

TEC - Instituto Tecnológico de Costa Rica – Escuela de Ingeniería en Computación

Carrera: Ingeniería en Computación

Curso: Programación Orientada a Objetos

Estudiantes: Alejandro Montero, Fabricio Monge

Profesor: William Mata Rodríguez

PROGRAMA 3: Mantenimiento de software KenKen (25%)

Fecha de entrega: 27 de noviembre 2023, 11:30 p.m.

IC 2101 Programación Orientada a Objetos

II Semestre, 2023

Escuela de Ingeniería en Computación, Tecnológico de Costa Rica

1 Tabla de contenido

2.

REQUERIMIENTOS DEL PROGRAMANUEVAS FUNCIONALIDADES	3
2.1 OPCIÓN DE CON DIFERENTES TAMAÑOS DE CUADRÍCULA	3
2.2 OTROS CAMBIOS EN LA OPCIÓN JUGAR	5
2.3 PODIO DEL NIVEL ACTUAL.....	5
2.4 PODIO GENERAL POR NIVEL Y CUADRÍCULA	6
2.5 PATRÓN BÁSICO DE DISEÑO	6
3. – TEMAS INVESTIGADOS (MATERIAL NO VISTO EN LAS LECCIONES) (0P o 25P)	6
3.1. Bibliografía.	7
4. SOLUCION (0P o 20P, Hacer Zoom)	8
5. CONCLUSIONES DEL TRABAJO	10
5.1 Problemas encontrados y soluciones a los mismos	10
6- TABLA DE REVISION DEL PROYECTO	12

2 – DEFINICIÓN DEL PROYECTO

En la ingeniería de software el ciclo de vida de desarrollo de software (SDLC: System Development Life Cycle) se refiere al conjunto de etapas necesarias para la creación y utilización de software a través del tiempo.

Hay diferentes metodologías que soportan este ciclo de vida, algunas de ellas son: modelo encascada (el más antiguo), modelo en espiral, prototipos, modelo incremental e iterativo, desarrollo ágil (usando herramientas como SCRUM, programación extrema-XP, Kanban, etc.).

El mantenimiento de software es una actividad natural del ciclo de vida del software, es muy común que nos encontremos trabajando en ello. Consiste en modificar el software por diversas razones:

- Corrección de errores
- Adaptar el software a nuevos requerimientos
- Mejoras (funcionales y técnicas)

En este proyecto vamos a hacer mantenimiento al software realizado en el proyecto anterior: específicamente vamos a hacer mejoras de dos tipos:

- A nivel funcional: se agregarán nuevas funcionalidades.
- A nivel técnico: se implementará algún patrón básico de diseño seleccionado por el programador.

REQUERIMIENTOS DEL PROGRAMA NUEVAS FUNCIONALIDADES

2.1 OPCIÓN DE CON DIFERENTES TAMAÑOS DE CUADRÍCULA

- 1) La versión actual del software es para jugar con el tamaño de cuadrícula 6x6. Esta nueva funcionalidad habilitará juegos con cuadrículas de tamaño 3x3, 4x4, 5x5, 6x6, 7x7, 8x8 y 9x9. Considere que debe modificar todos los componentes del software necesarios para proveer esta nueva funcionalidad, entre ellos: tamaño, despliegue y manejo de la cuadrícula según seleccione el jugador, panel de números permitidos y reglas de juego.

2) A la opción configurar se le agrega un nuevo parámetro:

	5. Tamaño de la cuadrícula	<input checked="" type="radio"/> 3x3
<input type="radio"/>		4x4
<input type="radio"/>		5x5
<input type="radio"/>		6x6
<input type="radio"/>		7x7
<input type="radio"/>		8x8
<input type="radio"/>		9x9

3) Modificación de la estructura del archivo de partidas que para esta versión se llamará kenken_partidas2023.xml: se agrega el dato de cuadrícula (antes del dato de nivel de dificultad):

```
<KenKen>
  <partida>
    <cuadrícula>tamañoCuadrícula</cuadrícula>          3x3/4x4/5x5/...
    <nivel de dificultad>nivelDificultad</nivel de dificultad>      fácil/intermedio/difícil
    <jaula>valor, operación aritmética, (fila, columna), (fila, columna)), ... </jaula>
    <jaula>valor, operación aritmética, (fila, columna), (fila, columna)), ... </jaula>
    ...
    <constantes>(constante, fila, columna), (constante, fila, columna), ...</constantes>
  </partida>
</KenKen>
```

