

Introducción a la Programación de Videojuegos

Evaluación II

Consigna

• Crear una clase que representará a un arma. Se propone que el diagrama de la clase sea el siguiente:

Weapon
Private:
<ul style="list-style-type: none">• <i>string</i> name• <i>float</i> attack
Public:
<ul style="list-style-type: none">• <i>Weapon</i> (<i>string</i> name, <i>float</i> attack)• <i>abstract float</i> GetAttackDamage (<i>AttackType</i> attackType, <i>ref bool</i> isCrit)

- Crear clases derivadas de *Weapon* para los siguientes tipos de arma:
 - *Sword*.
 - *Spear*.
 - *Axe*.

Hacer que cada una de ellas tenga un distinto modo de calcular el daño, haciendo que cada una maneje sus propios valores de *critical rate* y *critical damage*.

- Crear una clase que representará a una armadura. Se propone que el diagrama de la clase sea el siguiente:

Armor
Private:
<ul style="list-style-type: none"> • <i>string</i> name • <i>float</i> defense • <i>float</i> weight
Public:
<ul style="list-style-type: none"> • Armor (<i>string</i> name, <i>float</i> defense, <i>float</i> weight) • <i>float</i> MitigateDamage (<i>float</i> damage)

- Crear una clase que representará a un guerrero. Se propone que el diagrama de la clase sea el siguiente:

Warrior
Private:
<ul style="list-style-type: none"> • <i>string</i> name • <i>Weapon</i> weapon • <i>Armor</i> armor • <i>float</i> maxHealth • <i>float</i> currentHealth
Public:
<ul style="list-style-type: none"> • Warrior (<i>string</i> name, <i>float</i> maxHealth) • <i>void</i> SetWeapon (<i>Weapon</i> weapon) • <i>void</i> SetArmor (<i>Armor</i> armor) • <i>string</i> GetName () • <i>float</i> ReceiveDamage (<i>float</i> damage) • <i>float</i> Attack (<i>Warrior</i> warrior, <i>AttackType</i> attackType, <i>ref bool</i> isCrit) • <i>float</i> GetCurrentHealth () • <i>bool</i> IsAlive ()

- Crear una clase que sirva para crear a un guerrero. La misma debe estar compuesta por un guerrero, un arma y una armadura. Debe tener un método que permita dicha creación, pudiéndose elegir:
 - Nombre del guerrero.
 - Tipo de arma (lanza, espada o hacha).
 - Tipo de armadura (ligera, mediana, pesada).
- Crear una clase que sirva para manejar la lógica del juego. El mismo debe crearse con dos guerreros (jugadores 1 y 2), que se pasan como parámetros en el constructor. La clase debe mantener la cuenta del número de rondas, y por cada una debe permitir la realización de los ataques de los jugadores. Antes de atacar, se debe preguntar al jugador qué tipo de ataque se quiere realizar (rápido, normal o cargado). Dependiendo del tipo de ataque realizado, la acción deberá tener un bonus de chances de daño crítico o del daño crítico en sí. El ciclo de pelea se deberá repetir hasta que alguno de los personajes haya muerto. Indicar siempre el número de ronda actual. Además, entre cada ataque, mostrar:
 - Quién atacó a quién, y cuánto daño le hizo.
 - Si el ataque resultó en un crítico.
 - La vida restante del jugador atacado.
- El programa deberá iniciar con el creador de personajes. Hacer dos de éstos creadores, uno para cada jugador. Luego, crear a un objeto de la clase del juego, pasando como parámetros a los jugadores creados. Después, llamar a un método *“play”* que inicie el *loop* de batalla por rondas. Al final del ciclo, se debe mostrar el jugador ganador.

Aspectos a considerar

- **Cálculo de daño:** incluir un cálculo que haga que, según la defensa de la armadura, se mitigue hasta una cierta cantidad de daño recibido por el portador de la armadura. Hacer que exista un máximo posible de porcentaje mitigado (por ejemplo, 80%), con la finalidad de hacer que todo ataque siempre reste algo de vida. Hacer que la cantidad de daño mitigado sea proporcional a ese tope (si con una defensa igual al daño que se intenta

realizar se reduce el mismo en un 80%, pensar en cuánto se mitiga con el valor actual de defensa).

- **Daño crítico:** para sacar cuánto daño hace un jugador en un ataque, considerar los valores de daño crítico. Si las chances de crítico son de 70%, y el daño crítico es del 100%, hay una chance de 7 en 10 de realizar el doble de daño.
- **Tipos de ataque:** un ataque rápido podría otorgar más chances de daño crítico, pero nada de daño crítico adicional; un ataque cargado podría tener bonus de daño crítico, pero sin añadir chances de crítico al arma.
- **Creación de guerreros:** el creador deberá ser el encargado de crear a los guerreros y su equipamiento; usar un método específico de creación (*"CreateWarrior"*). Las armas deben ser creadas con *stats* determinados por la clase que crea a los personajes.

Criterios de Evaluación

- La resolución de la consigna base otorgará la condición básica de aprobación: **4 (cuatro)**.
- Agregar la aplicación eficiente de conceptos vistos y sugeridos (relaciones entre clases, encapsulamiento, y demás) y la prolijidad del código (el buen formato, nomenclatura de variables) otorgarán la condición de aprobación destacable: **7 (siete)**.
- Agregar la implementación apropiada de aquellas clases sobre las cuales no se otorgó un diagrama otorgará la condición de aprobación máxima: **10 (diez)**.