

C프로그래밍 실습

영어단어 게임 제작 프로젝트

최종 보고서

2023.12.24

김리하

204201

1. 프로젝트 목표

1) 배경 및 필요성

스마트폰이 일상화된 현재 비생산적인 스마트폰 이용이 늘어났다. 스마트폰을 이용하는 시간에 사용자에게 조금 더 생산적인 활동을 제공하기 위해 영단어 게임 제작을 결정함.

2) 프로젝트 목표

가볍게 플레이가 가능한 간단한 시스템의 영어 단어 외우기 게임으로 사용자가 일상 속에서 자연스럽게 플레이 하는 것을 목표로 함.

3) 차별점

기존 단어 외우기 게임들의 복잡하거나 사용 빈도가 낮은 기능들은 전부 제거하고 가볍고 필요한 기능들만 사용자의 피로도를 줄였음.

2. 기능 계획

1) 영어 단어 퀴즈 기능

- 무작위 영어 단어를 출력하여 뜻을 입력 받아 결과를 출력하는 기능

2) 오늘의 영어 질문

- 무작위 영어 질문이 제시되고 유저가 자신의 생각을 입력한다.
- txt 파일에 저장되어 유저는 자신의 과거 답변을 확인할 수 있다.
- 질문 예시) Which season do you like best? Why?

3. 기능 구현

(1) 영어 단어 퀴즈

- 입출력

출력: 무작위 영어 단어

입력: 영어 단어의 뜻

출력: 정답 결과

- 설명

무작위 영단어를 출력하면 유저는 답을 입력한다.

입력을 받으면 정답 결과를 출력한다.

오답일 시 정답을 같이 출력한다.

- 적용된 배운 내용

구조체: 단어 데이터 저장

함수: 단어 게임 실행을 함수로 제작. 랜덤 함수로 단어를 무작위로 출력

If문: 정답 구별을 위해 사용

- 코드 스크린샷

```
// 단어 게임 실행 함수 (메뉴 1번)
void word_quiz() {
    printf("영어 단어의 뜻을 맞춰보세요! \n");

    // 문제 개수 입력받기
    int questions_num;

    printf("몇 문제를 풀실건가요?: ");
    scanf_s("%d", &questions_num);

    // 맞은 문제 개수를 세는 변수
    int correct = 0;

    // 입력받은 수 만큼 단어 퀴즈 반복
    for (int i = 0; i < questions_num; i++) {

        int randomIndex = rand() % WORD_COUNT;
        char guess[MAX_WORD_LENGTH];

        printf("%s\n", dictionary[randomIndex].word);
        printf("뜻을 입력하세요: ");
        scanf_s("%s", guess, (int)sizeof(guess));

        if (strcmp(guess, dictionary[randomIndex].meaning) == 0) {
            printf("정답입니다!\n");
            correct++;
        }
        else {
            printf("틀렸습니다. 정답은 '%s'입니다.\n", dictionary[randomIndex].meaning);
        }

        printf("\n\n수고하셨습니다!\n\n %d문제 중 %d개를 맞췄어요!\n\n", questions_num, correct);
        printf("메인 메뉴로 돌아갑니다.\n\n");
    }
}
```

(2) 오늘의 영어 질문 출력

- 입출력

출력: 무작위 영어 질문

입력: 질문에 대한 답변

- 설명

유저에게 저장된 무작위 질문을 출력한다.

유저에게 답변을 입력 받는다.

질문과 답변은 txt 파일에 저장된다.

- 적용된 배운 내용

구조체: 질문 저장에 사용.

함수: 랜덤 함수 사용. 질문 출력을 함수로 제작.

파일: 질문 저장용으로 파일 사용

- 코드 스크린샷

```
// 영어 질문 실행 함수 (메뉴 2번)
void daily_questions() {

    // 오늘 날짜 출력 ex) 12월 10일의 질문
    current_time();
    void daily_questions()

    // 답변 저장용 리스트 초기화
    char save_answer[100] = "a";

    // questions_data에서 무작위 질문 출력 후 입력받기
    int randomIndex = rand() % QUESTION_COUNT;
    printf("%s\n", questions_data[randomIndex].eng_questions);
    getchar();
    scanf_s("%[^\n]s", save_answer, (int)sizeof(save_answer));

    // 답변 저장용 파일 생성 or 이어쓰기
    FILE* fp;
    fopen_s(&fp, "save_answer.txt", "a");

    if (fp == NULL) {
        printf("open 실패\n");
        return 0;
    }

    // 파일이 열리면 질문과 답변 저장
    while (1) {
        fputs(questions_data[randomIndex].eng_questions, fp);
        fputs(" ", fp);
        fputs(save_answer, fp);
        fputs("\n", fp);
        break;
    }

    fclose(fp);

    printf("저장되었습니다.\n메인 메뉴로 돌아갑니다.\n\n");
}
```

(3) 저장된 답변 출력

- 입출력

출력: txt 파일에 저장된 유저의 답변

- 설명

txt 파일에 저장된 유저의 답변을 출력한다.

