

Disseny i programació orientats a objectes

Projecte del primer semestre – Fases 1 i 2

Simple LSRPG

**Departament d'Enginyeria
La Salle – Universitat Ramon Llull
2 novembre de 2022**

Índex

1	Introducció	1
2	Especificació de requeriments	2
2.1	Persistència i inici del programa	2
2.2	Creació de personatge	2
2.3	Llistat de personatges	4
2.4	Crear una aventura.....	5
2.5	Jugar una aventura.....	6
2.5.1	Fase de preparació	6
2.5.2	Fase de batalla.....	7
2.5.3	Fase de descans curt	8
2.6	Sortir de l'execució	8
3	Accions	9
3.1	Atac.....	9
3.2	Curació.....	9
3.3	Suport	9
4	Classes de personatge	10
4.1	Aventurer	10
4.1.1	Punts de vida màxims.....	10
4.1.2	Valor d'iniciativa.....	10
4.1.3	Accions durant la preparació.....	10
4.1.4	Accions durant la batalla	10
4.1.5	Accions durant el descans curt.....	10
5	Persistència en JSON	11
5.1	Fitxer de monstres	12
5.2	Fitxer de personatges.....	13
5.3	Fitxer d'aventures	13
6	Exemples d'execució	14
6.1	Validació de la font de persistència	14
6.2	Menú principal	14
6.2.1	Funcionalitat de creació de personatge.....	15
6.2.2	Funcionalitat de llistat de personatges	16
6.2.3	Funcionalitat de creació d'una aventura.....	17
6.2.4	Funcionalitat de jugar una aventura	19

1 Introducció

Durant els últims anys, l'afició als jocs de rol de taula (TTRPG en anglès) ha augmentat de forma molt significativa. Plataformes com Twitch o YouTube tenen creadors especialitzats tant en l'*streaming* de partides de joc de rol, com en la formació de nous jugadors o debat de les normes.

Veient la situació actual, i sabent que la universitat té un club de rol (Play & Roll), des de l'assignatura hem pensat interessant poder desenvolupar el nostre propi sistema, que anomenarem *Simple LSRPG* (malauradament no tenim els drets per fer servir el nom *Simple RPG*).

Com sabem que tot sistema de joc nou s'haurà d'ajustar i balancejar per intentar evitar que alguns personatges estiguin OP (*overpowered*), volem plantejar una versió de prova o *alpha*, tot fent ús d'una temàtica retro, i implementat en una plataforma virtual.

Per aquest motiu, implementarem l'aplicació de *Simple LSRPG* amb el llenguatge Java, i donat que els requeriments són força extensos hem decidit dividir-ne el desenvolupament en dos *releases*:

- **Release inicial:** Definició i execució d'un MVP (*Minimum Viable Product*) amb les mecàniques del joc bàsiques. Aquesta *release* inicial es dissenyarà i implementarà com a part de les fases 1 i 2.
- **Release final:** Ampliació de les mecàniques de joc bàsiques per incorporar noves classes de personatge i tipologia de monstres. Aquesta *release* final es dissenyarà i implementarà com a part de les fases 3 i 4.

En els següents apartats d'aquest document es detallaran els requeriments de la primera *release* de l'aplicació i es donaran exemples de la seva execució.

Abans de començar, però, és interessant mencionar que per jugar a *Simple LSRPG* caldrà disposar d'un conjunt de daus. En el nostre programa virtualitzarem aquestes tirades de daus mitjançant l'ús de valors aleatoris. Sempre que es parli d'un llançament de daus, s'indicarà el nombre de cares que té el dau i s'utilitzarà una notació especial dX on X equival al nombre de cares que té el dau. Tot dau d' X cares podrà obtenir resultats de l'1 a X . D'aquesta forma, el llançament d'un dau de 6 cares [1..6] s'indicarà com a un $d6$, mentre que el llançament d'un dau de 8 cares [1..8] s'indicarà com a un $d8$.

2 Especificació de requeriments

Aquest apartat recull les funcionalitats que s'esperen de l'aplicació *Simple LSRPG*. Concretament, es descriu el que cal dissenyar i implementar per la *release* inicial, corresponent a les fases 1 i 2 de la pràctica.

2.1 Persistència i inici del programa

En aquesta primera *release* la persistència de les dades es farà en fitxers JSON, format que es troba explicat a l'apartat 5. Concretament, el programa haurà de disposar de 3 fitxers JSON on s'emmagatzemaran els monstres, els personatges creats i les aventures. Aquests fitxers sempre s'anomenaran amb el mateix nom: *monsters.json*, *characters.json* i *adventures.json*.

Si bé el sistema pot anar creant la resta de fitxers durant l'ús del programa, en el moment d'iniciar-lo caldrà comprovar que existeixi una font de persistència vàlida on buscar els monstres del joc. En cas que no existeixi cap font de persistència vàlida, caldrà mostrar un missatge d'error i finalitzar l'execució. El format exacte del JSON on es persisteixen els monstres es troba definit a l'apartat 5.1.

En tot el programa, quan calgui recuperar informació sobre aventures, monstres o personatges, caldrà anar a buscar sempre aquesta informació a la nostra font de persistència. No es considerarà una bona pràctica tenir informació en memòria RAM que no s'estigui fent servir.

Un cop comprovat que podem llegir monstres amb el nostre programa, caldrà mostrar a l'usuari un menú que permeti accedir a la resta de funcionalitats.

2.2 Creació de personatge

Una de les funcionalitat bàsiques de qualsevol joc de rol és el procés de creació d'un personatge. En aquesta opció del menú principal es guiarà a l'usuari a través del procés de creació de personatges. Les fases del procés s'anomenen a continuació en l'ordre en que cal executar-les.

Nom del personatge

En primer lloc, caldrà demanar a l'usuari quin nom vol donar-li al seu personatge. Aquest nom podrà contenir qualsevol caràcter que pugui considerar-se una lletra, tant majúscula com minúscula. El nom podrà portar qualsevol tipus d'accent i podrà consistir de múltiples paraules. Per exemple, tant *Finwë* com *Finrod Felagund* són noms de personatge vàlids.

No s'acceptaran noms que continguin números o caràcters especials. El fet de jugar amb personatges que es diguin, per exemple, *TheLegend27* o *Vegetta777*, pot ser molt graciós, però trenca amb la immersió del joc, de forma que només acceptarem noms que compleixin les normes anteriors. En cas de trobar un nom que contingui caràcters no acceptats, caldrà mostrar un missatge d'error a l'usuari i tornar al menú principal.

