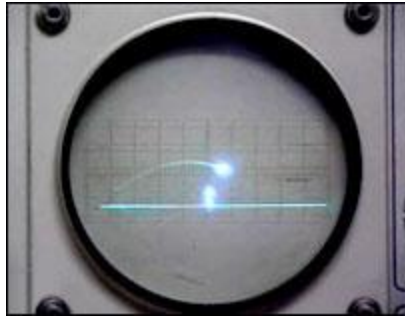




Proyecto 2 Septiembre – Diciembre 2023

Tennis for Two



1 Introducción

Tennis for Two es considerado el primer juego de video (el primer uso de una computadora de propósito general exclusivamente como entretenimiento). Su naturaleza sencilla lo hace ideal para portabilizar (de hecho, puede ser jugado [en internet](#)); sin embargo, el hecho de que está diseñado para ser jugado en un osciloscopio dificulta su uso en sistemas modernos. Por esta razón, se le pide que escriba un programa en MIPS que permita jugar *Tennis for Two* en una pantalla de 25 píxeles por 10 píxeles.

2 Requerimientos del programa

El programa debe funcionar en la consola MMIO simulada de MARS. El programa opera al ritmo de la velocidad de refrescamiento de la pantalla, la cual debe ser de 5 refrescamientos por segundo (es decir, se debe esperar 0,2 segundos). En ese momento debe imprimir el estado del juego, imprimiendo la letra 'O' para indicar los píxeles activos (los píxeles inactivos deben indicarse imprimiendo un espacio).

2.1 I/O

El campo de juego se marca con un "piso" de 25 píxeles con una "red" de 3 píxeles en el centro.

Tennis for Two usaba las perillas del osciloscopio para establecer el ángulo al cual la raqueta impactaba la pelota y el botón de confirmación para golpearla. Debido a que los dispositivos modernos son significativamente carentes en perillas, estableceremos tres ángulos que llamaremos

- Modo Underhand (representa impactar la pelota con la raqueta a un ángulo de 45° respecto a la vertical), el cual el jugador 1 activará con la tecla 'X' y el jugador 2 activará con la tecla 'M';
- Modo Forehand (representa impactar la pelota de tal forma que salga con un ángulo de 45° respecto a la vertical), el cual el jugador 1 activará con la tecla 'S' y el jugador 2 activará con la tecla 'K';
- Modo Backhand (representa impactar la pelota a un ángulo de 0° respecto a la vertical), el cual el jugador 1 activará con la tecla 'W' y el jugador 2 activará con la tecla 'O';

El botón de confirmación será la letra 'D' para el jugador 1, y la letra 'L' para el jugador 2.

Para simular el osciloscopio, la pelota debe dejar una estela de aproximadamente 10 píxeles, indicados con la letra 'o' minúscula.

2.2 Funcionamiento del Programa

La pelota siempre empieza con el jugador 1 en la mitad izquierda de la pantalla. La pelota comienza cada "servicio" a una altura de 4 píxeles y una distancia de 10 píxeles de la red. Al presionar 'D' comienza el servicio dependiendo del modo seleccionado (todos los juegos deben empezar con el modo en Forehand por defecto pero el jugador puede cambiarlo antes de presionar el botón de confirmación)

- En Underhand, la velocidad inicial es de 3 píxeles por en x y de 4 en y
- En Forehand, la velocidad inicial es de 3 píxeles en x y de 3 en y
- En Backhand, la velocidad inicial es de 3 píxeles en x y de 0 en y

Por cada refrescamiento, se debe actualizar la posición de la pelota y reducir la velocidad en y en 1 pixel para simular la gravedad. Note que esto rápidamente hará la velocidad en y negativa, permitiendo que la pelota inicialmente suba y luego vuelva a bajar.

Si la pelota golpea el piso ($y \leq 0$) se debe invertir el signo la velocidad en y. Esto se llama "perder el voleo".

Si la pelota golpea la red ($y \leq 3$ en el refrescamiento en el que la pelota pasa la red), se debe invertir el signo de la velocidad en x.

Si la pelota sale del área ($x < 0$ ó $x > 25$), comienza el servicio del jugador contrario. Esto se llama "fuera".

Si el jugador presiona su "botón de confirmación", se debe revisar si el toque es válido. Un toque es válido:

- Si la pelota está en la mitad de la cancha de ese jugador (ya pasó la red)
- Si la pelota no ha golpeado la red
- Si la pelota aún no ha golpeado el piso en esa mitad de la cancha dos veces
- Si el jugador aún no ha realizado un toque desde que la pelota entró en su mitad de la cancha

Si el toque es válido, se procede según el modo:

- Underhand hará *swap* de los valores de la velocidad en x y en y y les cambiará el signo
- Forehand calculará el promedio entre la velocidad en x y la velocidad en y asignará a ambos ese valor con el signo contrario
- Backhand invierte el signo la velocidad en x sin cambiar la velocidad en y

