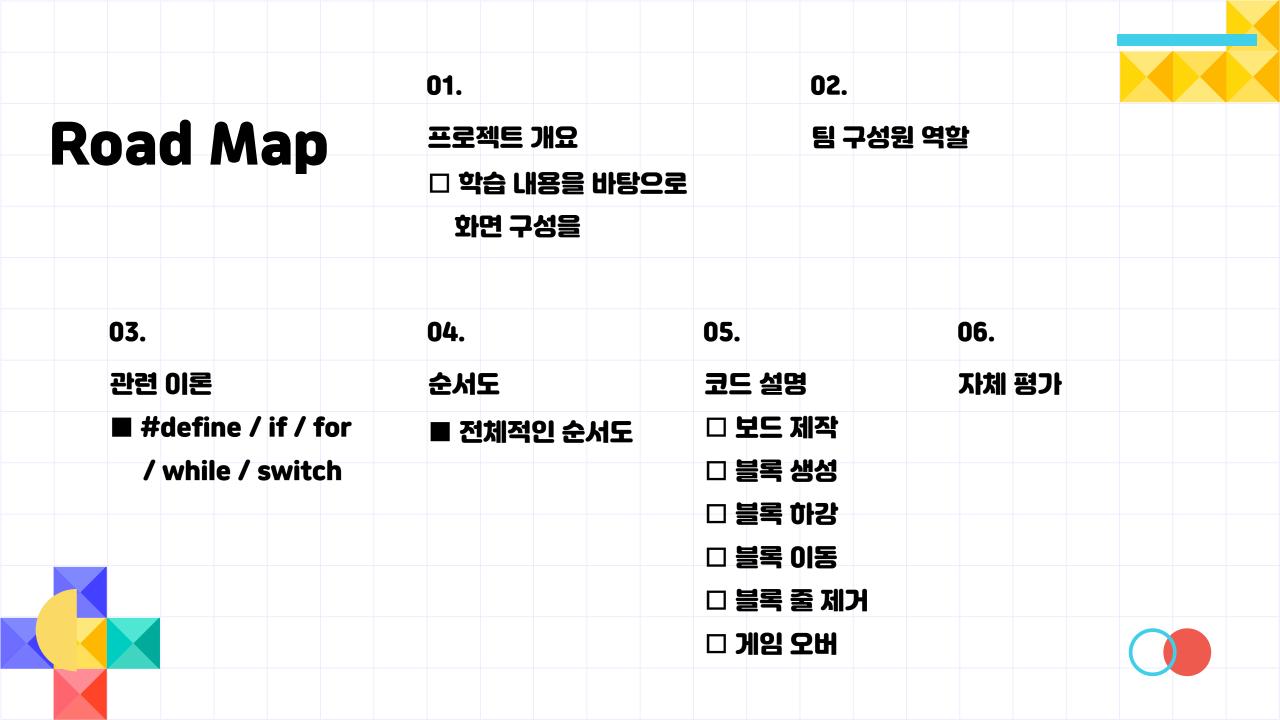


1조 안서준 / 오세룡 / 이범식 / 조영탁





프로젝트 개요

- #define을 이용하여 반복적인 값 입력을 줄 일 수 있다
- For 문을 이용해서 반복 출력, 연산이 가능하다
- If 문을 이용하여 조건을 걸어서 경우의 수를 만들 수 있다
- Switch 문을 이용하여 경우에 맞는 값 출력을 할 수 있다
- 배열을 이용하여 값을 저장하고 출력을 할 수 있다
- While 문을 이용하여 반복작업을 계속 호출해서 사용한다
- 이를 통해서 키보드의 입력과 CMD창에서 위치만
 알 수 있다면 테트리스를 만들 수 있다

