

Universidad ORT – Uruguay

Facultad de Ingeniería



Obligatorio – Ahorcado

Arquitectura en Sistemas Digitales

Alumnos:

Sebastián Effa Gallego (193248) – Ingeniería en Electrónica

Fernando A. Hernández Gobertti (173631) – Ingeniería en Telecomunicaciones

Tutor:

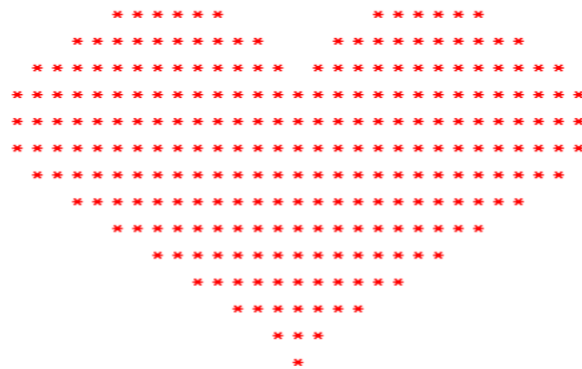
Ismael Garrido Friss de Kereki

Nicolás Levy Linares

Fecha: 23 de junio del 2016

Índice del Obligatorio:

Introducción al Obligatorio	3
Módulo interfaz.asm.....	4
Módulo funcionesGenerales.asm	8
Módulo leerArchivo.asm.....	11
Módulo bitmap.asm.....	12
Módulo dibujarModelosBitmap.asm	16
Módulo main.asm	18
Referencias	19

[illegible]

3

Módulo: interfaz.asm

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               %
%  | | _ _ _ _ _ _ _ _ _ _ | %
%  | | | | | | | | | | | %
%  | | | | | | | | | | | %
%  | | | | | | | | | | | %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

BIENVENIDO/A AL JUEGO DEL AHORCADO!!!

Trata de adivinar la palabra! Pero ten cuidado: Cada error te costara una vida y tendras 4 vidas iniciales. QUE TE DIVIERTAS!

El módulo presente se realizó con el objetivo de contener todo el código referido a la muestra en consola de información, tal sea información para interacción con el usuario referente a mensajes de bienvenida, instrucciones (representados en la figura superior), mensajes de error (o equivocación del usuario), dibujo del ahorcado y la palabra en proceso de ser adivinada (guiones, letras ya ingresadas, letras acertadas). Como se puede apreciar, todos estas funcionalidades forman parte de la nombrada Primer Parte del Obligatorio, consistente en la ejecución correcta del juego por consola del simulador, pero también, se consideró apropiado ingresar en este módulo una función tratante del procesamiento de la Tercer Parte del Obligatorio, esta última referida a brindar la elección al jugador de elegir entre dos teclados (el interno o el Keyboard MMIO Simulator). La consideración radica en el hecho de tener que tener que interactuar con el usuario mediante un cuadro de diálogo (“ConfirmDialog”), que es parte de la interfaz de usuario.

Para la versión final del Obligatorio entregada, este módulo contiene 3 funciones locales y 16 funciones globales, explicadas a continuación.

Nombre de la Función	Rango de Uso	Parámetros (Entrantes)	Objetivo de la Función	Retornos (Salientes)
dibujarModelos	Global	\$a0: Cantidad de Vidas Restantes que posee el usuario.	Compilar los diferentes modelos del ahorcado y dibujar en consola el correspondiente a la cantidad de vidas que el usuario posee en ese momento.	Ningún retorno. La función dibuja o escribe el ahorcado en consola y vuelve al main.

