
Información general

Materia: Laboratorio I

Comisión: 1-605

Turno: Noche

Docentes:

- Paula Ithurralde
- Miguel Silva
- Jhonathan Pécora
- Alejandro Beringher

Alumnos/as:

- Federico Carro
- Mariciel Oriana Marín Méndez
- Catalina Galgani

Link Github: <https://github.com/FedericoCarro/ProyectoFinalLab>

Descripción del proyecto

Proyecto: Juegos Arcade

Idea principal del proyecto:

- Realizar un programa que contenga 4 juegos distintos con los cuales los jugadores puedan interactuar varias veces si así lo quisieran o probar distintos juegos. El mismo será programado en Pseint.
- Juegos pensados:
 - Pacman
 - Laberinto
 - Batalla naval
 - Sudoku
- Cada juego será construido por cada uno/a de los participantes del grupo junto con la colaboración de los demás.

- **Menú principal interactivo**

Este menú se utilizará para la selección del juego que se quiera jugar en el momento.

```
*****
*                                                                 *
*                                                                 *
*                                JUEGOS ARCADE                      *
*                                                                 *
*                                1- LABERINTO DE MEEP              *
*                                                                 *
*                                2-PAC-MAN                        *
*                                                                 *
*                                3- SUDOKU                        *
*                                                                 *
*                                4- BATALLA NAVAL                 *
*                                                                 *
*                                                                 *
*                                                                 *
*****
```

- **1-EL LABERINTO DE MEEP**

Trata sobre un laberinto clásico, con la particularidad de que tiene la visión limitada por secciones, a medida que se mueve el personaje “Meep” van cambiando los cuadrantes que observa, en la ilustración se muestra un ejemplo de cómo dividirlo en esas secciones.

Funcionalidades principales:

- Movimiento del personaje “Meep” con letras y enter:
 - A: izquierda
 - S: abajo
 - D: derecha
 - W: arriba
- Nota técnica: se puede ingresar varias letras juntas y el mismo se moverá en las posibilidades ingresadas.

Condición de victoria:

- Encontrar la llave, llegar a la puerta y salir del laberinto.

Condición de derrota:

- No encontrar la llave y no poder salir del laberinto mas dejar de jugar el juego.

Funciones:

El juego cuenta con una cantidad de 11 funciones las cuales son:

Que no reciben, ni retornan:

- Función JuegoLaberinto
 - Esta es llamada desde la función principal de todo el paquete de juegos.
 - Dentro de la misma se encuentran las 3 opciones del switch segun que opción de menu
- Función JugarLaberinto
 - En la misma se inician variables, vectores, matrices que seran usadas durante el codigo para el funcionamiento del juego.
 - Se valida si el personaje agarro o no la llave en el juego
- Funcion Animacion_Inicio_Lab
 - Carga la visualización de la animación que se puede observar al inicio del juego.
- Función Instrucciones_Laberinto
 - Se visualizan las instrucciones del juego, los controles, la llave y el personaje de este. Más una leve descripción de cuál es el objetivo del mismo.
- Función Victoria_Laberinto
 - Animación de cuando terminas de jugar el laberinto

Que recibe valores:

- Función MovValidoLab(VectMovPosibles,PosX Por Referencia,PosY Por Referencia,MovSing,Contador Por Referencia)
 - Esta misma recibe la matriz VectMovPosibles para tener conocimiento de cuales son los movimientos validos segun la posicion. Recibe tambien PosX y PosY como punteros para poder editar su valor en caso de ser posible. Recibe el MovSing que corresponde al movimiento actual a analizar y recibe el contador de pasos como puntero para sumarle un valor en caso de ser valido el movimiento.
- Funcion MovimientosPosibles (Movimientos)
 - Recibe el vector de movimientos posibles del Meep el personaje que va por el laberinto, el cual será cargado posición por posición con el número que corresponde al switch que valida en la Función JuegoLaberinto si puede realizar el movimiento/s para el lugar que esta solicitando el usuario.
 - Se penso en cargar los valor de izquierda a derecha en el valor de x del juego.
- Funcion VisualPorBloques (PosX,PosY,Llave,Contador,Ceros,Carita)
 - Se carga la visual segun el bloque en el cual se encuentre el personaje.
 - La visual de cuando ve la llave, de a la aggarla, de continuar y todo lo que se ve a nivel visual en el juego.
 - Se carga en el tablero la visual del personaje segun los valores en X e Y en la matriz.

