

Estructuras usadas en el proyecto de estructuras lineales

Resumen del proyecto: El proyecto se trata de un minijuego desarrollado en JavaScript, html y css donde el objetivo es llenar estantes apilando bloques de un valor específico. Cuando el jugador llena exactamente el estante (es decir que la suma de los valores de los bloques apilados es igual al valor del estante), el jugador gana puntos iguales al valor del estante. El objetivo es acumular la mayor cantidad de puntos antes de que las vidas se acaben y termine el juego.

Para apilar un bloque en un estante, el jugador debe seleccionar previamente uno de los dos estantes que se le presentan, luego debe dirigirse a la parte izquierda de la interfaz donde podrá tomar un cañón y disparar a bolas marcadas con un valor aleatorio, cuando la bola es impactada, explota, y de forma automática se apila un bloque de valor igual al de la bola explotada al estante seleccionado. El método anterior es el principal para apilar bloques, llenar un estante y ganar puntos, aunque hay un método secundario para lograr apilar los bloques; se trata de explotar directamente las bolas con el mouse, para ello el jugador hace clic directamente en una bola para explotarla, una vez explotada, se crea un bloque del valor de la bola explotada y se almacena en un espacio aparte, el jugador podrá hacer uso de los bloques almacenados apilándolos en el estante seleccionado. Aunque es más fácil apilar con el mouse que disparando, la opción de apilar con el mouse esta limitada por tiempo y cantidad de bolas que se pueden explotar.

Puede pasar que, al apilar los bloques en un estante, la suma de los valores de los bloques exceda el valor del estante, por lo cual el jugador deberá desapilar los bloques excedentes para poder conseguir la suma exacta y ganar puntos.

El juego termina si se pierden cada una de las 7 vidas que tiene el jugador inicialmente; a medida que el jugador va ganando puntos, se le concede una vida más, aunque no se puede tener más que 7 vidas.

Luego de explicar un poco de que se trata el juego, vamos a ver las estructuras de datos utilizadas y explicar el por que y como se usaron dentro del juego:

Arreglos dinámicos: los arreglos dinámicos fueron utilizados para controlar las vidas, las bolas y los proyectiles disparados desde el cañón.

Para controlar las vidas se crea un Array en JavaScript donde se guardan los elementos html que representan las vidas del jugador; mediante un bucle temporizado, se toma la ultima vida del arreglo de vidas y se le aplica una animación visual para indicar al jugador que está perdiendo una vida, luego de que termina la animación, el elemento html que representa la vida se elimina completamente del programa (tanto del arreglo dinámico como del flujo de elementos html). Cuando un usuario gana una vida por acumulación de puntos, una nueva vida es creada y puesta siempre en el penúltimo lugar del arreglo de vidas, o si el arreglo ya tiene 7 vidas, se reemplaza la ultima vida que estaba en proceso de eliminarse por una nueva vida, para ello se elimina el ultimo elemento del arreglo de vidas y se agrega una nueva vida. Para hacer el mecanismo de vidas del jugador, se usó un arreglo dinámico ya que permite agregar un elemento en una parte específica del arreglo, lo cual es necesario ya que en general, el sistema para recuperar vidas agrega una nueva vida en la posición penúltima del arreglo.

Para realizar el mecanismo de explosión, se usaron dos arreglos dinámicos (dos Arrays en JavaScript), uno para guardar la información de las bolas y otro para la información de los proyectiles disparados desde el cañón. Haciendo uso de un bucle anidado, se recorren las dos listas verificando si la posición dentro de la pantalla de alguna de las bolas coincide con la posición de alguno de los proyectiles disparados; si esto se cumple, quiere decir que ha ocurrido una explosión, por lo cual tanto la bola como el proyectil se eliminan de sus respectivos arreglos. La razón por la cual se utilizaron arreglos dinámicos para simular las explosiones es por la necesidad de recorrer todas las bolas y proyectiles y saber si hubo una explosión, así como poder eliminar cualquier bola y proyectil dentro del arreglo.

Pilas: Cuando ocurre una explosión por disparo de proyectil un bloque se apila en el estante seleccionado, o cuando se saca del espacio auxiliar también se apila en el estante seleccionado, para ello se crea un Array en JavaScript hacer de estante guardando los bloques que se van apilando. Si el jugador excede el valor del estante, el jugador deberá desapilar los bloques excedentes para poder lograr la suma exacta, para ello se elimina el ultimo elemento que se agrego al arreglo haciendo uso del método pop() de los Arrays en JavaScript. Las pilas fueron utilizadas para representar

los estantes del juego, debido a que son ideales para simular el hecho de desapilar un bloque excedente.

Colas: Las colas fueron utilizadas para representar el espacio auxiliar donde se almacenan las bolas explotadas por el método del mouse. Se crea un Array en JavaScript, Cuando una bola es explotada por el método del mouse, se guarda un bloque de valor igual al de la bola explotada en este arreglo. Para el momento en el que el jugador decida hacer uso de los bloques almacenados, saca los bloques en el orden en que fueron ingresados, saliendo primero el bloque que entro en primer lugar dentro del arreglo; El bloque que va saliendo del espacio auxiliar, se va apilando en el estante seleccionado. La cola se eligió como estructura de datos en este caso debido a que satisface la necesidad de eliminar los elementos en orden de llegada.

Nota: para las tres estructuras de datos, se utilizaron los Arrays [] de JavaScript debido a que se pueden comportar de las tres maneras.