bienvenida	Global	Ningún parámetro.	Escribe en consola el String con el mensaje correspondiente de bienvenida, dado por stringBienvenida.	Ningún retorno. La función escribe en consola y vuelve al main.
instrucciones	Global	Ningún parámetro.	Escribe en consola el String con instrucciones sobre el juego, dado por stringInstrucciones.	Ningún retorno. La función escribe en consola y vuelve al main.
palabraSeleccionada	Global	Ningún parámetro.	Escribe en consola el String con la palabra seleccionada aleatoriamente, brindada por stringSeleccionada.	Ningún retorno. La función escribe en consola y vuelve al main.
mostrarPalabra	Global	Ningún parámetro.	Escribe en consola el String con la palabra a adivinar por el usuario separada por espacios.	Ningún retorno. La función escribe en consola y vuelve al main.
mostrarOpciones	Global	Ningún parámetro.	Escribe en consola el String referido a brindarle al usuario la opción de ingresar un carácter a verificar o el "0" para salir del juego (junto al "1" para debugear la palabra a adivinar). Esta información se encuentra en stringOpciones.	Ningún retorno. La función escribe en consola y vuelve al main.
errorPalabraSeleccionada	Global	Ningún parámetro.	Escribe en consola el String con el mensaje de error de elección de palabra.	Ningún retorno. La función escribe en consola y vuelve al main.
finGanador	Global	Ningún parámetro.	Escribe en consola el String con el mensaje final de victoria al jugador por haber	Ningún retorno. La función escribe en

			adivinado la palabra, mensaje brindado por stringFelicitaciones.	consola y vuelve al main.
finPerdedor	Global	Ningún parámetro.	Escribe en consola un String con el mensaje final de derrota al no haber el jugador adivinado la palabra deseada, mensaje brindado por stringGameOver.	Ningún retorno. La función escribe en consola y vuelve al main.
mostrarIngresoIncorrecto	Global	Ningún parámetro.	Escribe en consola un String con un mensaje de error debido a un inadecuado ingreso de carácter por el usuario (como un número diferente a 0 o 1, tildes, simbología en general diferente a letras) dado por stringIngresoIncorrecto.	Ningún retorno. La función escribe en consola y vuelve al main.
mostrarOpcionesTeclado	Global	\$a0: Dirección en memoria del String a desplegar en el cuadro de diálogo.	Mostrar en forma de cuadro de diálogo las posibilidades que posee el usuario en cuanto a qué teclado utilizar (teclado interno o Keyboard MMIO Simulator).	\$v0: Opción elegida por el usuario.
finGracias	Global	Ningún parámetro.	Escribe en consola el String con un mensaje de agradecimiento por haber testado y probado el producto, dado por stringGracias.	Ningún retorno. La función escribe en consola y vuelve al main.

mostrarPalabraAAdivinarFinal	Global	Ningún parámetro.	Escribe en consola al término del programa un String con un mensaje dado por stringPalabraFinal (indicando lo próximo a desplegar) junto a la palabra a adivinar (habiéndola el jugador descubierto o no).	Ningún retorno. La función escribe en consola y vuelve al main.
dibujarPaloVertical	Local, uso exclusivo en dibujarModelos	Ningún parámetro.	Escribe en consola un String compuesto por espacios antecediendo a una barra vertical (de nombre stringPaloModelo1).	Ningún retorno. La función escribe en consola y vuelve a dibujarModelos.
barraInvertida	Local, uso exclusivo en dibujarModelos	Ningún parámetro.	Escribe en consola el carácter (char) barra invertida. Posteriormente se descubrió una forma más sencilla de realizarlo, ingresando la secuencia "\\", la cual se utilizó para otras ocasiones pero se dejó en este caso a modo de ejemplo alternativo.	Ningún retorno. La función escribe en consola y vuelve a dibujarModelos.
espacio	Local, uso exclusivo en dibujarModelos	Ningún parámetro.	Escribe en consola un String compuesto por espacios (de nombre stringEspacio).	Ningún retorno. La función escribe en consola y vuelve a dibujarModelos.

```

Ingresa una letra para comenzar o ingresa '0' para SALIR DEL PROGRAMA :-(
}
La letra no se encuentra :(

La palabra a adivinar era:
P O R T U G U E S

```

En la última versión arribada y correspondientemente entregada, este módulo no posee ninguna función local y posee 5 funciones globales, analizadas posteriormente.

