

c프로그래밍 및 실습

생활영어 배우기

프로그램

진척 보고서 #2

제출일자: 2023.11.26

제출자명: 김진하

제출자 학번: 221473

1. 프로젝트 목표

번역기를 통한 영어가 아닌 실제 생활에서 많이 쓰이는 영어 표현을 배우며 고객의 생활 영어 향상을 목표로 한다.

1) 배경 및 필요성

영어회화는 어디에서나 중요하게 요구될 수 있는 역량 중 하나다. 하지만 단순히 번역기로 잘못 배운 영어회화는 상황에 맞지 않는 표현을 쓸 확률이 높아 실수를 하게 될 수도 있고, 누군가는 오해를 할 수도 있다. 이런 상황을 해결하기 위해 쉽게 접할 수 있는 생활영어 회화 배우기 프로그램을 만들었다.

2) 프로젝트 목표

사람들이 자주 쓰는 어색한 표현을

올바른 표현 및 상황에 맞는 표현법을 알려주며

고객의 실전 생활영어의 향상을 목표로 한다.

3) 차별점

기존 번역기 같은 경우 한국어로 문장을 입력하면 바로 영어로 번역되지만 단순히 한 문장만 보여주어 그 상황에 맞지 않는 표현을 쓸 수도 있고, 어색한 표현일 수도 있다. 이 문제를 해결하기 위해 상황에 적합한 표현과 어색한 표현 등등, 보여주며 자신의 언어 실수를 쉽게 이해하며 바로잡을 수 있고, 또한 그 나라의 문화를 이해할 수도 있다는 차별점이 있다.

2. 기능 계획

기능1: 영어 회화 배우기 챗터

세부기능1) 상황에 따른 영어 회화 수집

ex) 1. 인사하기 2. 물어보기 등등

세부기능 2) 상황에 따른 영어회화별 세분화하기

Ex)1. 인사하기

1)인사말 2)이름 묻기

기능2: 단어 배우기 챗터

세부기능 1) 자주 실수를 범하는 단어표현 보여주기

예) 1. 약속 2. 가다오다 등등

세부기능2) 정정하며 올바른 표현출력 및 설명

Ex) promise -> plan

구조체를 사용하여 영어회화 및 단어의 데이터를 저장할 예정

함수 계획:

Void printConversation: 대화 출력함수

Void PrintWord: 단어 출력함수

3. 진척사항

1) 기능 구현

study_eng.h 헤더파일

대화 데이터와 단어 데이터에 관련된 구조체 및 함수들을 정의하고 있다. 다른 소스 파일에서 이 구조체 및 함수들을 사용할 수 있도록 선언하고 정의하는 데 사용된다. 이 헤더 파일을 다른 소스 파일에 포함하여 그 안에 정의된 구조체나 함수들을 활용할 수 있다.

(1) 생활영어회화 출력 및 설명

이 구조체는 대화 데이터를 저장하기 위한 것으로, 세가지 멤버변수인 situation, expression, correction을 가지고 있다. 각각은 문자열을 저장할 수 있는 고정된 크기의 배열로 선언되어 있다.

입력: situatin: 영어 회화배우기 (상황) 문자열배열 정의

Expression: 영어 회화배우기 (표현) 문자열 배열 정의

Correction: 영어 회화 배우기(정정) 문자열 배열 정의

Word는 단어를 저장하는 배열, explation은 해당 단어의 설명을 저장하는 배열

배운 내용: 구조체, 변수

```
// 대화 데이터 구조체 정의
struct Conversation {
    char situation[50];
    char expression[100];
    char correction[100];
};

// 단어 데이터 구조체 정의
struct Word {
    char word[50];
    char explanation[100];
};
```

--함수 선언--

saveConversations: 대화 데이터를 저장하는 함수, Conversation 구조체의 배열과 배열의 크기를 인자로 받는다.

Void saveWords: 단어 데이터를 저장하는 함수로, Word 구조체의 배열과 배열의 크기를 인자로 받는다.

Void PirntConversation: 대화 데이터를 출력하는 함수로, Conversation 구조체의 포인터를 받는다.

