Simulador de batalla RPG

# Análisis del Sistema (30 puntos)

## 1.1 Requisitos funcionales del sistema (5 pts)

Escribir los puntos exactos que debe cumplir tu sistema.

Ejemplo: “El usuario puede crear una nueva solicitud de boletos”, etc.

* El usuario debe ser capaz de elegir la cantidad de unidades aliadas que quiere usar en el juego. Estas son “Hunter”, “Rogue”, “Warrior”, “Support”. Cada clase tiene ataques y objetos propios, además de diferentes stats.
* El programa genera una cantidad de enemigos aleatroai, siendo su tipo elegido aleatoriamente. Los tipos de enemigos son los siguientes: “Slime”, “Golem”, “Dragon”.
* En cada ronda del juego, el usuario puede ver la intención de los enemigos, es decir, sobre quien actuarán y de qué forma actuarán.
* El jugador entonces decide con cada personaje que acción quiere realizar: atacar, defender, o usar objeto.
* Una vez elegidos los comandos, se resuelva la cadena de ataque. Al finalizar esta, se revisa si aun quedan enemigos o aliados en pie. Si se da uno de estos dos casos, se muestra el mensaje de victoria o derrota según corresponda.

## 1.2 Clases necesarias y su propósito (5 pts)

|  |  |
| --- | --- |
| Clase | Propósito |
| Combatiente | Es la clase padre de todas las entidades en el juego, siendo abstracta. Cada combatiente tiene vida, ataque, defensa, evasión y uno o varios status. Además, cada una puede realizar una acción sobre otra entidad o sobre sí misma. |
| Acción | Esta clase representa la intención de una entidad hacia otra. Cuenta con un emisor, un tipo de acción (que puede ser ataque, bloqueo, curación o buff) y un recipiente, además de cierta velocidad. |
| Aliado | Esta clase hereda de Combatiente, y representa las unidades aliadas del jugador. Cada aliado cuenta con lo siguiente: un inventario, el cual tiene varios Items; una cantidad de mana, la cual se usar para hacer técnicas, y ciertas técnicas, las cuales son habilidades más fuertes que un ataque (depende de la clase, cuestan mana) |
| Técnica | Esta clase hereda de Acción, y representa una técnica usada por un aliado. Tiene cierto costo de mana, y es más potente que los ataques. |
| NPC aliado | Esta clase hereda de Aliado, y representa a una “mascota” invocada en el combate. Pertenece a los aliados, aunque no contiene técnicas ni inventario y solo puede atacar. |
| NPC enemigo | Esta clase hereda de Enemigo, y representa a una “mascota” invocada en el combate. Pertenece a los aliados, aunque no contiene habilidades enemigas y solo puede atacar. |
| Item | Esta clase representa los objetos que los aliados pueden usar. Cada item tiene un propósito diferente (curar, buff de cierto atributo, infligir daño) y una cantidad de existencias. |
| Inventario | Esta clase representa el inventario que un aliado tiene. Contiene varios Items, y cuenta con los siguientes métodos: revisar ítems y usar item |
| Enemigo | Esta clase hereda de Combatiente, y representa a los enemigos del jugador. Cada uno de ellos cuenta con una habilidad especial, la cual se usa bajo cierta condición y tiene determinado efecto. |
| Rogue | Esta clase hereda de aliado, y cuenta con sus propios ataques e inventario. Los Rogues se caracterizan por tener una alta evasión y ataque medio, aunque cuentan con baja defensa. Sus ataques son de alta velocidad, y cuentan también con ataques en área. |
| Hunter | Esta clase hereda de aliado, y cuenta con sus propios ataques e inventario. Los Hunters se caracterizan por invocar mascotas y tener ataques en área, aunque su defensa es media y su ataque principal no es muy fuerte. |
| Mage | Esta clase hereda de aliado, y cuenta con sus propios ataques e inventario. Los Mage se caracterizan por tener habilidades poderosas, las cuales pueden mejorar a aliados o protegerlos temporalmente. Sin embargo, sus ataques son bastante débiles, al igual que su defensa. |
| Warrior | Esta clase hereda de aliado, y cuenta con sus propios ataques e inventario. Los Warrior se caracterizan por tener una alta defensa y ataque, además de poder proteger a sus compañeros. Sin embargo, su velocidad es bastante lenta, y le cuesta atacar a grupos de enemigos pequeños. |
| Support | Esta clase hereda de aliado, y cuenta con sus propios ataques e inventario. Los Support cuentan con un inventario más grande que el resto de aliados, además de poder curar a sus compañeros e invocar NPC. Sin embargo, su ataque es débil, al igual que su defensa. |
