[75.07 / 95.02]

Algoritmos y programación III

Trabajo práctico 2 (enunciado)

Segundo cuatrimestre del año 2019

Índice

1. Objetivo

Desarrollar una aplicación de manera grupal aplicando todos los conceptos vistos en el curso, utilizando un lenguaje de tipado estático (Java) con un diseño del modelo orientado a objetos y trabajando con las técnicas de TDD e Integración Contínua.

2. Consigna general

Desarrollar la aplicación completa, incluyendo el **modelo** de clases e interfaz gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño.

3. Especificación de la aplicación a desarrollar

La aplicación consiste en un videojuego por turnos, de dos jugadores conformado de un tablero y distintas unidades sobre él. El objetivo del juego es destruir todas las unidades enemigas y gana el jugador que lo consigue primero.

El jugador dispone de distintas entidades para acomodar en el tablero, pero tiene una cantidad limitada de puntos para usar (cada entidad tiene un coste distinto de puntos dependiendo de su nivel).

El objetivo del juego es vencer al ejército enemigo (son entidades preestablecidas en el tablero que no pueden ser modificadas) acomodando las entidades aliadas en el tablero.

3.1. Entidades

- **Jugador:**
  Cada jugador dispone de 20 puntos para gastar en las entidades que quiera.

- **Tablero :**
  Es el campo donde se posicionan las unidades antes de comenzar la partida. Es cuadrado de 20 casilleros de lado. Estará dividido en 2 sectores.
  - Sector aliado:
  Es el sector en el cual se pueden acomodar las piezas del jugador actual.
  -Sector enemigo:
  Es el sector en el cual se posicionan las piezas del jugador contrincante.

  Cada jugador puede acomodar las piezas únicamente en su lado del tablero.

  -Penalización por estar en territorio enemigo:
  Tanto las unidades aliadas como enemigas sufrirán un 5% más de daño si están sobre territorio del otro bando.

- **Unidades :**

  - Todas las unidades se pueden desplazar de a un casillero por turno
  - Se puede mover una entidad por turno.

Soldado de infantería	
Costo	1
Atributos	- Vida: 100 - Daño cuerpo a cuerpo: 10 - Daño a distancia: 0
Comportamiento	- Puede atacar a un enemigo a corta distancia. - Si hay más de 3 Soldados contiguos (en cualquier dirección) se comportan como un **Batallón** y PUEDEN moverse los 3 al mismo tiempo en el mismo turno. [Esto significa que cada uno de los soldados se va a mover en la dirección solicitada. En caso que uno no pueda moverse al casillero, únicamente ese Soldado se quedará quieto, y los demás si se moverán]

Jinete	
Costo	3
Atributos	- Vida: 100 - Daño cuerpo a cuerpo: 5 - Daño a distancia: 15
Comportamiento	- Si hay al menos un Soldado de Infantería aliado cerca o no hay ningún enemigo cerca, su arma de ataque es un Arco y Flecha y únicamente puede atacar a enemigos en distancia media.. - Si no hay ningún aliado cercano y hay enemigos cercanos , su arma de ataque es una Espada y únicamente puede atacar a enemigos en distancia corta.

Curandero	
Costo	2
Atributos	- Vida: 75 - Curación: 15
Comportamiento	- Puede curar a una unidad Aliada (menos a la Catapulta) en una distancia cercana.

Catapulta	
Costo	5

Atributos	- Vida: 50 - Daño cuerpo a cuerpo: 0 - Daño a distancia: 20
Comportamiento	- No puede moverse en toda la partida. - Ataca en una distancia lejana únicamente. [Puede dañar tanto a Enemigos como Aliados] - Causa daño a la primera unidad enemiga alcanzada, y a todas las unidades directamente contiguas, y si a su vez la segunda unidad tiene otra unidad contigua, también causa el mismo daño (y así sucesivamente)

3.2. Flujo del programa

- Fase inicial (una por cada jugador): Cada jugador selecciona su nombre y posiciona en su lado del tablero las entidades que desee y dura hasta que lo anuncia el jugador.
- Fase de juego: Cuando ambos jugadores finalizaron su Fase inicial, se inicia ésta fase, la cual dura hasta que alguno de los jugadores gana. Consta de un turno por vez por usuario. El usuario que inicia es seleccionado de manera aleatoria.

