



UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES
CENTRO DE CIENCIAS BÁSICAS
ING. EN SISTEMAS COMPUTACIONALES

PROGRAMACIÓN III
PROYECTO FINAL
BEJEWELD

MAESTRO:
JOSÉ DE JESÚS PALOS GARCÍA

ALUMNOS:
GERARDO FEMAT DELGADO ID: 244371
EMMANUEL MUÑOZ CERDA
GRUPO: 4° C

AGUASCALIENTES, AGUASCALIENTES
06/06/2022

Introducción:

En este proyecto se creará el juego de “Bejeweled” de PopCap utilizando el lenguaje Java. De forma general se utilizan elementos como: JPanel, JFrame, Graphics, Threads, JButtons, Files entre otros. Para lograrlo se formaron equipos de dos personas, en este caso específico, cada integrante creó, modificó y ajustó elementos del código así como diseño de elementos visuales. El método utilizado en este trabajo es la creación de una clase por cada posible elemento necesario para el funcionamiento del juego; de esta manera se creó una estructura entendible alrededor de todo el programa. Durante el desarrollo del documento se irá explicando que función cumple cada una de las clases.

Por otro lado, fue necesario el realizar una investigación metódica, para aprender funciones necesarias que se incluyen en Java, obteniendo una mayor visión de las capacidades de este lenguaje interpretado. También se observaron comportamientos que no se pensaban fueran posibles entre gráficos y paneles, de modo que se tenga una vista mucho más uniforme a lo largo del programa.

El juego consta de un tablero en el cual se aprecian diferentes tipos de gemas. El objetivo del juego es hacer la mayor cantidad de puntos posible moviendo las gemas en columnas y/o filas de tres o más gemas del mismo tipo. Cada vez que se acomodan gemas parecidas de manera contigua, nuevas gemas se crean en la primera fila de la matriz, y todas las gemas que están arriba del lugar donde se encuentran las gemas contiguas, bajan. Este proceso se repite hasta que el usuario no puede hacer más movimientos, por lo que se usa un botón que intercambia las gemas de posición para que el usuario tenga la posibilidad de seguir jugando.

Enlace al proyecto: https://eduuaa-my.sharepoint.com/:f/g/personal/al244371_edu_uaa_mx/EjG7A9T8HNdKuaN-FhhjlkMBZB3uUIW7uNoCqZZRTHZqxA?e=wRxpga

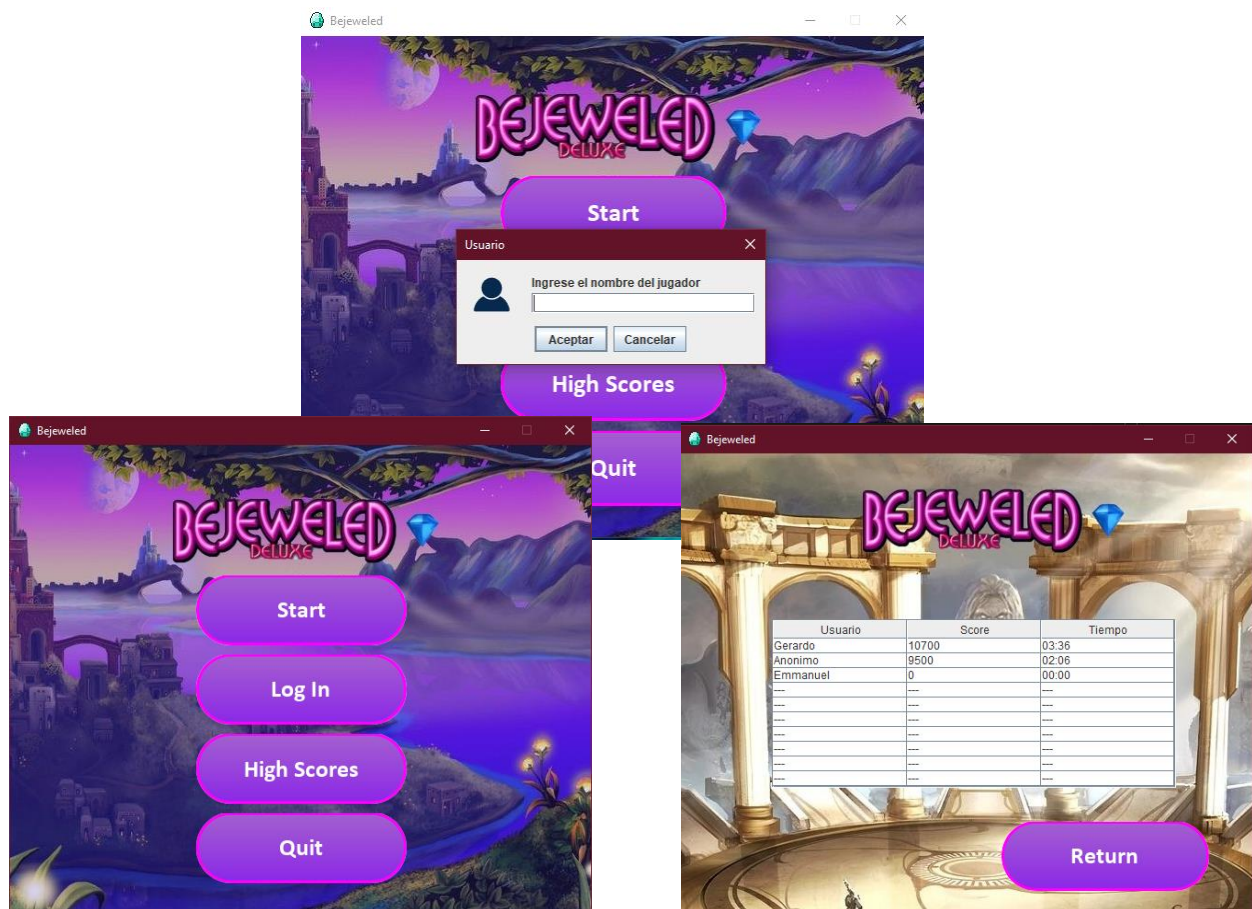
Desarrollo:

- Clase Main.java

Esta clase se encarga de ejecutar el programa y es la que se encarga de mostrarle todo al usuario final. Consta de la creación de un objeto de tipo “VentanaMenu”.

- Clase VentanaMenu.java

Esta clase hereda de JFrame y su función principal es crear una ventana de menú principal la cual consta de cuatro botones (Uno para iniciar el juego haciendo uso de un hilo, otro botón para ingresar al usuario, otro para mostrar las high scores, y un último botón para salir del programa.), así como de dos paneles JPanel, (uno de ellos es donde se encuentran todos los botones, y otro en donde se muestra la tabla de puntaje (PanelHighSores) y cada panel, tiene una imagen de fondo diferente. NOTA: Si el usuario no hace login, se utilizará un usuario anónimo.



- Clase Ventana.java

Esta clase hereda de JFrame y es en esta en donde se lleva a cabo el videojuego.

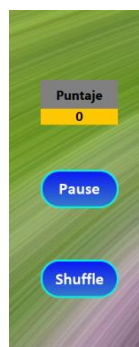
Consta de un panel MainPanelButtonsYLabels, el cual abarca un tercio de la pantalla en donde se encuentran los botones de pausa, shuffle y el marcador de puntaje los cuales se encuentran en la clase “MainPanelYButtons”.

Además de contar con el tablero de juego, pintado gracias a la función dibuja, la cual usa un objeto de tipo Graphics.



- Clase MainPanelYButtons.java

Esta clase hereda de JPanel y consta de un panel de un tercio de la pantalla principal y contiene dos JLabel, una que dice la palabra “Puntaje” y la otra que muestra en tiempo real, el puntaje obtenido por el jugador (el contenido de esta JLabel se usa para guardar el puntaje en el archivo). También contiene dos botones, uno que abre el menú de pausa el cual invoca una clase “PausaFrame” y shuffle que se utiliza para crear toda una nueva matriz en caso de que el usuario quiera replantear su juego, inclusive no encuentre movimientos, llama a una función de los Gráficos principales.



- Clase PausaFrame.java

Esta clase hereda de JInternalFrame, su función es que aparezca una ventana JFrame sobre la ventana principal del juego (Ventana.java). Esta consta de tres botones (JButton):

Un botón de despausa, el cual oculta el JInternalFrame y reanuda la partida del jugador.

Un botón que regresa a la pantalla de menú principal, llamando a guardar los puntos si es que el usuario hizo una mejor puntuación de la que tiene.