Void pirntWord: 단어 데이터를 출력하는 함수로, Word 구조체의 포인터를 받는다.

#ifndef, #define, #endif : 헤더 파일의 중복 포함을 방지하기 위한 헤더가드 이다.

```
// 함수 선언
void saveConversations(struct Conversation *conversations, int size);
void saveWords(struct Word *words, int size);
void printConversation(struct Conversation *convo);
void printWord(struct Word *wd);
```

```
// 대화 데이터 초기화
struct Conversation conversations[] = {
    {"인사말", "hey",
     "hey: 친해진 뒤 사용! \nhow are you: 전에 만난 적 있는 사람에게 사용!"},
    {"이름묻기", "hi my name is jina", "hi i'm jina"},
    {"오랜만에 만날때", "nice to meet you", "good to see you again"},
    {"첫 만남 무례한 질문", "how old are you\n\nWhat's your job",
     "정말 물어보고 싶을때->\nMay i ask you age?\n\n what do you do for a "
     "living?"},
    {"친해지기 위해 쓰는 표현", " how old are you ",
     "what do you do in your free time"},
    {"맛이 어떤지 물어보기", "is it delicious?", "how does it taste?"},
    {"주문시 시간이 필요할때", " wait a minute", "atually, i think we need anoter time"}
};
```

헤더파일에 정의된 구조체와 함수들을 구현한 C소스 코드

-대화 출력 함수 및 단어 출력 함수 정의

설명: 이 코드는 study_eng.h에 정의된 구조체인 Conversation과 Word를 활용하여 대화와 단어 데이터를 출력하는 함수들을 구현한다.

convo->situation은 convo가 가리키는 구조체의 situation 멤버에 접근하는 것입니다.

이것은 (*convo).situation과 동일한 의미

배운 것: 함수, 구조체, 포인터

```
// 대화 출력 함수
void printConversation(struct Conversation *convo) {
    printf("---%s---\n", convo->situation);
    printf("%s\n", convo->expression);
    printf("----정정----\n");
    printf("%s\n", convo->correction);
}

// 단어 출력 함수
void printWord(struct Word *wd) {
    printf("---%s---\n", wd->word);
    printf("%s\n", wd->explanation);
}
```

-대화 데이터를 파일에 저장하는 함수 정의

설명 :

파일을 열지 못했을 때 적절한 오류 메시지를 출력하고, 파일을 정상적으로 열었을 경우 해당 데이터를 파일에 쓴다. 마지막으로 파일을 닫아 리소스를 해제한다.

특정 데이터를 콘솔에 출력하거나 파일로 저장하는 함수들을 정의하고 있다.

세부 코드 설명:

SaveConversation함수는 대화 데이터를 Conversations.txt 파일에 저장한다.

fprintf()는 함수는 파일에 형식화된 데이터를 출력하는 데 사용된다.

설명: 구조체를 사용해 초기화한 대화 데이터를 파일에 저장하는 함수이다.

fprintf(): 파일에 데이터를 저장하기

fclose함수: 파일 포인터가 가리키는 파일을 저장하고 닫기

파일모드 w: 쓰기전용, (새 파일 생성, 기존파일이 있으면 덮어 씌)

지금까지 배운 내용:

Fprintf, 함수, 구조체, for문, fopen_s

.

배운 내용: 구조체, 포인터, 파일 입출력

```
// 대화 데이터를 파일에 저장하는 함수
void saveConversations(struct Conversation *conversations, int size) {
    FILE *file;
    fopen_s(&file, "conversations.txt", "w"); // 대화 데이터를 저장할 파일명
    if (file == NULL) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    for (int i = 0; i < size; ++i) {
        fprintf(file, "%s\n%s\n%s\n", conversations[i].situation,
            conversations[i].expression, conversations[i].correction);
    }

    fclose(file);
}
```

단어 데이터를 파일에 저장하는 함수 정의

설명: 특정 데이터를 콘솔에 출력하거나 파일로 저장하는 함수들을 정의하고 있다.