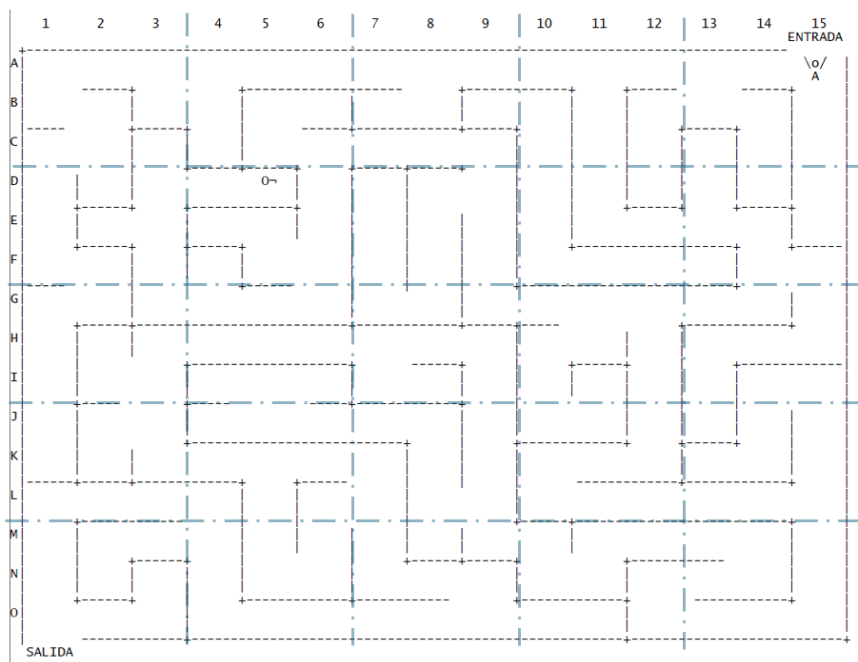
Que retorna valores:

- Función OP_Lab <- Menu_Lab
 - Retorna el valor seleccionado pro el usuario del menu del juego
 - Carga la visual del menu del juego
- Función Pausa<-PausaLaberinto

- Función que retorna el valor de la variable pausa para continuar el juego, revisar las instrucciones o salir del mismo.

Que reciben valores y retorna:

- Función Vertical<-MovimientoVertical(Vertical,MovPersonaje)
 - Recibe una variable de tipo carácter correspondiente a los movimientos en Y y el vector de movimientos posible más evalúa si la dirección es hacia arriba o abajo. Sumando o restando la posición en Y.
 - Retorna el valor de vertical.
- Función Horizontal<-MovimientoHorizontal(Horizontal,MovPersonaje)
 - Recibe una variable de tipo carácter correspondiente a los movimientos en X y el vector de movimientos posible más evalúa si la dirección es hacia derecha o izquierda. Sumando o restando la posición en X.
 - Retorna el valor horizontal.
- Funcion MovPersonaje<-CargaDeMovimiento(MovPersonaje)
 - Lee el carácter ingresado por el usuario para retornar el valor a la función JuegoLaberinto



VISUAL PSEINT:

- Algoritmo para que realice movimientos aleatorios el fantasma por el laberinto
- Conteo del puntaje a medida que el Pacman va avanzando y comiendo las monedas esparcidas por el laberinto.
- SuperPacman poder para matar los fantasmas y ganar aun mas puntaje.

Condición de victoria:

- Mientras el Pacman continúe con una vida como mínimo el misma al terminar de recolectar todas las monedas del tablero, volverá al inicio.
- **Nota técnica:**
 - al llegar a recolectar las 214 monedas se le suma 1 vida al Pacman pero se le fuerza el choque con un fantasma.

Condición de derrota:

- Si al Pacman lo toca un fantasma este pierde una vida y si el mismo se queda sin las 3 vidas el juego esta terminado.