Nombre de la Función	Rango de Uso	Parámetros (Entrantes)	Objetivo de la Función	Retornos (Salientes)
estudioLetra	Global	<p>\$a0: Letra ingresada.</p> <p>Uso interno del array global palabraAAdivinar.</p>	<p>Comprueba el motivo del carácter ingresado ("0": salir del programa; "1": muestra de palabra a adivinar por el usuario; o cualquier letra)</p> <p>realizando las acciones adecuadas correspondientes, junto a una verificación de ingreso correcto. En caso afirmativo, se transforma el carácter ingresado a mayúscula.</p>	<p>\$v0 = 0 (indicador de fin de juego).</p> <p>\$v0 = 1 (indicador que el carácter ingresado fue "1").</p> <p>\$v0 = 2 (indicador que el carácter ingresado es inválido).</p> <p>\$v0 = 3 (indicador que el carácter ingresado es correcto).</p>
letraEnPalabra	Global	<p>Ningún parámetro.</p> <p>Uso interno de arrays globales palabraAAdivinar, adivinandoPalabra, letrasIngresadas.</p>	<p>Analiza primeramente si el caracter ingresado ya fue ingresado. En caso afirmativo, el programa se dirige al módulo interfaz.asm para indicar al usuario el suceso. En caso negativo, se verifica si se encuentra en la palabra final a adivinar (palabraAAdivinar). Si el jugador ha acertado efectivamente en su elección, se recorre y suplanta el array adivinandoPalabra (inicialmente con los guiones bajos) con el caracter ingresado en la posición (o posiciones) correspondientes. Finalmente se guarda el carácter tratado en el array de letrasIngresadas para próximas jugadas.</p>	<p>\$v0 = 0 (el carácter ingresado no se encuentra en la buscada).</p> <p>\$v0 = 1 (el carácter ingresado se encuentra en la palabra final).</p> <p>\$v0 = 2 (el carácter ingresado ya ha sido ingresado previamente).</p>

eleccionPalabraArray	Global	Ningún parámetro. Uso interno de arrays globales palabraAAdivinar, palabrasConCeros.	Genera la elección aleatoria de una palabra del array palabrasConCeros (la cual contiene todas las posibles palabras leídas desde el archivo externo, separadas por 0), la transforma a mayúsculas y la carga en palabraAAdivinar (definiendo el array).	Ningún retorno. La función carga el array palabraAAdivinar con la palabra final a descubrir por el usuario y vuelve al main.
palabraGuiones	Global	Ningún parámetro. Uso interno de arrays globales palabraAAdivinar, adivinandoPalabra.	Define el array adivinandoPalabra en lo respectivo al largo de la palabra (dependiente de la cantidad de caracteres de palabraAAdivinar) cargando inicialmente con guiones bajos (a mostrar en el juego).	Ningún retorno. La función define adivinandoPalabra y vuelve al main.
verificarSiGana	Global	Ningún parámetro. Ningún parámetro. Uso interno del array global adivinandoPalabra.	Comprueba al final de cada turno si el usuario descubrió la palabra, para lo que la cantidad de guiones bajos en el array adivinandoPalabra debe ser cero.	\$v0 = 0 (aún existen guiones en adivinandoPalabra, por lo que el jugador no ha ganado). \$v0 = 1 (no hay guiones en adivinandoPalabra, causando que el jugador gane).

Módulo: leerArchivo.asm

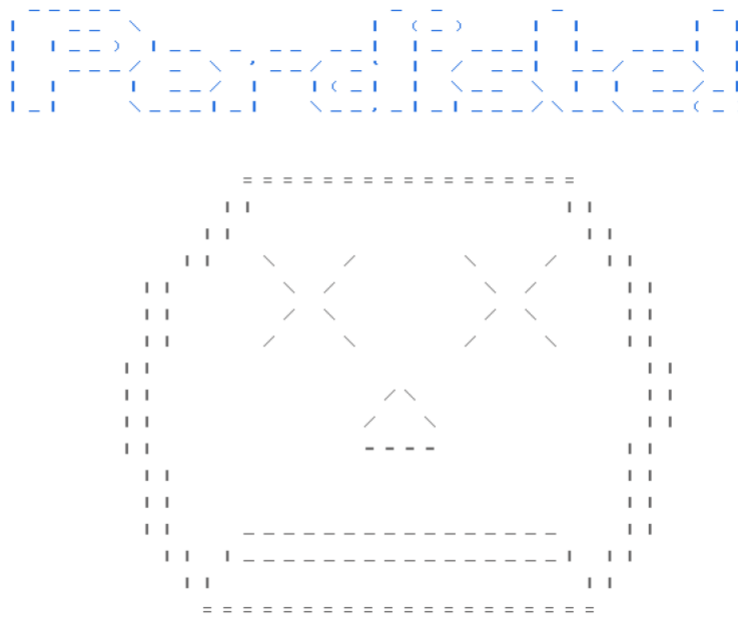
Obteniendo palabra a adivinar...
Exito al cargar la palabra, comienza el juego

