



Universidad Simón Bolívar
Depto. de Computación y T.I.
Organización del Computador (CI-3815)
Septiembre-Diciembre 2021

Proyecto # 2 (14%)

Manejo de interrupciones

Objetivo

Familiarizarse con los principios básicos de interrupciones y de operaciones de E/S (entrada/salida) asignada a memoria.

Objetivos Específicos

1. Implementar manejadores para las interrupciones de teclado (Receiver) y de reloj (Timer).
2. Simular la implementación de una consola mapeada a memoria e
3. Implementar el juego "*PacMan*"

Descripción del juego *PacMan*

Este juego ha existido por mucho tiempo y ha sido implementado para múltiples plataformas. Se define en base a un laberinto de tamaño $N \times M$, éste tendrá paredes al rededor e internamente. Estas paredes limitan los movimientos de los personajes.

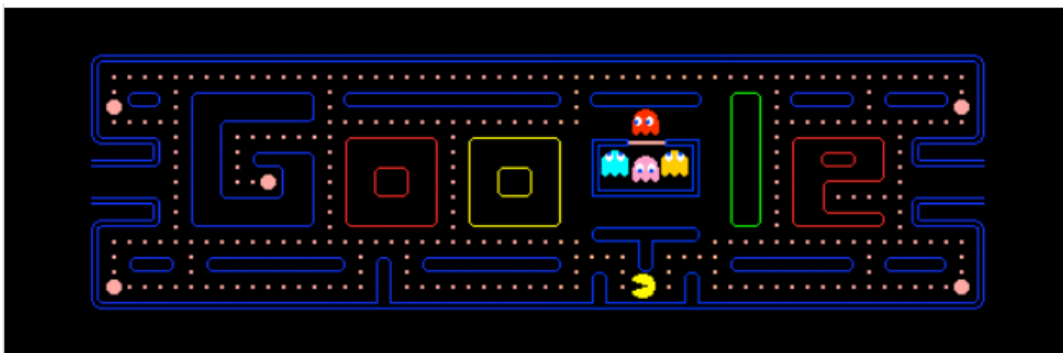


Figura 1

El juego tiene dos tipos de personajes:

1. El héroe: es el personaje el principal y lo llamaremos Pac-Man (color amarillo claro). Su objetivo es comerse los puntos amarillos

2. Los fantasmas o monstruos: son los enemigos. Tradicionalmente se les ha dado un nombre a cada uno: Shadow (Blinky), Speedy (Pinky), Bashful (Inky) y Pokey (Clyde). El objetivo de los enemigos es recorrer el laberinto y comerse a Pac-Man.

El juego termina cuando se consuman todas las vidas del Pac-Man. Una vida se consume cuando se cumple una de dos condiciones:

1. Pac-Man se come todos los puntos o
2. Alguno de los enemigos se come al personaje.

El jugador puede cambiar la dirección en la que se mueve Pack-Man usando las teclas direccionales del teclado (o **Up, Down, Left y Right. No case sensitive**). Los enemigos toman una dirección aleatoria en cada intersección. No pueden invertir la dirección a menos que sea la única opción.

Adicionalmente en el laberinto pueden existir portales: lugares que pueden ser usados por los personajes para trasportarse a otro portal.

En este enlace puede jugar una versión sencilla de este juego: <https://www.google.com/search?q=pacman&oq=pacman>.

Definiciones

Bitmap Display:

Es un dispositivo tipo monitor que puede ser conectado a un computador Mars, ver Figura 2. La interacción con este dispositivo se realiza a través de la memoria (DMA - Direct Memory Access).

