

TIC-TAC-TOE

Descripción del proyecto:

Tic-tac-toe se trata sobre el juego en el cual hay que colocar piezas en un tablero de 3x3, buscando alinear 3 piezas antes de que el jugador contrario lo haga. El juego se realiza por turnos. El proyecto se realizó en lenguaje ensamblador MIPS y se programo en MARS. Usando bitmap display como su interfaz gráfica.

Manual de uso:

Primera parte:

1. Se debe de descargar java (última versión) y el archivo para ejecutar MARS.
2. Se debe de descargar y guardar el archivo 3 en raya.asm
3. Después de ejecutar el archivo MARS, abrir el archivo de 3 en raya.asm.
4. Se debe de abrir el bitmap en MARS:
 - i. Ir a la opción Tools de MARS
 - ii. Seleccionar Bitmap display
 - iii. Ajustar los parámetros del tamaño de píxel, ambos en 32 y ajustar los parámetros del display, ambos en 512.
 - iv. Oprimir la opción de conectarlo con MIPS
5. Compilar el archivo y ejecutarlo.

Segunda parte:

1. Para jugar se debe de colocar el numero entre 0 y 8, los cuales corresponden a la casilla, en la terminal.
2. Después de colocar el número y presionar enter, la ficha se colocará en la interfaz.
3. Se busca posicionar 3 fichas alineadas.
4. En la terminal se mencionará que jugador tiene el turno.
5. Cuando se intenta colocar una pieza que ya se colocado antes, se volverá a preguntar hasta que ingrese un número valido.
6. Cuando uno de los dos jugadores gane, aparece una opción de reiniciar el juego. Ingrese s si desea reiniciar o n si desea terminar el juego.

División del archivo:

- **main:** Es el punto de entrada del programa. Este inicializa el juego llamando a `initTable` y llama las funciones correspondientes para crear el tablero. Además, junto con `Game` realiza la lógica principal del juego, donde pide la entrada al usuario, coloca la ficha, cambia de usuario y revisa el ganador.
- **initTable:** Inicializa el tablero y deja el espacio con 0's.
- **playerMovement:** Maneja el movimiento del jugador. Verifica si la celda seleccionada está vacía antes de realizar el movimiento. Si la celda está vacía, guarda el número del jugador en la celda correspondiente del tablero. Además, revisa si la entrada está dentro de los limites (0-8) si no vuelve al preguntar.

- **winner:** Verifica si hay un ganador. Verifica si algún jugador ha ganado comprobando las filas, columnas y diagonales. Si encuentra tres celdas consecutivas con el mismo valor, declara al ganador. Además, revisa si hay empate y pregunta si desea reiniciar el juego.
- **userInput:** Solicita y procesa la entrada del usuario. Imprime mensajes solicitando la entrada del usuario y lee el número ingresado. Actualiza el registro \$t2 con el valor leído.
- **changePlayer:** Alterna entre los jugadores. Cambia el jugador actual alternando entre el jugador 1 y el jugador 2. Actualiza el color y el número del jugador actual.
- **selected:** Selecciona la celda correspondiente al movimiento. Calcula la posición en el display para la celda seleccionada por el jugador. Ajusta las coordenadas para dibujar el movimiento en la pantalla.
- **drawMovement:** Dibuja el movimiento en la pantalla. Escribe el color correspondiente en la posición calculada del display.
- **createVerticalLine:** Dibuja líneas verticales en el tablero. Dibuja una línea vertical en la posición especificada del display. Repite el proceso hasta completar la línea.
- **createHorizontalLine:** Dibuja líneas horizontales en el tablero. Dibuja una línea horizontal en la posición especificada del display. Repite el proceso hasta completar la línea.
- **replayTable:** Reinicia display. Cambia los valores de display a 0's.
- **exit:** Finaliza la ejecución del programa.

Errores:

- No imprime user, pero si imprime lo que sigue.

Mejoras:

- Que imprima user.

Resultados:

Se logra lo que se propuso. Se puede jugar el 3 en raya o TikTakTok de forma completa y funcional, se muestran las fichas en el bitmap. Se puede identificar cuando gana y en tal caso reiniciar el juego.

