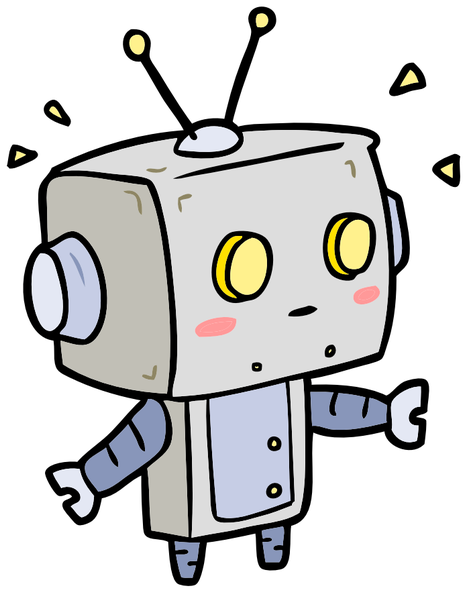
# Energy War

C++ Project

# ที่มาและความสำคัญ



โลกในปัจจุบันกำลังพังทลายด้วยน้ำมือของมนุษย์ หุ่นยนต์ในปี 2099 จึงย้อนเวลากลับมาล้างบางเหล่ามนุษย์เพื่อไม่ให้โลกถูกทำลาย

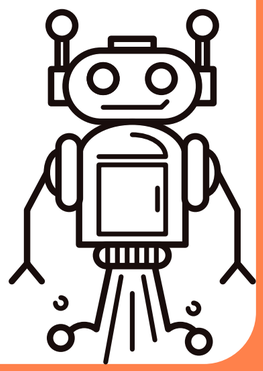
โดยจากที่กล่าวมา จึงเป็นสาเหตุและแนวคิดหลักของเกมที่ทำให้สมาชิกในกลุ่มตัดสินใจทำเกมแนว Turn base ออกมา เนื่องจากมีรูปแบบของเกมที่เล่นได้ง่าย และมีความน่าสนใจ อีกทั้งเนื่องจากแนวเกมดังกล่าว สามารถทำให้มีความหลากหลายของโครงสร้างการเขียนโปรแกรมได้ นี้จึงเป็นที่มาและความสำคัญของโครงงานดังกล่าว

# • รายละเอียด

- เกมต่อสู้ Turn-base ที่ต้องเล่นเป็นหุ่นยนต์ต่อสู้กับมนุษย์
- ทุกการกระทำของหุ่นยนต์จะต้องใช้พลังงาน บังเอิญตรงกับธีม ENERGY พอดีเลย
- มีด่านทั้งหมด 5 ด่าน แต่ละด่านก็จะมีจำนวนศัตรูและจำนวนระรอกแตกต่างกัน
- การต่อสู้ในแต่ละระรอกจะถูกแบ่งเป็นเทิร์น
- หุ่นยนต์มีทั้งหมด 3 ชนิด แต่ละชนิดจะมีค่าสถานะที่แตกต่างกัน สามารถทำการโจมตีได้เหมือนกันแต่จะมีทักษะพิเศษแตกต่างกัน
- ศัตรูจะมีทั้งหมด 4 ชนิด แต่ละชนิดมีความแตกต่างกันที่รูปร่างหน้าตาและค่าสถานะพลัง
- ค่าสถานะมีทั้งหมด 4 อย่างคือ HP(พลังชีวิต) DEF(พลังป้องกัน) ATK(พลังโจมตี) และ EN(พลังงาน)

# ข้อจำกัด

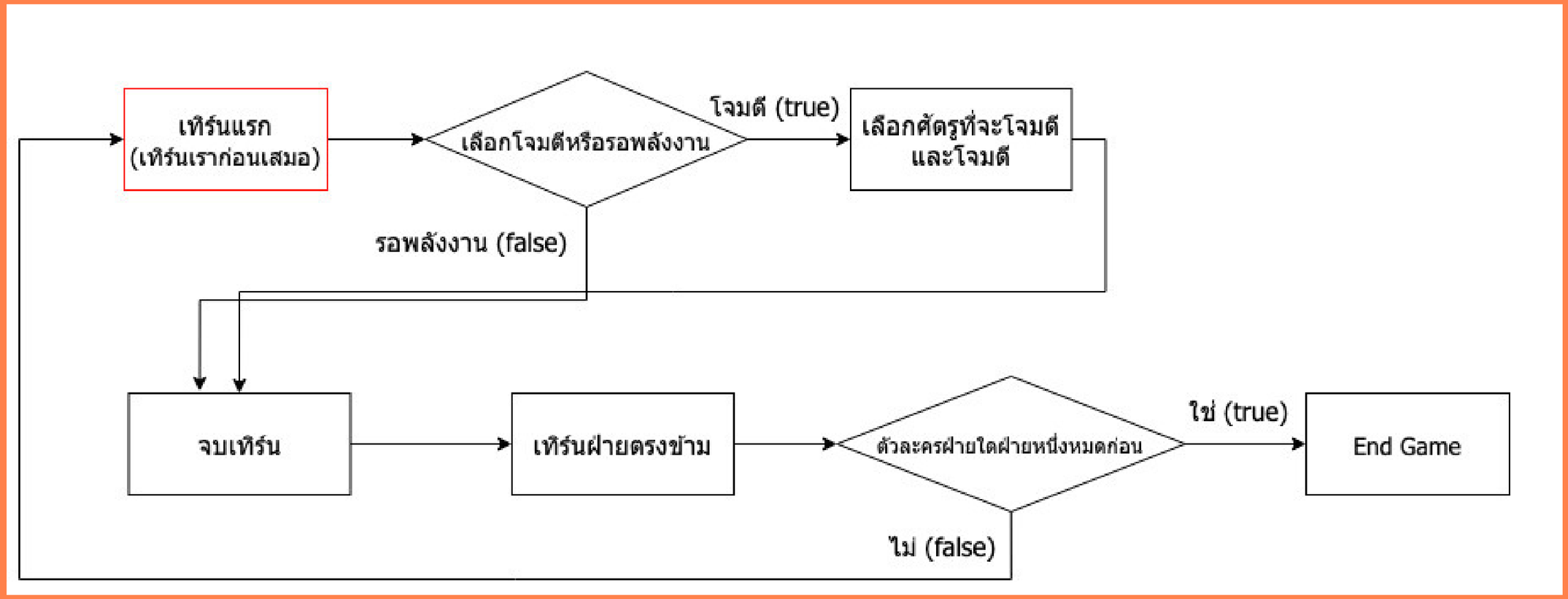
1. การเพิ่มจำนวนหุ่นยนต์ให้มากกว่า 3 ชนิด รวมถึงการแสดงผลหุ่นยนต์บนหน้าจอมากกว่า 3 ตัว จะต้องปรับแก้โปรแกรมหลายจุด
2. พวกศัตรูถูกออกแบบมาให้ทำได้แค่โจมตีเท่านั้น หากต้องการให้ศัตรูมีทักษะมากกว่าแค่การโจมตีจะต้องปรับแก้โปรแกรมหลายจุด
3. การทำงานของโปรแกรม จะต้องสั่งผ่านการใส่ตัวเลขลงไปเท่านั้น
4.หากทำการส่งค่าเข้าไปก่อนที่โปรแกรมจะขอจะทำให้โปรแกรมทำงานผิดพลาดได้
5.ถ้ากดเริ่มเกมไปแล้วไม่สามารถออกกลางคันได้จนกว่าจะชนะเกมหรือแพ้



# โชว์เกมมมมม

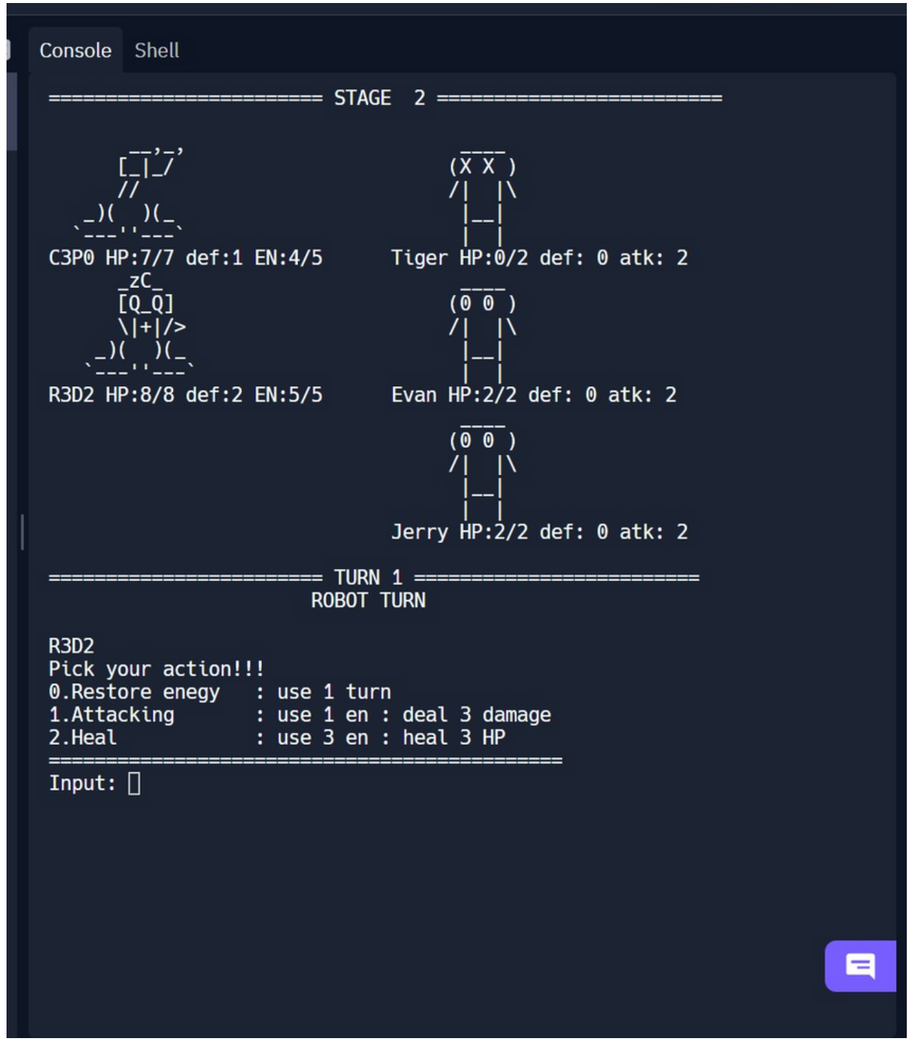
( Chow - Gemmmmm )

- **Flowchart ของเกมแบบย่อ**



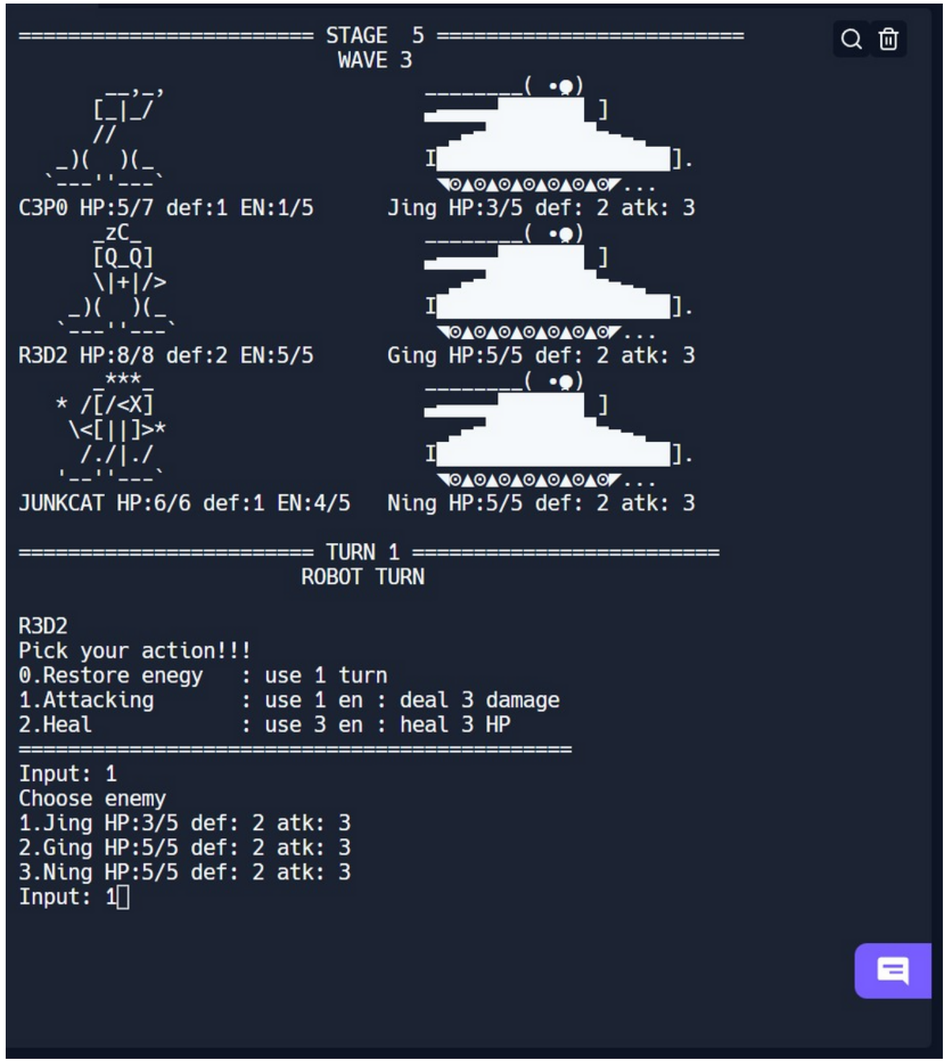
# Example


Console
Shell
========== STAGE 2 ==========
In 2099 , After successfully turning back time,
the first robot began surveying the area and
killing humans in order to prevent the planet
from collapsing.
ENTER 1 TO CONTINUE


Console
Shell
========== STAGE 2 ==========
C3P0 HP:7/7 def:1 EN:4/5
Tiger HP:0/2 def: 0 atk: 2
R3D2 HP:8/8 def:2 EN:5/5
Evan HP:2/2 def: 0 atk: 2
Jerry HP:2/2 def: 0 atk: 2
========== TURN 1 ==========
ROBOT TURN
R3D2
Pick your action!!!
0.Restore enegy : use 1 turn
1.Attacking : use 1 en : deal 3 damage
2.Heal : use 3 en : heal 3 HP
==========
Input:

# Example

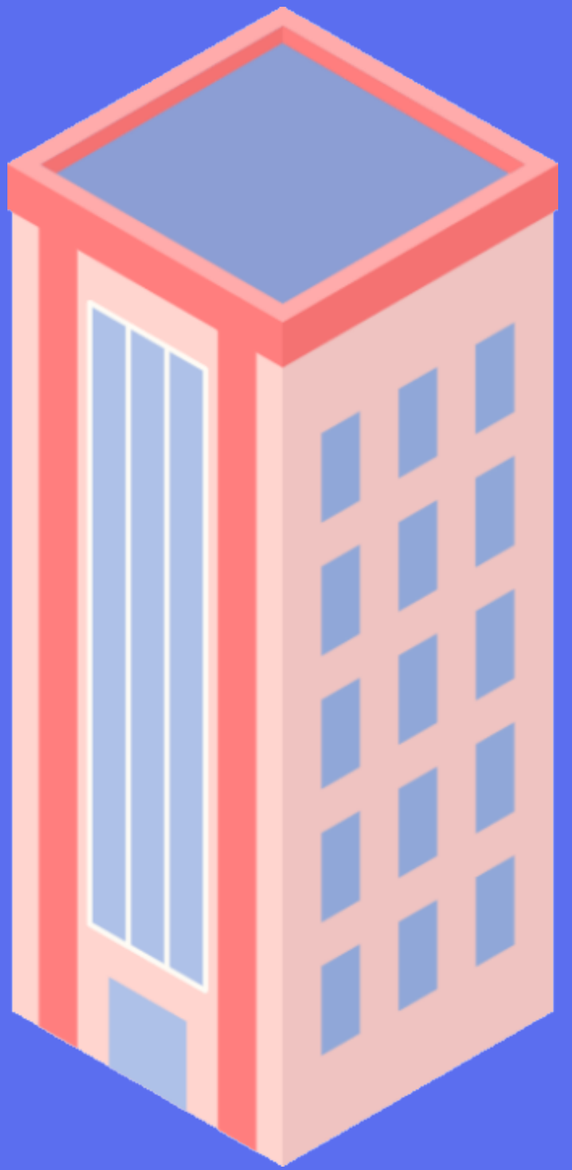

========== STAGE 4 ==========
WAVE 2
C3P0 HP:7/7 def:1 EN:2/5
Tony HP:2/2 def: 0 atk: 2
R3D2 HP:8/8 def:2 EN:4/5
Abby HP:4/4 def: 1 atk: 2
JUNKCAT HP:6/6 def:1 EN:5/5
Wendy HP:5/5 def: 2 atk: 3
========== TURN 1 ==========
ROBOT TURN
C3P0
Pick your action!!!
0.Restore enegy : use 1 turn
1.Attacking : use 1 en : deal 4 damage
2.Critical attack : use 3 en : deal 10 damage: cd 2 turn
==========
Input:


========== STAGE 5 ==========
WAVE 3
C3P0 HP:5/7 def:1 EN:1/5
Jing HP:3/5 def: 2 atk: 3
R3D2 HP:8/8 def:2 EN:5/5
Ging HP:5/5 def: 2 atk: 3
JUNKCAT HP:6/6 def:1 EN:4/5
Ning HP:5/5 def: 2 atk: 3
========== TURN 1 ==========
ROBOT TURN
R3D2
Pick your action!!!
0.Restore enegy : use 1 turn
1.Attacking : use 1 en : deal 3 damage
2.Heal : use 3 en : heal 3 HP
==========
Input: 1
Choose enemy
1.Jing HP:3/5 def: 2 atk: 3
2.Ging HP:5/5 def: 2 atk: 3
3.Ning HP:5/5 def: 2 atk: 3
Input: 1

# ไปดูกันเลยย

( Pai Du Kan Loei-ei )

# โครงสร้าง



# Class: Charactor

class character เป็นคลาสแม่ของ Robot และ Enemy
ทำหน้าที่เก็บและแสดงผลค่าสถานะต่างๆ คือ Name, Hp(health points), Df(defense) และ Atk(attack)

```
class character : public NODE     // mother class
{
protected:
  string name;
  int hp;       // health point
  int max_hp;   // hp สูงสุด
  int atk;      // attack point
  int df;       // defense point
  int cd_1 = 0; // cooldown skill1
  int po = 0;   // position

public:
  character(string n, int a, int c, int d) {
    max_hp = a;
    hp = a;
    df = c;
    atk = d;
    name = n;
  }
  int checkdeath() { //เช็คว่าตายหรือยัง
    if (hp <= 0) {
      return 1;
    } else
      return 0;
  }
  void setpo(int a) { po = a; } //ตั้งตำแหน่งในหน้าจอ
  int getpo() { return po; } //รับค่าตำแหน่ง
  int getcd_1() { return cd_1; }
  void changecd_1(int a) {
    cd_1 = cd_1 + a;
    if (cd_1 < 0) {
      cd_1 = 0;
    }
  }
  string getname() { return name; }
  int gethp() { return hp; }
  int getdf() { return df; }
  int getatk() { return atk; }
  void takedamg(int d) { //เลือดลด
    d = d - df;
    hp = hp - d;
    if (hp < 0) {
      hp = 0;
    }
    cout << name << " take " << d << " damage." << endl;
  }
  void takeheal(int d) { //รับฟื้นฟูhp
    if (d + hp > max_hp) {
      d = max_hp - hp;
      hp = max_hp;
    } else
      hp = hp + d;
    cout << name << " was healed " << d << " HP." << endl;
  }
  void show() {
    cout << name << " HP:" << hp << "/" << max_hp << " def: " << df
         << " atk: " << atk << endl;
  }
  string getstat() {
    string a = name + " HP:" + to_string(hp) + "/" + to_string(max_hp) +
               " def: " + to_string(df) + " atk: " + to_string(atk);
    return a;
  }
  ~character() {}
};
```

# Class: Robot

class Robot เป็นคลาสแม่ของ robot1 robot2 และrobot3 ทำหน้าที่จัดเก็บและแสดงผล รวมถึงคำนวนค่าสถานะพิเศษที่มีเฉพาะในตัวละครหุ่นยนต์ ซึ่งคือ En (Energy)

```
class Robot : public character //หุ่นยนต์ฝั่งเรา Inheritances class
{
protected:
  int max_en; // enegy สูงสุดที่สามารถเก็บได้
  int en;
  int en_c = 0;
  string type;

public:
  Robot(string n, string t, int h, int e, int d, int a)
      : character(n, h, d, a) {
    max_en = e;
    en = e;
    type = t;
  }
  int attack(void) {
    int attack;
    srand(time(NULL));
    attack = atk;
    this->usedenergy(1);
    cout << "You just attacking on enemy with " << attack
         << " damage.\nUse 1 enegy " << name << " has " << en << " enegy left "
         << endl;
    return attack;
  }
```

```
  string gettype() { return type; }
  int geten() { return en; }
  bool check_reenergy(int t) {
    if ((en_c + 1) == t) {
      en_c = 0;
      return true;
    } else {
      en_c = t;
      return false;
    }
  }
  void usedenergy(int x) { en = en - x; }
  void renew_energy() {
    int d = 3;
    if (en + d > max_en) {
      d = max_en - en;
      en = max_en;
    } else {
      en = en + d;
    }
    cout << name << " restored " << d << " enegy." << endl;
    sleep_for(seconds(1));
  }
  ~Robot() {}
;
```

# Class: Enemy

class enemy ทำหน้าที่เก็บประเภทของศัตรู และใช้สำหรับแสดงผลการโจมตีของฝั่งศัตรู

```
class enemy : public character //ฝั่งศัตรู //Inheritances class from character
{
private:
  string type;
public:
  enemy(string t, string n, int h, int d, int a) : character(n, h, d, a) {
    type = t;
  }
  int attack(void) {
    int attack;
    srand(time(NULL));
    attack = 1 * atk;
    return attack;
  }
  ~enemy() {}
};
```

# Class: Node

```
class NODE {
  NODE *next;

public:
  NODE();
  virtual void show() {}
  void insert(NODE *&);
  NODE *move_next();
  virtual int takedamg(int) {}
  virtual void takeheal(int) {}
  virtual int checkdeath() {}
  virtual int attack() {}
  virtual int skill_1() {}
  virtual void RobotAction() {}
  virtual int geten() {}
  virtual string gettype() {}
  virtual string getname() {}
  virtual string getstat() {}
  virtual int getcd_1() {}
  virtual int getpo() {}
  virtual void setpo(int) {}
  virtual void renew_energy() {}
  virtual void changecd_1(int) {}
  virtual string *getnormal() {}
  virtual string *getshake() {}
  virtual string *getdeath() {}
  virtual string *getattack_1() {}
  virtual string *getattack_2() {}
  virtual string *getskill(){}
  virtual string   *getreenergy() {}
  virtual void gettakedamg(int){}
  virtual ~NODE();
};

NODE::NODE() { next = NULL; }
NODE::~NODE() {}
NODE *NODE::move_next() { return next; }
void NODE::insert(NODE *&x) { x->next = this; }
```

# Class: LL (Links list)

```
class LL {
  NODE *hol;
  int size;

public:
  void add_node(NODE *&);
  void show_all();
  void show_name();
  void attacked(int, int, screen *);
  void all_attacked(int, screen *);
  void healed(int, int, screen *);
  void checkdeath(screen *);
  int getsize();
  string getname();
  NODE *gethol();
  void robotaction(LL *, screen *, LL *);
  void enemyaction(LL *, screen *);
  void robotscreen(screen *);
  void enemyscreen(screen *);
  void datascreen(screen *);
  void clear();
  ~LL();
  LL();
};
```

```
void LL::robotscreen(screen *S) {
  NODE *t = hol;
  int i;
  for (i = 0; i < size; i++) {
    t->setpo(i + 1);
    S->receive(t->getpo(), t->getnormal());
    S->receive_data(t);
    t = t->move_next();
  }
  S->show();
}

void LL::enemyscreen(screen *S) {
  NODE *t = hol;
  int i;
  for (i = 0; i < size; i++) {
    t->setpo(-(i + 1));
    S->receive(t->getpo(), t->getnormal());
    S->receive_data(t);
    t = t->move_next();
  }
  S->show();
}

void LL::datascreen(screen *S) {
  NODE *t = hol;
  int i;
  for (i = 0; i < size; i++) {
    S->receive_data(t);
    t = t->move_next();
  }
  S->show();
}
```

# การทำงานของคลาสตัวละคร

C++ Project

# Robot 1

เก็บรูปแบบการแสดงผลของตัวละครในการกระทำต่างๆ และค่าพลังงานของหุ่นยนต์

```
class robot1 : public Robot // Inheritances class from robot
{
private:
  string normal[5] = {"       __,_,    ", "      [_|_/     ", "      //       ",
                      "    _)(  )(_    ", "  `---''---`    "};
  string shake[5] = {"      __>_<     ", "     [_|_/      ", "     //        ",
                     "   _)(  )(_     ", " `---''---`     "};
  string death[5] = {"       __x_x    ", "      [_|_/     ", "      //       ",
                     "    _)(  )(_    ", "  `---''---`    "};
  string criti[5] = {"       __>_< /", "       [_|_/ / ", "       //   /  ",
                     "    _)(  ) |    ", "   `---''---`   "};
  string attack_1[5] = {"        __,_,   ", "       [_|_/    ", "       //  /    ",
                        "    _)(  ) |    ", "   `---''---`   "};
  string attack_2[5] = {"        __,_,   ", "       [_|_/    ", "       //       ",
                        "    _)(  ) /)   ", "   `---''---`   "};
  string reenergy[5] = {"   *  __^_^     ", "     [_|_/  *   ", "    * ||        ",
                        "    _)(  )(_    ", "  `---''---`    "};

public:
  robot1(string n = "C3P0", string t = "Robot1", int h = 7, int e = 5,
         int d = 1, int a = 4)
      : Robot(n, t, h, e, d, a) {}
  string getstat() {
    string a = "C3P0 HP:" + to_string(hp) + "/" + to_string(max_hp) +
               " def:" + to_string(df) + " EN:" + to_string(en) + "/" +
               to_string(max_en);
    //"C3P0 HP:5/5 def:5/5 EN:5/5"
    return a;
  }
  void RobotAction(void) {
    cout << "Pick your action!!!" << endl;
    cout << "0.Restore enegy   : use 1 turn" << endl;
    cout << "1.Attacking       : use 1 en : deal " << atk << " damage" << endl;
    cout << "2.Critical attack : use 3 en : deal " << atk * 2.5 << " damage";
    if (cd_1 > 0) {
      cout << ": cd " << cd_1 << " turn";
    }
    cout << endl << "=========================================" << endl;
  }
```

# Robot 2

เก็บรูปแบบการแสดงผลของตัวละครในการกระทำต่างๆ และค่าพลังงานของหุ่นยนต์ตัวที่ 2

```
class robot2 : public Robot // Inheritances class from robot
{
private:
  string normal[5] = {"      _zC_     ", "      [Q_Q]    ", "      \\|+|/>   ",
                      "    _)(  )(_   ", "   `---''---`  "};
  string shake[5] = {"     _zC_      ",
                     "     [v_v]     ",
                     "      |/+|/    ",
                     "   _)(  )(_    ",
                     "  `---''---`   "};
  string heal[5] = {"       _zC_    ", "    +  [*_*]   ", "       \\|+|/>+ ",
                    "     _)(  )(_  ", "    `---''---` "};
  string attack_1[5] = {"       _zC_    ", "       [Q_Q]0  ",
                        "       \\|+|/^  ", "     _)(  )(_  ",
                        "    `---''---` "};
  string attack_2[5] = {"       _zC_    ", "       [Q_Q]   ",
                        "       \\|+|/>00", "     _)(  )(_  ",
                        "    `---''---` "};
  string reenergy[5] = {"      _zC_     ", "      [-_-]zZ  ",
                        "      \\|+|/    ", "    _)(  )(_   ",
                        "   `---''---`  "};
  string death[5] = {"      _zC_     ", "      [XoX]    ", "       |/+|/   ",
                     "    _)(  )(_   ", "    _)(  )(_   "};

public:
  robot2(string n = "R3D2", string t = "Robot2", int h = 8, int e = 5,
         int d = 2, int a = 3)
      : Robot(n, t, h, e, d, a) {}
  int skill_1() {
    int heal;
    srand(time(NULL));
    heal = (1) * atk;
    this->usedenergy(3);
    cout << "You just use heal on an ally with " << heal
         << " damage.\nUse 3 enegy " << name << " has " << en << " energy left "
         << endl;
    return heal;
  }
```

