

Trabajo Mastermind

Antonio María Franqués García y Eduardo Pastor Picazo, grupo A2A.

Presentación

En este proyecto hemos programado el conocido juego del Mastermind utilizando el método gráfico que el Easy68k pone a nuestra disposición. Las partes más notables de nuestro trabajo son:

- Interfaz completamente gráfica. El usuario puede realizar todos los pasos del juego utilizando el ratón. Desde la elección de las características disponibles para la partida hasta la configuración de sus respuestas. La presentación de las combinaciones y de los colores tanto disponibles como las combinaciones de las respuestas se realizan gráficamente para facilitar la entrada de datos por parte del usuario y la asimilación de las pistas ofrecidas para encontrar la solución.
- Combinaciones aleatorias. Cada partida comprenderá un reto diferente gracias a la randomización de las combinaciones.
- Configuración de la partida. El jugador puede elegir, el número de bolas con las que jugar (4,5 ó 6), los turnos disponibles para descubrir la combinación secreta (12 ó 16) y si permite la repetición de bolas en la combinación a descubrir.

Funciones básicas y mejoradas

El funcionamiento básico del programa se basaba en:

- La correcta obtención y almacenamiento de las combinaciones, tanto la generada aleatoriamente como la introducida por el usuario.
- Verificación de si la combinación introducida por el jugador cada turno coincide con la solución, y de no ser así saber obtener las pistas que ayudarán a encontrarla si se siguen.
- Comprobación de si se consigue en un número de turnos menor del que se piden.

Finalmente hemos podido implementar de forma satisfactoria:

- Interfaz completamente gráfica.
- Posibilidad para el jugador de definir algunas condiciones de la partida.
- Implementación de sonidos.

Dificultades encontradas durante la realización

Debido a la intención de mejorar el programa todo lo posible dentro de nuestras posibilidades no hemos podido evitar encontrarnos con ciertas dificultades o criterios muy importantes a tener en cuenta que han marcado nuestro modo de trabajo

- A la hora de programar en el entorno gráfico nos hemos visto algo perdidos. Sin embargo todas las dudas que nos surgían sobre cómo hacer que las cosas aparecieran

por pantalla se dispaban con bastante rapidez al recurrir a la ayuda que el propio easy68k ofrece sobre sus funciones, en especial las relacionadas con el TRAP #15.

- El tamaño de las variables y de los parámetros también ha sido un tema crucial que hemos tenido en cuenta. Así es, pues debido a esto hemos sido capaces de utilizar la memoria de una manera más eficiente.

Rutinas empleadas y su organización

Para este proyecto hemos utilizado varias rutinas que debía seguir el programa en función de cómo se configurara su funcionamiento y otros que serán de uso básico para el funcionamiento. Las rutinas más importantes son las siguientes:

- Generación aleatoria de las combinaciones que serán la solución del juego (inicio en INIRND). Dicha combinación dependerá de si el jugador desea que los valores presentes en dicha combinación se repitan o no. En cualquier caso, para conseguir las combinaciones hemos usado una subrutina ofrecida por el profesor, por lo que no conocemos completamente su funcionamiento.
- Muestreo de cada ficha que se selecciona en pantalla (inicio en PFICHAS). Esta subrutina dibuja una ficha redonda basándose en la posición actual del cursor. El color de la ficha pintada se configura usando el TRAP #15 con el código D0=81 justo antes de llamar a esta subrutina.
- Muestreo de las pistas por pantalla (inicio en PPISTAS). La función de esta subrutina es dibujar las pistas necesarias para ganar el juego a la misma altura que la combinación introducida por el usuario. Al igual que PFICHAS el color con el que pintará cada vez viene dado justo antes de su llamada.