- 적용된 배운 내용

파일: 저장된 데이터를 읽어오기 위해 사용

함수: 질문 출력 기능을 함수로 제작.

- 코드 스크린샷

```
void show_past_answer() {  
    // 답변을 불러올 때 사용할 임시 리스트  
    char show_ans[1000];  
  
    FILE* fp;  
  
    fopen_s(&fp, "save_answer.txt", "r");  
  
    if (fp == NULL) {  
        printf("opne 실패\n");  
        return 0;  
    }  
  
    // txt 파일에 저장된 질문과 답변을 불러와서 출력  
    while (1) {  
        fgets(show_ans, 1000, fp);  
        printf("%s\n", show_ans);  
        if (feof(fp)) break;  
    }  
  
    fclose(fp);  
}
```

4. 테스트 결과

(1) 영어 단어 퀴즈

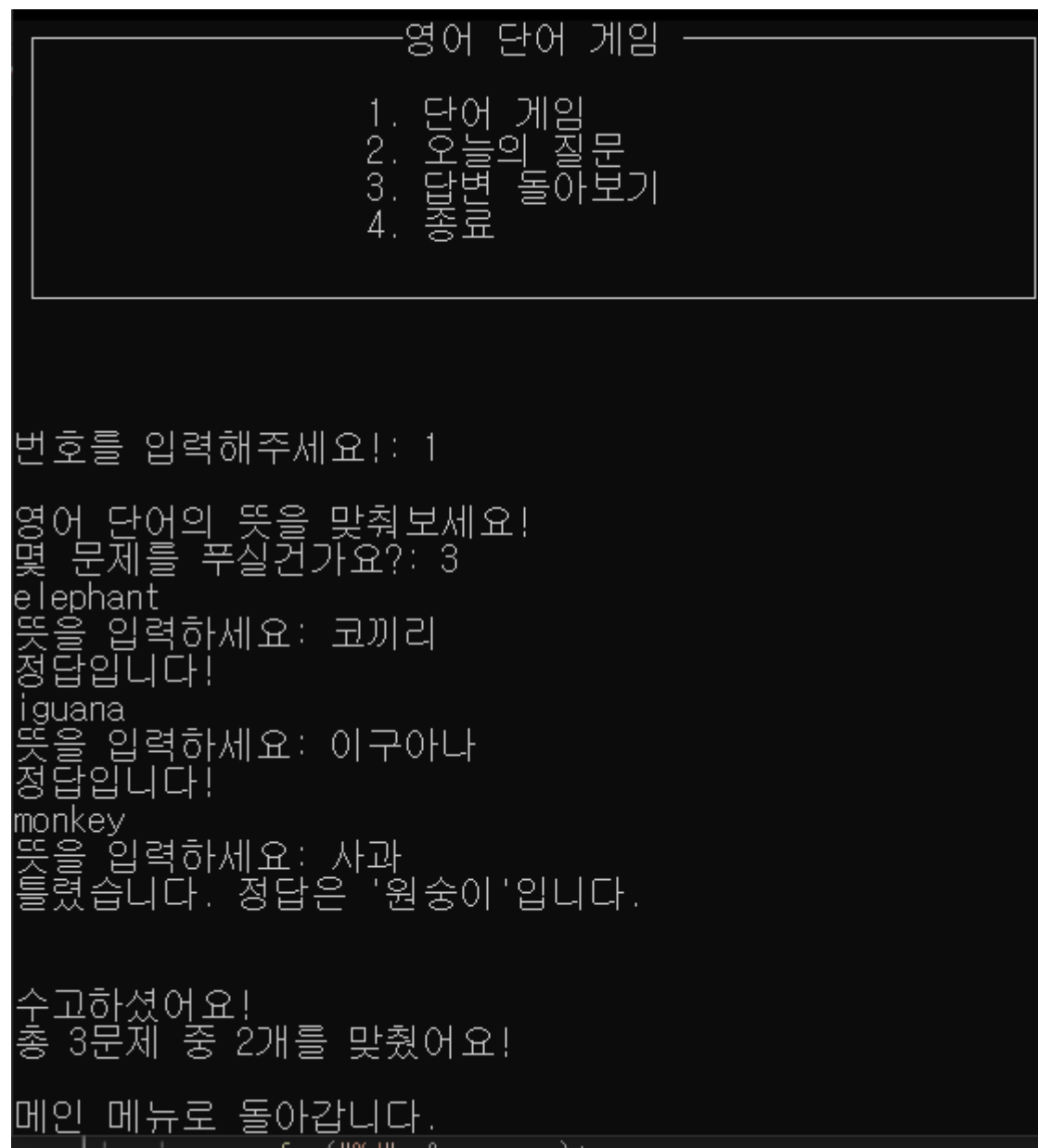
- 설명

무작위 영단어를 출력하면 유저는 답을 입력한다.

입력을 받으면 정답 결과를 출력한다.

