STUCOM
Centre d'Estudis
www.stucom.com

Homologat i concertat per
la Generalitat de Catalunya

Dept. d'Informàtica i Telecomunicacions	Curs 2018-2019
Grup: DAM2T	
M09 - Serveis i procesos - U F 1	
Nom professor/a: Daniel Moreno	

Examen - DeepDungeonsOfDoom

Estructura:

La práctica a realizar consiste en un juego RPG. Este juego constará de tres partes: **Dungeon**, **Inventory** y **Shop**.

Dungeon: El usuario podrá ir a pelear contra los monstruos que residen en el dungeon. Cada monstruo tiene un nivel de ataque que le permite utilizar más o menos dados en su tirada. Además, cada monstruo ofrece cierto nivel de experiencia y monedas. Por lo tanto, según el nivel de nuestro personaje deberemos tener en cuenta a que monstruo vamos a enfrentarnos. Si durante la batalla conseguimos vencer a nuestro enemigo obtendremos la recompensa y experiencia. Al contrario, si nos acaba matando deberemos empezar el juego de nuevo.

Inventory: Cada personaje tendrá un Stuff asociado a su perfil: guerrero, arquero o mago. Cómo es obvio un mago inicialmente tendrá un cetro para atacar, un arquero un arco… y así con todos los ítems de su inventario. En la sección de Shop veremos que estos ítems se pueden cambiar y adquirir otros, por lo tanto, podríamos tener un mago que utiliza una espada. El inventario nos permite ver que atributos conseguimos en base a los ítems que tenemos, además de nuestra vida, experiencia o monedas disponibles. Cuando un personaje realice una compra en la tienda, el inventario deberá cambiar al ítem escogido y reflejar los cambios en sus stats.

Shop: En este apartado el personaje podrá ver todos los ítems disponibles para comprar. En función de las monedas que disponga podrá comprar mejores o peores ítems. Cuando compre un ítem se le ha de restar el importe de su bolsa y añadir este ítem a su stuff para que en el apartado Inventory se vea reflejado el nuevo ítem.

Dinámica:

Un **personaje** tiene una vida, nombre, dinero, exp y un conjuto de ítems que podríamos agrupar en algo que llamamos "Stuff". Los ítems le ayudan a calcular sus habilidades. Estas habilidades se calculan con la suma de los diferentes atributos que tienen los ítems. Por lo tanto, gracias a estos ítems podremos saber los atributos de : Ataque (ATQ), Defensa (DEF), Magia (MAG), Suerte (LCK).

Un **ítem** cuesta un dinero y se compone de diferentes propiedades: Ataque (ATQ), Defensa (DEF), Magia (MAG), Suerte (LCK).

Un **monstruo** es un ser que tiene un nombre, ataque vida y si lo derrotamos nos dará dinero y experiencia para el héroe.

Combate:

Los héroes y los monstruos lanzarán dados en base a un cálculo para el combate de rangos. Un ejemplo:

Daño <= 10 -> 1 dado
Daño > 10 y <= 25 -> 2 dados
Daño > 25 -> 3 dados

Según los parámetros que tenga el personaje obtendremos un resultado. Los monstruos tendrán en cuenta para su cálculo el ataque, pero los supervivientes la suma de ataque, defensa, magia y suerte divido entre 4. Ejemplo:

Ataque : 3 ; Defensa: 9 ; Magia: 1 ; Suerte: 5
(3 + 9 + 1 + 5) / 4 = 6

En este caso nuestro héroe tiraría 1 dado, parece poco pero… hay que ser un valiente.

Nota: Tenéis que pensar que cuando diseñáis un juego, el usuario debe tener cierto riesgo de perder, por ello si el damos muchas facilidades al principio, existe mucha probabilidad de que vea muy sencillo nuestro juego y no consigamos llamar su atención y "picarlo" a querer superarse. Por eso, os dejo elegir a vosotros en parte los valores de los ítems, stats, etc.. para que cuándo vuestros compañeros prueben vuestra app os den un feedback de la jugabilidad, experiencia en base dificultad y equilibro del juego. Haced todas las modificaciones que creáis oportunas, incluso podéis cambiar mi sistema de calcular el daño.