| Slime | Esta clase hereda de enemigo. Se caracteriza por tener un medio y una velocidad media. Además de ello, es capaz de curarse en vez de atacar. Su habilidad especial le permite partirse en 2 slimes más pequeños al tener el 50%, siendo que las copias no tienen esta habilidad. |
| Slime Boss | Esta clase hereda de slime. Se caracteriza por la capacidad de cada slime hijo de dividirse otra vez, además de en su ataque tiene cierta probabilidad de disolver Items del inventario de a quien ataque. |
| Witch | Esta clase hereda de enemigo. Se caracteriza por tener un ataque débil, aunque puede envenenar a los aliados o invocar esqueletos. Su habilidad especial es pasiva, la cual provoca que su evasión aumente cada vez que es atacada con una técnica. Sin embargo, esta evasión se reinicia cuando un aliado le hace daño con un ataque. |
| Witch Boss | Esta clase hereda de Witch. Se caracteriza por la capacidad de invocar esqueletos más fuertes. Además, al tener un 50% de vida, drena el alma de los aliados, curándose en el proceso (una sola vez). |
| Goblin | Esta clase hereda de enemigo. Se caracteriza por tener un ataque mediocre, aunque puede robar a los aliados de sus objetos. Su habilidad especial se activa al momento de tener 50% de vida, con lo cual en cada turno de ahora en adelante tiene cierta probabilidad de escapar de la batalla. |
| Goblin Boss | Esta clase hereda de Goblin. Se caracteriza por poder atacar múltiples veces. Además de ello, como habilidad especial, puede usar 3 de los objetos que ha robado en los enemigos o aliados. |
| Golem | Esta clase hereda de enemigo. Se caracteriza por tener una alta vida y defensa, aunque sus ataques son lentos. Sin embargo, sus ataques pueden infligir daño en área. Su habilidad especial es pasiva, siendo que tiene una defensa extra contra ataques físicos. |
| Golem Boss | Esta clase hereda de Golem. Se caracteriza por tener una resistencia física aún mayor, además de poder absorber todos los NPC del juego para recuperar vida, cuantas veces quiera. |
| Habilidad enemiga | Esta clase representa las habilidades que un enemigo puede tener. Algunas se activan una vez, mientras que otras se activan varias veces o son pasivas. Los Bosses tienen 2 habilidades pasivas, siendo la primera una versión amplificada de su contraparte y una especial de jefe. |
| Status | Esta clase representa efectos que suceden en un combatiente durante determinada cantidad de rondas. Pueden activarse al final de cada ronda, o cuando su duración termina. Es posible eliminar los Status por diferentes métodos. |
| Controlador de crear combatientes | Esta clase se encarga de crear los combatientes del juego al momento de iniciar la partida. |
| Controlador de elegir acción aliada | Esta clase se encarga de elegir las acciones de los aliados en cada ronda. |
| Controlador de acciones enemigas. | Esta clase se encarga de elegir las acciones de los enemigos en cada ronda. |
| Controlador de combate | Esta clase se encarga de resolver la línea de comandos de cada ronda, después de que los aliados hayan elegido sus acciones. |
| Controlador de fin de ronda | Esta clase se encarga de revisar a todos los combatientes al final de la ronda, para ver si se activa alguna habilidad especial de los enemigos o para restar vida por envenenamiento. |
| Consola | Esta clase se encarga de enviar los mensajes necesarios al usuario. |
| Main | Esta clase se encarga de dirigir el flujo del combate conectándose con los 5 controladores y la Consola. |

## 1.3 Atributos de cada clase (10 pts)

Clase: Combatiente

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Nombre | String | Protegida | Es el nombre del combatiente. |
| Vida | Int | Protegida | Es la cantidad de vida de un combatiente. Si se acaba, el combatiente no puede continuar luchando y se elimina del juego. |
| Ataque | Int | Protegida | Es la cantidad de ataque de un combatiente. |
| Defensa | int | Protegida | Es la cantidad de defensa de un combatiente. |
| Evasión | Int | Protegida | Es la probabilidad de evasión de un combatiente. Nunca puede ser mayor de 80 ni menor de 0. |
| Lista de status | ArrayList <Status> | Protegida | Es una lista de todos los Status que un combatiente tiene. Al final de la ronda, se revisa esta lista de status. |

