#### 1. INTRODUCCIÓN

El programa desarrollado es un videojuego que consiste básicamente en construir líneas de 5 o más círculos de colores y rectángulos de al menos 2x2 cuadros, en un tablero de 9x9 cuadros. El objetivo del juego es acumular la mayor cantidad de puntos posibles al formar las líneas y los rectángulos. Este juego es una variante del juego Color Lines, originalmente hecho para MS-DOS. Se cumple con todos los objetivos y especificaciones planteados.

Este informe cuenta con la especificación del diseño y análisis descendente aplicado para el desarrollo del programa. También cuenta con la información del estado actual del programa, sus alcances, limitaciones y características especiales. Además se tratan las experiencias y dificultades que se presentaron en el desarrollo de este programa.

#### 2. DISEÑO

El problema fue dividido en 14 subprogramas:

* cargar\_imagen(nombre)
* cargar\_Objeto(Tipo, CoordenadaX, CoordenadaY)
* Inicializar\_Tablero(Tablero)
* Objetos\_proximos(Tablero)
* Agregar\_proximos(Tablero, Prox\_Obj)
* Jugada\_Valida(Tablero, columnaActual, filaActual)
* Mover\_Objetos(Tablero, turno)
* Tipos\_Eliminacion(Tablero, Objeto, i, j, Aux)
* Puntaje(Eliminados)
* Eliminar\_Objetos(Tablero)
* Fin\_Juego(Tablero)
* Obtener\_Record(puntosT)
* Cerrar\_Juego()
* main()

El subprograma llamado main() contiene al algoritmo principal del programa, básicamente encierra al programa principal y es la estructura más compleja de todo el programa. El resto de los subprogramas son utilizados para resolver problemas pequeños. A continuación se explicará detalladamente el funcionamiento de la función main() y posteriormente una breve explicación de lo que hace cada uno de los otros subprogramas.

##### 2.1. main():

El subprograma más importante es main() el cual es en esencia el programa principal, ya que en ella se realizan todos los procedimientos. Al inicio de esta función se establecen las siguientes variables y constantes locales:

Variables:

* turno: Esta variable sirve para contar los turnos que lleva el juego.
* Tablero: Es la matriz 9x9 que representa al tablero de juego.
* FinDelJuego: Variable para verificar si el juego terminó.
* Fuente: Variable para almacenar la fuente de letra que se usará para imprimir en la ventana en la ventana del juego.

Constantes:

* alto: Tamaño en píxeles del alto de la ventana.
* ancho: Tamaño en píxeles del alto de la ventana.
* CoordenadaXY: Coordenadas del tablero en píxeles para posicionar los objetos.
* CoordenadaY\_ObjetoProx: Coordenada Y en píxeles para posicionar los próximos objetos en ventana.
* CoordenadaX\_ObjetoProx: Coordenadas X en píxeles para posicionar los próximos objetos en ventana.
* Reloj: Reloj importado desde pygame.
* segundos: Contador de segundos.
* NEGRO: Variable para almacenar la codificación del color negro en pygame.

Luego se invoca la función Inicializar\_Tablero con Tablero como parámetro, lo cual nos devolverá el tablero inicializado con 3 objetos en él. Seguidamente se invoca la función próximos objetos, la cual calcula cuales son los próximos objetos en aparecer, y estos son almacenados en la variable Prox\_Obj. Después se inicializa el bucle para la ejecución del juego. Dentro de él lo primero que se hace es verificar si el usuario ha cerrado la ventana mediante la función Cerrar\_Juego, luego se inicializa la ventana y se carga el fondo del tablero. Se establece una secuencia if, donde si el turno es igual a 0 (inicio del juego) entonces se muestran en pantalla los objetos iniciales en el tablero, los próximos objetos en salir y el puntaje que se inicializa en 0, y se aumenta la variable turno en 1.

La siguiente sentencia if verifica si el turno es mayor que 0 (el juego ya ha sido inicializado) y que la variable FinDelJuego sea False (que el juego aun no haya terminado) de ser así entonces se invoca la función Mover\_Objetos con los parámetros Tablero y Turno, y esta devuelve el tablero actualizado con el movimiento hecho por el usuario y la variable turno aumentada en 1.

Luego se invoca la función Eliminar\_Objetos la cual verifica las condiciones para eliminar los objetos, y en caso de que se cumplan los elimina y calcula los puntos obtenidos por haberlos eliminado. La función anterior retorna el tablero actualizado (se almacena en la variable Tablero), una matriz de valores booleanos que reflejan si ocurrió alguna eliminación (se almacena en la variable E), y los puntos obtenidos (se almacena en la variable puntos).

