recorrerAtras(Lista* l).

Tablero:

Atributos: inicio, actual, columna1, columna2, columna3, niveles.

Métodos: crearTablero(int levels), imprimirTablero(Tablero* t), iniciarJuego(Tablero* t).

Funciones Principales:

crearNodo(int valor): Crea un nuevo nodo con el valor especificado.

crearLista(): Crea una nueva lista vacía.

crearTablero(int levels): Crea un nuevo tablero con el número especificado de niveles.

*agregarAlFinal(Lista l, Nodo n):*** Agrega un nodo al final de la lista.

*agregarAlInicio(Lista l, Nodo n):*** Agrega un nodo al inicio de la lista.

*len(Lista l):** Obtiene la longitud de la lista.

*recorrerAdelante(Lista l):** Recorre la lista hacia adelante e imprime sus elementos.

*recorrerAtras(Lista l):** Recorre la lista hacia atrás e imprime sus elementos.

*imprimirTablero(Tablero t):** Imprime el estado actual del tablero.

*iniciarJuego(Tablero t):** Inicia el juego permitiendo a los jugadores realizar movimientos y actualizando el estado del tablero.

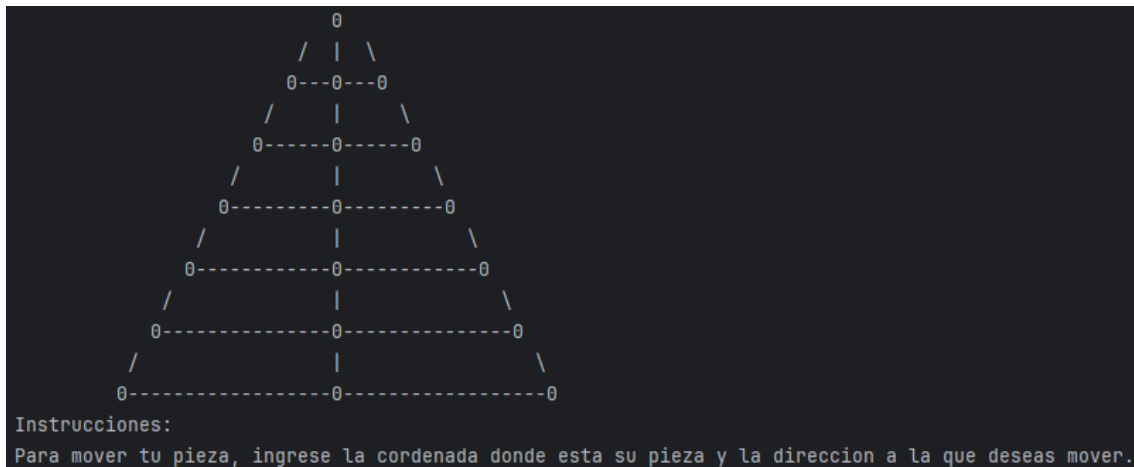
Funciones Complementarias:

main(): Función principal que crea un tablero, imprime su estado e inicia el juego.

calloc(), printf(), scanf(): Funciones de la biblioteca estándar de C utilizadas para la asignación dinámica de memoria, imprimir en consola y leer desde el teclado.

Instrucciones del Juego

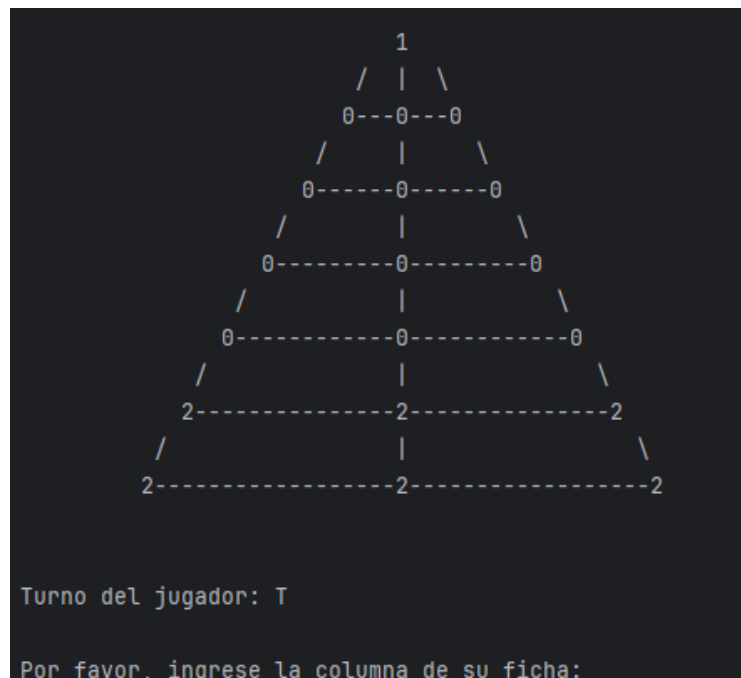
1. Se despliega el tablero con las instrucciones.



2. Los jugadores deciden quién jugará con el tigre y quién con los leopardos. El tigre se coloca inicialmente en la parte superior del triángulo, y los leopardos se colocan abajo en el tablero.

El tigre se ha colocado en la cuspide del tablero.
Los leopardos han tomado su posicion, empieza el juego!

3. Se despliega el tablero en su forma inicial.



4. El juego turna automáticamente a los jugadores, quienes ingresan desde el teclado las coordenadas de la ficha que desean mover y su destino.

```
      1
     / | \
    0---0---0
     / | \
    0-----0-----0
     / | \
    0-----0-----0
     / | \
    0-----0-----0
     / | \
    2-----2-----2
     / | \
    2-----2-----2

Turno del jugador: T

Por favor, ingrese la columna de su ficha:1
Por favor, ingrese la fila de su ficha:0
Por favor, ingrese la direccion de su
movimiento:2
```

5. Las posiciones se actualizan y se pasa a la siguiente ronda.

```
      0
     / | \
    0---0---0
     / | \
    0-----0-----0
     / | \
    0-----0-----0
     / | \
    0-----0-----0
     / | \
    2-----2-----2
     / | \
    2-----2-----2

Turno del jugador: L

Por favor, ingrese la columna de su ficha:|
```