

CI3815 – Curso de Organización del Computador

Proyecto # 2 (Trimestre Septiembre-Diciembre 2012)

Objetivo

El objetivo de este Proyecto es que el estudiante adquiera un conjunto de destrezas básicas de la programación en lenguaje ensamblador MIPS, y que comprenda y maneje los mecanismos de generación y manejo de interrupciones y excepciones que le permitan tratar adecuadamente la ocurrencia de eventos asíncronos a su programa.

Actividades

Pac-Man es uno de los videojuegos *arcade* más famosos de todos los tiempos, diseñado por Tōru Iwatani y lanzado en 1980.

En este proyecto, se requiere que usted diseñe e implemente una versión simplificada del juego Pac-Man, que se presenta a continuación.

Descripción del juego

En la versión de Pac-Man que usted debe desarrollar, Pac-Man recorre un laberinto formado por una cuadrícula. Cada casilla de esta cuadrícula contiene un punto que Pac-Man debe comerse, o bien una pared. Las casillas pueden también contener cerezas, que Pac-Man debe comerse para ganar puntaje.

El objetivo de Pac-Man es comerse todos los puntos y cerezas del laberinto. Al lograr esto, Pac-Man avanza al próximo nivel.

Pac-Man tiene tres enemigos que habitan el laberinto, llamados Blinky, Pinky e Inky. Pac-Man pierde una vida cuando es tocado por un enemigo. Al perder tres vidas, termina el juego.

A continuación se describe en detalle cada uno de los elementos del juego.

Laberinto



Figura 1: Pantalla del juego Pac-Man original

```
XXXXXXXXXXXXXXXXXXXXX
Xaaaaaaaaaaaaaaaaa$aX
XaXXXXXaaaaaaaaXXXXXaX
XaaXaa*aaaaaaaaaaaXaaX
XaaXaaaaaaXXaaaaaaXaaX
XaaXa$aaaaaXXaaaaaaXaaX
XaaXaaaaXXXXXXaaaaXaaX
XaaXaaaaaaXXaaaaaaXaaX
XaaXaaaaaaXXaaa*aaXaaX
XaaXa*aaaaaaaaaaaaXaaX
XaaXaaaaaaaooooooooXaaX
XaXXXXXaaaaoXXXXXXaaX
Xa*aa$aaaaaooooo<aa*aX
XaaaaaaaaaaaaaaaaaaaaX
XXXXXXXXXXXXXXXXXXXXX
```

Figura 2: Ejemplo de archivo pac.txt

El juego tiene varios niveles, cada uno con su propio laberinto o mapa. El programa lee por defecto los mapas de todos los niveles de un archivo llamado `pac.txt`. Este archivo contiene el mapa de cada uno de los niveles, con una línea en blanco entre cada mapa. Un ejemplo de este archivo se muestra en la figura 2.

En el mapa se especifica la posición de todas las paredes, así como la posición inicial de Pac-Man, los fantasmas y las cerezas. Pac-Man se representa por el carácter `<` y los fantasmas por el carácter `$`. Las paredes se representan por la letra `X` y las cerezas por el símbolo `*`. Las casillas que no representan paredes (ni contienen un personaje) tienen cada una comida para Pac-Man, representada por la letra `a` y cuando Pac-Man pasa sobre ellas y se la come su representación debe cambiar a la letra `o`. Las cerezas al ser comidas pasan a tomar la forma de `(o)` también.

Justo al iniciarse, el programa pedirá un nombre de archivo del cual cargar los niveles. Si el usuario presiona Enter sin especificar ningún nombre, se cargará por defecto el archivo de nombre `pac.txt` que esté en el mismo directorio donde se invocó al programa. En cambio, si el usuario especifica un nombre de archivo, se cargará el archivo indicado.

Pac-Man

Pac-Man (`<`) es el personaje principal del juego. Puede desplazarse en cuatro direcciones. El usuario pulsa las teclas `w` y `s` para desplazarse en dirección norte y sur, y las teclas `k` y `l` para desplazarse en dirección oeste y este, respectivamente. Pac-Man no puede atravesar las paredes (`X`). Desplazarse a una casilla donde se encuentra una pared (interna o no) no está permitido.

Cuando Pac-Man se desplaza a una casilla que contiene una comida (`a`) o una cereza (`*`), se los come y la casilla queda en `(o)`. Cada comida le suma un `(1)` punto al puntaje (*score*) del jugador, mientras que las cerezas le suman cien `(100)` puntos.