3.3. Distancias

- Distancia Cercana: consta de una distancia de 1 a 2 casilleros (en cualquier dirección)
- Distancia Media: consta de una distancia de 3 a 5 (en cualquier dirección)
- Distancia Lejana: consta de una distancia de 6 a Infinito (en cualquier dirección)

4. Interfaz gráfica

La interacción entre el usuarios y la aplicación deberá ser mediante una interfaz gráfica intuitiva. Consistirá en una aplicación de escritorio utilizando **JavaFX** y se pondrá mucho énfasis y se evaluará como parte de la consigna su **usabilidad**.

5. Herramientas

1. **JDK (Java Development Kit)**: Versión 1.7 o superior que incluya JavaFX.
2. **JUnit**: Framework de pruebas unitarias para Java.
3. **IDE (Entorno de desarrollo integrado)**: Su uso es opcional y cada integrante del grupo puede utilizar uno distinto o incluso el editor de texto que más le guste. Lo importante es que el repositorio de las entregas no contenga ningún archivo de ningún IDE y que la construcción y ejecución de la aplicación sea totalmente independiente del entorno de desarrollo. Algunos de los IDEs más populares son:
    a. Eclipse
    b. IntelliJ
    c. Netbeans

4. **Herramienta de construcción:**: Se deberán incluir todos los archivos XML necesarios para la compilación y construcción automatizada de la aplicación. El informe deberá contener instrucciones acerca de los comandos necesarios (preferentemente también en el archivo README.md del repositorio). Puede utilizarse Apache Ant con Ivy o Maven.
5. **Repositorio**: Todas las entregas deberán ser subidas a un repositorio único en GitHub para todo el grupo en donde quedarán registrados los aportes de cada miembro. El repositorio puede ser público o privado. En caso de ser privado debe agregarse al docente corrector como colaborador del repositorio.
6. **Git**: Herramienta de control de versiones
7. **Herramienta de integración contínua**: Deberá estar configurada de manera tal que cada *commit* dispare la compilación, construcción y ejecución de las pruebas unitarias automáticamente. Algunas de las más populares son:
    a. Travis-CI
    b. Jenkins
    c. Circle-CI

6. Entregables

Para cada entrega se deberá subir lo siguiente al repositorio:

1. Código fuente de la aplicación completa, incluyendo también: código de la pruebas, archivos de recursos.
2. Script para compilación y ejecución (Ant o Maven).
3. Informe, acorde a lo especificado en este documento (en las primeras entregas se podrá incluir solamente un enlace a Overleaf o a Google Docs en donde confeccionen el informe e incluir el archivo PDF solamente en la entrega final).

No se deberá incluir ningún archivo compilado (formato .class) ni tampoco aquellos propios de algún IDE (por ejemplo .idea). Tampoco se deberá incluir archivos de diagramas UML propios de alguna herramienta. Todos los diagramas deben ser exportados como imágenes de manera tal que sea transparente la herramienta que hayan utilizado para crearlos.

7. Formas de entrega

Habrá **4 entregas formales** que tendrán una calificación de **APROBADO o NO APROBADO** en el momento de la entrega. Además se contará con una entrega 0 preliminar.

Aquél grupo que acumule 2 no aprobados, quedará automáticamente desaprobado con la consiguiente pérdida de regularidad en la materia de **todos** los integrantes del grupo. *En cada entrega se deberá incluir el informe actualizado (preferentemente impreso).*

8. Evaluación

El día del vencimiento de cada entrega, cada ayudante convocará a los integrantes de su grupo, solicitará el informe correspondiente e iniciará la corrección mediante una entrevista grupal.

Es imprescindible la presencia de todos los integrantes del grupo el día de cada corrección.

Se evaluará el trabajo grupal y a cada integrante en forma individual. El objetivo de esto es comprender la dinámica de trabajo del equipo y los roles que ha desempeñado cada integrante del grupo. Para que el alumno apruebe el trabajo práctico debe estar aprobado

en los dos aspectos: grupal e individual (se revisarán los *commits* de cada integrante en el repositorio).

Dentro de los ítems a chequear el ayudante evaluará aspectos formales (como ser la forma de presentación del informe), aspectos funcionales: que se resuelva el problema planteado y aspectos operativos: que el TP funcione integrado.

9. Casos de prueba para cada entrega

Se sobreentiende que cada entrega consta de las **pruebas + el código** que hace pasar dichas pruebas).

