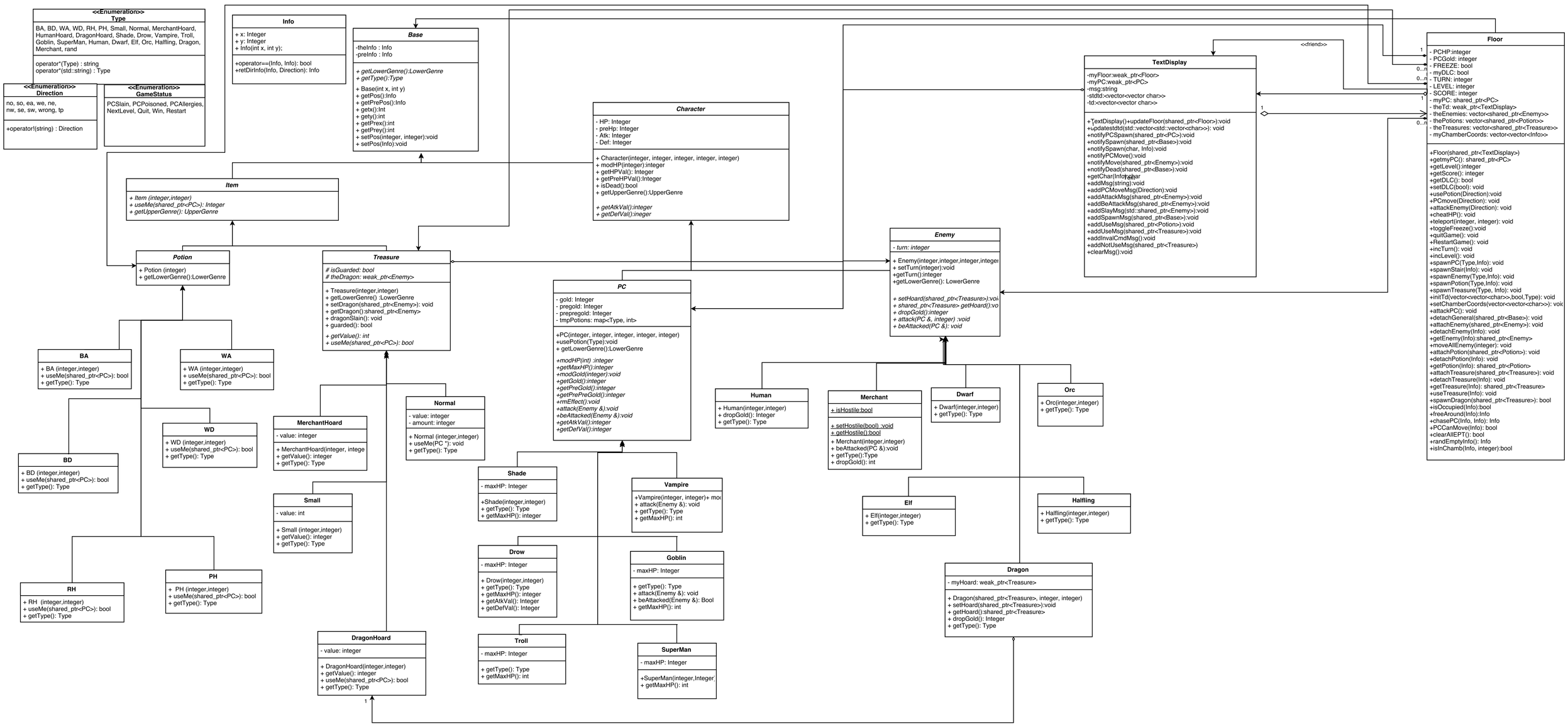
<<Enumeration>>
Type
BA, BD, WA, WD, RH, PH, Small, Normal, MerchantHoard, HumanHoard, DragonHoard, Shade, Drow, Vampire, Troll, Goblin, SuperMan, Human, Dwarf, Elf, Orc, Halfling, Dragon, Merchant, rand
operator*(Type) : string
operator*(std::string) : Type
<<Enumeration>>
Direction
no, so, ea, we, ne, nw, se, sw, wrong, tp
+operator!(string) : Direction
<<Enumeration>>
GameStatus
PCSlain, PCPoisoned, PCAllergies, NextLevel, Quit, Win, Restart
Info
+ x: Integer
+ y: Integer
+ Info(int x, int y);
+operator==(Info, Info): bool
+retDirInfo(Info, Direction): Info
Base
-theInfo : Info
-preInfo : Info
+ getLowerGenre():LowerGenre
+ getType():Type
+ Base(int x, int y)
+ getPos():Info
+ getPrePos():Info
+ getx():Int
+ gety():int
+ getPrex():int
+ getPrey():int
+ setPos(integer, integer):void
+ setPos(Info):void
Character
- HP: Integer
- preHp: Integer
- Atk: Integer
- Def: Integer
+ Character(integer, integer, integer, integer, integer)
+ modHP(integer):integer
+ getHPVal(): Integer
+ getPreHPVal():Integer
+ isDead():bool
+ getUpperGenre():UpperGenre
+ getAtkVal():integer
+ getDefVal():integer
TextDisplay
-myFloor:weak_ptr<Floor>
-myPC:weak_ptr<PC>
-msg:string
-stdtd:<vector<vector char>>
-td:<vector<vector char>>
+TextDisplay()+updateFloor(shared_ptr<Floor>):void
+updateStdtd(std::vector<std::vector<char>>): void
+notifyPCSpawn(shared_ptr<PC>):void
+notifySpawn(shared_ptr<Base>):void
+notifySpawn(char, Info):void
+notifyPCMove():void
+notifyMove(shared_ptr<Enemy>):void
+notifyDead(shared_ptr<Base>):void
+getChar(Info):char
+addMsg(string):void
+addPCMoveMsg(Direction):void
+addAttackMsg(shared_ptr<Enemy>):void
+addBeAttackMsg(shared_ptr<Enemy>):void
+addSlayMsg(std::shared_ptr<Enemy>):void
+addSpawnMsg(shared_ptr<Base>):void
+addUseMsg(shared_ptr<Potion>):void
+addUseMsg(shared_ptr<Treasure>):void
+addInvalCmdMsg():void
+addNotUseMsg(shared_ptr<Treasure>)
+clearMsg():void
<<friend>>
Floor
- PCHP:integer
- PCGold: integer
- FREEZE: bool
- myDLC: bool
- TURN: integer
- LEVEL: integer
- SCORE: integer
- myPC: shared_ptr<PC>
- theTd: weak_ptr<TextDisplay>
- theEnemies: vector<shared_ptr<Enemy>>
- thePotions: vector<shared_ptr<Potion>>
- theTreasures: vector<shared_ptr<Treasure>>