També caldrà assegurar que els noms respectin el format universal d'aquests, és a dir, han de començar amb una majúscula al principi de cada nova paraula i la resta de lletres hauran d'estar en minúscula. En cas de trobar un nom que estigui incorrectament formatat, s'acceptarà, però el nostre programa haurà de modificar el seu format per ajustar-nos al format correcte. Per exemple, un personatge anomenat *sTaXx*, caldrà donar-li el format correcte perquè s'anomeni *Staxx*.

Per últim, caldrà assegurar que aquest nom sigui únic. Això vol dir que caldrà comprovar que no existeixi ja en el sistema un personatge amb el mateix nom exacte. En cas que el nom no es trobi disponible, caldrà mostrar un missatge d'error a l'usuari i tornar al menú principal.

Nom del jugador

És important pel sistema poder identificar qui és el propietari de cadascun dels personatges. Per aquest motiu, caldrà demanar a l'usuari que introdueixi el seu nom, o un àlies que l'identifiqui. No posarem limitacions al nom del jugador i acceptarem, sense jutjar, qualsevol nom o àlies que l'usuari decideixi introduir.

Nivell inicial del personatge

Tot i que normalment els personatges es creen a nivell 1, oferirem la possibilitat als jugadors de crear els seus personatges a un nivell superior. El nivell màxim que pot tenir un personatge serà 10, de forma que caldrà comprovar en tot moment que l'usuari introdueixi un valor vàlid [1 – 10]. En cas d'introduir un valor incorrecte, caldrà mostrar un missatge d'error i tornar a demanar el nivell del personatge a l'usuari.

Si un personatge és creat amb un nivell superior a 1, caldrà assignar-li tants punts d'experiència com siguin necessaris per arribar al nivell determinat. Més endavant, a l'apartat 2.5.3, s'explicarà en detall el funcionament dels punts d'experiència i els nivells d'un personatge.

Estadístiques del personatge

Tot personatge tindrà 3 estadístiques o característiques. Aquestes estadístiques serveixen per descriure numèricament les capacitats i habilitats del personatge. En concret, les 3 característiques seran Cos (*Body*), Ment (*Mind*) i Esperit (*Spirit*).

Aquests valors es generaran de forma automàtica sense interacció amb l'usuari. Per cada estadística a generar caldrà simular el llançament de 2 daus de 6 cares (2d6). Tal com es pot veure en la Taula 1, la suma dels resultats dels daus determinarà el valor assignat a aquest personatge en aquesta estadística.

Taula 1: Generació d'estadístiques del personatge.

SUMA DELS DAUS	VALOR ESTADÍSTICA
2	-1
ENTRE 3 I 5	+0
ENTRE 6 I 9	+1
ENTRE 10 I 11	+2
12	+3

Un cop el programa hagi generat les tres característiques d'un personatge, només ens quedarà assignar-li una classe de personatge.

Classe de personatge

La majoria dels jocs de rol permeten als seus jugadors triar la classe de personatge. La classe d'un personatge és allò que li permet a un personatge disposar d'un conjunt d'habilitats. Per exemple, en molts jocs un personatge pot triar si vol ser un guerrer i lluitar amb espasa i escut, o si vol ser un mag i ser capaç d'utilitzar les seves dotes arcanes.

En aquesta versió inicial del joc no volem complicar-nos massa i definirem per defecte que tots els personatges nous tenen la classe Aventura. En futures iteracions es definiran noves classes de personatges per tal de poder ampliar el ventall d'habilitats.

Durant aquesta fase del procés de creació d'un personatge no hi haurà interacció amb l'usuari. El programa automàticament assignarà al personatge la classe Aventura.

Un cop arribats en aquest punt, el personatge es pot considerar que ha estat creat correctament. Caldrà informar a l'usuari que el procés ha estat correcte i emmagatzemar el personatge a persistència.

2.3 Llistat de personatges

Aquesta funcionalitat del menú principal permetrà a l'usuari consultar els noms dels personatges creats que es trobin actualment emmagatzemats a persistència.

En primer lloc, caldrà demanar a l'usuari que introdueixi el nom d'un jugador. Si l'usuari prem la tecla *enter* i no introdueix cap nom de jugador, el programa entendrà que vol mostrar tots els personatges sense aplicar cap filtrat. Si, per contra, l'usuari introdueix el nom d'un jugador, caldrà mostrar tots els personatges que siguin propietat d'aquest jugador.

Per tal de cercar el nom del jugador s'aplicarà un criteri *case insensitive*, que significa que la comparació que es faci tractarà com a iguals una lletra majúscula o minúscula ('a' == 'A'). També mostrarem els personatges de jugadors que continguin parcialment en el seu nom el text cercat. Per exemple, si busquem el text *Gon*, també haurem de mostrar tots els personatges del jugador *AegonTheConqueror*, ja que conté el text *Gon*.

La llista amb els noms dels personatges caldrà que estigui numerada. Si la cerca no troba cap personatge que compleixi amb el criteri establert, caldrà mostrar un missatge a l'usuari indicant que no s'han trobat personatges amb aquest criteri, i tornar al menú inicial.

Informació completa del personatge

Un cop es mostri el llistat amb tots els noms dels personatges, l'usuari podrà introduir el número associat a un dels personatges de la llista. Si, per exemple, la cerca retorna 6 personatges, l'usuari podrà introduir un número entre l'1 i el 6. Quan l'usuari introdueixi un número vàlid, caldrà mostrar-li la informació completa del personatge. En cas contrari, caldrà mostrar un missatge d'error, i tornar al menú inicial.

Un cop l'usuari hagi escollit un número vàlid de la llista de noms de personatges obtinguda de la cerca anterior, caldrà mostrar tota la informació relacionada amb aquest personatge.

1. Nom del personatge
2. Nom del jugador propietari del personatge
3. Classe del personatge
4. Nivell del personatge
5. Punts d'experiència actuals del personatge
6. Estadístiques del personatge (Cos, Ment i Esperit)

Després de mostrar tota la informació del personatge, caldrà oferir l'opció a l'usuari d'esborrar-lo. Si l'escull, l'usuari haurà de confirmar l'eliminació del personatge introduint-ne el nom exacte, que es mostrarà per pantalla. En cas d'haver-hi qualsevol error en el nom, caldrà mostrar un missatge informatiu i tornar a sol·licitar la confirmació amb el nom. Si l'usuari prem *enter* sense introduir cap caràcter, es cancel·larà l'eliminació i el programa tornarà al menú principal.