2.3 Corrida de ejemplo

Diagrama de una cancha de tenis simulada. Se muestra una línea horizontal representando la cancha, con una fila de 26 caracteres 'o' que forman la estela de la pelota. Debajo de la estela, hay una 'D' que indica la posición del jugador 1. Encima de la estela, hay tres 'o' que indican la posición de la pelota en tres momentos diferentes.


```

      oo
      ooo
      oo
      o
      O
      O
      O
      O
Ooooooooooooooooooooooooooooo
W

```

```

      ooo
      oo
      o
      o
      o
      O
      O
      O
      O
Ooooooooooooooooooooooooooooo
D

```

```

      o
      o
      o
      o
      O
      O
      O
      O
      O
Ooooooooooooooooooooooooooooo

```

```

      o O O
      o oO
      o O
Ooooooooooooooooooooooooooooo
D

```

```

      O
      oo
      oo O
      o oO
      o O
Ooooooooooooooooooooooooooooo

```

```

      O
      ooo
      oo
      oo O
      oo
      O
Ooooooooooooooooooooooooooooo

```

DDD

```
Oooo
  ooo
    oo
      O
      O
      O
Ooooooooooooooooooooooooooooo
```

```
oooo
  oo
    O
    O
    O
Ooooooooooooooooooooooooooooo
```

```

      O
    O
    O
    O
Ooooooooooooooooooooooooooooo
```

Obsérvese que si la pelota excede la altura de la pantalla, no se considera fuera del área de juego

2.4 0,2 segundos

La espera de 0,2 segundos se puede realizar con la llamada a sistema 32, la cual detiene la ejecución del programa por la cantidad de milisegundos en \$a0. Se puede invocar con:

```
li $v0,32
li $a0,200
syscall #sleep
```

Para asegurar que los refrescamientos, en efecto, ocurran cada 0,2 segundos, debe contemplarse el tiempo de efectuar el refrescamiento como tal. Para ello, se almacena la hora en milisegundos al inicio del refrescamiento, y nuevamente al final. La hora se puede obtener con la llamada a sistema 30, la cual coloca esta información en \$a1:

```
li $v0,30
syscall #time
```

Se resta la hora obtenida al inicio de la obtenida al final para obtener el tiempo que tomó el refrescamiento, y se resta este número del resultado, así:

```

li $v0,30
syscall #time
move $s6,$a1

#####
##Espacio para Código de refrescamiento##
#####

li $v0,30
syscall #time
sub $s6,$s6,$a1 #tiempo transcurrido

li $t7,200
sub $a0,$t7,$s6 #tiempo restante
li $v0,32
syscall #sleep

```

Nótese, sin embargo, que el tiempo de reanudar el programa luego de un *sleep* es no-determinístico, por lo que puede ser útil medir el tiempo del *sleep* y sumar el exceso a lo que se le restará al siguiente *sleep* (sin embargo, esto no es obligatorio).

3 Requerimientos del Informe

Debe entregar un informe donde se detallen las decisiones tomadas en el proyecto incluyendo:

- Introducción, dando la motivación y estructura del informe
- Diseño de la estructura de datos para almacenar los elementos del juego
- Dificultades Encontradas en la implementación u otras decisiones posteriores
- Conclusiones, dando el estado actual del programa y las lecciones aprendidas

4 Evaluación

El proyecto tiene una ponderación de 10 puntos. Se asignarán

- 3 puntos por código
 - 1 punto por tener un manejador de la entrada y salida usando el simulador MMIO
 - 1 punto por la forma de almacenar el campo de juego
 - 1 punto por la forma de actualizar el campo de juego
- 4 puntos por ejecución
 - 1 punto por el formato de impresión
 - 1 punto por cambiar el modo correctamente
 - 1 punto por detectar el fin del servicio correctamente
 - 1 punto por el movimiento correcto de la pelota
- 3 puntos por informe
 - 1 punto por la introducción y conclusión
 - 1 punto por sus estructuras de datos
 - 1 punto por sus decisiones de diseño e implementación

Todo su código debe estar debidamente documentado y siguiendo las convenciones de MIPS. El programa debe correr sin errores.

5 Extra Credit

Modifique el programa para llevar la puntuación de los jugadores. La puntuación de cada jugador en tenis se escribe en el formato

Game XX

Set Y

Match Z

Donde XX es 00 si el jugador no ha ganado ningún servicio, 15 si el jugador ha ganado un servicio, 30 si ha ganado dos, y 40 si ha ganado tres (recuerde que el servicio se gana sólo cuando la pelota sale del área del juego). Al ganar el cuarto servicio, se incrementa el número de sets (Y) ganado y la puntuación (XX) vuelve a cero. Al ganar el séptimo set, se incrementa el número de matches (Z) y sets vuelve a cero.

Sin embargo, si al ganar el sexto set, el otro jugador tiene menos de 5 sets ganados, el jugador que ganó el sexto set, gana también el match. La necesidad de jugar un séptimo set se llama un *tiebreak*.

El primer jugador en ganar tres matches, gana.

Cada uno de estos elementos lo hará acreedor de un punto adicional para un total de 5 puntos adicionales posibles.