- myChamberCoords: vector<vector<Info>>
+Floor(shared_ptr<TextDisplay>)
+getmyPC(): shared_ptr<PC>
+getLevel():integer
+getScore(): integer
+getDLC(): bool
+setDLC(bool): void
+usePotion(Direction):void
+PCmove(Direction): void
+attackEnemy(Direction): void
+cheatHP(): void
+teleport(integer, integer): void
+toggleFreeze():void
+quitGame(): void
+RestartGame(): void
+incTurn(): void
+incLevel(): void
+spawnPC(Type,Info): void
+spawnStair(Info): void
+spawnEnemy(Type,Info): void
+spawnPotion(Type,Info): void
+spawnTreasure(Type, Info): void
+initTd(vector<vector<char>>,bool,Type): void
+setChamberCoords(vector<vector<char>>): void
+attackPC(): void
+detachGeneral(shared_ptr<Base>): void
+attachEnemy(shared_ptr<Enemy>): void
+detachEnemy(Info): void
+getEnemy(Info):shared_ptr<Enemy>
+moveAllEnemy(integer): void
+attachPotion(shared_ptr<Potion>): void
+detachPotion(Info): void
+getPotion(Info): shared_ptr<Potion>
+attachTreasure(shared_ptr<Treasure>): void
+detachTreasure(Info): void
+getTreasure(Info): shared_ptr<Treasure>
+useTreasure(Info): void
+spawnDragon(shared_ptr<Treasure>): bool
+isOccupied(Info):bool
+freeAround(Info):Info
+chasePC(Info, Info): Info
+PCCanMove(Info): bool
+clearAllEPT(): bool
+randEmptyInfo(): Info
+isInChamb(Info, integer):bool
Item
+ Item (integer,integer)
+ useMe(shared_ptr<PC>): Integer
+ getUpperGenre(): UpperGenre
Potion
+ Potion (integer)
+ getLowerGenre():LowerGenre
Treasure
isGuarded: bool
theDragon: weak_ptr<Enemy>
+ Treasure(integer,integer)
+ getLowerGenre() :LowerGenre
+ setDragon(shared_ptr<Enemy>): void
+ getDragon():shared_ptr<Enemy>
+ dragonSlain(): void
+ guarded(): bool
+ getValue(): int
+ useMe(shared_ptr<PC>): bool
Enemy
- turn: integer
+ Enemy(integer,integer,integer,integer
+ setTurn(integer):void
+getTurn():integer
+getLowerGenre(): LowerGenre
+ setHoard(shared_ptr<Treasure>):void
+ shared_ptr<Treasure> getHoard():void
+ dropGold():integer
+ attack(PC &, integer) :void
+ beAttacked(PC &): void
PC
- gold: Integer
- pregold: Integer
- prepregold: Integer
- tmpPotions: map<Type, int>
+PC(integer, integer, integer, integer, integer)
+usePotion(Type):void
+ getLowerGenre():LowerGenre
+modHP(int) :integer
+getMaxHP():integer
+modGold(integer):void
+getGold():integer
+getPreGold():integer
