Documento de Análisis

da.acostac1

c.caro

j.serratos

Iteración 1:

Se determinó como rol el dominio general del juego lights out

Roles: Juego Lights Out

Responsabilidades:

1. Crear Tablero con especificaciones
2. Rellenar Tablero
3. Determinar tamaño
4. Reasignar número de jugadas
5. Reasignar nombre de jugador
6. Determinar dificultad
7. Crear y manipular onclick events
8. Imprimir top 10 de puntajes
9. Reiniciar juego
10. Terminar juego
11. Guardar registro de puntaje de un usuario
12. Actualizar registros descendentemente (mayor a menor)

Iteración 2:

Se definieron conceptos que cumplen o son parte de las responsabilidades con sus debidos estereotipos y descripción

InterfazTaller4: Controller

Es el encargado de iniciar, reiniciar y terminar el juego

Panel Informacion: Information Holder

Contiene el número de movimientos de un jugador y su nombre

Panel Opciones: Information Holder

Contiene información que determina qué actividad se va a realizar, por ejemplo, la actividad de nuevo Juego, reiniciar Juego, Ver top10 y cambiar jugador.

Panel Tabla: Information Holder

Encargado de guardar el formato de la tabla según las especificaciones dadas desde la clase Panel Menu

Panel Menu: Information Holder

Contiene la información sobre las características de la interfaz, su tamaño y un índice de dificultad del juego

Calculadora de puntos: Controller

Calcula a partir de los movimientos del usuario, los puntos obtenidos en el juego

Manipulador de botones: Controller

Se encarga de actualizar y manipular los resultados de los botones presentes en Panel Menu

Creador de objetos: Controller

Crea los objetos que hacen parte del juego, es decir, cada casilla, usando Java2D y les da propiedades (tamaño, color, letra, etc)

Reflexión:

Ventajas:

Los information holder cubren 2 responsabilidades relacionadas entre sí, por lo que se facilita al no tener que implementar otras clases

Desventajas:

InterfazTaller4 tiene muchas responsabilidades asignadas, no es un modelo centralizado. Sus métodos dependen del funcionamiento de otras clases

Trade-offs:

InterfazTaller4 pasó a ser Controller porque al solo ser InformacionHolder, la clase Panel Tabla debía hacer todos los cálculos y por tanto, tener como atributos a las demás clases. Lo cual, era más complejo y extenso.

Proyecto

Iteración 3:

1. Crear, reiniciar y terminar el juego : InterfazTaller4 -> Controller
2. Crear, rellenar y actualizar Tablero con especificaciones : InterfazTaller4->Controller
3. Determinar tamaño y dificultad : Panel Menu -> Information Holder
4. Reasignar número de jugadas y a el jugador : Panel Informacion->Información Holder
5. Crear y manipular onclick events : Panel Tabla -> Information Holder
6. Imprimir top 10 de puntajes : InterfazTaller4 -> Controller
7. Guardar registro de puntaje de un usuario -> Clase del paquete modelo : Information Holder de Registros
8. Actualizar registros descendentemente (mayor a menor) -> Clase del paquete modelo : Controller

Restricciones:

* Debe crearse en Java2D
* La ventana del Top10 DEBE usar un elemento de tipo JList y tiene que mostrar los resultados de alguna manera personalizada (no puede ser la visualización por defecto de un JList).
* Tener listeners u otros manipuladores de eventos
* No modificar las clases del paquete modelo
* No pueden haber más elementos de lo solicitado

Diagrama de Clases

Diagrama

Descripción generada automáticamente