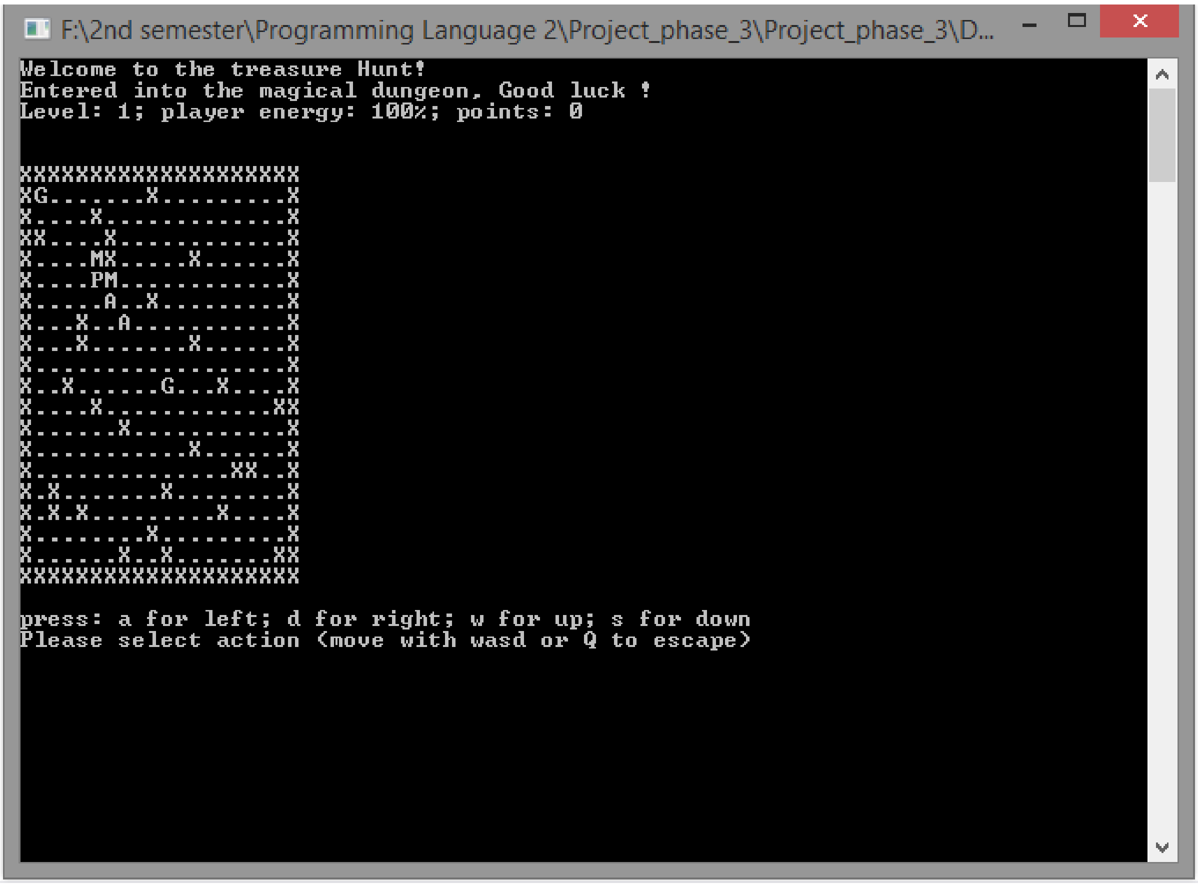
Summary of the Game:

1) This project was planned and implemented by me in about 1 month.
2) At the beginning, cerate 2 dimension array map (width: 20, Height: 20), player character with position value1 and position value2, sign, point and energy.
3) I have placed walls (**X**) on all around its borders. Then I also placed some extra walls inside to complete it. I also placed Gems denoted by '**G**'; Monster denoted by '**M**' and player denoted by '**P**'; MagicApple by 'A'.

4) When you enter on the game, it would be like as this picture:



F:\2nd semester\Programming Language 2\Project_phase_3\Project_phase_3\D...

```
Welcome to the treasure Hunt!
Entered into the magical dungeon, Good luck !
Level: 1; player energy: 100%; points: 0


XXXXXXXXXXXXXXXXXXXX
XG.......X.........X
X....X.............X
XX....X............X
X....MX.....X......X
X....PM............X
X.....A..X.........X
X...X..A...........X
X...X.......X......X
X..................X
X..X......G...X....X
X....X............XX
X......X...........X
X...........X......X
X..............XX..X
X.X.......X........X
X.X.X.........X....X
X........X.........X
X......X..X.......XX
XXXXXXXXXXXXXXXXXXXX

press: a for left; d for right; w for up; s for down
Please select action (move with wasd or Q to escape)
```

In here,

- A is Magic Apple; it will help you increase your energy.
- M is Monster; it will decrease your energy.
- G is Gem; the more gem you have the more point for you.
- P is player; it is you on the map.

