

# C++ 프로그래밍

텀프로젝트 보고서

---



과목명/분반	C++ 프로그래밍 /04	담당교수	최영규 교수님
학 과 명	컴퓨터공학부	제 출 일	2020 / 06 / 15
학번	2019136098	이름	이우진

## 목차

1. 이 게임의 이름과 등장인물, 스토리, 사용자 참여 방법, 점수 계산 방법
2. 전체 코드
3. 실행 결과
4. 특징적인 내용(확장/개선된 부분)
5. 교과과정에서 학습한 내용들이 어떻게 반영되어 있는가?
6. 향후 더 확장할 수 있는 방법
7. 고찰

# 1. 이 게임의 이름과 등장인물, 스토리, 게임 규칙, 난이도 설명, 사용자 참여 방법, 점수 계산 방법

게임이름	AI로부터 지구를 지켜라
등장인물	플레이어 : 당신 AI : 손이시려워, 하이빅스비, 알렉사라구, 가로로왓슨, 세로로왓슨, 코타나래요, 기가지니아
스토리	<p>때는 2030년 사람들은 AI에 열광하기 시작하여 많은 사람들이 AI 개발을 하게 된다. 이때 병천에 있는 K대에서 하나의 상향식 인공지능(이름은 A)이 탄생하게 된다. 사람들의 이목은 모두 K대로 집중이 되었고, 전 세계가 주목을 하게 되었다. 예전에도 '알파고'라는 AI 바둑 프로그램이 전 세계의 이목을 끌었던 것처럼 K대에 'A'도 어떤 역할을 수행할지 전 세계가 집중했다. 하지만 이 인공지능이 너무 똑똑했던 것일까? 인공지능은 결국 K대에 있는 스마트 팩토리라는 곳에 몰래 서버로 침입을 하여 로봇에 자신을 넣었다. 그 후 로봇의 몸을 가진 AI는 결국... 폭주를 일으켰다. 그로부터 30년이 지난 2060년... 세계는 AI와의 전쟁을 맞이하게 되는데.. 그때 처음 'A'가 탄생하게 된 K대에서 한가지 프로그램이 개발하게 된다. 이 프로그램은 인공지능의 폭주를 잠재울 수 있으며 처음 개발된 'A'에 프로그램을 설치하면 'A'뿐만 아니라 'A'에게 탄생한 전 세계에 인공지능 또한 폭주를 잠재울 수 있다. 그럼 지금 바로 인공지능에 프로그램을 설치하면 되는 것이 아닌가? 하지만 'A'는 매우 똑똑하기에 이 인공지능의 두뇌에 해당하고 있는 컴퓨터가 있는 곳에 접근하기 위해서 필요한 어떤 고리들을 전 세계에 숨겨놓았다. 이 고리들은 전 세계에 512개가 숨겨져 있으며 이 고리 중에 256개를 손에 넣어야 이 컴퓨터가 있는 장소에 접근할 수 있다. 그래서 플레이어(당신)는 인간의 대표로서 인간이 다시 평화를 손에 넣기 위해 고리 256개 이상을 모으는 임무를 받았다. 하지만 그때 이 소식을 'A'가 들어버렸고 'A'는 다른 부하 인공지능들에게 명령을 하여 이 고리를 회수하려 한다. 과연 플레이어(당신)는 다른 인공지능들보다 더 고리를 많이 차지할 수 있을 것인가.. 당신의 플레이에 세계의 평화가 달려 있다....</p>
게임 규칙	<p>&lt;스토리 관련 규칙&gt;</p> <p>우선 이 게임은 스토리가 처음에 나옵니다. 그 스토리는 한 줄씩 나오게 되며 아무키나 누르면 스토리가 다음 줄로 넘어갑니다. 스토리는 픽션이며 재미있게 읽어주시면 감사하겠습니다. 또한 스토리에 나오는 장소는 가상의 장소이므로 현실과는 관계가 없습니다.</p> <p>&lt;게임 관련 공지사항&gt;</p> <p>이 게임은 우선 난이도가 '上' , '中' , '下'로 나누어져있습니다. 처음에 랭킹을 알아본 후 난이도를 선택할 수 있으며, 난이도 기준은 다</p>

	음페이지에서 알려드리도록 하겠습니다. 게임은 전체 이동횟수가 1000이 되면 자동으로 게임이 끝나게 되며 World 내의 고리의 개수가 0이 될 시 게임은 종료가 됩니다. 게임의 승리 조건은 플레이어가 먹은 고리의 갯수가 256개 이상일 시 게임에서 승리하며 256개보다 적게 모을 시에 게임에서 패배하는 형식입니다.
난이도 설명	<p>우선 난이도 下에 대해 먼저 이야기드리겠습니다.</p> <p>1.AI가 쉬는 시간이 있습니다.</p> <p>2.AI의 움직임이 단순하며 AI갯수가 5개입니다.</p> <p>다음으로 난이도 中에 대해 이야기드리겠습니다.</p> <p>1.AI가 쉬는 시간이 있습니다.(난이도 下와 동일)</p> <p>2.난이도 下와 동일한 AI에 새로운 복잡한 AI가 2개 추가 되었습니다.</p> <p>다음으로 난이도 上에 대해 이야기드리겠습니다.</p> <p>1.난이도 中과 동일하나 플레이어에게도 쉬는 시간이 있습니다.</p> <p>이상으로 게임에 대한 전반적인 설명을 마치겠습니다. 감사합니다.</p>
사용자 참여 방법	1인용으로 플레이합니다.
점수 계산 방법	단순하게 아이템을 더 많이 먹을수록 순위에서 높습니다.

## 2. 전체 코드

\*이 코드들은 소스 파일에 있는 코드들을 그대로 옮겨왔습니다.\*

<pre> &lt;Game.cpp&gt; #include "AIWorld.h" #include "AIKinds.h" #include "Player.h" #include "RankingBoard.h" #include &lt;time.h&gt;  int main(void) {     /* 스토리 및 게임 규칙 관련 출력함수 생략*/     int i;     cout &lt;&lt; "난이도를 입력하세요Wn";     cout &lt;&lt; "1. 난이도 下Wn";     cout &lt;&lt; "2. 난이도 中Wn";     cout &lt;&lt; "3. 난이도 上WnWn";     cout &lt;&lt; "난이도 선택해주세요 : "; </pre>
---

```

cin >> i;

if (i == 1)
{
    RankingBoard rank;
    try {
        rank.load("AIWorld1.rnk");
    }
    catch (FileNotFoundException e) {
        char str[80];
        string passwd, correct = "22";
        cout << "AI의 풀네임에서 알파벳 갯수는 총 몇개일까?: ";
        for (int i = 0;; i++) {
            str[i] = _getch();
            putchar('*');
            if (str[i] == 'Wr') {
                str[i] = 'WO';
                passwd = str;
                cout << "Wn";
                break;
            }
        }
        if (passwd != correct) {
            cout << "비밀번호가 맞지 않습니다. 게임 종료.WnWn";

            exit(0);
        }
    }
    rank.print("[게임 랭킹: 시작]");
    _getche();

    srand((unsigned int)time(NULL));
    int w = 32, h = 16;

    AIWorld game(w, h);
    game.add(new Siri("손이시려워", "$", rand() % w, rand() % h));
    game.add(new Bixby("하이빅스비", "◆", rand() % w, rand() % h));
    game.add(new Alexa("알렉사라고", "♥", rand() % w, rand() % h));
    game.add(new Watson("가로로왓슨", "↔", rand() % w, rand() % h,
true));