오답일 시 정답을 같이 출력한다.

- 테스트 결과 스크린샷



(2) 오늘의 영어 질문 출력

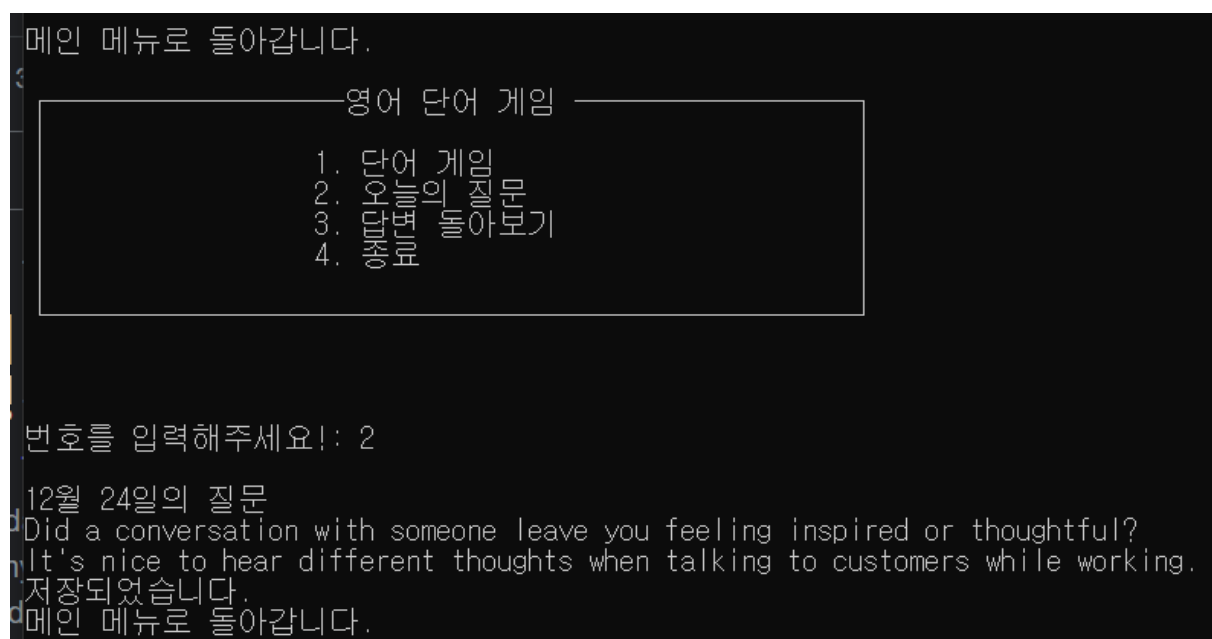
- 설명

유저에게 저장된 무작위 질문을 출력한다.

유저에게 답변을 입력 받는다.

질문과 답변은 txt 파일에 저장된다.

- 테스트 결과 스크린샷

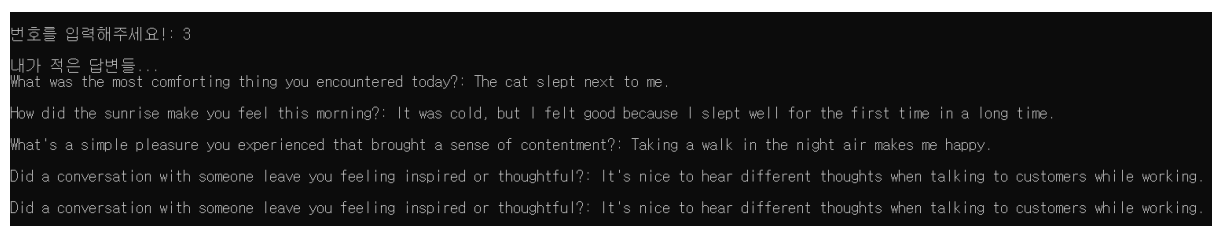


(3) 저장된 답변 출력

- 설명

txt 파일에 저장된 유저의 답변을 출력한다.

- 테스트 결과 스크린샷



5. 계획 대비 변경 사항

1) 변경 내역 제목

- 이전: 퀴즈의 기록을 저장하여 통계 확인 가능
- 이후: 기록 저장 기능 미제작
- 사유: 각 단어의 개수도 많고 그에 따른 데이터 조작이 힘들어 제작을 취소했습니다.

6. 느낀점

파이썬과 자바 기초 수업을 들은 후 수강한 수업이었다. C언어는 처음이었지만 기초 수업이 도움이 되어서 초반에는 무리가 없었다. 하지만 중반부터 어려움을 겪었고, 여러 기능을 사용하는 개인 프로젝트가 처음이라 많은 시행착오가 있었다.

프로젝트를 통해 여러 기능을 사용하고 깃을 이용하면서 전에 수강했던 기초수업보다 많은 걸 얻어갔다. 아직 멀었지만, 껍데기만 아는 느낌에서 프로그래밍 언어를 더 자세히 알 수 있었다.