

ESCUELA COLOMBIANA DE INGENIERÍA
PROGRAMACIÓN ORIENTADA A OBJETOS
PROYECTO
MACANA-ORJUELA

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el Problem C de la maratón de programación internacional 2017 Mission Improbable. En esta versión vamos a visualizar dos zonas: la zona de la bodega y la zona de plan de robo.

PRIMER CICLO

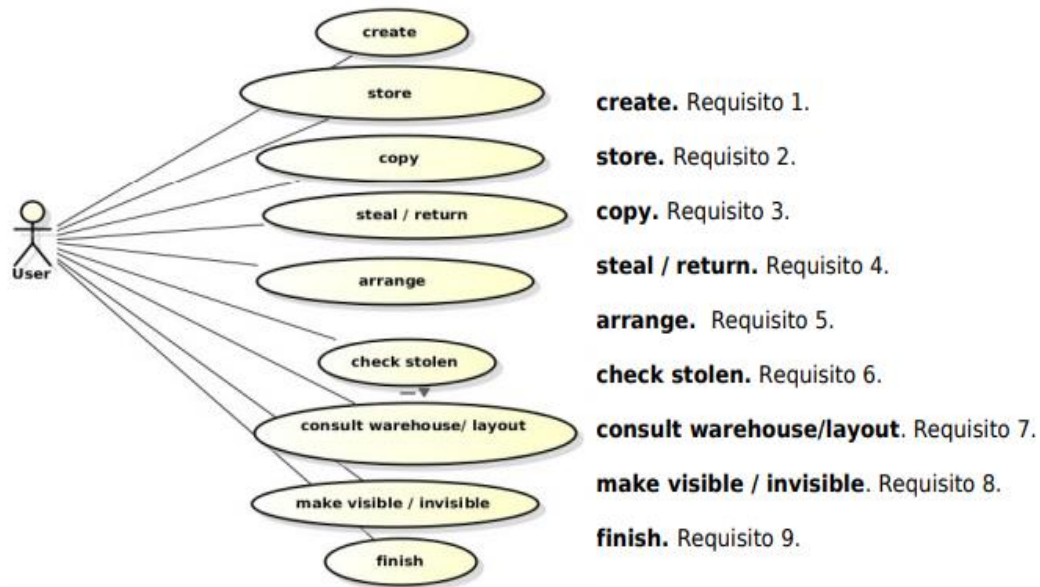
Los requisitos para el primer ciclo de desarrollo están indicados a continuación. No olviden que siempre hay un requisito implícito: el de EXTENSIBILIDAD.

REQUISITOS FUNCIONALES

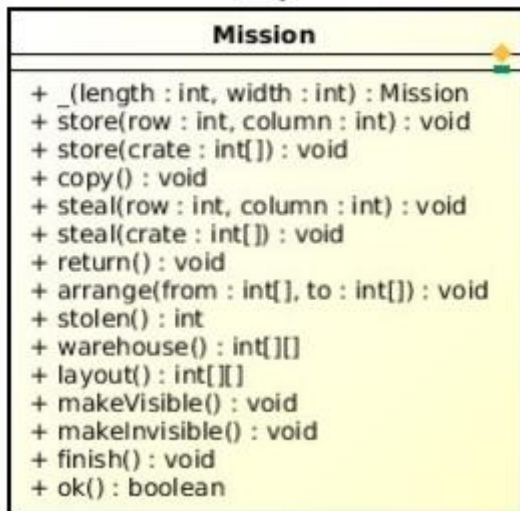
El simulador debe permitir:

1. Crear un juego
2. Almacenar una caja en la bodega
3. Copiar el estado actual de la bodega en la zona de plan de robo
4. Robar y devolver una caja en el plan de robo
5. Reorganizar una caja en el plan de robo
6. Consultar las piezas robadas en el último plan
7. Consultar las cajas que hay en bodega y en el plan de robo
8. Hacer visible o invisible el simulador
9. Terminar el simulador





REQUISITOS DE DISEÑO Y CONSTRUCCIÓN



REQUISITOS DE USABILIDAD

1. En las zonas de bodega y plan se deben presentar las imágenes de las tres cámaras.
2. Las zonas de bodega y plan deben ser de diferentes colores.
3. En la zona de plan se deben mostrar las cajas que han sido robadas.
4. La zona de plan debe tomar color rojo cuando sea diferente a la zona de bodega.
5. Si la acción no se puede realizar se le debe presentar un mensaje especial, sólo si el simulador es visible.
6. Las posiciones se dan en términos de (fila,columna) iniciando desde 1.