세부 코드 설명 : saveWords()함수는 word 배열을 받아서 파일에 단어 데이터를 저장한다.

Fprintf() 함수는 파일에 형식화된 데이터를 출력하는 데 사용된다.

설명: 구조체를 사용해 초기화한 단어 데이터를 파일에 저장하는 함수이다.

.위 함수는 구조체 포인터를 받아서 해당 데이터를 파일에 저장하게 된다. 포인터는 데이터를 전달하고 함수가 해당 데이터를 조직할 수 있게끔 도와준다.

fprintf() :파일에 데이터를 저장하기

fclose함수: 파일 포인터가 가리키는 파일을 저장하고 닫기

파일모드 **w**: 쓰기전용, (새 파일 생성, 기존파일이 있으면 덮어쓰)

지금까지 배운 내용: 형식을 지정해 파일 입출력하기, **fprintf**, **fopen_s**, 반복문, 함수,

fclose

배운내용 : 구조체, 함수, 파일 입출력

```
// 단어 데이터를 파일에 저장하는 함수
void saveWords(struct Word *words, int size) {
    FILE *file;
    fopen_s(&file, "words.txt", "w"); // 단어 데이터를 저장할 파일명
    if (file == NULL) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    for (int i = 0; i < size; ++i) {
        fprintf(file, "%s\n%s\n", words[i].word, words[i].explanation);
    }

    fclose(file);
}
```

Main.c

이 코드는 생활영어를 학습하는 프로그램으로 특정 대화와 단어를 선택하고 해당 정보를 출력하는 기능을 가지고 있다.

1. 구조체 및 포인터 초기화

Conversations 및 **words**는 각각 대화와 단어를 가리키는 구조체 포인터이다.

동적할당으로 메모리를 할당하고 데이터를 저장할 공간을 마련하는 용도로 선언되었다.

배운내용: 구조체, 포인터

```
// 대화 데이터 초기화 (포인터로 변경)
struct Conversation *conversations;

// 단어 데이터 초기화 (포인터로 변경)
struct Word *words;
```

2. 메모리 동적 할당

Malloc()함수를 사용하여 conversations와 words가 가리키는 메모리 영역에 대화와 단어를 저장할 공간을 할당한다. 대화 데이터에는 9개의 대화가 , 단어 데이터에는 20개의 단어가 저장도리 수 있는 공간을 할당한다.

배운내용: 동적 메모리 할당

```
// main함수
int main() {
    // 대화와 단어 배열에 대한 메모리 동적 할당
    conversation = malloc(sizeof(struct Conversation) * 9);
    words = malloc(sizeof(struct Word) * 20);
```

```
// 메모리 할당이 제대로 이루어졌는지 확인
if (conversations == NULL || words == NULL) {
    printf("메모리 할당에 실패했습니다.\n");
    return 1; // 프로그램 종료, 오류 코드 반환
}
```

구조체배열을 이용해 대화 데이터 및 단어 데이터 초기화

설명: 프로그램이 사용할 대화와 단어 데이터를 초기화하는 역할을 한다. 이러한 초기화를 통해 프로그램이 사용할 데이터가 미리 정의되고, 이후 사용자가 선택한 기능에 따라 이러한 데이터를 활용할 수 있다.

대화 데이터 초기화 설명: 구조체배열을 이용해 conversation구조체 배열을 초기화 한다.

각 요소는 대화 정보를 포함하며 첫번째 필드는 상황을, 두 번째 필드는 표현을 세 번째 필드는 정정을 나타낸다.

단어 데이터 초기화 설명: 구조체 배열을 이용해 Word 구조체 배열을 초기화 하는 부분이다. 각 요소는 단어와 해당 단어의 설명을 나타내며, 첫 번째 필드는 단어를 두 번째 필드는 정정된 단어를 나타낸다.

