

Trabajo Práctico Pacman 75.10

Primera Iteración

[1 Implementación pedida](#)

[1.1. Modelo de dominio](#)

[1.2. Test unitarios](#)

[2. Aclaraciones sobre la implementación](#)

[3. Criterios de corrección](#)

[4. Herramientas a utilizar](#)

[5. Sobre la entrega](#)

Implementación del Pacman 75.10 - Primera iteración

1 Implementación pedida

1.1. Modelo de dominio

En esta primera iteración se pide diseñar e implementar solamente un fantasma del juego, contemplando el tiempo que transcurre que afecta el comportamiento del fantasma.

Se pide modelar dejando de lado temas de performance por el momento.

El fantasma del juego está caracterizado por poder estar:

- Muerto
- Como fantasma cazador, con niveles de agresividad
- Como fantasma Presa

El fantasma se puede mover, puede ser comido, puede incrementar su ira. Por naturaleza al comienzo es cazador y puede ser convertido en presa en cualquier momento. Siendo cazador, cada X1, X2 y X3 intervalos incrementa su ira, luego de X3 permanece en ese nivel de ira o agresividad.

Si un fantasma esta muerto no puede incrementar su ira. No puede ser comido ni tampoco puede ser convertido en presa. Al morirse debe reiniciar el nivel de agresividad y en un tiempo X4 debe volver a “revivir” como cazador para comenzar el ciclo.

Si un fantasma es cazador no puede ser comido, puede incrementar su nivel de agresividad (hasta un máximo) y puede ser convertido en presa.

Si un fantasma es presa puede ser comido (y debe estar muerto luego de ser comido), no puede incrementar su nivel de agresividad y no puede ser convertido a presa. Al cabo de un tiempo x (ver aclaraciones) debe volver a un naturaleza cazador.

No se pide nada de implementación de interfaz gráfica ni ninguna otra funcionalidad del juego que la anteriormente descrita.

1.2. Test unitarios

Se piden un set de test unitarios sobre el modelo de dominio descrito en el punto 1.1

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.

Se debe contemplar un conjunto amplio y abarcativo de casos de prueba sobre el modelo de dominio.

2. Aclaraciones sobre la implementación

Si bien las siguientes etapas contemplaran como darle inteligencia de robot al fantasma, meterlo en el laberinto clásico del juego con sus bolitas (bolones y frutas) y agregar el Pacman, para esta entrega solo se pide implementar del dominio del Pacman descrito en el punto 1.1, permitiendo con los tests validar las reglas del mismo.

- Todos los ticks o intervalos de tiempo, transiciones, etc, deben ser configurables.

3. Criterios de corrección

- El diseño del dominio
- El diseño de código
- Los test unitarios
- Cumplir con toda la funcionalidad descrita
- El informe completo. Carátula, índice, enunciado, 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 entregar en un único archivo comprimido, el informe completo junto al repositorio local git en el que trabajaron (incluyendo el directorio .git), a través del campus.