El módulo leerArchivo.asm consiste en realizar puramente la Segunda Parte del Obligatorio, es decir, leer la información dentro de un archivo cuya dirección en disco debe ser especificada por el jugador y su posterior tratamiento, referido este a la transformación de la información leída hacia un formato correcto a utilizar por el resto del programa, de tal forma de no alterar lo programado anteriormente. El archivo debe ser, por tanto, de extensión .txt y codificación Unicode (si se corre en Windows, en el caso de macOS se debe guardar como texto sin formato) y debe estar compuesto por una palabra (sin espacios ni tildes o números) por línea tal como es especificado en el manual de usuario anexo al trabajo. Tener además, las precauciones indicadas en el manual de usuario para evitar un corrompimiento del programa.

En el módulo final concluido se puede apreciar que no contiene ninguna función local y una única función global, consistente en aquella que genera el objetivo final de todo el módulo.

Nombre de la Función	Rango de Uso	Parámetros (Entrantes)	Objetivo de la Función	Retornos (Salientes)
leerArchivo	Global	No hay parámetros. Se considera esta función como existente por sí sola, independientemente del programa, considerando su funcionalidad.	Abrir, leer y cerrar un archivo cuya dirección es ingresada por el usuario. Procesamiento de datos leídos para relacionarse armoniosamente con formato trabajado en la Primer Parte del Obligatorio (array palabrasConCeros).	Ningún retorno. La función cumple su cometido, finalmente cargando de caracteres el array palabrasConCeros (a procesar por módulo funcionesGenerales.asm) y vuelve al main.

Módulo: bitmap.asm



En el módulo presente se encuentra la mayoría de las funciones referidas a la Cuarta Parte del Obligatorio, en la que en efecto se solicita la recreación del juego en el display del simulador (bitmap), habiendo sido anteriormente realizado para mostrar en consola. Se remarca el hecho que no la totalidad de las funciones en las que interviene el mapeo del Bitmap son mostradas en este módulo, puesto que el dibujo del ahorcado, debido a su complejidad particular y a fin de no extender en gran medida el módulo actual, se prefirió exponer en el siguiente módulo tratado. Consecuentemente, este módulo se encarga de realizar el paralelismo entre lo mostrado en consolas y lo desplegado en el bitmap, efectuando acciones tales como el dibujo o muestra de una palabra (para lo cual se necesitó la función correspondiente al mapeo de una letra), y la muestra en general de la diferente simbología utilizada en el transcurso del trabajo, tales como el corazón visto en páginas anteriores o la cara apreciada claramente en la figura superior.

