TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

ĐẠI HỌC QUỐC GIA TP.HCM

Lớp 20CLC08

BÁO CÁO ĐỒ ÁN

THỰC HÀNH CUỐI KỲ

GV hướng dẫn: *Nguyễn Hải Đăng – Đỗ Trọng Lễ*

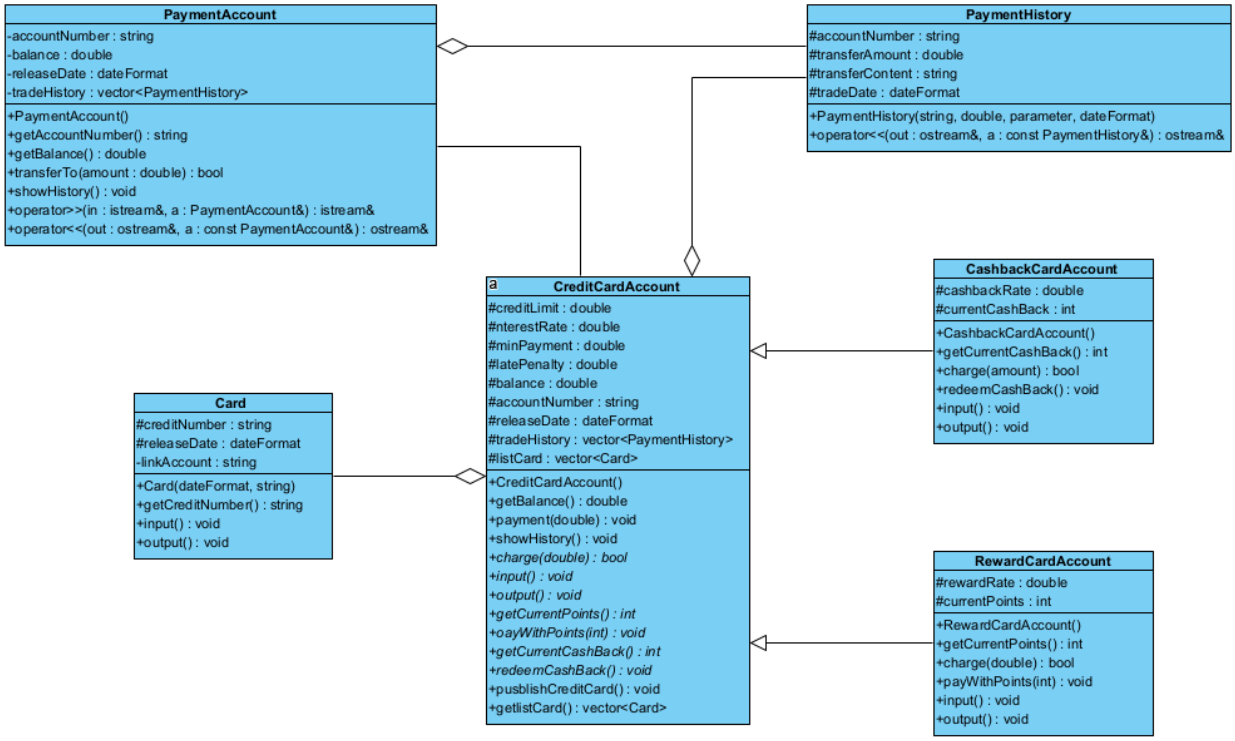
Sinh viên thực hiện: *20127039 – Trần Đàm Gia Huy*

Thành phố Hồ Chí Minh, ngày 25 tháng 12 năm 2021

MỤC LỤC

BÀI 1: TÀI KHOẢN NGÂN HÀNG

Các đối tượng:



- **PaymentAccount:** Tài khoản thanh toán, được sử dụng để thanh toán các tài khoản tín dụng
- **CreditCardAccount:** Tài khoản thẻ tín dụng, được sử dụng để tạo tài khoản tín dụng nhằm quản lý sự chi tiêu trước cho người dùng và phát hành thẻ tín dùng trực tiếp (Danh sách thẻ tín dụng được quản lý **bởi CreditCardAccount** nên có quan hệ sở hữu aggregation). Đồng thời **CreditCardAccount** cũng là lớp cha cho các lớp con **RewardCardAccount** và **CashbackCardAccount** kế thừa và phát triển
- **PaymentHistory:** chứa thông tin lịch sử giao dịch, mỗi tài khoản đều sở hữu danh sách các giao dịch.
- **Card:** Chứa thông cần thiết của 1 thẻ tín dụng
- **RewardCardAccount:** Tài khoản được kế thừa từ **CreditCardAccount** đồng thời có thêm chức năng tích điểm trên mỗi giao dịch
- **CashbackCardAccount:** Tài khoản được kế thừa từ **CreditCardAccount** đồng thời có thêm chức năng hoàn tiền cho người dùng trên mỗi giao dịch

Cách thức chương trình hoạt động:

- Đầu tiên người dùng sẽ đăng ký **tài khoản thanh toán** (Nhập từ bàn phím)
- Sau đó người dùng sẽ được lựa chọn đắng ký 1 trong 3 tài khoản (**Tài khoản thẻ tín dụng, Tài khoản thẻ tích điểm, Tài khoản thẻ hoàn tiền**)
  - Tài khoản thẻ tín dụng:
    - Đăng kí tài khoản và phát hành thẻ tín dụng theo số lượng mình mong muốn
    - Chọn thẻ tín dụng để sử dụng
    - Tiêu 1 số tiền để số dư trong thẻ tín dụng tăng lên. Trong bài này em sẽ cho tiêu 3 lần (1000000vnd, 3000000vnd, 5000000vnd). Tổng là 9000000vnd
    - Hiển thị lịch sử giao dịch của thẻ tín dụng đó (sao kê)
    - Thanh toán khoản tiền thẻ tín dụng bằng thẻ thanh toán
    - Hiển thị lịch sử giao dịch của thẻ thanh toán
  - Tài khoản thẻ tích điểm:
    - Đăng kí tài khoản và phát hành thẻ tích điểm theo số lượng mong muốn
    - Chọn thẻ tích điểm để sử dụng
    - Tiêu 1 số tiền để số dư và số điểm tích trong thẻ tích điểm tăng lên. Trong bài này em sẽ cho tiêu 3 lần (1000000vnd, 3000000vnd, 5000000vnd). Tổng là 9000000vnd
    - Sử dụng số điểm tích để thanh toán (Trong bài em ví dụ cụ thể là lấy số điểm tích hiện tại để thanh toán cho 100 points)
    - Hiển thị lịch sử giao dịch của thẻ tích điểm đó (sao kê)
    - Thanh toán khoản tiền thẻ tích điểm bằng thẻ thanh toán
    - Hiển thị lịch sử giao dịch của thẻ thanh toán
  - Tài khoản thẻ hoàn tiền:
    - Đăng kí tài khoản và phát hành thẻ hoàn tiền theo số lượng mong muốn
    - Chọn thẻ hoàn tiền để sử dụng
    - Tiêu 1 số tiền để số dư và số tiền hoàn trong thẻ hoàn tiền tăng lên. Trong bài này em sẽ cho tiêu 3 lần (1000000vnd, 3000000vnd, 5000000vnd). Tổng là 9000000vnd
    - Hiển thị số tiền có thể hoàn
    - Tiến hành hoàn tiền (Số tiền hoàn sẽ trừ vào số dư và số tiền hoàn bằng 0)
    - Hiển thị lịch sử giao dịch của thẻ hoàn tiền đó (sao kê)
    - Thanh toán khoản tiền thẻ hoàn tiền bằng thẻ thanh toán
    - Hiển thị lịch sử giao dịch của thẻ thanh toán

Demo minh họa

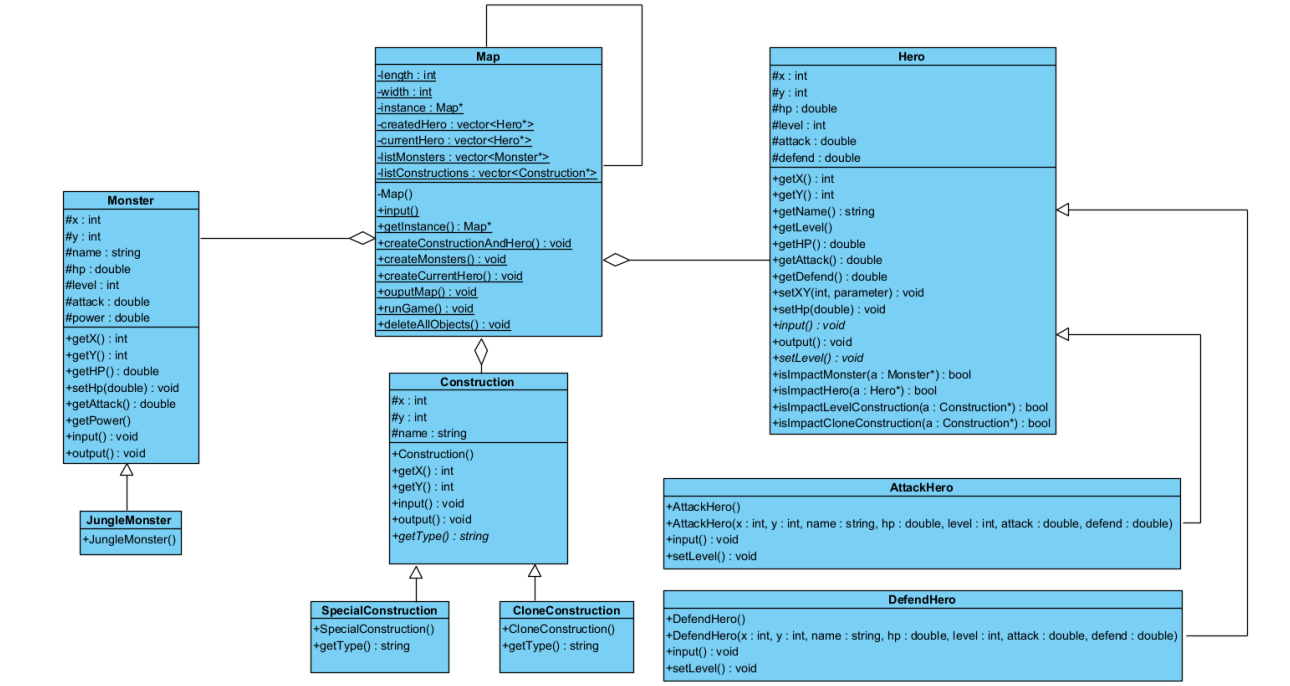
Tài khoản hoàn tiền và sử dụng Payment Account thanh toán

```
------------PAYMENT ACCOUNT-------------
Account number: GiaHuy
Balance: 100000000
Release date: 11/9/2002

-----------------MENU-------------------
1. Credit Card Account
2. Reward Card Account
3. Cash back Card Account
4. Exit
Input choose: 3

------------CASH BACK CARD ACCOUNT-------------
Input credit account number: 123456789
Input interest rate: 1.2
Input cash back rate: 0.5
Input day: 11
Input month: 09
Input year: 2002
Account number: 123456789
Balance: 0
Credit limit: 30000000
Interest rate: 1
Min payment: 0
Late penalty 2000000
Cash back rate: 1
Current cash back: 0
Release date: 11/9/2002

------------PUBLISH CASH BACK CARD-------------
-----INPUT-----
Input the number of cards to publish ( max 6 ): 2
Card 1
Input credit card number: giahuy1
Card 2
Input credit card number: giahuy2

-----OUTPUT-----
Card 1
Credit card number: giahuy1
Release date: 11/9/2002
Linked with ( Account number ): 123456789
Card 2
Credit card number: giahuy2
Release date: 11/9/2002
Linked with ( Account number ): 123456789

------------USE CASH BACK CARD-------------
Input cash back card to use: giahuy2

------------CASH BACK-------------
Cash back is available: 4500000
Redeem cash back
Account number: 123456789
Balance: 4500000
Credit limit: 30000000
Interest rate: 1
Min payment: 0
Late penalty 2000000
Cash back rate: 1
Current cash back: 0
Release date: 11/9/2002
```

```
------------HISTORY OF CASH BACK CARD-------------
Credit card number: giahuy2
Release date: 11/9/2002
Linked with ( Account number ): 123456789

------TRANSACTION 1--------
Account number: 123456789
Transfer amount: +1000000
Transfer content: Receive
Trade date: 11/9/2002

------TRANSACTION 2--------
Account number: 123456789
Transfer amount: +3000000
Transfer content: Receive
Trade date: 11/9/2002

------TRANSACTION 3--------
Account number: 123456789
Transfer amount: +5000000
Transfer content: Receive
Trade date: 11/9/2002


------------PAY CASH BACK CARD BY PAYMENT ACCOUNT------
-----PAYMENT ACCOUNT------
Account number: 123456789
Balance: 0
Credit limit: 30000000
Interest rate: 1
Min payment: 0
Late penalty 2000000
Cash back rate: 1
Current cash back: 0
Release date: 11/9/2002
Account number: GiaHuy
Balance: 95500000
Release date: 11/9/2002

-----HISTORY OF PAYMENT ACCOUNT------
------TRANSACTION 1--------
Account number: GiaHuy
Transfer amount: -4500000
Transfer content: Payment
Trade date: 11/9/2002

Press any key to continue . . .
```

BÀI 2: TRÒ CHƠI CHIẾN THUẬT ĐẠI CHIẾN HAI THẾ GIỚI

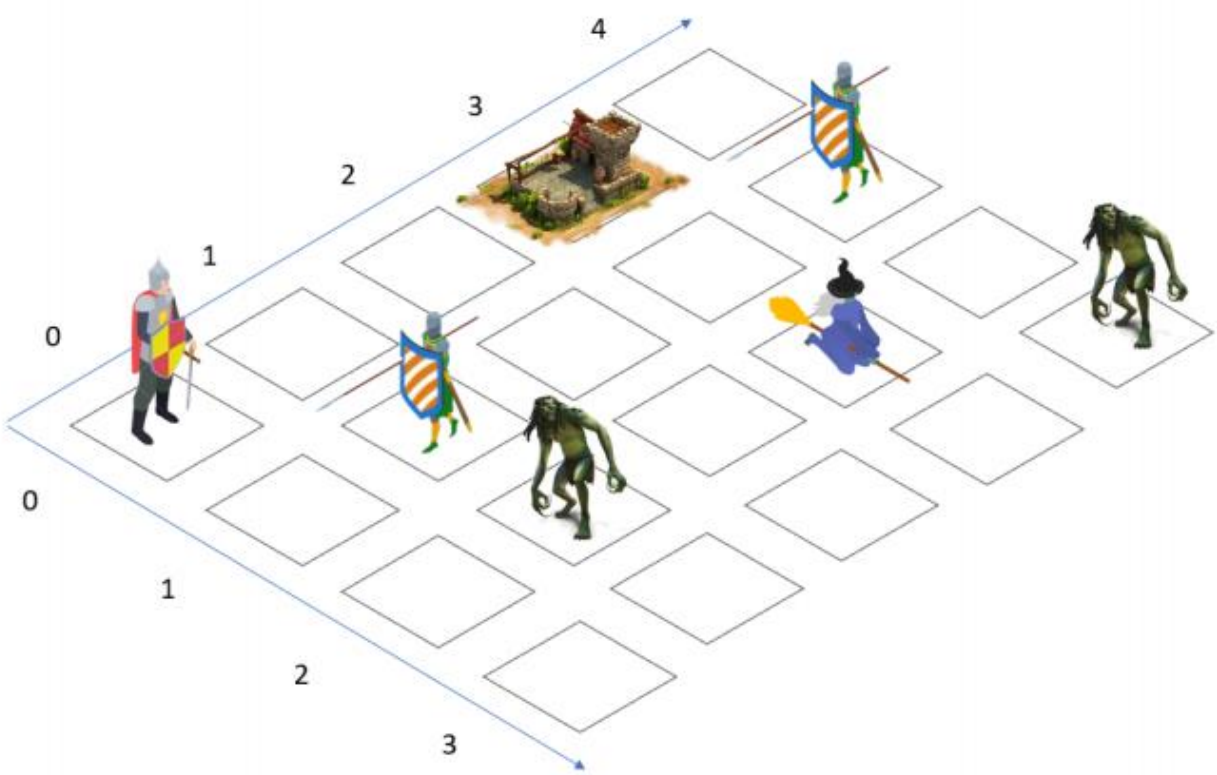
Các đối tượng



- **Hero:** Lớp hero có các thuộc tính riêng biệt như tọa độ, hp, level, chỉ số tấn công và phòng thủ. Lớp này hỗ trợ các phương thức nhập xuất thông tin hero và check các vị trí xem có đụng độ quái vật, các công trình hoặc các hero tại vị trí khác hay không.
- **Attack Hero:** Kế thừa lớp **Hero** và có chỉ số tấn công cao hơn phòng thủ (phòng thủ = tấn công /5). Khi lên level thì phòng thủ +1, tấn công +5.
- **Defend Hero:** Kế thừa lớp **Hero** và có chỉ số phòng thủ cao hơn tấn công (tấn công = phòng thủ /5). Khi lên level thì phòng thủ +5, tấn công +1.
- **Construction:** Lớp **Construction** cho biết các thuộc tính và phương thức của công trình. Một công trình có các thuộc tính tọa độ và tên gọi riêng. Ngoài ra còn có các phuơng hức get, set và lấy kiểu công trình hỗ trợ cho việc xử lý.
- **SpecialConstruction** : Kế thừa lớp **Construction** và dùng để nâng cấp **Hero**
- **CloneConstruction** : Kế thừa lớp **Construction** và dùng để nhân bản **Hero**
- **Monster**: Lớp **Monster** cho biết các thuộc tính và phương thức của quái vật. Một quái vật có các thuộc tính tọa độ, tên gọi riêng, level, chỉ số tấn công và điểm nộ. Ngoài ra còn có các phuơng thức get, set và nhập xuất hỗ trợ cho việc xử lý.
- **JungleMonster:** Kế thừa lớp **Monster**
- **Map**: Đóng vai trò điều phối toàn bộ trò chơi**. Map** có các thuộc tính gồm list hero được công trình tạo ra, list người chơi hiện tại, list quái vật và list công trình. Do đó tất cả các lớp trong trò chơi đều có quan hệ **aggregation** với **Map**

- Do toàn bộ trò chơi chỉ có **1 map duy nhất** nên áp dụng mẫu thiết kế hướng đối tượng **Singleton** là rất hợp lý. Điều đó giúp đảm bảo chỉ có **1 bản đồ**. Ngoài ra Map còn có các phương để khởi tạo game và phương thức runGame để chạy trò chơi. Cuối cùng là phương thức **deleteAllObjects** dưới dạng static vì nếu để trong **destructor** thì khi gọi **destructor** sẽ gặp tình trạng tự bản thân thành phần này gọi xóa chính nó, dẫn đến việc **lặp vô hạn**.

Kịch bản trò chơi



- Bản đồ sẽ có chiều dài chiều rộng là **length, width** do người dùng nhập.
- Người dùng nhập vị trí x,y **player** muốn di chuyển đến.
- Nếu va chạm với binh lính khác thì **player** thu nạp binh lính (**Hero**) đó vào đội của mình
- Nếu va chạm với công trình level thì tất cả **Hero** trong đội sẽ tăng 1 level đồng thời nếu là **Attack Hero** thì điểm tấn công +5, phòng thủ +1 và **Defend Hero** thì ngược lại. Sau đó công trình sẽ biến mất.
- Nếu va chạm với công trình clone thì tất cả số lượng **Hero** trong đội sẽ được nhân bản lên x2 lần, đồng thời công trình đó sẽ biến mất
- Nếu va chạm với quái rừng thì tất cả Hero và quái rừng sẽ tấn công nhau
- **Quái rừng** tấn công **Hero** bằng chỉ số tấn công của quái + điểm nộ x2
- **Hero** mất 1 lượng máu bằng chỉ số tấn công của quái + điểm nộ x2 – chỉ số phòng thủ **Hero**
- **Hero** tấn công **quái rừng** bằng chỉ số tấn công của **Hero**
- **Quái rừng** mất 1 lượng máu bằng chỉ số tấn công của **Hero**
- Trò chơi **chiến thắng** khi tất cả quái rừng bị tiêu diệt và ít nhất 1 Hero còn sống
- Trò chơi **thua** khi tất cả người chơi bị tiêu diệt

Cài đặt trò chơi

Cài đặt các lớp cần thiết cho trò chơi

Hero

```
//------------CLASS HERO------------
class Hero
{
protected:
	int x, y;
	string name;
	double hp;
	int level;
	double attack, defend; // atk status, def status
public:
	//Setter
	void setXY(int, int);
	void setHp(double);
	virtual void setLevel();

	//Getter
	int getX();
	int getY();
	int getLevel();
	string getName();
	double getHP();
	double getAttack();
	double getDefend();

	//Input, output
	virtual void input();
	void output();

	//Check impact
	bool isImpactMonster(Monster*); // Impact with monsters
	bool isImpactHero(Hero*); // Impact with other heroes
	bool isImpactLevelConstruction(Construction*); // Impact with level construction
	bool isImpactCloneConstruction(Construction*); // Impact with clone construction
};
```

```
//------------CLASS ATTACKHERO------------
class AttackHero : public Hero
{
public:
	AttackHero();
	AttackHero(int, int, string, double, int,
	double, double);
	void input();
	void setLevel();
};
```

```
//------------CLASS DEFENDHERO------------
class DefendHero : public Hero
{
public:
	DefendHero();
	AttackHero(int, int, string, double, int,
	double, double);
	void input();
	void setLevel();
};
```

Construction

```
//------------CLASS CONSTRUCTION------------
class Construction
{
protected:
	int x, y;
	string name;
public:
	Construction();

	//Getter
	int getX();
	int getY();

	//Input, Output
	void input();
	void output();

	//Use to distinguish between each other
	virtual string getType();
};
```

```
//-----CLASS SPECIALCONSTRUCTION----
class SpecialConstruction:public Construction
{
public:
	SpecialConstruction();
	string getType();
};
```

```
//-----CLASS CLONECONSTRUCTION-----
class CloneConstruction :public Construction
{
public:
	CloneConstruction();
	string getType();
};
```

Monster

```
class Monster
{
protected:
        int x, y;
        string name;
        double hp;
        int level;
        double attack; // atk status
        double power; // power status
public:
        //Getter
        int getX();
        int getY();
        double getHP();
        double getAttack();
        double getPower() ;

        //Setter
        void setHp(double);

        //Input, output
        void input();
        void output();
};
```

```
class JungleMonster :public Monster
{
public:
        JungleMonster();
};
```

Map

```
class Map // Use singleton to create only map
{
private:
	static int length, width;
	static Map* instance; //Only object
	static vector<Hero*> createdHero; // List hero created by Constructions
	static vector<Hero*> currentHero; // Your team
	static vector<Monster*> listMonsters; // List Monsters
	static vector<Construction*> listConstructions; // List Constructions
	Map(); // Private Constructor
public:
	void input();
	static Map* getInstance(); // Create only object
	static void createConstructionAndHero(); // Create Constructions and Heroes
	static void createMonsters(); // Create Monsters
	static void createCurrentHero(); // Create Player
	static void outputMap(); // Output all objects on map
	static void runGame(); // Run game
	static void deleteAllObjects(); // Free static memory
};
```

Khởi tạo toàn bộ đối tượng cho trò chơi

```
void Map::createConstructionAndHero() {

	//Create Constructions
	int n;
	cout << "--------------CONSTRUCTIONS AND HEROES--------------\n";
	cout << "Input the number of create hero constructions: "; cin >> n;
	for (int i = 0; i < n; i++) {
		listConstructions.push_back(new Construction());
		while (true)
		{
			cout << "\n---------CONSTRUCTIONS--------\n";
			listConstructions[i]->input();
			if (listConstructions[i]->getX() > length || listConstructions[i]->getX() < 0 ||
			listConstructions[i]->getY() > width || listConstructions[i]->getX() < 0)
				cout << "Position is out of range, Input again" << endl;
			else break;
		}
		//Create Hero
		int choose;
		cout << "\n---------HEROES--------\n";
		cout << "1. Attack hero" << endl;
		cout << "2. Defend hero" << endl;
		cout << "Input type of hero: "; cin >> choose; cin.ignore(1);
		if (choose == 1) {
			createdHero.push_back(new AttackHero());
			while (true) {
			createdHero[i]->input();
			if (createdHero[i]->getX() > length || createdHero[i]->getX() < 0 ||
			createdHero[i]->getY() > width || createdHero[i]->getX() < 0)
				cout << "Position is out of range, Input again" << endl;
			else break;
			}
		}
		else {
			createdHero.push_back(new DefendHero());
			while (true) {
				createdHero[i]->input();
				if (createdHero[i]->getX() > length || createdHero[i]->getX() < 0 ||
				createdHero[i]->getY() > width || createdHero[i]->getX() < 0)
					cout << "Position is out of range, Input again" << endl;
				else break;
			}
		}
	}

	//Create Level Constructions
	int m;
	cout << "\nInput the number of leveling constructions: "; cin >> m;
	for (int i = n; i < n + m; i++) {
		cout << "\n---------CONSTRUCTIONS--------\n";
		listConstructions.push_back(new SpecialConstruction());
		while (true) {
			listConstructions[i]->input();
			if (listConstructions[i]->getX() > length || listConstructions[i]->getX() < 0 ||
			listConstructions[i]->getY() > width || listConstructions[i]->getX() < 0)
				cout << "Position is out of range, Input again" << endl;
			else break;
		}
	}

	//Create Clone Constructions
	int z;
	cout << "\nInput the number of clone constructions: "; cin >> z;
	for (int i = n + m; i < n + m + z; i++) {
		cout << "\n---------CONSTRUCTIONS--------\n";
		listConstructions.push_back(new CloneConstruction());
		while (true) {
			listConstructions[i]->input();
			if (listConstructions[i]->getX() > length || listConstructions[i]->getX() < 0 ||
			listConstructions[i]->getY() > width || listConstructions[i]->getX() < 0)
				cout << "Position is out of range, Input again" << endl;
			else break;
		}
	}
}
```

(Tất cả tọa độ khi nhập đều không được trùng nhau)

Khởi tạo danh sách các công trình

- Đầu tiên ta sẽ khởi tạo danh sách công trình tạo ra **Hero** và khởi tạo ra **Hero**

- Sau đó ta khởi tạo danh sách công trình tăng level cho **Hero**

- Cuối cùng là tạo danh sách công trình nhân bản **Hero.** Ta chỉ cần **1 vector<Construction*> listConstructions** để lưu 3 loại công trình gồm **Construction, SpecialConstruction** và **CloneConstructio**n do áp dụng **tính kế thừa** cho **SpecialConstruction** và **CloneConstruction**. Nếu không ta phải cần **3 vector** để lưu công trình.

Khởi tạo danh sách quái vật

```
// Create Monsters
void Map::createMonsters() {
	cout << "\n--------------MONSTERS---------------\n";
	int n;
	cout << "Input the number of monsters: "; cin >> n; cin.ignore();
	for (int i = 0; i < n; i++) {
		cout << "-----MONSTERS-----\n";
		listMonsters.push_back(new JungleMonster());
		while (true) {
			listMonsters[i]->input();
			if (listMonsters[i]->getX() > length || listMonsters[i]->getX() < 0
			|| listMonsters[i]->getY() > width || listMonsters[i]->getX() < 0)
				cout << "Position is out of range, Input again" << endl;
			else break;
		}
	}
}
```

Tạo danh sách **quái vật** tại những vị trí người dùng qui định

Khởi tạo người chơi

```
// Create Player
void Map::createCurrentHero() {
	cout << "\n--------------PLAYER---------------\n";
	currentHero.push_back(new AttackHero());
	while (true) {
		currentHero[0]->input();
		if (currentHero[0]->getX() > length || currentHero[0]->getX() < 0
		|| currentHero[0]->getY() > width || currentHero[0]->getX() < 0)
			cout << "Position is out of range, Input again" << endl;
		else break;
	}
}
```

Tạo danh sách **Hero** của team hiện tại với player là **Hero** vị trí đầu tiên. Do sử dụng **tính kế thừa Hero** cho 2 lớp **AttackHero** và **DefendHero** nên chỉ cần **1 vector<Hero*> currentHero** để lưu danh sách Hero thay vì phải sử dụng **2 vector vector<AttackHero> currentHero** và **vector<DefendHero> currentHero.**

Chạy trò chơi

Di chuyển

```
//Move your team
	//Move your team
	int x, y;
	while (true) {
		cout << "Input position your team want to move" << endl;
		cout << "Input x: "; cin >> x;
		cout << "Input y: "; cin >> y;
		if (x < 0 || y < 0 || x>length || y>width) continue;
		else {
			for(int i=0; i<currentHero.size(); i++)
				currentHero[i]->setXY(x, y);
			break;
		}
	}
```

Nhập vị trí player muốn đến (**Hero** đầu tiên) và set vị trí đó cho toàn bộ **Hero** trong team

Va chạm với Hero

```
//Heroes
for (int i = 0; i < createdHero.size(); i++)
{
	if (currentHero[0]->isImpactHero(createdHero[i])) //Check impact with other heroes
	{
		cout << "\n--------NEW HERO ENGAGES IN YOUR TEAM--------\n";
		currentHero.push_back(createdHero[i]); //Add hero to your team
		//Delete hero in list hero created by construction
		createdHero.erase(createdHero.begin() + i);
	}
}
```

Nếu player va chạm với **Hero** khác thì thu nhận Hero đó vào team. Đồng thời xóa **Hero** đó ra khỏi danh sách **Hero** được tạo bởi công trình