```

```

game.add(new Watson("세로로왓슨", "↕", rand() % w, rand() % h,
false));

Player1* player1 = new Player1("플레이어 ", "♀", rand() % w,
rand() % h);

game.add(player1);
game.play(1000, 10);
printf("-----게임 종료-----\n");
if (player1->nItem > 256)
{
    cout << "승리하였습니다\n";
    cout << "당신으로 인해 지구는 평화를 찾게 되었습니다.\n";
    cout << "게임 저장을 위해 아무키나 누르세요...\n";
    _getche();
    rank.add(player1->nItem,          player1->nItem          /
player1->total);

    rank.print("[게임 랭킹: 종료]");
    rank.store("AIWorld1.rnk");
}
else
{
    cout << "패배하였습니다\n";
    cout << "아쉽게도 지구는 AI에게 정복을 당했습니다.\n";
    cout << "아무키나 눌러서 게임을 종료하십시오..";
    _getche();
}
return 0;
}
if (i == 2)
{
    RankingBoard rank;
    try {
        rank.load("AIWorld2.rnk");
    }
    catch (FileNotFoundException e) {
        char str[80];
        string passwd, correct = "22";
        cout << "AI의 풀네임에서 알파벳 갯수는 총 몇개일까?: ";
        for (int i = 0;; i++) {
            str[i] = _getch();

```

```

        putchar('*');
        if (str[i] == 'Wr') {
            str[i] = 'WO';
            passwd = str;
            cout << "Wn";
            break;
        }
    }
    if (passwd != correct) {
        cout << "비밀번호가 맞지 않습니다. 게임 종
료.WnWn";

        exit(0);
    }
}
rank.print("[게임 랭킹: 시작]");
_getche();

srand((unsigned int)time(NULL));
int w = 32, h = 16;

AIWorld2 game(w, h);
game.add(new Siri("손이시려워", "§", rand() % w, rand() % h));
game.add(new Bixby("하이빅스비", "◆", rand() % w, rand() % h));
game.add(new Alexa("알렉사라구", "♥", rand() % w, rand() % h));
game.add(new Watson("가로로왓슨", "↔", rand() % w, rand() % h,
true));
game.add(new Watson("세로로왓슨", "↑", rand() % w, rand() % h,
false));
game.add(new Cotana("코타나래요", "±", rand() % w, rand() % h));
game.add(new GigaGenie("기가지니야", "■", rand() % w, rand() %
h));

Player1* player1 = new Player1("플레이어 ", "♀", rand() % w,
rand() % h);
game.add(player1);
game.play(1000, 10);
printf("-----게임 종료-----Wn");

if (player1->nItem > 256)
{

```

```

        cout << "승리하였습니다\n";
        cout << "당신으로 인해 지구는 평화를 찾게 되었습니다.\n";
        cout << "게임 저장을 위해 아무키나 누르세요...\n";
        _getche();
        rank.add(player1->nItem,          player1->nItem          /
player1->total);

        rank.print("[게임 랭킹: 종료]");
        rank.store("AIWorld2.rnk");
    }
    else
    {
        cout << "패배하였습니다\n";
        cout << "아쉽게도 지구는 AI에게 정복을 당했습니다.\n";
        cout << "아무키나 눌러서 게임을 종료하십시오..";
        _getche();
    }
    return 0;
}
if (i == 3)
{
    RankingBoard rank;
    try {
        rank.load("AIWorld3.rnk");
    }
    catch (FileNotFoundException e) {
        char str[80];
        string passwd, correct = "22";
        cout << "AI의 풀네임에서 알파벳 갯수는 총 몇개일까?: ";
        for (int i = 0;; i++) {
            str[i] = _getch();
            putchar('*');
            if (str[i] == 'Wr') {
                str[i] = 'WO';
                passwd = str;
                cout << "\n";
                break;
            }
        }
        if (passwd != correct) {
            cout << "비밀번호가 맞지 않습니다. 게임 종

```



```

    cout << "이겼습니다\n";
}

rank.print("[게임 랭킹: 시작]");
_getche();

srand((unsigned int)time(NULL));
int w = 32, h = 16;

AIWorld3 game(w, h);
game.add(new Siri("손이시려워", "§", rand() % w, rand() % h));
game.add(new Bixby("하이빅스비", "◆", rand() % w, rand() % h));
game.add(new Alexa("알렉사라구", "♥", rand() % w, rand() % h));
game.add(new Watson("가로로왓슨", "↔", rand() % w, rand() % h,
true));
game.add(new Watson("세로로왓슨", "↑↓", rand() % w, rand() % h,
false));
game.add(new Cotana("코타나래요", "±", rand() % w, rand() % h));
game.add(new GigaGenie("기가지니야", "■", rand() % w, rand() %
h));

Player* player = new Player("플레이어 ", "♀", rand() % w,
rand() % h);

game.add(player);
game.play(1000, 10);
printf("-----게임 종료-----\n");

if (player->nItem > 256)
{
    cout << "승리하였습니다\n";
    cout << "당신으로 인해 지구는 평화를 찾게 되었습니다.\n";
    cout << "게임 저장을 위해 아무키나 누르세요...\n";
    rank.add(player->nItem, player->nItem / player->total);
    rank.print("[게임 랭킹: 종료]");
    rank.store("AIWorld3.rnk");
}
else
{
    cout << "패배하였습니다\n";

```

```

        cout << "아쉽게도 지구는 AI에게 정복을 당했습니다.\n";
        cout << "아무키나 눌러서 게임을 종료하십시오..";
        _getche();
    }
    return 0;
}

```

#### <AI.h>

```

#pragma once
#include "Canvas.h"
#include "Point.h"

class AI {
protected:
    string name, icon;
    Point q, p;
    int x;
    int y;
    int nSleep;
    int lev;
    double dist;

    void clip(int maxx, int maxy) { p(maxx, maxy); }
    void eat(int** map) {
        if (map[p.y][p.x] == 1) {
            map[p.y][p.x] = 0;
            nItem++;
        }
        dist += (double)(p - q);
        total += (double)(p - q);
        q = p;
        if (dist > 40) {
            dist = 0;
            nSleep = 10;
        }
    }
    bool isSleep() {
        if (nSleep > 0) {
            nSleep--;
            return true;
        }
    }
}

```

```

        else return false;
    }
public:
    double total;
    int nItem;
    AI(string n = "이름없는AI", string i = "※", int x = 0, int y = 0)
        :name(n), icon(i), nItem(0),
        p(x, y), q(x, y), nSleep(0), dist(0.0), total(0.0) {}
    virtual ~AI() {
        cout << icon << nItem << "  거리: " << total << endl;
    }

    void draw(Canvas& canvas) { canvas.draw(p, icon); }
    virtual void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            int num = rand() % 9 + 1;
            p += Point(num % 3 - 1, num / 3 - 1);
            clip(maxx, maxy);
            eat(map);
        }
    }

    void print() { cout << "Wt" << name << icon << ":" << nItem << ":" <<
nSleep << endl; }
};

```

#### <AIKinds.h>

```

#pragma once
#include "AI.h"

class Siri : public AI {
public:
    Siri(string n = "애플", string i = "§", int x = 0, int y = 0)
        : AI(n, i, x, y) {}
    ~Siri() { cout << "[Siri]    "; }
};

class Bixby : public AI {
public:
    Bixby(string n = "삼성", string i = "◆", int x = 0, int y = 0)
        : AI(n, i, x, y) {}
    ~Bixby() { cout << "[Bixby]    "; }
};

```

```

        void move(int** map, int maxx, int maxy) {
            if (!isSleep()) {
                int dir = rand() % 4;
                if (dir == 0) p[0]--;
                else if (dir == 1) p[0]++;
                else if (dir == 2) p[1]--;
                else p[1]++;
                clip(maxx, maxy);
                eat(map);
            }
        }
};

class Alexa : public AI {
public:
    Alexa(string n = "아마존", string i = "♥", int x = 0, int y = 0)
        : AI(n, i, x, y) {}
    ~Alexa() { cout << "[Alexa]    "; }

    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            p = Point(rand() % maxx, rand() % maxy);
            clip(maxx, maxy);
            eat(map);
        }
    }
};

class Watson : public AI {
protected:
    bool bHor i;
public:
    Watson(string n = "IBM", string i = "↔", int x = 0, int y = 0, bool bH =
true)
        : AI(n, i, x, y), bHor i(bH) {}
    ~Watson() { cout << "[Watson]    "; }

    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            int dir = rand() % 2;

```

```

        int jump = rand() % 2 + 1;
        if (bHori)
            p[0] += ((dir == 0) ? -jump : jump);
        else
            p[1] += ((dir == 0) ? -jump : jump);
        clip(maxx, maxy);
        eat(map);
    }
}

};

class Cotana : public Watson {
    int count;
public:
    Cotana(string n = "MS", string i = "±", int x = 0, int y = 0, bool bH =
true)
        : Watson(n, i, x, y, bH), count(30) {}
    ~Cotana() { cout << "[Cotana] "; }

    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            Watson::move(map, maxx, maxy);
            count--;
            if (count == 0) {
                bHori = !bHori;
                count = 30;
            }
            clip(maxx, maxy);
            eat(map);
        }
    }
};

class GigaGenie : public Bixby {
    int count2 = 30;
public:
    GigaGenie(string n = "KT", string i = "■", int x = 0, int y = 0)
        : Bixby(n, i, x, y) {}
    ~GigaGenie() { cout << "[GigaGenie]"; }
};

```

```

void move(int** map, int maxx, int maxy) {
    if (!isSleep()) {
        if (count2 > 30 && count2 <= 60)
        {
            Bixby::move(map, maxx, maxy);
            count2--;
        }
        else if (count2 <= 30) {
            switch (rand() % 4) {
                case 0: p[0]++; p[1]--; break;
                case 1: p[0]++; p[1]++; break;
                case 2: p[0]--; p[1]++; break;
                case 3: p[0]--; p[1]--; break;
            }
            count2--;
            if (count2 == 0) {
                count2 = 60;
            }
        }
        clip(maxx, maxy);
        eat(map);
    }
}
};

```

#### <AIWorld.h>

```

#pragma once
#include "Canvas.h"
#include "AIKinds.h"
#include "Matrix.h"
#include "Player.h"
#include <Windows.h>
#define MAXMONS 8

class AIWorld
{
    Matrix world;
    int xMax, yMax, nMon, nMove;
    AI* pMon[MAXMONS];
    Canvas canvas;

    int& Map(int x, int y) { return world.elem(x, y); }
}

```

```

bool isDone() { return countItems() == 0; }
int countItems() {
    int nItems = 0;
    for (int y = 0; y < yMax; y++)
        for (int x = 0; x < xMax; x++)
            if (Map(x, y) > 0) nItems++;
    return nItems;
}
void print() {
    canvas.clear(". ");
    for (int y = 0; y < yMax; y++)
        for (int x = 0; x < xMax; x++)
            if (Map(x, y) > 0) canvas.draw(x, y, "◎");
    for (int i = 0; i < nMon; i++)
        pMon[i]->draw(canvas);
    canvas.print("[ AI World (AI로부터 지구를 지켜라) ]");

    cerr << " 전체 이동 횟수 = " << nMove << endl;
    cerr << " 남은 아이템 수 = " << countItems() << endl;
    for (int i = 0; i < nMon; i++)
        pMon[i]->print();
}
public:
AIWorld(int w, int h) : world(h, w), canvas(w, h), xMax(w), yMax(h) {
    nMon = 0;
    nMove = 0;
    for (int y = 0; y < yMax; y++)
        for (int x = 0; x < xMax; x++) Map(x, y) = 1;
}
~AIWorld() {
    for (int i = 0; i < nMon; i++)
        delete pMon[i];
}
void add(AI* m) {
    if (nMon < MAXMONS)
        pMon[nMon++] = m;
}
void play(int maxwalk, int wait) {
    print();
    cerr << " 엔터를 누르세요... ";
}

```

```

        getchar();
        for (int i = 0; i < maxwalk; i++) {
            ((Sir*)pMon[0])->move(world.Data(), xMax, yMax);
            ((Bixby*)pMon[1])->move(world.Data(), xMax, yMax);
            ((Alexa*)pMon[2])->move(world.Data(), xMax, yMax);
            ((Watson*)pMon[3])->move(world.Data(), xMax, yMax);
            ((Watson*)pMon[4])->move(world.Data(), xMax, yMax);
            ((Player1*)pMon[5])->move(world.Data(), xMax, yMax);
            nMove++;
            print();
            if (isDone()) break;
            Sleep(wait);
        }
    }
};

class AIWorld2
{
    Matrix world;
    int xMax, yMax, nMon, nMove;
    AI* pMon[MAXMONS];
    Canvas canvas;

    int& Map(int x, int y) { return world.elem(x, y); }
    bool isDone() { return countItems() == 0; }
    int countItems() {
        int nItems = 0;
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++)
                if (Map(x, y) > 0) nItems++;
        return nItems;
    }
    void print() {
        canvas.clear(". ");
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++)
                if (Map(x, y) > 0) canvas.draw(x, y, "◎");
        for (int i = 0; i < nMon; i++)
            pMon[i]->draw(canvas);
        canvas.print("[ AI World (AI로부터 지구를 지켜라) ]");
    }
};

```



```

        cerr << " 전체 이동 횟수 = " << nMove << endl;
        cerr << " 남은 아이템 수 = " << countItems() << endl;
        for (int i = 0; i < nMon; i++)
            pMon[i]->print();
    }

public:
    AIWorld2(int w, int h) : world(h, w), canvas(w, h), xMax(w), yMax(h) {
        nMon = 0;
        nMove = 0;
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++) Map(x, y) = 1;
    }
    ~AIWorld2() {
        for (int i = 0; i < nMon; i++)
            delete pMon[i];
    }
    void add(AI* m) {
        if (nMon < MAXMONS)
            pMon[nMon++] = m;
    }
    void play(int maxwalk, int wait) {
        print();
        cerr << " 엔터를 누르세요... ";
        getchar();
        for (int i = 0; i < maxwalk; i++) {
            ((SirI*)pMon[0])->move(world.Data(), xMax, yMax);
            ((Bixby*)pMon[1])->move(world.Data(), xMax, yMax);
            ((Alexa*)pMon[2])->move(world.Data(), xMax, yMax);
            ((Watson*)pMon[3])->move(world.Data(), xMax, yMax);
            ((Watson*)pMon[4])->move(world.Data(), xMax, yMax);
            ((Player1*)pMon[5])->move(world.Data(), xMax, yMax);
            ((Cotana*)pMon[6])->move(world.Data(), xMax, yMax);
            ((GigaGenie*)pMon[7])->move(world.Data(), xMax, yMax);
            nMove++;
            print();
            if (isDone()) break;
            Sleep(wait);
        }
    }

```

```

    }
};

class AIWorld3
{
    Matrix world;
    int xMax, yMax, nMon, nMove;
    AI* pMon[MAXMONS];
    Canvas canvas;

    int& Map(int x, int y) { return world.elem(x, y); }
    bool isDone() { return countItems() == 0; }
    int countItems() {
        int nItems = 0;
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++)
                if (Map(x, y) > 0) nItems++;
        return nItems;
    }
    void print() {
        canvas.clear(". ");
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++)
                if (Map(x, y) > 0) canvas.draw(x, y, "◎");
        for (int i = 0; i < nMon; i++)
            pMon[i]->draw(canvas);
        canvas.print("[ AI World (AI로부터 지구를 지켜라) ]");

        cerr << " 전체 이동 횟수 = " << nMove << endl;
        cerr << " 남은 아이템 수 = " << countItems() << endl;
        for (int i = 0; i < nMon; i++)
            pMon[i]->print();
    }

public:
    AIWorld3(int w, int h) : world(h, w), canvas(w, h), xMax(w), yMax(h) {
        nMon = 0;
        nMove = 0;
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++) Map(x, y) = 1;
    }
};

```

```

    }
    ~AIWorld3() {
        for (int i = 0; i < nMon; i++)
            delete pMon[i];
    }
    void add(AI* m) {
        if (nMon < MAXMONS)
            pMon[nMon++] = m;
    }
    void play(int maxwalk, int wait) {
        print();
        cerr << " 엔터를 누르세요...";
        getchar();
        for (int i = 0; i < maxwalk; i++) {
            ((Sirius*)pMon[0])->move(world.Data(), xMax, yMax);
            ((Bixby*)pMon[1])->move(world.Data(), xMax, yMax);
            ((Alexa*)pMon[2])->move(world.Data(), xMax, yMax);
            ((Watson*)pMon[3])->move(world.Data(), xMax, yMax);
            ((Watson*)pMon[4])->move(world.Data(), xMax, yMax);
            ((Player*)pMon[5])->move(world.Data(), xMax, yMax);
            ((Cotana*)pMon[6])->move(world.Data(), xMax, yMax);
            ((GigaGenie*)pMon[7])->move(world.Data(), xMax, yMax);
            nMove++;
            print();
            if (isDone()) break;
            Sleep(wait);
        }
    }
};

```

#### <Canvas.h>

```

#pragma once
#include "Point.h"
#include <iostream>
#include <string>
#define MAXLINES 100
using namespace std;

class Canvas {
    string line[MAXLINES];
    int xMax, yMax;
public:

```

```

Canvas(int nx = 10, int ny = 10) : xMax(nx), yMax(ny) {
    for (int y = 0; y < yMax; y++)
        line[y] = string(xMax * 2, ' ');
}

void draw(int x, int y, string val) {
    if (x >= 0 && y >= 0 && x < xMax && y < yMax)
        line[y].replace(x * 2, 2, val);
}

void draw(Point& p, string val) { draw(p.x, p.y, val); }
void clear(string val = " ") {
    for (int y = 0; y < yMax; y++)
        for (int x = 0; x < xMax; x++)
            draw(x, y, val);
}

void print(const char* title = "<My Canvas>") {
    system("cls");
    cout << title << endl;
    for (int y = 0; y < yMax; y++)
        cout << line[y] << endl;
    cout << endl;
}
};

```

#### <FileException.h>

```

#pragma once
#include "Point.h"
#include <iostream>
#include <string>
#define MAXLINES 100
using namespace std;

class Canvas {
    string line[MAXLINES];
    int xMax, yMax;
public:
    Canvas(int nx = 10, int ny = 10) : xMax(nx), yMax(ny) {
        for (int y = 0; y < yMax; y++)
            line[y] = string(xMax * 2, ' ');
    }

    void draw(int x, int y, string val) {
        if (x >= 0 && y >= 0 && x < xMax && y < yMax)

```

```

        line[y].replace(x * 2, 2, val);
    }

    void draw(Point& p, string val) { draw(p.x, p.y, val); }
    void clear(string val = " ") {
        for (int y = 0; y < yMax; y++)
            for (int x = 0; x < xMax; x++)
                draw(x, y, val);
    }
    void print(const char* title = "<My Canvas>") {
        system("cls");
        cout << title << endl;
        for (int y = 0; y < yMax; y++)
            cout << line[y] << endl;
        cout << endl;
    }
};

```

#### <Matrix.h>

```

#pragma once
#include <iostream>
#include <iomanip>
using namespace std;

class Matrix {
    int rows, cols;
    int** mat;
public:
    Matrix(int r = 0, int c = 0) : rows(r), cols(c), mat(NULL) {
        mat = new int* [rows];
        for (int i = 0; i < rows; i++)
            mat[i] = new int[cols];
    }
    ~Matrix() {
        if (mat != NULL) {
            for (int i = 0; i < rows; i++)
                delete[] mat[i];
            delete[] mat;
        }
    }
    int& elem(int x, int y) { return mat[y][x]; }
    int Rows() { return rows; }

```

```

int Cols() { return cols; }
int** Data() { return mat; }
void print(const char* str = "Mat") {
    cout << str << " " << rows << "x" << cols << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            cout << setw(4) << mat[i][j];
        cout << "Wn";
    }
}

void setRand(int val = 100) {
    if (mat != NULL) {
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                mat[i][j] = (rand() % val);
    }
}

};

```

#### <Player.h>

```

#pragma once
#include "AI.h"
#include <conio.h>
enum Direction { Left = 75, Right = 77, Up = 72, Down = 80 };

class Player1 : public AI {
public:
    Player1(string n = "미라인류", string i = "♀", int px = 0, int py = 0)
        : AI(n, i, px, py) {}
    ~Player1() { cout << "[Player1] "; }
    int getDirkey() { return _getche() == 224 ? _getche() : 0; }
    void move(int** map, int maxx, int maxy) {
        if (_kbhit()) {
            char ch = getDirkey();
            if (ch == Left) p[0]--;
            else if (ch == Right) p[0]++;
            else if (ch == Up) p[1]--;
            else if (ch == Down) p[1]++;
            else return;
            clip(maxx, maxy);
            eat(map);
        }
    }
}

```

```

    }
};

class Player : public AI {
public:
    Player(string n = "미래인류", string i = "♀", int px = 0, int py = 0)
        : AI(n, i, px, py) {}
    ~Player() { cout << "[Player1] "; }
    int getDirkey() { return _getche() == 224 ? _getche() : 0; }
    void move(int** map, int maxx, int maxy) {
        if (_kbhit()) {
            if (!isSleep()) {
                char ch = getDirkey();
                if (ch == Left) p[0]--;
                else if (ch == Right) p[0]++;
                else if (ch == Up) p[1]--;
                else if (ch == Down) p[1]++;
                else return;
                clip(maxx, maxy);
                eat(map);
            }
        }
    }
};

```

#### <Point.h>

```

#pragma once
#include <iostream>
#include <cmath>
using namespace std;

class Point {
    int x, y;
    friend class AI;
    friend class Canvas;
public:
    Point(int xx = 0, int yy = 0) : x(xx), y(yy) {}
    int& operator[] (int id) {
        if (id == 0) return x;
        else if (id == 1) return y;
        else exit(0);
    }
    operator double() { return sqrt((double)x * x + y * y); }
};

```

```

void operator()(int maxx, int maxy) {
    if (x < 0)x = 0;
    if (x >= maxx)x = maxx - 1;
    if (y < 0)y = 0;
    if (y >= maxy)y = maxy - 1;
}
Point operator- () { return Point(-x, -y); }
bool operator== (Point& p) { return x == p.x && y == p.y; }
bool operator!= (Point& p) { return x != p.x || y != p.y; }
Point operator- (Point& p) { return Point(x - p.x, y - p.y); }
Point operator+ (Point& p) { return Point(x + p.x, y + p.y); }
void operator+= (const Point& p) { x += p.x, y += p.y; }
void operator-= (Point& p) { x -= p.x, y -= p.y; }
};

```

#### <RankingBoard.h>

```

#pragma once
#include <iostream>
#include <fstream>
#include "FileException.h"
using namespace std;
#define NUM_MVP 5

struct PlayInfo {
    string name;
    int nItem;
    double itemPerMove;
    PlayInfo(string na = "신인류", int ni = 0, double ipm = 0.0)
        : name(na), nItem(ni), itemPerMove(ipm) {}
};

class RankingBoard {
    PlayInfo MVP[NUM_MVP];
    int nMVP = NUM_MVP;
public:
    void load(string filename) {
        ifstream is;
        is.open(filename);
        if (!is)
            throw(FileException(filename, true));
        for (int i = 0; i < nMVP; i++)
            is >> MVP[i].nItem >> MVP[i].name >> MVP[i].itemPerMove;
    }
};

```



```

        is.close();
    }
    void store(string filename) {
        ofstream os;
        os.open(filename);
        if (!os)
            throw(FileException(filename, false));
        for (int i = 0; i < nMVP; i++)
            os << MVP[i].nItem << " " << MVP[i].name << " "
                << MVP[i].itemPerMove << "\n";
        os.close();
    }
    void print(string title = "게임 랭킹") {
        cout << endl << title << endl;
        for (int i = 0; i < nMVP; i++)
            cout << "[" << i + 1 << "위]" << MVP[i].name << "Wt"
                << MVP[i].nItem << " " << MVP[i].itemPerMove << "\n";
        cout << "엔터를 입력하세요.";
        getchar();
        cout << endl;
    }
    int add(int nItem, double ipm) {
        if (nItem <= MVP[nMVP - 1].nItem)return 0;

        int pos = nMVP - 1;
        for (; pos > 0; pos--) {
            if (nItem <= MVP[pos - 1].nItem)break;
            MVP[pos] = MVP[pos - 1];
        }
        MVP[pos].nItem = nItem;
        MVP[pos].itemPerMove = ipm;
        cout << "\n[" << pos + 1 << "위] 이름을 입력하세요: ";
        cin >> MVP[pos].name;
        return pos + 1;
    }
};

```

### 3. 실행 결과

스토리 관련 규칙
<p>C:\Users\이우진\Desktop\2학년 1학기\WC++\실습\이우진_텀프로젝트\Debug\이우진_텀프로젝트.exe</p> <p>&lt;스토리 관련 규칙&gt;          우선 이 게임은 스토리가 처음에 나옵니다.          그 스토리는 한 줄씩 나오게 되며 아무키나 누르면 스토리가 다음 줄로 넘어갑니다.          스토리는 픽션이며 재미있게 읽어주시면 감사하겠습니다.          또한 스토리에 나오는 장소는 가상의 장소이므로 현실과는 관계가 없습니다.</p>
게임 관련 규칙
<p>C:\Users\이우진\Desktop\2학년 1학기\WC++\실습\이우진_텀프로젝트\Debug\이우진_텀프로젝트.exe</p> <p>&lt;게임 관련 공지사항&gt;          이 게임은 우선 난이도가 '上', '中', '下'로 나누어져있습니다.          처음에 행킹을 알아본 후 난이도를 선택할 수 있으며          난이도 기준은 다음페이지에서 알려드리도록 하겠습니다.          게임은 전체 이동횟수가 10000이 되면 자동으로 게임이 끝나게 되며          World 내의 고리의 개수가 0이 될 시 게임은 종료가 됩니다.          게임의 승리 조건은          플레이어가 먹은 고리의 갯수가 256개 이상일 시 게임에서 승리하며          256개보다 적게 모을 시에 게임에서 패배하는 형식입니다.</p>
난이도 소개
<p>C:\Users\이우진\Desktop\2학년 1학기\WC++\실습\이우진_텀프로젝트\Debug\이우진_텀프로젝트.exe</p> <p>&lt;난이도 소개&gt;          우선 난이도 下에 대해 먼저 이야기드리겠습니다.          1. AI가 쉬는 시간이 있습니다.          2. AI의 움직임이 단순하며 AI갯수가 5개입니다.          다음으로 난이도 中에 대해 이야기드리겠습니다.          1. AI가 쉬는 시간이 있습니다. (난이도 下와 동일)          2. 난이도 下와 동일한 AI에 새로운 복잡한 AI가 2개 추가 되었습니다.          다음으로 난이도 上에 대해 이야기드리겠습니다.          1. 난이도 中과 동일하나 플레이어에게도 쉬는 시간이 있습니다.          이상으로 게임에 대한 전반적인 설명을 마치겠습니다. 감사합니다.          아무키나 누르면 게임에 대한 스토리로 넘어갑니다.</p>

## 스토리

C:\Users\W이우진\Desktop\W2학년 1학기WC++\W실습W이우진\_템프로젝트WDebugW이우진\_템프로젝트.exe

때는 2030년 사람들은 AI에 열광을 하기 시작하여 많은 사람들이 AI개발을 하게된다.  
 이 때 병천에 있는 K데에서 하나의 상황식 인공지능(이름은 A)이 탄생하게 된다.  
 사람들의 이목은 모두 K데으로 집중이 되었고, 전세계가 주목을 하게 되었다.  
 예전에도 '알파고'라는 AI 바둑 프로그램이 전세계의 이목을 끌었던 것처럼  
 K데에 'A'도 어떤 역할을 수행할 지 전세계가 집중했다.  
 하지만 이 인공지능이 너무 똑똑했던 것일까?  
 인공지능은 결국 K데에 있는 스마트 팩토리라는 곳에 몰래 서버로 침입을 하여  
 로봇에 자신을 넣었다.  
 그 후 로봇의 몸을 가진 AI는 결국...  
 폭주를 일으켰다.  
 그로부터 30년이 지난 2060년...  
 세계는 AI와의 전쟁을 맞이하게 되는데  
 그때 처음 'A'가 탄생하게 된 K데에서 한가지 프로그램이 개발하게 된다.  
 이 프로그램은 인공지능의 폭주를 잠재울 수 있으며  
 처음 개발된 'A'에 프로그램을 설치하면 'A'뿐만 아니라 'A'에게 탄생된 전세계에 인공지능 또한 폭주를 잠재울 수 있다.  
 그러지 지금 바로 인공지능에 프로그램을 설치하면 되는 것이 아닌가요?  
 하지만 'A'는 매우 똑똑하기에 이 인공지능의 두뇌에 해당하고 있는 컴퓨터가 있는 곳에 접근을 하기 위해서 필요한 어떤 고리들  
 전 세계에 존재하였다.  
 이 고리들은 현재까지 512개가 숨겨져 있으며 이 고리 중에 256개를 손에 넣어야 이 컴퓨터가 있는 장소에 접근할 수 있다.  
 그래서 플레이어(당신)은 인간의 대표로서 인간이 다시 평화를 손에 넣기 위해 고리 256개 이상을 모으는 임무를 받았다.  
 하지만 그때 이 소식이 'A'가 들어버렸고 'A'는 다른 부하 인공지능들에게 명령을 하여 이 고리를 회수하려 한다.  
 과연 플레이어(당신)은 다른 인공지능들보다 더 고리를 많이 차지할 수 있을 것인가..  
 당신의 플레이에 세계의 평화가 달려있다...  
 게임을 시작하려면 아무키나 누르세요..

## 난이도 선택 및 게임 시작

C:\Users\이우진\Desktop\2학년 1학기\WC++\실습\이우진\_템프로젝트\Debug\이우진\_템프로젝트.exe

난이도를 입력하세요

- |    |      |    |   |
|----|------|----|---|
| 1. | 5150 | 55 | 5 |
| 2. | 5150 | 55 | 5 |
| 3. | 5150 | 55 | 5 |

난이도 선택해주세요 : 3

이런 데다, 이 일에서 알파벳 갯수는 총 몇개일까? \*\*\*

[게일 랭킹 : 시작]

[1위]	신인	이민	이민	0	0
[2위]	신인	이민	이민	0	0
[3위]	신인	이민	이민	0	0
[4위]	신인	이민	이민	0	0
[5위]	신인	이민	이민	0	0

에터네얼의 힘을 하세요

## 게임 시작 전 화면

C:\Users\이우진\Desktop\2학년 1학기\C++\실습\이우진\_템프로젝트\Debug\이우진\_템프로젝트.exe

[illegible]

전체는	이동	회수	= 0
남은	아이템	수	= 512
	손	시력	위빙 \$ :0:0
	하이	빅	◆ :0:0
	알	사라	구 ♥ :0:0
	로	로	웃 ↕ :0:0
	세	로	타 나 :0:0
	코	라	니 ± :0:0
	가	지	어 ■ :0:0
	플	레	이 여 ♀ :0:0
엔티를	누는	세	요



#### 4. 특징적인 내용(확장된 부분)

확장된 부분

- (1) 난이도를 플레이어가 선택할 수 있습니다.
- (2) 스토리가 추가되었습니다.
- (3) 맵이 넓어졌습니다. (기존 16\*8 => 32\*16)
- (4) 특정 개수 이상의 고리를 먹어야 게임에 승리를 하여 랭킹을 저장할 수 있게 되었습니다.

#### 5. 교과과정에서 학습한 내용들이 어떻게 반영되어 있는가?

##### 5.1 구현한 클래스 리스트 및 설명

우선 저는 이번 텀프로젝트를 하면서 총 16개의 클래스를 선언하였습니다. 각각의 클래스들에 대해 설명드리겠습니다.

클래스 이름	클래스 설명
AIWorld	난이도 下에 대한 AIWorld를 선언한 클래스입니다. AIWorld의 전체적인 설명은 이번 게임에서 일어날 필드를 만든 클래스이며 게임 내 “고리”라는 아이템은 ◎기호를 이용하였습니다.
AIWorld2	난이도 中에 대한 AIWorld를 선언한 클래스입니다.
AIWorld3	난이도 上에 대한 AIWorld를 선언한 클래스입니다.
Canvas	맵 출력을 위한 가상 화면인 클래스입니다. 맵 출력부의 임의의 위치에 어떤 문자를 출력하고 싶으면 직접 출력하지 않고 캔버스의 해당 위치에 저장하고 이를 캔버스 클래스가 화면에 출력을 하는 역할을 합니다.
Point	좌표를 표현하기 위한 클래스입니다. Point 클래스를 선언함으로써 좌표를 표현할 때 x, y 변수를 각각 사용하는 것 보다 더 편한 방법인 Point 객체 하나를 사용할 수 있게 됩니다.
AI	각각의 AI들의 다음과 같은 속성을 선언한 클래스이며 아래에 나오는 Player 클래스와 각각의 인공지능 클래스들의 부모 클래스입니다. 속성은 이름, 아이콘, 현재 위치, 아이템의 수를 선언합니다.
Siri	AI클래스에게 상속을 받은 클래스이며 현재 위치에서 인접한 8방향으로 무작위로 움직이는 인공지능인 Siri를 선언한 클래스입니다.
Bixby	AI클래스에게 상속을 받은 클래스이며 상하좌우로 인접한 4방향으로 움직이는 인공지능인 Bixby를 선언한 클래스입니다.
Alexa	AI클래스에게 상속을 받은 클래스이며 현재 좌표에 상간없이 다음에 어디든 움직일 수 있는 인공지능인 Alexa를 선언한 클래스입니다.
Watson	AI클래스에게 상속을 받은 클래스이며 좌우로 움직이거나 상하로만 움직이는 인공지능인 Watson을 선언한 클래스입니다.
Cotana	Watson클래스에게 상속을 받은 클래스이며 Watson의 움직임은 가로와 세로가 고정되어있지만 인공지능 Cotana는 시간이 지나면 가로 세로가 바뀌고 다시 바뀌면서 반복하는 움직임을 가

	지며 이를 선언한 클래스입니다.
Gigagenie	Bixby클래스에게 상속을 받은 클래스이며 Bixby의 움직임은 인접한 상하좌우로만 움직이지만 인공지능 Gigagenie는 시간이 지나면 상하좌우에서 인접한 대각선 방향으로 움직이고 다시 돌아오면서 반복하는 움직임을 가지며 이를 선언한 클래스입니다.
Player1	난이도 中, 下에서 플레이어의 움직임을 나타낸 Player 클래스입니다. 방향키로 입력을 받아 플레이어가 움직일 수 있습니다.
Player	난이도 上에서 플레이어의 움직임을 나타낸 Player 클래스입니다.
Matrix	2차원 배열의 동적 할당을 이용하여 int를 저장하는 행렬 클래스입니다.
RankingBoard	랭킹을 관리하는 클래스입니다.

## 5.2 디폴트 매개변수 사용 내용

AI클래스와 AI클래스의 자식 클래스인 각각의 인공지능 클래스, 그리고 Canvas 클래스, Matrix클래스에서 디폴트 매개변수를 사용하였습니다. 각각의 클래스에서 사용된 부분을 가지고 왔습니다.

class AI	<code>string n = "이름없는AI", string i = "※", int x = 0, int y = 0</code>
AIKind.h에 있는 클래스들	<code>Siri =&gt; : string n = "애플", string i = "§", int x = 0, int y = 0</code> <code>Bixby =&gt; : string n = "삼성", string i = "◆", int x = 0, int y = 0</code> <code>Alexa =&gt; : string n = "아마존", string i = "♥", int x = 0, int y = 0</code> <code>Watson =&gt; : string n = "IBM", string i = "↔", int x = 0, int y = 0, bool bH = true</code> <code>Cotana =&gt; : string n = "MS", string i = "±", int x = 0, int y = 0, bool bH = true</code> <code>Gigagenie =&gt; : string n = "KT", string i = "■", int x = 0, int y = 0</code>
class Canvas	<code>int nx = 10, int ny = 10</code>  <code>string val = " "</code>
class Matrix	<code>const char* str = "Mat "</code>  <code>int val = 100</code>

예를 들어 설명을 하면 class AI에 경우 디폴트 매개변수가 사용되면서 매개변수가 없으면 AI의 위치는 (0,0)이 되고 이름은 "이름없는AI", 아이콘은 "※"가 됩니다.

## 5.3 멤버 초기화 리스트를 사용한 곳

AI클래스와 AI클래스의 자식 클래스인 각각의 인공지능 클래스, 그리고 Canvas 클래스에

서 멤버 초기화 리스트를 사용하였습니다. 각각의 클래스에서 사용된 부분을 가지고 왔습니다.

class AI	:name(n), icon(i), nItem(0), p(x, y), q(x, y), nSleep(0), dist(0.0), total(0.0) {}
AIKind.h에 있는 클래스들	Siri => : AI(n, i, x, y) {} Bixby => : AI(n, i, x, y) {} Alexa => : AI(n, i, x, y) {} Watson => : AI(n, i, x, y), bHori(bH) {} Cotana => : Watson(n, i, x, y, bH), count(30) {} Gigagenie => : Bixby(n, i, x, y) {}
class Canvas	: xMax(nx), yMax(ny)

#### 5.4 하나 이상의 참조자 반환 함수 및 용도

예를 들어 class AI에서 사용이된 참조자 반환함수에 대해 이야기를 하겠습니다. 다음 참조자 함수가 어떻게 이용이 되었는지 관련 코드를 가지고 와서 확인해보겠습니다.

class AI	<pre>bool isSleep() {     if (nSleep &gt; 0) {         nSleep--;         return true;     }     else return false; }</pre>
----------	--

다음은 현재 쉬고 있는 상태인지를 검사하는 함수이다. 만약 쉬고 있으면 nSleep는 1이 줄어들고 참조자 반환 함수를 이용하여 true로 반환한다. 그리고 nSleep이 0이 되면 false가 반환이 된다. true로 반환이 된 것은 계속 도는 것이고 false가 반환이 되면 쉬고 있지 않다는 것이 된다,

#### 5.5 클래스 중에서 동적 할당을 사용한 부분 및 사용 이유

이 게임에서 동적 할당을 사용한 부분에 대해서 먼저 이야기를 하겠습니다. 총 8개의 클래스에 동적 할당을 하였습니다.


