

Proyecto #3 de diseño: lenguaje ensamblador ARM

I. Competencias a desarrollar:

Diseñar un programa básico en lenguaje ensamblador del procesador ARM que implemente el juego asignado.

II. Condiciones y fechas de entrega:

- El proyecto se realiza en parejas seleccionadas por los estudiantes.
- Entrega y presentación:** durante los períodos asignados durante la SEMANA 20
- Presentación:** por medio de sesión vía Blue Jeans, deben estar presentes todos los integrantes del grupo y demostrar el funcionamiento de su programa.
- Material a entregar en Canvas** (*ver detalles en la evaluación*)
 - Programas fuente (.s)
 - Documento del diseño en formato PDF que incluye todo lo necesario para explicar el trabajo realizado: especificación del uso de los registros, diagrama de flujo del programa hecho en una herramienta de software o algoritmo narrativo, ordenado y con la simbología correcta, conclusiones, bibliografía citada de forma correcta y de al menos 2 sitios confiables.

III. Instrucciones

Su proyecto consiste en elaborar un juego en lenguaje ensamblador de ARM. Deberá permitir jugar a dos personas como mínimo. Uno de los jugadores puede ser la computadora. El programa debe cumplir los requerimientos de funcionamiento, programación defensiva y uso de subrutinas propias.

Se requiere construir una interfaz gráfica (GUI) para interactuar con el usuario utilizando caracteres ASCII, diseñada por los estudiantes. El objetivo principal es que su juego funcione y sea divertido. **Además, se solicita utilizar la técnica de “ASCII art” para realizar alguna gráfica alusiva al juego, que usa caracteres ASCII para dibujar, consulte en los sitios:**

- <http://www.glassgiant.com/ascii/>
- <http://picascii.com/>

IV. Temarios:

1. Battleship

Se juega entre dos jugadores y el objetivo es hundir los barcos del otro jugador. En un tablero de 4x4, cada jugador elegirá la posición donde colocará un barco, seleccionando las posiciones contiguas para este. En un primer prompt, el jugador 1 selecciona de forma consecutiva las 3 posiciones contiguas horizontales o verticales del barco. Luego de seleccionadas, el juego debe desplegar un tablero con el barco (indicado +) en la posición seleccionada. Si el jugador 1 no confirma la posición de su primer barco, se reinicia el proceso. Si confirma la posición, se borra el tablero del jugador 1 y se repite todo el proceso para el jugador 2. En cada turno, se despliega el tablero de juego de cada jugador con el estado exitoso o no de sus elecciones anteriores. El jugador elige 3 posiciones en cualquiera de las casillas de su tablero para tratar de hundir el barco de su oponente. Luego de la selección, se mostrará su tablero con una falla con una “o” y un acierto con una “x” y este se borrará luego de 5 segundos, dejando la pantalla sin tablero y con un mensaje invitando al siguiente jugador. El juego termina cuando un jugador hunde el barco del oponente. EXTRA: usar un tablero con mayor número de casillas y mayor número de barcos de 3 posiciones.

| | | | |
|--|---|--|--|
| | | | |
| | + | | |
| | + | | |
| | + | | |

Tablero con la posición para barco jugador 1

| | | | |
|---|---|---|--|
| | o | o | |
| | | o | |
| | x | | |
| o | | | |

Tablero jugador 2 con sus aciertos y fallas

2. GO

Se juega entre dos jugadores. El objetivo del juego es convertir todas las fichas de un jugador a fichas del otro jugador. En un tablero de 5x5 cada jugador coloca una ficha por turno. A un jugador le corresponde la ficha (O) y a otro la ficha (X). Si un jugador coloca las fichas del otro entre dos fichas suyas, todas las fichas atrapadas se convierten a fichas del jugador. Esto se puede hacer de forma vertical u horizontal. El juego termina cuando no queden más posiciones por jugar y gana el jugador con más fichas en el tablero. EXTRA: extender la función de captura de fichas a posiciones diagonales.