* * * TETRIS * * *

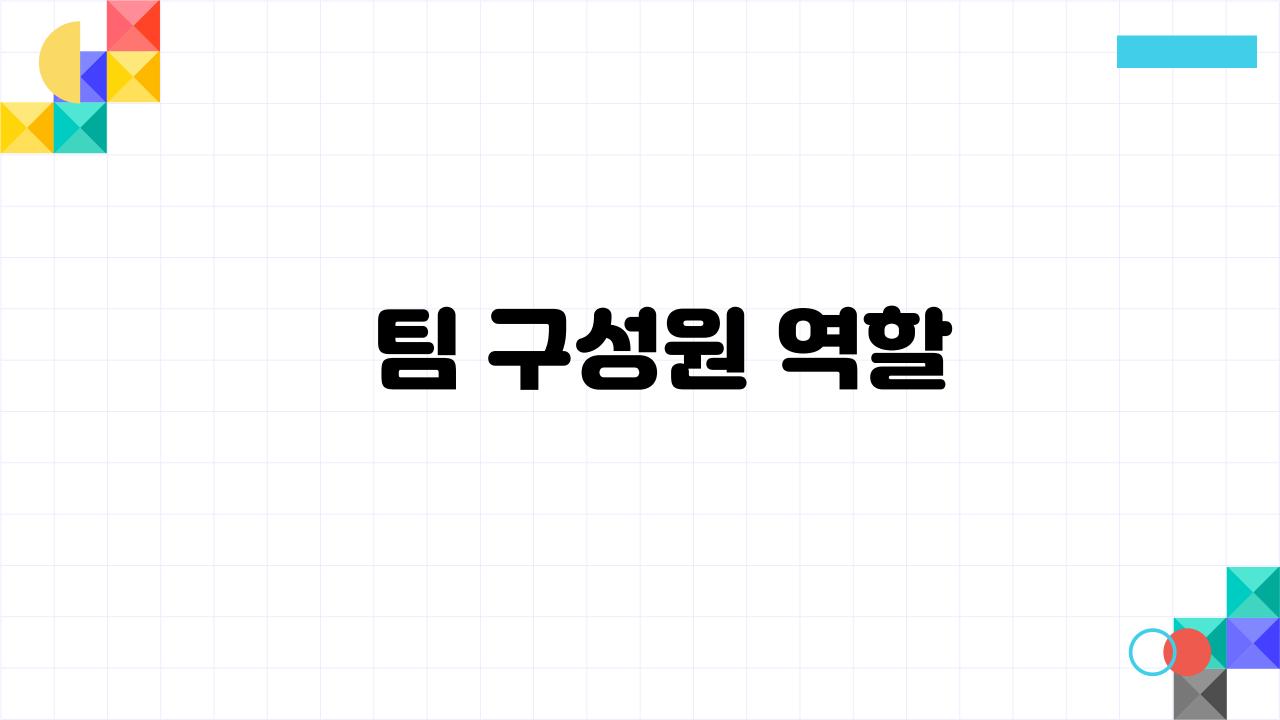
Space : 날취

C : 음악 On/Off P : 일시정지 ESC : 게임 종료

☆ Press Enter ☆

최고점: 1900

C드라이브에 tetris.wav가 있다면 C 입력



역할

안서준

- · 게임보드, 임시보드, 추가보드
- · 오버라인까지 차면 게임오버
- · 추가보드에 점수 및 삭제한 줄 수 출력
- ・ 보드 최신화 및 출력

