

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

-----oOo-----



PHƯƠNG PHÁP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
Đề án: Thi cuối kì môn học

Sinh viên thực hiện:

- Nguyễn Thanh Phong (20127274)

Mục Lục

I.	Mô tả đồ án	3
II.	Sơ đồ UML và mô tả chi tiết.....	4
	1. Sơ đồ UML.....	4
	2. Mô tả chi tiết	5
III.	Khả năng mở rộng	6
IV.	Phần chưa hoàn thành	7

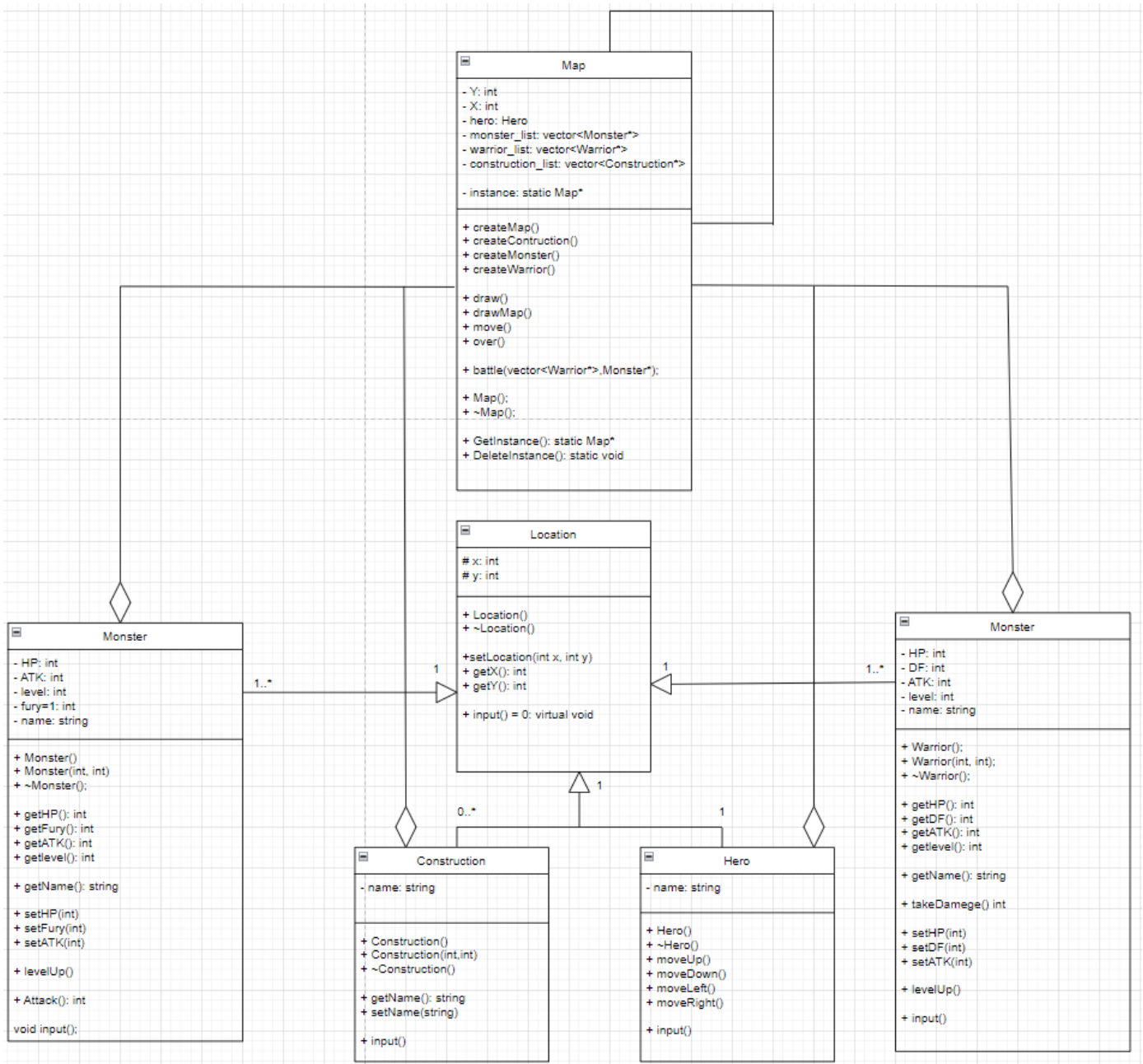
A. Đồ án

I. Mô tả đồ án

Khởi đầu trò chơi, người chơi sẽ nhập kích thước của map. Tiếp theo thì người chơi sẽ nhập thông tin và tọa độ của các chiến binh, quái vật, công trình. Trò chơi sẽ bắt đầu khi người chơi nhập thành công. Người chơi sẽ nhập “W”, “A”, “S”, “D” để di chuyển đến các địa điểm trên bảng đồ. Khi di chuyển đến các ô quái vật và khiêu chiến thành công thì sẽ chiến thắng. Ngược lại, nếu tất cả các chiến binh bị tiêu diệt thì sẽ thua.

II. Sơ đồ UML và mô tả chi tiết

a. Sơ đồ UML



* ghi chú: Do trên draw.io khi viết nghiêng thì sẽ nghiêng toàn bộ khung text nên em không viết nghiêng cho phương thức thuần ảo được, em để luôn virtual

b. Mô tả chi tiết

1. Lớp Location

Lớp cha Location chứa thuộc tính `int x`, `int y` để cho việc các lớp khác kế thừa, bên cạnh đó để thuận tiện cho việc nhập, lớp Location còn có phương thức thuần ảo `input()`.

2. Lớp Monster

Được kế thừa từ lớp Location, lớp Monster có các thuộc tính đề yêu cầu như: `hp`, `atk`, `level`, `name` và `fury` mặc định là 1. Bên cạnh các phương thức cơ bản `get`, `set`, `constructor` thì có thêm 2 phương thức là `Attack()`, giá trị trả về là lượng sát thương của đối tượng `Monster(int)` theo công thức đề bài yêu cầu và `levelUp()` tăng chỉ số và `level` khi đạt đủ điều kiện