Conclusiones:

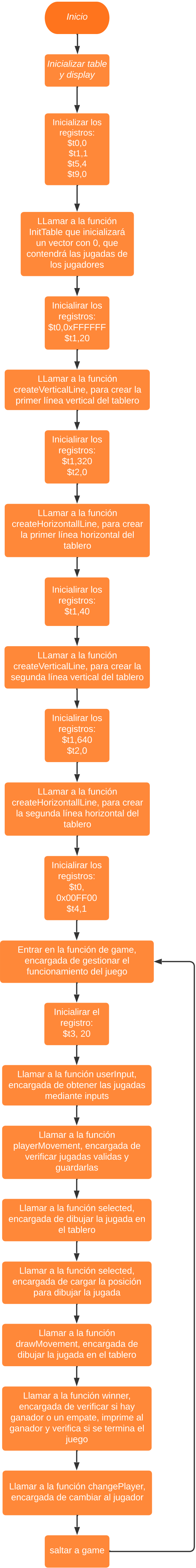
Este proyecto fomento en nosotros el conocimiento y el uso de lenguaje ensamblador, en MIPS. Nos impulsó a investigar las formas de interfaces graficas y su uso en MARS. Y consecuentemente estos aprendizajes nos ayudara a mejorar nuestras capacidades de programación y razonamiento para el futuro.

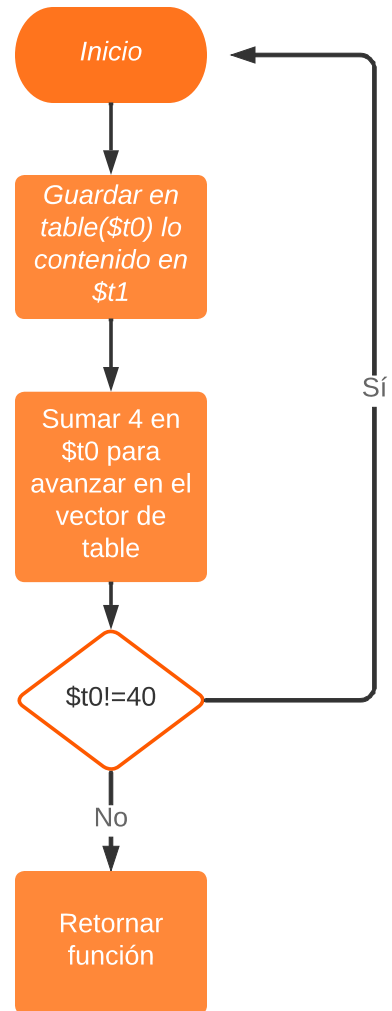
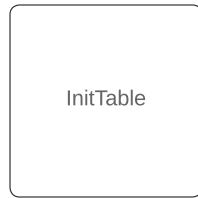
Autores:

Axel Rojas Retana C36944

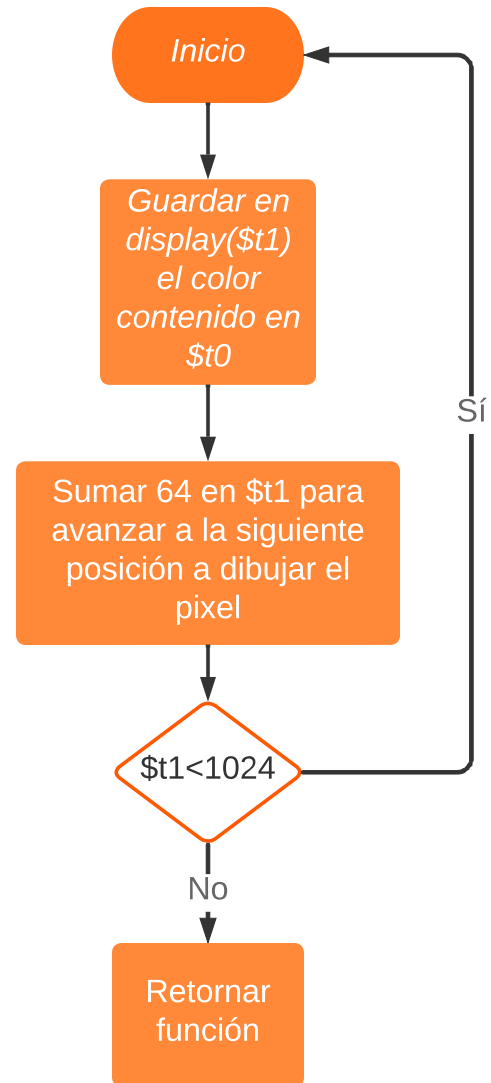
José Pablo Brenes Coto C31289

Diagramas:

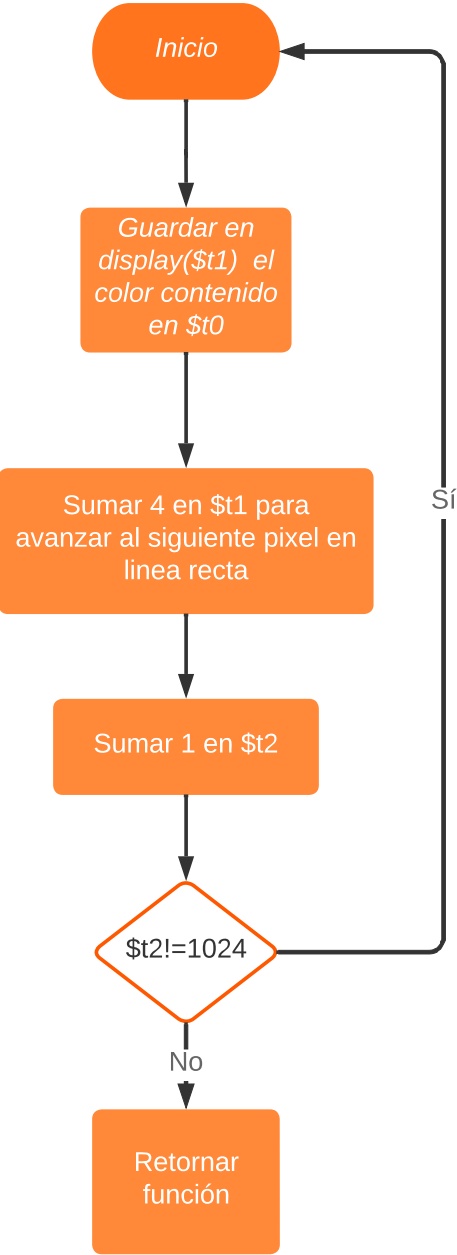




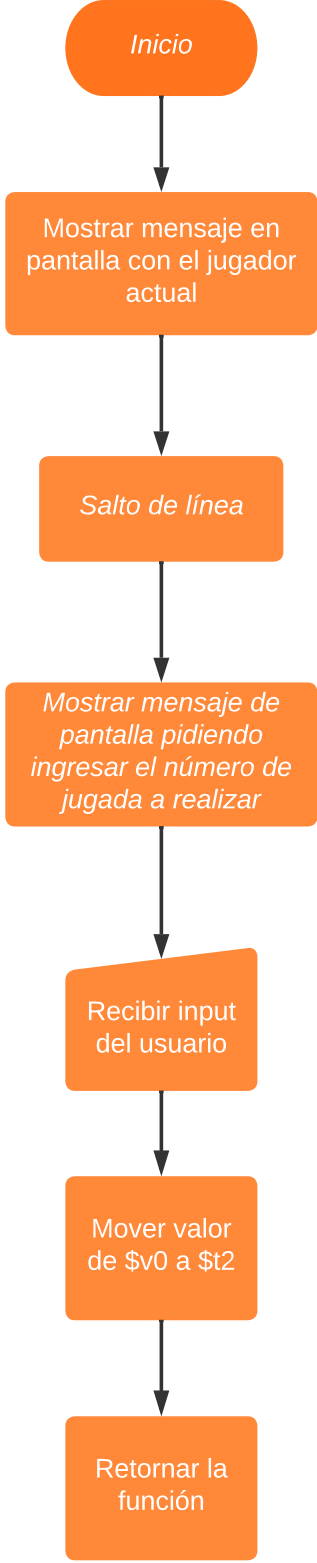
CreateVerticalLine