+getPrePreGold():integer
+rmEffect():void
+attack(Enemy &):void
+beAttacked(Enemy &):void
+getAtkVal():integer
+getDefVal():integer
BA
+ BA (integer,integer)
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
WA
+ WA (integer,integer)
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
WD
+ WD (integer,integer)
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
BD
+ BD (integer,integer)
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
RH
+ RH (integer,integer)
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
PH
+ PH (integer,integer)
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
MerchantHoard
- value: integer
+ MerchantHoard(integer, intege
+ getValue(): integer
+ getType(): Type
Normal
- value: integer
- amount: integer
+ Normal (integer,integer)
+ useMe(PC *): void
+ getType(): Type
Small
- value: int
+ Small (integer,integer)
+ getValue(): integer
+ getType(): Type
DragonHoard
- value: integer
+ DragonHoard(integer,integer)
+ getValue(): integer
+ useMe(shared_ptr<PC>): bool
+ getType(): Type
1
Shade
- maxHP: Integer
+Shade(integer,integer)
+ getType(): Type
+ getMaxHP(): integer
Vampire
+Vampire(integer, integer)+ mod
+ attack(Enemy &): void
+ getType(): Type
+ getMaxHP(): int
Drow
- maxHP: Integer
+ Drow(integer,integer)
+ getType(): Type
+ getMaxHP(): integer
+ getAtkVal(): Integer
+ getDefVal(): Integer
Goblin
- maxHP: Integer
+ getType(): Type
+ attack(Enemy &): void
+ beAttacked(Enemy &): Bool
+ getMaxHP(): int
Troll
- maxHP: Integer
+ getType(): Type
+ getMaxHP(): int
SuperMan
- maxHP: Integer
+SuperMan(integer,Integer,
+ getMaxHP(): int
Human
+ Human(integer,integer)
+ dropGold(): Integer
+ getType(): Type
Merchant
+ isHostile:bool
+ setHostile(bool) :void
+ getHostile():bool
+ Merchant(integer,integer)
+ beAttacked(PC &):void
+ getType():Type
+ dropGold(): int
Dwarf
+ Dwarf(integer,integer)
+ getType(): Type
Orc
+ Orc(integer,integer)
+ getType(): Type
Elf
+ Elf(integer,integer)
+ getType(): Type
Halfling
+ Halfling(integer,integer)
+ getType(): Type
Dragon
- myHoard: weak_ptr<Treasure>
+ Dragon(shared_ptr<Treasure>, integer, integer)
+ setHoard(shared_ptr<Treasure>):void
+ getHoard():shared_ptr<Treasure>
+ dropGold(): Integer
+ getType(): Type
1
0...n
0...n
1
1
0...n