Ejemplo para una partida con tamaño de cuadrícula 6:

```
<KenKen>
  <partida>
    <cuadrícula>6x6</cuadrícula>
    <nivel de dificultad>fácil</nivel de dificultad>
    <jaula>11, +, (1,1), (1,2), (2,1)</jaula>
    <jaula>120, x, (1,3), (2,2), (2,3)</jaula>
    <jaula>3, +, (1,4), (2,4)</jaula>
    <jaula>2, -, (1, 6), (2, 6)</jaula>
    <jaula>11, +, (2, 5), (3, 5)</jaula>
    <jaula>3, +, (3, 1), (3, 2)</jaula>
    <jaula>15, x, (3, 3), (3, 4)</jaula>
    <jaula>1, -, (3,4), (4,4)</jaula>
    <jaula>72, x, (4,1), (4,2), (5,1)</jaula>
    <jaula>8, x, (4,5), (4,6)</jaula>
    <jaula>3, +, (5,2), (5,3)</jaula>
    <jaula>13, +, (5,4), (6,4), (6,5)</jaula>
```

```

    <jaula>9, +, (5,6), (6,6)</jaula>
    <jaula>2, /, (6,1), (6,2)</jaula>
    <constantes>(3, 1, 5), (6, 3, 6), (1, 6, 3)</constantes>
  </partida>
</KenKen>

```

2.2 OTROS CAMBIOS EN LA OPCIÓN JUGAR

- 4) Agregar el dato Nombre del jugador (string de 1 a 40 caracteres): antes de iniciar el juego el jugador debe obligatoriamente dar un nombre.

5) Podio

Se refiere a los mejores 3 jugadores de cada nivel de dificultad para cada tamaño de la cuadrícula. Los mejores jugadores son los que completan el juego en el menor tiempo. El programa registra esta información en el archivo “kenken2023podio.dat” cuando el jugador usa el botón VALIDAR JUEGO.

En ese momento el programa debe determinar si el jugador debe registrarse en el Podio. Las partidas que no usan reloj no entran al Podio. Si se tienen los 3 mejores jugadores y el jugador actual hace un mejor tiempo que esas marcas, hay que eliminar la marca con mayor tiempo para seguir teniendo un máximo de 3 por nivel y cuadrícula. La marca contiene el nombre del jugador y el tiempo (horas, minutos, segundos) que el jugador usó para completar un juego. Registre también la fecha y hora del sistema. Note que si usa el timer hay que calcular la duración del juego (al tiempo establecido inicialmente se le resta el tiempo en que se detuvo el timer).

6) Botón nuevo

PODIO

Esta opción se puede usar en cualquier momento del juego. Detiene el reloj si lo está usando. Despliega los registros de los mejores 3 jugadores por cada nivel de dificultad para cada tamaño de cuadrícula. Esta información se toma del archivo “kenken2023podio.dat”.

La primera información que debe desplegarse es la del podio correspondiente al nivel y tamaño de la cuadrícula indicada en la opción jugar. Por ejemplo, si está en el nivel FÁCIL con una cuadrícula 6 x 6 se despliega primero ese podio y luego los demás:

2.3 PODIO DEL NIVEL ACTUAL

<i>NIVEL FÁCIL</i>		<i>JUGADOR</i>	<i>TIEMPO</i>
6 x 6	ORO	Nombre del jugador	0:10:11
	PLATA	Nombre del jugador	0:10:15

BRONCE Nombre del jugador 0:11:20

2.4 PODIO GENERAL POR NIVEL Y CUADRÍCULA

<i>NIVEL DIFÍCIL:</i>		<i>JUGADOR</i>	<i>TIEMPO</i>
3 x 3	ORO	Nombre del jugador	0:05:15
	PLATA	Nombre del jugador	0:05:55
	BRONCE	Nombre del jugador	0:06:59
6 x 6	ORO	Nombre del jugador	0:35:19
	PLATA	Nombre del jugador	0:36:38
<i>NIVEL INTERMEDIO:</i>		<i>JUGADOR</i>	<i>TIEMPO</i>
6 x 6	ORO	Nombre del jugador	0:30:19
	PLATA	Nombre del jugador	0:31:38
	BRONCE	Nombre del jugador	0:32:00
<i>NIVEL FÁCIL:</i>		<i>JUGADOR</i>	<i>TIEMPO</i>
4 x 4	ORO	Nombre del jugador	0:04:00
	PLATA	Nombre del jugador	0:04:49