2.4 Crear una aventura

Des del menú principal, l'usuari podrà crear una nova aventura. Una aventura consistirà en un conjunt de combats, entre 1 i 4, contra un nombre indeterminat de monstres (mínim 1).

Nom de l'aventura

La primera cosa que caldrà demanar a l'usuari serà el nom de l'aventura. Caldrà assegurar en tot moment que el nom de l'aventura sigui únic i no existeixi ja una aventura amb el mateix nom. En el suposat que ja existeixi una aventura amb aquest nom, caldrà mostrar un missatge d'error i tornar al menú principal.

Nombre de combats

A continuació, l'usuari haurà d'indicar el nombre de combats que vol que aquesta aventura tingui. Caldrà acceptar qualsevol valor numèric entre 1 i 4. Qualsevol altre valor serà considerat un error. En cas d'error, caldrà mostrar un missatge informatiu a l'usuari i tornar a demanar el nombre de combats. Si l'usuari segueix cometent errors, quan aquest cometi el tercer error es tornarà al menú principal, en lloc de seguir demanat aquest valor.

Creació d'un combat

Per cada un dels possibles combats que l'usuari hagi decidit crear, caldrà, en primer lloc, mostrar la informació actual del combat que s'està creant. La informació mínima que haurà de mostrar cada un dels combats haurà de ser la següent:

1. Número de combat actual (1 – 4)
2. Llistat numerat de monstres actualment en el combat (nom i quantitat)
3. Menú per poder afegir, eliminar monstres o continuar la creació de l'aventura

Un cop s'hagin creat tots els combats, es considerarà que l'aventura s'ha creat amb èxit i es procedirà a emmagatzemar-la a persistència.

Afegir monstres

Quan un usuari decideixi afegir monstres a un combat, es mostrarà una llista numerada amb tots els monstres disponibles al sistema. Per cada monstre caldrà indicar el nom del monstre i el seu nivell de dificultat.

Un cop llistats tots els monstres, caldrà preguntar a l'usuari quin d'ells vol afegir. L'usuari triarà un monstre a partir del número que aparegui a la llista. A continuació, se li preguntarà quants monstres d'aquest tipus vol afegir de cop.

Si el monstre a afegir ja existia a la llista de monstres, caldrà acumular la quantitat nova a la ja existent.

L'única limitació en l'afegit de monstres és que mai es podrà afegir més d'un monstre amb un nivell de dificultat *Boss*. En cas d'intentar afegir més d'un monstre de dificultat *Boss*, ja sigui per quantitat o varietat, caldrà mostrar un missatge d'error i sortir d'aquest menú sense afegir res, però seguint el procés de creació de l'aventura.

Un cop afegit el(s) monstre(s), caldrà actualitzar la informació del combat actual.

Eliminar monstres

De la mateixa forma que es poden afegir monstres a un combat, volem afegir l'opció d'eliminar monstres. En aquest cas, caldrà demanar el número pertinent, segons el llistat de monstres actualment en el combat.

Un cop triat el monstre, s'eliminaran tots els monstres d'aquest tipus del combat actual.

Un cop eliminat el(s) monstre(s), caldrà actualitzar la informació del combat actual.

2.5 Jugar una aventura

Per poder jugar una aventura el sistema necessita tenir un mínim de 3 personatges. Mentre el sistema no tingui suficients personatges emmagatzemats a persistència aquesta opció s'ha de mostrar desactivada amb un missatge indicant que no hi ha suficients personatges.

Quan un usuari decideixi jugar una aventura haurà de triar dues coses: l'aventura i la *party* (o grup de personatges que van junts a una aventura).

En primer lloc, caldrà mostrar-li un llistat amb totes les aventures disponibles i haurà de poder-ne triar una. Cal implementar tots els controls d'error necessaris per assegurar el correcte funcionament d'aquesta funcionalitat.

En segon lloc, caldrà fer el mateix amb tots els personatges. L'usuari haurà de triar entre 3 i 5 personatges dels disponibles. A partir d'aquest moment, aquest grup de personatges conformarà la nostra *party*. Cal implementar tots els controls d'error necessaris per tal d'assegurar que es poden triar correctament els personatges de la *party*, així com assegurar que un personatge no pot estar repetit (no, el conjur per clonar personatges encara no s'ha inventat en aquest joc).

Un cop escollida l'aventura i la *party*, caldrà inicialitzar els punts de vida màxims i actuals de cada personatge. Per fer-ho, se seguirà la fórmula indicada per la seva classe, descrita a l'apartat 4.

Després d'inicialitzar totes les dades es procedirà a iniciar l'aventura amb el primer combat. El funcionament intern de cada combat s'explica a continuació. Un cop acabat cadascun dels combats de l'aventura, caldrà procedir al següent combat de la llista o a la pantalla de victòria/derrota en funció del resultat del combat i de si és l'últim o no.

Cada combat està format per 3 fases: Preparació, Batalla i Descans curt.

Durant aquestes fases els personatges realitzaran les accions que els pertorqui segons la classe de personatge que tinguin associada. Els tipus d'accions disponibles i els seus efectes generals es descriuen a l'apartat 3. Cadascuna de les classes de personatge descriuran les accions a realitzar a cada fase, així com els seus efectes. Recordeu que actualment només es defineix la classe Aventurer, a l'apartat 4.1.

A continuació es descriuen en detall les fases d'un combat.

2.5.1 Fase de preparació

Durant la fase de preparació tots els personatges conscients realitzaran una acció segons el que determini la seva classe de personatge. Per exemple, els Aventurers faran una acció de suport anomenada *Self-motivated*.

Un cop realitzades totes les accions corresponents a la fase de preparació, es procedirà a determinar l'ordre d'iniciativa. L'ordre d'iniciativa indicarà l'ordre en que personatges i monstres realitzaran les accions que pertorquin durant la següent fase, és a dir la fase de batalla.

Per determinar l'ordre d'iniciativa cada personatge i cada monstre que participi en el combat haurà de tirar un dau de 12 cares (d12) i sumar el seu valor d'iniciativa. Tots els monstres del sistema tindran ja un valor d'iniciativa assignat. Per contra, cada personatge determinarà el seu valor d'iniciativa segons el que especifiqui la seva classe de personatge. Per exemple, el valor d'iniciativa d'un Aventurer és equivalent a la seva estadística d'Esperit ($d12 + \text{Esperit}$).