Observaciones:

- La práctica para considerarse valida a corregir debe hacer uso de clases, constructores, métodos, objetos…
- Vosotros decidís cuantas clases vais a crear. Cómo ayuda algunas de las clases que podríais tener que utilizar son:
  - Heroe, Item, Monstruo, Stuff, Dado
- Tareas:
  - **Items**:
    - Definir cuáles serán los ítems para cada personaje.
    - Definir los valores de ataque, defensa.. de cada ítem.
    - Definir el precio de los ítems.
  - **Personajes**:
    - El personaje ha de empezar con un número X de monedas, esta cantidad inicial la decidís vosotros en función del valor que definàis para cada item.
    - El personaje debe iniciar con una cantidad de exp igual a 0.
    - Cada personaje tiene unos stats diferentes, por ello elegid cuál será la equipación y en base a los resultados, este personaje deberá tener más o menos vida. No se ha de poner los mejores puntos y vida a un personaje y el resto estar desfavorecidos, pensad que vuestro juego debe ser jugable y atraer al usuario.
    - Si el personaje muere durante un combate, el usuario ha de volver a empezar.
  - **Monstruos:**
    - Elegid los valores que va a tener cada monstruo ataque, oro que dan, experiencia etc..

Puntuación de la práctica:

En el **dungeon**, la selección de los monstruos se debe utilizar pickerView. Para el combate se debe tener en cuenta el rango de ataque y mostrar uno, dos o tres dados según este rango. El lanzamiento de los dados se realiza con pickerView y el action de un botón que simula la tirada tanto del monstruo como del usuario. El código para realizar la animación os lo facilito yo. **(5,5 puntos).**

En el **Inventory**, los valores deben ser calculados automáticamente en función de los ítems, mostrar experiencia, vida y dinero según las estadísticas del personaje **(2 puntos)**

En el **shop**, se debe utilizar pickerView para mostrar los objetos. Al seleccionar un elemento y hacer clic a comprar se debe restar del saldo y aplicarle al héroe su nuevo ítem. **(2 puntos)**

Vídeo de demostración de funcionalidad de la app:

https://www.youtube.com/watch?v=QbMLlDLVyXU

Ampliaciones posibles:

- Implementar un sonido a cada ataque cómo mínimo durante la pelea en el dungeon. Realizar la parte del dungeon es requisito indispensable para obtener esta puntuación. También podéis implementar más sonidos en diferentes puntos de la partida. **(0.5 puntos).**
- Implementar un sistema de ranking de puntuaciones con el nombre del jugador y la puntuación de experiencia que han ido consiguiendo los diferentes héroes(jugadores) que se han ido enfrentando a vuestra app.
- Implementar un sistema que guarde la información del héroe al cerrar la partida.
- Sistema de mensajería mediante Alerts al morir, comprar….
- Uso de gifs para animar al héroe, monstruo…
- Uso de pociones que se puedan comprar en la tienda y aumenten los valores del héroe, le den una oportunidad de volver a la vida, etc..
    - Un contador de cuenta atrás, para calcular cuándo terminan los efectos de las pociones.

Notas:

Qué incluye el enunciado:

-Conjunto de assets.
-Sonido de ataque.
-Código de animación de los dados.
-Ejemplo de creación de una vista personalizada para el pickerView.

Disponéis de todos los assets de personajes e ítems. Diseñar nuevos assets para vuestros personajes o ítems no se contabilizará si la práctica no funciona a la perfección. Además, el diseño de nuevos assets no debe perjudicar el desarrollo de la parte de programación ni es justificante de la falta de tiempo, ni aportará más nota. Por lo tanto, no perdáis tiempo para desarrollar.