# Robot 3

เก็บรูปแบบการแสดงผลของตัวละครในการกระทำต่างๆ และค่าพลังงานของหุ่นยนต์ตัวที่ 3

```
class robot3 : public Robot // Inheritances class from robot
{
private:
  string normal[5] = {"      _***_     ", "   * /[/<X]     ", "    \\<[||]>*   ",
                      "     /./|./     ", "   '--''---`    "};
  string shake[5] = {"     _***_      ", "    /['><] *    ", "   |_>O|]>/     ",
                     "     |.|.|      ", "    '--''---`   "};
  string bomb[5] = {"    _***_       ", "   /[c<X]  _*_ ", "   |_[||]-<[_] ",
                    "    |./|.|      ", "   '--''---`    "};
  string attack_1[5] = {"      _***_     ", "     /[;<X]     ", "    |_[||]>o*  ",
                        "      |./|.|   ", "     '--''---` "};
  string attack_2[5] = {"      _***_     ", "     /[;<X] ,   ", "    |_[||]O))* ",
                        "      |./|.| ' ", "     '--''---` "};
  string reenergy[5] = {"      _***_     ", "     /[;--]     ", "    |_>||]<     ",
                        "      |.|.|     ", "     '--''---   "};
  string death[5] = {"      _***_     ", "     /['XX]     ", "    |_>O|]>     ",
                     "      |.|.|     ", "     '--''---` "};

public:
  robot3(string n = "JUNKCAT", string t = "Robot3", int h = 6, int e = 5,
         int d = 1, int a = 5)
      : Robot(n, t, h, e, d, a) {}
  void RobotAction(void) {
    cout << "Pick your action!!!" << endl;
    cout << "0.Restore enegy    : use 1 turn" << endl;
    cout << "1.Attacking        : use 1 en : deal " << atk << " damage" << endl;
    cout << "2.Fire in the hole: use 3 en : deal " << atk << " damage";
    if (cd_1 > 0) {
      cout << ": cd " << cd_1 << " turn";
    }
    cout << endl << "=========================================" << endl;
  }
```

# Human

เก็บรูปแบบการแสดงผลของ human ในการกระทำต่างๆ

```
class human : public enemy // Inheritances class from enemy
{
private:
  string normal[5] = {"      ____     ", "     (0 0 )   ", "     /|  |\\    ",
                      "      |__|      ", "       |  |     "};
  string shake[5] = {"       ____    ", "      (> < )  ", "      /|  |\\   ",
                     "       |__|    ", "       /  /     "};
  string death[5] = {"      ____     ", "     (X X )   ", "     /|  |\\    ",
                     "      |__|      ", "       |  |     "};
  string attack_1[5] = {"     ____      ", "    (0 0 )    ", "    -| \\|     ",
                        "     |__|      ", "      |  |      "};
  string attack_2[5] = {"     ____      ", "    (0 0 )    ", "    --| \\|     ",
                        "     |__|      ", "      |  |      "};
public:
  human(string n, string t = "human", int h = 2, int d = 0, int a = 2)
      : enemy(t, n, h, d, a) {}
  ~human() {
    if (hp > 0) {
    } else {
      sleep_for(milliseconds(500));
      cout << endl << name << " : Noooooooo!!!!" << endl;
      sleep_for(seconds(1));
      cout << endl << "Human " << name << " is dead." << endl;
      sleep_for(seconds(1));
    }
  }
```

# Fat Human

เก็บรูปแบบการแสดงผลของ fat human ในการกระทำต่างๆ

```
class fat_human : public enemy // Inheritances class from enemy
{
private:
  string normal[5] = {"      |||||    ", "     ( o o )    ",
                      "    //     \\\\   ", "      \\ ooo /      ",
                      "     ||    ||    "};
  string shake[5] = {"       |||||   ", "      ( @ @')   ",
                     "     //     \\\\  ", "       \\ ooo /     ",
                     "      //    //   "};
  string death[5] = {"      |||||    ", "     ( x x )    ",
                     "    //     \\\\    ", "      \\ ooo /      ",
                     "     ||    ||    "};
  string attack_1[5] = {"     |||||     ", "  \\ ( o o )     ",
                        "   \\\/     \\\\    ", "    \\ ooo /       ",
                        "    ||    ||     "};
  string attack_2[5] = {"     |||||     ", "    ( o o )     ",
                        "   //     \\ \\    ", "  / \\ ooo /       ",
                        "    ||    ||     "};
public:
  fat_human(string n, string t = "fat human", int h = 4, int d = 1, int a = 2)
      : enemy(t, n, h, d, a) {}
  ~fat_human() {
    if (hp > 0) {
    } else {
      sleep_for(milliseconds(500));
      cout << endl << name << " : My child, I'm sorry..." << endl;
      sleep_for(seconds(1));
      cout << endl << "Fat human " << name << " is dead." << endl;
      sleep_for(seconds(1));
    }
  }
```

# Soldier Human

เก็บรูปแบบการแสดงผลของ soldier human ในการกระทำต่างๆ

```
class soldier_human : public enemy // Inheritances class from enemy
{
private:
  string normal[5] = {"    ________( •●)   ", "  ▄▄▄▄▄▄▄▄▄▄▄▄ ] ",
                      "   ▄▄▄▄▄▄▄▄▄▄▄▄▄▄  ", "  I▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄]. ",
                      "  ◥⊙▲⊙▲⊙▲⊙▲⊙▲⊙◤...    "};
  string shake[5] = {"    ________( •●)  ", "  ▄▄▄▄▄▄▄▄▄▄▄▄ ]",
                     "   ▄▄▄▄▄▄▄▄▄▄▄▄▄▄ ", "  I▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄].",
                     "  ◥⊙▲⊙▲⊙▲⊙▲⊙▲⊙◤...   "};
  string death[5] = {"   ________( X )   ", "  ▄▄▄▄▄▄▄▄▄▄▄▄ ] ",
                     "   ▄▄▄▄▄▄▄▄▄▄▄▄▄▄  ", "  I▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄]. ",
                     "  ◥⊙▲⊙▲⊙▲⊙▲⊙▲⊙◤...    "};
  string attack_1[5] = {" |________( •●)    ", "-o▄▄▄▄▄▄▄▄▄▄▄▄ ]  ",
                        " | ▄▄▄▄▄▄▄▄▄▄▄▄▄▄  ", "  I▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄].  ",
                        "  ◥⊙▲⊙▲⊙▲⊙▲⊙▲⊙◤...    "};
  string attack_2[5] = {"    ________( •●)   ", "***▄▄▄▄▄▄▄▄▄▄▄▄ ] ",
                        "   ▄▄▄▄▄▄▄▄▄▄▄▄▄▄  ", "  I▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄]. ",
                        "  ◥⊙▲⊙▲⊙▲⊙▲⊙▲⊙◤...    "};
public:
  soldier_human(string n, string t = "Soldier human", int h = 5, int d = 2,
                int a = 3)
      : enemy(t, n, h, d, a) {}
  ~soldier_human() {
    if (hp > 0) {
    } else {
      sleep_for(milliseconds(500));
      cout << endl << name << " : mission...";
      sleep_for(milliseconds(500));
      cout << "failed" << endl;
      sleep_for(seconds(1));
      cout << endl << "Soldier human " << name << " is dead." << endl;
      sleep_for(seconds(1));
    }
  }
```