5) After step 2, game is ready for playing, now function move() is running, depend on input of player, player character will move left, right, down , up by

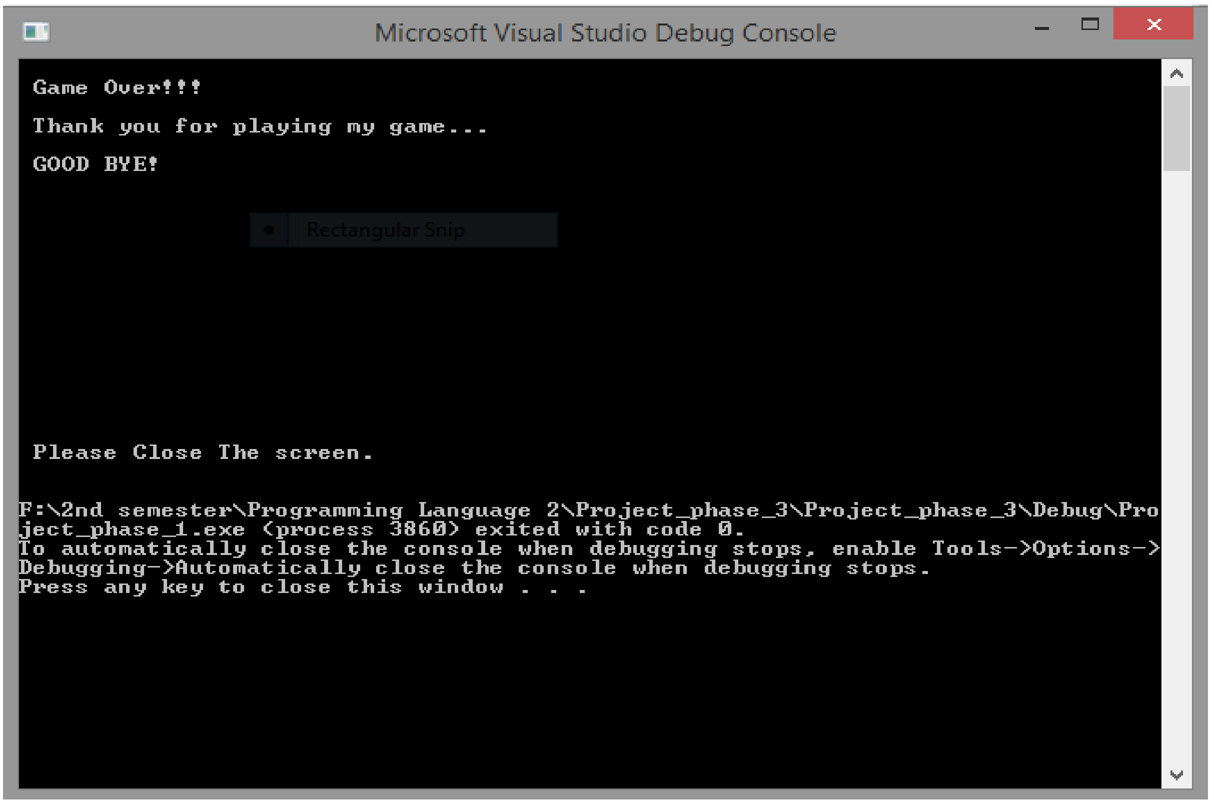
using switch cases. In switch case, there are 6 cases, move up, move down, move left, move right, invalid input and quit game.

6) In each movement case, there are some scenarios that player 'P' will meet monster, apple, gem. Each time player character moves, player will lose 5 energy, meet monster and fight will cost 5 energy, eat apple gain 10 energy, find gem will gain 10 points. Every time player move, program will set player character's symbol to new position, set old position to dot '.' ; then move new position.

In here,

- Press a for move left
- Press s for move down
- Press d for move right
- Press w for move up
- Press q for quit game
- Game Over when your energy = 0
- there will be more gem, magic apple, monster and bigger map when you go to higher level

7) At the end of the game, it would be like as:



```
 Game Over!!!

 Thank you for playing my game...

 GOOD BYE!





 Please Close The screen.

F:\2nd semester\Programming Language 2\Project_phase_3\Project_phase_3\Debug\Pro
ject_phase_1.exe (process 3860) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Classes and Their members:

In this phase I developed seven classes: Game, Gem, position, player, MagicApple, Monster, map.

```
class position  //class Position
{
private:

public:
       int value1, value2;
       position();
       position(int m, int n);
       int getPositionX() const;
       int getPositionY() const;
};

class Game    //class Game
{
private:
       char sign;
       position p;

public:
       Game();
       Game(int m, int n, char sign);
       position getPosition();
       virtual void setPosition(int m, int n);
       virtual char getSympol() const; //virtual ~Game();
};

class Gem :public Game  //class Gem
{
private:
       int point;

public:
       Gem(int bbb, int aaa, char sign, int point);
       Gem();
       int getPoint() const;       //void setPosition(int x, int y);
                               //position getPosition(
};

class MagicApple :public Game   //class MagicApple
{
private:
       int energy;

public:
       MagicApple(int bbb, int aaa, char sign, int energy);
       MagicApple();
       int getEnergy() const;
};


class Monster :public Game  //class Monster
{
private:
       int strength;

public:
       Monster(int bbb, int aaa, char sign, int strength);
       Monster();
       int getStrength() const;
};

class Player :public Game  // class Player
{
private:

public:
       int point, energy;
       Player(int bbb, int aaa, char sign, int point, int energy);
       Player();        //void setPosition(int m, int n);
                      //int getPositionX();
       int getPoints() const;
       int getEnergy() const; //int getEnergy() const;

};

class map :public Game, position  //class Map
{
private:
       static int level;

public:
       int h, w; // height(h) , width(w)
       char** letter;
       map(int w, int h); // height(h) , width(w)
       map(); //void random();
       void step_player(Monster* m, int shape1, Player& p, MagicApple* a, int shape2,
Gem* g, int shape3);
       int getHeight() const;
       int getWidth() const;
       void mapSet(/*Monster &m,Player &p*/);
       void print(Player& p);
       int getLevel() const;
       ~map();
};
```

I can judge myself as having done this task good and I also learned a lot through this project.