Si Pac-Man se come una cereza sólo aumenta su puntaje. No pedimos que en el proyecto Pac-Man se pueda comer a los fantasmas, como si ocurre en el juego original.

Fantasmas

Pac-Man tiene tres enemigos “fantasma” llamados Blinky, Pinky e Inky. Los enemigos tienen las mismas restricciones de movimiento de Pac-Man, es decir, no pueden atravesar paredes. Los enemigos se desplazan en línea recta y cuando llegan a un obstáculo como una pared eligen un nuevo camino de forma pseudo-aleatoria.

La selección pseudo-aleatoria del nuevo camino que siguen los enemigos viene dada por un algoritmo que le será suministrado. Un enemigo siempre comienza a moverse en dirección norte y se desplaza en línea recta hasta encontrar una pared. Cuando no le es posible seguir en línea recta, la dirección a tomar se decide al azar (con el algoritmo suministrado) entre las direcciones libres.

Cuando un enemigo coincide con Pac-Man en la misma casilla, el enemigo se lo come y Pac-Man pierde una vida. Pac-Man regresa entonces a su posición inicial y se mantiene

el estado del resto del juego: los comidas ya comidas permanecen (o), y los fantasmas (\$) en su misma posición.

Cuando Pac-Man pierde dos vidas, el juego termina y se muestra un mensaje de *Game Over* y la puntuación final del usuario. La puntuación no se acumula por niveles.

Niveles

Pac-Man avanza de nivel cuando se come todos los puntos y cerezas del laberinto. El juego se gana cuando Pac-Man culmina el último nivel.

Generación de números aleatorios

Existen métodos para generar fácilmente una secuencia de números pseudo-aleatorios en un computador. En la carpeta del proyecto hay un programa de nombre *aleatorio.s* que contiene una implementación de un generador de números pseudo-aleatorios, usted debe tomar este programa como base y modificarlo ligeramente para que sirva para generar los valores que su proyecto necesita. La semilla inicial siempre debe ser la misma que allí aparece y cada vez que se requiera un nuevo número pseudo-aleatorio, se debe usar el valor de la búsqueda anterior como una nueva semilla, tal como está actualmente en el programa.

Aspectos Adicionales de la Entrega del Proyecto:

- Su programa consistirá solo en un programa manejador de interrupciones y usará un programa principal que le será suministrado en la sección de documentos de aula virtual en la carpeta de *proyect2* y que tiene por nombre *main.s*. El programa que usted entregue que debe tener el manejador de interrupciones debe tener por nombre *proyecto2GrupoN.s*, donde N es su número de grupo. Para correr el programa, usted cargará este archivo fuente como manejador de excepciones de QtSpim, a través de las opciones del simulador.
- Dado que su programa estará contenido en el manejador de interrupciones, el programa principal que se ejecuta en QtSpim (*main.s*) será un programa *dummy* que solamente imprimirá el mensaje de FIN.
- Usted debe subir el código fuente de su proyecto en la sección de grupos a más tardar el jueves 29 de noviembre a las 11 de la noche al directorio de su grupo en Aula Virtual. Nombre el archivo *proyecto2GrupoN.s*, donde N es su número de grupo. No debe comprimir el archivo a un archivo ZIP, ni a cualquier otro tipo del archivo como .RAR.
- La eficiencia del programa será valorada en la nota del proyecto. En aula virtual hay un documento de buenas prácticas de programación en SPIM que obligatoriamente deben seguir.

- El proyecto es en grupos de hasta dos (2) integrantes, y pueden juntarse con compañeros de diferentes secciones.
- Debido a que el proyecto debe ser realizado en equipo, se asume que cada uno de los miembros del mismo debe conocer plenamente todos y cada uno de los detalles de implementación del proyecto, y podrá ser interrogado al respecto durante la corrección del mismo. Aquellas personas que no muestren un dominio de los detalles del proyecto no tendrán puntos en la evaluación.
- Cualquier caso de copia de proyectos será severamente castigado, no será evaluado ninguno de los proyectos involucrados y serán aplicadas las sanciones correspondientes establecidas en los reglamentos de la universidad.
- Cualquier posible modificación a este documento, será enviado como una notificación a través de Aula Virtual, esté atento a estas notificaciones.

GDTOC