Funciones:

Que retornan vacío:

- Funcion JuegoPacman
 - Se ven las opciones de Menú del juego y su bienvenida
 - El menu se visualiza llamando la funcion OpcionPacman<-- MenuPacman
- Función InicioPacman
 - Es donde se definen variables, vectores y matrices principales del juego, más mediate ciclos se cargan sus datos iniciales.
 - Se llaman a las funciones correspondientes para que continúe el juego, pero ya estás puedan recibir cargados e iniciadas las variables, vectores y matrices
 - Se validan movimientos del pacman, más a consecuencia los movimientos que deben hacer los fantasmas
 - Se valida si el pacman tiene el poder o no activado.
 - Se valida el choque entre fantasma y pacman. Si el pacman tiene o no el poder activa, más esto se verá reflejado en si se le debe sacar o no una vida al mismo.
- Función Intrucciones
 - Es la opción 2 del menu que se visualiza dentro de la función JuegoPacman
 - Aca se explican que controles se deben usar y como se mueve el Pacman
 - Se presentan los personajes
 - Más se explica que el juego contiene opción de Pausa.
- Función AnimacionPacman
 - La misma contiene ciclos para realizar una animación mediante repetir el tablero con monedas y sin para que pareciera que se está recargando el tablero.

Que retornan un valor:

- Función OpcionPacman<-- MenuPacman
 - Carga la visual del menu que se llama en la funcion JuegoPacman
 - Retorna la opción ingresada por el usuario.
- Función Pausa<--PausaPacman
 - Al usuario ingresar P en vez de cualquiera de los otros caracteres con los cuales se juega el pacman este vendra a la función Pausa<--PausaPacman y se carga la visual de Pausa del juego, mas el jugador ingresar la opción que desea se leera la misma y se retornara el valor a la función

Que reciben valores, pero no retornan:

- Función CartelPuntaje(puntaje,Vidas)
 - Se llama a esta función cuando el pacman choca con un fantasma y se visualiza la cantidad de vidas que tiene, más el puntaje realizado hasta el momento.
- Función visual(pacman,VectMovPacmanPosibles,pacmanx,pacmany,PosFant,puntaje,vidas,PacmanEnojado,movimiento)
 - Recibe los valores tanto del vector de movimientos posibles del pacman, la posición en x e y del pacman, el valor de vidas que le quedan, la matriz pacman (Vector general del tablero en donde están las monedas, lugares vacíos, fantasmas y pacman) PacmanEnojado como variable booleana de valor lógico, el carácter de movimiento para que el personaje de pacman se muestre su boca para la izquierda o derecha y PosFant Vector de posiciones de los 4 fantasmas
 - Dentro de la misma se recibe el valor de movimiento ingresado por el usuario en la función InicioPacman y dentro de esta se valida la visual del pacman.
 - Si el pacman está normal se visualiza de la siguiente manera “(·<” y los fantasmas “/ü\”, si el pacman está enojado (agarro el superpoder) se visualiza de la siguiente manera “(*<” y los fantasmas “\ü/”
 - Dentro de esta función se carga el tablero y se ubica cada posición de la matriz dentro de la visual, más las vidas y el puntaje
- Función MovimientosDisponiblesPacman(MovimientosPacman)
 - Recibe el vector de movimientos posibles del pacman el cual será cargado posición por posición con el número que corresponde al switch que valida si puede realizar el movimiento o no el pacman.
 - Se pensó en cargar los valores de izquierda a derecha en el valor de x del juego.

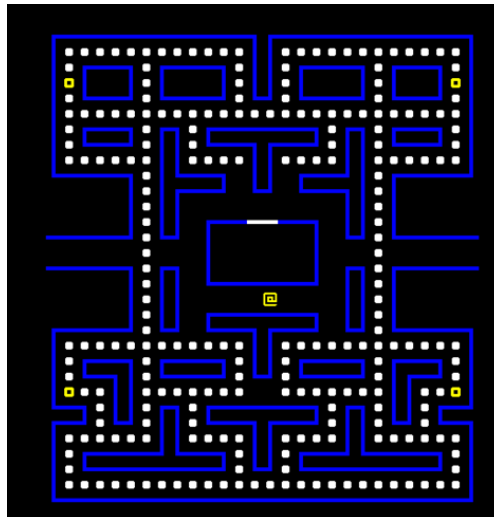
Que reciben valores y retornan valores:

- Función MovValido<-SegunVecMov(VectMovPacmanPosibles,pacmanx Por Referencia, pacmany Por Referencia, movimiento Por Referencia)
 - Recibe el vector de movimientos posibles del pacman, mas 3 punteros (2 enteros que son de la posición en x e y del pacman, 1 de carácter que represan el valor ingresado por el usuario)
 - En esta función se evalúa mediante un switch si el movimiento ingresado por el usuario es posible o no. Si no es posible por valor logico se devuelve falso y el pacman no se va

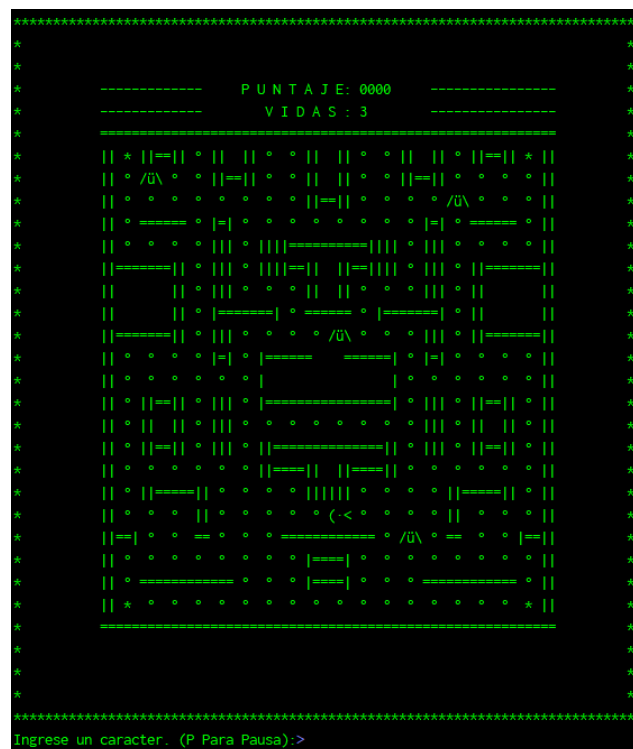
a mover. Por el contrario va a moverse segun el caracter ingresado y su dirección correspondiente.

- Función MovInicial<-BFS (VectMovPacmanPosibles, OrigenX,OrigenY,DestinoX,DestinoY,Distancia Por Referencia)
 - Es para definir cual es el movimiento mas corto entre el fantasma y el pacman
 - Variables origen es desde donde parte y destino hasta donde llega, mas distancia es para guardar la distancia entre origen y destino.

VISUAL ORIGINAL:



VISUAL PSEINT:



- 3-SUDOKU

Puzzle numérico en donde no se pueden repetir números entre filas, columnas y cuadrantes de 3x3, se deberán completar todas las casillas para poder ganar.

Funcionalidades principales:

- Es completar un tablero de 9x9 realizado por 9 cuadrantes de 3x3 sin repetir ningún número del 1 al 9, de forma horizontal y vertical. Mas a su vez dentro del cuadrante de 3x3 que se encuentran.

Condición de victoria:

- Completar el tablero de forma correcta.

Condición de derrota:

- Equivocarse 3 veces en el ingreso de los números.

Funciones:

Que no recibe ningun valor, ni retorna:

- Función JuegoSudoku
 - Se definen matrices, vectores y distintos tipos de variables que serán utilizadas durante el juego
 - Se valida mediante switch la opción de menú ingresada por el usuario en la función menú
 - Depende la opción seleccionada se lo deriva a las funciones correspondientes
- Función inicio_sud
 - Visualización de la animación del inicio del juego al seleccionarlo
- Función Inst_sud
 - Si el usuario en las opciones de menú selecciona ver las instrucciones se llamará a esta función donde se puede visualizar las instrucciones y el objetivo del juego.
- Función Carga_sud
 - Pantalla para simular que el juego se está cargando.

