

Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Practica de Laboratorio 18:
Definición de Clases de Usuario
Herencia

I

OBJETIVOS

- Que el alumno demuestre poder crear “clases definidas por el programador”
- Implementar métodos para las clases definidas por el programador
- Utilizar el mecanismo de Herencia de clases

II

ACTIVIDADES

- En la historia, los ejércitos estaban conformados por diferentes tipos de soldados, que tenían similitudes, pero también particularidades.
- Basándose en la clase Soldado crear las clases Espadachín, Arquero y Caballero. Las tres clases heredan de la superclase Soldado, pero aumentan atributos y métodos propios, o sobrescriben métodos heredados.
- Los espadachines tienen como atributo particular "longitud de espada" y como acción "crear un muro de escudos" que es un tipo de defensa en particular.
- Los caballeros pueden alternar sus armas entre espada y lanza, además de desmontar (sólo se realiza cuando está montando e implica defender y cambiar de arma a espada), montar (sólo se realiza cuando está desmontado e implica montar, cambiar de arma a lanza y investir). El caballero también puede investir, ya sea montando o desmontando, cuando es desmontado equivale a atacar 2 veces, pero cuando está montando implica a atacar 3 veces.
- Los arqueros tienen un número de flechas disponibles las cuales pueden dispararse y se agotan cuando lo hacen.
- Basarse en los laboratorios anteriores.
- Realizar el diagrama de clases de UML.
- Tendrá 2 Ejércitos que pueden ser constituidos sólo por espadachines, caballeros y arqueros (usar una estructura de datos por cada tipo de soldado). Cada ejército tendrá n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Espadachin0X1, Arquero1X1, Caballero2X2, etc., un

valor de puntos de vida autogenerado aleatoriamente: Espadachín [3..4], Arquero [1..3] y Caballero [3..5], la fila y columna también autogenerados aleatoriamente (no puede haber 2 soldados en el mismo cuadrado) y valores autogenerados para el resto de atributos. Mostrar el tablero, distinguiendo los ejércitos y los tipos de soldados creados.

- El juego es humano contra humano y consistirá en mover un soldado por cada turno de cada jugador. Se puede mover en cualquier dirección, Ud. deberá darle la coordenada del soldado a mover y la dirección de movimiento, el programa deberá verificar que hay un soldado del ejército que tenga el turno en dicha posición y que el movimiento es válido (no puede haber 2 soldados del mismo ejército en el cuadrado), pidiendo ingresar nuevos datos si no es así. Cuando un soldado se mueve a una posición donde hay un soldado rival, se produce una batalla y gana el soldado basado en la siguiente métrica: la sumatoria de los niveles de vida de los dos soldados que se enfrentan son el 100% y se le debe dar la probabilidad correspondiente de vencer para cada soldado (ejemplo S1:5 S2:3, las probabilidades de vencer serían S1:62.5% E2:37.5%) y de acuerdo a dichas probabilidades se decidirá el ganador. El ganador ocupará dicho cuadrado (como bono, se le aumentará el nivel de vida actual en 1) y el perdedor desaparecerá. Para cada batalla se deberá explicar por qué ganó uno de los soldados. Gana el juego quien deje al otro ejército sin soldados. Después de cada movida se deberá mostrar el Mapa en su estado correspondiente. Hacer un programa iterativo.



	A	B	C	D	E	F	G	H	I	J
1							E3			
2		A3		A1						
3							E3			A1
4				L2						
5										
6	C5					L1		L2		
7				E4						
8	C5									
9										
10						E4				A1