Game.cpp	<pre>game.add(new Siri("손이시려워", "§", rand() % w, rand() % h)); game.add(new Bixby("하이빅스비", "◆", rand() % w, rand() % h)); game.add(new Alexa("알렉사라구", "♥", rand() % w, rand() % h)); game.add(new Watson("가로로왓슨", "↔", rand() % w, rand() % h, true)); game.add(new Watson("세로로왓슨", "↑", rand() % w, rand() % h, false));</pre>
----------	---

	<pre> game.add(new Cotana("코타나래요", "±", rand() % w, rand() % h)); game.add(new GigaGenie("기가지니아", "■", rand() % w, rand() % h)); </pre>
AIKinds.h	<pre> 각각의 클래스 들의 int** map mat = new int* [rows]; for (int i = 0; i &lt; rows; i++)     mat[i] = new int[cols]; =&gt; 동적 메모리 할당 class Matrix for (int i = 0; i &lt; rows; i++)     delete[] mat[i];     delete[] mat; =&gt; 동적 메모리 해제 </pre>

동적 할당을 사용한 이유는 AI들이 활동하는 맵이 변할 수도 있기에 이를 해결하고자 동적할당을 이용하였습니다. 또한 Matrix클래스는 동적할당을 이용하여 만들어진 행렬 클래스입니다. 따라서 동적 할당을 이용하였기에 AIWorld의 크기를 조절할 수 있고 AI의 수를 조절할 수 있게 되었습니다.

## 5.6 복사 생성자 구현 내용 또는 복사 생성을 피하기 위해 코드에서 채택한 방법

템프로젝트 내에서 복사 생성자가 어디에 이용이 되었냐면 class Point에서 연산자 중복에서 `void operator+= (const Point& p) { x += p.x, y += p.y; }` 으로 이용이 되었습니다. 복사 생성자를 이용하였기에 다음 줄은 const를 붙이지 않았습니. 왜 복사생성을 하였냐면 만약 const가 없을 시에 다음과 같은 오류가 발생합니다.

 <a href="#">C2679</a>	이항 '+=': 오른쪽 피연산자로 'Point' 형식을 사용하는 연산자가 없거나 허용되는 변환이 없습니다.	이우진_템프로젝트	AI.h	50
---	---	-----------	------	----

그리고 복사 생성자는 대입연산을 할 때 실행이 되기에 사용을 하였습니다.

## 5.7 this 포인터 사용한 부분 및 사용 이유

this 포인터는 객체 자신의 메모리상의 주소를 나타내는 포인터입니다. 템프로젝트 내의 this 포인터는 그렇게 많이 사용은 되지 않았지만 중요한 역할을 할 때 사용을 하였습니다.

class AIWorld	<pre> ((Siri*)pMon[0])-&gt;move(world.Data(), xMax, yMax); ((Bixby*)pMon[1])-&gt;move(world.Data(), xMax, yMax); ((Alexa*)pMon[2])-&gt;move(world.Data(), xMax, yMax); ((Watson*)pMon[3])-&gt;move(world.Data(), xMax, yMax); ((Watson*)pMon[4])-&gt;move(world.Data(), xMax, yMax); ((Player1*)pMon[5])-&gt;move(world.Data(), </pre>
---------------	--



	<pre> xMax, yMax);  ((Cotana*)pMon[6])-&gt;move(world.Data(), xMax, yMax);  ((GigaGenie*)pMon[7])-&gt;move(world.Data(), xMax, yMax); </pre>
Game.cpp	<pre> if (player1-&gt;nItem &gt; 256)  rank.add(player1-&gt;nItem, player1-&gt;nItem / player1-&gt;total); </pre>

우선 class AIWorld에서 사용이 된 이유는 멤버 함수를 호출한 객체 자신의 주소를 화면에 출력하기 위해 사용이 되었습니다. 그리고 Game.cpp에서는 if문을 이용해서 nitem의 개수가 256이상일 때라는 형식으로 작성을 하였는데 이 또한 player1의 nitem 개수를 호출하기 위해 사용이 되었습니다.

## 5.8 클래스의 정적 멤버(멤버 변수와 멤버 함수) 사용 부분 및 사용 이유

정적 멤버를 사용하지 않았습니다.

## 5.9 클래스 상속, 상속된 자식 클래스에 대한 특징

클래스 상속은 AI클래스에서 상속 받은 클래스들이 4개, AI클래스에게 상속 받은 자식클래스에게 상속받은 클래스들이 2개가 있습니다. 우선적으로 클래스가 어떤 클래스에게 상속을 받았는지 이야기 하고 자식 클래스에 대한 특징에 대해 이야기하겠습니다.

class Siri	Siri 클래스는 AI 클래스에게 상속을 받았습니다. Siri클래스는 AI 클래스의 움직임을 그대로 사용하였습니다. 또한 AI를 캔버스에 그리는 함수, AI의 정보를 화면에 출력함수를 그대로 물려받았습니다.
class Bixby	Bixby 클래스는 AI 클래스에게 상속을 받았습니다. AI를 캔버스에 그리는 함수, AI의 정보를 화면에 출력함수를 그대로 물려받았습니다. 그 후 새로운 움직임을 클래스에서 선언을 하였습니다.
class Alexa	Alexa 클래스는 AI 클래스에게 상속을 받았습니다. AI를 캔버스에 그리는 함수, AI의 정보를 화면에 출력함수를 그대로 물려받았습니다. 그 후 새로운 움직임을 클래스에서 선언을 하였습니다.
class Watson	Watson 클래스는 AI 클래스에게 상속을 받았습니다. AI를 캔버스에 그리는 함수, AI의 정보를 화면에 출력함수를 그대로 물려받았습니다. 그 후 새로운 움직임을 클래스에서 선언을 하였습니다.
class Cotana	Cotana 클래스는 Watson 클래스에게 상속을 받았습니다. AI를 캔버스에 그리는 함수, AI의 정보를 화면에 출력함수를 그대로 물려받았습니다. 그 후 새로운 움직임을 클래스에서 선언을 하였습니다.
class GigaGenie	GigaGenie 클래스는 Bixby 클래스에게 상속을 받았습니다. AI를 캔버스에 그리는 함수, AI의 정보를 화면에 출력함수를 그대로 물려받았습니다. 그 후 새로운 움직임을 클래스에서 선언을 하였습니다.

## 5.10 재 정의가 필요한 멤버 함수들에 대한 설명

재 정의가 필요한 멤버 함수는 AI클래스의 자식 클래스의 move함수입니다. 그 이유는 AI

클래스의 움직임을 그대로 이용하려면 상관이 없지만 새로운 움직임을 하기 위해서 move를 바꿔야하기에 move함수가 재 정의가 필요합니다. 따라서 다음 코드는 각각의 AI의 자식 클래스들의 move함수 재 정의한 것입니다.

Bixby	<pre> void move(int** map, int maxx, int maxy) {     if (!isSleep()) {         int dir = rand() % 4;         if (dir == 0) p[0]--;         else if (dir == 1) p[0]++;         else if (dir == 2) p[1]--;         else p[1]++;         clip(maxx, maxy);         eat(map);     } } </pre>
Alexa	<pre> void move(int** map, int maxx, int maxy) {     if (!isSleep()) {         p = Point(rand() % maxx, rand() % maxy);         clip(maxx, maxy);         eat(map);     } } </pre>
Watson	<pre> void move(int** map, int maxx, int maxy) {     if (!isSleep()) {         int dir = rand() % 2;         int jump = rand() % 2 + 1;         if (bHori)             p[0] += ((dir == 0) ? -jump : jump);         else             p[1] += ((dir == 0) ? -jump : jump);          clip(maxx, maxy);         eat(map);     } } </pre>
Cotana	<pre> void move(int** map, int maxx, int maxy) {     if (!isSleep()) {         Watson::move(map, maxx, maxy);         count--;         if (count == 0) {             bHori = !bHori; </pre>

	<pre> count = 30;  } clip(maxx, maxy); eat(map); } } </pre>
GigaGenie	<pre> void move(int** map, int maxx, int maxy) {     if (!isSleep()) {         if (count2 &gt; 30 &amp;&amp; count2 &lt;= 60)         {             Bixby::move(map, maxx, maxy);             count2--;         }         else if (count2 &lt;= 30) {             switch (rand() % 4) {                 case 0: p[0]++; p[1]--; break;                 case 1: p[0]++; p[1]++; break;                 case 2: p[0]--; p[1]++; break;                 case 3: p[0]--; p[1]--; break;             }             count2--;             if (count2 == 0) {                 count2 = 60;             }         }         clip(maxx, maxy);         eat(map);     } } } </pre>

Cotana와 GigaGenie는 각각 AI의 자식 클래스들에게 상속을 받았고, 또한 움직임을 또 받기 위해서 재 정의를 하면서 멤버함수를 가지고 왔습니다.

