

ESCUELA COLOMBIANA DE INGENIERÍA

PROGRAMACIÓN ORIENTADA A OBJETOS

PROYECTO INICIAL Ciclo No. 2 2024-2

REFACTORING Y EXTENSIÓN

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el Problem F de la maratón de programación internacional 2023 **Tilting Tiles**. En este simulador se desea probar una un pegante especial, el **gummy glue**, que si se aplica sobre una baldosa hace que ésta y las baldosas vecinas se peguen.

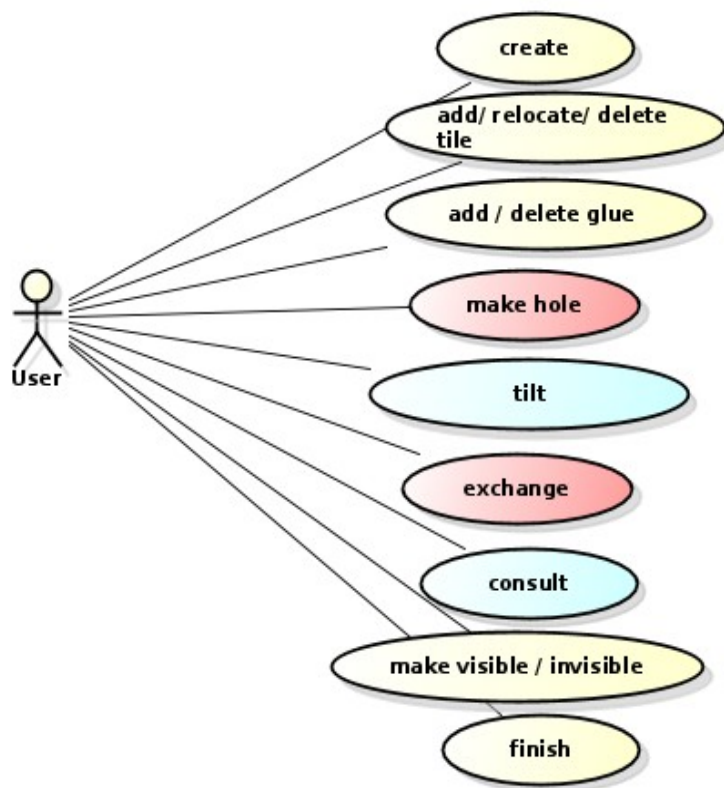
SEGUNDO CICLO

Los requisitos para el primer ciclo de desarrollo están indicados a continuación. Siempre hay un requisito implícito: el de **EXTENSIBILIDAD**.

MUY IMPORTANTE PLANIFICAR LOS MINI-CICLOS PARA ORIENTAR EL DESARROLLO
En esta entrega NO deben resolver el problema de la maratón sólo deben construir el simulador.

REQUISITOS FUNCIONALES

8. Debe permitir intercambiar los tableros de referencia y edición
9. Debe permitir ladear inteligentemente el tablero
10. Debe permitir agujerear una celda vacía
11. Debe permitir consultar las baldosas que no se pueden mover al ladear el tablero
12. Debe permitir consultar el número de baldosas que faltan por ubicar



- **exchange** Nuevo. Corresponde a requisito 8.
- **tilt**. Extensión. Corresponde a requisito 9.
- **make hole**. Nuevo. Corresponde a requisito 10
- **consult**. Extensión. Corresponde a los requisitos 11 y 12.

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

Puzzle
<pre>+ _ (h : int, w : int) : Puzzle + _ (ending : char[][]) : Puzzle + _ (starting : char[][], ending : char[][]) : Puzzle + addTile(row : int, column : int, color : String) : void + deleteTile(row : int, column : int) : void + relocateTile(from : int[], to : int[]) : void + addGlue(row : int, column : int) : void + deleteGlue(row : int, column : int) : void + makeHole(row : int, column : int) : void + tilt(direction : char) : void + tilt() : void + exchange() : void + isGoal() : boolean + actualArrangement() : char[][] + fixedTiles() : int[][][] + misplacedTiles() : int + makeVisible() : void + makeInvisible() : void + finish() : void + ok() : boolean</pre>

-
- Las posiciones deben retornarse ordenadas por filas columnas

REQUISITOS DE USABILIDAD

- Las celdas agujereadas deben lucir diferentes.
- Las baldosas que caen por los agujeros desaparecen.
- Las baldosas que no se pueden mover deben parpadear al hacerse la consulta, si el simulador está visible.

REQUISITOS DE ENTREGA

Los productos esperados para esta entrega son:

1. Diseño completo en la herramienta astah
2. Implementación siguiendo los estándares de codificación y documentación de java.
3. Casos de pruebas de unidad de su proyecto: **Puzzlecc2Test**
Las pruebas de unidad deben ser en modo invisible.
No olviden diseñar las pruebas considerando dos preguntas: ¿qué debería hacer? ¿qué no debería hacer?
4. Propuesta de casos de prueba para incluir en la clase **Puzzlecc2Test**
La clase **Puzzlecc2Test** será una creación colectiva usando el wiki correspondiente.
Los nombres de los casos de prueba deberán incluir la identificación de los autores. Por ejemplo, **accordingDAShould....** (DA: Iniciales de los primeros apellidos en orden alfabético).
5. Documento de retrospectiva del proyecto. (Ver ciclo uno)
Es necesario incluir la retrospectiva de este ciclo y de los anteriores.
6. No olviden preparar dos pruebas de aceptación para la presentación

Los productos los deben publicar en el espacio preparado en moodle en un archivo .zip con un nombre igual a la concatenación de los apellidos de los autores, ordenados alfabéticamente.

MUY IMPORTANTE DEFINIR EL ESTADO EN TÉRMINOS DE LOS MINI-CICLOS PLANIFICADOS

Publicar productos a revisión

:

Semana 06 21 de septiembre