**Práctica 2: Snake**

* **Diagrama de flujo**

Diagrama

Descripción generada automáticamente

* **Diseño**

En este caso el código trabaja mediante un arreglo que guarda la posición de cada una de las partes de la serpiente, iniciando con la cabeza en la posición 0 y añadiendo detrás de este el resto de las direcciones dentro de los leds. Cabe destacar que en dicho arreglo está directamente el número que representa la posición en los leds en lugar de las coordenadas X y Y por separado, por lo que para obtener estos valores es necesario usar módulo y división.

Para conocer si una manzana fue consumida o si alguna colisión con el resto del cuerpo sucedió basta con comparar los valores de la cabeza con estos, por lo que se necesita recorrer toda la serpiente tanto para buscar colisiones como para actualizar las nuevas posiciones de cada uno de estos.

Al iniciarse, el juego se encarga de limpiar por completo el tablero de leds, por lo que se puede ejecutar varias veces el código sin la necesidad de cerrarlo y volverlo a abrir, una vez hecho esto se muestra la serpiente completamente estática con 2 segmentos en la esquina superior de la pantalla y no es sino hasta que se interactúa con el dpad para que la víbora empiece a moverse y que aparezca la primera de las manzanas.

* **Requisitos del simulador**

Imagen que contiene Patrón de fondo

Descripción generada automáticamenteDadas las características de nuestro programa y las especificaciones dadas en un inicio es necesario que el tablero de leds tenga unas medidas divisibles entre 2, es decir, pueden ir desde 4 hasta 256 pero tanto la altura como el ancho deben ser múltiplos de 2, esto como ya se mencionó es a causa de que tanto las manzanas como la serpiente trabajan mediante bloques de 2x2, por lo que usar saltos o atributos de solo 1 tamaño en los leds complicaría enormemente las cosas, en este caso mostramos un tablero de 16x16.

Forma

Descripción generada automáticamenteA su vez es necesaria la creación de un dpad para controlar los movimientos de la serpiente, este ya por su parte tiene por defecto el añadido de ser controlado por las teclas WASD o las flechas, por ende, en caso de requerir mayor velocidad con las direcciones estas teclas pueden ser usadas, cabe destacar que gracias al código no importa si se quiera ir a una dirección invalida, como querer ir a la derecha mientras se avanza a la izquierda, dado que dicha entrada simplemente se ignorará.

Texto

Descripción generada automáticamenteFinalmente, nuestro sistema de puntajes es algo sencillo dado que solo se muestra al final de la ejecución, es decir, al chocarse consigo mismo, y da como resultado la cantidad de manzanas consumidas durante la partida, dicho puntaje aparece simplemente en la consola.

* **Configuraciones de cache (Primera parte)**

Para esta parte del reporte y en base al requisito mínimo dado en la actividad, las siguientes pruebas dentro del caché serán realizadas en un tablero de leds de tamaño 26x36, además de que como está especificado al final de la actividad, cada uno de los bloques o vías tendrán un tamaño de 2 palabras.

Mapeo directo 4 líneas/bloques:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Asdads

Asdasd

Asdasd

Asociativa con 2 conjuntos y 2 vías (ways):

Tabla

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Asdsd

Asdas

Asdasd

Totalmente asociativa con 4 líneas/bloques:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Asda

Asdas

Adsa

* **Configuraciones de cache (Segunda parte)**

Mapeo directo 16 líneas/bloques:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Gráfico

Descripción generada automáticamente con confianza media

Asd

Asd

Asd

Asociativa con 4 conjuntos y 4 vías (ways):

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Gráfico

Descripción generada automáticamente con confianza media

Asdasd

Asdasd

Asda

Totalmente asociativa con 16 líneas/bloques:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Asd

asd

asd

* **Optimización de cache**

asdasd

* **Conclusiones**

**Alberto Renteria Camacho:**

Asdasd

Asdas

asdas

**Yael Alejandro Rodríguez Barreto:**

Si bien el desarrollo de la práctica se aleja de las cosas vistas en el anterior trabajo, sí se acerca en gran medida a todo lo repasado durante esta última parte del semestre, pasando por cosas como de nuevo trabajar con la herramienta Ripes, pero en esta ocasión mediante el lenguaje C, el manejo de las entradas y salidas de datos en este como lo son los leds y el pad, hasta cosas todavía más reciente como el funcionamiento de la caché en la lectura y escritura de datos dentro de la memoria.

En este caso y como era de esperarse gran parte del foco de la actividad se centra tanto en el análisis como lectura de información dentro del caché, haciendo uso de todo lo aprendido tanto para entender como este trabaja (Cosas como cache de mapeo directo, asociativa y por conjunto) y cuáles de estos métodos son mejores para la optimización de memoria.

Cabe destacar que, además de lo anterior, no solo basta con entender cómo es que estos trabajan, sino también de cómo se conforman para realizar un análisis completo y acorde a lo solicitado desde un inicio.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza media