개발 완료 보고서

제출일 : 2019년 8월 2일

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **팀명** | | **참여 인원** | | |
| IoTeam | | 심민선, 강진광, 고선주, 김태완, 오금성, 조민권, 오지구, 김주현, 박소희, 김현서 문선철, 김경훈, 박종화, 김길환, 양승호, 김규완, 장진혁 | | |
| **개발프로젝트 소개** | | | | |
| **프로젝트 명** | Ayotee Word Processor Project | | | |
| **활동일시** | 7 / 16 ~ 8 / 2 | | **장소** | 광주인력개발원 공학1관 융합1실 |
| **주요 주제** | C++ 개발 환경에서 메모장과 같은 워드프로세서 구현 | | | |
| **개발목적** | 클래스 간의 연결, 상속 부분을 이해하고 C++ 프로그래밍을 이해한다. | | | |
|
| **파트별**  **임무분담** | UI | 메인 인터페이스, 입력모드, 상태바 등 사용자 편의 화면 출력 구현 파트 | | |
| 편집1 | 에디터 내 정렬기능, 편집 기능 구현 파트 | | |
| 편집2 | 문자열 변경, 문자열 찾기, 폰트 기울기, 굵기 글자색, 밑줄 구현 파트 | | |
| 파일 | 저장, 불러오기, 삭제, 이메일 전송, 오토 세이브, 폴더 내 파일리스트 구현 파트 | | |
| **개발환경** | Windows OS / Linux SFTP Server(Putty) / Dev C++, Notepad++, Clion | | | |
|
| **일정표** |  | | | |
|
| **요구분석서** |  | | | |
| **코드**  **설명** | **UI** | 김태완 | ex)  **Class Notepad**   * 프로그램 실행시 기본적인 바탕이 되는 메모장 기본 틀 * **void Box(void)**   + 메뉴바와 상태바를 출력하는 함수 * **void PrintNotepad(void)**   + 메모장의 가장 기본이 되는 틀 * **string PrintBlank(void)**   + 공백을 출력해주는 함수   **Class Menu**   * 프로그램 실행시 기 * **void Box(void)**   + 메뉴바와 상태바를 출력하는 함수 * **void PrintNotepad(void)**   + 메모장의 가장 기본이 되는 틀 * **string PrintBlank(void)**   + 공백을 출력해주는 함수 | |
| 오금성 | **Class uiBox**   * Box형 레이아웃을 만들어 주요 메서드를 받아와서 출력하는 클래스 * **void setBox(int x, int y, int width , int height);**   + 레이아웃을 만들기 위한 박스 만들기 * **void inputHorizon(std::string str, int num);**   + setBox() 란 메서드 안에 사용될 수평선 * **void inputVertical(std::string str, int num, int num2, int x, int y);**   + setBox() 메서드 안에 사용될 수직선   + num = str이 들어갈 갯수   + num2 = 양쪽 수직선을 채울 빈칸의 숫자   + x,y 는 setBox() 에 따른 위치를 입력 받아 조정 * **void uiHelp();**   + 도움말 UI를 실행하면 나타나는 메서드 * **void uiHelpMenu();**   + 메뉴에 대한 도움말 UI ( 벡터에 값을 입력해 연속적으로 출력하는 방식) * **void uiHelpEdit();**   + 편집모드에 대한 도움말 UI ( 메뉴와 방식 동일 ) * **void uiEmail();**   + 이메일 UI ( 이메일 클래스에 있는 기능을 각각 실행하기 때문에 출력 위치 조정과 레이아웃에 대한 크기를 정해준 형태) * **void uiTextlist(int Mode = 0, int startPoint = 0);**   + 모드 ( 0 : 목록 , 1 : 삭제, 2 : 열기, 3 : 뒤로 가기 )   + startPoint를 설정해 주고 maxPage - endPoint 만큼 파일 리스트를 읽어와서 출력한 후 다음 페이지 이동 시 그 다음 목록이 출력 * **std::string selectFile(int Mode = 0);**   + 실행 시 uiTextlist() 를 출력하고 방향키를 입력해 선택한 파일을 반환하는 메서드   + 모드는 uiTextlist()와 동일 * **void uiDelete();**   + uilist를 만들기 전 버전으로 파일명을 직접입력해서 삭제하는 메서드 * **void uiDeleteTest();**   + selectFile() 을 실행하여 선택한 파일을 delt.cpp 에 있는 delete 하는 메서드를 실행 * **int uiQuit();**   + 종료 선택 시 1을 반환하는 메서드 * **void uiEnter();**   + 키 입력시 다음 메서드가 실행될 수 있도록 화면 정지 상태를 만듦 * **int uiChoice(std::vector <std::string> choice);**   + 벡터에 출력될 문자열을 정해진 위치에 나타내고, 방향키로 좌, 우 선택하여 enter 입력시 좌측 위치의 값에 따라 반환함. * **void uiCalculation();**   + 계산기 메뉴얼에 따른 입력과 출력 레이아웃 출력 * **void versionPrint();**   + 버전에 따른 정보를 출력 | |
| 오지구 | **Class Title**   * 프로그램 실행시 첫 화면 * **int title\_print()** * 함수를 모아 출력하고 위에서 출력하고 밑으로 떨어지는등 효과를 담당 * **void loading\_pan()** * loading 문자를 하나씩 출력하면서 원하는 부분에 색을 칠함 * **void sample()//샘플 1,2,3 있음** * 로고, 상황에 따라 3개로 나뉨   **Class Creator**   * 만든이 출력 * **void creator\_print()** * 함수를 모아 출력하는 함수 * **void enter()** * 만든이 출력중 언제든 엔터 입력시 탈출 * **void cre\_print(int a, int b, int num)** * 3줄로 만들어지는 문자를 출력하고 출력 위치가 일정이상 낮다면 그 부분은 출력하지 않아서 밑에서 올라오는 것 처럼 만듬 * **void cre\_name(int a, int num)** * **cre\_print()**안에 두고 상황에 맞춰 소속 팀원의 이름을 출력   **Class Cursor**   * 커서 이동 * **void set()** * 초기에 위치할 커서 위치, 커서가 이동 가능한 최대 위치, 메뉴숫자, 한번 입력시 이동할 거리 등 이곳에서 수정하면 나머지 함수도 한꺼번에 수정 * **void cur\_take()** * 커서 위치에 따라서 다른 사람들의 함수를 호출 * **void paint\_color()** * 커서위치에 따라 메뉴에 색을 칠함 * **void hide(int n)** * 커서를 숨기거나 보이게함 * **void cur(int xi,int yi)** * 커서 위치 변경, 표시 * **void cur\_name()** * 커서 위치에 따라 메뉴의 파일들을 팝업창으로 보여줌 | |
| 문선철 | **Class Timer**   * 스레드를 통해 실시간으로 노트패드의 상태를 나타내는 클래스 * **void SetMode(int i);**   + 현제 상태(메뉴와 입력창)를 지정한 변수를 받는 함수 * **int GetMode(void);**   + 리턴해 주는 함수 * **void Refresh(int i);**   + 키를 입력 시 받은 값을 반환해 타이머를 갱신해 주는 함수 * **void Print(void);**   + 타이머를 돌리고 상태창을 출력할 내용을 담아놓은 함수 * **void lockMutex();**   + 다른 동작과 연동되지 않도록 뮤텍스를 걸고 * **void unlockMutex();**   + 뮤텍스를 끈다.   **Class Calendar**   * 달력 * **int PrintCal(void);**   + 달력을 만들어가는 함수 | |
| **편집1** | 고선주 | **Class Editor**   * 편집 기능 관련 수행 클래스 * **void blockText(int currentLine, int currentPos\_y);** * 블럭관련 기능 | |
| 김경훈 | **Class Editor**   * 편집관련기능 수행 클래스 * **void addBulletPoint(void);** * 글머리기호 추가 함수 * **void removeBulletPoint(void);**   + 글머리기호 제거 함수   **Class Undoer**   * 되돌리기 기능 수행 클래스 * **UndoSet \*getUndoSet();**   + 되돌리기 시 되돌리기 취소에 스택을 쌓고, 되돌리기를 실행 * **UndoSet \*getRedoSet();**   + 되돌리기 취소 시 되돌리기에 스택을 쌓고, 되돌리기 취소를 실행 * **void push(UndoSet \*uds);**   + 특정 저장조건시 되돌리기 스택에 전체글의 내용을 저장 | |