배운 내용: 구조체 배열

```
// 대화 데이터 초기화
struct Conversation conversations[] = {
    {"인사말", "hey",
     "hey: 친해진 뒤 사용! Wnhow are you: 전에 만난 적 있는 사람에게 "
     "사용!"},
    {"이름물기", "hi my name is jina", "hi i'm jina"},
    {"오랜만에 만날때", "nice to meet you", "good to see you again"},
    {"첫 만남 무례한 질문", "how old are youWnWhat's your job",
     "정말 물어보고 싶을때->WnMay i ask you age?Wn what do you do for a "
     "living?"},
    {"친해지기 위해 쓰는 표현", "how old are you",
     "what do you do in your free time"},
    {"맛이 어떨지 물어보기", "is it delicious?", "how does it taste?"},
    {"주문시 시간이 필요할때", "wait a minute",
     "atually, i think we need anoter time"},
    {"메뉴 결정을 했을때", "I decided", "I made a decision"},
    {"서비스인지 물어보기", "it's service?",
     "it's on us/ it's on the house"};

// 단어 데이터 초기화
struct Word words[] = {
    {"가다 오다",
     "come: 화자와 청자가 가까워질 때 사용Wngo: 반대로 멀어질 때 사용"},
    {"패딩-padding", "puffer jacket"},
    {"우리 동네-my neighboor", "my neighborhood"},
    {"버스 정류장-bus station", "bus stop"},
    {"후드티-hood-T", "hoodie"},
    {"킥보드-kick board", "kick scooter"},
    {"미끄러지기-slide", "scooter"}
```

입력:

choice: 첫 번째 입력을 받기 위한 변수

Choicee: 회화 배우기의 카테고리별 입력을 받기 위한 변수

Choic: 2번째 회화 배우기의 카테고리별 입력을 받기 위한 변수

Cho: 회화 배우기의 카테고리별 입력을 받기 위한 변수

설명:

입력된 값을 받기 위해 choice 정수형 변수로 선언하였다.

Do while 반복문을 사용하여 만약 3종료하기를 누르면 종료된다.

Choicee, choic, cho 도 모두 입력된 값을 받기위한 변수이다.

적용된 배운 내용: 변수선언, 출력함수, 입력함수, do while 반복문

1. 변수 초기화 및 메뉴 출력

입력: 사용자 선택을 저장하는 변수들을 초기화

Choice : 첫 번째 메뉴의 선택을 저장할 변수

Subchoice: 첫 번째 메뉴의 1 번을 선택시 나오는 메뉴의 선택을 저장할 변수

Choicee: 위의 메뉴에서 받은 선택을 저장할 변수

Choic , cho : 각 카테고리 별로 선택을 저장할 변수

출력: printf함수를 사용하여 콘솔에 출력된다.

설명: 위의 번호중에서 1) 회화 배우기 2)단어 배우기 3)종료하기

만약 1을 선택하면 1)인사말, 2)이름 묻기 3)오랜만에 만날 때 번호가 뜨고 거기서 1개를 선택할 수 있다.

코드 설명: siwtch문을 이용해 각각 케이스 별로 입력을 받고 출력을 한다.

Do while반복문을 활용해 사용자가 프로그램을 종료하려고 할 때까지 반복되도록 한다.

적용된 배운 내용: switch 조건문, 변수선언, do while반복문

```

int choice = 0;
int choicee = 0;
int choic = 0;
int cho = 0;
printf("----쉽게 배우는 생활영어----\n\n");
do {
    printf("1) 회화 배우기  2) 단어 배우기  3) 종료하기\n\n");
    scanf_s("%d", &choice);

    switch (choice) {
        // 생활영어 회화 배우기 챕터
        case 1: {
            int subChoice = 0;
            do {
                printf("\n <<생활 영어 회화 배우기>>  \n");
                printf("1) 인사하기  2) 물어보기  3) 주문하기  4) 종료하기\n\n");
                printf("선택: ");
                scanf_s("%d", &subChoice);
            }
        }
    }
}

```

회화 배우기 챕터

쉽게 배우는 생활영어 1. 회화 배우기 선택 시 활용되는 코드

출력: 만약 사용자가 번호를 선택 시 printCoversation의 함수가 불러와져 관련 데이터가 콘솔에 출력된다.

