

TETRIS EN MIPS

Autor: Jesús Moreno Ramírez

DNI: 21068175Z

Correo UM: jesus.morenor1@um.es

**Asignatura: Estructura y Tecnología de
los Computadores**

Profesor: Alberto Ros Bardisa

Correo UM (profesor): aros@um.es

ÍNDICE

PORTADA	1
ÍNDICE	2
INTRODUCCIÓN	3
PUNTUACIÓN	3
GAME OVER	3
LINEA	4
ELIMINAR LÍNEA	4
PIEZA SIGUIENTE	5
CONCLUSIÓN	5

INTRODUCCIÓN

En este documento se explicara paso a paso las funciones implementadas para el correcto funcionamiento del **tetris** en el **ensamblador del MIPS**. Las funciones básicas, las que tenían que ser traducidas del **Tetris.c**, no serán explicadas en este documento ya que su funcionamiento está explicado en el PDF que sirve de enunciado para esta práctica. Las funciones que han sido implementadas son **game_over**, **linea**, **eliminar_linea**, **pieza_siguiente**, **puntuación**.

puntuación

Para la puntuación no se ha creado ninguna función en especial, tan solo se han creado dos datos almacenados en memoria uno es una cadena de texto (**puntuacion**) y el otro es un número(**puntos**). Cuando una línea es completada se modifica el valor de **puntos** sumando 10 y también será modificado cuando se coloque una pieza en el campo sumando 1 al valor actual de **puntos**. Para mostrar **puntuacion** y **puntos**, se hace a la vez que se dibuja el campo, para dibujar **puntos** se usa la función **integer_to_string_v4** de la tercera práctica para pasar de entero a texto y dibujarlo.

game_over

En esta función se tenía que lograr que cuando ya no se puedan bajar más piezas se acabe el juego y muestre un mensaje indicando el fin del juego. Para saber si el juego ha terminado lo que he hecho es

llamar a la función **comprobar_pieza** para ver si se puede dibujar en la primera fila, es decir si no hay ninguna pieza en esa posición. Si no se puede poner la pieza se dibujan las cadenas de texto del fin de partida y se espera a que pulse una tecla para llevar al jugador al menú de inicio (se llama a la función de **acabar_partida**).

linea

A esta función se llama cuando colocamos una pieza en el campo, antes de crear una nueva pieza. El funcionamiento es sencillo va recorriendo el campo desde abajo hacia arriba. Empieza a recorrer una línea desde la izquierda hacia derecha, vamos llamando a la función **get_pixel** para que nos devuelva el pixel en el que nos hallamos si devuelve un pixel vacío entonces se pasa a la fila de arriba y así hasta llegar al principio del campo. Si llega al final de una línea y no ha encontrado un pixel vacío entonces se considera que la línea está completa y se llaman a la función **eliminar_linea** y se incrementa el valor de **puntos** en 10. Aunque se haya encontrado una línea completa se seguirá comprobando si las líneas superiores están completas o no.

eliminar_linea

Cuando se ha encontrado una línea completa es necesario eliminar esa línea, es así como funciona el tetris. Para eso recorreremos la línea que está completa copiando la línea de arriba, es decir, hacemos un **get_pixel** para obtener el píxel que hay exactamente encima del que queremos borrar, con **set_pixel** lo colocamos donde

queremos borrar . Si la línea completa es la primera línea en vez de copiar la línea de arriba lo que se hará es llenar esa línea de pixels vacíos.

pieza_siguiente

Para conseguir mostrar la pieza siguiente lo que se ha hecho es crear un dato en memoria que es **pieza_next** donde se almacenará la pieza siguiente. Para mostrar la pieza siguiente lo que se ha hecho es crear una “caja” para contener a **pieza_next** cuando se dibuja el campo se dibuja el “contenedor” de **pieza_next** con la respectiva pieza dentro. Antes de dibujar se tiene que copiar en **pieza_actual** la pieza siguiente, almacenada en **pieza_next**, luego se creará una pieza aleatoria la cual se almacenará en **pieza_next**.

CONCLUSIÓN

En general estoy bastante contento con el trabajo ha sido algo entretenido y conforme se ha ido trabajando se ha ido haciendo menos tedioso. No ha sido tan difícil como las prácticas donde era algo nuevo y bastante lioso, se planteaba como un reto. Una vez que se tiene claros algunos conceptos como el uso de la pila y los bucles la dificultad es la misma que programar en un lenguaje de alto nivel, más largo de programar pero casi sin dificultad.