Clase: Acción

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Nombre | String | Protegida | Es el nombre de la acción. |
| Velocidad | Int | Protegida | Es la velocidad con la que se ejecuta la acción. |
| Intención | ArrayList <Int> | Protegida | Es el propósito de la acción. La intención depende del siguiente código:   * 1 = infligir daño * 2 = infligir daño porcentual * 3 = curar * 4 = buffear ataque * 5 = buffear defensa * 6 = buffear evasion * 7 = añadir veneno * 8 = quitar veneno * 9 = invocar NPC |
| Tipo | Int | Protegida | Es el tipo correspondiente a la acción. El tipo de acción depende del siguiente código:   * 0 = atacar * 1 = protegerse * 2 = usar técnica * 3 = usar objeto * 4 = usar habilidad especial * 5 = invocar npc * 6 = recibir efecto de Status |
| Emisor | Combatiente | Protegida | Es quien emite la acción. En caso de que el emisor se encuentre muerto al momento de resolverse la acción, esta se cancela. |
| Recipiente | ArrayList <Combatiente> | Protegida | Es quien o quienes reciben la acción. Si todos los recipientes se encuentran muertos al momento de resolverse la acción, esta se cancela. |

Clase: Aliado

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Inventario | Inventario | Protegida | Es el inventario del aliado. Dentro de este inventario, se cuenta con varios Items, los cuales tienen gran variedad de efectos. |
| Lista de técnicas | ArrayList <Técnica> | Protegida | Es la lista de técnicas que el aliado puede usar. Las técnicas son propias de cada clase. |
| Max maná | Int | Protegida | Es la cantidad máxima de maná que el aliado puede tener. |
| Maná | Int | Protegida | Es la cantidad de maná que el aliado tiene. Dicho maná es necesario para usar habilidades. |

Clase: Técnica

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Costo de mana | Inventario | Privada | Es la cantidad de maná que se necesita para usar la técnica. |

Clase: NPC aliado

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Maestro | Combatiente | Privada | Es el combatiente que invocó a este NPC. Si el maestro muere, el NPC también muere. |

Clase: NPC enemigo

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Maestro | Combatiente | Privada | Es el combatiente que invocó a este NPC. Si el maestro muere, el NPC también muere. |

Clase: Item

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Cantidad de usos | int | Privada | Es la cantidad de veces que el Item puede usarse. |
| Posible objetivo | Int | Privada | Representa con qué personajes puede usarse el objeto. Se basa en el siguiente código:  1 = 1 aliado  2 = todos los aliados  3 = 1 enemigo  4 = todos los enemigos  5 = todos los combatientes |
| Cantidad de efecto | int | Privada | Representa la potencia que el objeto tiene. Las acciones dependen de su emisor y su recipiente para saber su efecto, pero con los efectos se aplica la cantidad de efecto sin tomar esto en cuenta. |
| Intención | int | Privada | Representa el efecto que usar el objeto tiene. Se basa en el mismo código de las acciones. |

Clase: Inventario

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de Items | ArrayList <Item> | Privada | Son los diferentes objetos que el Inventario contiene. |

Clase: Enemigo

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de habilidades enemigas | ArrayList <Habilidad enemiga> | Protegida | Es la lista de habilidades enemigas que el enemigo puede usar. Generalmente, consiste de 1 o 2 habilidades. |

Clase: Rogue

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguna |  |  |  |

Clase: Hunter

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Máximo de mascotas | Int | Privada | Es el limite de mascotas que un Hunter puede tener al mismo tiempo. |

Clase: Mage

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguna |  |  |  |

Clase: Warrior

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguna |  |  |  |

Clase: Support

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguna |  |  |  |

Clase: Slime

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguna |  |  |  |

Clase: Slime Boss

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguna |  |  |  |

Clase: Witch

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Máximo de mascotas | int | Protegida | Es el máximo de NPC que una Witch puede invocar al mismo tiempo. |

Clase: Witch Boss

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |

Clase: Goblin

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Objetos robados | ArrayList <Item> | Protegida | Es la lista de Items que un goblin le ha robado a los jugadores. |
| Aliados a los que se ha robado | ArrayList <Aliado> | Protegida | Es la lista de Aliados a los cuales el Goblin le ha robado. Es posible repetir los aliados a los que se le ha robado. |