설명: 회화 배우기를 사용자가 선택하게 된다면 인사하기 물어보기 주문하기를 선택할 수 있는 서브 메뉴가 있다. 사용자는 서브 메뉴에서 원하는 옵션을 선택할 수 있다. 선택에 따라 대화 데이터가 출력된다.

배운 내용: 반복문, if 조건문, 포인터

```

printf("----쉽게 배우는 생활영어----\n\n");
do {
    printf("1) 회화 배우기   2) 단어 배우기   3) 종료하기\n\n");
    scanf_s("%d", &choice);

    switch (choice) {
        // 생활영어 회화 배우기 챕터
        case 1: {
            int subChoice = 0;
            do {
                printf("\n <<생활 영어 회화 배우기>>   \n");
                printf("1) 인사하기  2) 물어보기  3) 주문하기  4) 종료하기\n\n");
                printf("선택: ");
                scanf_s("%d", &subChoice);

                if (subChoice == 1) {
                    printf(" ---인사하기--- \n");
                    printf("1) 인사말\n 2) 이름 묻기\n 3) 오랜만에 만날 때\n");
                    scanf_s("%d", &choicee);

                    if (choicee == 1) {
                        printConversation(&conversations[choicee - 1]);
                    } else if (choicee == 2) {
                        printConversation(&conversations[choicee - 1]);
                    } else {
                        printConversation(&conversations[choicee - 1]);
                    }
                }
            }
        }
    }
}

```

```

    } else if (subChoice == 2) {
        printf("---물어보기---\n\n");
        printf(
            "1) 첫 만남 무례한 질문\n 2) 친해지기 위해서 쓰는 표현\n 3) "
            "맛이 어떨지 물어보기\n");
        scanf_s("%d", &choic);

        if (choic == 1) {
            printConversation(&conversations[4 - choic]);
        } else if (choic == 2) {
            printConversation(&conversations[6 - choic]);
        } else {
            printConversation(&conversations[8 - choic]);
        }
    } else if (subChoice == 3) {
        printf("----주문하기----\n\n");
        printf(
            "1) 주문시 시간이 필요할때\n 2) 결정을 했을때\n 3) 서비스인지 "
            "물어보기");
        scanf_s("%d", &cho);

        if (cho == 1) {
            printConversation(&conversations[7 - cho]);
        } else if (cho == 2) {
            printConversation(&conversations[9 - cho]);
        } else if (cho == 3) {
            printConversation(&conversations[11 - cho]);
        } else {
            printf("잘못된 선택입니다.");
            break;
        }
    }
}

```

기능2: 단어배우기 챗터

설명: 일상용어, 건물, 회화할 때 주의할 용어를 선택할 수 있는 서브 메뉴가 있다. 사용자는 서브 메뉴에서 원하는 옵션을 선택할 수 있다. 선택에 따라 단어 데이터가 출력된다.

입력: 사용자가 원하는 메뉴와 서브메뉴에서 숫자를 선택하고 저장된다.

만약 없는 번호의 값을 입력하면 잘못된 선택이라 뜨고 다시 번호를 입력받는다.

출력: word구조체의 주소를 printWord 함수에 전달한다. 여기서 예를 들어 cho+9는 건물 단어들이 words배열의 인덱스 10부터 시작한다고 표시된 것이다.