| | | | | |
|---|---|--|--|--|
| X | O | | | |
| | O | | | |
| | X | | | |
| | | | | |
| | | | | |

| | | | | |
|---|---|---|--|--|
| X | X | X | | |
| | O | | | |
| | X | | | |
| | | | | |
| | | | | |

3. Pi-DOTS

Lo juega solamente un jugador a la vez. Se genera aleatoriamente una matriz de caracteres. El jugador debe unir la mayor cantidad de caracteres del mismo tipo con líneas verticales y horizontales, cuyas posiciones se ingresarán por teclado. Cuando conecta los caracteres del mismo tipo, estos desaparecen de las posibilidades de ser elegidos para unir, y se acumula 1 punto por cada letra unida. Si el jugador conecta 20 caracteres, entonces termina esa ronda. Cada error de ingreso para unir puntos descuenta 3 puntos del total acumulado. Para referencia del funcionamiento del programa puede visitar: <http://www.manualpc.com/tutorial-de-two-dots/>

4. Trivia-ARM

Se juega entre dos jugadores. El objetivo del juego es adquirir el personaje en cada categoría, las cuales están divididas: Arte, Literatura, Geografía, Ciencia y Tecnología, Historia, Entretenimiento y Deportes. Para cada categoría se incluye una serie de preguntas de trivia. El personaje correspondiente a cada categoría será ganado luego de responder correctamente tres preguntas de esa misma categoría. La ronda será ganada por el primer jugador que logre adquirir 3 personajes. Se deberán incluir animaciones básicas para representar a los personajes a ganar. Puede dirigirse al siguiente link para tener una referencia del funcionamiento del juego: <https://www.triviacrack.com/>

5. ARM-Ztype

El juego requiere de 1 jugador. Consiste en “destruir” las palabras que son enviadas por la computadora en forma de ticker tape (texto que se desplaza horizontalmente) de texto que se mueven de izquierda a derecha. Al iniciar el juego aparece la primera palabra con su(s) espacios en blanco y cada 5 segundos o luego de que el jugador intente adivinar, aparece una nueva palabra. La destrucción de palabras se realiza ingresando la letra o letras faltantes (máximo 2) en el orden de ausencia en la palabra. Cada palabra está precedida por un numeral entre paréntesis, por lo que el jugador debe indicar primero el número de la palabra a eliminar y luego la cadena de caracteres faltantes, todo en la misma línea y sin espacios en blanco. Al eliminar cualquier palabra, se debe recalcular el orden de las demás dentro de la secuencia, para poder desplegarla al jugador. El juego acaba cuando cualquier palabra no adivinada llega al lado derecho de la pantalla. EXTRA: El juego debe tener al menos dos niveles de dificultad, los cuales se diferenciarán por la velocidad de avance de las palabras y por el número de letras faltantes. El jugador acumula puntos según el número de letras de la palabra. Para mayor referencia del funcionamiento del juego, dirigirse a: <https://zty.pe/>

| | |
|-------------------------------|--|
| T=-0 s | Debes “destruir” las palabras completándolas. Selecciona el orden de la palabra seguido de sus caracteres faltantes: 1ER Listo para jugar? (Y/N) Raspi:> Y |
| T = 0 s | (1)_rnitorr_nco Raspi:> |
| T = 5 s timer | (2)ba_l (1)_rnitorr_nco Raspi:>2L |
| T= 8 s Intento de jugador | ¡Intenta de nuevo! (3) t_levisi_n (2)ba_l (1)_rnitorr_nco Raspi:>2U |
| T= 11 s Intento de jugador | Correcto: BAUL (2) t_levisi_n (1)_rnitorr_nco Raspi:> |
| T = 16 s timer | (3) z_p_to (2) t_levisi_n (1)_rnitorr_nco Raspi:> |

6. Grupo de 3 personas: 2 fotos 1 palabra (2 Pics 1 Word)

Se juega entre dos jugadores, la computadora no puede ser uno de ellos. Cada jugador contará con tres turnos de forma alternada; en cada turno se muestran en pantalla dos imágenes (diseñadas con ASCII Art) que tienen algún tema en común; debajo de las imágenes habrá un tablero con letras en desorden, para formar una palabra (la palabra está relacionada a las imágenes). El jugador deberá ingresar por teclado la palabra que responda al acertijo; si el jugador responde correctamente, entonces obtiene tres puntos. Gana el jugador que obtenga más puntos al final de cada ronda (una ronda = 6 turnos, 3 de cada jugador).