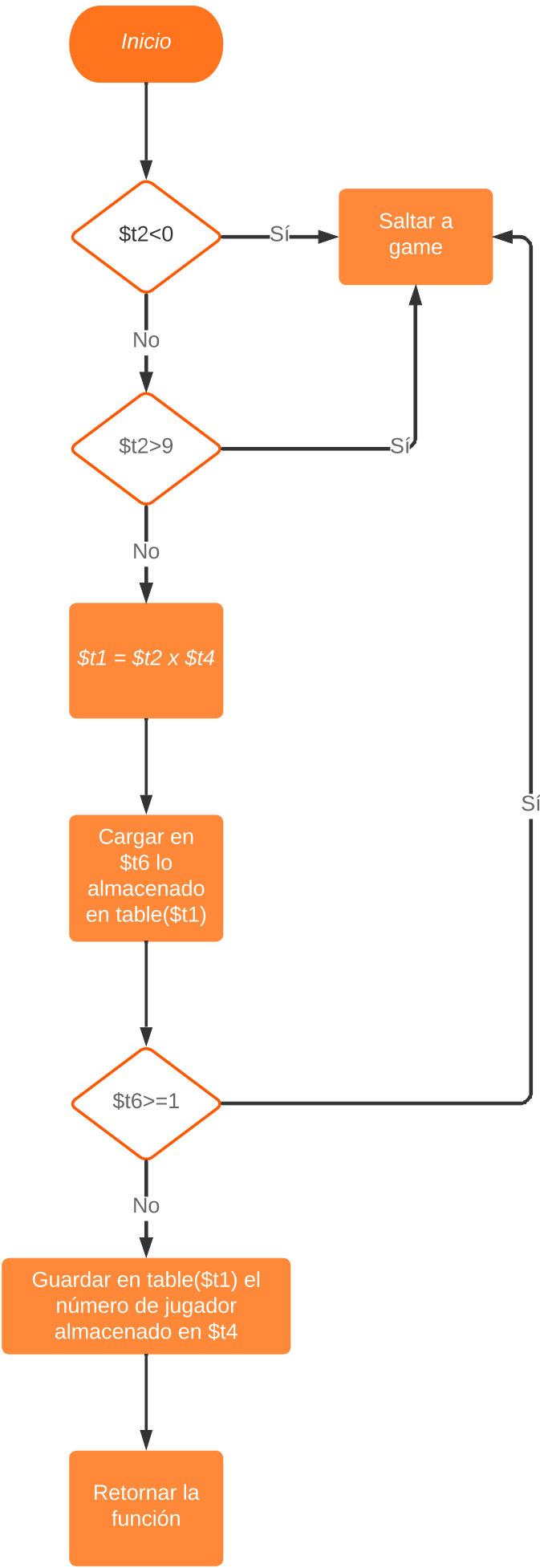


CreateHorizontalLine

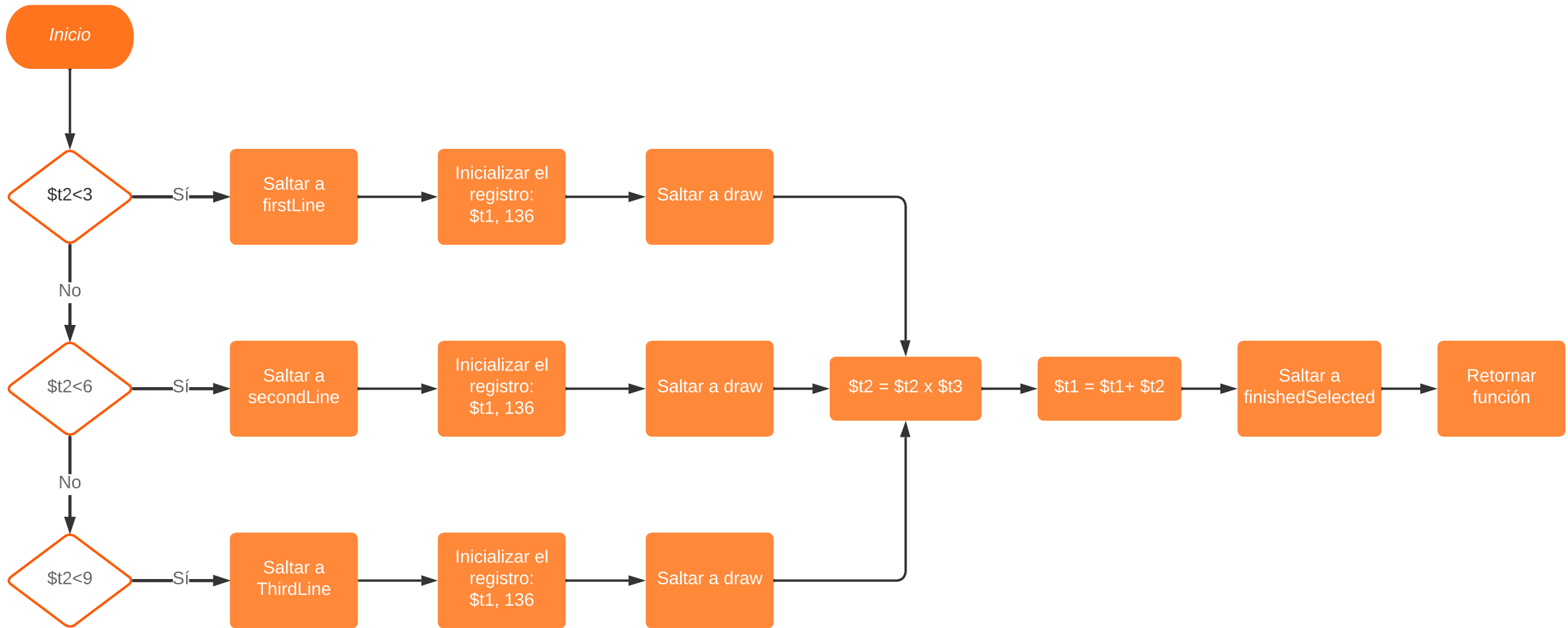