| 강진광 | **class Text**  - 텍스트를 저장할 벡터와 타이핑에 관련된 인덱스 값을 관리하는 클래스  **- void CorrectionText();**  : 텍스트 내 불필요한 시스템 문자를 제거 하고 버퍼 인덱스 값을 초기화  **class Editor**  - 편집과 관련된 모든 기능을 수행하는 클래스  **- void KoreanInputConversion();**  : 모드 선택 메뉴에서 한글을 입력하였을 경우 영문 값으로 변경처리  - **void writeText();**  : 입력 모드에서 입력된 값에 따라 호출할 함수를 판별, 연결한다  **- int countCharacter(int size=-1, int line=-1) const;**  : 특정 라인의 특정 위치에 해당하는 총 문자 갯수를 반환  **- int countCharacter(std::string &text) const**;  : 특정 텍스트의 전체 문자 갯수를 반환  **- void handlePressEnter();**  : 입력 모드에서 엔터를 입력하였을 경우를 개행 처리  **- void handlePressTap();**  : 입력 모드에서 탭을 입력하였을 경우 현재 커서 위치로부터 공백 5칸을 채운다  **- void deleteCharacter();**  : 입력 모드에서 백스페이스를 입력하였을 경우 한글/영문 판별 하여 삭제  **- void handleKorean();**  : 입력 모드에서 한글을 입력한 경우 string 에 채워넣는다  **- void handleEnglish();**  : 입력 모드에서 영문을 입력한 경우 string 에 채워넣는다  **- void correctionBufferIndex\_x();**  : index 값이 범위를 초과하지 않도록 보정한다  **- int getIndex\_xByNumOfChar(int numOfCharacter, int line ) const;**  : 특정 문자 갯수에 해당하는 인덱스 값을 반환  **- void handleExceedWidthLimit();**  : 텍스트들이 화면상에 보이는 테두리를 벗어나지 않도록 조정한다  **- void handleIndexMoving();**  : 방향키를 입력하였을 떄 현재 가르키고 있는 string 내부 인덱스 값의 이동을 처리  **- void handleCursorMoving();**  : 인덱스 값을 기준으로 화면상에 보이는 커서를 이동시킨다  **- void setAlignType();**  : 좌, 우, 중앙 정렬을 설정한다  **- int getAlignedSpace(int line) const;**  : 설정된 정렬 값에 따라 텍스트를 찍을 x 좌표값을 조정  **- void setBulletPoint();**  : 글머리기호 삽입을 위한 UI를 화면상에 띄운다  **- void getUndoAndRedo();**  : 되돌리기, 되돌리기 취소 UI를 화면상에 띄운다  **- void blockText(int currentLine, int currentPos\_y);**  : 텍스트의 특정 범위를 블록 처리한다  **- void blockAllText();**  : 텍스트 전체를 블록설정 한다  **- void copyText();**  : 블록 설정된 부분을 복사할 수 있도록 버퍼에 저장  **- void cutText();**  : 블록 설정된 부분을 잘라낼 수 있도록 버퍼에 저장하고 삭제  **- void pasteText();**  : 버퍼에 저장된 내용을 현재 커서 위치에 붙여 넣는다  **- void translateText();**  : 블록 설정된 부분을 영/한, 한/영 번역한다  **- void findString();**  : 특정 문자를 찾아서 커서를 해당 위치로 이동시킨다  **- void findAndReplace();**  : 문자열 찾아서 바꾼다  **- translate.py**  : 파이썬 google translate 라이브러리를 활용, 번역할 문장을 socket으로 전달 받아 번역 후 C++ 클라이언트로 전송한다  **class TypingPractice**  - 타이핑 연습기능 구현 클래스  **- void readTextFile(const std::vector <std::string> \*editorTexts );**  **:** 타이핑 할 텍스트를 불러온다  **- void printSpeedAndAccuracy();**  : 쓰레드로 구동하여 빠르기와 정확도를 실시간으로 출력  **- void typeText(const int currentLine );**  : 타이핑 입력을 처리  **- void checkError(const int currentLine);**  : 타이핑 된 텍스트의 오류를 감지  **- void markErrorPoint(int xPos);**  : 감지된 오류 갯수만큼 V 표시를 출력  **class Gui**  -. 편집모드와 관련된 모든 UI 구현을 위한 클래스  - **void MakeBox(int width, int height, int boxPos\_x, int boxPos\_y, std::string str, int setStrPos\_x = CENTER, int setStrPos\_y = TOP );**  : 문자열을 매개변수로 받아 Box UI에 함께 출력 시킨다  - **void MakeBox(int width, int height, int boxPos\_x, int boxPos\_y, std::vector <std::string> &strs, int setStrPos\_x = CENTER, int setStrPos\_y = TOP);**  : 여러개의 문자열을 하나의 벡터로 전달받아 Box UI 내부에 출력시킨다  - **void MakeTable(std::vector <std::string> \*textBuffer, int currentLine);**  : 편집모드에서 표그리기 기능을 수행한다  **class BottomUI**  - 하단 UI를 출력하는 클래스  **- void fontThickness();**  : 현재 설정된 폰트 굵기 값을 출력  **- void fontUnderline();**  : 현재 설정된 폰트 밑줄라인 값을 출력  **- void fontForegroundColor();**  : 현재 설정된 폰트 글자색을 출력  **- void fontBackgroungColor();**  : 현재 설정된 글자 배경색을 출력  **- void SetMode(int mode);**  : 현재 모드상태(메뉴/입력)를 설정  **- void PrintBottomUI();**  : 하단 UI 와 관련된 전체를 통합 출력 | |
| 김길환 | **Class Editor**   * 편집모드 종합 클래스 * **void cutText(void)**   - 자르기 기능  - void copyText(void)  - 자르기의 복사기능  - void deleteCharacter(void)  - 자르기의 복사 이후 원본 제거 기능   * **void setAlignType(void)**   - 왼쪽 정렬 / 가운데 정렬 / 오른쪽 정렬 선택  - int getAlignedspace(int line)  - 기본값 : 왼쪽 정렬  - 가운데 정렬, 오른쪽 정렬 선택시 계산 | |
| 양승호 | **Class Editor**   * 편집관련기능 수행 클래스 * **void copyText();**    + : 블록 설정된 부분을 복사할 수 있도록 벡터에 저장   (방식은 다르나 이미 만들어버리심) | |
| **편집2** | 장진혁 | ex)  **Class MINI\_GAME**   * 베네치아 실행에 필요한 변수,함수가 들어있는 클래스 * **void playing(MINI\_GAME\* );** * 함수들을 조합하여 게임을 실행하는 함수 * **void drop\_word(void);** * 떨어지는 단어와 생성되는 단어를 관리하는 함수 * **void write\_word(void);** * 플레이어의 입력을 담당하는 함수 * **void check(void);** * 한글의 문자열을 체크하여 지울지 결정하는 함수 * **void screen(int );** * 게임의 초기(기본적인)화면을 출력하는 함수 * **void life(void);** * 목숨 수에 따른 그 출력을 담당하는 함수 * **void water\_level(int );** * 집 주변 물의 수위를 출력하는함수 * **void ending(int );** * 게임이 끝난후 엔딩을 출력하는 함수 * **void language(void);** * 플레이할 언어의 설정을 하는 함수 * **void rank(void);** * 플레이어의 점수(시간)을 포함한 top5의 랭킹을 출력하는 함수 * **void input\_rank(void);** * 플레이어의 이름,점수(시간)을 랭크에 대입하는 함수 * **void name\_input(void);** * 플레이어의 이름을 입력하는 함수 * **void exchange(int ,int );** * 랭킹을 위한 버블정렬을 하는 함수 | |
| 김주현 | **Class Font\_Setting**   * 글씨굵기, 색변경, 밑줄 기능 담당하는 클래스 * **Font\_Setting()**   + 초기화 세팅 생성자 * **void ToggleThickness()** * 글씨 굵기 온/오프 함수 * **void ToggleUnderLine()** * 밑줄 온/오프 함수 * **void ToggleReverseColor()** * 색반전 온/오프 함수 * **void SetColor()** * 색 선택 및 색 코드 설정 함수 * **void ApplyFontSetting()** * 표시 출력 세팅 함수 * **void ResetSetting()** * 표시 효과 리셋 함수 * **bool GetThicknessState()** * 글씨 굵기 상태 함수 * **bool GetUnderLineState()** * 밑줄 상태 함수 * **int GetBackgroundColorValue()** * 배경색 값 리턴 * **int GetForegroundColorValue()** * 글씨색 값 리턴 | |
| 박종화 | **Class Editor**   * 편집1 클래스에 병합 * **Editor::findAndReplace()** * 찾고자 하는 문자열을 입력 후 바꾸고싶은 문자열을 입력하면 입력된 문자열에 따라 바뀌게 됨.(문자열 바꾸기)   **Class CInterface**   * 진수 변환 제외로 인한 ui에 병합   **Class CCharStack**   * 연산자를 저장하기 위한 스택 클래스   **Class CIntStack**   * 숫자를 저장하기 위한 스택 클래스   **Class CCalc**   * ui클래스에 병합 * **void Rearrange()**   + 입력받은 str을 temp값으로 복사하여 스트링 문자열을 인덱스 하나씩 배분하여 식을 재배열 하여 CCharStack클래스와 ClntStack 클래스에 나누어 저장하게 되는 함수 * **int Priority(char ch)** * 연산자 우선순위를 반환하는 함수(자연로그나 sin,cos,tan등이 가장 우선순위가 높고 그다음으로 \*,/ 그다음으로 +,-순으로 정해져 있다.) * **void CalcFormula()**   + Rearrange()함수에서 재배열된 reFormula를 대상으로 계산식을 수행하는 함수 * **void CalcSqrt(float tmp) / CalcLog(float tmp) / CalcTri(char ch, float tmp)**   + 루트와 자연로그, 삼각함수를 리턴해주는 함수   **Class uibox**   * **void uiCalculation()**   + 메인에 있던 계산기 실행하는 소스를 uibox클래스에 옮겨놓은 함수 | |
| 김규완 | **Class Editor**   * 편집 클래스에 병합 * **void findString()** * 현재 작성중인 내용 중에서 호출하여 원하는 문자열 입력시 커서가 찾은 위치로 이동.   **Class Exchanger**   * 환율 계산기 * **void Exchanger()**   - 생성자   * **Run()**   - 클래스 실행 함수   * **print\_screen(void);**   - 테두리 상자 그리는 함수   * **refresh\_screen(void);**   - 화면에 메뉴를 출력하는 함수   * **set\_force(void);**   - 값을 강제로 변환하는 함수   * **input\_value(void);**   - 수치를 입력하는 함수   * **select\_country(void);**   - 국가 선택   * **calculate(void);**   - 환율 계산   * **print\_stat(void);**   - 화면에 결과를 출력하는 함수   * **check\_invalid\_value(void);**   - 수치를 입력할때 에러 방지 함수 | |
| **파일** | 조민권 | ex)  **Class Textlist**   * 파일 리스트 출력 기능 * **void list\_dir();**   + opendir, readdir, closedir, 을 사용하여 폴더 내 특정 파일 리스트 출력(.txt, .c, .cpp)   **Class Autosave**   * 자동 파일입출력 * **void Auto();**   + 쓰레드를 사용하여 특정 카운터가 될 시 현재 작업중인 벡터를 파일입출력을 사용하여 텍스트문서로 저장(백업)   **Class Filedata**   * 프로그램 내의 텍스트, 파일 연관 기능 * **void Setfilename(string filename\_);** * **void Getfilename(void);**   + 입력받거나 불러온 파일의 이름을 여러번 반환받아야 하는 번거로움을 해결해주는 함수   + Setfilename에서 string으로 파일명을 초기화한 후 Getfilename으로 반환하여 쉽게 파일명을 사용하도록 함 * **void Findnameas();**   + ‘다른 이름 저장’ - 저장 할 파일명을 입력 받아 폴더내의 파일과 이름을 비교하여 중복여부 판단   + 중복 시 중복임을 알려주고 다시 입력받게 함   + 비중복 시 save함수호출; * **void Findname();**   + ‘저장’ - 새 파일과, 작업 중인 파일을 비교함   + 저장하지 않고 작업했을 경우, Findnameas 함수호출   + 저장되어 있는 파일인 경우 현재 열린 파일의 이름으로 덮어쓰기 저장 * **void save();**   + 작업중인 텍스트 vector를 전부 파일 입출력기능으로 텍스트 파일 저장   + 입력과 불러온 파일의 확장자명에 따라 생성 또는 덮어씌워지는 파일이 다름 * **void load(std::string wantName);**   + ‘불러오기’ - 파일리스트를 띄워 현재 폴더내에 불러오기 가능한 파일 목록을 선택하여 vector에 저장하여 텍스트 입력란에 출력 * **void new();**   + ‘새 파일’ - 텍스트 입력란과 파일을 초기화   + 텍스트 입력 중 새 파일을 불러오면 현재의 작업물의 저장 의사를 물어봄   + 새 파일을 불러올 시 vector를 초기화하고, 현재 파일명을 초기화하여 파일 생성가능한 상태로 만듬 | |
| 심민선 | ex)  **Class email**   * 보내는 사람 이메일과 smtp코드. 받는 사람 이메일, 첨부파일 명을 입력 받아 바이너리 파일로 생성시킨다. * **void FileSend(void)** * 시스템 명령어로 email.py를 실행시켜 파일전송을 진행한다. * **void AdEmail\_send(void)**   + 보내는 사람 이메일을 입력 받아 바이너리 파일로 저장한다. * **void AdEmail\_reci(void)**   + 받는 사람 이메일을 입력 받아 바이너리 파일로 저장한다. * **void AdEmail\_code(void)** * 보내는 사람 이메일 SMTP코드를 입력 받아 바이너리 파일로 저장한다. * **void AdEmail\_text(void)** * 전송하고자 하는 텍스트 파일명을 입력받아 바이너리 파일로 저장한다.   **email.py**   * email 클래스에서 생성된 바이너리 파일들을 아스키 코드로 읽어들여서 첨부할 파일과 함께 이메일로 전송한다.   **Class delt**   * 선택한 텍스트를 제거한다. * **void FileDel(void)**   + CommendMake()를 실행, 반환 받은 check값으로 제거와 에러 메시지 출력을 판단하여 실행한다. * **int CommedMake(void)**   + FileFind()를 실행, 반환 받은 check 값으로 제거 커맨드 생성을 결정하여 실행한다. * **int FileFind(char path[], char name[])**   + 파일의 존재 여부를 확인하여 그 판단한 값을 반환한다.   **weather.py**   * 네이버 ‘날씨’ 키워드 검색 페이지에서 당일 날씨와 온도 데이터를 파싱하여 txt 파일로 저장한다.   **install.sh**   * 지정 디렉토리를 생성하여 .deb 파일을 압축해제한다. | |
| 김현서 | **Class save**  -void save\_mod(void)  -파일을 저장하고 다른이름으로도 저장  **Class load**  void load\_mod(void)  -파일을 불러와서 내용을 출력한다. | |
| 박소희 | **Class cover**   * 파일 중복확인과 덮어쓰기, 자동저장 * **int inspection()**   -파일 열기 결과를 이용해 중복 확인하는 함수  - **void overlap()**  -사용자의 선택에 따라 파일 덮어쓰기 또는 자동저장을 하는 함수 | |
| **후기** | 김태완 | 3주 간 대 장정 프로젝트의 UI파트장을 맡게되었습니다. 자발적으로 지원해서 하게 되었지만 한편으론 부담감도 느꼈습니다. 처음에는 UI파트는 할게 별로 없고 그다지 중요하지 않겠구나 라고 생각했지만 처음 퀄리티 좌지우지 하는게 UI 부분이란 걸 알게 되었습니다. 또 UI 파트가 처음 만들고 끝나는 게 아니라 거의 모든 파트 부분에 들어가 있어서 상당 부분 애먹었습니다.  제가 무리한 부탁을 많이 했지만 적극적으로 수용하여 부탁을 들어준 파트원 덕분에 파트장을 편하게 한 것 같아 기쁩니다. 개인적으로 프로젝트를 진행하면서 발생한 문제점이 두 가지 정도 있는 것 같습니다.  첫 번째는 UI파트 전부를 헤더파일 하나에넣어서 처리하면 다른 파트에서 처리할 때 편할 것 같아 그렇게 했지만 정반대로 헤더파일에서 헤더파일을 서로 참조 할 때 문제점이 많이 발생하였습니다 .  두 번째는 추가사항을 고려하지 않고 코딩하여 나중에 추가하려고 할 때 많은 부분을 고쳐야 했고 유지보수에 어려움을 겪을 수 있다고 생각했습니다.  다음 프로젝트를 진행할 때는 이러한 문제점을 생각하면서 코딩해야 겠다는 생각을 하였습니다.  마지막 마무리까지 하고 싶었는 데 개인적인 일로 파트장으로 떠넘기 듯이 한 것 같아 많이 미안한 마음입니다.그래도 성공적으로 UI파트를 마무리해서 뿌듯합니다. | | |
| 오금성 | 어쩌다 보니 UI 파트를 맡아 팀장도 떠넘기 듯이 시작한 프로젝트가 어느새 완료 시점까지 다가오게 되어 새삼 스럽고, 막상 만들어 가면서 팀원들과 많은 소통을 해야하는 부분들이 있었지만 어떤 식으로 해야할지 구체적으로 정하지 못해서 자꾸 주위를 맴돌았던 것 같다.  버그 수정과 다른 파트와 연계된 부분들을 수정함에 있어서 다른 파트의 소스들을 보며 이해해야하는 부분들이 많이 필요해서 자꾸 보다 보니 실력에 많은 도움이 된 것 같다.  마지막까지 함께 하지 못한 우리 UI 파트 팀장이 개인 일정으로 같이 마무리 하지 못하게 되어 많이 아쉽고 대리로 파트장 임무 수행을 하며 느낀 바로는 프로젝트 기간에 많은 부담을 느꼈을 거라 생각한다. | | |
| 오지구 | ui파트 팀이 되서 ui를 다양하게 테마로 꾸미려 했으나 결국 문자 크기 변경등의 방법을 찾지 못해 아쉽게도 다양한 테마를 만들지 못하고 몇몇 부분에서 미흡했던것 같습니다.  그렇지만 그렇게 만든 타이틀과 효과들을 보여줬을때 주변에서 잘했다고 칭찬 해줘서 그나마 자신 가질수 있었다고 생각합니다  가장 부족했던 점은 각종 변수 명을 임의로 정해서 다른 사람이 봤을때 알아보기 힘들었다는 것입니다. 평소 나만 알아볼수 있다거나 유지보수가 힘들다는 말을 들어 신경을 쓰고 주석을 열심히 달았는데도 부족했던것 같아 좀 더 주의를 기울여야 겠습니다. | | |
| 문선철 | 과거에 사용했던 스레드와는 다른 형태를 사용해 보았습니다. 그 외에도 다른 파일을 켜는 system이나 다른 새로운 기능을 지닌 함수를 사용하고 그 외에 다른 방식으로 값을 전달해 줘서 값을 바꾸는 등의 방법에도 다양한 것이 있다는 것을 깨달았습니다. 과거에 사용해 보았던 달력 만들기를 다시금 만들면서 코딩은 쉬지 않고 계속 해야 한다는 것을 깨닫는 시간을 가질 수 있었습니다. | | |
| 고선주 | 프로젝트 기간 동안 할당된 파트를 맡아서 진행 해보았지만 생각 했던것보다 구현이 많이 어려웠었고 구현환경을 동일한 환경에서 진행되지 않아 소스 코드를 합치는데에도 많은 어려움이 있었습니다.  이번 프로젝트를 진행하면서 스스로 많이 부족하다는 것을 느끼게 되었고 다음 번 프로젝트에서는 맡은 파트를 잘 구현할 수 있도록 좀더 열심히 공부를 해야 겠습니다. | | |
| 김경훈 | 저에게 3주의 프로젝트 중 첫 주는 예비군, 두번째 주는 구현기간, 세번째 주는 사실상 프로그래밍과는 관련이 적은 매뉴얼 작성 + 허투루 쓴 시간이었던 것 같습니다.  프로젝트 진행간에 중간에 참여한다는 것은 맡은 역할이 큰 것이 아니었음에도 불구하고 정확한 역할과 프로젝트 자체의 전반적인 이해가 힘들었으며, 다른 인원의 구현속도 등에있어서도 부담이 적지않았습니다.  결과적으로는 맡은 바 임무에 있어서는 완수했다는 뿌듯함이 있지만서도, 나름대로 지금 껏 개인시간을 투자했다고 생각했지만, 적용능력이 낮아서인지 앞으로 다른 인원들의 눈에 띄는 구현속도, 실력에 뒤지지않고 따라잡을 수 있을까라는 걱정과 후회도 많이 남습니다. | | |
| 김길환 | C가 아닌 C++로의 프로젝트라 C++은 제대로 아는거 하나 없어 걱정 많이 하며 시작했는데 그래도 C와 비교하며 이리 저리 찾아보며 C++의 기초를 스스로 학습할 수 있었던 계기가 되었던 거 같지만 그래도 뭔가 깔끔하게 학습한게 아니라 막히는 부분만 학습하다 보니 전체적인 기초 학습은 한번 필요 하다는 느낌도 들고 단기간 코딩학습은 실패하더라도 역시 만들어 보는게 제일 빠르다는 생각이 들었습니다. | | |
| 강진광 | 3주간의 긴 프로젝트가 끝났습니다. 처음 시작할때는 파이썬 수업과 겹치다보니 집중도 안되고 의욕도 많이 떨어져서 잘 할 수 있을지 걱정이 되었지만 방학 기간에 들어서서서 프로젝트와 관련된 코딩에만 집중할 수 있는 환경이 되니 슬슬 재미가 붙기 시작하면서 끝마칠 때까지 즐겁게 코딩할 수 있었습니다. 완성본도 처음 생각했던 것보다 훨씬더 훌륭하게 만들어진 것 같아 뿌듯했습니다. | | |
| 양승호 | 3주짜리 프로젝트가 끝났다는 것이 이제서야 실감이 나는것 같습니다.  사실 프로그래밍을 하였어도 본 프로그램에 들어가지 않은게 있어서 좀 그렇기도 한데 적극적으로 어필하지않은 제 탓인것 같기도 하고, 제 파트를 만듬과 동시에 타 인원의 소스코드가 들어가면서 ‘더 해서 뭐하냐 이미 다 만들어졌을텐데’ 하는 생각으로 논 적도 많습니다.  그래서 그런가 이제와서 후기쓰려고 회상해보니 후회가 많은 프로젝트가 된 것 같고, 다른 팀원들의 속도와 실력에 뒤쳐진것 같다는 느낌도 많이 받고 있습니다. | | |
| 장진혁 | 3주간에 프로젝트로 c++에 관한 정보를 많이 습득한 것 같습니다.  이 프로젝트가 iot반에 좋은 영향을 준 것 같습니다. 서울에 갔다 오고 난 뒤부터는 빼먹은 만큼 열심히 해야겠다. 라고 생각은 했지만 막상 그 걸 채우기가 쉽지 않았습니다. 그러나 옆에서 팀원들과, 형 누나들의 행동에 어떻게든 뭐라도 할 수 있었던 것 같습니다. | | |
| 김주현 | 지금까지 진행한 프로젝트 중 가장 긴 3주라는 기간 동안 진행하는 대규모 프로젝트였습니다. 프로젝트를 하면서 맡은 파트에 대한 아쉬움이 없다고 할 수 는 없지만 자료를 찾아보고 직접 실행이 가능한지 시도해보고  모르는 부분을 물어보고 만들어가면서 완성되어가는 결과물을 보니 뿌듯했고 좋은 경험을 한 것 같습니다. 다른 팀원들이 하는 것을 보며 많이 배운 것 같고 아직 부족한 점이 많다는 것을 느꼈습니다. 다음 프로젝트를 위해 더 열심히 공부해야겠다고 생각했습니다. | | |
| 박종화 | c로 기존처럼 코드를 짜는것과 c++로 객채지향적으로 코드를 짜는것은 정말 다르다는 것을 뼈저리게 깨달았습니다. 변수의 사용부터 소스의 결합까지 어려운것 투성이였고 이해가 안되는것도 정말 많았습니다. 그때마다 주변의 도움을 받으며 잘 넘기기도 하였고 여러가지를 배우면서 조금 더 소스가 가다듬어지고 정리가 되면서 많은 것을 배운 것 같습니다. 아직 부족한게 많아 구글의 도움을 많이 받지만 조금더 익숙해지고 학습하다 보면 c언어를 짜듯이 c++도 자연스럽게 할 수 있을 것 같습니다. | | |
| 김규완 | 3주라는 기간이 무색하게 시간이 정말 빠르게 흘러 갔습니다.  처음에는 분량이 많고 코드를 합칠때 안될 것 같다고 생각했지만  대신에 여러 팀원과 힘을 합쳐서 프로젝트를 완성시켜 보니 오류도 있는 것 같았지만 대신 그만큼 빠르게 고쳐지는 것 이었습니다. | | |
| 조민권 | 3주동안 큰 불화이벤트 없이 서로 도와가며 하나, 둘씩 테두리 잡아가고 기능들을 넣으며 완성돼가는 프로그램을 보니 짜릿한 희열감을 느끼며 너무 재미있었습니다.  개발 중 저에게 간혹 오던 슬럼프가 이번에는 찾아오지 않고, 끝까지 물고 늘어져 개발해 가는 팀원들을 보니 앞으로도 팀원들에게 민폐가 되지 않도록 열심히 잘 해야겠습니다. | | |
| 박소희 | 이번 프로젝트 처음 시작할 때는 약간의 걱정이 있었지만, 성공적으로 끝내어져서 기쁘고  팀장분과 다른 분들이 잘 이끌어 주셔서 이런 결과가 나올 수 있었던 것 같습니다  저도 노력해서 한 몫 제대로 하도록 공부하고 노력하겠습니다  이번에 소통을 원활하게 하지 못한 거 같아 다음번에는 좀 더 소통하며 열심히 할 것입니다. | | |
| 심민선 | 팀장으로서 할 일과 파트원으로서 할 일이 맞물린 나머지 조금 정신없던 3주가 아니었나 싶습니다. 팀장으로서 역할을 주고 전체 스케줄을 진행하고 파트별 문제점을 조율하고 책임을 져야 했던 하루하루가 값진 경험이 되었습니다. 하지만 파트원으로서 구현 목적에 있어서는 조금 무책임한 감이 있었던 것 같습니다. 파이썬이든 쉘 스크립트든 이것저것 집어오긴 했지만, 깊이가 부족했던 것 같습니다. 그럼에도 불구하고 완성된 프로그램을 보면서 여기에 내가 한몫했다는 생각을 하니 너무 즐겁네요. 3주 동안 진행한 프로젝트 잘 마무리된 것 같아서 기쁩니다. 모두 고생하셨습니다. | | |
| 김현서 | c++을 배워보지 않아 생소하고 어려웠습니다. 자꾸 c처럼 짜지더라고요. 처음엔 힘들었는데 팀원들이 도와주고 공부하고 열심히 한 덕분에 제가 했던 부분에 있어서는 완벽하게 이해할 수 있었습니다. 이제 조금정도는 c++로 무언가 하는데 익숙해질 것 같습니다. 좀 더 공부해서 다음에는 좀 더 도움이 될수 있도록 열심히 해야겠다는 생각이 듭니다. 모두들 고생많으셨습니다. | | |
| **비고** |  |  | | |