Se lee la matriz E y en caso de que todos sus valores sean False (osea que no ocurrió ninguna eliminación) se agregan los próximos objetos al tablero y se verifica nuevamente si se produce alguna eliminación. Luego se verifica que el juego aun no haya terminado y se calculan los próximos objetos. Se cargan entonces los objetos en el tablero, los próximos objetos y los puntos para mostrarlos en la ventana.

Lo siguiente que encontramos es una sentencia if que verifica si FinDelJuego es True (si ya terminó el juego), de ser asi ejecuta una secuencia de if interna donde la primera guardia verifica que la variable segundos sea menor a 1, de ser así muestra la pantalla de game over y se calcula si el jugador a logrado establecer un nuevo record y de ser así se solicitara el nombre del jugador por consola. La siguiente guardia verifica que la variable segundos sea igual a 1 (que ya se mostró la pantalla de game over) y que la variable New\_Record sea True (el usuario hizo un nuevo record), de ser así se muestra una pantalla que felicita al usuario por haber logrado el puntaje más alto en el juego, y esta se muestra hasta que el usuario cierre la ventana o la consola. La ultima guardia evaluar si la variable segundos es igual a 1 (que ya se mostró la pantalla de game over) y que la variable New\_Record sea False (el usuario no hizo un nuevo record), de ser así entonces se muestra una pantalla con el record actual y el nombre del jugador que lo hizo, además de un mensaje por no haberlo superado. Por último se refresca la ventana.

##### 2.2 SUBPROGRAMAS

* cargar\_imagen(nombre): Esta función lo único que hace es cargar imágenes para mostrarlas en la pantalla. Su parámetro es el nombre/dirección de la imagen y devuelve la imagen.
* cargar\_Objeto(Tipo, CoordenadaX, CoordenadaY): Esta función es la encargada de procesar todos los objetos del tablero. Sus parámetros son el tipo de objeto, y el número de fila y columna donde se encuentra. La función devuelve la imagen del objeto y la posición en píxeles.
* Inicializar\_Tablero(Tablero): Este subprograma sirve para cargar el tablero inicial de juego, que contiene tres objetos aleatorios además de los próximo objetos. Su parámetro es la matriz Tablero y retorna esta misma variable actualizada con los objetos.
* Objetos\_proximos(Tablero): Esta función calcula los próximos objetos que se agregaran al tablero. Solo tiene la matriz Tablero como parámetro y devuelve una matriz que contiene los valores correspondientes a los próximos objetos.
* Agregar\_proximos(Tablero, Prox\_Obj): Este subprograma se encarga de agregar los próximos objetos al tablero. Sus parámetros son la matriz Tablero y el arreglo que contiene los valores de los próximos objetos. Retorna la matriz Tablero con los objetos agregados.
* Jugada\_Valida(Tablero, columnaActual, filaActual): Función que se encarga de verificar si el movimiento que quiere hacer el jugador es válido. Sus parámetros son la matriz Tablero, el número de columna y el de fila del objeto que se desea mover. Retorna un valor booleano que de ser True la jugada es válida, en caso contrario no lo es.
* Mover\_Objetos(Tablero, turno): Subprograma que se encarga de actualizar la posición del objeto que el usuario movió. Sus parámetros son la matriz Tablero y el contador de turnos. Retorna el tablero con la posición del objeto movido actualizada y la variable turno incrementada en 1.
* Tipos\_Eliminacion(Tablero, Objeto, i, j, Aux) : En este subprograma se realizan todos los procesos de eliminación. Tiene como parámetros la matriz Tablero, la variable Objeto que contiene a alguno de los 7 tipos de objetos del juego, las variables i y j que son el número de fila y columna respectivamente de donde se encuentra el objeto en el Tablero, y la variable Aux que es una copia de la matriz Tablero. El subprograma comienza con la asignación de False a la variable Elim (variable que se usa para saber si hubo una eliminación o no), luego se divide por una sentencia if, donde en la primera guardia se verifica que el valor de Objeto sea diferente de 7 (que el objeto no es un cuadrado), de ser así se realizan 4 procesos diferentes para eliminar líneas verticales, horizontales o diagonales de 5 o más pelotas si es que estas existen. La eliminación de las líneas se realiza en la matriz Aux y en cada eliminación el valor de la variable Elim cambia a True (es decir que hubo una eliminación). La siguiente guardia verifica que Objeto sea igual a 7 (el objeto es un cuadrado), de ser así se verifica que exista un rectángulo de al menos 2x2 cuadrados, en y tal caso se elimina de la matriz Aux y se cambia el valor de Elim a True (ocurrió una eliminación). La función retorna la variable Aux y la variable Elim.
* Puntaje(Eliminados) : Función que se encarga de contabilizar los puntos obtenidos luego de haber logrado eliminar un rectángulo o una línea de 5 o más pelotas. Su único parámetro es una variable que contiene el número de objetos que el jugador elimino en el turno. Retorna los puntos obtenidos.
* Eliminar\_Objetos(Tablero): Esta función va de la mano con la función Tipos\_Eliminación y Puntaje. Sirve para actualizar el tablero luego de haber calculado las eliminaciones con la función anterior. Retorna la matriz Tablero, un arreglo de valores booleanos que reflejan si ocurrió alguna eliminación en el turno y los puntos obtenidos en el turno.
* Fin\_Juego(Tablero): Este subprograma sirve para verificar si el juego ya ha terminado. Su único parámetro es la Matriz tablero y retorna False en caso de que el juego aun no haya terminado, en caso contrario retorna True.
* Obtener\_Record(puntosT): Esta función calcula i el jugador ha logrado establecer un nuevo récord. Su único parámetro son los puntos totales que obtuvo el jugador en todo el juego. Retorna el récord vigente y una variable booleana que es True en caso de haber un nuevo récord o false si no lo hubo.
* Cerrar\_Juego(): Este subprograma solo funciona para verificar si el usuario ha cerrado la ventana de juego, no posee parámetros ni tampoco retorna ningún valor. Es utilizada en múltiples partes del código para garantizar que la ventana se cierre correctamente cuando el usuario así lo desee.

