

MEMORIA S.MICRO: BUBBLE SHOOTER

Bertol Gómez, Martín
Sánchez Périz, Ixeia
3GTA1

ESTRUCTURA DEL CÓDIGO

En esta memoria, vamos a explicar en líneas generales el funcionamiento de nuestro programa, sin incidir en cómo hace cada uno de los pasos (Para ello, se pueden ver directamente las líneas de código de la subrutina a la que nos referimos en cada momento). Nuestro programa sigue los siguientes pasos:

- Dibuja el recuadro blanco de fondo para delimitar la zona del juego.
- Salta a la subrutina SHOWMAP, que es la que nos dibuja el mapa que tenemos guardado en la dirección que indica MAPA.
- Va a la subrutina PRIMERABOLA para dibujar la primera bola que vamos a lanzar (como después de la primera bola, el color de esta será el que nos muestra la bola pequeña, hemos asignado un color a la primera bola para no tener que recurrir al programa de aleatorio solo para sacar una bola).
- Salta a la subrutina BALL, que dibuja la bola pequeña (esta bola y las sucesivas sí que son aleatorias).
- Entra en el bucle LOOP, que espera a que el jugador haga click en la pantalla.
- Tras el click, identifica cuál es la posición en la que se pretende colocar la bola con el algoritmo que hemos implementado.
- Una vez obtenida la dirección, la guarda en el dato GUARDADIRECCION y comprueba si la puede dibujar o no en función de si ese hueco está vacío o ya tiene una bola.
- En caso de que el hueco esté vacío dibuja la bola y mueve el color de la bola pequeña (guardado en COLORBOLA) a la bola grande (con la etiqueta COLORGRANDE).
- Vuelve a dibujar la bola grande y la bola pequeña saltando a las subrutinas BOLAGRANDE y BALL respectivamente.
- Introducimos un delay para que el color aleatorio que se observa mientras el programa espera el click del jugador coincida con el que se lanza, ya que sino cambiaba demasiado rápido y se colocaba un color nuevo.

- Salta a la subrutina EXPLOTAR, que sirve para comparar las bolas adyacentes del mismo color que la bola disparada (ha de tener en cuenta si la bola disparada está en una fila par o impar), metiendo posteriormente un 1 en la dirección correspondiente de MAPACOLOR.
- Va a la subrutina RECORREMAPACOLOR, que sirve para identificar bolas del mismo color adyacentes a las adyacentes.
- Repite esta subrutina 10 veces para asegurarnos de que ya hemos cogido todas las bolas que se tienen que explotar.
- Si tiene al menos tres bolas adyacentes, salta a la subrutina VACIARMAPA3, que sirve para explotar las bolas.
- De no tener tres bolas del mismo color adyacentes o tras hacer el paso anterior, va a la subrutina COMPROBACIONDESCOLGADAS, en la que se comprueba si hay bolas descolgadas y nos aseguramos de que la primera fila del mapa auxiliar que utilizamos en esta subrutina, MAPADESCOLGADO, este rellena de unos.
- Salta a la subrutina DESCOLGADAS, que guarda un 0 en las direcciones de las bolas que no tienen camino hasta la parte de arriba del mapa.
- Repite esta subrutina 10 veces para asegurarnos de que hemos hecho bien los caminos de unos hasta la parte superior del mapa
- Va a la subrutina VACIARMAPA5, que elimina las bolas de MAPADESCOLGADO en las que hay un 0.
- Salta a NUEVAVUELTA, donde limpia los dos mapas auxiliares.
- Realiza un delay para evitar errores varios.
- Mediante un contador llamado CONTADORBAJARFILA, comprobamos si debemos bajar ya la fila nueva o todavía tenemos que esperar.
- En el caso de no tener que bajar una fila, salta directamente a NOBAJARFILA, para mostrar por pantalla la puntuación y volver al LOOP, para esperar el siguiente click.

- En el caso de tener que bajar una fila, entramos en BAJARFILA, donde primero movemos todas las filas una fila hacia abajo y, después, creamos una fila con bolas aleatorias utilizando la subrutina BALL2.
- Tras esto, volvemos a dibujar el recuadro blanco y el mapa, preparando así el juego para el próximo click, y volvemos a empezar.

COMENTARIOS ADICIONALES

Si en algún momento se lanza una bola y queda alguna bola descolgada sin eliminarse, se tendrá que esperar al siguiente lanzamiento para que se eliminen, ya que no hemos añadido ningún recorrido de mapa hasta después del lanzamiento. Pero este error es momentáneo.

Cabe destacar que falta por implementar algún que otro algoritmo como mejora del juego, que debido a la falta de tiempo y a la extensión del trabajo no hemos terminado. Por ejemplo, habíamos pensado en impedir el lanzamiento en aquellas posiciones que estuvieran bloqueadas (sería suficiente con comparar las dos inferiores dependiendo de si es fila par o impar). También mostrar un contador por pantalla para mostrar la puntuación acumulada del jugador.

Además, no existe botón de RESET o de FIN DEL JUEGO. Es decir, si se quiere terminar el juego se deberá cerrar la simulación y volver a ejecutarla. Si se pierde la partida y todas las filas están llenas de bolas, el juego deberá cerrarse manualmente.

Por último, hemos intentado implementar la puntuación, pero tenemos algún error y por ello hemos comentado el código que hace referencia a este algoritmo.