3. Lớp Warrior

Tương tự như lớp Monster, lớp Warrior cũng được kế thừa từ lớp Monster, các thuộc tính cơ bản cũng giống lớp Monster chỉ khác Warrior có `DF` còn Monster có `fury`. Ngoài các phương thức cơ bản ra, Warrior có 1 phương thức khác là `takeDamage()`, trả về số lượng máu + giáp (`int`).

4. Lớp Construction

Được kế thừa từ lớp Location, không có các phương thức khác ngoài các phương thức cơ bản

5. Lớp Hero

Được kế thừa từ lớp Location, có các phương thức `moveUp()`, `moveDown()`, `moveLeft()`, `moveRight()` để thay đổi vị trí `x`, `y` tùy theo phương thức

6. Lớp Map

Là lớp quan trọng nhất, lớp này chứa hoạt động của trò chơi

Lớp Map được thiết kế theo mẫu thiết kế Singleton (Singleton Design Pattern) để đảm bảo Map chỉ có 1 đối tượng duy nhất cho toàn bộ chương trình

Lớp Map có các thuộc tính như diện tích của bản đồ (`int x`, `int y`), có 1 nhân vật người chơi (`hero`), có danh sách con trỏ quái vật, chiến binh, công trình(`vector<Monster*>`, `vector<Warrior*>`, `vector<Construction*>`)

Lớp Map có các phương thức như :

- `createMap()`: Dùng để nhập diện tích của bản đồ
- `createConstruction()`: Dùng để tạo các công trình. Đầu tiên chương trình sẽ tạo ngẫu nhiên số lượng công trình dựa theo diện tích. Tiếp theo là cho người chơi nhập thông tin của các công trình
- `createMonster()`: Dùng để tạo quái vật. Do được chạy đầu tiên nên toàn bộ quái vật sẽ được tạo vị trí 1 cách ngẫu nhiên khác ô bắt đầu của người chơi (`!= (0,0)`)
- `createWarrior()`: Dùng để tạo chiến binh. Tương tự như tạo công trình, phải nhập thủ công
- `move()`: Nhập kí tự từ bàn phím (`W,A,S,D`) để di chuyển Hero
- `draw()`: Phương thức này sẽ vẽ trò chơi.

