

Trabajo Práctico Pacman 75.10

Tercera Iteración

[1 Implementación pedida](#)

[1.1 Interfaz Gráfica](#)

[1.2. Juego \(Vidas, Puntos, Niveles\)](#)

[1.3. Frutas](#)

[1.4. Test unitarios](#)

[2. Aclaraciones sobre la implementación](#)

[3. Criterios de corrección](#)

[4. Herramientas a utilizar](#)

[5. Sobre la entrega](#)

Implementación del Pacman 75.10 - Tercera iteración

1 Implementación pedida

Se pide extender el modelo de dominio realizado en la iteración 1 y 2 para soportar los siguientes requerimientos

1.1 Interfaz Gráfica

Se requiere agregar una UI que tome los movimientos del pacman de la misma, y visualice el estado del laberinto en todo momento.

Los laberintos se siguen levantando de archivos XML para su definición, pero ya no son necesarios los xmls de salida generados luego de cada iteración.

1.2. Juego (Vidas, Puntos, Niveles)

Se solicita agregar 3 vidas al pacman, llevar el puntaje, mínimamente dando punto a cada bolita, a los bolones, y por cada fantasma comido.

Definir en constantes o configuración los valores, tiempos, lógica etc que requieran para definir cuestiones no claramente definidas y requeridas para el juego y la UI.

Al terminar de comer las bolitas del laberinto, se vuelve a continuar en el siguiente laberinto. Los laberintos se modelan en el formato xml definido, los mismos se muestran como distintos niveles en forma secuencial, si hay uno, siempre el mismo, si hay más de uno siempre en forma secuencial y cíclica entre ellos, hasta que le pacman pierda todas su vidas y ahí finalice el juego.

1.3. Frutas

Aparece otro componente que es la fruta, la misma cada X tiempo aparece en la pantalla y se mueve por el laberinto a velocidad de 1. Al comerla da Z puntos. Siempre avanza, al llegar a un cruce elige uno de los posibles caminos de salida en forma random.

1.4. Test unitarios

Se piden un set de test unitarios sobre el modelo de dominio descrito.

El set de test unitarios que se desarrolle debe representar un código valioso, debe motivar al equipo de trabajo a mantenerlo en el tiempo por la utilidad que brinda. Debe seguir las buenas prácticas de implementación de test unitarios.

2. Aclaraciones sobre la implementación

Queda fuera de la iteración los siguientes requerimientos:

- Lógica de cómo un fantasma muerto, regresa al inicio (como se desplazaría de muerto a su lugar de inicio) Directamente desaparece y aparece cuando revive del punto de inicio.
- Otras estrategias de los fantasmas cazadores de cómo atacar al pac-man (si eligen el camino más corto o más largo, si lo encierran en grupo, etc)
- En caso de tomar una decisión de implementación que esté fuera del alcance descrito, esta no debe contradecir ningún requerimiento pedido y se debe enunciar en el informe la hipótesis que la justifique, las mismas deben validarse con el ayudante.

3. Criterios de corrección

- El diseño del dominio
- El diseño de código
- Los test unitarios
- Cumplir con toda la funcionalidad descrita, ambiente y buenas prácticas definidas.

- El informe completo. Carátula, índice, enunciado, justificaciones y decisiones para la resolución acompañada si es necesario de diagramas UML con descripciones orientadas a los contenidos de la materia, extractos de códigos de ejemplo, y conclusiones.

4. Herramientas a utilizar

- Maven ≥ 2
- JUnit 4.xx
- Repositorio Git

5. Sobre la entrega

Se debe etiquetar en el repositorio la entrega antes de la fecha definida.