Los alumnos del curso 2 que tengan correctores asignados el martes a la noche tendrán la entrega presencial los días martes.

Entrega 1 (jueves 7 de Noviembre)

- Pruebas entidades:
    - Una Unidad movible se puede mover en todas las direcciones.
    - Una Unidad movible no puede moverse a un casillero ocupado
    - Un Soldado de infantería aliado ataca a una pieza enemiga y se verifica que se resta la vida correspondiente.
    - Un Jinete aliado ataca a una pieza enemiga y se verifica que se resta la vida correspondiente.
    - Una catapulta aliada ataca a una pieza enemiga y se verifica que se resta la vida correspondiente.
    - Un Curandero aliado cura a una pieza aliada y se verifica que se suma la vida correspondiente.
- Pruebas tablero:
    - Se coloca una pieza aliada en un casillero del sector aliado vacío con éxito.
    - Se verifica que no se puede colocar una pieza aliada en un casillero del sector aliado ocupado.
    - Se verifica que no se puede colocar una pieza aliada en un casillero del sector enemigo.
    - Correcta creación e inicialización del tablero.
- Pruebas Jugador:
    - Se verifica que no puede tomar más entidades de lo que sus puntos le permiten.
    - Se verifica que el jugador que se queda sin entidades, es el perdedor.

Entrega 2 (jueves 21 de noviembre)

- Pruebas entidades:
    - Soldado:
        - se verifica que 3 soldados contiguos pueden moverse al mismo tiempo en la misma dirección con una sola acción.
        - Teniendo 3 soldados contiguos, y un obstáculo (una entidad distinta a los otros dos soldados) obstruyendo a uno de los 3, se verifica que al mover el Batallón, se mueven 2 soldados y uno se queda quieto.
        - (En la situación anterior) Se verifica que el Batallón se disuelve, al quedar separados los 3 soldados.

- habiendo 4 soldados contiguos, se verifica que al mover un Batallón se mueven únicamente 3.
- Jinete:
  - Un jinete sin aliados en distancia corta y un enemigo en distancia corta, ataca con su espada al enemigo y se verifica que se realiza correctamente el ataque.
  - Un jinete sin aliados en distancia corta y un enemigo en distancia corta y otro enemigo en distancia media, trata de atacar al enemigo en distancia media y se verifica que no se puede realizar el ataque

Entrega 3 (jueves 28 de noviembre)

1. Interfaz gráfica inicial básica: comienzo del juego y visualización del tablero e interfaz de usuario básica.
2. Modelo del manejo de turnos en el juego.

Entrega 4 - Final: (jueves 5 de diciembre)

Trabajo Práctico completo funcionando, con interfaz gráfica final, sonidos e informe completo.

Pre-entregas: (jueves 31 de octubre y 14 de noviembre)

Las pre-entregas no son obligatorias, sirve para ir validando el avance así como resolver dudas que surjan durante el desarrollo del trabajo. Es recomendable que los equipos se presenten.

Tiempo total de desarrollo del trabajo práctico:

4 semanas + 2 pre entregas

10. Informe

El informe deberá estar subdividido en las siguientes secciones:

Supuestos

Documentar todos los supuestos hechos sobre el enunciado. Asegurarse de validar con los docentes.

Diagramas de clases

Varios diagramas de clases, mostrando la relación estática entre las clases. Pueden agregar todo el texto necesario para aclarar y explicar su diseño de manera tal que logre el modelo logre comunicarse de manera efectiva.

Diagramas de secuencia

Varios diagramas de secuencia, mostrando la relación dinámica entre distintos objetos planteando una gran cantidad de escenarios que contemplen las secuencias más interesantes del modelo.

Diagrama de paquetes

Incluir un diagrama de paquetes UML para mostrar el acoplamiento de su trabajo.

Diagramas de estado

Incluir diagramas de estados, mostrando tanto los estados como las distintas transiciones para varias entidades del modelo.

Detalles de implementación

Deben detallar/explicar qué estrategias utilizaron para resolver todos los puntos más conflictivos del trabajo práctico. Mencionar qué patrones de diseño fueron utilizados y por qué motivos.

Excepciones

Explicar las excepciones creadas, con qué fin fueron creadas y cómo y dónde se las atrapa explicando qué acciones se toman al respecto una vez capturadas.