지금까지 배운 내용: 반복문, 조건문, 구조체 배열, 포인터

```
// 생활영어 단어 배우기 챗터
case 2: {
    int subChoice = 0;
    printf("<< 생활영어 단어 배우기 >> \n");
    do {
        printf(
            "1) 일상 용어   2) 건물   3) 회화할 때 주의할 용어   4) "
            "종료하기\n\n");
        printf("선택: ");
        scanf_s("%d", &subChoice);

        switch (subChoice) {
            case 1: {
                printf(
                    "\n1) 가다오다   2) 패딩   3) 우리동네   4) 버스 정류장   5) "
                    "후드티\n");
                printf(
                    "6) 킥보드   7) 미혼   8) 미팅   9) 요리사   10) 토스트\n");
                printf("선택: ");
                scanf_s("%d", &cho);

                if (cho >= 1 && cho <= 10) {
                    printWord(&words[cho - 1]);
                } else {
                    printf("잘못된 선택입니다. 다시 선택하세요.\n");
                }
            }
        }
    } while (subChoice != 0);
}
```

```

        printf("잘못된 선택입니다. 다시 선택하세요.\n");
    }
    break;
}
case 2: {
    printf(
        "1) 화장실 2) 아파트 3) 원룸 4) 콘도 5) 빌라/맨션\n");
    printf("선택: ");
    scanf_s("%d", &cho);

    if (cho >= 1 && cho <= 5) {
        printWord(&words[cho + 9]); // 인덱스 10부터 건물 단어 시작
    } else {
        printf("잘못된 선택입니다. 다시 선택하세요.\n");
    }
    break;
}
case 3: {
    printf(
        "1) 한 시간 2) 이틀 전에 3) 10분 후에 4) 그냥 5) "
        "한국 사람들\n");
    printf("선택: ");
    scanf_s("%d", &cho);

    if (cho >= 1 && cho <= 5) {
        printWord(&words[cho + 14]); // 인덱스 15부터 회화할 때 주의할
        // 용어 시작
    } else {
        printf("잘못된 선택입니다. 다시 선택하세요.\n");
    }
}

```

프로그램 종료하기

설명: 첫번째 메뉴로 돌아가서 사용자가 3번을 선택한다면 위 프로그램은 종료된다.

코드 세부 설명: While문: 3번을 입력 받았기에 반복문은 조건에 따라 종료된다.

출력: printf 함수를 통해 문장이 콘솔에 출력된다.

```

case 3:
    printf(" 프로그램을 종료합니다.\n");
    break;
default:
    printf("잘못된 선택입니다. 다시 선택하세요.\n");
}
while (choice != 3);

```

프로그램 종료될 때 현재 메모리에 저장된 대화와 단어 데이터를 파일로 저장하는 역할을 한다.

설명: 함수를 호출하여 대화 데이터와 단어 데이터를 파일에 저장하는 역할을 한다.

배열의 요소 수를 자동으로 계산하여 함수에 전달하여 코드를 더 유연하고 일반화된 방식으로 작성하는 방법입니다. 즉, 배열 크기가 변경되더라도 이 코드 부분은 자동으로 새로운 배열 크기에 맞추어 동작함

코드 세부 설명:

saveConversations 함수는 대화 데이터를 파일에 저장하는 함수, 이 함수는 conversation 배열의 크기를 계산하여 그 크기에 맞게 파일에 데이터를 저장한다.

saveWords함수는 단어 데이터를 파일에 저장하는 함수, 이 함수는 words배열의 크기를 계산하여 그 크기에 맞게 파일에 데이터를 저장한다.

위의 코드는 배열의 크기를 계산하는 방법으로 `sizeof(array) / sizeof(array[0])`

를 사용한다. 이 방법은 배열의 총 요소 수를 해당 데이터 타입의 크기로 나누어 배열의 크기를 계산한다.

- ➔ 프로그램 종료 시에 데이터를 영구적으로 보존하기 위해 파일에 저장하는 중요한 단계이다. 저장된 데이터는 이후에 프로그램을 실행하거나 다른 곳에서 읽을 수 있도록 보존된다.

지금까지 배운 내용: 함수호출

```
// 프로그램 종료 시 대화와 단어 데이터를 파일에 저장
saveConversations(conversations,
                  sizeof(conversations) / sizeof(conversations[0]));
saveWords(words, sizeof(words) / sizeof(words[0]));
```

프로그램이 종료되기전에 동적으로 할당된 메모리를 해제하는 역할

코드 설명: free()함수를 호출함으로써 conversation와 words가 가리키고 있는 메모리 블록이 해제되어, 해당 메모리 공간은 이후 다른 목적으로 재사용 될 수 있게 된다.

```
// 동적으로 할당한 메모리 해제
free(conversations);
free(words);
```


함수의 구조체 배열을 포인터로 변경하기

설명: 함수의 매개변수로 구조체 포인터를 받도록 변경 메인 함수 내에서 이러한 함수들을 호출할 때 구조체 배열 대신 포인터를 전달해야 된다.

예를들어

`printConversation(conversations[choi-1];`를 `printConversation(&conversations[choicee -1];`로 변경해야 한다.

`convo->situation`은 `convo`가 가리키는 구조체의 `situation` 멤버에 접근하는 것입니다.

이것은 `(*convo).situation`과 동일한 의미를 가지고 있다.

배운 내용: 함수의 구조체 배열을 포인터로 변경하기

```
// 함수 선언
void saveConversations(struct Conversation *conversations, int size);
void saveWords(struct Word *words, int size);
void printConversation(struct Conversation *convo);
void printWord(struct Word *wd);
```

2) 테스트 결과

(1) 생활 영어회화 배우기 챗터

- 첫 화면 1, 2, 3 중에 고를 수 있다.

```
----쉽게 배우는 생활영어----  
1) 회화 배우기    2) 단어 배우기    3) 종료하기
```

-

1. 회화 배우기 챗터

-1을 선택시

```
<<생활 영어 회화 배우기>>  
1) 인사하기  2) 물어보기  3) 주문하기  4) 종료하기  
선택: |
```

```
선택: 1  
---인사하기---  
1) 인사말  
2) 이름물기  
3) 오랜만에 만날 때  
|
```

인사말인 경우

```
1  
-----인사말-----  
hey  
----정정----  
hey: 친해진 뒤 사용!  
how are you: 전에 만난 적 있는 사람에게 사용!
```

이름 묻기 경우

```
2
---이름 묻기---
hi my name is jina
----정정----
hi i'm jina
```

오래만에 만날 때 경우

```
3
---오랜만에 만날때---
nice to meet you
----정정----
good to see you again
```

2 물어보기 선택시

```
선택: 2
---물어보기---
```

1) 첫 만남 무례한 질문 2) 친해지기 위해서 쓰는 표현 3) 맛이 어떤지 물어보기

-1을 선택시

```
-----첫 만남 무례한 질문-----
how old are you
What's your job
----정정----
정말 물어보고 싶을때->
May i ask you age?
what do you do for a living?
```

-2를 선택시

```
-----친해지기 위해 쓰는 표현---
how old are you
----정정----
what do you do in your free time
```

2

3을 선택시

```
---맛이 어떤지 물어보기---  
is it delicious?  
----정정----  
how does it taste?
```

3 주문하기 선택시

```
선택: 3  
---주문하기---  
  
1) 주문시 시간이 필요할 때  
2) 결정을 했을 때  
3) 서비스인지 물어보기
```

-1을 선택시

```
선택: 3  
---주문하기---  
  
1) 주문시 시간이 필요할 때1  
---주문시 시간이 필요할 때---  
wait a minute  
----정정----  
atually, i think we need anoter time
```

결정을 했을 때 인경우

```
3) 서비스인지 물어보기2  
---메뉴 결정을 했을 때---  
I decided  
----정정----  
I made a decision
```

서비스인지 물어보기인 경우

```
3) 서비스인지 물어보기3  
---서비스인지 물어보기---  
it's service?  
----정정----  
it's on us/ it's on the house
```

4 종료하기 선택시

```
<<생활 영어 회화 배우기>>  
1) 인사하기 2) 물어보기 3) 주문하기 4) 종료하기  
  
선택: 4  
1) 회화 배우기 2) 단어 배우기 3) 종료하기
```

기능 2) 단어 배우기 챗터

-2를 선택시

```
선택: 4  
1) 회화 배우기 2) 단어 배우기 3) 종료하기  
  
2  
<< 생활영어 단어 배우기 >>  
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기  
  
선택: |
```

-1을 선택 시

```
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기  
  
선택: 1  
  
1) 가다오다 2) 패딩 3) 우리동네 4) 버스 정류장 5) 후드티  
6) 킥보드 7) 미혼 8) 미팅 9) 요리사 10) 토스트  
선택: |
```