NOTA: su programa deberá tener un banco de al menos 30 imágenes que serán seleccionadas aleatoriamente.

V. Evaluación

Cada miembro del grupo presentará una parte del proyecto, teniendo una nota individual, la cual servirá para **determinar el porcentaje de conocimiento que tenga de la totalidad del mismo**. *Es decir, si por ejemplo la nota obtenida en el proyecto es de 80 puntos, y el estudiante posee un 50% de conocimiento del código, la nota será de 40 puntos.*

Para poder acreditar la nota del proyecto es necesario: 1) entregar el informe y el programa realizado **en Canvas** y 2) Presentarse el día de la entrega del proyecto en el periodo doble del viernes 26 de abril.

Cualquier indicio de copia entre estudiantes de esta u otra sección, o de cualquier tipo con datos encontrados en internet o proporcionados por otros estudiantes, implica un cero automático en la nota de TODA la actividad, para todos los involucrados.

| Criterios | Nivel 3 Experto 😊 | Nivel 2 Aprendiz 😬 | Nivel 1 Novato 😞 |
|--|---|--|---|
| Funcionamiento del programa 40% | El programa funciona con todos sus requerimientos: ingreso de datos, despliegue de resultados y salida correcta al sistema operativo. 40% | El programa funciona entre el 80% y 60% de sus requerimientos. 30% | El programa funciona en menos del 50% de sus requerimientos. 12% |
| Interfaz con el usuario 20% | La interfaz indica claramente los avances del juego, es clara y amigable, está muy completa y se encuentra diseñada de forma creativa. Utiliza ASCIIart con una figura alusiva al juego 20% | La interfaz Indica parcialmente los avances del juego, es medianamente amigable con el usuario, está incompleta o le falta creatividad. No utiliza ASCIIart. 12% | La interfaz no indica los avances del juego, es poco amigable con el usuario y/o está muy incompleta. No utiliza ASCIIart. 4% |
| Programación defensiva y ayuda del uso del programa 10% | El programa tiene muy buena programación defensiva en todos los ingresos de datos, y proporciona mensajes oportunos ante situaciones inesperadas. El programa da instrucciones de ayuda para utilizar el juego. 10% | El programa tiene programación defensiva en la mayoría de los ingresos de datos, y proporciona algunos mensajes oportunos ante situaciones inesperadas. El programa da una ayuda regular para utilizar el juego. 6% | El programa tiene muy poca o ninguna programación defensiva en los ingresos de datos, y proporciona pocos o ningún mensaje oportuno ante situaciones inesperadas. El programa da muy poca ayuda para utilizar el juego. 2% |
| Documentación y orden del programa 10% | La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes. Los nombres de las variables son significativos. La presentación del programa es muy clara y ordenada, y utiliza una tabulación adecuada. 10% | Falta documentación en el encabezado o en bloques de código Los nombres de las variables son medianamente significativos. La presentación del programa es regularmente clara y ordenada. La tabulación es aceptable. 6% | Falta gran parte de la documentación del código Los nombres de las variables no expresan ningún significado. La presentación del programa es confusa y desordenada. No hay tabulación de las instrucciones. 2% |
| Uso de subrutinas y programación estructurada (no cuentan las subrutinas vistas en clase) 10% | El programa es 100% estructurado con más de dos (2) subrutinas que utilizan el formato ABI. Las subrutinas tienen en el encabezado el nombre del estudiante que las diseñó. 10% | El programa es casi completamente estructurado con mínimo dos (2) subrutinas que utilizan el formato ABI. 6% | El programa no está estructurado o las subrutinas no utilizan el formato ABI. 2% |
| Documento de análisis y diseño 10% | El documento incluye todo lo necesario para explicar el trabajo realizado: especificación del uso de los registros, diagrama de flujo del programa hecho en una herramienta de software o algoritmo narrativo, ordenado y con la simbología correcta, conclusiones, bibliografía citada de forma correcta y de al menos 2 sitios confiables. 10% | El documento incluye entre el 80% y 60% de los aspectos necesarios para explicar el trabajo realizado. 6% | El documento incluye menos del 50% de los aspectos para explicar el trabajo realizado. 2% |