Va chạm với công trình

```
	//Construction
	for (int i = 0; i < listConstructions.size(); i++)
	{
		//Level construction
		//Check impact with level construction
		if (currentHero[0]->isImpactLevelConstruction(listConstructions[i]))
		{
			cout << "\n----------YOUR TEAM HAS BEEN LEVEL UP----------\n";
			for (int j = 0; j < currentHero.size(); j++)
				currentHero[j]->setLevel(); //Increase level of heroes
			//Delete construction has been used
			listConstructions.erase(listConstructions.begin() + i);
		}

		//Clone construction
		//Check impact with clone construction
		if (currentHero[0]->isImpactCloneConstruction(listConstructions[i]))
		{
			cout << "\n----------YOUR TEAM HAS BEEN CLONE (x2)----------\n";
			int n = currentHero.size();
			for (int j = 0; j < n; j++) {
				//Clone heroes
				if (currentHero[i]->getAttack() > currentHero[i]->getDefend())
currentHero.push_back(new AttackHero(currentHero[i]->getX(), currentHero[i]->getY(),
currentHero[i]->getName(), currentHero[i]->getHP(), currentHero[i]->getLevel(), currentHero[i]-
>getAttack(), currentHero[i]->getDefend()));
				else currentHero.push_back(new DefendHero(currentHero[i]->getX(),
currentHero[i]->getY(), currentHero[i]->getName(), currentHero[i]->getHP(), currentHero[i]-
>getLevel(), currentHero[i]->getAttack(), currentHero[i]->getDefend()));
			}
			//Delete construction has been used
			listConstructions.erase(listConstructions.begin() + i);
			break;
		}
	}
```

Nếu va chạm với công trình level thì tất cả **Hero** trong đội sẽ tăng 1 level đồng thời nếu là **Attack Hero** thì điểm tấn công +5, phòng thủ +1 và **Defend Hero** thì ngược lại. Sau đó công trình sẽ biến mất.

Nếu va chạm với công trình clone thì tất cả số lượng Hero trong đội sẽ được nhân bản lên x2 lần, đồng thời công trình đó sẽ biến mất. Khi **clone** thì em sẽ sao chép **giá trị**, không sao chép **địa chỉ** vì như vậy sẽ bị **trỏ chung vùng nhớ**, do đó khi clone thì em sẽ **push_back(new….)** thay vì **push_back(…).**

Va chạm với quái vật

```
//Monster
for (int i = 0; i < listMonsters.size(); i++)
{
	if (currentHero[0]->isImpactMonster(listMonsters[i])) //Check impact
	{
		cout << "\n----ATTACKING BETWWEEN MONSTER AND HUMAN----\n";
		for (int j = 0; j < currentHero.size(); j++)
		{
			//Check if hp's and defend status heroes greater than damage of monsters
			if (listMonsters[i]->getAttack() + listMonsters[i]->getPower() * 2 -
				currentHero[j]->getDefend() > currentHero[j]->getHP()) {
				//if not, decrease hp's heroes
				currentHero[j]->setHp(listMonsters[i]->getAttack() +
				listMonsters[i]->getPower() * 2 - currentHero[j]->getDefend());
			}
			listMonsters[i]->setHp(currentHero[j]->getAttack());
			//Check each hero in your team is dead or not
			if (currentHero[j]->getHP() <= 0) {
				// If dead delete hero from your team
				currentHero.erase(currentHero.begin() + j
				j--;
			}
		}
		//Check each monster is dead or not
		if (listMonsters[i]->getHP() <= 0) {
			// If dead delete monster from list monster
			listMonsters.erase(listMonsters.begin() + i);
			i--;
		}
	}
}
```

- Nếu va chạm với **quái rừng** thì tất cả **Hero** và **quái rừng** sẽ tấn công nhau
- **Quái rừng** tấn công **Hero** bằng chỉ số tấn công của quái + điểm nộ x2
- **Hero** mất 1 lượng máu bằng chỉ số tấn công của quái + điểm nộ x2 – chỉ số phòng thủ **Hero**
- **Hero** tấn công **quái rừng** bằng chỉ số tấn công của **Hero**
- **Quái rừng** mất 1 lượng máu bằng chỉ số tấn công của **Hero**

Xuất danh sách tất cả đối tượng trên map

```
//Output all objects on map
void Map::outputMap() {

	//Output constructions
	cout << "\n--------------CONSTRUCTIONS---------------\n";
	for (int i = 0; i < listConstructions.size(); i++) {
		cout << "-----CONSTRUCTION-----\n";
		listConstructions[i]->output();
		cout << "Type: " << listConstructions[i]->getType() << endl;
	}

	//Output Monsters
	cout << "\n----------------MONSTERS----------------\n";
	for (int i = 0; i < listMonsters.size(); i++) {
		cout << "-----MONSTER-----\n";
		listMonsters[i]->output();
		cout << endl;
	}

	//Output heroes created by constructions
	cout << "\n----------------HEROES----------------\n";
	for (int i = 0; i < createdHero.size(); i++) {
		cout << "-----HERO-----\n";
		createdHero[i]->output();
		cout << endl;
	}

	//Output your team
	cout << "\n----------------YOUR TEAM----------------\n";
	for (int i = 0; i < currentHero.size(); i++) {
		cout << "-----HERO-----\n";
		currentHero[i]->output();
		cout << endl;
	}
	//Win or Lose
	if (listMonsters.size() == 0 && currentHero.size() != 0)
		cout << "\n----------------HUMAN WIN--------------------\n";
	if ( currentHero.size() == 0)
		cout << "\n----------------HUMAN LOSE--------------------\n";

}
```

- Trò chơi chiến thắng khi tất cả quái rừng bị tiêu diệt và ít nhất 1 **Hero** còn sống
- Trò chơi thua khi tất cả người chơi bị tiêu diệt

Giải phóng bộ nhớ

```
//Free memory
void Map::deleteAllObjects(){
	//Free constructions
	for (int i = 0; i < listConstructions.size(); i++) {
		delete listConstructions[i];
		listConstructions[i] = nullptr;
	}
	//Free monsters
	for (int i = 0; i < listMonsters.size(); i++) {
		delete listMonsters[i];
		listMonsters[i] = nullptr;
	}
	//Free heroes created by constructions
	for (int i = 0; i < createdHero.size(); i++) {
		delete createdHero[i];
		createdHero[i] = nullptr;
	}
	//Free all heroes in team
	for (int i = 0; i < currentHero.size(); i++) {
		delete currentHero[i];
		currentHero[i] = nullptr;
	}
}
```