2.5 PATRÓN BÁSICO DE DISEÑO

El programador selecciona algún patrón básico de diseño. Ejemplos: patrón Singleton para aplicarlo de tal forma que garantice la creación de solo una instancia con la lista de partidas de juego almacenadas en el archivo `kenken_partidas2023.xml`, patrón Factory para crear objetos según la cuadrícula que se requiera en el momento para jugar (3x3, 4x4, etc.), patrón Observer para tener la lista de jugadores de tal forma que cuando ocurra un cambio en algún podio se les informe mediante un mensaje o correo, etc.

3. – TEMAS INVESTIGADOS (MATERIAL NO VISTO EN LAS LECCIONES) (0P o 25P)

Formato XML: Se investigó acerca de archivos que poseen formato XML para la carga de partidas. Descubrimos que es un formato organizado mediante una jerarquía de etiquetas y

se utiliza en muchos softwares y programas para comunicar información. En el proyecto se utilizó para gestionar la forma de cargar las partidas en el tablero.

Lectura de archivos XML: Se investigó acerca de dos enfoques para leer archivos XML, el primero fue JAXB, el cual es una librería que nos permite mapear XML a clases de manera sencilla. Sin embargo, este no viene instalado por default y en lugar de instalar sus dependencias y gastar tiempo montándolo, optamos por trabajar con DOM (Document Object Model). Esta otra manera de leer archivos XML nos permite analizar y manipular el archivo de manera que poseemos más control a la hora de leer y procesar los datos por medio de las diferentes etiquetas y elementos. DOM es más manual ya que hay que programar más, sin embargo, la flexibilidad a la hora de actualizar el código que lee el archivo XML hizo que valiese la pena.

Para recuperar el contenido del XML, DOM posee diferentes funciones para recuperar elementos por el nombre de su etiqueta, por ejemplo: “root.getElementsByTagName(String x)”. Utilizando esto en conjunto a “(Element).item(index)” se extrajo la información relevante a la partida y se cargó en las clases correspondientes.

Reproducción de Audio: Java posee librerías propias para reproducir audio. Se pueden acceder por medio del comando “import javax.sound.sampled.*”. Esto nos permite cargar archivos de audio, crear clips de audio, abrir el audio y asociarlo con el clip, así como reproducirlo. Para reproducir el audio solamente se tiene que dar el camino al archivo a utilizar, en nuestro caso fue un .wav.

Apertura de PDFs: Java también tiene librerías implementadas por Apache que sirven para la manipulación de PDFs. En nuestro caso se utilizó PDFBox, con la cual se puede escribir y leer archivos PDF, añadir contenido a páginas, crear nuevas páginas, extraer texto, extraer imágenes, entre otras funciones. Realmente es una biblioteca bastante completa para trabajar con PDFs, sin embargo, se utilizó solamente para abrir el manual de usuario del proyecto.

3.1. Bibliografía.

Apache PDFBox / A Java PDF Library. (s. f.).

<https://pdfbox.apache.org/>

Java Sound Technology. (s. f.).

<https://docs.oracle.com/javase/8/docs/technotes/guides/sound/index.html>

XML Parser. (s. f.).

https://www.w3schools.com/xml/xml_parser.asp

Wikipedia contributors. (2023, 22 octubre). *XML*. Wikipedia.

<https://en.wikipedia.org/wiki/XML>

Timer (Java Platform SE 7). (2020, 24 junio).

<https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/java/util/Timer.html>

Duration (Java Platform SE 8). (2023, 4 octubre).

<https://docs.oracle.com/javase/8/docs/api/java/time/Duration.html>

Instant (Java Platform SE 8). (2023, 4 octubre).