Que recibe valores, pero no retorna:

- Función JugarSudoku(MatrizVacía, MatrizResuelta, VectorCoordenadas, puntos_sud, vidas_sud Por Referencia)
 - Recibe la matriz vacía y la matriz con los valores que la resuelven, más el vector de coordenadas, los puntos del sudoku y por puntero la vida del mismo.
 - Es donde se reciben las coordenadas ingresadas por el usuario, se verifican y se pide el ingreso del número, más este se valida si es correcto su ingreso en esa coordenada o incorrecto.
 - Donde se realiza la suma de puntos si está bien ingresada la coordenada o pérdida de vida si no lo está.
 - Se llaman a las funciones de victoria o derrota si así fuera necesario
- Función ResolverMatriz(Sud)

- La función recibe la matriz con los valores que la resuelven
- Se carga valida y carga los valores correctos que deben ir en los cuadrantes de 3x3 y a su vez en cada subcuadrante de 3x3.
- Se valida que no se repitan, que sea correcto como se carga al azar cada número en cada coordenada. Si la solución de esa carga es válida o no.
- Función CompletarMatriz(MatrizVacía,MatrizResuelta,puntos_sud Por Referencia)
 - Según el nivel de dificultad que haya elegido el usuario se valida como se cargara la visual del tablero
 - Se carga la matriz vacía y se cargan los valores también a matriz resuelta convirtiendolo a texto así al ingresar coordenadas y el número se valida correctamente el tipo de dato ingresado.
- Función MostrarMatriz(MatrizResuelta,puntos_sud,vidas_sud Por Referencia)
 - Recibe la matriz resuelta ya cargada con los datos iniciales y si ya está jugando el usuario los ingresados por él. Los puntos acumulados y las vidas restantes como un puntero ya que es un valor que se va a ir modificando mientras se juega según lo ingresado por el usuario en otra función.
 - Es la visual del tablero de juego.
- Función perdiste_sud(puntos_sud)
 - Recibe los puntos acumulados y se visualiza una animación de que el jugador perdió, pero visualizará cuantos puntos acumuló.
- Función ganaste_sud(puntos_sud)
 - Recibe los puntos acumulados y se visualiza una animación de que el jugador ganó pero visualizará cuantos puntos acumuló.
- Función ErrorSud(vidas Por Referencia)
 - Recibe el valor de las vidas como puntero, más realiza una animación de que perdiste una vida.

Que no recibe valores, pero retorna:

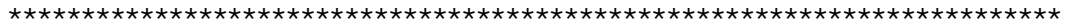
- Función Op_sud <- Menu_sud
 - Se visualiza el menú del juego y las opciones
 - Se lee la opción ingresada por el usuario y se retorna su valor así si es dentro del switch de opciones que se valida en la función JuegoSudoku arrancará el juego, leerá las instrucciones o saldrá del mismo al menú de los demás juegos.
- Función Op_sud <- Menu_Pausa_sud
 - Si selecciona la Pausa durante el juego visualizará esta pantalla, más se le pedirá al usuario que ingrese alguna de las opciones para continuar con el juego, salir del mismo o ver las instrucciones.
- Función niv<-sel_niv
 - El menú de selección de dificultad del juego
 - Se retornará la opción seleccionada y según esta la cantidad de números que se podrán visualizar en el sudoku para poder resolverlo.

Que recibe valores y retorna:

- Función CoordenadaValida<-IngresoDeCoordenadas_SUDOKU(VectorCoordenadas Por Referencia,pausa Por Referencia)
 - Donde el usuario ingresara la coordenada y luego la retornaría a la función JugarSudoku
 - Se carga el vector con las coordenadas posibles
 - Se carga la variable lógica para retornar si la coordenada no es valida
- Función Num<-IngresoDeNumero_SUDOKU(MatrizVacía,puntos_sud Por Referencia,vidas_sud)
 - Se recibe la matriz vacía, los puntos mediante puntero y las vidas como entero
 - Se retornara el valor ingresado del numero para ubicarlo en la coordena seleccionada en la anterior función
 - Se valida que sea un numero del 1 al 9

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

VISUAL PSEINT:



- Atacar y hundir toda la flota enemiga.

Condición de derrota:

- Que te ataquen e hundan tu flota.