Main

```
#include "Map.h"
int main() {

	Map* a = Map::getInstance();

	cout << "INPUT MAP" << endl;
	a->input();
	//Initialize the game
	a->createConstructionAndHero();
	a->createMonsters();
	a->createCurrentHero();
	a->outputMap();

	//Play game
	while (true)
	{
		int choose;
		cout << "\n----------MENU----------\n";
		cout << "1. Move" << endl;
		cout << "2. Exit" << endl;
		cout << "Input choose: "; cin >> choose;
		if (choose == 1) {
			a->runGame();
			a->outputMap();
		}
		else {
			a->deleteAllObjects();
			delete a;
			a = nullptr;
			return 0;
		}
	}
}
```

Nhận xét

Những phần đã làm được

- Trong toàn bộ đồ án em đã thiết kế theo đúng tinh thần hướng đối tượng
- Thiết kế đúng theo kịch bản đề bài và có những phần sáng tạo riêng
- Áp dụng **kỹ thuật kế thừa, đa hình** và mẫu thiết kế **Singleton** đảm bao chỉ tạo ra 1 đối tượng **Map** duy nhất trong toàn bộ chương trình
- Tất cả **bộ nhớ cấp phát** đều được giải phóng
- Hàm **main** tổ chức gọn gàng
- Mã nguồn có chú thích đầy đủ
- **UML** rõ ràng, thể hiện rõ quan hệ giữa các lớp
- Thiết kế công trình nhân bản **Hero** trong phần yêu cầu nâng cao

Những phần chưa làm được

- Ban đầu khi khởi tạo các đối tượng trong trò chơi, người dùng sẽ mặc định là khởi tạo các công trình, **Hero** được tạo bởi công trình, quái vật tại các **vị trí khác nhau**. Em chưa kiểm tra được nếu người dùng khởi tạo đối tường trùng vị trí đã chọn trước đó
- Chưa cài đặt được nhân vật **quái vật phù thủy** trong phần yêu cầu nâng cao. Tuy nhiên em xin đóng góp 1 số ý tượng như sau:
    - Để **quái vật phù thủy** tạo ra quái vật rừng rậm và **quái vật phù thủy** khác thì mẫu thiết kế phù hợp nhất sẽ là **Composite**
    - Quái vật phù thủy sẽ gọi **đệ qui** hàm **input** và **output** chính nó để tạo ra nó hoặc tạo ra các **quái vật rừng rậm** bằng phương thức của **quái vật rừng rậm**
    - Sử dụng **vòng lặp** trong list **quái vật rừng rậm** để tính tổng lượng máu và sức mạnh của tất cả quái vật và tiếp tục gọi đệ qui nếu có **phù thủy** quản lý quái vật, từ đó gán cho **phù thủy**

Đoạn mã nguồn tâm đắc

```
//Monster
for (int i = 0; i < listMonsters.size(); i++)
{
	if (currentHero[0]->isImpactMonster(listMonsters[i])) //Check impact
	{
		cout << "\n----ATTACKING BETWWEEN MONSTER AND HUMAN----\n";
		for (int j = 0; j < currentHero.size(); j++)
		{
			//Check if hp's and defend status heroes greater than damage of monsters
			if (listMonsters[i]->getAttack() + listMonsters[i]->getPower() * 2 -
				currentHero[j]->getDefend() > currentHero[j]->getHP()) {
				//if not, decrease hp's heroes
				currentHero[j]->setHp(listMonsters[i]->getAttack() +
				listMonsters[i]->getPower() * 2 - currentHero[j]->getDefend());
			}
			listMonsters[i]->setHp(currentHero[j]->getAttack());
			//Check each hero in your team is dead or not
			if (currentHero[j]->getHP() <= 0) {
				// If dead delete hero from your team
				currentHero.erase(currentHero.begin() + j
				j--;
			}
		}
		//Check each monster is dead or not
		if (listMonsters[i]->getHP() <= 0) {
			// If dead delete monster from list monster
			listMonsters.erase(listMonsters.begin() + i);
			i--;
		}
	}
}
```

Đây chính là mã nguồn chiến đấu giữa quân đoàn **Hero** và những **Quái vật** mà em tâm đắc nhất. Khi tấn công cả 2 phe đều bị trừ 1 lượng HP, tuy nhiên nếu sự tấn công của quái vật quá thấp (thấp hơn chỉ số phòng thủ của **Hero** thì **Hero** sẽ **không bị mất máu**). Khi máu của quái vật bé hơn hoặc bằng 0 thì quái vật đó sẽ chết và xóa khỏi list quái vật trên bản đồ. Tương tự với **Hero** trong team. Đây có thể là đoạn code quan trọng và đóng vai trò một trong những linh hồn của trò chơi.

Demo minh họa

```
--------------CONSTRUCTIONS---------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create
-----CONSTRUCTION-----
Position: (1,3)
Name: Construction to level
Type: Level
-----CONSTRUCTION-----
Position: (1,4)
Name: Construction to clone
Type: Clone

----------------MONSTERS-----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------
-----HERO-----
Position: (1,2)
Name: Hero A
HP: 100
Level: 10
Attack / Defend: 20/100


----------------YOUR TEAM-----------------
-----HERO-----
Position: (2,1)
Name: Gia Huy
HP: 1000
Level: 7
Attack / Defend: 100/20


----------MENU------------
1. Move
2. Exit
Input choose: 1
Input position your team want to move
Input x: 1
Input y: 2
```

```
------------NEW HERO ENGAGES IN YOUR TEAM------

--------------CONSTRUCTIONS---------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create
-----CONSTRUCTION-----
Position: (1,3)
Name: Construction to level
Type: Level
-----CONSTRUCTION-----
Position: (1,4)
Name: Construction to clone
Type: Clone

----------------MONSTERS-----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM-----------------
-----HERO-----
Position: (1,2)
Name: Gia Huy
HP: 1000
Level: 7
Attack / Defend: 100/20

-----HERO-----
Position: (1,2)
Name: Hero A
HP: 100
Level: 10
Attack / Defend: 20/100


----------MENU------------
1. Move
2. Exit
Input choose:
```