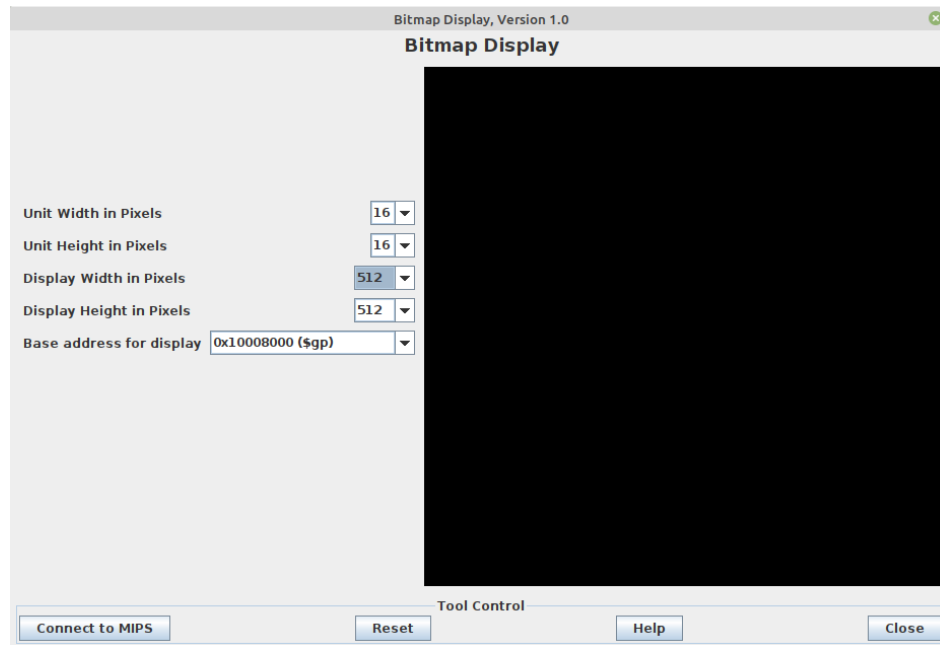
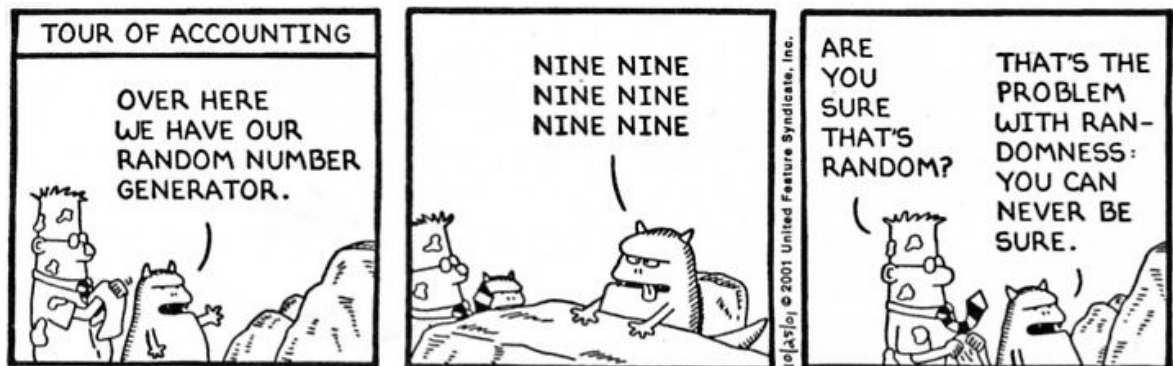


Figura 2

La memoria asignada a este dispositivo es un bloque contiguo de bytes que pueden verse como una matriz de dos dimensiones (llamémosla $\mathbf{MAT}_{n \times m}$). Este dispositivo tiene una resolución de N líneas y M columnas y un refrescamiento automático: una vez cada S segundos. La fila i de la matriz $\mathbf{MAT}_{n \times m}$ corresponderá con la fila i del display, de la misma forma sucede con las columnas. Para detalles sobre funcionamiento, lea la información presentada en la opción *Help*.

Número pseudo-aleatorio:



Un número pseudo-aleatorio es un número generado en un proceso que parece producir números al azar, pero no lo hace realmente. Las secuencias de números *pseudo-aleatorios* no muestran ningún patrón o regularidad aparente desde un punto de vista estadístico, a pesar de haber sido generadas por un algoritmo, en el que las mismas condiciones iniciales producen siempre el mismo resultado.

Los mecanismos de generación de números aleatorios que se utilizan en la mayoría de los sistemas informáticos son en realidad procesos *pseudo-aleatorios*. Muchos de estos sistemas se valen de elementos del sistema de operación o el *hardware*, como el reloj. En otras palabras, un generador de números *pseudo-aleatorios* básico puede hacer uso del valor obtenido del reloj del sistema. Y con éste hacer alguna operación como el módulo.