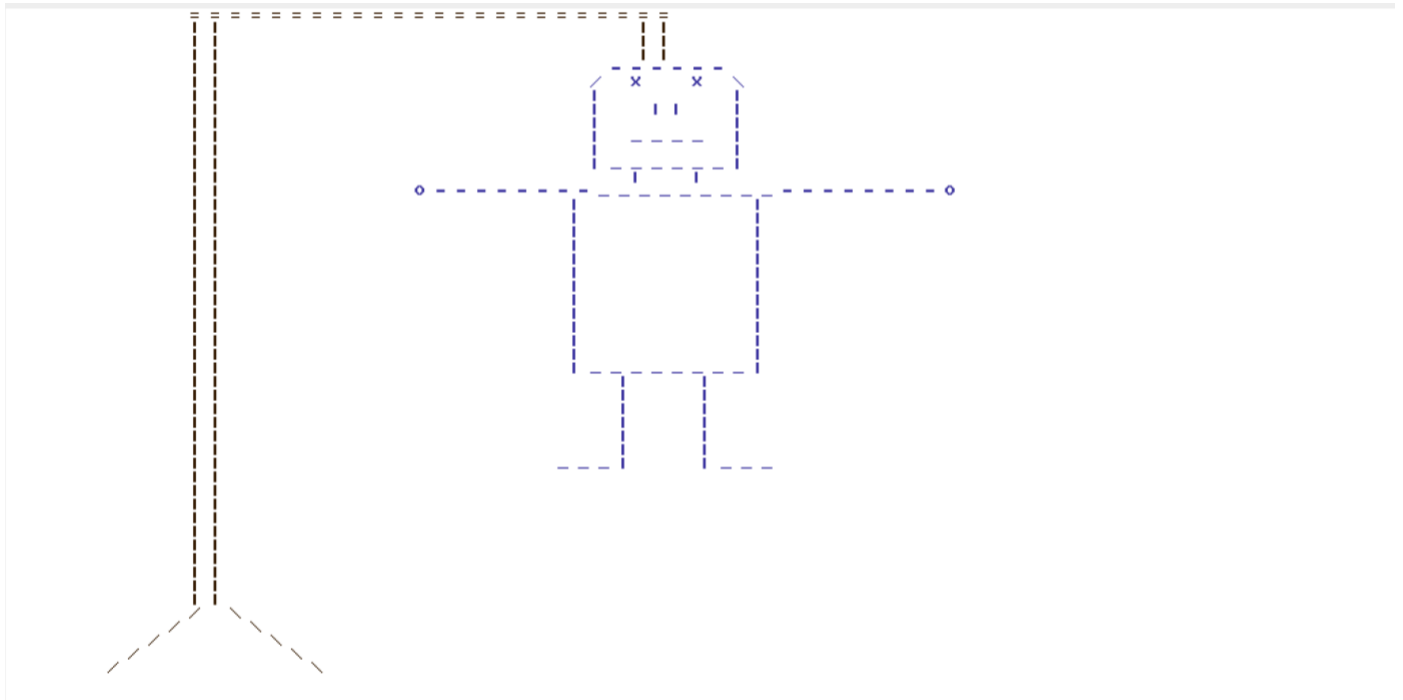
En el módulo último entregado se puede apreciar una función local y 14 funciones globales, explicitadas en la tabla siguiente.

Nombre brindada a la Función	Rango de Uso	Parámetros (Entrantes)	Objetivo de la Función	Retornos (Salientes)
dibujarLetra	Local	<p>\$a0: Dirección de la letra de la font (array de 8 words).</p> <p>\$a1: Fila posición en pantalla.</p> <p>\$a2: Columna posición en pantalla.</p> <p>\$a3: Color en el que se dibujará la letra.</p>	<p>Dibujar en el Bitmap display la letra descrita en la posición y color indicados.</p> <p>(la posición establece el comienzo de la letra, es decir, el extremo superior izquierdo de esta)</p>	Ningún retorno. La función dibuja en el Bitmap.
dibujarPalabra	Global	<p>\$a0: Dirección de la palabra a dibujar (string).</p> <p>\$a1: Fila posición en pantalla.</p> <p>\$a2: Columna posición en pantalla.</p> <p>\$a3: Color en el que se dibujará la letra.</p>	<p>Dibujar en el Bitmap display la palabra o string descrito en la posición y color indicados.</p> <p>(la posición establece el comienzo de la palabra, es decir, el extremo superior izquierdo de esta, tener en cuenta el largo del string para que este quepa en un "renglón" de la pantalla)</p>	Ningún retorno. La función dibuja en el Bitmap.
dibujarAdivinandoPalabra	Global	Sin parámetros.	<p>Dibuja en el Bitmap "adivinandoPalabra", string que contiene la palabra con guiones en proceso de adivinación.</p>	Ningún retorno. La función dibuja en el Bitmap.