Clase: Goblin Boss

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |

Clase: Golem

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Resistencia física | Int | Protegida | Es la cantidad de resistencia adicional que un golem tiene para ataques físicos. |

Clase: Golem Boss

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |

Clase: Habilidad enemiga

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Cantidad de activaciones | Int | Privada | Es la cantidad de veces que una habilidad enemiga puede activarse. Puede ser un número fijo, o ser “null”, en cuyo caso la habilidad puede activarse cuantas veces se quiera. |
| Forma de activación | Int | Privada | Es la forma en que la habilidad se activa. Se basa en el siguiente código:   * 1 = al recibir daño * 2 = al tener 50% de vida * 3 = al tener 0% de vida |

Clase: Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Tipo de Status | int | Privada | Es el tipo de status que un Combatiente sufre. En base a él se decide el nombre, velocidad e intención de la Acción generada por el Status. Se basa en el siguiente código:  1 = envenenamiento  2 = vitalidad (curación periódica) |
| Duración | int | Privada | Es la cantidad de rondas que este status dura. Al final de cada ronda, la duración decrece en 1. Si llega a 0, se elimina el status. |
| Tipo de activación | int | privada | Es la forma en que se activa el status. Se basa en el siguiente código:   * 1 = Cada ronda * 2 = Al terminarse la activación |

Clase: Controlador de crear combatientes

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de combatientes | ArrayList <Combatientes> | Privada | Es la lista de combatientes que se usarán durante la partida |

Clase: Controlador de elegir acción aliada

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de combatientes | ArrayList <Combatientes> | Privada | Es la lista de combatientes que se usan en la partida. De ellos, solo los aliados eligen sus acciones. |
| Lista de acciones | ArrayList <Accion> | Privada | Es la lista de acciones que se resolverán en el combate. |

Clase: Controlador de acciones enemigas

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de combatientes | ArrayList <Combatientes> | Privada | Es la lista de combatientes que se usarán durante la partida. De ellos, solo los enemigos eligen sus acciones de forma aleatoria. |
| Lista de acciones | ArrayList <Accion> | Privada | Es la lista de acciones que se resolverán en el combate. |

Clase: Controlador de combate

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de combatientes | ArrayList <Combatientes> | Privada | Es la lista de combatientes que se usarán durante la partida |
| Lista de acciones | ArrayList <Accion> | Privada | Es la lista de acciones que se resolverán en el combate. |

Clase: Controlador de fin de ronda

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de combatientes | ArrayList <Combatientes> | Privada | Es la lista de combatientes que se usarán durante la partida |

Clase: Consola

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Teclado | Scanner | Privada | Es la forma en que el programa se comunica con la consola del sistema. |

Clase: Main

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Visibilidad** | **Propósito** |
| Lista de combatientes | ArrayList <Combatiente> | Privada | Es la lista de combatientes que se usarán en el juego. |
| Lista de acciones | ArrayList  <Accion> | Privada | Es la lista de acciones que deben resolverse en 1 ronda. |

### 1.4 Métodos de cada clase (10 pts)

Clase: Combatiente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Recibir daño | * Daño: int | Boolean | Publica | Es la forma en que un combatiente pierde vida. Si el combatiente tiene más de 0 de vida, regresa false. De lo contrario, regresa True. |
| Recibir curación | * Curación: int | void | Publica | Es la forma en que un combatiente puede recuperar vida. En caso de que la curación exceda el máximo de vida de un combatiente, se pierde el exceso de vida. |
| Recibir buff | * Atributo: ArrayList <int> * Cantidad:   int | void | Publica | Representa la forma en que se mejoran los stats de un combatiente. El atributo es el stat que se mejora, mientras que la cantidad es en cuanto se mejora. El atributo se basa en el siguiente código:   * 1 = ataque * 2 = defensa * 3 = evasión |
| Agregar Status | * Status nuevo: Status | boolean | Publica | Este método se usa para agregar un nuevo Status a la lista de Status del Combatiente. Un tipo de Status no se puede repetir en un Combatiente, por lo que si este caso se da se regresa False. De lo contrario, se regresa True. |
| Remover Status | * Tipo de Status: int | Boolean | Publica | Este método se usa para remover un Status del Combatiente. Para ello, busca un Status que coincida con el tipo ingresado, y lo elimina. Si no lo encuentra regresa False. De lo contrario, regresa True. |