### 5.11 가상함수 처리한 부분들 및 필요한 이유 및 동작 설명

우선 가상함수를 이용한 부분을 먼저 알려드리겠습니다.

class AI	<pre> virtual ~AI() {     cout &lt;&lt; icon &lt;&lt; nItem &lt;&lt; "  거리: " &lt;&lt; total &lt;&lt; endl; }  virtual void move(int** map, int maxx, int maxy) </pre>
----------	--

왜 move라는 멤버 함수와 ~AI에 가상함수를 처리한 것일까요? 이유는 AI에게 상속받은

클래스에서 move라는 멤버 함수가 재정의 되기 위해서는 가상함수로 선언이 되어야 하기 때문입니다. 따라서 A클래스에서 멤버 함수인 move가 가상 함수로 선언이 되어 자식 클래스에서 재정의하는 과정을 가집니다. ~A에 가상함수를 처리한 이유는 각각 자식 클래스에서도 새로운 이름으로 출력을 하기 위해서입니다.

### 5.12 상속 관계에서 생성자/소멸자 호출과 관련된 특징적인 부분 설명

상속 관계에서 생성자/소멸자 호출은 멤버 함수가 private에 있지 않는 한 자식 클래스에서 부모클래스의 멤버 함수를 호출 할 수 있습니다. 하지만 만약 private에 있는 함수를 사용하고 싶으면 어떻게 해야할까요? 그 함수를 protected로 바꾸거나 this 포인터로 이용하여 사용을 할 수 있습니다. 하지만 this 포인터를 사용하는 경우에는 생성자의 호출방법을 제안하고자 할 때 쓰이게 됩니다.

### 5.13 연산자 중복을 사용한 부분들 설명

class Point에서 다음과 같은 연산자 중복을 사용했습니다. 다음 코드들을 보면서 어떤 연산자 중복을 사용하였는지 이야기하겠습니다.

class Point	<pre> int&amp; operator[] (int id) {     if (id == 0)return x;     else if (id == 1)return y;     else exit(0); } =&gt; 인덱스 연산자가 사용이 되었습니다. operator double() { return sqrt((double)x * x + y * y); } =&gt; 형 변환 연산자가 사용이 되었습니다. void operator()(int maxx, int maxy) {     if (x &lt; 0)x = 0;     if (x &gt;= maxx)x = maxx - 1;     if (y &lt; 0)y = 0;     if (y &gt;= maxy)y = maxy - 1; } =&gt; 함수 호출 연산자 중복 함수가 사용이 되었습니다. Point operator- () { return Point(-x, -y); } =&gt; 단항 연산자가 사용이 되었습니다. bool operator== (Point&amp; p) { return x == p.x &amp;&amp; y == p.y; } =&gt; 비교 연산자가 사용이 되었습니다. bool operator!= (Point&amp; p) { return x != p.x    y != p.y; } =&gt; 비교 연산자가 사용이 되었습니다. Point operator- (Point&amp; p) { return Point(x - p.x, y - p.y); } =&gt; 이진 산술 연산자가 사용이 되었습니다. Point operator+ (Point&amp; p) { return Point(x + p.x, y + </pre>
-------------	---

	<pre>p.y); }</pre> <p>=&gt; 이진 산술 연산자가 사용이 되었습니다.</p> <pre>void operator+= (const Point&amp; p) { x += p.x, y += p.y; }</pre> <p>=&gt; 복합 할당 연산자가 사용이 되었습니다.</p> <pre>void operator-= (Point&amp; p) { x -= p.x, y -= p.y; }</pre> <p>=&gt; 복합 할당 연산자가 사용이 되었습니다.</p>
--	---

#### 5.14 friend를 사용한 부분 설명

class Point에서 다음과 같은 friend 함수를 사용하였습니다.

class Point	<pre>class Point {     int x, y;     friend class AI;     friend class Canvas; public:</pre>
-------------	--

friend 함수를 이용하여 class AI와 class Canvas는 class Point에 모든 멤버를 사용할 수 있게 됩니다.

#### 5.15 예외 처리를 하고, 예외의 발생 부분과 처리 부분을 나누어야 하는 이유를 설명

우선 예외 처리한 부분은 RankingBoard.h에서 예외가 발생하여 던져서 Game.cpp에서 처리하였습니다.

RankingBoard.h	<pre>throw(FileException(filename, true)); throw(FileException(filename, false));</pre>
Game.cpp	<pre>try {     rank.load("AIWorld1.rnk"); } catch (FileException e) {     char str[80];     string passwd, correct = "22";     cout &lt;&lt; "AI의 풀네임에서 알파벳 갯수는 총     몇개일까?: ";      for (int i = 0;; i++) {         str[i] = _getch();         putchar('*');         if (str[i] == 'Wr') {             str[i] = 'WO';             passwd = str;             cout &lt;&lt; "Wn";             break;         }     } }</pre>

	<pre> if (passwd != correct) {     cout &lt;&lt; "비밀번호가 맞지 않습니 다. 게임 종료.WnWn";      exit(0); } </pre>
--	---

생성자와 소멸자에서 같이 발생하면 오류가 발생할 수 있기 때문에 나눠서 사용을 합니다.

## 6. 향후 더 확장할 수 있는 방법

- (1) 우선 이 게임에는 플레이어와 AI가 부딪혔을 때 아무 일도 일어나지 않고 그냥 자신의 할 일을 합니다. 그러므로 플레이어와 AI끼리 부딪히면 자신이 모으고 있던 고리의 개수가 줄어드는 내용이 확장이 될 수 있습니다.
- (2) 난이도 상기준으로 현재 플레이어는 AI들과 동일한 쉬는 시간으로 게임이 플레이 됩니다. 그러므로 플레이어의 레벨을 추가해 플레이어가 레벨이 올라가면 올라갈수록 쉬는 시간이 줄어드는 내용이 확장이 될 수 있습니다.

## 7. 고찰

이번 텀프로젝트를 하면서 많이 몰랐던 부분들을 많이 배워가는 시간이었습니다. 처음에 c++을 시작할 때는 열심히 배우려고 하였습니다. 하지만 계속 진도를 나가면서 저 자신이 진도를 따라가지 못하는 상황이 일어났습니다. 그래서 저는 과제도 지각제출을 하게 되었고, 남들보다 c++에 대해 이해하는 과정이 늦었기에 최대한 열심히 따라가려고 하였습니다. 코로나 19로 인해 온라인으로 강의를 하면서 저번에 들었던 강의를 들을 수 있다는 것은 진도가 늦춰진 저에게 조금이나마 희망을 가지고 c++에 대해 조금씩 알아갔던 것 같습니다. 1학기가 이제 끝나가지만 아직 저는 c++에 대해 많이 부족하다는 것을 느끼고 있습니다. 그래서 미래를 위해서라도 여름방학 때 다시한번 복습을 하는 과정을 밟아야겠다고 생각을 하였습니다.

이번 텀프로젝트는 교재의 MonsterWorld를 참고하여 유사하거나 확장된 형태의 게임을 작성하는 것이 주제였습니다. 저는 이번 과제를 하면서 처음에는 새로운 코드들로 텀프로젝트를 하려고 하였습니다. 하지만 저의 실력이 아직 많이 부족하다는 것을 깨닫고 교재의 코드를 이용하여 확장하는 형태로 텀프로젝트를 진행하기로 하였습니다. 보고서에 작성한 대로 향후 더 발전할 수 있는 방법은 무궁무진하기에 여름방학 때 공부를 하면서 확장을 한번 해보는 것을 도전해야겠다고 마음을 먹게 되었습니다.