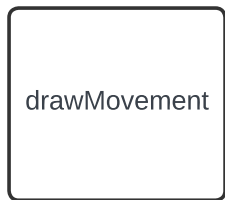
userInput

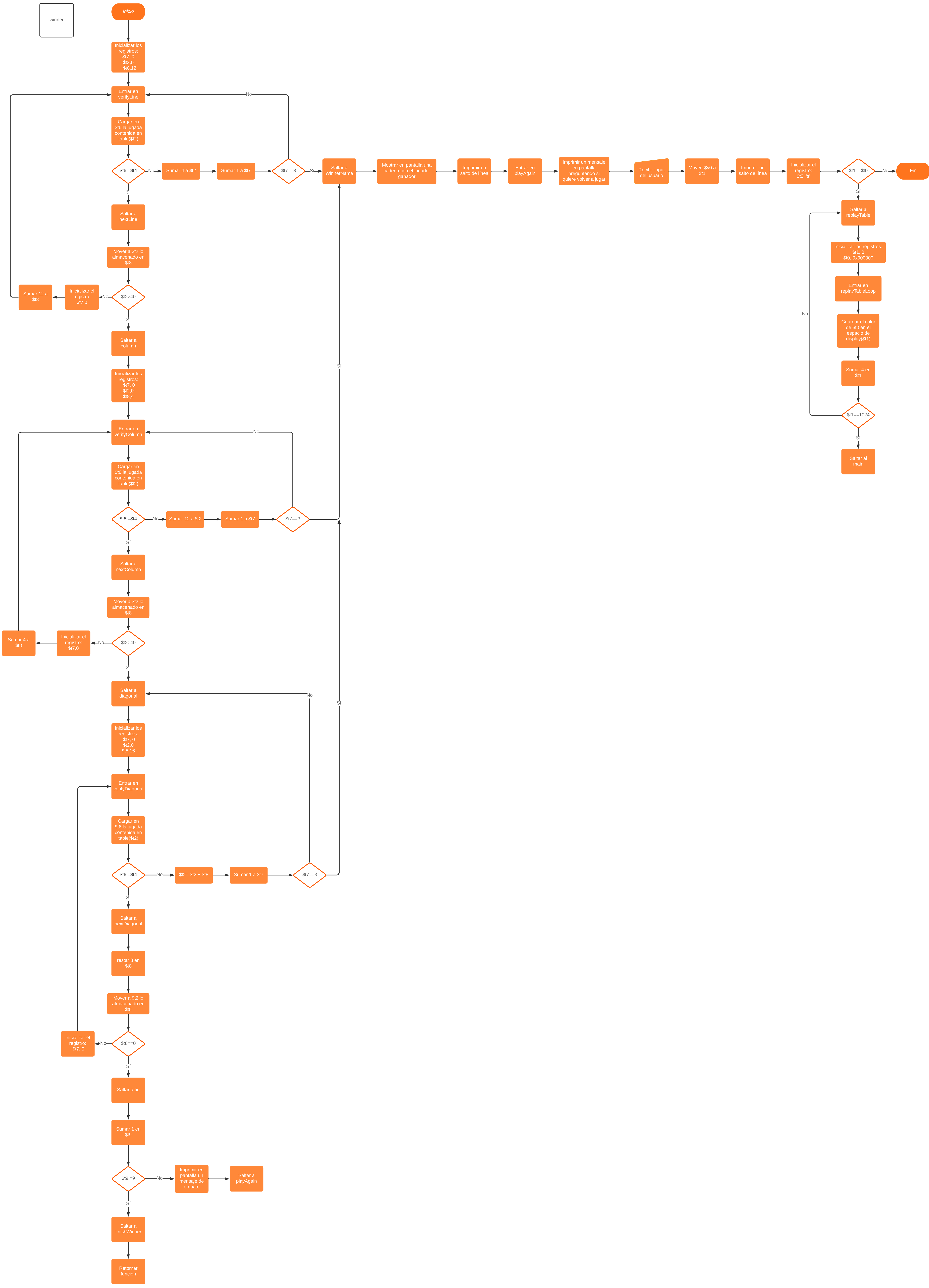




Selected







changePlayer

