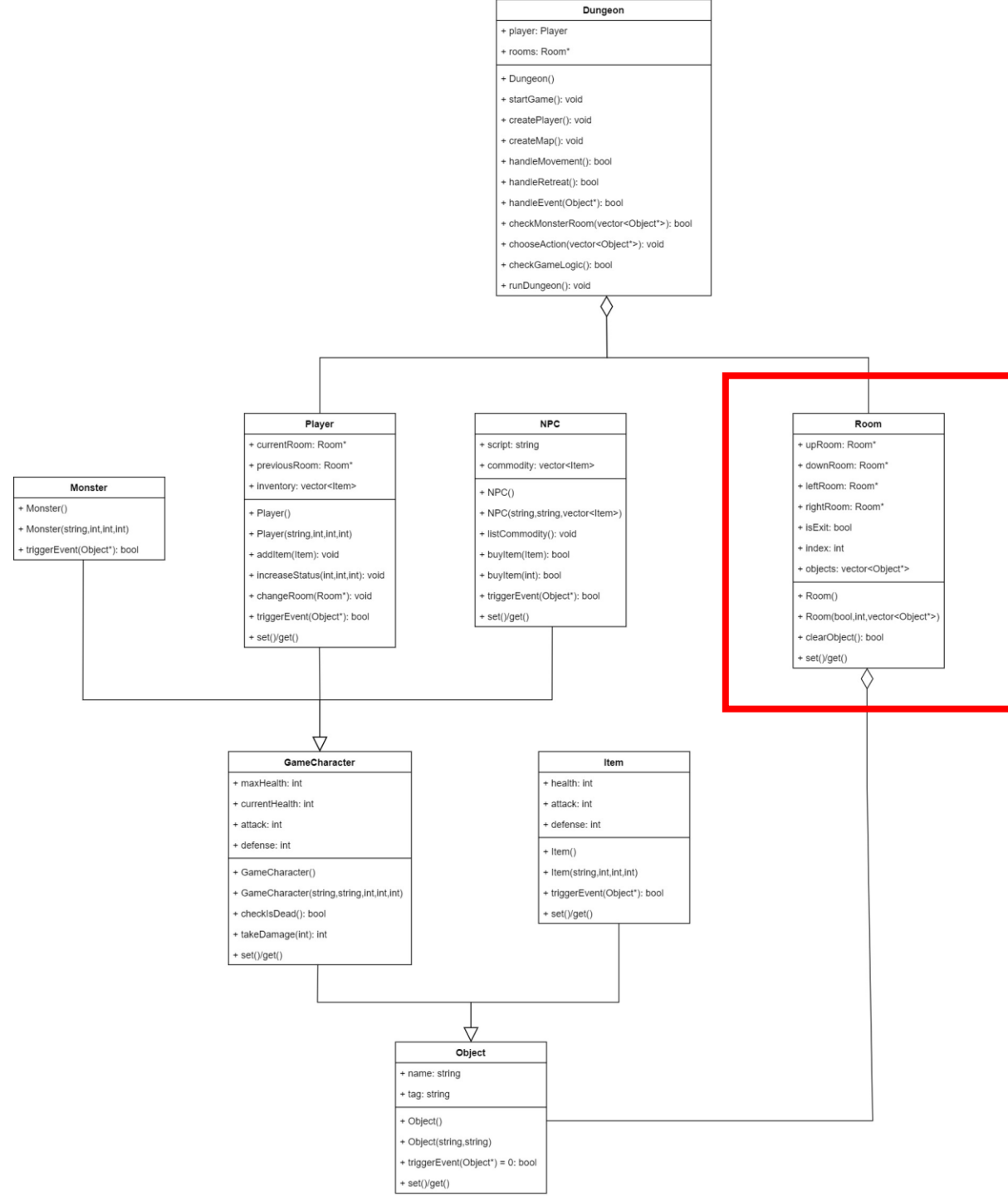


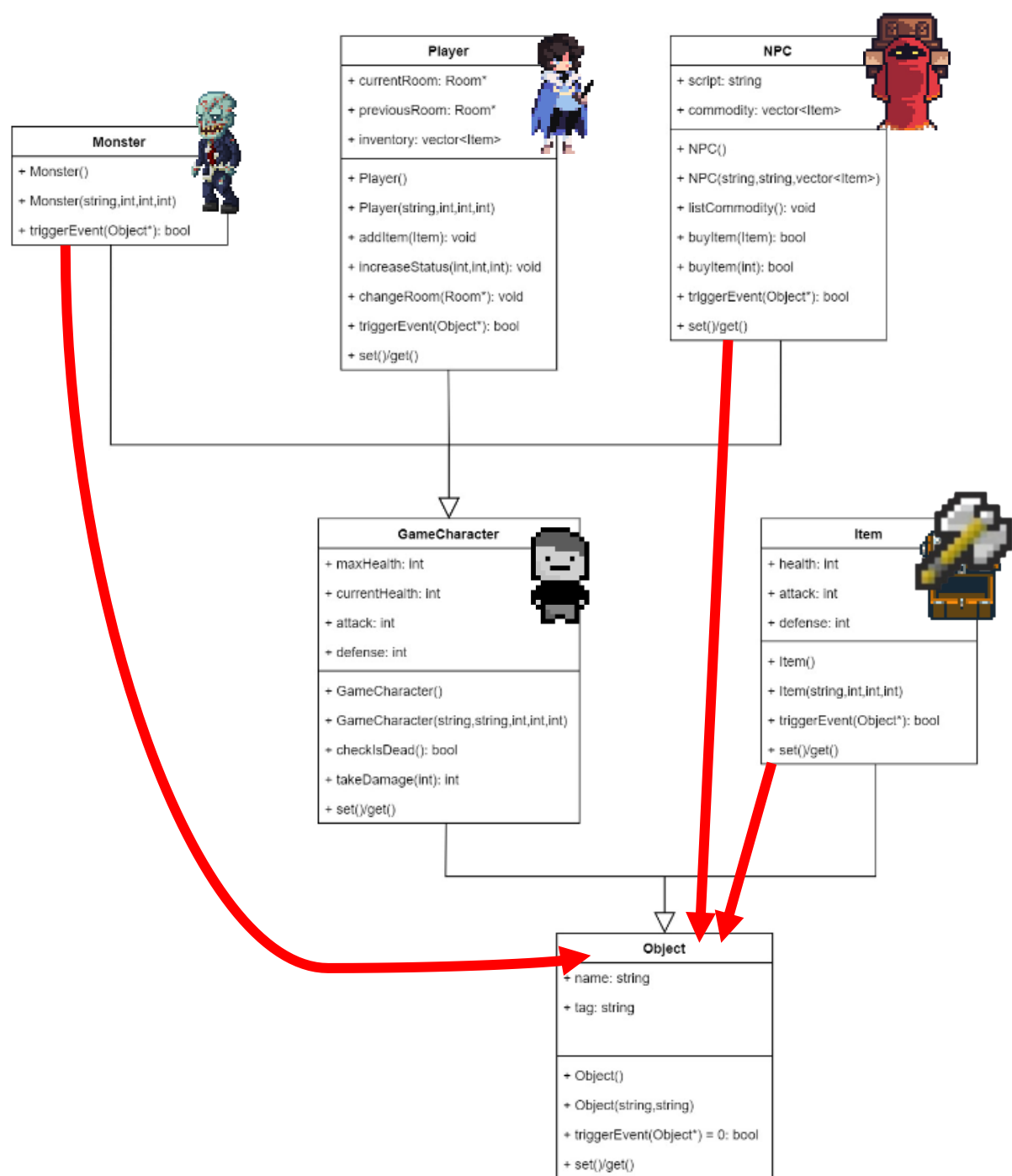
Dungeon

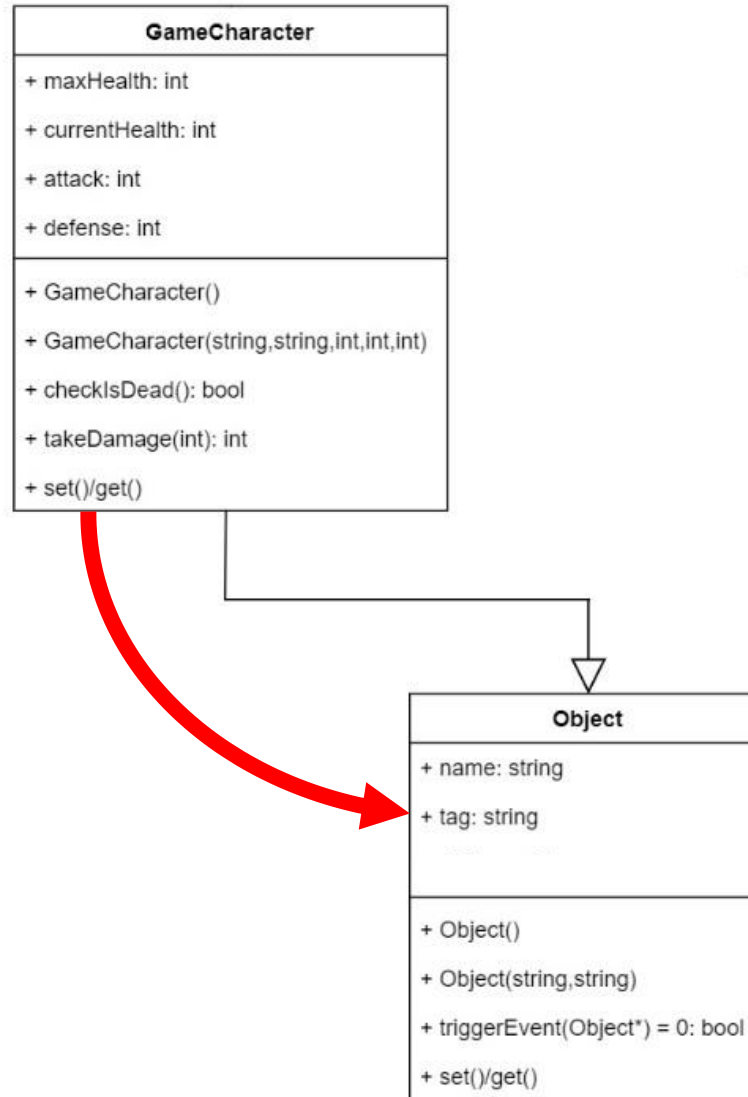


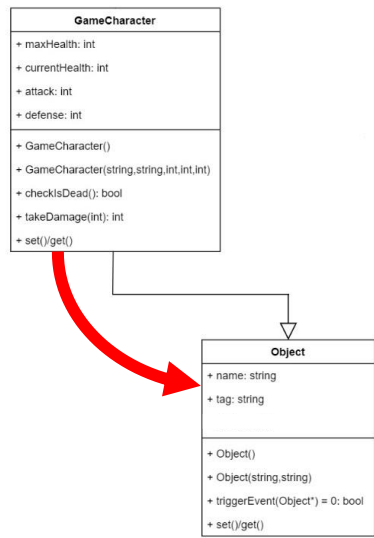
Room

- + upRoom: Room*
- + downRoom: Room*
- + leftRoom: Room*
- + rightRoom: Room*
- + isExit: bool
- + index: int
- + objects: vector<Object*>

- + Room()
- + Room(bool,int,vector<Object*>)
- + clearObject(): bool
- + set()/get()







```
41      GameCharacter* charactor = new GameCharacter();
```

GameCharacter*



```
43      Object* obj = (Object*) charactor;
```

Object*



```
43      Object* obj = (Object*) charactor;
```

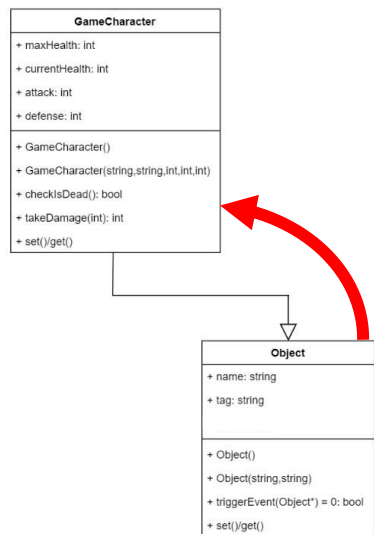
Object*



```
45      GameCharacter* role = (GameCharacter*) obj;
```

GameCharacter*





45

```
Object* obj = new Object();
```

※ 假設Object不是abstract class
否則會compile error

Object*



47

```
GameCharacter* charactor = (GameCharacter*) obj;
```

GameCharacter*



不安全的轉換

Object*



name	tag	maxhealth	currenthealth	attack	defense
------	-----	-----------	---------------	--------	---------

GameCharactor*

name	tag	maxhealth	currenthealth	attack	defense
------	-----	-----------	---------------	--------	---------



Object*



name	tag
------	-----

GameCharactor*

name	tag	maxhealth	currenthealth	attack	defense
------	-----	-----------	---------------	--------	---------

不安全

無型別檢查，直接進行型別轉換

```
41  GameCharacter* character = new GameCharacter();  
42  Object* obj = (Object*) character;
```

有型別檢查，若有問題則回傳空指標

```
44  GameCharacter* character = new GameCharacter();  
45  Object* obj = dynamic_cast<Object*>(character);
```

```
40     Object* obj = new Object();
41     //GameCharactor* charactor = new GameCharactor();
42     GameCharactor* noTypeCheck = (GameCharactor*) obj;
43     GameCharactor* TypeCheck = dynamic_cast<GameCharactor*>(obj);
44     cout << "    obj's addr: " << obj << endl;
45     cout << "no Type Check: " << noTypeCheck << endl;
46     cout << "    Type Check: " << TypeCheck << endl;
```

```
obj's addr: 0xed1be8
no Type Check: 0xed1be8
Type Check: 0
```

1. 子類轉父類(ex GameCharactor -> Object)

不會有問題，可以直接轉

2. 父類轉子類(ex Object -> GameCharactor)

須注意不要存取不存在的屬性

3. 若不確定，可使用dynamic_cast