

Proyecto 3: Rabbit Chase

Organización de Computadoras y Assembler

Pablo Sao - 11530

Mirka Monzón - 18139

Instrucciones del juego

El objetivo del juego es atrapar un “conejo” antes que escape del tablero de 5x5. El jugador (H) selecciona como punto de partida cualquier posición de inicio de las filas o columnas externas. El conejo (B) aparece en el centro del tablero. Para su siguiente movimiento, el jugador puede elegir únicamente casillas vecinas a su última posición. En su siguiente movimiento, el conejo elige de forma aleatoria cualquiera de los 9 cuadros vecinos y decide una dirección. En el tercer movimiento, el conejo solamente podrá elegir una de las 3 casillas vecinas en la dirección elegida. El jugador gana si elige la misma casilla elegida por el conejo en el siguiente movimiento (X). Se acaba el juego si el conejo sale del tablero (B!) alcanzando cualquiera de las posiciones en las filas o columnas externas.

EXTRA: aumentar el tamaño del tablero y permitir que el conejo se mueva en cualquier dirección.

		H		
		B		

Primer movimiento de jugador (H) y conejo (B)

	H			
	B<			

Segundo movimiento de jugador (H) y conejo (B)

X				

Conejo atrapado

H				
B!				

Conejo escapó

Algoritmo

Secuencia Instrucción

- 1 Se inicia el juego
- 2 Escribe matriz vacía, con el jugador en el centro de la fila más alejada del tablero.
- 3 Jugador se posiciona en la casilla donde desea iniciar.
- 4 Se valida que solo seleccione las posiciones en la parte exterior del tablero (columnas y filas más alejadas del centro).
- 5 Se espera por parte del jugador el ingreso de la tecla para iniciar el juego.
- 6 Al seleccionar la tecla, se posiciona al conejo en el centro del tablero.
- 7 Se muestra en pantalla el tablero.
- 8 Se espera entrada del jugador para seleccionar una casilla vecina.
- 9 Se muestra la pantalla del tablero con el movimiento del jugador.
- 10 Si el jugador selecciona la casilla donde está posicionado el conejo:
 - 10.A se muestra tablero
 - 10.B se indica al jugador que ha ganado.
 - 10.C vuelve a la pantalla del inicio del juego.
- 11 El conejo selecciona una casilla vecina
- 12 Valida si el conejo selecciona la casilla donde se encuentra posicionado el jugador.
- 13 Si el conejo selecciona la casilla donde está posicionado el jugador:
 - 13.A se muestra tablero
 - 13.B se indica al jugador que ha ganado.
 - 13.C vuelve a la pantalla del inicio del juego.
- 14 Si el conejo no selecciona una posición del jugador, realiza movimiento.
- 15 Muestra tablero
- 16 regresa a la secuencia 8.

Recomendaciones

- No se presentó algún impedimento en el desarrollo del proyecto, salvo el comprender como realizar la generación de números aleatorios con assembler, debido a lo que se estuvo investigando, este solo se genera una sola vez, donde es necesario incorporar otros componentes para que siempre se genere un número aleatorio, donde sería de bastante ayuda el poder ver

en una clase como realizar este tipo de archivos, o ponerlo a disposición de los alumnos al momento en el que se asigna el proyecto 3.

- Al revisar los diferentes enunciados de los proyectos, hay unos que solo necesitan utilizar un vector, donde sería de mayor utilidad que los proyectos futuros incluyan un concepto de matriz de información, ya que esto será de bastante ayuda para poder comprender y reforzar temas del lenguaje de programación.
- De igual forma sería de bastante importancia recalcar a los alumnos que tengan en cuenta el concepto de “máquinas de estado” para desarrollar el proyecto.

Referencias

Bandara, N. (2017). *Answer to How to Print colored String in ARM Assembly*. Extraído de: <https://stackoverflow.com/questions/46084673/how-to-print-colored-string-in-arm-assembly>

Barnes, R. (2015). *Answer to List of ANSI Color Scape Sequences*. Extraído de: <https://stackoverflow.com/questions/4842424/list-of-ansi-color-escape-sequences>

Bramley, J. (2019). *Condition Codes 1: Condition Flags and Codes*. Extraído de: <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/condition-codes-1-condition-flags-and-codes>

Manzke, M. (2007). *2 - Dimensional Arrays*. Extraído de: https://www.scss.tcd.ie/Michael.Manzke/1ba3/1BA3_33_hll.pdf

Smith, B. (2014). *Raspberry Pi Assembly Language*. Segunda edición. California, Estados Unidos: CreateSpace. 53, 72-73, 88 - 94 pp.