# Super Human

เก็บรูปแบบการแสดงผลของ super human (boss) ในการกระทำต่างๆ

```
class super_human : public enemy // Inheritances class from enemy
{
private:
  string normal[5] = {
      "           _^_^_            ", "    *    ^, |@o@|  ,^        ",
      "       /VA^/|-v-|\\^AV\\   *  ", "      /VAV /| | |\\ VAV\\     ",
      "   * /AVAV/ /_I_\\ \\VAVA\\    "};
  string shake[5] = {"           _^_^_       ", "    ouch    |>o<|       ",
                     "         (\\    /)     ", "          (\\   /)      ",
                     "          (V)        "};
  string death[5] = {
      "           _^_^_            ", "    *    ^, |X_X|  ,^        ",
      "       /VA^/|-v-|\\^AV\\   *  ", "      /VAV /| | |\\ VAV\\     ",
      "   * /AVAV/ /_I_\\ \\VAVA\\    "};
  string attack_1[5] = {"           _^_^_       "
                        "           |@o@|       ",
                        "           (   )       ",
                        "            \\ /        ",
                        "             V         "};
  string attack_2[5] = {
      "           _^_^_            ", "      (!! ^___ |@o@| ___^     ",
      "       (!! \\  \\ _(_ /  /   ", "       (!! \\ \\  )  / /     ",
      "         (!!  \\VVVVV/       "};
public:
  super_human(string n, string t = "Super human", int h = 10, int d = 3,
              int a = 4)
      : enemy(t, n, h, d, a) {}
  ~super_human() {
    if (hp > 0) {
    } else {
      sleep_for(milliseconds(500));
      cout << endl << name << " : Superhuman....never...lost." << endl;
      sleep_for(seconds(1));
      cout << endl << "Super human " << name << " is dead." << endl;
      sleep_for(seconds(3));
    }
  }
```

# การทำงานของระบบการต่อสู้

C++ Project

# 1.เข้ามาที่ฟังก์ชัน stage1_battle()

เล่าเนื้อเรื่องประจำstage

```
int stage1_battle() // 1 Robot , 2 Human , 1 Fat human
{
  string a = "0";
  while (a != "1") {
    system("clear");
    cout << "======================== STAGE  1 ========================" << endl
         << endl;
    cout << stage1() << endl << endl;
    cout << "ENTER 1 TO CONTINUE" << endl;
    cin >> a;
  }
  LL A;
  LL R;
  int i;
  NODE *t;
  NODE *p;
  t = new human("Robert");
  A.add_node(t);
  t = new human("Teon");
  A.add_node(t);
  t = new fat_human("Raul");
  A.add_node(t);
  t = new robot1;
  R.add_node(t);
  screen S;
  S.getstage("STAGE  1");
  R.robotscreen(&S);
  A.enemyscreen(&S);
  if (battle(&A, &R, &S) == 1) {
    cout << "You win." << endl;
    return 1;
  } else {
    cout << "You lose." << endl;
    return 0;
  }
  sleep_for(seconds(3));
}
```

สร้างลิ้งค์ลิสสำหรับหุ่นยนต์และศัตรู

# 2.เข้ามาที่ฟังก์ชัน battle(LL *A, LL*R, screen *S)

สร้างลูปวนเทิร์น

จบเทิร์น 10 แล้วแพ้ทันที

```
int battle(LL *A, LL *R, screen *S) {
  for (int turn = 1; turn <= 10; turn++) {
    A->datascreen(S);
    R->datascreen(S);
    S->getturn("TURN " + to_string(turn), "ROBOT TURN");
    int k = 0;
    sleep_for(seconds(1));
    R->robotaction(A, S, R);
    S->getturn("TURN " + to_string(turn), "ENEMY TURN");
    A->enemyaction(R, S);
    sleep_for(seconds(1));
    if (R->getsize() <= 0) {
      return 0;
    } else if (A->getsize() <= 0) {
      return 1;
    } else if (turn == 10) {
      cout << "Time is over" << endl;
      return 0;
    }
  }
  return 0;
}
```

หุ่นยนต์ทำaction

แล้วตามด้วยศัตรูทำaction

# 3.เข้ามาที่ฟังก์ชัน robotaction ในคลาส LL

```
void LL::robotaction(LL *A, screen *M, LL *R) {
  NODE *t = hol;
  int i, k, a;
  for (i = 0; i < size; i++) {
    M->receive_data(t);
    M->show();
    t->changecd_1(-1); //ทุกเทิร์น คูลดาวน์ลดอัตโนมัต
    k = 0;
    if (A->getsize() == 0) {
      cout << "no enemy left." << endl;
      t->renew_energy();
      M->receive(t->getpo(), t->getreenergy());
      M->show();
      sleep_for(seconds(1));
      M->receive(t->getpo(), t->getnormal());
      M->show();
    } else {
      cout << t->getname() << endl;
      sleep_for(milliseconds(500));
      if (t->geten() <= 0) {
        cout << "Wait for next turn, Energy isn't enough." << endl;
        t->renew_energy();
        M->receive(t->getpo(), t->getreenergy());
        M->show();
        sleep_for(seconds(1));
        M->receive(t->getpo(), t->getnormal());
        M->show();
      } else {
        int RA = 0;
        do {
          a = 0;
          try {
            t->RobotAction();
            cout << "Input: ";
            cin >> RA;
            if (cin.fail())
              throw 4;
            if (RA > 2 || RA < 0)
              throw "input choose be more or equal to 0 and less or equal to 2";
```

```
void RobotAction(void) {
  cout << "Pick your action!!!" << endl;
  cout << "0.Restore enegy   : use 1 turn" << endl;
  cout << "1.Attacking       : use 1 en : deal " << atk << " damage" << endl;
  cout << "2.Critical attack : use 3 en : deal " << atk * 2.5 << " damage";
  if (cd_1 > 0) {
    cout << ": cd " << cd_1 << " turn";
  }
  cout << endl << "========================================" << endl;
}
```

```
void RobotAction(void) {
  cout << "Pick your action!!!" << endl;
  cout << "0.Restore enegy   : use 1 turn" << endl;
  cout << "1.Attacking       : use 1 en : deal " << atk << " damage" << endl;
  cout << "2.Heal            : use 3 en : heal " << atk << " HP";
  if (cd_1 > 0) {
    cout << ": cd " << cd_1 << " turn";
  }
  cout << endl << "========================================" << endl;
}
```

```
void RobotAction(void) {
  cout << "Pick your action!!!" << endl;
  cout << "0.Restore enegy   : use 1 turn" << endl;
  cout << "1.Attacking       : use 1 en : deal " << atk << " damage" << endl;
  cout << "2.Fire in the hole: use 3 en : deal " << atk << " damage";
  if (cd_1 > 0) {
    cout << ": cd " << cd_1 << " turn";
  }
  cout << endl << "========================================" << endl;
}
```

# 4.switch case เลือก action ใน robotaction

4.1 เลือก case 0

ไปต่อที่ฟังก์ชัน renew_energy ในคลาสRobot

```
switch (RA) {
case 0:
  t->renew_energy();
  M->receive(t->getpo(), t->getreenergy());
  M->show();
  sleep_for(seconds(1));
  M->receive(t->getpo(), t->getnormal());
  M->show();
  break;
```

```
void renew_energy() {
  int d = 3;
  if (en + d > max_en) {
    d = max_en - en;
    en = max_en;
  } else {
    en = en + d;
  }
  cout << name << " restored " << d << " enegy." << endl;
  sleep_for(seconds(1));
}
```