오세룡

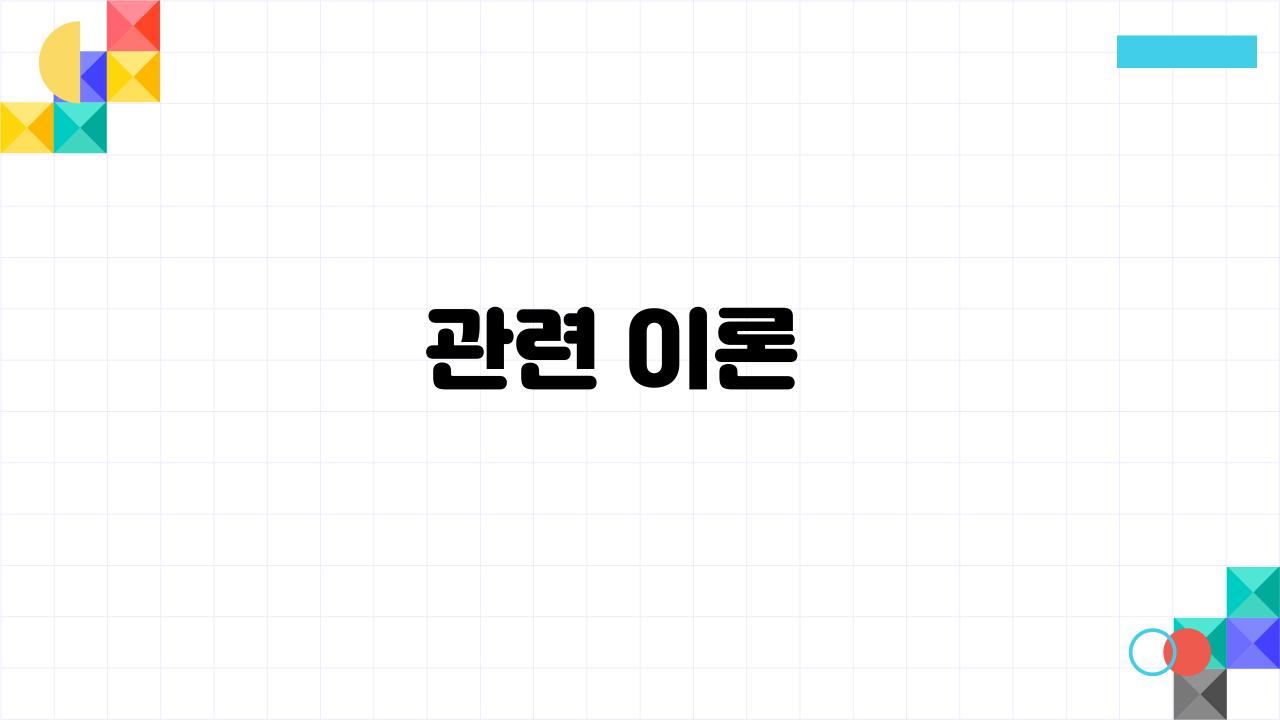
- ・ 블록 생성 및 색상
- 블록 회전
- ・ 블록 충돌 체크
- 일정 시간마다 블록 하강
- 블록 미리보기

이범식

- · 콘솔 세팅 (콘솔사이 즈, 스크롤바, 커서)
- ・ 블록 이동 및 고정
- · 배경 음악 및 ON/Off 기능
- 순서도

조영탁

- · 시작화면, 정지화면, 종료화면
- · 블록 한 번에 바닥에 낙하
- 한 줄 다 차면 줄 삭제및 점수, 속도 증가
- · 파일 입출력 및 최고 점 출력