<https://docs.oracle.com/javase/8/docs/api/java/time/Instant.html>

4. SOLUCION (0P o 20P, Hacer Zoom)

5. CONCLUSIONES DEL TRABAJO

5.1 Problemas encontrados y soluciones a los mismos

- **Objetos Timer, Duration e Instant:** Durante el desarrollo del cronómetro en Java, nos enfrentamos al problema de gestionar la precisión al utilizar las clases Duration, Timer e Instant. Un problema surgió al intentar detener estos objetos para poder reiniciar el cronómetro o timer, ya que en casos no detenía el timer o cronómetro actual y al generar otro, el Label encargado de mostrar el tiempo se actualizaba en dos momentos diferentes, mostrando 2 cronómetros que llevaban el tiempo de manera diferente. La solución a este problema fue validar en diferentes casos del programa que se eliminara el objeto actual antes de llamar al siguiente. Por ejemplo, al terminarse el tiempo y querer continuar con un cronómetro, hay que cargar de manera negativa el tiempo del timer para poder generar un cronómetro con el tiempo que llevaba el timer.
- **Lectura de archivos XML:** En un inicio se tuvo la idea de trabajar con la Librería JAXB, sin embargo, al ver que se ocupaba instalar más dependencias y no estar seguros de cuáles funcionaban se decidió trabajar con DOM. Se desperdició mucho tiempo intentando trabajar con JAXB solo porque se había investigado que era más fluido trabajar con clases. Finalmente, se solucionó cambiando a DOM e investigando como incorporar las clases de celda y partida en el lector de XML, ya que realmente no era tan complicado y DOM se prestó para una mayor flexibilidad al tener que programar todo de manera más manual que JAXB.

Aprendizajes Obtenidos

- Para los objetos Timer, Duration e Instant, Se debe llevar un buen control de los objetos para que no termine en desastre. Se debe estar seguro de que los procesos anteriores han sido terminados antes de volver a iniciar otros que utilicen las mismas implementaciones como en el caso de los relojes.
- En el tema de los archivos XML, se aprendió que algunas herramientas más llamativas no siempre son mejores. Tratamos de usar JAXB porque parecía más

sofisticado y útil con el tema de clases, pero nos dimos cuenta de que la simplicidad de DOM era lo que necesitábamos. Nos enseñó que entender las bases y manejar las cosas a menudo da más flexibilidad y es más práctico. Fue una lección sobre no complicarse y apegarse a lo que funciona en el momento, aunque al inicio pueda parecer complicado.

- Se modificó la forma de generar los botones, puesto que había que añadir nuevos tamaños para la creación de los otros tableros, estos se crean mediante un for, donde se añade la función de interacción, la cual ahora está en un archivo aparte y se llama desde el botón, se pasaron otras funciones a este archivo aparte para algunas funciones de juego como deshacer y rehacer las jugadas.
- El modelo de observador se implementó en las configuraciones, y esto significa que existe un observador y un objeto específico siempre y cuando la aplicación ejecute esta ventana, el observador notifica los cambios de la configuración para hacer el objeto data en otro archivo que recibe la notificación del observador.

6- TABLA DE REVISION DEL PROYECTO

Concepto	Puntos	% de avance 100%/0	Puntos obtenidos	Análisis de resultados
Opción Jugar (7 cuadrículas):		100%		
Desplegar cuadriculado	7			
Desplegar información de cada casilla según partida	7			
Enmarcado de las jaulas según partida	7			
Desplegar el resto de la opción	7			
Iniciar Juego:		100%		
Actualizar casillas (colocar y borrar números)	7			
Controles del proceso	7			
Validar Juego	7	100%		
Opción 5 de configurar	5	100%		
Modificación archivo de partidas kenken_2023.xml	3	100%		
Nombre jugador	2	100%		
Crear podio en opción VALIDAR JUEGO	6	100%		
Botón Podio		100%		
Desplegar primero podio nivel actual	5			
Desplegar otros podios	5			
Patrón básico de diseño seleccionado por el programador	15	100%		
Ayuda (despliegue del PDF del manual de usuario)	2	100%		
Documentación del proyecto: Manual de usuario	4	100%		

Resto de la documentación	4			
TOTAL	100			
Funcionalidades desarrolladas adicionalmente				