dibujarLetrasIngresadas	Global	Sin parámetros.	Dibuja en el Bitmap "letrasIngresadas", string que contiene las letras ya ingresadas por el usuario.	Ningún retorno. La función dibuja en el Bitmap.
dibujarStringLetrasIngresadas	Global	Sin parámetros.	Dibuja en el Bitmap "stringLetras", string que indica las letras ya ingresadas por el usuario.	Ningún retorno. La función dibuja en el Bitmap.
dibujarStringPalabraAAdivinar	Global	Sin parámetros.	Dibuja en el Bitmap "stringPalabraAAdivinar", string que indica la palabra a adivinar por el usuario.	Ningún retorno. La función dibuja en el Bitmap.
dibujarPalabraAAdivinar	Global	Sin parámetros.	Dibuja en el Bitmap "palabraAAdivinar", string que contiene la palabra a adivinar por el usuario.	Ningún retorno. La función dibuja en el Bitmap.
limpiarPantalla	Global	\$a0: Color a aplicar de fondo.	Limpia la pantalla por completo pintando (o sobrescribiendo) todos los píxeles con el color indicado.	Ningún retorno. La función dibuja en el Bitmap.
dibujarGameOver	Global	Sin parámetros.	Dibuja en el Bitmap el string "Perdiste!" en modo ascii art que indica al usuario el fin del juego.	Ningún retorno. La función dibuja en el Bitmap.
dibujarFelicitaciones	Global	Sin parámetros.	Dibuja en el Bitmap el string "Felicitaciones!" en modo ascii art que indica al usuario	Ningún retorno. La función dibuja en el

			el fin del juego.	Bitmap.
dibujarHeart	Global	Sin parámetros.	Dibuja en el Bitmap un corazón en modo ascii art junto al string "Felicitaciones!"	Ningún retorno. La función dibuja en el Bitmap.
dibujarBienvenida	Global	Sin parámetros.	Dibuja en el Bitmap un cartel en modo ascii art que da la bienvenida al usuario	Ningún retorno. La función dibuja en el Bitmap.
dibujarInstrucciones	Global	Sin parámetros.	Dibuja en el Bitmap varios string que indican unas breves instrucciones para el usuario junto al cartel de bienvenida.	Ningún retorno. La función dibuja en el Bitmap.
dibujarDead	Global	Sin parámetros.	Dibuja en el Bitmap una cara de ahorcado en modo ascii art junto al string "Perdiste!"	Ningún retorno. La función dibuja en el Bitmap.
dibujarGracias	Global	Sin parámetros.	Dibuja en el Bitmap el string "Gracias por jugar!" en modo ascii art que indica al usuario el fin del juego ya sea habiendo ganado o perdido.	Ningún retorno. La función dibuja en el Bitmap.

Módulo: dibujarModelosBitmap.asm



El módulo `dibujarModelosBitmap.asm` trata de la realización exclusiva del dibujo del ahorcado en el display Bitmap del simulador, continuando y finalizando así la realización de la Cuarta Parte del Obligatorio. Con tal efecto, para lo cual se utilizó de manera extensiva la función `dibujarPalabra` presente en el módulo `bitmap.asm` previamente analizado. Se procuró nuevamente reutilizar la lógica creada para la representación del dibujo referido en la Primer Parte del Obligatorio, explicitada también en secciones anteriores y apreciable en el código brindado a la disposición del lector.

En este último módulo analizado se puede apreciar una cantidad de 368 líneas de código (considerando nuevamente las líneas de código vacías), 2 funciones locales y una única función global, consistente en aquella que genera el objetivo final de todo el módulo.

