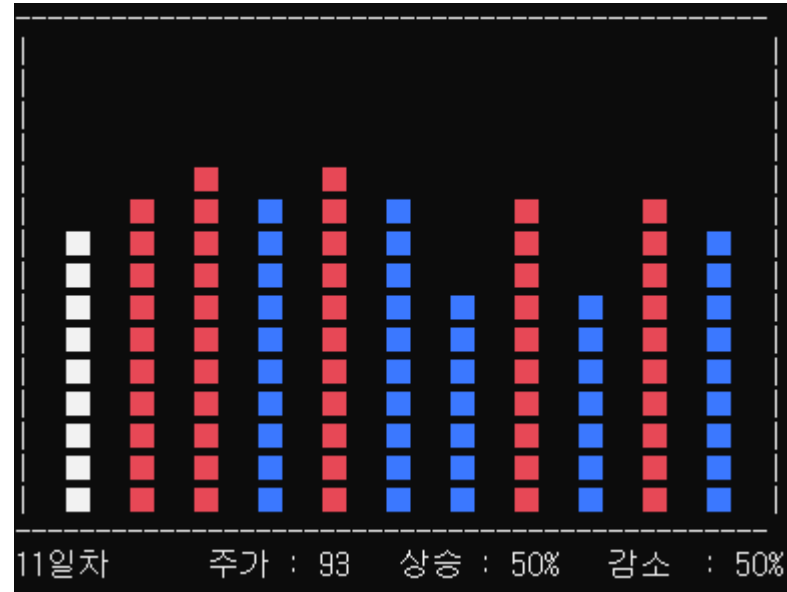


주식 투기 게임



역할분담

이원종 : 주식 관련 함수 담당
전기범 : 전체 시스템 및 디버그
이재주 : 그래픽 구현 담당
김영주 : 결과값 계산, 도트찍기

```

#pragma once
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <ctime>
#include <windows.h>
#include <iomanip>

#define START 90 // 시작점
#define PERSTART 50
#define DATE 11
#define STARTMONEY 5000

using namespace std;

class Datas {
private:
    int per; //확률
    int    updown; //주가증감
    int    price[DATE]; //역대주가

    int money; //자금
    int stock; //소지 주
    int total; //총 재산

    int day;

```

헤더파일 data.h

```

public:
    void setper(int per);
    void setupdown(int updown);
    void setprice();
    void setprice(int price);
    void setmoney(int money);
    void setstock(int stock);
    void settotal(int total);
    void setday(int day);
    void setgetprice(int allprice[]);

    int getprice();
    int getper();
    int getupdown();
    int getmoney();
    int getstock();
    int gettotal();
    int getday();

};

```

```
#include "data.h"
```

소스파일 data.cpp

```
void Datas::setper(int per) { this->per = per; }
void Datas::setupdown(int updown) { this->updown = updown; }
void Datas::setprice() { price[0] = START; }
void Datas::setprice(int price) { this->price[day + 1] = price; }
void Datas::setmoney(int money) { this->money = money; }
void Datas::setstock(int stock) { this->stock = stock; }
void Datas::setday(int day) { this->day = day; }
void Datas::settotal(int total) { this->total = total; }
void Datas::setgetprice(int allprice[])
{
    for (int i = 0; i < 11; i++)
        allprice[i] = price[i];
}

int Datas::getper() { return per; }
int Datas::getupdown() { return updown; }
int Datas::getprice() { return price[day]; }
int Datas::getmoney() { return money; }
int Datas::getstock() { return stock; }
int Datas::gettotal() { return total; }
int Datas::getday() { return day; }
```

소스파일 func.cpp

```
#include "data.h"

extern int player, buy[], key;
extern Datas dt[4];

void Rand_Graph() // 주식 랜덤 함수

void color_check() //

void draw_in() //

void ui() //그래픽 구현

void player_data() //

void ui_sec() //

void typing() //키보드 조작 인식

void playerdata() //유저 정보 저장

void final() //최종 결과 계산

void gameover() //우승자 그래픽
```

소스파일 main.h - 전체

```
#include "data.h"

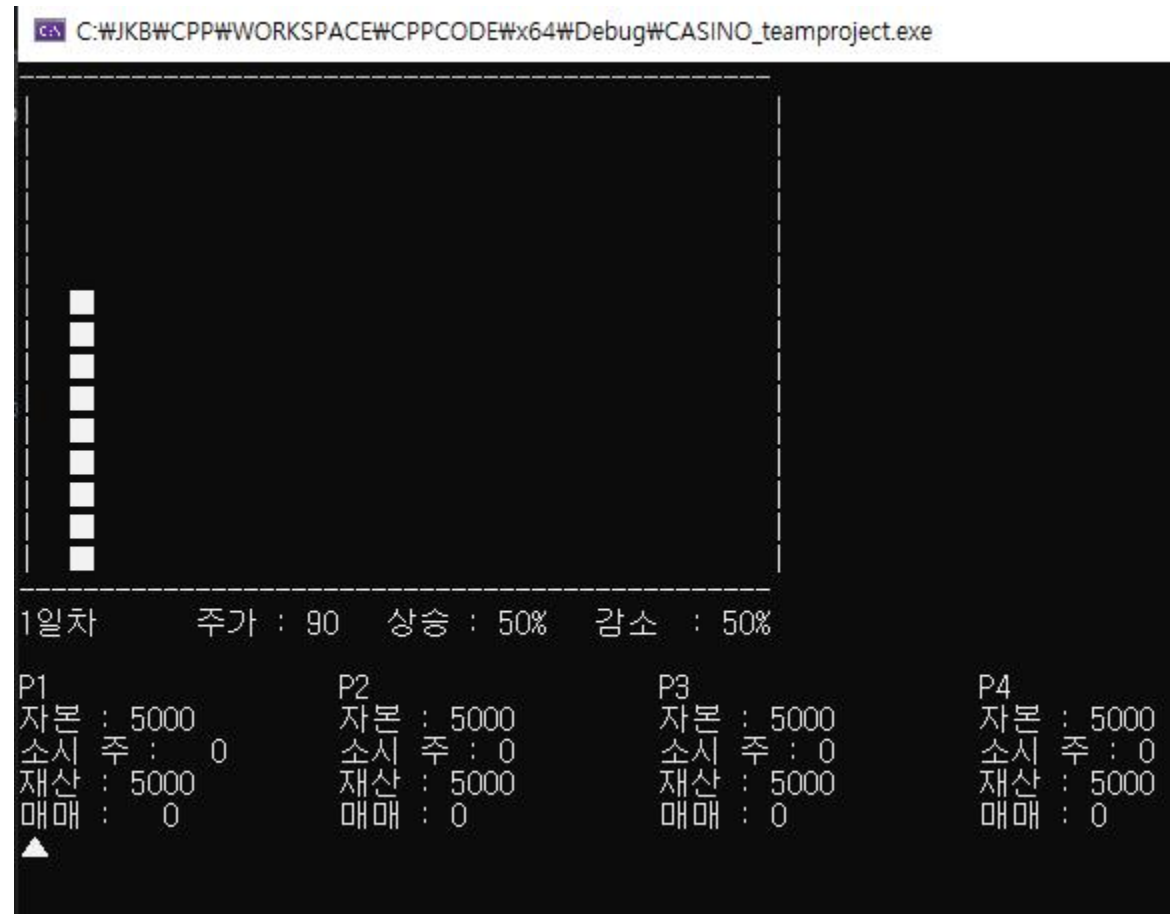
int player = 1, buy[5], key = 0;
Datas dt[5]; //dt[0] = 시스템, dt[1] = p1, dt[2] = p2, dt[3] = p3, dt[4] = p4

extern void Rand_Graph(); extern void ui();
extern void typing(); extern void playerdata();
extern int final(); extern void gameover(int);
extern void ui_sec();

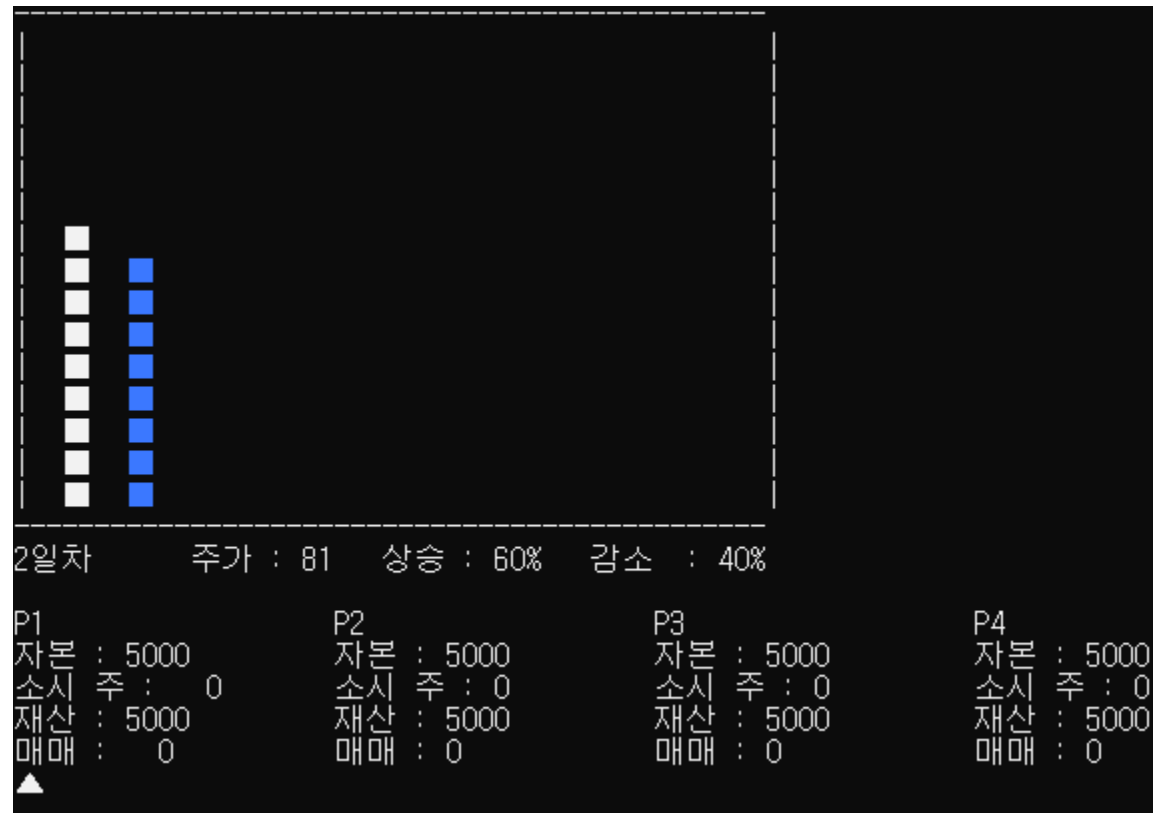
int main()
{
    for (int i = 1; i <= 4; i++)
        dt[i].setmoney(STARTMONEY);
    dt[0].setprice();
    dt[0].setper(PERSTART);
    playerdata();

    while (dt[0].getday() < 10)
    {
        system("cls");
        ui();
        ui_sec();
        typing();
        Rand_Graph();
        playerdata();
        dt[0].setday(dt[0].getday() + 1); //dt[0] 1 dt[10] 11
        for (int i = 1; i <= 4; i++)
            dt[i].settotal(dt[i].getmoney() + dt[i].getstock() * dt[0].getprice());
    }
    system("cls");
    ui();
    ui_sec();
    gameover(final());
}
```

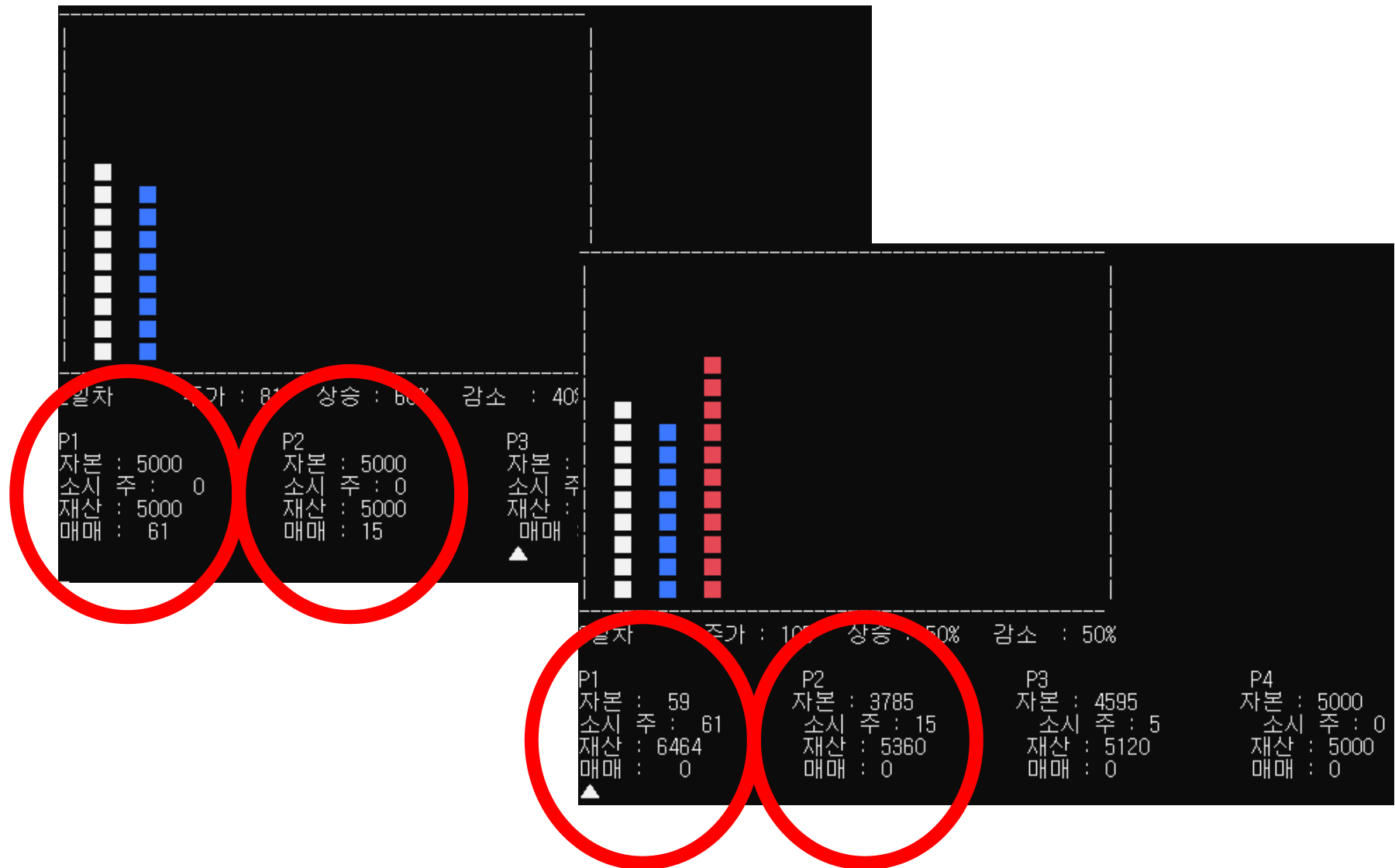
기능 - 초기화면



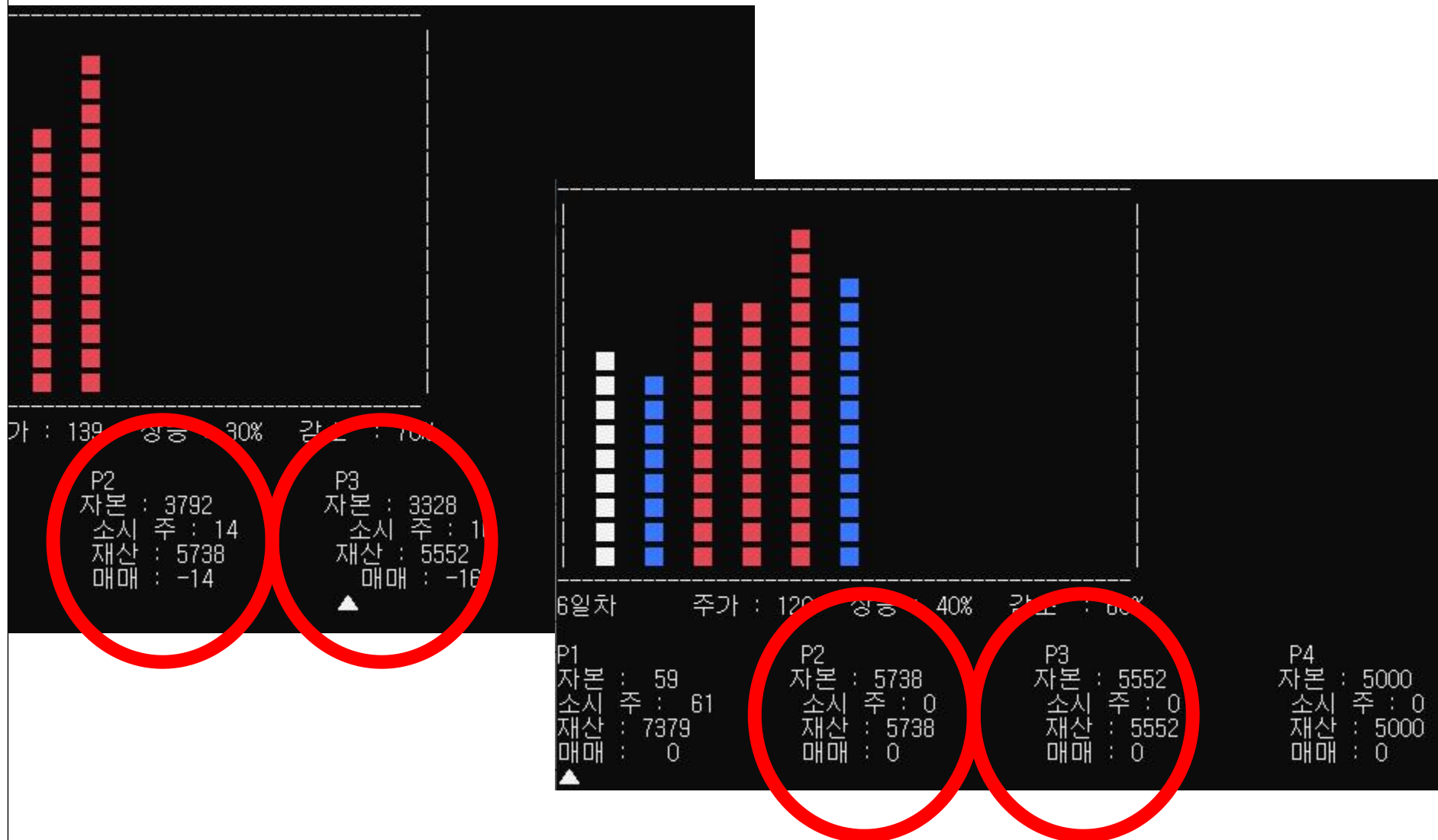
기능 - 그래프 진행



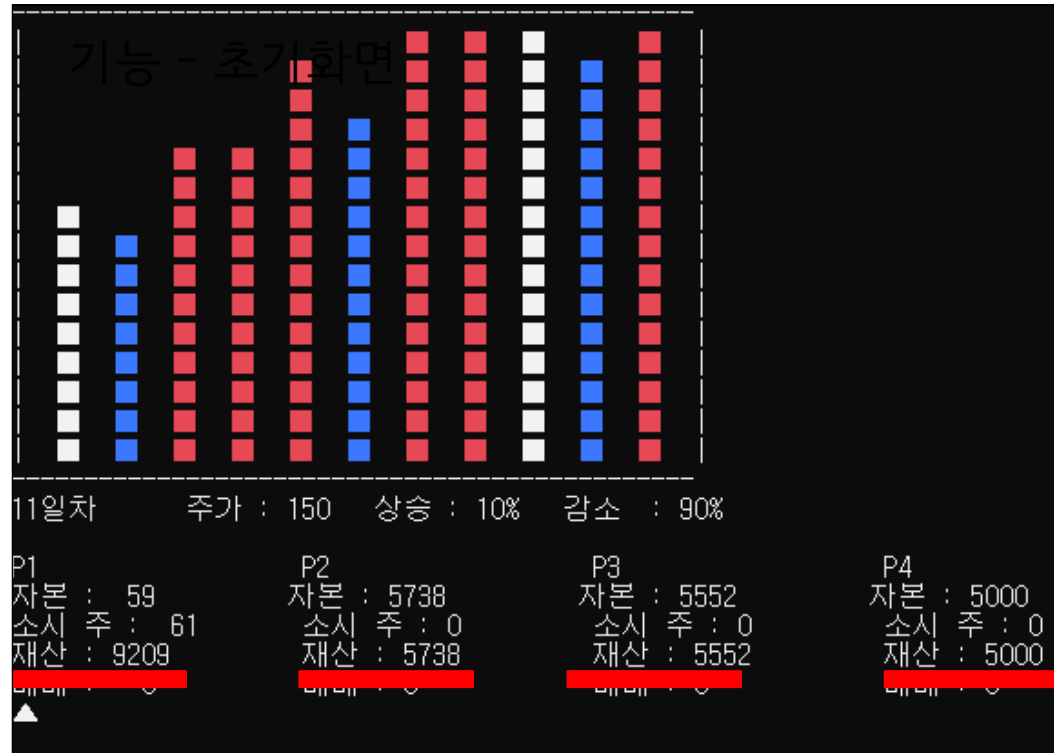
기능 - 가즈아 (구매)



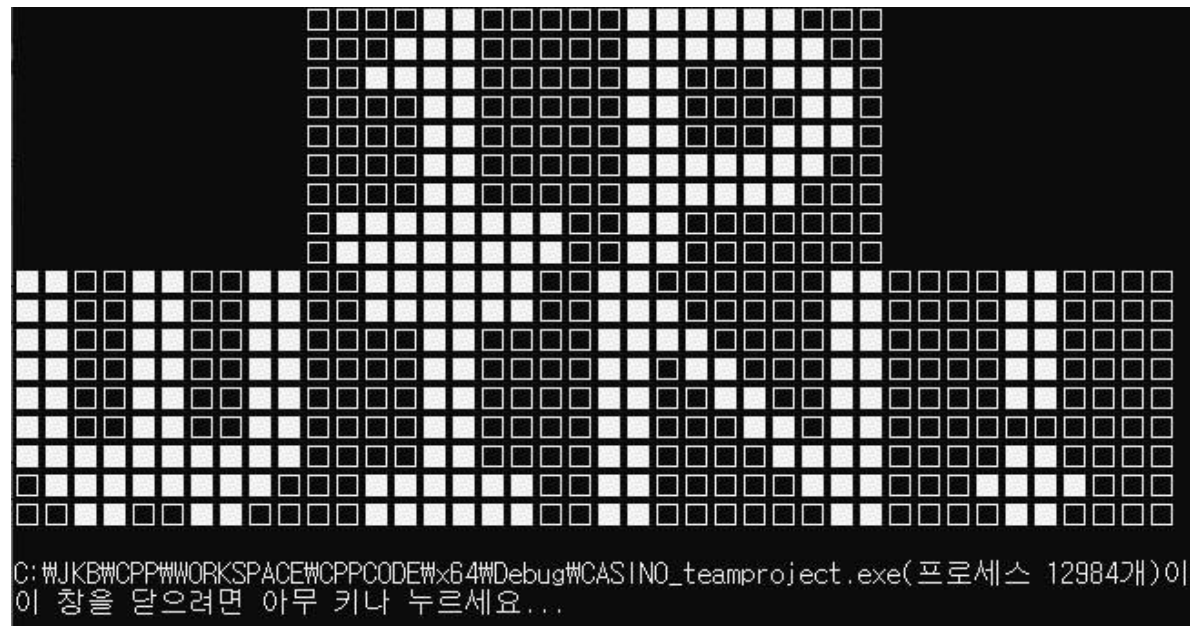
기능 - 손절 (판매)



기능 - 초기화면



기능 - 결과



힘들었던점

- 진행도중 필요함수가 더 발생
- 함수가 많아 디버그 시간낭비
- 각자 코드를 작성시 지역변수를 사용하여 클래스 변수로 합칠때 애로사항 발생