#### 3. ESTADO ACTUAL DEL PROYECTO

El proyecto se encuentra completo. Cumple con todos los parámetros establecidos para la entrega. Muestra correctamente les elementos en la pantalla, permite al usuario moverlos siempre que la jugada sea válida, elimina correctamente líneas horizontales, verticales, diagonales de 5 o más pelotas así como combinaciones de estas líneas, y también es capaz de eliminar correctamente rectángulos de al menos 2x2 cuadrados. Cuenta con sistema de puntuación enlazado a las eliminaciones y muestra el puntaje en pantalla. Además el programa lee correctamente el archivo de texto que contiene el record (crea el archivo en caso de que este no exista), compara la información obtenida en el archivo con la del jugador, determina si se ha producido un nuevo record, sobrescribe el archivo y muestra la información en la pantalla.

Además de lo pedido el programa cuenta con características extras como las siguientes:

* Muestra el número de turnos que lleva el juego.
* Muestra mensajes de error personalizados para cada error de input que el usuario pueda cometer.
* En caso de haber eliminado objetos se imprime en pantalla el número de objetos eliminados y los puntos obtenidos.
* Cuenta con pantalla de game over.
* Cuenta con dos pantallas alternativas de record, una para cuando el jugador logra superar el record actual, y otra para cuando no logra hacerlo.
* La ventana puede cerrarse en cualquier momento. Pero mientras el juego no haya terminado es necesario ingresar alguna coordenada para que el cierre se haga efectivo. Además muestra un mensaje en la consola informando que ce ha cerrado la ventana.
* Le solicita al jugador un apodo para guardar el record, en caso de haber logrado un nuevo record

#### 4. CONCLUSIONES

Los resultados obtenidos son satisfactorios aunque se presentaron algunas dificultades en la parte gráfica del programa por la poca experiencia trabajando con pygame, sin embargo esto se solucionó rápidamente. El movimiento de los objetos fue uno de los apartados del programa que también dio problemas al inicio. La experiencia desarrollando este programa ha sido muy buena, este juego fue muy divertido y emocionante de hacer ya que no fue una actividad monótona sino que hubo que pensar bastante, buscar mucha información y aprender a usar muchas herramientas nuevas. La única recomendación que creemos sería buena implementar en el curso es enseñar más sobre librerías como pygame para que su uso sea más fácil al desarrollar programas en python.

#### 5. FUENTES CONSULTADAS

www.pygame.org

www.razonartificial.com

es.wikipedia.org/wiki/Pygame