Y por último un botón que sale del programa que da un mensaje de salida.



- Clase Gráficos.java

Dentro de esta clase se encuentra gran parte de la lógica del funcionamiento del juego junto a la clase "Scanner", esta crea una matriz de Gemas al iniciar, verifica que no haya más de tres gemas con mismo Id juntas y si existen las elimina hasta que no queden en el tablero. Una vez comprobado, inicia los gráficos y sonidos del juego, mediante un mouseListener detecta que gemas está presionando el jugador, si presiona dos contiguas el juego verificará si se puede realizar el movimiento mediante la clase "scanner", si está permitido entonces iniciará la eliminación, si no retornará la gema a su posición inicial.

El juego tendrá la posibilidad de generar gemas especiales de dos tipos: Joyas de energía juntando cuatro en línea o cinco en forma de L o T las cuales explotan en un radio de tres por tres casillas, y los hypercubos que eliminan todas aquellas gemas con la que se intercambió.

Las gemas de energía se verán representadas con un destello y los hypercubos con un cubo especial que siempre está girando. Cuando alguno de estos destruye a otro

especial se efectuará su función de forma recursiva, es decir, si uno de energía destruye otro con energía entonces el otro también explotará en su rango.

Cuando las gemas son eliminadas mediante una función recursiva la verificación de conjuntos de tres o más gemas será llamado de forma recursiva hasta que en todo el tablero no se encuentren más conjuntos. Una vez finalizado el proceso se detendrá la ejecución por un segundo y se mostrará que gemas serán eliminadas y hacia donde bajarán. También pinta el fondo y el tiempo que lleva el jugador en partida.

Contiene algunos métodos como obtener puntos, obtener el tiempo en formato String y el llamado a el ciclo principal para iniciar el juego, estos métodos sirven para guardar los datos de la partida en el archivo.

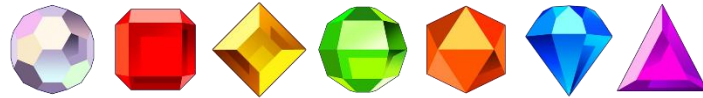


- Clase Scanner.java

Esta se encarga de verificar que se puedan hacer los movimientos entre las gemas y también de eliminar de forma lógica las gemas, analiza cada una de las gemas hacia arriba, abajo, izquierda y derecha y detecta cuantas gemas iguales o especiales de este mismo color hay para eliminarlas de forma lógica, si existen gemas especiales llama a su función de eliminar gemas especiales. Contiene las funciones en polimorfismo, unas son utilizadas para antes de iniciar el juego y otra para hacer representaciones graficas con sonidos. Contiene gran parte de la lógica del juego.

- Clase Gemas.java

Esta clase es la que se encarga de representar a todos los objetos que estarán en el tablero de juego. Esta clase se encarga de darles una posición actual tanto en una matriz lógica, así como en una matriz de pixeles, lo que facilita operaciones de movimiento de dichos objetos. También cuenta con una función que les da una imagen aleatoria. Además de tener varios setters y getters para modificar sus datos y poder realizar sus movimientos.



- Clase Musica.java

Esta clase es la encargada de darle sonido al juego, es utilizada a lo largo de todo el programa y consta de un clip de audio que se ejecuta en la pantalla principal del juego, así como los múltiples sonidos. Es un manejador de música sencillo al que sólo se le debe enviar un archivo y se encarga de crear el clip de audio.

- Clase Reloj.java

Clase que cuenta el tiempo y muestra en pantalla el contador mediante gráficos, debe ser utilizada junto a un hilo para que constantemente cuente sin sufrir de las interrupciones de otros elementos. Tiene funciones para detener el reloj y continuarlo mediante waits y notify, así como una función que retorna su conteo actual en forma de String para ser almacenado en un archivo. Muestra el tiempo en formato Minutos:Segundos, no muestra horas por lo que los minutos se pueden extender de forma indefinida.