Enunciado del Proyecto

Ud debe implementar las siguientes actividades:

Actividad 1: La interfaz gráfica

Defina y programe un módulo que realice los movimientos de los personajes: debe ser implementado en su totalidad en el área de texto (.text) del archivo Main.s como una función llamada PacMan (Ver archivo Main.s anexo). Debe usar el dispositivo BitmapDisplay para implementar su tablero. Considere la siguiente definición de tablero en su implementación:

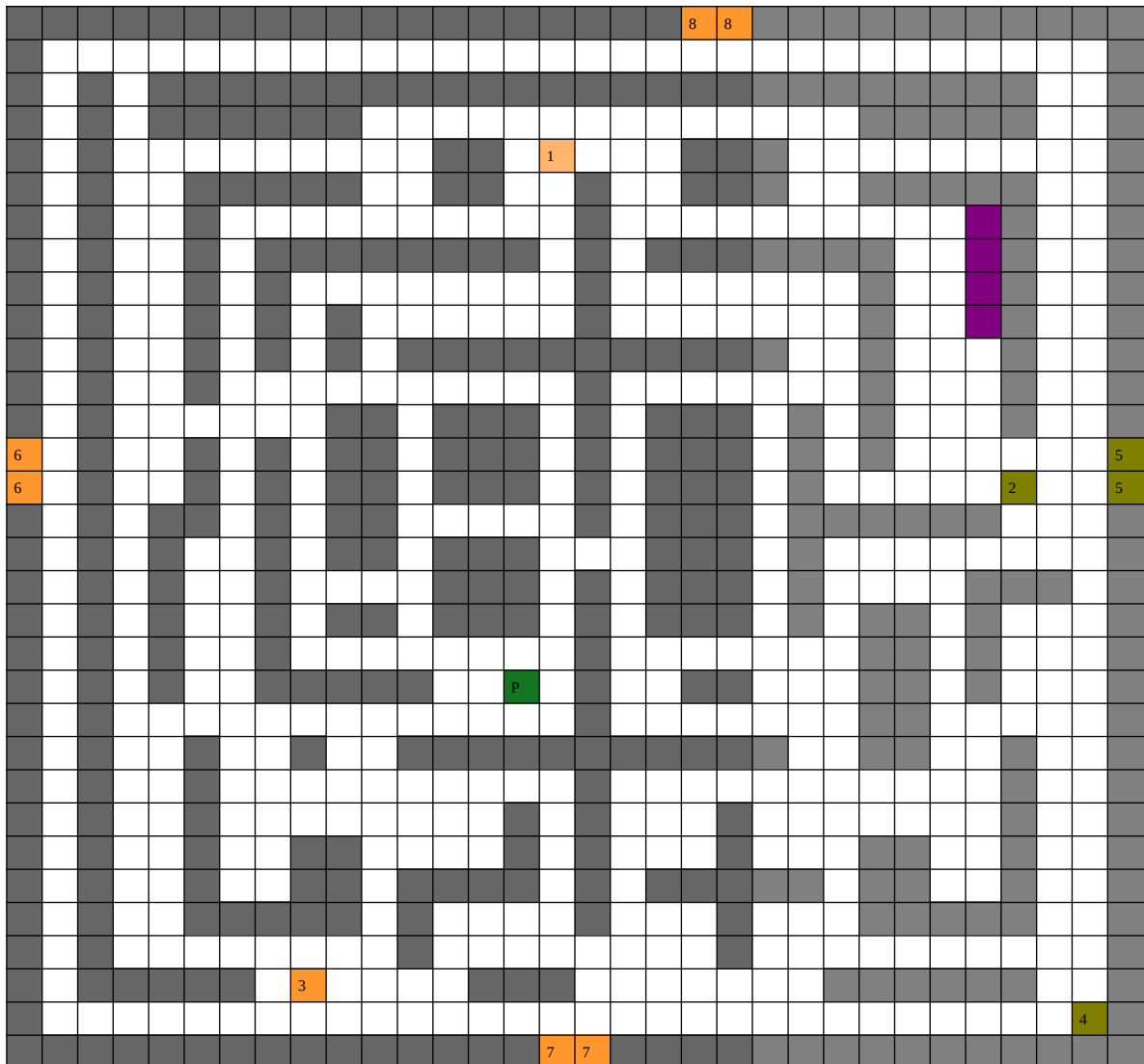


Figura 3: tablero

Donde:

- Los cuadros de color gris representan las paredes.
- Los cuadros en blanco serán el “alimento” del Pac-Man
- El cuadrado verde representa dónde debe aparecer el Pac-Man al comienzo de cada juego.
- Los cuadros magenta representan los puntos de inicio de los enemigos.
- Los cuadros naranja representan los portales.

Utilice los siguientes valores en su implementación:

MAT = 0x10008000

S = 1

C = ?

N = M = 512 / 16

D = 'A'

$$V=3$$

Su implementación debe permitir modificar los valores **MAT**, **D**, **C**, **V** y **S** de forma muy sencilla¹. De requerirlo use la esquina inferior izquierda como las coordenadas (0, 0).

