Universidad Simón Bolívar
Departamento de Computación y
Tecnología de la Información
CI3815 - Organización del Computador
Enero-Marzo 2018

Proyecto 2 (10 %)
Interrupciones

Objetivo

El objetivo de este proyecto es que el estudiante adquiera destrezas en la programación de rutinas manejadoras de interrupciones en lenguaje ensamblador MIPS.

Descripción del juego

Breakout es un juego por computadora en el cual el usuario hace mover una barra horizontal en la parte inferior de la pantalla. Una pelota se desplaza por el área vacía de la pantalla y rebota contra las paredes y la barra. En la parte superior de la pantalla hay varias filas de ladrillos, uno de estos se rompe cuando la pelota choca con él. Un ejemplo del aspecto en pantalla que puede tener el juego se muestra a continuación.

Figura 1: Captura de pantalla de una implementación de Breakout

El objetivo del juego es romper todos los ladrillos. Si la pelota se sale por la parte inferior de la pantalla sin haber sido alcanzada por la barra, se pierde una vida.

Especificación

En este proyecto, se requiere que usted implemente el juego Breakout en MARS, mediante la programación de un manejador de interrupciones y la herramienta Bitmap Display que permite simular un dispositivo mapeado a memoria.

Las teclas A y a se utilizarán para mover la barra hacia la izquierda y las teclas D y d la derecha. La tecla Q se utilizará para salir del juego, y la barra espaciadora para pausarlo. Cuando el usuario gane el juego usted deberá mostrar una pantalla adecuada de libre diseño. Cuando el usuario pierda, se debe

mostrar una notificación y, si quedan vidas disponibles, reiniciar el juego cuando el usuario presione cualquier tecla. El número de vidas iniciales será de 3.

Movimiento de la pelota

La pelota se moverá aproximadamente una vez cada T ciclos de reloj (registro count). Este valor podrá variar a lo largo del juego como se especifica más adelante.

El movimiento de la pelota se puede describir mediante dos vectores Vx y Vy que representan la velocidad horizontal y vertical de la pelota, respectivamente. El sentido positivo es hacia abajo y hacia la derecha. En cada movimiento, la pelota se desplazará Vx casillas en sentido horizontal y Vy casillas en sentido vertical.

El valor de Vx respecto a Vy determinará la dirección en que se desplaza a la pelota. Por ejemplo, si Vx = 0 y Vy = -2, la pelota se estará desplazando en sentido Norte. Mientras que si Vx = 1 y Vy = 1, la pelota se moverá hacia el Sureste. En la siguiente tabla se resumen todas las combinaciones admitidas de velocidad para la pelota.

Vx	Vy	Dirección
-1	1	SO
-1	-1	NO
0	-2	N
0	2	S
1	1	SE
1	-1	NE

Al chocar con la barra, la posición de contacto con la barra determinará el ángulo con que saldrá rebotando la pelota. La barra se dividirá en cinco segmentos como se indica a continuación:

Al chocar con el centro de la barra (posición 3) la pelota saldrá en dirección N. Al chocar en la posición 1 o 5 saldrá hacia el NO o NE, respectivamente. Si choca en la posición 2 saldrá con Vx = -1 y Vy = -2, mientras que en la posición cuatro, con Vx = 1 y Vy = -2.

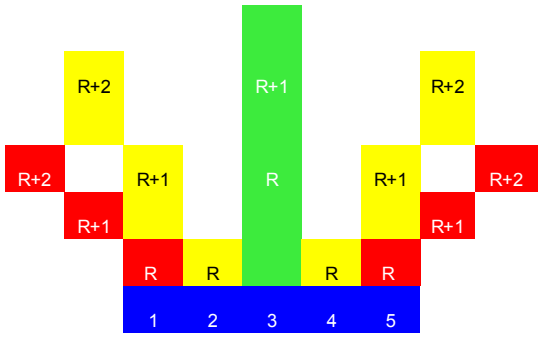

Figura 2: Cambio de movimiento de la pelota al chocar

Imagine la cuadrícula superior como la consola. Cada celda representa una unidad de desplazamiento. En azul está la barra y sus posibles 5 posiciones de impacto de la pelota. En rojo la trayectoria que seguirá la pelota de impactar en 1 ó 5 en el instante de tiempo R y R+1, en verde la representación del rebote para el caso de impacto en 3 y en amarillo para el caso 2 y 4.

Al inicio de la partida la pelota se moverá desde el centro del tablero en dirección aleatoria (Ver syscalls 40, 41 y 42).

Cuando la pelota choca con una pared, esta ejercerá una fuerza Normal perpendicular a ella. Esto implica que la componente de velocidad paralela a la pared no se verá afectada, mientras que la componente de velocidad perpendicular a la pared invertirá su sentido y mantendrá su magnitud. Por ejemplo, si la pelota tiene velocidad Vx = -1 y Vy = 2, después del chocar con la pared izquierda su velocidad será Vx = 1 y Vy = 2.