Tiempo: 01:31

- Clase MyFileManager.java

Esta clase es la encargada de las operaciones de entrada y salida de datos a un archivo de texto llamado "Scores.txt", todas sus funciones son estáticas. Es ampliamente usado a lo largo de todo el programa, generalmente, las operaciones de escritura en el archivo se dan cuando el programa está por terminar su ejecución o se crea un nuevo usuario en el archivo, también cuando se termina la partida del usuario pero se sigue con la ejecución del programa. La operación de lectura se usa al momento de modificar la tabla de high scores, lo que permite que se muestren de manera fácil. Tiene varios

métodos que ayuda a leer los datos que necesitamos, modificarlos y escribirlos en el archivo. Utiliza una lógica de separadores por tabulación.

- Clase MouseButtonListener.java

Es una clase genérica que ayuda a la creación de botones durante todo el juego, su función crea un MouseListener que hace que se agranden y se hagan pequeños los botones cuando el mouse pasar por ellos. También tiene una función de salida del programa, ejecuta un audio de despedida y cierra el juego.

- Clase VentanaHilo.java

Esta clase es la encargada de crear un hilo de ejecución a la hora de cerrar la pantalla de menú principal e iniciar la ventana principal del juego, todo esto, para que la ejecución del programa no presente problemas y la nueva ventana pueda Invocar su paint.

- Clase RelojHilo.java

Esta clase ejecuta en un hilo el reloj que se le envíe y es necesario que el reloj se utilice junto a esta, pues la clase "Reloj" está diseñada para funcionar en un hilo con funciones wait y notify. Lo único que hace es llamar a una función de Reloj llamada contar.

Nota: Cada vez que se cierra el juego o se regresa al menú durante la ejecución del juego, MyFileManager guarda los datos de la partida si es que el usuario hizo una mejor puntuación que la registrada.

Conclusiones:

Emmanuel:

Crear un juego no es nada sencillo, especialmente cuando se hace en Java, un lenguaje que no está orientado a la creación de videojuegos (como lo puede llegar a ser C# o incluso C++). Sin embargo, es precisamente este reto el que logro llevar al equipo a un nivel de exigencia alto, en donde la adaptación fue clave y permitió que se aprendiera sobre la marcha.

Es importante destacar que al ser el primer semestre en la carrera en donde se ve de lleno la programación orientada a objetos, la curva de aprendizaje fue muy pronunciada, por lo que en un inicio costo algo de esfuerzo entender todos los temas satisfactoriamente.

Gerardo:

La creación del juego fue bastante tediosa, sobre todo la lógica de movimiento, comprobaciones y gemas especiales. Cada una de las clases fue creada pensando en subdividir el programa lo más posible para que su mantenimiento o ajustes antes de la entrega también lo fuera, pienso que lo logramos bastante bien. De forma general no fue nada sencillo aplicar los conocimientos que obtuvimos durante el curso, pues fue necesario hacer consultas de forma constante a la documentación y sobre todo buscar referencias de cómo crear muchas de las clases, tanto de los hilos como botones y sobre todo de archivos, pues muchas de las funciones que utilizamos nunca tuvimos alguna aproximación.

Aunque Java contenga muchas funciones que ayudan a mostrar gráficos, pienso que no es el mejor lenguaje para crear un videojuego al igual que mi compañero, de cualquier forma, mediante la realización del proyecto conocí muchas funciones de Java.

References

- Creating Hover Effect for JButton*. (2008). Retrieved from Daniweb:
<https://www.daniweb.com/programming/software-development/threads/124033/creating-hover-effect-for-jbutton>
- Oracle. (2022, Junio 5). *Java® Platform, Standard Edition & Java Development Kit*. Retrieved from <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>
- PRAMODBABLAD. (2017, Enero 19). *How To Replace Specific String In Text File In Java?* Retrieved from Java Concept Of The Day:
<https://javaconceptoftheday.com/modify-replace-specific-string-in-text-file-in-java/>
- SpolidSoul. (n.d.). *DevianArt*. Retrieved from Bejeweled 3 Gems:
<https://www.deviantart.com/spoildsoul/art/Bejeweled-3-Gems-Hypercube-209086831>
- The VG Resource. (n.d.). *Resource*. Retrieved from The Sounds: https://www.sounds-resource.com/pc_computer/bejeweled3/sound/37398/
- Yadav, C. (2019, Enero 8). *tutorialspint*. Retrieved from Swap elements of ArrayList with Java Collections: <https://www.tutorialspoint.com/swap-elements-of-arraylist-with-java-collections#:~:text=In%20order%20to%20swap%20elements,specified%20positions%20in%20the%20list.>