El funcionamiento del programa se puede dividir en los siguientes grandes bloques:

- Configuración de la pantalla, los botones de opciones y la petición de la configuración de juego al usuario.
- Dibujado del tablero de juego y creación de una combinación aleatoria teniendo en cuenta ciertas configuraciones introducidas por el usuario.
- Petición de la combinación a probar por el usuario y su representación por pantalla.
- Verificación de si la combinación es la que estamos buscando. Este algoritmo será explicado en el punto “El corazón”, debido a que es seguramente una de las partes más importantes del programa.
- De ser así, que muestre un mensaje de victoria, la combinación correcta y un sonido triunfal. En caso contrario comprobar cuanto de esa combinación es correcto y mostrarlo por pantalla en forma de elipses blancas y negras.
- En caso de no haber encontrado la combinación antes de que se acaben los turnos se mostrará un mensaje que indicará la falta de éxito en el juego, la combinación correcta y una melodía trágica.
- Habiendo ganado o perdido se pedirá si se quiere volver a jugar, volviendo a la pantalla de configuración de la partida.

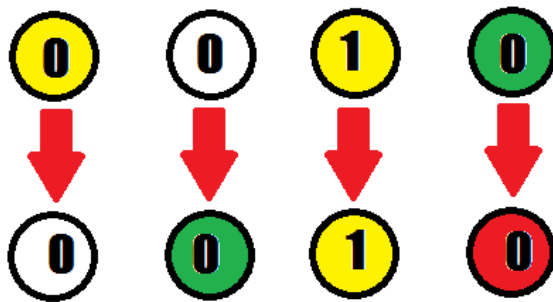
El corazón

En este apartado se explicará el funcionamiento del algoritmo que permite la correcta asimilación de la combinación introducida por el usuario, llevando a la victoria, la derrota o marcando el camino que debe seguir el jugador. Siendo un apartado tan importante ha parecido correcto dedicarle un espacio aparte. Este bloque se divide en varios pasos.

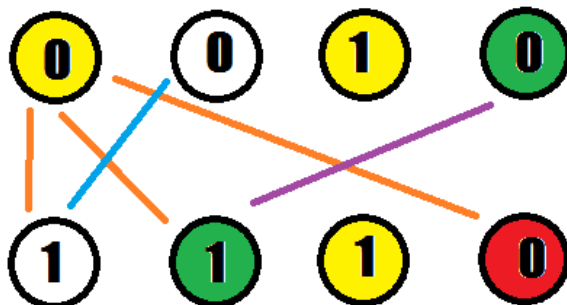
1º Verificación en bloque: Lo primero que haremos al recibir la combinación introducida por el usuario será el comprobar si la combinación es completamente correcta. En el caso de que una sola de las posiciones de las fichas introducidas no coincidiera con la combinación generada se pasaría directamente al siguiente paso. Sin embargo, en el mejor de los casos (todas las introducidas están bien posicionadas) podríamos pasar directamente a anunciar el triunfo del jugador.

2º Comprobación de los turnos restantes: En el caso de que la combinación no fuera correcta y no se dispusieran de más turnos el programa indicaría directamente la pérdida del juego.

3º Verificación en paralelo: Aquí cada ficha de la combinación se compara con la ficha en la misma posición que ella. En caso de ser iguales ambas contarían con un contador asociado que las marcaría como “verificadas en la misma posición”, de no ser así recibirían una marca contraria.



4º Verificación en abanico: A continuación se comprobarán todas las bolas de la combinación introducida por el usuario por cada bola de la solución que se compruebe. Es decir, cada vez que se compruebe una bola de la solución que no tuviera su homónima en la comprobación en paralelo mirará a todas las introducidas por el usuario y que no esté marcada buscando su homónima. En este caso sólo las bolas de la combinación introducida por el usuario serán marcadas conforme vayan encontrando sus correspondientes, evitando así que posteriores bolas las tomen como suyas.



5º Cálculo de bolas de ayuda: En este bloque del programa usaremos los contadores asociados a la solución y a la combinación del usuario para calcular las pistas que podremos mostrar por pantalla (negras para color correcto y bien posicionados, y blancas por tener sólo el color correcto). El número de bolas negras será igual al número de contadores asociados a la solución,

mientras que el número de bolas blancas se calcula restando al número de contadores asociados a la combinación del usuario los contadores asociados a la solución. En este caso será:

Nº Bolas negras: 1

Nº Bolas blancas: $3-1=2$

Tras esto se reduce en 1 el número de turnos disponibles, finalizando así la comprobación y asimilación de los datos introducidos.