Clase: Acción

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Aliado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Usar objeto | * Nombre de objeto: String * Objetivos: ArrayList <Combatientes> | Acción | Publica | Es la forma en que un Aliado puede usar un objeto sobre otro combatiente. |
| Usar técnica | * Nombre de la técnica: String * Objetivos: ArrayList <Combatientes> | Acción | Publica | Representa la forma en que un Aliado usa una técnica sobre otro Combatiente. |
| Buscar técnica | * Nombre de la técnica: String | Técnica | Protegida | Es un método auxiliar, que permite encontrar un objeto de Técnica en base a su nombre. |
| Verificar gasto de maná | * Nombre de la técnica: String | boolean | Publica | Es un método auxiliar, cuyo objetivo es verificar si el Aliado tiene suficiente maná para usar una técnica. |
| Gastar maná | * Cantidad de mana: int | void | Protegida | Cuando un Aliado usa una técnica, este proceso realiza el gasto de mana correspondiente. |

Clase: NPC aliado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: NPC enemigo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Item

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| usar | * Ninguno | void | Publica | Cuando el Item se usa para generar una acción, se disminuye en 1 la cantidad de usos que tiene. |

Clase: Inventario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Mostrar lista de Items | * Ninguno | String | Publica | Devuelve un mensaje que contiene todos los objetos que hay en el inventario. |
| Buscar Item | * Nombre de Item: String | Item | Publica | Busca si el Item ingresado existe en el Inventario. En caso de existir, devuelve dicho Item. |
| Anadir Item | * Item nuevo: Item | void | Publica | Agrega un Item a el Inventario |
| Usar Item | * Nombre de Item: String | Item | Publica | Usa el Item que se encuentra en el inventario, y lo devuelve. |
| Perder item aleatorio | * Ninguno | Item | Publica | Quita un Item del inventario, sin usar su efecto |

Clase: Enemigo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Usar habilidad | * Ninguno | Accion | Publica | Este método se usa al final de la ronda. Sirve para verificar si la habilidad de un enemigo debe activarse, dado que cierta condición se dé. |

Clase: Rogue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Hunter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Mage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Warrior

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Slime

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Dividirse | * Ninguno | ArrayList <Slime> | publica | Este método regresa dos copias del slime, siendo que cada una tiene la mitad de vida del slime original. Sin embargo, estas copias no pueden usar este método. |

Clase: Slime Boss

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Witch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Witch Boss

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

Clase: Goblin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Robar objeto | * Nombre de aliado: String * Objeto robado: Item | void | Publica | Al robarle un objeto al Aliado, este se guarda en la lista de objetos robados del Goblin |

Clase: Goblin Boss

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Usar objetos | * ninguno | void | Publica | El Goblin Boss usa hasta tres de los objetos que ha robado. Los objetivos son completamente aleatorios, por lo que puede sanar a Aliados o a Enemigos por igual. |

Clase: Golem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Recibir daño | * Ataque recibido: Acción | Boolean | Publica | Cuando el Golem va a recibir daño, se verifica si el ataque fue físico. En caso de serlo, se aplica la reducción de la habilidad especial antes de recibirse. |

Clase: Golem Boss

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Remover NPC | * ninguno | Acción | Pública | Genera una acción que consiste en un ataque porcentual del 100% en todos los NPC. |
| Curacion por NPC | * Cantidad de NPC: int | void | Pública | El Golem Boss se recupera en cierta cantidad de vida, según cuantos NPC haya absorbido. |

Clase: Habilidad enemiga

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Generar Acción | * Recipientes: ArrayList <Combatiente> | Acción | Publica | Genera un objeto de Acción, el cual tiene como emisor al enemigo que tenga la habilidad enemiga. |

Clase: Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Generar Acción | * ninguno | Acción | Publica | Genera un objeto de Acción, el cual no tiene un emisor. Su recipiente es el Combatiente al que está unido. |