2.5.2 Fase de batalla

Tota batalla es jugarà per torns. Durant cada torn tots els personatges i monstres que no es trobin inconscients o morts actuaran seguint l'ordre d'iniciativa establert.

Els personatges realitzaran una acció segons el que determini la seva classe de personatge. Per exemple, els Aventurers faran una acció d'atac anomenada *Sword slash*.

Tots els monstres sempre faran una sola acció per torn. Concretament, faran una acció d'atac contra un únic personatge de la *party*. Els monstres escolliran el personatge a atacar de forma aleatòria, tot ignorant als personatges que es trobin inconscients. El mal que faran serà aleatori segons el tipus de dau que aparegui en la seva descripció. Com s'explica més en detall a l'apartat 3.1, hi ha tres tipus de mal: físic, màgic o psíquic. Tot monstre indicarà sempre en la seva descripció el tipus de mal que fan els seus atacs.

Inconsciència i mort

Tot monstre que vegi reduït el seu nombre de punts de vida actuals a una xifra inferior a 1, es considerarà que ha estat mort. Un monstre mort no podrà ser objectiu de cap acció, ni participarà en la batalla.

Tot personatge que vegi reduït el seu nombre de punts de vida actuals a una xifra inferior a 1, es considerarà que ha quedat inconscient. Un personatge inconscient no podrà ser objectiu de cap acció d'atac, no participarà en la batalla, i els seus punts de vida actuals seran 0 (un personatge mai pot tenir un valor de punts de vida actuals inferior a 0). Si un personatge inconscient és objectiu d'una acció de curació que incrementi els seus punts de vida actuals per sobre de 0, automàticament deixarà d'estar inconscient i tornarà a actuar amb normalitat.

Final de la batalla

La fase de batalla finalitzarà quan tots els monstres hagin estat morts o quan tots els personatges de la *party* hagin resultat inconscients. Aquest últim fet es coneix com a TPU (*Total Party Unconscious*). En cas de donar-se un TPU s'acabarà tant el combat com l'aventura de forma immediata. És a dir, es mostrarà un missatge de derrota i l'aplicació tornarà al menú principal.

En cas que els monstres estiguin morts, la fase de batalla es considerarà una victòria i es passarà a la següent fase del combat, el descans curt.

2.5.3 Fase de descans curt

Aquesta és la darrera fase del combat (excepte en el cas de TPU). En aquesta fase tots els personatges conscients realitzaran una acció segons el que determini la seva classe de personatge. Per exemple, els Aventurers faran una acció de curació anomenada *Bandage time*.

Seguidament, tots els personatges de la *party* guanyaran tants punts d'experiència com indiqui la suma dels punts d'experiència de tots els monstres morts. Aquest augment d'experiència pot ocasionar que un personatge pugui de nivell.

Pujar de nivell

Un personatge pujarà de nivell cada vegada que sumi un total de 100 punts d'experiència. Observeu la Taula 2 com a referència.

Quan un personatge pugui de nivell durant una aventura, automàticament es recalcularà la seva vida màxima i es modificarà la seva vida actual perquè sigui igual a la seva vida màxima (encara que estigui inconscient).

Taula 2: Relació entre punts d'experiència i nivell de personatge

PUNTS D'EXPERIÈNCIA	NIVELL DE PERSONATGE	PUNTS D'EXPERIÈNCIA	NIVELL DE PERSONATGE
0 – 99	1	500 – 599	6
100 – 199	2	600 – 699	7
200 – 299	3	700 – 799	8
300 – 399	4	800 – 899	9
400 – 499	5	900+	10

2.6 Sortir de l'execució

Des del menú principal, caldrà donar l'opció d'aturar l'execució sortint del programa. En aquesta situació caldrà assegurar que tota la informació del sistema que sigui important per a següents execucions estigui guardada al medi de persistència.

3 Accions

En aquest apartat s'explicaran en termes generals els diferents tipus d'accions que es poden realitzar en el joc. Tota acció tindrà un o múltiples objectius (*targets*) i un efecte a aplicar a tots els objectius escollits. A l'apartat 4 es descriuran els tipus d'accions que realitzen els Aventurers en les diferents fases del combat, així com les seves particularitats.

3.1 Atac

L'acció d'atac consistirà en intentar colpejar a tots els objectius de l'acció. Tot atac resultarà en una quantitat de mal que caldrà restar dels punts de vida actuals de cada un dels objectius. Aquest mal podrà ser d'un dels 3 tipus diferents definits en el joc: físic, màgic o psíquic. Aquest tipus de mal serà important en el futur, ja que en les properes versions del sistema podran existir classes o monstres resistents a certs tipus de mal.

Quan un personatge o un monstre realitzi una acció d'atac, primer caldrà llançar un dau de 10 cares (d10). Qualsevol resultat entre 2 i 10 es considerarà un impacte. Si per contra, la tirada (d10) resulta en un 1, es considerarà que l'atac ha fallat. Qualsevol atac fallat representarà sempre un mal de 0 punts. A més, un resultat de 10 a la tirada es considerarà un impacte crític. En cas d'impacte crític, caldrà calcular el mal amb normalitat i multiplicar-lo per 2.

Quan una classe impacti amb una acció d'atac (2 – 10 al d10), aquesta haurà de llençar un dau (d6 pels Aventurers) i sumar una de les seves estadístiques (Cos pels Aventurers) al resultat obtingut. El valor resultant de la suma serà la quantitat de punts de mal que rebran tots els seus objectius. El tipus de mal dependrà de la classe (físic pels Aventurers).

Com s'ha mencionat a l'apartat 2.5.2, quan un monstre impacti amb una acció d'atac (2 – 10 al d10), aquest haurà de llençar el tipus de dau que especifiqui la seva descripció. El valor resultat del dau serà la quantitat de punts de mal que rebrà el seu objectiu. El tipus de mal també estarà especificat a la descripció del monstre.

3.2 Curació

Una acció de curació buscarà recuperar la vida dels objectius de l'acció. Tota acció de curació restaurarà punts de vida actuals als seus objectius. Una classe o monstre no podrà tenir mai més punts de vida actuals que els seus punts de vida màxims. Si una acció de curació intenta superar aquest valor màxim, els punts de curació restants es perdran.

Quan una classe realitzi una acció de curació, aquesta haurà de llençar un dau (d8 pels Aventurers) i sumar la seva estadística de Ment al resultat obtingut. El valor resultant de la suma serà la quantitat de punts de curació que es restauraran a tots els seus objectius.

3.3 Suport

Una acció de suport s'encarregarà d'augmentar o disminuir una de les estadístiques dels seus objectius.

Quan una classe realitzi una acció de suport haurà d'indicar quin valor augmentarà (Cos, Ment, Esperit) i en quina quantitat (positiva o negativa).

Els efectes de tota acció de suport desapareixen al final del combat en que s'han utilitzat, és a dir després de la fase de descans curt corresponent.

4 Classes de personatge

En aquesta primera edició del programa només disposarem d'una única classe de personatge. A continuació es descriuen totes les característiques i habilitats associades a aquesta classe.

4.1 Aventurer

Un Aventurer és una persona amb les ganes i l'energia per decidir arriscar la seva vida per trobar riquesa, fama, o vivències personals fora de l'habitual. Sense un especial entrenament en l'art de l'esgrima i sent un inepte en l'art de la màgia, s'enfronta al perill amb més ganes que habilitat.

4.1.1 Punts de vida màxims

Els punts de vida màxim d'un Aventurer es calcularan a partir de la fórmula que es veu a continuació. Recordeu que al principi d'un aventura els personatges comencen al màxim de vida.

$$\text{Punts vida màxims} = (10 + \text{Cos}) * N,$$

on N equival al Nivell del Personatge

4.1.2 Valor d'iniciativa

La iniciativa d'un Aventurer es calcularà a partir de la seva estadística Esperit:

$$\text{Iniciativa} = d12 + \text{Esperit}$$

4.1.3 Accions durant la preparació

Abans de la batalla, els aventurers utilitzen una estona per auto-motivar-se. Segons han llegit en molts llibres, estar motivat en una batalla augmenta el rendiment de l'Aventurer, de forma que es prenen molt seriosament aquesta tasca.

Self-motivated

L'Aventurer realitzarà una acció de suport amb ell com a únic objectiu. Aquesta acció de suport augmenta l'estadística Esperit en +1 punts.

4.1.4 Accions durant la batalla

Durant la batalla, un Aventurer no té gaire coneixements tàctics, de forma que únicament es dedicarà a atacar a l'enemic que tingui una quantitat de punts de vida actuals més baixa.

Sword slash

L'Aventurer realitzarà un atac *Sword slash* contra un únic objectiu, calculant el mal de l'atac com es descriu en l'apartat 3.1 amb un d6 i utilitzant la seva estadística Cos com a valor d'atac. El mal de l'atac sempre serà de tipus físic.

$$\text{Mal} = d6 + \text{Cos}$$

4.1.5 Accions durant el descans curt

Després d'una dura batalla, l'Aventurer aprofita per descansar una estona i aplicar-se un bon grapat de vendes (o tiretes, si ho preferiu).

Bandage time

En aquest sentit, realitzarà una acció de curació on ell mateix serà l'únic objectiu de l'acció seguint el procés descrit a l'apartat 3.2.

$$\text{Curació} = d8 + \text{Ment}$$

5 Persistència en JSON

En diverses funcionalitats d'aquest projecte es demana persistir informació. En aquesta primera fase tota la informació que calgui persistir es desarà en fitxers JSON. Aquest format, les sigles del qual signifiquen *JavaScript Object Notation*, representa entitats, arrays i valors primitius de forma estructurada, sovint fent servir parells clau-valor.

En aquest format no hi ha gairebé cap restricció, i es poden barrejar tipus de dades fàcilment. Per exemple, es pot tenir un array on les posicions desin entitats totalment diferents.

A continuació disposeu d'un exemple en el que es persisteixen diferents instàncies de les classes que es poden apreciar al diagrama de la Figura 1. La Figura 2 exemplifica un possible format JSON per ambdues, tot i que també es podrien fer servir fitxers separats.

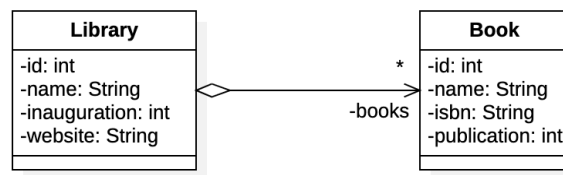


Figura 1. Diagrama UML de l'exemple de persistència.

```
[
  {
    "id": 1,
    "name": "CCUC - CSUC",
    "inauguration": 1996,
    "website": "https://ccuc.csuc.cat",
    "books": [
      {
        "id": 3,
        "name": "Learning Java",
        "isbn": "978-1565927186",
        "publication": 2000
      }
    ]
  },
  {
    "id": 2,
    "name": "Aladí - Xarxa de biblioteques municipals",
    "inauguration": 1986,
    "website": "https://aladi.diba.cat",
    "books": [
      {
        "id": 1,
        "name": "The Neverending Story",
        "isbn": "978-0140386332",
        "publication": 1993
      },
      {
        "id": 2,
        "name": "The Lord of the Rings",
        "isbn": "978-0007136582",
        "publication": 2002
      }
    ]
  }
]
```

Figura 2. Exemple de fitxer en format JSON per persistir la informació de biblioteques i llibres.

5.1 Fitxer de monstres

Tal com s'explica a l'apartat 2.1, abans de començar el programa caldrà assegurar que es disposa d'una font de persistència de dades. Aquesta font de persistència servirà per obtenir la informació dels monstres que existeixen en el programa. En aquesta primera *release* del programa disposarem d'una única font de dades, que consistirà en un fitxer JSON.

Aquest fitxer emmagatzemarà tots els monstres del programa en un array d'objectes, on cadascun dels elements de l'array serà un monstre disponible.

A continuació, en la Figura 3, es pot veure un exemple de la informació que pot contenir aquest fitxer. Tal com s'observa, tots els monstres disposen d'un conjunt de característiques comunes:

- **Nom:** Serà únic, de forma que mai trobarem a dos monstres amb el mateix nom.
- **Dificultat:** El *challenge* indicarà com de fort serà aquest monstre. Hi ha tres possibles valors: *Minion*, *Lieutenant* i *Boss*.
- **Experiència:** Quantitat de punts d'experiència que obté cada personatge de la *party* després de derrotar un monstre d'aquest tipus.
- **Punts de vida:** La quantitat de vida màxima a la que començarà el combat un monstre d'aquest tipus.
- **Valor d'iniciativa:** Valor a afegir a la tirada d'iniciativa d'aquest monstre. Tingueu present que pot ser negatiu.
- **Dau de mal:** Tipus de dau que es llençarà per determinar el mal que fan els atacs del monstre. Recordeu que la nomenclatura d6 vol dir que es llança un dau de 6 cares [1..6].
- **Tipus de mal:** La categoria a la que correspon el mal que fa el monstre. Hi ha tres possibles valors: *Physical*, *Magical* i *Psychical*.

```
[
  {
    "name": "Orc",
    "challenge": "Minion",
    "experience": 3,
    "hitPoints": 9,
    "initiative": -1,
    "damageDice": "d6",
    "damageType": "Physical"
  },
  {
    "name": "Morgoth",
    "challenge": "Boss",
    "experience": 168,
    "hitPoints": 100,
    "initiative": 4,
    "damageDice": "d20",
    "damageType": "Psychical"
  }
]
```

Figura 3. Exemple de fitxer JSON amb el llistat de monstres.

5.2 Fitxer de personatges

Per altra banda, en l'apartat 2.1 també es parla d'un fitxer de personatges en format JSON que caldrà fer servir per emmagatzemar la informació del personatges que es vagin creant en el sistema.

Aquest fitxer emmagatzemarà tots els personatges del programa en un array d'objectes, on cadascun dels elements de l'array serà un personatge disponible.

A continuació, en la Figura 4, es pot veure un exemple d'aquest fitxer.

- **Nom:** Serà únic, de forma que mai trobarem a dos personatges amb el mateix nom.
- **Nom del jugador:** Nom o àlies del jugador que ha creat el personatge.
- **Experiència:** Punts d'experiència acumulats pel personatge al llarg de les diferents aventures. Es representarà numèricament.
- **Estadístiques de Cos, Ment i Esperit:** Estadístiques en format numèric on cada un dels camps corresponents representa el valor de Cos, Ment i Esperit del personatge.
- **Classe de personatge:** Text amb el nom en anglès corresponent a la classe del personatge associat. Pels aventurers farem servir el text "Adventurer".

```
[
  {
    "name": "Amrod",
    "player": "SonsOfFeanor",
    "xp": 229,
    "body": 2,
    "mind": 0,
    "spirit": 0,
    "class": "Adventurer"
  },
  {
    "name": "Caranthir",
    "player": "SonsOfFeanor",
    "xp": 306,
    "body": 2,
    "mind": 0,
    "spirit": 1,
    "class": "Adventurer"
  }
]
```

Figura 4. Exemple de fitxer JSON amb el llistat de personatges.

5.3 Fitxer d'aventures

Finalment, en l'apartat 2.1 també es parla d'un tercer fitxer en format JSON que caldrà fer servir per emmagatzemar la informació de les aventures que es vagin creant en el sistema.

Aquest fitxer haurà d'emmagatzemar totes les aventures del programa en un array d'objectes, on cadascun dels elements de l'array serà una aventura disponible.

Teniu llibertat per decidir quin és el millor format per emmagatzemar les aventures en el fitxer.

6 Exemples d'execució

A continuació es mostren exemples d'execució de les funcionalitats prèviament descrites, amb les dades introduïdes per l'usuari en negreta. En general, el format i contingut dels missatges és lliure, sempre que siguin suficientment detallats per ser autoexplicatius.

Important: Cal comprovar l'entrada de dades de l'usuari en tots els moments prèviament descrits, cancel·lant la funcionalitat quan hi hagi un error (i informant-ne per pantalla). També és necessari validar l'entrada de l'usuari quan navega pels diferents menús de l'aplicació.

6.1 Validació de la font de persistència

La Figura 5 il·lustra la fase inicial de validació de la font de persistència. A continuació, es mostra un exemple de com actuar en cas que la font no estigui disponible.



Figura 5. Validació correcta de la font de persistència.

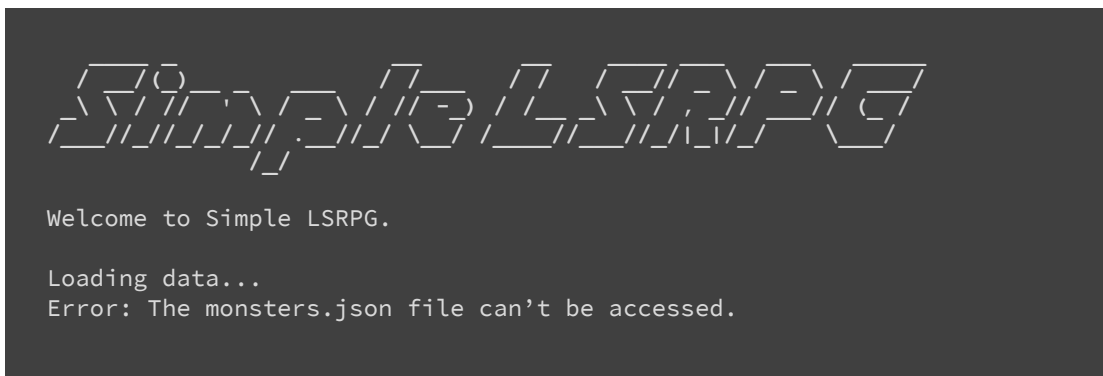


Figura 6. Error en la validació de la font de persistència.

6.2 Menú principal

Un cop validada la font de persistència de monstres, mostrarem el menú principal del programa. A la Figura 7 se n'inclou un exemple, juntament amb el comportament de la funcionalitat de sortir.

A continuació es detallen les quatre funcionalitats principals del programa.


```
The tavern keeper looks at you and says:
"Welcome adventurer! How can I help you?"

    1) Character creation
    2) List characters
    3) Create an adventure
    4) Start an adventure
    5) Exit

Your answer: 5

Tavern keeper: "Are you leaving already? See you soon, adventurer."
```

Figura 7. Menú principal, incloent la funcionalitat de sortir.

6.2.1 Funcionalitat de creació de personatge

Un cop l'usuari triï l'opció 1 del menú principal (Figura 7), el programa procedirà a demanar les dades necessàries, tal com està descrit a l'apartat 2.2. En aquest exemple es mostrarà el procés sense errors en la creació d'un personatge.

```
(...)
Your answer: 1

Tavern keeper: "Oh, so you are new to this land."
"What's your name?"

-> Enter your name: Finrod Felagund

Tavern keeper: "Hello, Finrod Felagund, be welcome."
"And now, if I may break the fourth wall, who is your Player?"

-> Enter the player's name: Eru Ilúvatar

Tavern keeper: "I see, I see..."
"Now, are you an experienced adventurer?"

-> Enter the character's level [1..10]: 3

Tavern keeper: "Oh, so you are level 3!"
"Great, let me get a closer look at you..."

Generating your stats...

Body:   You rolled 2 (1 and 1).
Mind:   You rolled 11 (5 and 6).
Spirit: You rolled 8 (2 and 6).

Your stats are:
- Body: -1
- Mind: +2
- Spirit: +1

The new character Finrod Felagund has been created.
```

Figura 8. Procés de creació d'un personatge.

6.2.2 Funcionalitat de llistat de personatges

En la Figura 9 es mostra un exemple de la funcionalitat que permet als usuaris consultar un llistat de personatges.

```
(...)  
Your answer: 2  
  
Tavern keeper: "Lads! They want to see you!"  
"Who piques your interest?"  
  
-> Enter the name of the Player to filter: Eru  
  
You watch as some adventurers get up from their chairs and approach you.  
  
1. Galadriel  
2. Finrod Felagund  
3. Túrin Turambar  
4. Niënor Níniel  
5. Elendil  
  
0. Back  
  
Who would you like to meet [0..5]:
```

Figura 9. Llistat de personatges d'un jugador, amb comprovació parcial del nom.

A continuació, en la Figura 10 es mostra un segon exemple quan l'usuari prem *enter* en lloc d'introduir un nom de jugador.

```
(...)  
Your answer: 2  
  
Tavern keeper: "Lads! The Boss wants to see you, come here!"  
"Who piques your interest?"  
  
-> Enter the name of the Player to filter:  
  
You watch as all adventurers get up from their chairs and approach you.  
  
1. Hellboy  
2. Galadriel  
3. Luke Skywalker  
4. Finrod Felagund  
5. Túrin Turambar  
6. Niënor Níniel  
7. Percival Fredrickstein von Musel Klossowski de Rolo III  
8. Elendil  
9. Vi  
10. Jinx  
  
0. Back  
  
Who would you like to meet [0..10]:
```

Figura 10. Llistat de tots els personatges.

Informació completa del personatge

Després de l'exemple de la Figura 10, si l'usuari introdueix un número vàlid accedirà a la vista d'informació completa d'un personatge. En la Figura 11 es pot veure un exemple d'aquest procés on, a més a més, es decideix esborrar el personatge.

```
(...)  
Who would you like to meet [0..10]: 10  
  
Tavern keeper: "Hey Jinx get here; the boss wants to see you!"  
  
* Name: Jinx  
* Player: IPlayLOLInClass  
* Class: Adventurer  
* Level: 2  
* XP: 132  
* Body: +1  
* Mind: -1  
* Spirit: +3  
  
[Enter name to delete, or press enter to cancel]  
Do you want to delete Jinx? Jinx  
  
Tavern keeper: "I'm sorry kiddo, but you have to leave."  
  
Character Jinx left the Guild.
```

Figura 11. Informació completa d'un personatge i procés d'esborrat.

6.2.3 Funcionalitat de creació d'una aventura

De forma similar a la funcionalitat de creació de personatges, un usuari pot crear noves aventures. En la Figura 12 es mostra un exemple d'aquest procés sense errors d'usuari.

```
(...)  
Your answer: 3  
  
Tavern keeper: "Planning an adventure? Good luck with that!"  
  
-> Name your adventure: Strolling through Mordor  
  
Tavern keeper: "You plan to undertake Strolling through Mordor, really?"  
"How long will that take?"  
  
-> How many encounters do you want [1..4]: 2  
  
Tavern keeper: "2 encounters? That is too much for me..."  
  
* Encounter 1 / 2  
* Monsters in encounter  
# Empty  
  
1. Add monster  
2. Remove monster  
3. Continue  
  
-> Enter an option [1..3]: 1
```

```

1. Orc (Minion)
2. Easterling (Minion)
3. Warg (Minion)
4. Uruk (Minion)
5. Troll (Lieutenant)
6. Nazgul (Lieutenant)
7. Gothmog (Boss)
8. Morgoth (Boss)

-> Choose a monster to add [1..7]: 1
-> How many Orc(s) do you want to add: 2

* Encounter 1 / 2
* Monsters in encounter
  1. Orc (x2)

1. Add monster
2. Remove monster
3. Continue

-> Enter an option [1..3]: 3

* Encounter 2 / 2
* Monsters in encounter
  # Empty

1. Add monster
2. Remove monster
3. Continue

(...) // Skip

-> Enter an option [1..3]: 3

Tavern keeper: "Great plan lad! I hope you won't die!"

The new adventure Strolling through Mordor has been created.

```

Figura 12. Procés de creació d'una aventura.

Eliminar monstres

Durant el procés de creació d'una aventura que s'ha pogut veure en la Figura 12, existeix la possibilitat d'esborrar monstres d'un combat. Aquest procés s'il·lustra a la Figura 13.

```

* Encounter 2 / 2
* Monsters in encounter
  1. Orc (x5)
  2. Nazgul (x9)
  3. Morgoth (x1)

1. Add monster
2. Remove monster
3. Continue

-> Enter an option [1..3]: 2
-> Which monster do you want to delete: 2

9 Nazgul were removed from the encounter.

```

```

* Encounter 2 / 2
* Monsters in encounter
  1. Orc (x5)
  2. Morgoth (x1)

1. Add monster
2. Remove monster
3. Continue

-> Enter an option [1..3]:

```

Figura 13. Procés d'esborrat de monstres durant la creació d'una aventura.

6.2.4 Funcionalitat de jugar una aventura

Per poder jugar una aventura, en primer lloc, cal assegurar que el sistema disposi de suficients personatges. Si el sistema no disposés d'un mínim de 3 personatges ja creats, es mostraria el missatge que es veu en la Figura 14.

```

The tavern keeper looks at you and says:
"Welcome adventurer! How can I help you?"

1) Character creation
2) List characters
3) Create an adventure
4) Start an adventure (disabled: create 3 characters first)
5) Exit

Your answer:

```

Figura 14: Menú inicial quan encara no hi ha 3 personatges creats.