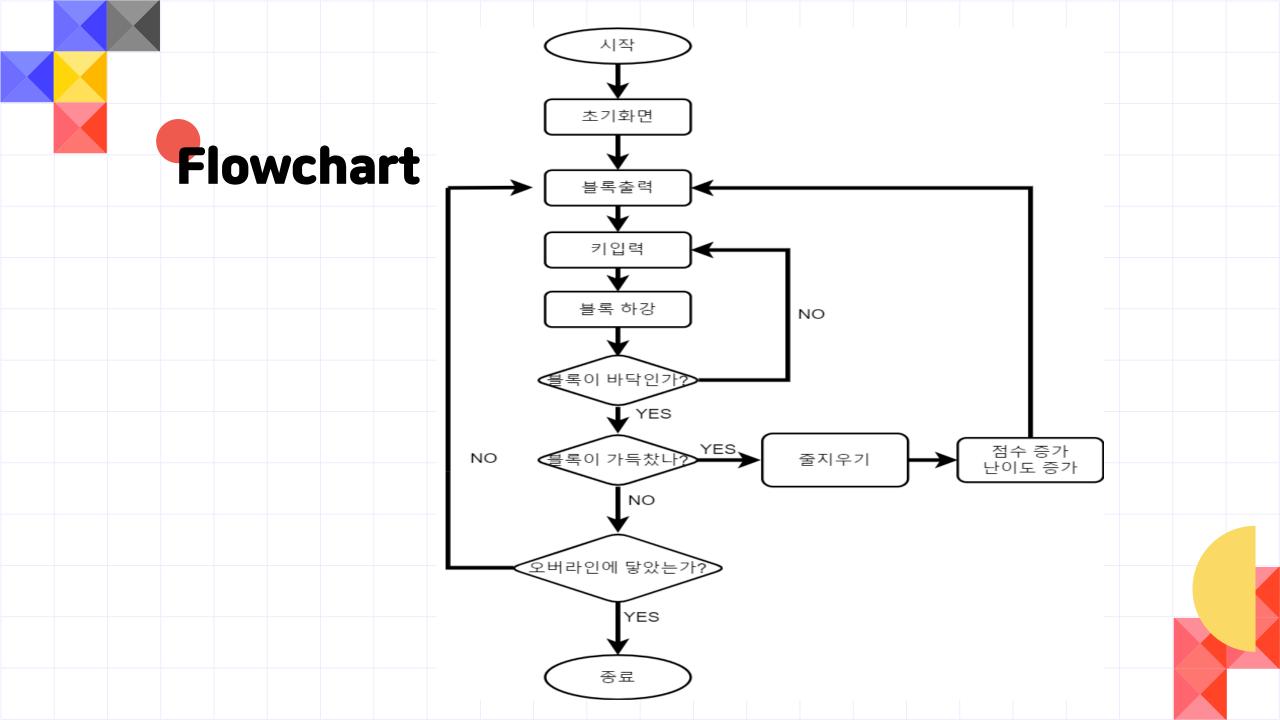


#define, Void Funtion()

#define을 매크로 상수를 사용해서 키보드 입력의 ASCII코드 값 저장 Function()을 만들어서 main()에 호출 다양한 함수를 통해서 연산 및 결과 출력









Main()

Main()함수에 들어가는 함수들 실행

```
srand(time(NULL));
SetConsole();
StartScreen();
ResetGameBoard();
ResetCopyGameBoard();
PrintGameBoard();
PrintNextBoard();
PrintScore();
NewBlock();
while (1) {
    if (flagNewBlock == 1)
        NewBlock();
    Keyboard();
    Drop();
    OverLineDraw();
    LineClean();
    PrintScore();
    GameOverCheck();
    PrintGameBoard();
    Sleep(150 - speed);
```

□ void main() {

cnt = 7;