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

1. En su desarrollo debe respetar las decisiones de diseño presentes en este diagrama de clases para la clase principal. El método ok retorna si la última acción se pudo realizar o no.
2. Las clases se deben construir reutilizando los componentes del proyecto shapes que sean necesarios.
3. El paquete shapes puede ser extendido, si se requieren otras funcionalidades. Incluyan en la retrospectiva las extensiones y su justificación.
4. Las clases deben tener la documentación estándar de java. No olvidar revisar la documentación generada.
5. Las clases se deben construir en BlueJ. El nombre del nuevo proyecto debe ser Mission.

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.

RETROSPECTIVA CICLO 1

1. ¿Cuáles fueron los mini-ciclos definidos? Justifícalos.

1.Mission,,makeVisible , makeInvisible(Aquí creamos la matriz inicial de la bodega y del plan de robo para poder crear todo el juego,la visibilidad de las figuras de las cámaras, sus matrices)
store,store,copy(Almacenamos las cajas para poder actualizar la matriz de la bodega)
steal,steal,return,arrange, stolen(Aquí se roban las cajas y se devuelve el valor de las cajas robadas y en caso de tener que devolver la caja se puede realizar ya que ya contamos con los métodos anteriores.)
warehouse,layout,finish,ok(Este muestra las matrices y es la terminación del juego).

2. ¿Cuál es el estado actual del laboratorio en términos de mini-ciclos? ¿por qué?

Se realizaron todos los métodos menos el método store(crate) ya que no se logró entender la función de ese método, sin embargo, los demás métodos se realizaron y funcionan correctamente, y cuentan con su debida documentación.

3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)

30 horas

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Lograr implementar todos los métodos porque no entendíamos al inicio el funcionamiento del juego, una vez entendida la lógica del juego logramos implementarlo todo mostrando sus figuras y matrices.

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

La mayor dificultad que se nos presentó fue entender la lógica del juego, entender cómo funcionaba cada cámara, para resolverlo buscamos videos en youtube, investigamos en internet y recurrimos a nuestra docente para lograr aclarar las dudas.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Lo bueno que hicimos como equipo fue que alternamos roles para la elaboración del proyecto, escuchamos las opiniones, discutimos las ideas y nos ayudamos mutuamente. Nos comprometemos a preguntar con más tiempo de anterioridad las dudas para tener más tiempo para implementar las soluciones.

7. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

Programación a par debido a que logramos avanzar mucho más rápido con el proyecto trabajando de este modo y logramos resultados más eficaces.

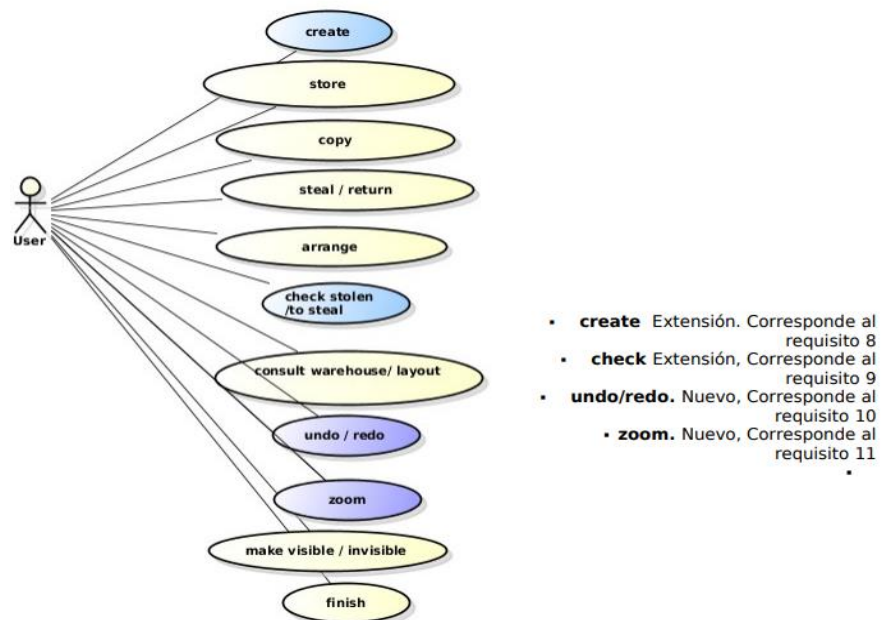
SEGUNDO CICLO

Los requisitos para el primer ciclo de desarrollo están indicados a continuación. No olviden que siempre hay un requisito implícito: el de EXTENSIBILIDAD.