가다오다

```
선택: 1
---가다 오다---
come: 화자와 청자가 가까워질 때 사용
go: 반대로 멀어질 때 사용
```

패딩

```
선택: 2
---패딩-padding---
puffer jacket
```

우리동네

```
선택: 3
---우리 동네-my neighboor---
my neighborhood
```

버스 정류장

```
선택: 4
---버스 정류장-bus station---
bus stop
```

후드티

```
선택: 5
---후드티-hood-T---
hoodie
```

킥보드

```
선택: 6
---킥보드-kick board---
kick scooter
```

미혼

```
선택: 7
---미혼- solo---
single
```

미팅

```
선택: 8
---미팅-meeting---
blind date
```

요리사

```
선택: 9
---요리사-cooker---
cook
```

토스트

```
0) 크로issant 7) 미혼
선택: 10
---토스트-toast---
grilled sandwich
```

2를 선택시 건물 용어

1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기

선택: 2

1) 화장실 2) 아파트 3) 원룸 4) 콘도 5) 빌라/맨션

선택: |

1 화장실

```
선택: 1
---화장실-toilet---
bathroom
```

1) 일상 용어 2) 건물

2 아파트

```
1) 화장실    2) 아파트    3) 원룸
선택: 2
---아파트-apart---
apartment/building/complex
```

1) 일상 용어 2) 건물

3 원룸

```
선택: 3
---원룸-one room---
studio apartment
```

4 콘도

```
선택: 4
---콘도-condo---
hotel
```

5 빌라/맨션

```
선택: 5
---빌라-villa/mansion---
apartment
```

3을 선택 시 회화할 때 주의할 용어

```
gritted sandwich
1) 일상 용어    2) 건물    3) 회화할 때 주의할 용어    4) 종료하기

선택: 3
1) 한 시간    2) 이틀 전에    3) 10분 후에    4) 그냥    5) 한국 사람들
선택: |
```

1 한시간

```
선택: 1
---한시간 - a hour ---
an hour
```

2 이틀 전에

```
---이틀전 에-two days before---
two days ago
```


3 10분전에

```
---10분 후에-after 10 minutes---  
in 10 minutes  
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기
```

4 그냥

```
---그냥-just---  
just because  
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기
```

5 한국사람들

```
---한국사람들-Korean people---  
Koreans  
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기
```

종료하기 선택시

```
apartment  
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기  
선택: 4  
프로그램을 종료합니다.  
1) 회화 배우기 2) 단어 배우기 3) 종료하기  
선택: 3  
프로그램을 종료합니다.  
C:\Users\LG\source\repos\Project82\x64\Debug\Project82.exe(프로세스 2342  
8개)이(가) 종료되었습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 누르세요...|
```

4. 계획 대비 변경 사항

1) 어색한 상황의 대화 대신 정확한 단어와 표현만 보여주기

- 이전: 회화 배우기 챗터와, 단어배우기 챗터에서 어색한 표현 법을 쓰는 대화를 출력한다.
- 이후: 대화 대신 어색한 단어나 문장만 쓴다.
- 사유: 공간 부족으로 인한 문제점, 어색하거나 잘못된 표현을 사용한 대화를 계속해서 보여주면 뻘한 내용만 나오기 때문에 대화를 삭제했다.

2) 단어를 입력 받는 대신 자주 실수하는 단어표현 몇 가지 보여주기

- 이전: 고객이 단어를 입력하면 상황에 따른 비슷한 단어 사용을 추천하며 예시 문장을 보여준다.
- 이후: 대부분이 실수할 수 있는 단어 표현만 몇 가지 보여주고, 그에 따라서 정확한 표현을 보여준다.
- 사유: 공간 부족으로 인한 문제점

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/6	11/12	11/19	11/24	11/29	12/1~12/15	12/22
제안서 작성		완료							
기능1	세부기능1		진행중						
	세부기능2			진행중					
기능2	세부기능1				진행중				
	세부기능2					진행중			
중간 구현물							완료		
중간 보고서							완료		
최종 구현물 및 보고서								----->	