Notas:

1. Utilice la etiqueta **C** como base para la conversión con los tics de reloj. Este valor va a depender del computador usado para correr su proyecto².
2. **V** es el número de vidas que tendrá el Pac-Man,
3. **D** tendrá la dirección actual del Pac-Man. Valores posibles son:
 1. **A**rriba
 2. **A**bajo
 3. **D**erecha
 4. **I**zquierda

Pac-Man se mueve un paso cada S segundos en la dirección indicada por D. Si Pac-Man se come todos los puntos amarillos, el juego recommienza. El juego sólo termina cuando se presiona la tecla 'q' o se consuman todas las vidas de Pac-Man.

Por eficiencia: Sólo se debe escribir (sw) en el Display para realizar cambios. En otras palabras, sólo actualice cuando haya un cambio en la disposición de los personajes y alimentos.

Actividades 2: Manejador de Teclado (Receiver) y de Reloj (Timer)

La implementación del juego requiere los siguientes dos elementos para la atención de los eventos:

- **Manejador de interrupción de teclado:** en el área .ktext del archivo myexception.s. Debe reconocer las teclas direccionales, las letras correspondiente a la definición de la etiqueta D, la 'p' y la 'q'.
- **Manejador de interrupción de reloj:** en el área de .ktext del archivo myexception.s.
 - Note que este manejador es el responsable de indicar cuándo deben moverse los jugadores. Sin embargo, el movimiento es realizado en el Main.

Reglas modo básico:

- En cualquier momento puede presionarse la tecla 'p' para pausar el juego, o continuarlo si ya está pausado.
- En cualquier momento puede presionarse la tecla 'q' para terminar el juego. Debe colocarse 0x00 en la dirección de la etiqueta *seguir*.

No habrá distinción entre mayúsculas y minúsculas.

1 Modificar más de una línea de código, para cambiar uno de esos valores, no es considerado sencillo.
2 Este valor requerirá ser actualizado dependiendo del computador usado para ejecutar el proyecto.

La visualización de los elementos del juego:

- Pac-Man será de color amarillo claro
- Los enemigos serán de color: azul, rojo, verde y marrón.
- Las paredes serán de color Gris oscuro.
- Los portales serán de color naranja

Modalidades:

Cada grupo se le asignará un sub-conjunto de los portales. Envíe un correo a su profesor indicando los integrantes de su equipo para realizar esta asignación.

NOTAS IMPORTANTES:

1. Ud. debe usar la versión de Mars publicada con este enunciado (*MarsTimer.jar*), si tiene algún inconveniente con ésta, debe notificarlo a su profesor de inmediato.
2. No seguir las especificaciones aquí dadas podría resultar en la no corrección de su proyecto.
3. Si tiene preguntas, se les recomienda aclararlas con tiempo.
4. Ud debe implementar los manejadores de interrupción en el archivo `myexception.s`
5. El simulador del dispositivo definido utilizará `el Bitmap Display` ~~la consola de Mars para desplegarse.~~
6. El programa principal que se debe correr puede ser cualquiera, y la labor que éste lleve a cabo debe ejecutarse “normalmente”.

Recomendaciones

1. Comience con tiempo.
2. Estructure bien su código.
3. Trabaje en forma ordenada e incremental
4. Pruebe que cada una de sus funciones funciona correctamente
5. **Respete** la especificación dada en el presente enunciado: No cambie el nombre de las rutinas NI las definiciones aquí establecidas.
6. Tenga presente que es mejor tener más funciones pequeñas que menos funciones largas.

Entrega

El proyecto debe ser entregado **hasta las 8 am del lunes de la semana 12** al correo electrónico `eduardo AT usb.ve` y debe adjuntar un único archivo con nombre `Proy-Apellido1-Apellido2.tar` que contenga **un directorio/carpeta con el nombre `Proy2-Apellido1-Apellido2`**. **Éste debe contener:**

1. **Los fuentes en formato .s:** Estos archivos serán la implementación de su proyecto y se usará *para ejecutarlo*.

2. **Los fuentes en formato .pdf:** Estos archivos serán la implementación de su proyecto y será usado para *evaluar su código*.
3. Los conjuntos de archivos en los puntos 1 y 2 deben corresponder con la misma versión de su proyecto. Alguna discrepancia puede resultar en la **no corrección** de su proyecto.
4. Su código debe estar debidamente documentado y estructurado.

EB/EB