En cas contrari, si ja existeixen 3 personatges creats, es podrà procedir a iniciar el procés de triar l'aventura que es vol jugar i el grup de personatges o *party* que participaran en ella. En la Figura 15 es pot observar un exemple de procés de selecció d'una aventura i creació de la *party*. Per simplificar l'exemple, s'ha resumit el procés iteratiu de triar 5 personatges.

```

(...)
Your answer: 4

Tavern keeper: "So, you are looking to go on an adventure?"
"Where do you fancy going?"

Available adventures:
  1. Dagor-nuin-Giliath - The Battle under the Stars
  2. The Siege of Angband
  3. Nirnaeth Arnoediad - The Battle of Unnumbered Tears
  4. The War of Wrath
  5. The Battle of Five Armies
  6. The Fellowship of the Ring

-> Choose an adventure: 1

```

Tavern keeper: "Dagor-nuin-Giliath – The Battle under the Stars it is!"
"And how many people shall join you?"

-> Choose a number of characters [3..5]: 5

Tavern keeper: "Great, 5 it is."
"Who among these lads shall join you?"

Your party (0 / 5):

1. Empty
2. Empty
3. Empty
4. Empty
5. Empty

Available characters:

1. Fëanor
2. Galadriel
3. Maedhros
4. Finrod Felagund
5. Túrin Turambar
6. Niënor Níniel
7. Caranthir
8. Elendil
9. Curufin
10. Amrod

-> Choose character 1 in your party: 1

Your party (1 / 5):

1. Fëanor
2. Empty
3. Empty
4. Empty
5. Empty

Available characters:

1. Fëanor
2. Galadriel
3. Maedhros
4. Finrod Felagund
5. Túrin Turambar
6. Niënor Níniel
7. Caranthir
8. Elendil
9. Curufin
10. Amrod

-> Choose character 2 in your party: 3

(SKIP)

Your party (5 / 5):

1. Fëanor
2. Maedhros
3. Caranthir
4. Curufin

```
5. Amrod
-----

Tavern keeper: "Great, good luck on your adventure lads!"

The "Dagor-nuin-Giliath - The Battle under the Stars" will start soon...
```

Figura 15: Exemple d'elecció d'una aventura i selecció dels personatges.

A continuació es procedirà a mostrar un exemple on la mateixa *party* que s'ha creat durant l'exemple de la Figura 15, executa un dels combats configurats dins l'aventura triada.

The “Dagor-nuin-Giliath – The Battle under the Stars” will start soon...

Starting Encounter 1:

- 6x Orc
 - 1x Troll
-

*** Preparation stage ***

Fëanor uses Self-Motivated. Their Spirit increases in +1.
Maedhros uses Self-Motivated. Their Spirit increases in +1.
Caranthir uses Self-Motivated. Their Spirit increases in +1.
Curufin uses Self-Motivated. Their Spirit increases in +1.
Amrod uses Self-Motivated. Their Spirit increases in +1.

Rolling initiative...

- 13 Amrod
- 12 Orc
- 11 Fëanor
- 10 Caranthir
- 10 Orc
- 10 Maedhros
- 10 Curufin
- 4 Troll
- 3 Orc
- 2 Orc
- 1 Orc
- 1 Orc

*** Combat stage ***

Round 1:

Party:

- Fëanor 55 / 55 hit points
- Maedhros 40 / 40 hit points
- Caranthir 48 / 48 hit points
- Curufin 33 / 33 hit points
- Amrod 24 / 24 hit points

Amrod attacks Orc with Sword slash.
Hits and deals 4 physical damage.

Orc attacks Curufin.
Hits and deals 3 physical damage.

Fëanor attacks Orc with Sword slash.
Hits and deals 7 physical damage.
Orc dies.

Curufin attacks Orc with Sword slash.
Critical hit and deals 12 physical damage.
Orc dies.

Orc attacks Maedhros.
Hits and deals 2 physical damage.

Maedhros attacks Orc with Sword slash.

Hits and deals 6 physical damage.

Curufin attacks Orc with Sword slash.
Hits and deals 5 physical damage.
Orc dies.

Troll attacks Amrod.
Critical hit and deals 29 physical damage.
Amrod falls unconscious.

Orc attacks Maedhros.
Fails and deals 0 physical damage.

End of round 1.

Round 2:

Party:

- Fëanor	55 / 55 hit points
- Maedhros	38 / 40 hit points
- Caranthir	48 / 48 hit points
- Curufin	30 / 33 hit points
- Amrod	0 / 24 hit points

(SKIP)

End of round 4.
All enemies are defeated.

*** Short rest stage ***

Fëanor gains 22 xp.
Maedhros gains 22 xp.
Caranthir gains 22 xp.
Curufin gains 22 xp. Curufin levels up. They are now lvl 4!
Amrod gains 22 xp.

Fëanor uses Bandage time. Heals 7 hit points.
Maedhros uses Bandage time. Heals 4 hit points.
Caranthir uses Bandage time. Heals 6 hit points.
Curufin uses Bandage time. Heals 9 hit points.
Amrod is unconscious.

Starting Encounter 2:
- 3x Orc
- 2x Easterling
- 1x Gothmog

(SKIP)

End of round 9.
All enemies are defeated.

*** Short rest stage ***

Fëanor gains 61 xp.
Maedhros gains 61 xp. Maedhros levels up. They are now lvl 5!
Caranthir gains 61 xp.

```
Curufin gains 61 xp.  
Amrod gains 61 xp. Amrod levels up. They are now lvl 3!  
  
Fëanor is unconscious  
Maedhros uses Bandage time. Heals 7 hit points.  
Caranthir uses Bandage time. Heals 2 hit points.  
Curufin is unconscious  
Amrod uses Bandage time. Heals 3 hit points.  
  
Congratulations, your party completed "Dagor-nuin-Giliath - The Battle  
under the Stars"
```

Figura 16: Exemple d'un combat amb les seves fases

Si en algun moment la *party* fos derrotada, caldria mostrar un missatge indicant que s'ha donat un TPU, finalitzar l'aventura i tornar al menú principal.

```
Tavern keeper: "Lad, wake up. Yes, your party fell unconscious."  
"Don't worry, you are safe back at the Tavern."
```

Figura 17: Exemple d'una aventura finalitzada per un TPU.