# 4.switch case เลือก action ใน robotaction

4.2 เลือก case 1

ไปต่อที่ฟังก์ชัน attacked ในคลาสLL

```
case 1:
  int choose;
  do { // ---> exception
    cout << "Choose enemy" << endl;
    A->show_all();
    cout << "Input: ";
    a = 0;
    try {
      cin >> choose;
      if (cin.fail())
        throw 4;
      if (choose > A->getsize() || choose <= 0)
        throw "input choose be more than 0 and less or equal to ";

    } catch (int e) {
      cout << "Input choose be integer" << endl;
      cin.clear();
      cin.ignore(50, '\n');
      sleep_for(seconds(1));
      system("clear");
      M->show();
      a = 1;
    } catch (const char *e) {
      cin.clear();
      cin.ignore(50, '\n');
      int H = A->getsize();
      cout << e << H << endl;
      sleep_for(seconds(2));
      system("clear");
      M->show();
      a = 1;
    }
  } while (a);
  M->attack(t);
  A->attacked(choose, t->attack(), M);
  break;
```

```
void LL::attacked(int x, int a, screen *M) {
  NODE *t = hol;
  for (int i = 1; i < x; i++) {
    t = t->move_next();
  }
  int b = t->takedamg(a);
  M->receive_data(t);
  M->shake(t);
  t->gettakedamg(b);
}
```

ไปต่อที่ฟังก์ชัน takedamg และ gettakedamg ในคลาส character

# 4.switch case เลือก action ใน robotaction

4.3 เลือก case 2

เช็คว่าเป็นหุ่นยนต์ประเภทไหน

```
case 2:
  int Choose;
  if (t->gettype() == "Robot1") {
    do { // ---> exception
      cout << "Choose enemy" << endl;
      A->show_all();
      cout << "Input: ";
      a = 0;
      try {
        cin >> Choose;
        if (cin.fail())
          throw 4;
        if (Choose > A->getsize() || Choose <= 0)
          throw "input choose be more than 0 and less or equal to ";

      } catch (int e) {
        cout << "Input choose be integer" << endl;
        cin.clear();
        cin.ignore(50, '\n');
        sleep_for(seconds(1));
        system("clear");
        M->show();
        a = 1;
      } catch (const char *e) {
        cin.clear();
        cin.ignore(50, '\n');
        int H = A->getsize();
        cout << e << H << endl;
        sleep_for(seconds(2));
        system("clear");
        M->show();
        a = 1;
      }
    } while (a);
    M->skill(t);
    A->attacked(Choose, t->skill_1(), M);
  } else if (t->gettype() == "Robot2") {
    do { // ---> exception
      cout << "Choose an ally" << endl;
      this->show_name();
      cout << "Input: ";
      a = 0;
      try {
        cin >> Choose;
        if (cin.fail())
          throw 4;
        if (Choose > R->getsize() || Choose <= 0)
          throw "input choose be more than 0 and less or equal to ";

      } catch (int e) {
        cout << "Input choose be integer" << endl;
        cin.clear();
        cin.ignore(50, '\n');
        sleep_for(seconds(1));
        system("clear");
        M->show();
        a = 1;
      } catch (const char *e) {
        cin.clear();
        cin.ignore(50, '\n');
        int H = R->getsize();
        cout << e << H << endl;
        sleep_for(seconds(2));
        system("clear");
        M->show();
        a = 1;
      }
    } while (a);
    M->skill(t);
    this->healed(Choose, t->skill_1(), M);
  } else if (t->gettype() == "Robot3") {
    M->skill(t);
    A->all_attacked(t->skill_1(), M);
  }
  t->changecd_1(3); //ใช้สกิล ติดcd 2 เทิร์น
```

# 5.ฟังก์ชัน checkdeath ใน LL

ตรวจสอบว่าตัวละครนั้นตายหรือยัง ถ้าใครตายแล้วก็ลบออกจากลิ้งค์ลิส

```
void LL::checkdeath(screen *M) {
  NODE *t = hol;
  NODE *p;
  int S = size;
  int i;
  for (i = 0; i < S; i++) {
    if (t->checkdeath() == 1) {
      M->receive_data(t);
      M->receive(t->getpo(), t->getdeath());
      if (t == hol) {
        hol = t->move_next();
        NODE *temp = t;
        delete temp;
        t = hol;
        size--;
      } else if (t->move_next() != NULL) {
        NODE *temp = t;
        t = t->move_next();
        t->insert(p);
        delete temp;
        size--;
      } else {
        delete t;
        size--;
      }
    } else {
      p = t;
      t = t->move_next();
    }
  }
}
```

```
int checkdeath() { //เช็คว่าตายหรือยัง
  if (hp <= 0) {
    return 1;
  } else
    return 0;
}
```

# 6.เข้ามาที่ฟังก์ชัน enemyaction ในคลาส LL

ศัตรูสุ่มโจมตีหุ่นยนต์

```
void LL::enemyaction(LL *R, screen *M) {
  M->show();
  if (size == 0) {
    cout << "All enemy dead." << endl;
  } else if (R->size == 0) {
  } else {
    NODE *t = hol;
    int i;
    for (i = 0; i < size; i++) {
      if (R->size > 0) {
        sleep_for(seconds(1));
        srand(time(NULL));
        int robotnum = rand() % R->size + 1;
        int action = 1;
        M->attack(t);
        cout << t->getname() << " attack!!!" << endl;
        sleep_for(seconds(1));
        R->attacked(robotnum, t->attack(), M);
        if (t->move_next() != NULL) {
          t = t->move_next();
        }
        R->checkdeath(M);
      }
    }
  }
  cout << endl << "END TURN" << endl;
  sleep_for(seconds(2));
}
```

# 7.เข้ามาที่ฟังก์ชัน stage2_battle()

มี 2 wave

หลังจากชนะ wave 1 เอาลิงค์ลิสรับศัตรูใหม่ต่อ

```
int stage2_battle() // 2 Robot ,  Wave1(3Human) ,Wave2(1Human,2Fat human)
{
  string a = "0";
  while (a != "1") {
    system("clear");
    cout << "====================== STAGE  2 ======================" << endl
         << endl;
    cout << stage2() << endl << endl;
    cout << "ENTER 1 TO CONTINUE" << endl;
    cin >> a;
  }
  LL A;
  LL R;
  int i;
  NODE *t;
  NODE *p;
  t = new human("Jerry");
  A.add_node(t);
  t = new human("Evan");
  A.add_node(t);
  t = new human("Tiger");
  A.add_node(t);
  t = new robot2;
  R.add_node(t);
  t = new robot1;
  R.add_node(t);
  screen S;
  S.getstage("STAGE  2");
  S.getwave("WAVE 1");
  R.robotscreen(&S);
  A.enemyscreen(&S);
```

```
  if (battle(&A, &R, &S) == 1) {
    cout << "You win." << endl;
    sleep_for(seconds(2));
    S.getwave("WAVE 2");
    t = new human("Gundam");
    A.add_node(t);
    t = new fat_human("Lumie");
    A.add_node(t);
    t = new fat_human("Mark");
    A.add_node(t);
    A.enemyscreen(&S);
    if (battle(&A, &R, &S) == 1) {
      cout << "You win." << endl;
      sleep_for(seconds(2));
      return 1;
    } else {
      cout << "You lose." << endl;
      return 0;
      sleep_for(seconds(3));
    }
  } else {
    cout << "You lose." << endl;
    return 0;
    sleep_for(seconds(3));
  }
}
```