Funciones:

El juego cuenta con un total de 19 funciones las cuales son:

Que no recibe ningun valor, ni retorna:

- Función JuegoBatallaNaval
 - Se inicia las matrices, vectores y varibales del juego.
 - Se cargan son vaores iniciales
 - Se valida la opcion de menu seleccionada
 - Se valida a quien le toca jugar si esta en la opción de dos jugadores
 - Se valida las coordenadas de la CPU si selecciono la ocpion jugar contra CPU
 - Los disparos que hara la CPU. Más los que realiza el jugador.
 - Si le erra al barco o le pega.
 - Más si ingresa o no pausa
- Función BarcoDañado
 - Se llama está función para simular la animación de como le cae el proyectil al barco (sea del jugador o su rival)
- Función BarcoErrado
 - Se llama a esta función para simular como el proyectil cae en agua y se erra ese tiro.
- Función CoordenadaRepetida
 - Simula una animación que da aviso de que la coordenada elegida ya estaba seleccionada asique es repita
- Función ErrorDeBarco
 - Simula una animación que da aviso de que la coordenada elegida y el barco seleccionado no entran en esa coordenada.
- Función NoTieneBarco
 - Simula una animación que da aviso que el barco elegido con esa longitud no es posible utilizarlo porque ya no hay cantidad disponible.
- Función MenuInstrucciones
 - Se llama a esta función para que se visualicen las instrucciones del juego y el objetivo de este.
- Función Reubicar
 - Se le consulta al jugador si está conforme con las ubicaciones de sus barcos o quiere reubicarlos
- Función GanadorJ1
 - Visualización si quien gano es el jugador 1
- Función GanadorJ2
 - Visualización si quien gano es el jugador 2

Que recibe valores, no retorna:

- Función CargaDeTableroAzar (TableroCarga,VectorCoordenadas,Jugador,PosicionBarco)
 - Se carga los barcos que ingresara la CPU
 - Se valida que no queden ensimismados, que no traspasen los bordes y se guardan como quedan colocados
- Función CargaDeTablero (TableroCarga,VectorCoordenadas,Jugador,PosicionBarco,salir Por Referencia)
 - La carga del tablero que realiza el jugador de forma manual
 - Se solicita seleccionar la longitud del barco barco y la coordenada principal en la que se ingresa la primera posición y se llenan las demás
 - Se comprueba si donde se está eligiendo es correcto y valido. No queden superpuestos y no queden traspasando paredes
- Función GuardadoCoord (VectorCoordenadas,UltCoord,Jugador)
 - Se guardan las coordenadas del jugador ya ingresadas para que estas no se repitan o no queden grabadas y pisen las anteriores ya realizadas
- Función BARCOSDISP (BarcoDisp,PosicionBarco)
 - Se valida que barco sigue disponibles para su carga en el tablero
- Función VisualBN (Tablero,BarcosDisponibles,Jugador)
 - Se carga la visual del tablero de la batalla naval
 - Se visualiza los barcos disponibles, sus longitudes y que jugador es el que debe de cargar
- Función TableroDeJuego(TableroBatalla,Jugador,BarcoDisp,UltCoord)
 - Se visualizan los dos tableros, el del jugador y su rival
 - La última coordenada ingresada por el usuario y el estado de los barcos del rival
- Función TurnoDeJugador(Jugador)
 - Se visualiza una simulación de animación para a que jugador le toca

Que no reciba variables, pero retorna un valor

- Función op<-SeleccionJugadores
 - Se visualiza el menu inicial del juego
 - Se lee la opción ingresada por el usuario para retornarla a la función correspondiente y que se verifique con un switch la opción ingresada.
- Función op<-MenuPausa
 - Se llama a está función cuando se pone en pausa el juego y se lee la opción ingresada por el usuario.

Que reciben valores y retornan valores

- Función CoordenadaValida<-IngresoDeCoordenadas(VectorCoordenadas,pausa Por Referencia)
 - Función donde se le pide al usuario ingresar la coordenada y se valida mediante el vector de coordenadas que sea correcto lo ingresado por el mismo.
 - De ser una coordenada no correspondiente se le avisara al jugador que seleccione otra y no se tomara como valida

12 Ana



bruno

```
*****  
*                                                                 *  
*                                                                 *  
*                                                                 *  
*                                                                 *  
*                                                                 *  
*                                                                 *  
*                                                                 *  
*                                                                 *  
*      Jugador   1              [1] [2] [3] [4]  
* Barcos Disponibles:    1 - 1 - 2 - 1  
*  
*          A     B     C     D     E     F     G     H     I     J  
* 1-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 2-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 3-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 4-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 5-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 6-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 7-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 8-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 9-  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
* 10- [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
*  
*  
*  
*  
*  
*  
*  
*  
*  
*  
*  
*****
```

Ingrese la longitud del barco a colocar (1-2-3-4):>