Khi move vị trí **Hero Gia Huy** đến vị trí **Hero khác** (cụ thể là **Hero A** có tọa độ (1,2)) thì **Hero Gia Huy** sẽ thu nhận **Hero A** vào team. Lúc này team đã trở thành **2** thành viên và **Hero A** trong **list Hero** do công trình tạo ra sẽ bị xóa đi.

```
--------------CONSTRUCTIONS--------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create
-----CONSTRUCTION-----
Position: (1,3)
Name: Construction to level
Type: Level
-----CONSTRUCTION-----
Position: (1,4)
Name: Construction to clone
Type: Clone

----------------MONSTERS----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM----------------
-----HERO-----
Position: (1,2)
Name: Gia Huy
HP: 1000
Level: 7
Attack / Defend: 100/20

-----HERO-----
Position: (1,2)
Name: Hero A
HP: 100
Level: 10
Attack / Defend: 20/100


----------MENU------------
1. Move
2. Exit
Input choose: 1
Input position your team want to move
Input x: 1
Input y: 3
```

```
------------YOUR TEAM HAS BEEN LEVEL UP----

--------------CONSTRUCTIONS---------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create
-----CONSTRUCTION-----
Position: (1,4)
Name: Construction to clone
Type: Clone

----------------MONSTERS-----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM-----------------
-----HERO-----
Position: (1,3)
Name: Gia Huy
HP: 1000
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,3)
Name: Hero A
HP: 100
Level: 11
Attack / Defend: 21/105


----------MENU------------
1. Move
2. Exit
Input choose:
```

Sau khi move team đến vị trí công trình level thì toàn bộ **Hero** trong team sẽ tăng 1 level, chỉ số **attack** và **defend** của mỗi loại **Attack Hero** và **Defend Hero** trong team sẽ tăng theo. Đồng thời công trình level sẽ biến mất và xóa khỏi danh sách list công trình.

```
--------------CONSTRUCTIONS------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create
-----CONSTRUCTION-----
Position: (1,4)
Name: Construction to clone
Type: Clone

----------------MONSTERS---------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM--------------
-----HERO-----
Position: (1,3)
Name: Gia Huy
HP: 1000
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,3)
Name: Hero A
HP: 100
Level: 11
Attack / Defend: 21/105


----------MENU------------
1. Move
2. Exit
Input choose: 1
Input position your team want to move
Input x: 1
Input y: 4
```

```
-----------YOUR TEAM HAS BEEN CLONE (x2)--

--------------CONSTRUCTIONS---------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create

----------------MONSTERS-----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM-----------------
-----HERO-----
Position: (1,4)
Name: Gia Huy
HP: 1000
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,4)
Name: Hero A
HP: 100
Level: 11
Attack / Defend: 21/105

-----HERO-----
Position: (1,4)
Name: Gia Huy
HP: 1000
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,4)
Name: Hero A
HP: 100
Level: 11
Attack / Defend: 21/105


----------MENU------------
1. Move
2. Exit
Input choose:
```

Sau khi team move đến vị trí công trình clone thì công trình sẽ tiến hành nhân bản toàn bộ **Hero** trong team **x2 lần.**

```
--------------CONSTRUCTIONS--------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create

----------------MONSTERS-----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 1000
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM----------------
-----HERO-----
Position: (1,4)
Name: Gia Huy
HP: 1000
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,4)
Name: Hero A
HP: 100
Level: 11
Attack / Defend: 21/105

-----HERO-----
Position: (1,4)
Name: Gia Huy
HP: 1000
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,4)
Name: Hero A
HP: 100
Level: 11
Attack / Defend: 21/105


----------MENU------------
1. Move
2. Exit
Input choose: 1
Input position your team want to move
Input x: 1
Input y: 5
```

```
------ATTACKING BETWWEEN MONSTER AND HUMAN------

--------------CONSTRUCTIONS--------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create

----------------MONSTERS-----------------
-----MONSTER-----
Position: (1,5)
Name: Monster A
HP: 832
Level: 10
Attack / power: 100/5


----------------HEROES-----------------

----------------YOUR TEAM----------------
-----HERO-----
Position: (1,5)
Name: Gia Huy
HP: 911
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,5)
Name: Hero A
HP: 95
Level: 11
Attack / Defend: 21/105

-----HERO-----
Position: (1,5)
Name: Hero A
HP: 95
Level: 11
Attack / Defend: 21/105

-----HERO-----
Position: (1,5)
Name: Hero A
HP: 95
Level: 11
Attack / Defend: 21/105


----------MENU------------
1. Move
2. Exit
Input choose:
```

Ta move **team** đến vị trí quái thì cả 2 phe **tấn công lẫn nhau**

Quái rừng tấn công Hero bằng chỉ số tấn công của quái + điểm nộ x2. **Hero** mất 1 lượng máu bằng chỉ số tấn công của **quái rừng** + điểm nộ x2 – chỉ số phòng thủ **Hero.**

Hero tấn công quái rừng bằng chỉ số tấn công của **Hero**. **Quái rừng** mất 1 lượng máu bằng chỉ số tấn công của **Hero**

```
------ATTACKING BETWWEEN MONSTER AND HUMAN------

--------------CONSTRUCTIONS---------------
-----CONSTRUCTION-----
Position: (1,1)
Name: Construction to create
Type: Create

----------------MONSTERS-----------------

----------------HEROES-----------------

----------------YOUR TEAM-----------------
-----HERO-----
Position: (1,5)
Name: Gia Huy
HP: 466
Level: 8
Attack / Defend: 105/21

-----HERO-----
Position: (1,5)
Name: Hero A
HP: 70
Level: 11
Attack / Defend: 21/105

-----HERO-----
Position: (1,5)
Name: Hero A
HP: 70
Level: 11
Attack / Defend: 21/105

-----HERO-----
Position: (1,5)
Name: Hero A
HP: 70
Level: 11
Attack / Defend: 21/105


-----------------HUMAN WIN--------------------

----------MENU------------
1. Move
2. Exit
Input choose:
```

Sau khi cho cả 2 phe chiến đấu n lần với nhau đến khi quái vật chết thì trò chơi sẽ chiến thắng thuộc về phe **con người**. **Quái vật** chết sẽ bị xóa khỏi danh sách.