# การทำงาน ของระบบ การแสดงผล

C++ Project

# 1. เข้ามาที่class screen

ใน screen จะประกอบไปด้วย Array ของ string เพื่อใช้เก็บค่าการแสดงผลแต่ละตำแหน่ง

```
class screen {
private:
  string free = "                ";
  string stage;
  string wave = "                ";
  string turn[2];
  string po1[6] = {"                ", "                ", "                ",
                   "                ", "                ", "                "};
  string po2[6] = {"                ", "                ", "                ",
                   "                ", "                ", "
"};
  string po3[6] = {"                ", "                ", "                ",
                   "                ", "                ", "
    "};
  string po_1[6] = {"                ", "                ", "                ",
                    "                ", "                ", "                "};
  string po_2[6] = {"                ", "                ", "                ",
                    "                ", "                ", "                "};
  string po_3[6] = {"                ", "                ", "                ",
                    "                ", "                ", "                "};
  string battledata[10];

public:
```

# 2. function receive & receive_data

มี function receive เพื่อรับค่าการแสดงผลในแต่ละตำแหน่งโดยรับค่ามาจาก function ภายนอก

```
47
48 void screen::receive(int po, string *A) {
49   switch (po) {
50   case 1:
51     for (int i = 0; i < 5; i++) {
52       po1[i] = *(A + i);
53     }
54     break;
55   case 2:
56     for (int i = 0; i < 5; i++) {
57       po2[i] = *(A + i);
58     }
59     break;
60   case 3:
61     for (int i = 0; i < 5; i++) {
62       po3[i] = *(A + i);
63     }
64     break;
65   case -1:
66     for (int i = 0; i < 5; i++) {
67       po_1[i] = *(A + i);
68     }
69     break;
70   case -2:
71     for (int i = 0; i < 5; i++) {
72       po_2[i] = *(A + i);
73     }
74     break;
75   case -3:
76     for (int i = 0; i < 5; i++) {
77       po_3[i] = *(A + i);
78     }
```

```
void screen::receive_data(NODE *t) {
  int po = t->getpo();
  switch (po) {
  case 1:
    po1[5] = t->getstat();
    break;
  case 2:
    po2[5] = t->getstat();
    break;
  case 3:
    po3[5] = t->getstat();
    break;
  case -1:
    po_1[5] = t->getstat();
    break;
  case -2:
    po_2[5] = t->getstat();
    break;
  case -3:
    po_3[5] = t->getstat();
    break;
  }
}
```

# 3.การแสดงผล function show screen

หลังจากได้รับค่าของแต่ละตำแหน่งมาเก็บไว้แล้ว

```
void screen::show() {
  system("clear");
  cout << stage << endl << wave << endl;
  for (int i = 0; i < 6; i++) {
    if (i == 5) {
      cout << po1[i] << "       " << po_1[i] << endl;
    } else {
      cout << po1[i] << free << po_1[i] << endl;
    }
  }
  for (int i = 0; i < 6; i++) {
    if (i == 5) {
      cout << po2[i] << "       " << po_2[i] << endl;
    } else {
      cout << po2[i] << free << po_2[i] << endl;
    }
  }
  for (int i = 0; i < 6; i++) {
    if (i == 5) {
      cout << po3[i] << "    " << po_3[i] << endl;
    } else {
      cout << po3[i] << free << po_3[i] << endl;
    }
  }
  cout << endl << turn[0] << endl << turn[1] << endl<< endl;
}
```

ในฟังก์ชั่น show() จะแสดงผลด้วยการพิมพ์แต่ละตำแหน่งโดยใช้ for loop

# 4.ประยุกต์การแสดงผล

จะเห็นว่าจะใช้receive รับค่าแสดงผลในแต่ละตำแหน่งมาและใช้ show()

```
void screen::shake(NODE *t) {
  int po = t->getpo();
  sleep_for(milliseconds(350));
  this->receive(po, t->getshake());
  this->receive_data(t);
  this->show();
  sleep_for(milliseconds(450));
  this->receive(po, t->getnormal());
  this->show();
}
```

```
void screen::attack(NODE *t) {
  int po = t->getpo();
  this->receive(po, t->getattack_1());
  this->receive_data(t);
  this->show();
  sleep_for(milliseconds(250));
  this->receive(po, t->getattack_2());
  this->show();
  sleep_for(milliseconds(650));
  this->receive(po, t->getnormal());
  this->show();
}
```

```
void screen::skill(NODE *t) {
  int po = t->getpo();
  this->receive(po, t->getskill());
  this->receive_data(t);
  this->show();
  sleep_for(milliseconds(1000));
  this->receive(po, t->getnormal());
  this->show();
}
```

ใช้ในการเปลี่ยนท่าทางของแต่ละตำแหน่งได้ และใช้กับfunction sleep_for เพื่อเพิ่มมิติให้กับการแสดงผล

# Exception Handling

## ดักข้อผิดพลาดจากการรับค่า (cin) แทบทุกจุดในโปรแกรม

```
int b;
do {
  int menu;
  system("clear"); // clear screen
  openwindow();
  cout << "--> 1 START GAME <--" << endl;
  cout << "--> 2 HOW TO PLAY <--" << endl;
  cout << "--> 3 ABOUT US <--" << endl;
  cout << "--> 4 EXIT <--" << endl;
  cout << "======================" << endl;
  b = 0;
  try {
    cin >> menu;
    if (cin.fail())
      throw "Invalid input";
    if (menu > 4 || menu < 1)
      throw "Wrong number please try again";
  } catch (int e) {
    cout << e << endl;
    cin.clear();
    cin.ignore(50, '\n');
    sleep_for(seconds(1));
    system("clear");
    b = 1;
  } catch (const char *e) {
    cout << e << endl;
    cin.clear();
    cin.ignore(50, '\n');
    sleep_for(seconds(2));
    system("clear");
    b = 1;
  }
```

```
int RA = 0;
do {
  a = 0;
  try {
    t->RobotAction();
    cout << "Input: ";
    cin >> RA;
    if (cin.fail())
      throw 4;
    if (RA > 2 || RA < 0)
      throw "input choose be more or equal to 0 and less or equal to 2";
    if (RA == 2 && t->getcd_1() > 0) {
      cout << "This skill is not ready.\nWait " << t->getcd_1()
           << " turn.";
      throw " ";
    }
    if (RA == 2 && t->geten() < 3) {
      throw "Energy isn't enough.";
    }

  } catch (int e) {
    cout << "Invalid input " << endl;
    cin.clear();
    cin.ignore(50, '\n');
    sleep_for(seconds(1));
    system("clear");
    M->show();
    a = 1;
  } catch (const char *e) {
    cin.clear();
    cin.ignore(50, '\n');
    cout << e << endl;
    sleep_for(seconds(2));
    system("clear");
    M->show();
    a = 1;
  }
} while (a);
```

# ผู้จัดทำ

**6413110 นายวัชรศักย์ พรหมณี**
-ออกแบบและจัดการระบบต่อสู้

**6413112 นายศศิศ ศรีรัตน์**
-จัดการเกี่ยวกับคลาสตัวละคร

**6413210 นายกวิน เก่งเกตุ**
-จัดการเกี่ยวกับลิงค์ลิส

**6413223 นายรวิพล ฉายรักษา**
-ออกแบบและจัดการระบบแสดงผลบนหน้าจอ