Nombre de la Función	Rango de Uso	Parámetros (Entrantes)	Objetivo de la Función	Retornos (Salientes)
dibujarModelosBitmap	Global	Registro \$a0: Cantidad de vidas que posee el usuario.	Compilar los diferentes modelos del ahorcado y dibujar o mapear en el Bitmap el correspondiente acorde a la cantidad de vidas que el usuario posee en ese momento.	Ningún retorno. La función dibuja o escribe en el Bitmap del simulador y vuelve al main.
elegirColor	Local, uso exclusivo en dibujarModelosBitmap	Cantidad de vidas que posee el usuario.	Dependiendo de la cantidad de vidas restantes del usuario, se define el color con el cual dibujar al ahorcado.	Ningún retorno. La función define el color y vuelve a dibujarModelosBitmap.
parametrosModelo1	Local, uso exclusivo en dibujarModelosBitmap	Sin parámetros.	Define el color y parámetros para la muestra en Bitmap Display del palo vertical del ahorcado.	Ningún retorno. La función dibuja o escribe en el Bitmap del simulador y vuelve a dibujarModelosBitmap.

Módulo: main.asm

El main.asm es el módulo en el que se agrupan todos los demás en forma de funciones para dar forma y vida al juego, asociando los parámetros y retornos de estas entre sí y a las variables locales utilizadas.

Consiste en la concatenación de todas las llamadas a las funciones necesarias para ejecutar el juego paso a paso: dar unas breves instrucciones iniciales, leer el archivo de palabras (si es posible, ver manual), tomar una aleatoriamente, mostrar diálogo para elección de teclado, dibujar ahorcado, pedir ingreso de letras con su consecuente tratamiento, verificar luego del ingreso de cada letra si el jugador gana o pierde o simplemente puede continuar ingresando letras, en tal caso se repite el código desde el dibujo del ahorcado; y finalmente en el caso de que el jugador gane o pierda transmitirle tal información y dar por finalizado el juego.

A medida que se realizaban las partes requeridas por la letra y los módulos que estas conllevan, su implementación en el main (juego principal) era tan sencilla como incluir, intercalando, llamadas a funciones de los nuevos módulos entre las llamadas ya existentes y algún detalle extra como en el caso de la implementación de la lectura de archivo que hubo que cambiar el array de palabras predefinido por un espacio libre que la nueva función de lectura de archivo llenará con las palabras leídas; o en el caso del teclado MMIO, que con una simple y única implementación, no fue necesaria la creación de un nuevo módulo y se optó por escribir las pocas líneas de código requeridas directamente en el main.

Este módulo por ser el principal y no tener grandes y/o complejas responsabilidades no posee ninguna función, únicamente es destacable la etiqueta “main” que indica al compilador por donde comenzar a ensamblar el programa.

Referencias:

Mars (MIPS Assembler and Runtime Simulator):

Página principal: <http://courses.missouristate.edu/KenVollmar/MARS/>

Creador: Missouri State University (<http://www.missouristate.edu>)

Actualizada: 14 de Abril del 2016

Leída: 4 de mayo del 2016

Wilkinson:

Keyboard MMIO Simulator: <http://wilkinsonj.people.cofc.edu/mmio.html>

Realizada o Actualizada: No hay datos

Leída: 10 de mayo del 2016

Patorjk:

Página principal: <http://patorjk.com>

Generador ASCII Text Art: <http://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Something%20>

Realizada o Actualizada: No hay datos

Leída: 20 de mayo del 2016

RapidTables:

Página principal: <http://www.rapidtables.com>

Paleta RGB: http://www.rapidtables.com/web/color/RGB_Color.htm#feedback

Realizada o Actualizada: No hay datos

Leída: 30 de mayo del 2016

Stack Overflow:

Página principal: <http://stackoverflow.com>

Bitmap Display de Mars: <http://stackoverflow.com/questions/28066572/bitmap-display-on-mars-simulator>

Realizada: 21 de Enero del 2015

Leída: 30 de mayo del 2016

Code Project:

Página principal: <http://www.codeproject.com>

Ejemplo de la Viborita: <http://www.codeproject.com/Questions/1001151/drawing-a-x-unit-in-BitMap-Display-MIPS>

Realizada: 27 de Junio del 2015

Leída: 30 de mayo del 2016

Github:

Página principal: <https://github.com>

Font 8x8 (indicada en la letra del trabajo): https://github.com/dhepper/font8x8/blob/master/font8x8_basic.h

Realizada: 20 de Febrero del 2012

Leída: 4 de junio del 2016