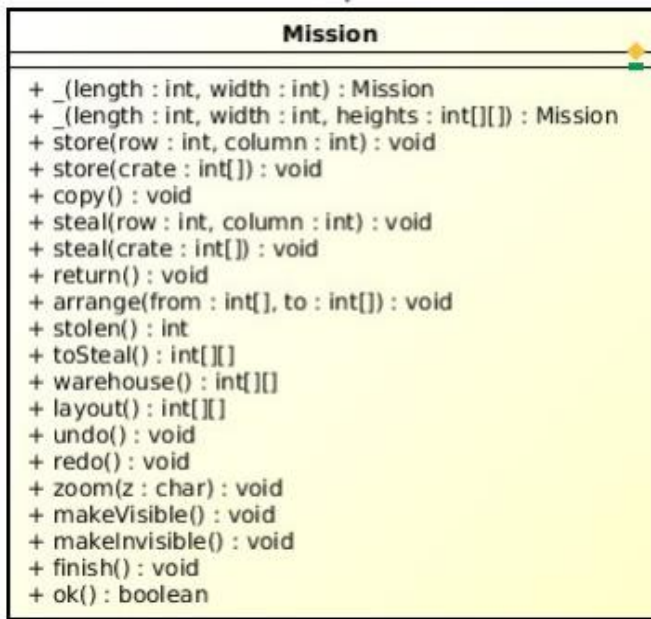
REQUISITOS FUNCIONALES

8. Debe permitir crear el simulador dado el plano de la bodega
9. Debe permitir consultar la posición de las cajas que se han robado en el plan de robo
10. Debe permitir deshacer y rehacer las acciones del simulador
11. Debe permitir agrandar o achicar las imagenes

CICLO 2



REQUISITOS DE DISEÑO Y CONSTRUCCIÓN



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

heights del segundo creador sigue las especificaciones del enunciado de la maratón. z en zoom es + o - Si se modifica la zona de la bodega se debe reiniciar la zona de plan de robo

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 Mission 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 MissionTestC2. 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 MissionTestC2 será una creación colectiva usando el wiki correspondiente. Los

nombres de los casos de prueba deberán incluir la identificación de los autores. Por ejemplo, segunDAdeberia (DA: Iniciales de los primeros apellidos en orden alfabético).

4. Documento de retrospectiva del proyecto. (Ver ciclo uno) Es necesario incluir la retrospectiva de este ciclo y de los anteriores.

5. No olviden preparar dos pruebas de aceptación para la presentación

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 de este ciclo y de los anteriores.

RETROSPECTIVA CICLO 2

1. ¿Cuáles fueron los mini-ciclos definidos? Justifícalos.

1.Mission,toSteal(Aquí solicitamos la matriz inicial de la bodega para poder crear todo el juego,y se pueden consultar las posiciones de las cajas robadas; se realizó en el miniciclo 1 porque ya teníamos los métodos del ciclo 1 para poder realizar lo.)

2.zoom,undo,redo(permite deshacer y rehacer las acciones del simulador, permite agrandar o achicar las imagenes)

2. ¿Cuál es el estado actual del laboratorio en términos de mini-ciclos? ¿por qué?

Se realizaron todos los métodos y funcionan correctamente, y cuentan con su debida documentación.

3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)

16 horas

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Lograr implementar todos los métodos y sus diagramas, porque tuvimos muchas dudas en sintaxis y en la lógica de esos métodos.

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

La implementación de los métodos undo y redo, consultar información en internet y buscar guía con los laboratorios ya resueltos.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Lo bueno que hicimos como equipo fue que alternamos roles para la elaboración del proyecto, escuchamos las opiniones, discutimos las ideas y nos ayudamos mutuamente. Nos comprometemos a preguntar con más tiempo de anterioridad las dudas para tener más tiempo para implementar las soluciones.

7. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

UnitTest porque logramos entender la importancia de implementar primero las pruebas, y cómo estas pruebas pueden ayudarnos a entender mejor el proyecto e implementar nuestro código de una manera más sencilla al observar las pruebas.