2016182041 조영환

CPU: 라이젠 3600XT

메모리 16GB

성능표

|  |  |  |
| --- | --- | --- |
| 멀티 스레드 | 멀티 스레드(시야처리) | 멀티 스레드(시야+섹터링) |
|  |  |  |

Do\_move 에서 추가

// 섹터1만 검사

if (players[p\_id].x < 195 && players[p\_id].y < 195)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터2만 검사

else if (players[p\_id].x > 205 && players[p\_id].y < 195)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터3만 검사

else if (players[p\_id].x > 205 && players[p\_id].y > 205)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터4만 검사

else if (players[p\_id].x < 195 && players[p\_id].y >205)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터 1,2 검사

else if (players[p\_id].x >= 195 && players[p\_id].x <= 205 && players[p\_id].y < 195)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터 3,4 검사

else if (players[p\_id].x >= 195 && players[p\_id].x <= 205 && players[p\_id].y > 205)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터 1,4 검사

else if (players[p\_id].x < 195 && players[p\_id].y >= 195 && players[p\_id].y <= 205)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터 2,3 검사

else if (players[p\_id].x > 205 && players[p\_id].y >= 195 && players[p\_id].y <= 205)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

// 섹터 1,2,3,4 검사

else if (players[p\_id].x >= 195 && players[p\_id].x <= 205 && players[p\_id].y >= 205 && players[p\_id].y <= 205)

{

for (auto& pl : players) {

if (pl.id == p\_id)continue;

//if (pl.id != p\_id) break;

if ((pl.m\_state == PLST\_INGAME) && can\_see(p\_id, pl.id)) {

new\_vl.insert(pl.id);

}

}

}

send\_move\_packet(p\_id, p\_id);

섹터링을 4분활 하였습니다.

|  |  |
| --- | --- |
| S1 | S2 |
| S3 | S4 |

그후 겹쳐 있는 공간까지 포함하여 총 9개 구역을 검사하는 공간을 만들었습니다.

|  |  |  |
| --- | --- | --- |
| S1검사 | S1,S2검사 | S2검사 |
| S1,S3검사 | 모두검사 | S2,S4검사 |
| S3검사 | S3,S4검사 | S4검사 |

S1~S4까지 모두 검사하는 범위는 X,Y가 195~205사이를 검사하고 플레이어 위치가 임의로 나눈 범위에 있으면 그 범위안에 시야가 보여지도록 하였습니다.