El mismo principio se aplica cuando la pelota choca con un ladrillo. Adicionalmente, en este caso, el ladrillo es destruido y T será incrementado en 1.

Detalles de Implementación

Para la visualización del juego, se hará uso de la herramienta *Bitmap Display* disponible en Mars. Esta herramienta simula una pantalla con mapa de bits en donde cada palabra del espacio de direcciones específicado corresponde a un pixel de la pantalla. Los pixeles están ordenados por fila comenzando en la esquina superior izquierda. La herramienta permite establecer el tamaño en pixeles de la pantalla a simular y la unidad de los pixeles. El valor que se almacene en esa palabra es interpretado como un color RGB de 24-bits con el componente Rojo en los bits 16-23, el verde en los bits 8-15 y el azul en los bits 0-7 (p.ejm. el color rojo equivale al patrón 0xFF0000). Cada vez que se escribe una palabra en el espacio de direcciones correspondiente al *display*, la posición en la pantalla será renderizado con el color que su valor representa.

La pantalla de juego se representará como una matriz de pixeles de ancho M y alto N, estos valores serán fijos, Ver Figura 3. Note que existen 4 filas de ladrillos, cada ladrillo debe ser de color diferente a sus vecinos (horizontales y verticales). En diferentes tonos de azul se muestra la barra, cada color representa las secciones descritas. Observe los valores que se deben usar para configurar las dimensiones de la herramienta Bitmap Display.

Ud debe definir la etiqueta global T con valor inicial 10, el usuario puede incrementar este valor con la tecla u o U y decrementarlo con la tecla l o L . El valor del incremento/decremento se tomará de la etiqueta global INCREMENTO y su valor inicial será 100. Note que esto permitirá ajustar el comportamiento del juego en base a la velocidad de la máquina donde sea ejecutado.

Nota:
Para este proyecto Ud debe usar la implementación de Mars publicada en la siguiente dirección: https://ldc.usb.ve/~eduardo/Cursos/CI3815/MarsTimer/MarsTimer.jar

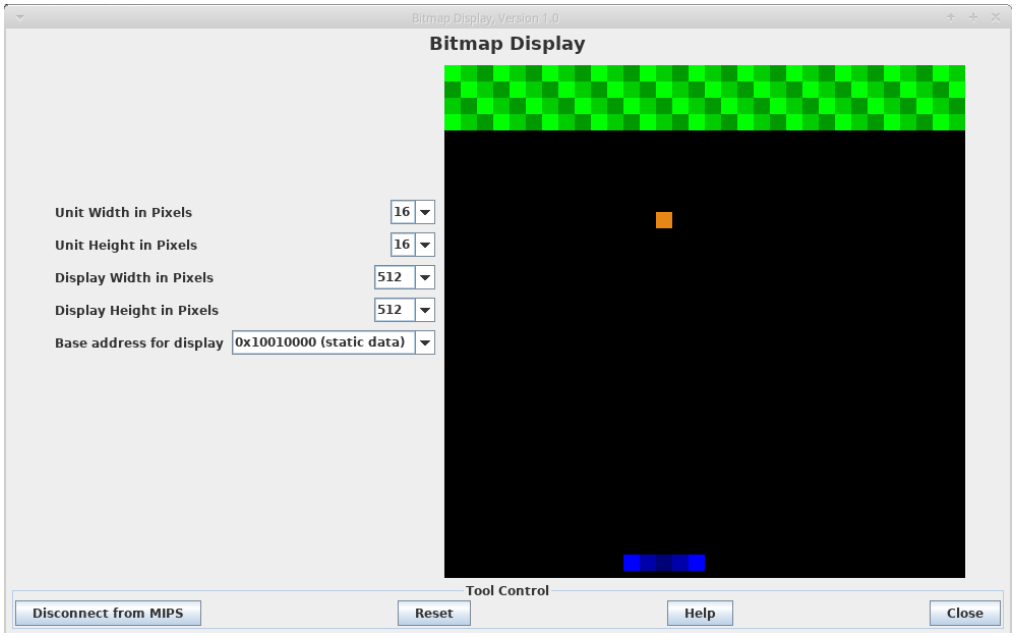

Figura 3: Configuración inicial del juego.

Para la implementación de su juego deberá trabajar sobre dos archivos:

- breakout.asm en donde se incluye gran parte de la lógica del juego
- Exception.asm manejador de excepciones que realiza el tratamiento adecuado de las interrupciones de reloj y teclado.

Fecha de entrega: Jueves 22 de marzo hasta las 11:55 pm a través de la plataforma Moodle.

Referencia

http://www.aeonity.com/ab/games/arcade-classics/classic-breakout.php