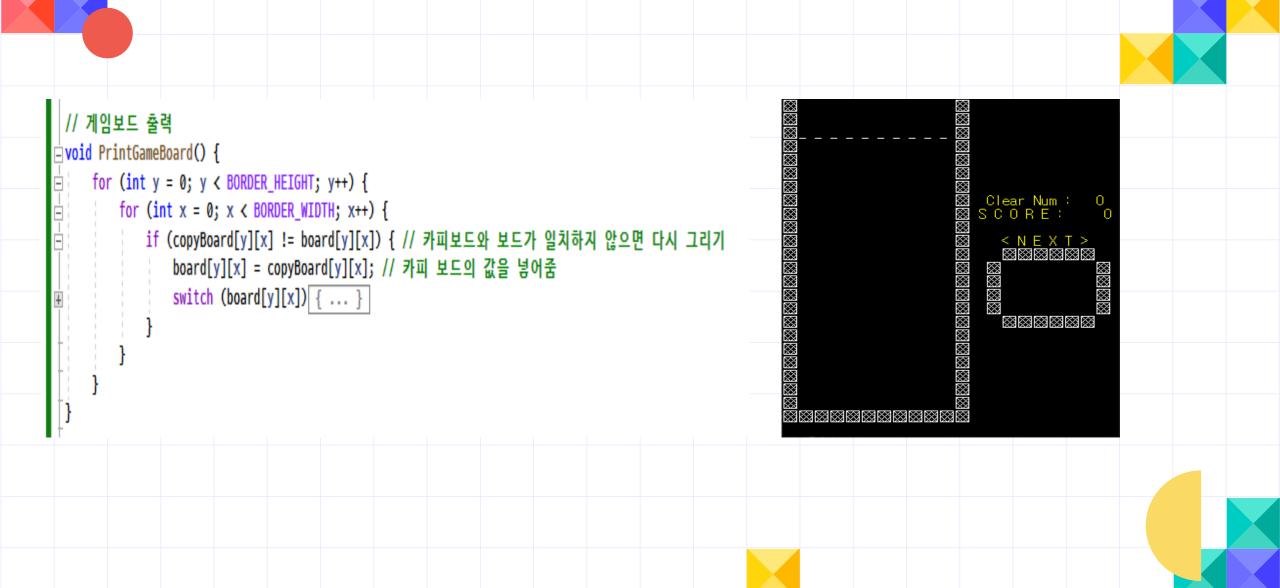
블록 판 제작

블록판 제작 / 블록판 출력

2차원 배열을 통해서 X, Y축에 입력을 받고 출력을 합니다

- Board(y)(x)에 해당하는 곳에 모양을 출력 합니다
- Board 함수 안에는 벽을 설정하기 위함

```
□void ResetCopyGameBoard()
     for (int y = 0; y < BORDER_HEIGHT; y++) {
         for (int x = 0; x \in BORDER_WIDTH; x++) {
             copyBoard[y][x] = EMPTY;
     for (int y = 0; y < BORDER_HEIGHT; y++) {
         for (int x = 0; x \in BORDER_WIDTH; x++) {
             if (x == 0 ; x == BORDER_WIDTH - 1) // 좌-우 벽 그리기
                 copyBoard[y][x] = WALL;
             if (y == BORDER_HEIGHT - 1) // 하단 벽 그리기
                 copyBoard[y][x] = WALL;
             if (y == 2 && x != 0 && x != BORDER_WIDTH - 1) // 게임 오버라인 그리기
                 copyBoard[y][x] = OVERLINE;
```



블록 생성

BlockShape[][][][]을 작성하여 4차원 배열을 생성하고 7가지 블록 모양 표현

[블록 종류][블록 회전][블록 모양][블록 모양]

```
{0, 0, 0, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 0, 0, 0,}
{0, 0, 0, 0,}
{0, 0, 1, 0,}
{1, 1, 1, 1,}
{0, 0, 0, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 1, 0, 0,}
{0, 0, 0, 0, 0,}
{0, 1, 0, 0,}
{0, 1, 0, 0,}
{0, 1, 0, 0,}
{0, 1, 0, 0,}
{0, 1, 0, 0,}
```

```
// [블록 종류][블록 회전][블록 모양][블록 모양]
int BlockShape[7][4][4][4] = {
       // ㅁ 블록
           {0, 0, 0, 0},
           {0, 1, 1, 0},
           {0, 1, 1, 0},
           {0, 0, 0, 0}
           {0, 0, 0, 0}.
           {0, 1, 1, 0},
           {0, 1, 1, 0},
           {0, 0, 0, 0}
           {0, 0, 0, 0},
           {0, 1, 1, 0},
           {0, 1, 1, 0},
           {0, 0, 0, 0}
           {0, 0, 0, 0},
           {0, 1, 1, 0},
           {0, 1, 1, 0},
           {0, 0, 0, 0}
  },
  for (int v = 0; v < 4; v++) {
     for (int x = 0; x < 4; x++) {
          GotoXY(15 + x, 12 + y);
          if (BlockShape[nextBlockType][0][y][x] == 1) {
              printf("■");
```

블록 하강

블록이 자동으로 바닥으로 내려오게 한다

• Drop() 함수가 main() 함수 안에 while문에서 계속 반복됩니다

```
// 바닥에 닿아도 즉시 고정 안됨
□void Drop() {
     if (CrushCheck(0, 1, 0) == TRUE && flagCrush == 0) {
         Move(0, 1, 0);
     else if (CrushCheck(0, 1, 0) == TRUE && flagCrush == 1) {
         flagCrush = 0;
     else if (CrushCheck(0, 1, 0) == FALSE && flagCrush == 0) {
         flagCrush = 1;
     else if (CrushCheck(0, 1, 0) == FALSE && flagCrush == 1) {
         FixBlock();
         flagCrush = 0;
```

블록 키 입력

키보드 입력 값을 매크로 상수로 선언하고

#define Enter 13

#define Esc 27

switch 문을 사용하여 그 상황에 맞는 경우를 실행

```
#define ARROW 224

#define LEFT 75

#define RIGHT 77

#define UP 72

#define DOWN 80

#define SPACEBAR 32

#define C 99

#define Z 122

#define P 112
```

```
// 키보드 입력
□void Keyboard() {
     if (_kbhit()) {
         int nkey = _getch();
         if (nkey == ARROW) {
             nkey = _getch();
             switch (nkey) {
             case LEFT:
                 if (CrushCheck(-1, 0, 0) == TRUE) {
                     Move(-1, 0, 0);
                 break;
             case RIGHT:
                 if (CrushCheck(1, 0, 0) == TRUE) {
                     Move(1, 0, 0);
                 break;
             case DOWN:
                 if (CrushCheck(0, 1, 0) == TRUE) {
                     Move(0, 1, 0);
                 break;
```

