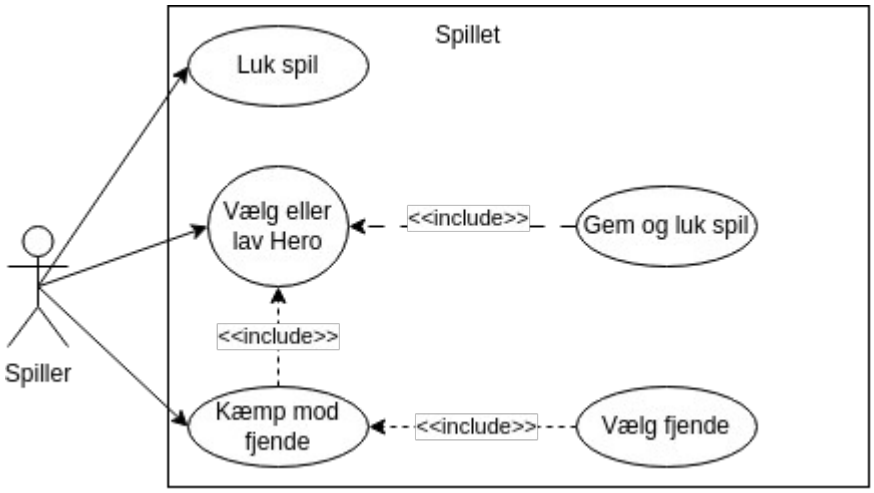
Første iteration

Første iteration skal indeholde muligheden for at lave en karakter eller vælge en tidligere karakter. Derefter skal man kunne vælge at kæmpe mod en fjende eller forlade og gemme sine fremskridt. Hver fjende har en mængde xp som gives til ens karakter når fjenden bliver besejret.
Ud fra at dette er der lavet tre diagrammer:

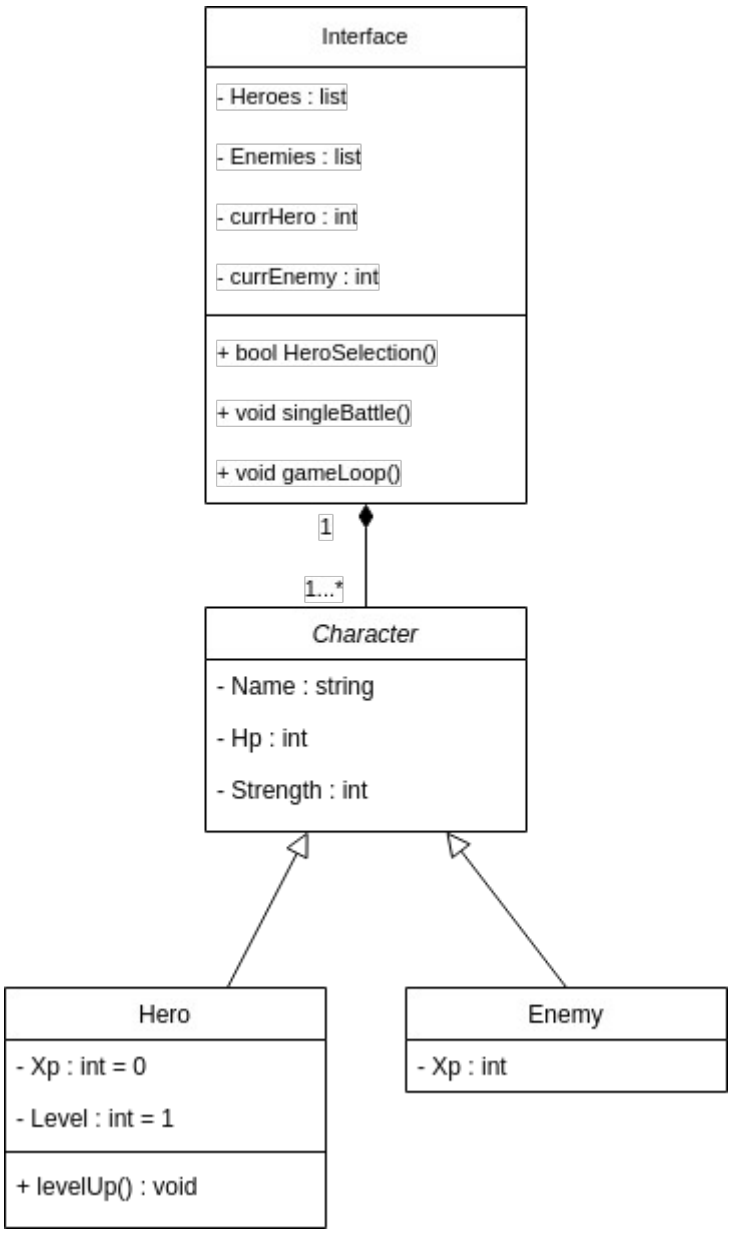


Diagrammet viser hvad man som spiller har mulighed for. Man kan i starten kun vælge mellem at lukke spillet eller vælge en karakter. Derefter har man mulighed for at vælge mellem at kæmpe mod en fjende eller lukke spillet og gemme.
Hvis der vælges at kæmpe mod en fjende kan man derefter vælge hvilken fjende man vil kæmpe imod.

Hero	
PK	**Hero_id int NOT NULL**
	Name char
	Hp int NOT NULL
	Strength int NOT NULL
	Level int NOT NULL
	Xp int

Enemy	
PK	**Enemy_id int NOT NULL**
	Name char(125)
	Hp int NOT NULL
	Strength int NOT NULL
	Xp int NOT NULL

Databaserne der indeholder karaktererne og fjenderne har ikke noget med hinanden at gøre men eksisterer separat fra hinanden.



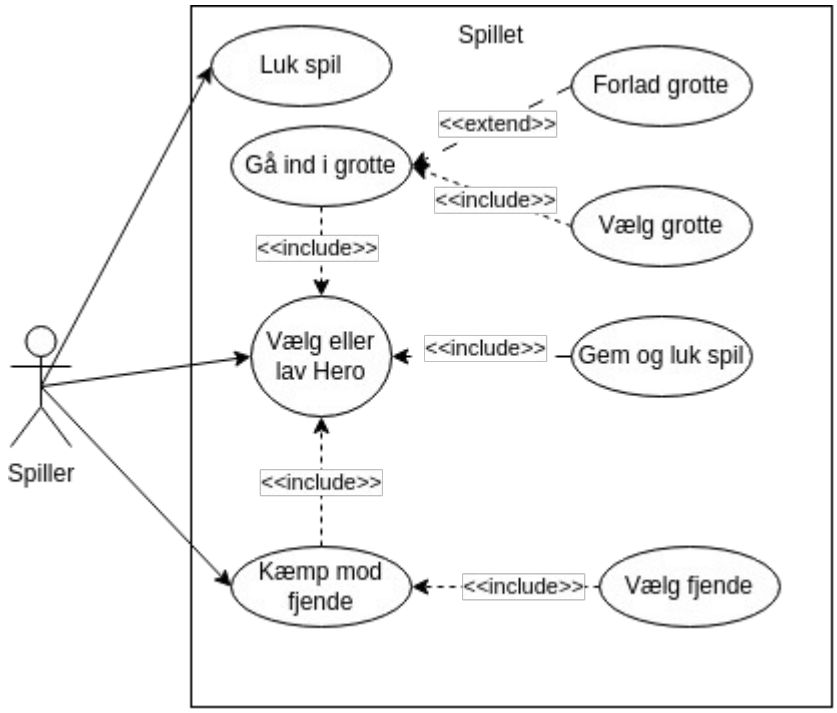
Der er lavet fire klasser i programmet til at håndtere spillet.
Den første klasse hvor det hele foregår er i Interface, der håndtere hvilke karakterer man kan vælge, og hvilke fjende man kan kæmpe imod. Derudover håndterer den også logikken for kampen.

Klassen Character indeholder nogle standard variabler som både Hero og Enemy klassen benytter sig af og er derfor lavet sådan at de to klasser kan nedarve disse variable samt getter funktionerne.

Klassen Hero har en unik funktion i og med at ens karakter kan nå et højere niveau: Level.
Klassen Enemy har noget xp som bliver sat ud fra databasen.

Anden iteration

Anden iteration tilføjer grotter til spillet. Udover at man kan kæmpe mod enkelte fjender er det nu også muligt at gå ind i en grotte, og ved at kæmpe sig til enden kan man blive belønnet med en mængde guld. I dette spil bliver grottens fjender tilfældigt genereret ud fra grottens id.
En grotte indeholder et tilfældigt antal fjender, mellem 4 og 7, hvor den sidste fjende er den sværeste. Fjenderne bliver valgt ud fra gottens ID hvor den nemmeste grotte kan indeholde den svageste og næst svageste, med den tredje svgeste som boss til sidst.
Nummer to grotte kan indeholde den næst svageste og tredje svageste, med den 4 svageste som boss til sidst.
Grotten med dragen indeholder kun Ape King fjender, med dragen som boss til sidst.
Ud fra dette er der lavet tre diagrammer:



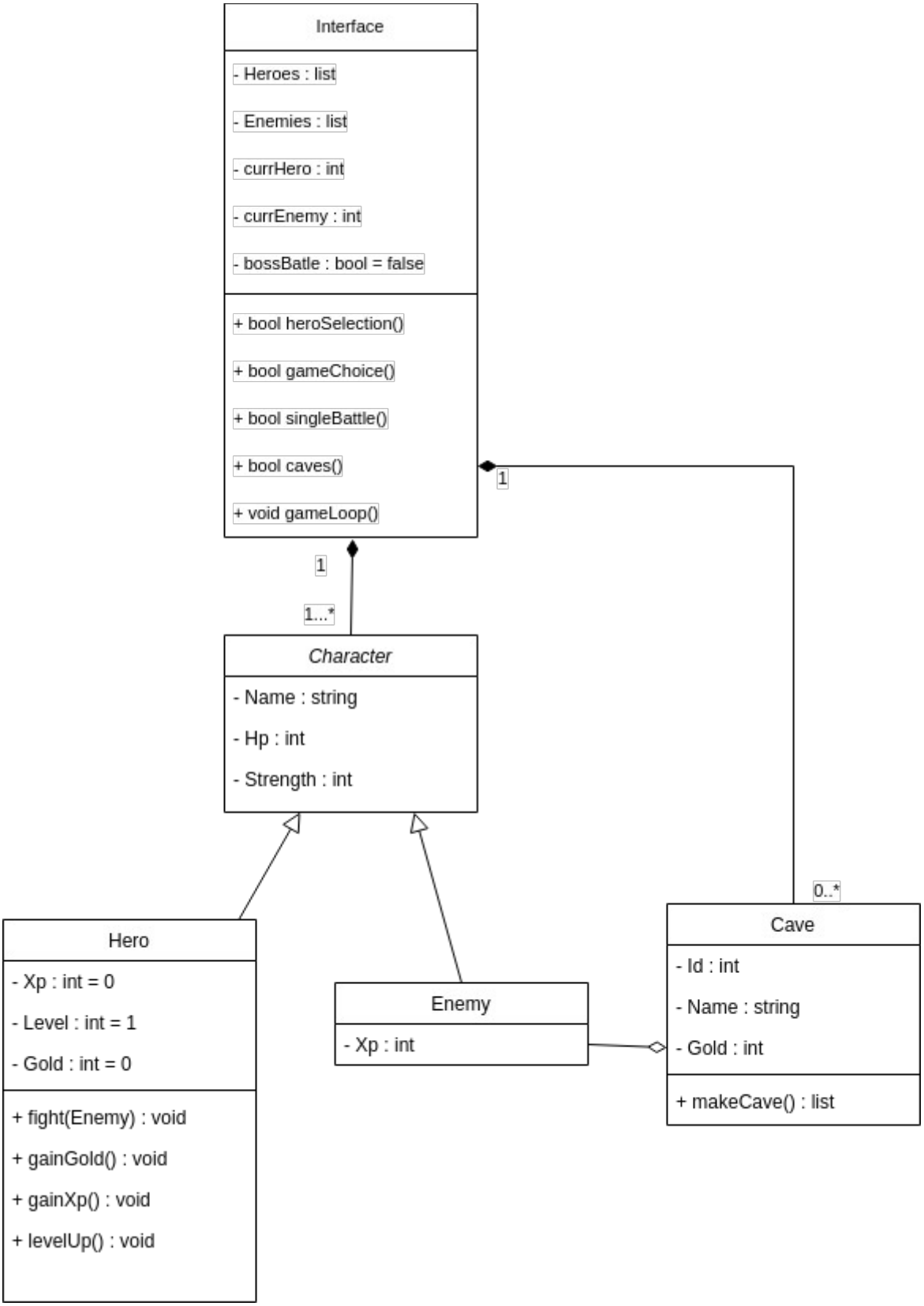
I use-case diagrammet kan det ses at man stadig skal vælge en karakter, men derefter kan man også vælge at gå ind i en grotte. Hvis dette er det valg man tager, så er der mulighed for at forlade grotten, eller at vælge hvilken grotte man vil udfordre.

Hero	
PK	**Hero_id int NOT NULL**
	Name char
	Hp int NOT NULL
	Strength int NOT NULL
	Level int NOT NULL
	Xp int
	Gold int

Cave	
PK	**Cave_id int NOT NULL**
	Name char(125)
	Gold int

Enemy	
PK	**Enemy_id int NOT NULL**
	Name char(125)
	Hp int NOT NULL
	Strength int NOT NULL
	Xp int NOT NULL

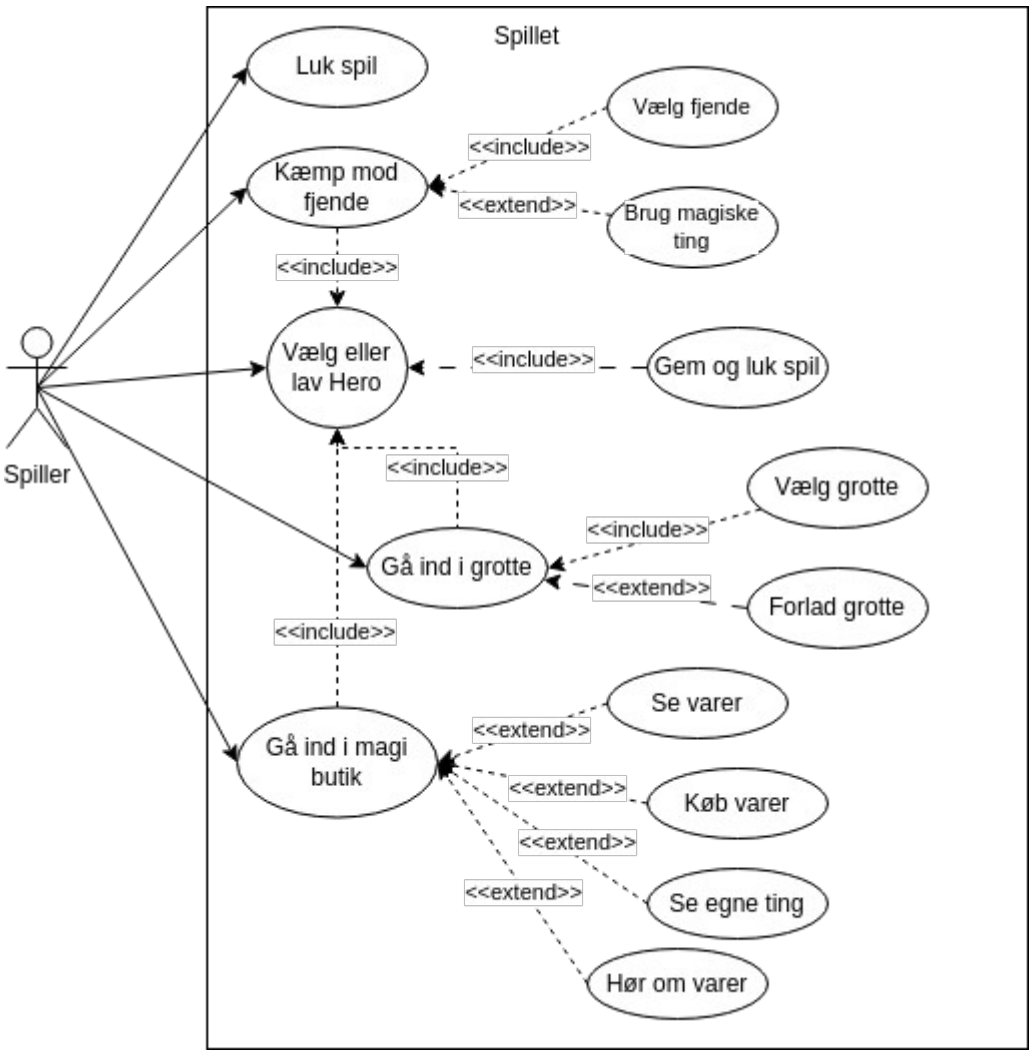
Da fjenderne i grotten bliver tilfældigt generet i koden, så er der stadig ingen af databaserne, der benytter hinanden



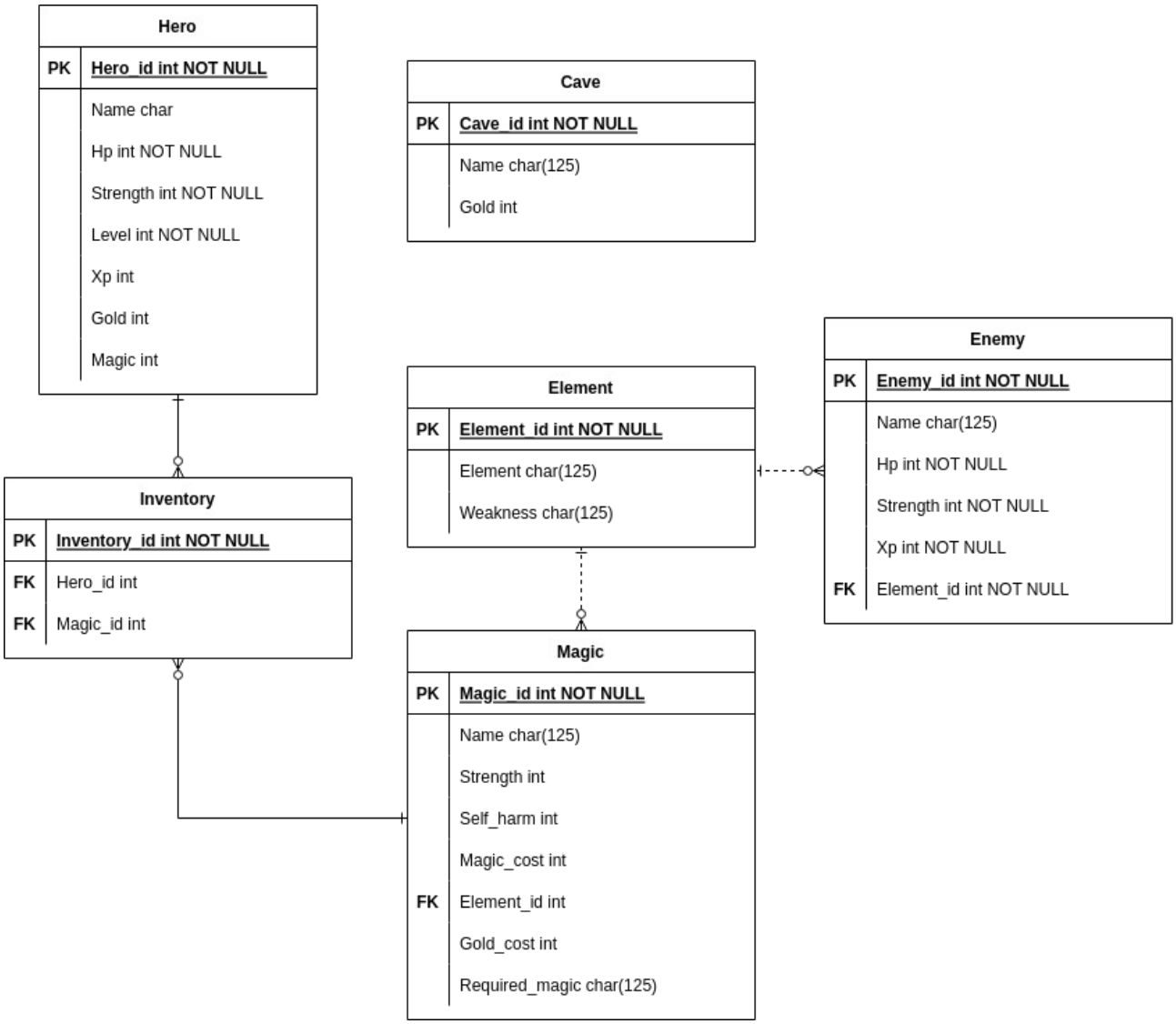
Der er blevet oprettet en klasse mere, hvor grotterne bliver fyldt med fjender ud fra den tidligere nævnte logik. Grotten indeholder et id, et navn og noget guld. Grotten er afhængig af fjender, da der ikke kan konstrueres en grotte uden fjender, men fjender kan godt eksistere uden for grotten. Fight-funktionen er blevet rykket over til Hero, hvori logikken nu bliver håndteret og der skal bruges en Enemy for at kunne lave kaldet på denne funktion.

Tredje iteration

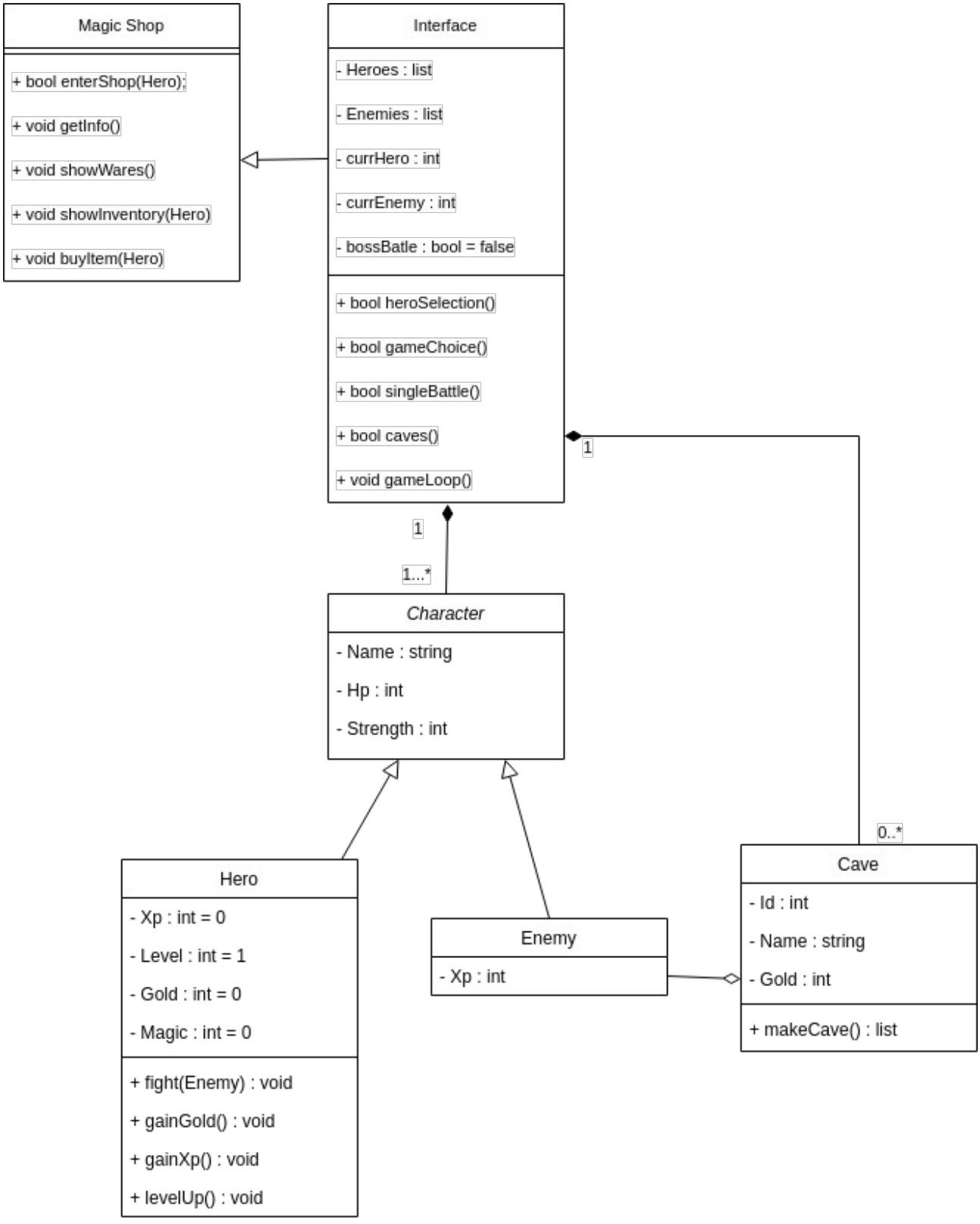
I tredje iteration tilføjes et magisystem, hvor de forskellige magier har styrker og svagheder, fjender har et element tilknyttet sig, og der er mulighed for at købe magi i en magisk butik.
I magibtuikken er der mulighed for at se hvilke varer der er, hvad de forskellige varer kræver og gør, købe ting og se hvilke ting man selv har.
Ud fra dette er der lavet tre diagrammer:



Butikken er tilføjet til use-casen og når man kæmper mod en fjende har man nu mulighed for at vælge at bruge sine magiske ting.



Der er tilføjet tre nye tabeller i databasen.
Den første der er tilføjet er Element, der indeholder navn på elementet og navn på hvilket element den er svag over for. Enemy tabellen referer til element_id'et for det element som enemy tilhører.
Den anden tabel er Magic. Denne tabel indeholder alt information om de ting, der kan købes i magibutikken. Derudover referer den også med en foreign key til element ud fra hvilket element magien er.
Den tredje tabel er Inventory. Denne holder styr på hvilke magier der tilhører en karakter. Denne referer både til hero_id og magic_id.



I UML diagrammet er der tilføjet en ny klasse: Magic Shop. Denne klasses egenskaber arves af interface sådan så alt der har med butikken at gøre kan kaldes direkte fra interface.
Der er også tilføjet Magic som variabel til Hero.