Clase: Controlador de crear combatientes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Generar combatiente | * Tipo de combatiente: int | Combatiente | Privada | Este método genera un nuevo Combatiente del tipo ingresado, en base al siguiente código:   * 1 = Mage * 2 = Hunter * 3 = Rogue * 4 = Warrior * 5 = Support * 6 = Slime * 7 = Golem * 8 = Witch * 9 = Goblin * 10 = Slime Boss * 11 = Golem Boss * 12 = Witch Boss * 13 = Goblin Boss * 14 = NPC aliado * 15 = NPC enemigo |
| Generar nombre | * ninguno | String | Privada | Este método auxiliar se usa para generar un nombre aleatorio para el Combatiente |
| Generar combatientes | * ninguno | ArrayList <Combatiente> | Publica | Este método genera los combatientes que se usarán en el juego |

Clase: Controlador de elegir acción aliada

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Elegir acción | * emisor: Combatiente | Acción | Publica | Genera un objeto de Acción, el cual tiene como emisor a un Combatiente. |
| Elegir objetivo | * posibles Objetivos: ArrayList <Combatiente> | Combatiente | Privada | Es un método auxiliar, cuyo propósito es elegir un Combatiente de la lista de posibles objetivos. |

Clase: Controlador de acciones enemigas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Elegir acción | * emisor: Combatiente | Acción | Publica | Genera un objeto de Acción, el cual tiene como emisor a un Combatiente. |
| Elegir objetivo | * posibles Objetivos: ArrayList <Combatiente> | Combatiente | Privada | Es un método auxiliar, cuyo propósito es elegir un Combatiente aleatorio de la lista de posibles objetivos. |

Clase: Controlador de combate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Resolver línea de acciones | * Lista de acciones: ArrayList <Accion> | void | Publica | Este método resuelve todas las acciones en la lista de acciones, en cierto orden. |
| Resolver acción | * accion: Accion | void | Privada | Este método resuelve una acción. |

Clase: Controlador de fin de ronda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Generar acciones | * Lista de Combatientes: ArrayList <Combatiente> | ArrayList <Acción> | Publica | Se revisa a todos los Combatientes, para ver si pueden generar una acción al final de la ronda. |
| Generar acción | * Combatiente: Combatiente | ArrayList <Acción> | Privada | Se revisa si el Combatiente puede generar una o más acciones al finalizar la ronda. |

Clase: Consola

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Mensaje de bienvenida | * ninguno | void | Publica | Muestra al usuario el mensaje de bienvenida del programa. |
| Generar Combatientes | * ninguno | void | Publica | Le pide al usuario los datos necesarios para generar los combatientes del combate. |
| Turno de aliado | * aliado: Aliado | void | Publica | Muestra al usuario que puede hacer cierto aliado, y le pide que elija que hará el aliado. |
| Verificar acciones de combate | * lista de Acciones: ArrayList <Accion> | void | Publica | Muestra al usuario que acciones están preparadas para hacerse en el combate. |
| Verificar Combatientes | * Lista de Combatientes: ArrayList <Combatiente> | void | Publica | Muestra al usuario todas las unidades que se encuentran en combate. |
| Verificar Combatiente | * Combatiente: Combatiente | void | Publica | Muestra al usuario el estado actual de un Combatiente. |
| Resultado de acción | * Acción: Acción | void | Publica | Muestra al usuario el resultado de una acción |

Clase: Main

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **Parámetros (tipo de dato)** | **Resultado (tipo de dato)** | **Visibilidad** | **Propósito** |
| Ninguno |  |  |  |  |

### 2. Diseño: Diagrama de Clases (30 puntos)

* Asegúrate de mostrar atributos y métodos con visibilidad (+, -).
* Indica relaciones entre clases (asociación, agregación, etc.).
* Incluye el driver program (Main).

<https://lucid.app/lucidchart/66e30f2f-93f7-49db-9056-0bff8b4bd3ad/edit?viewport_loc=-880%2C-1471%2C4897%2C2227%2C0_0&invitationId=inv_5e1669fe-f390-49f9-bec3-eabca47e1422>

### 3. Programa (40 puntos)

En cada archivo `.java`, asegurarse de incluir:

* Las clases necesarias.
* Uso adecuado de objetos.

Menú que debe implementar el driver program:

* 1. Nuevo comprador
  2. Nueva solicitud de boletos
  3. …
  4. …
  5. Salir

GitHub: colocar aquí la URL:

<https://github.com/Oscarro11/Simulador-RPG>

# Checklist antes de entregar

Está claro el análisis?

El diagrama tiene los elementos UML correctamente?

Subiste tu código a GitHub con todo lo necesario?