블록 줄 제거

가로축의 블록이 10개가 다 차면 한 줄을 지운다

최대 4줄까지 지울 수 있고 지워진 블록 위의 블록은 아래로 내린다

```
// 줄 완성시 삭제 및 점수 및 난이도 증가
☆void LineClean() {
    int blockCount = 0; // 한 줄의 블록 수
    int clearLine = 0; // 삭제할 줄 위치
    int clearCount = 0; // 삭제한 횟수
    for (int y = BORDER_HEIGHT - 2; y > 2 && clearCount < 4; y--) { // 4회 삭제하였다면 체크 안함
        for (int x = 1; x < BORDER_WIDTH - 1; x++) { // 한 줄의 블록 수 체크
           if (copyBoard[y][x] == FBLOCK)
              blockCount++;
       if (blockCount == 10) { // 줄 완성시 삭제
           for (clearLine = y; clearLine > 2; clearLine--) // 위의 줄을 내리기
              for (int x = 1; x \in BORDER_WIDTH - 1; x++) {
                  if (clearLine == 3) // 가장 윗 줄이면 내리지 않고 비우기
                     copyBoard[clearLine][x] = EMPTY;
                  else
                     copyBoard[clearLine][x] = copyBoard[clearLine - 1][x];
           clearCount++; // 삭제한 횟수 체크
           y++; // 삭제한 줄은 다시 체크
           speedUp += 1; // 난이도 증가에 필요한 줄 수 증가
           clearNum += 1;
        blockCount = 0; // 한 줄의 블록 수 체크 초기화
```

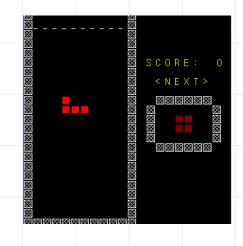
게임 오버

게임 오버

고정된 블록이 경계선에 닿으면 게임이 종료된다

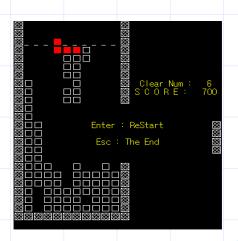
```
// 오버라인에 닿으면 게임 오버
□void GameOverCheck() {
    for (int x = 1; x < BORDER_WIDTH - 1; x++) {
        if (copyBoard[2][x] == FBLOCK) { // Y가 2인 위치에 고정된 블록이 있으면
           if (score > bestScore)
               Ranking(); // 랭킹
           for (int i = 0; i < 7; i++) {
               GotoXY(3, 10 + i);
                                                      \n");
               printf("
           GotoXY(8, 12);
           printf("Enter : ReStart");
           GotoXY(8, 14);
           printf(" Esc : The End");
           Sleep(1000); // 잠시 정지
           // 재시작 시 화면 초기화
           while (1) { ... }
```

실행 화면



블록 생성

블록 하강

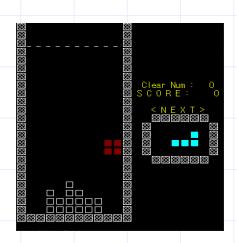


게임 오버

SCORE:

< N E X T >

블록 판 제작



블록 키 입력

블록 줄 제거

< N E X T >





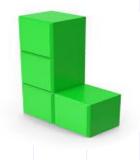
안서준

전체적인 틀을 구성하여서 조금 씩 코드를 추가하면 정상적으로 돌아갈 줄 알았지만 그 과정에서 발생하는 오류가 정말 많았다. 수정을 해도 팀원들과 계속 범위 를 지정하지 않으니깐 오류가 많 아서 소통의 중요성을 알았다.



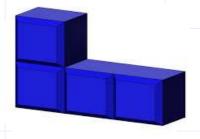
이범식

팀원들이 각자 맡은 작업을 팀원들이 수행을 하고 오류 수정을 하고 필요한 자료를 찾고 하나씩 해 내 가면서 팀워크의 중요성과 아직 모자란 부분이 많다고 느끼고 더 분발 해야겠다고 느꼈다.



오세룡

해야할 일을 나누고 어떤 일을 할지 정하는 과정에서 어떻게 나눠야 하나 정하는것과 제작하면서 생긴 원하지 않던 현상들을 수정하는게 힘들었지 만 완성하고 나니 나름 잘 만들어진 것 같다.



조영탁

처음에는 어려운 목표라고 생각하였으나, 모두 힘을 합쳐 노력하니 생각 보다 진행 속도가 빨라서 놀랐고, 이 로 인해 팀 프로젝트의 효과와 중요성 에 대하여 많이 깨닫게 되었다.