- Ý tưởng: Sẽ hiển thị các đối tượng theo số (Hero: 1, Warrior: 2, Monster: 3, Construction: 4) và cho Hero di chuyển dựa trên các nút W,A,S,D, các sự kiện diễn ra khi người chơi di chuyển đến các ô tương ứng.
 - Chi tiết: Đầu tiên sẽ tạo mảng int 2 chiều và toàn bộ mảng là 0 (ô trống). Tiếp theo là gán số tương ứng của các đối tượng vào mảng (1, 2, 3, 4). Sau bước này ta sẽ có mảng int 2 chiều chứa biểu tượng tượng trưng cho từng đối tượng trên bản đồ. Sau đó là lấy di chuyển của người chơi. Tiếp là thu nhận lính nếu di chuyển đến vị trí lính, chiến đấu với quái vật nếu đến vị trí quái, nâng cấp lính nếu đến công trình. Cuối cùng là xóa map hiện tại để lập lại vòng lặp trên và vòng lặp sẽ kết thúc khi chiến binh hi sinh tất cả hoặc không còn quái trên bản đồ. Ngoài ra khi di chuyển 3 lần thì lính của người chơi thu nhận được và quái sẽ được tăng 1 cấp
 - over(): Phương thức in ra màn hình là người chơi đã chiến thắng hoặc thua
 - battle(): Toàn bộ lính bạn thu nhận được sẽ đánh với 1 quái vật. Cách thức khiêu chiến là 1 vs 1, người đầu tiên trong đội quân bạn sẽ ra nghênh chiến với quái vật, nếu máu bé hơn 0 thì sẽ đến người tiếp theo. Khiêu chiến sẽ kết thúc khi quái bị tiêu diệt hoặc đội quân người chơi không còn một ai.
 - drawMap(): Tương tự như phần đầu của phương thức draw(). Phương thức drawMap() dùng để vẽ lại những đối tượng đã được nhập trước đó cho người chơi dễ dàng quan sát.
7. Hàm main(): dùng để chạy game hoàn chỉnh. Đầu tiên tạo 1 đối tượng map, khởi tạo map khởi tạo các đối tượng liên quan

III. Khả năng mở rộng

1. Thiết kế lại mã nguồn

- Do thời gian có hạn nên em chưa kịp trình bày code một cách đẹp mắt nên nếu có thêm thời gian, em sẽ thêm 1 class Game để chứa những phương thức hoạt động của trò chơi trong đó, hàm main và class Map sẽ gọn và dễ hiểu hơn

2. Thêm chức năng save, load game và tính điểm

- Đọc ghi file các thuộc tính của các đối tượng hiện có khi trò chơi đang hoạt động, tính điểm dựa trên thời gian trò chơi diễn ra + độ lớn của bản đồ

3. Mở rộng mã nguồn

- Cho phép các đối tượng Warrior, Construction, Monster khởi tạo vị trí 1 cách ngẫu nhiên và các thuộc tính cũng ngẫu nhiên tùy vào độ khó
- Tạo thêm các lớp con của Warrior, Construction, Monster để thêm tính đa dạng
- Thiết kế đồ họa cho trò chơi
- Thiết kế những Item chức năng ở các ô ngẫu nhiên, khi đến các ô đó người chơi được nhận các thuộc tính buff hoặc debuff

- IV. Phần chưa hoàn thành
- Phần nâng cao trong đề bài