

ESCUELA COLOMBIANA DE INGENIERÍA

PROGRAMACIÓN ORIENTADA A OBJETOS

PROYECTO INICIAL Ciclo No. 2 2019-01

REFACTORING Y EXTENSIÓN

PROYECTO INICIAL

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el **Problema E** de la maratón de programación internacional 2018 **Getting a Jump on Crime**.

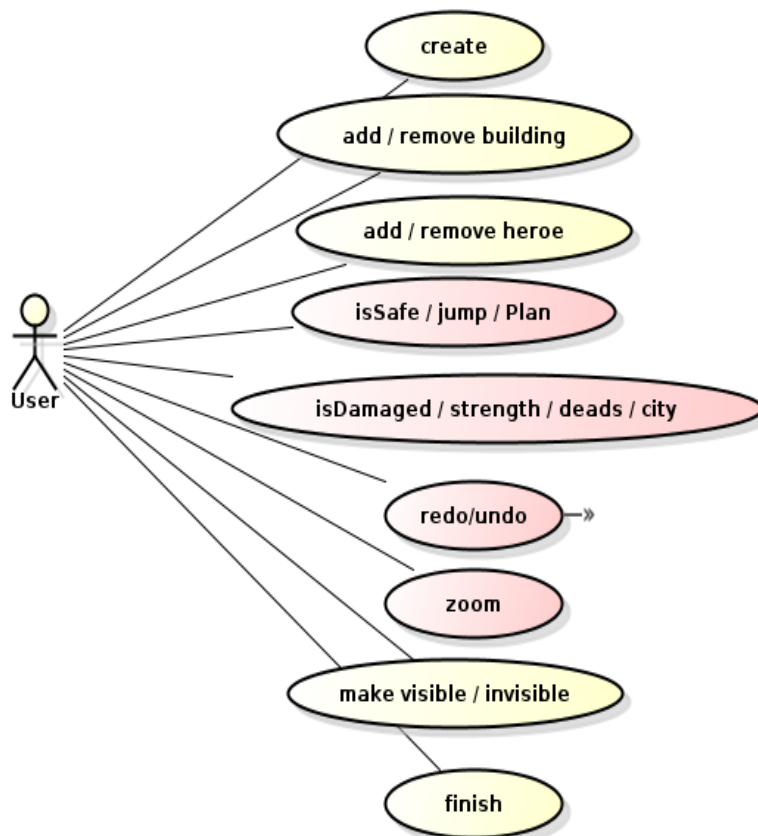
SEGUNDO CICLO

El objetivo de este ciclo es extender el simulador para cumplir nuevos requisitos y perfeccionarlo para garantizar la calidad del mismo considerando los criterios de corrección y extensibilidad.

En esta entrega NO deben resolver el problema de la maratón

REQUISITOS FUNCIONALES

8. Debe permitir planear y dar un “buen” salto
9. Debe permitir agrandar o achicar la representación
10. Debe permitir deshacer o rehacer la última acción
11. Debe permitir consultar la información de los elementos de la ciudad



- **isSafe/ jump /Plan.** Extensión. Corresponde al requisito 8
- **zoom.** Nuevo. Corresponde al requisito 10
- **redo/ undo** Nuevo, Corresponde al requisito 10
- **isDamage, strength, deads, city** Extensión, Corresponde al requisito 11

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

| CityOfHeroes |
|--|
| <pre>+ (width : int, height : int) : CityOfHeroes + addBuilding(x : int, width : int, height : int, hardness : int) : void + addHeroe(color : String, hidingBuilding : int, strength : int) : void + deads() : String[] + jump(heroe : String, velocity : int, angle : int, slow : boolean) : void + jump(heroe : String, building : int) : void + jump(heroe : String) : void + jumpPlan(heroe : String, building : int) : int[] + isDamaged(position : int) : boolean + isSafeJump(heroe : String, velocity : int, angle : int) : boolean + isSafeJump(heroe : String, building : int) : boolean + city() : int[][][] + removeBuilding(position : int) : void + removeHeroe(color : String) : void + strength(color : String) : int + makeVisible() : void + makeInvisible() : void + redo() : void + undo() : void + zoom(z : char) : void + finish() : void + ok() : boolean</pre> |

En el desarrollo deben respetar las decisiones de diseño presentes en este diagrama.

La consulta **city** debe retornar un vector con dos vectores. El primero con la información de los edificios [x, ancho, alto, dureza] ordenado por posición en x. El segundo con la información de los heroes [edificio, fortaleza] ordenado por el número de edificio.

El plan retorna un par (velocidad, ángulo)

El carácter del zoom es + o -

Los productos esperados para esta entrega son:

1. Diseño completo en la herramienta astah
Diagrama de clases con atributos y métodos privados y públicos.
En astah, crear un diagrama de clases (dejar el nombre por omisión)
(De las clases de **shapes** incluya únicamente el detalle (atributos - métodos) de lo que extendieron)
Diagrama de secuencia de cada uno de los métodos de la clase **CityOfHeroes**
En astah, crear los diagramas de secuencia en los métodos (dejar el nombre por omisión)
(Parar en los componentes de **shapes**)
2. Código siguiendo los estándares de documentación de java.
No olviden que el código de los métodos no debe ocupar más de una pantalla.
3. Propuesta de casos de prueba para clase **CityOfHeroesTestC2**.
Las pruebas se deben preparar en modo invisible.
No olviden diseñar las pruebas considerando dos preguntas: ¿qué debería hacer? ¿qué no debería hacer?
Las clase **CityOfHeroesTestC2** será una creación colectiva usando el wiki correspondiente.
Los nombres de los casos de prueba deberán incluir la identificación de los autores. Por ejemplo, **segunDA** debería (DA: Iniciales de los primeros apellidos en orden alfabético).

REQUISITOS DE ENTREGA

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. **Es necesario incluir la retrospectiva con las respuestas de los ciclos anteriores.**

Publicar productos a revisión

:

Semana 06 : Jueves 21 de febrero