C프로그래밍및실습

닉네임 자동 추천 프로그램 개발

최종 보고서

제출일자: 2023-12-24
제출자명: 채수아
제출자학번: 231126

1. 프로젝트 목표

1) 배경 및 필요성

각종 사이트에 가입하거나 게임 프로필을 처음 만들 때, 사용자의 닉네임이 필요할 때가 있다. 그런데 보통 닉네임은 다른 사람과 중복으로 지을 수 없다. 자신이 하고싶은 닉네임은 항상 다른 사람이 먼저 사용중인 경우가 허다하다. 때문에 우리는 매번 닉네임을 어떻게 깔끔하게 변형해야 할지 고민에 빠지게 된다. 이 문제를 해결하기 위해, 큰 고민 없이 원래 자신의 닉네임과 몇 가지 취향을 입력하면 자동으로 새로운 닉네임을 추천해 주는 프로그램을 개발할 필요가 있다.

2) 프로젝트 목표

사용자의 원래 닉네임과 취향을 수집하여, 사용자의 큰 고민이 필요 없도록 깔끔한 닉네임을 몇 가지 추천해주는 프로그램을 만드는 것.

3) 차별점

기존의 일부 게임들은 사용자가 입력한 닉네임이 이미 존재할 경우 다른 닉네임을 추천해주는 기능이 있다. 그런데 이들은 아예 관련 없는 닉네임을 추천하거나, '닉네임123', 'a닉네임'과 같은 식으로 의미 없는 변형을 하곤 했다. 하지만 우리가 개발할 프로그램은 이런 수준에 그치지 않고, 사용자가 최대한 선호하는 느낌을 살려 닉네임을 추천해준다는 부분에서 차별점이 있다.

2. 기능 계획

1) 사용자의 닉네임과 취향 수집

- 사용자가 사용하고 싶어하는 기존의 닉네임을 수집하고, 어떤 느낌의 닉네임으로 변형을 원하는지를 묻는다.

2) 사용자의 새 닉네임 생성

- 사용자의 새로운 닉네임을 여러 개 만든다.

(1) 사용자의 닉네임과 수식어의 조합

- 취향 별 닉네임 스타일을 분류하여 사용자 닉네임에 추가할 수식어를 미리 준비하여, 사용자의 닉네임과 조합하고 최종적으로 닉네임을 완성한다.

3) 사용자의 마음에 들지 않을 경우 닉네임을 재생성

- 추천된 닉네임이 사용자의 마음에 들지 않으면 닉네임을 다시 조합하여 만들어 준다.

3. 기능 구현

(1) 사용자의 닉네임과 취향 수집

- 입출력

사용자 자신의 닉네임을 nickname배열에 입력한다. 사용자가 종류 선택을 위해 1~3 범위의 정수를 변수 choice에 입력한다. 사용자가 선택한 스타일에 따라 이후 해당하는 함수를 실행한다.

- 설명

코드가 실행되면 사용자는 자신의 닉네임을 입력한다. 그리고 닉네임 스타일에 따른 메뉴가 사용자에게 보인다. 사용자의 선택을 토대로 이후에 닉네임이 새로 만들어지게 된다.

- 적용된 배운 내용

사용자의 닉네임을 담는 1차원 배열 nickname이 사용되었다. 문자열 메뉴를 출력하기 위해 2차원 배열 name_style이 사용되었다. if문을 사용하여 사용자가 입력한 것을 판별하고 예외 처리한다. 그리고 while문을 사용하여 사용자가 올바른 입력을 할 때까지 버퍼를 제거한다.

- 코드 스크린샷

```
while (1) {
    printf(
        "============================================================₩n₩n"
        "영문 닉네임 자동 추천 프로그램, '뉴네임메이커'가 실행되었습니다.₩n"
        "사용하고자 하는 자신의 고유 영문 닉네임을 입력해주세요. (최대 "
        "%d자)₩n₩n"
        "============================================================₩n₩n",
        MAX_NICKNAME_LENGTH);

    // 구조체 포인터를 사용
    NICKNAME *nickname_maker = (NICKNAME *)malloc(sizeof(NICKNAME));

    if (nickname_maker == NULL) {
        return 1;
    }

    // 사용자에게 닉네임을 입력받음
    printf("자신의 닉네임: ");
    char temp[MAX_NICKNAME_LENGTH + 1];
    scanf_s("%20s", temp, (int)sizeof(temp));
    char ch = getchar();

    nickname_maker->user_nickname =
        (char *)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(nickname_maker->user_nickname, strlen(temp) + 1, temp);

    printf("₩n입력된 닉네임: %s₩n₩n", nickname_maker->user_nickname);
    printf(
        "당신은 어떤 스타일의 닉네임을 선호하시나요?₩n"
        "다음의 종류 중 원하는 것을 선택하세요.₩n₩n"
        "1. %s  2. %s  3. %s₩n₩n",
        name_style[0], name_style[1], name_style[2]);
```

```
        printf("종류 선택 (1-3): ");

        // 사용자가 선호하는 스타일을 선택, 사용자가 1, 2, 3이 아닌 다른 것을
        // 입력할 시 예외 처리
        if (scanf_s("%d", &choice) != 1 || choice < 1 || choice > 3) {
            printf("₩n잘못된 선택입니다. 1부터 3까지의 숫자를 선택해주세요.₩n₩n");
            while (getchar() != '₩n')
                ;
        } else
            break;
    }

    printf("₩n선택된 종류: %d. %s₩n₩n", choice, name_style[choice - 1]);
```

(2) 사용자의 새 닉네임 생성

- 입출력

함수 *NameMakerChar()에 사용자의 닉네임, 꾸밀 문자들을 보내고 조합하여 새 닉네임을 만들어낸다. 함수 *NameMakerStr()에 사용자의 닉네임, 꾸밀 문자열들, 문자열의 개수를 보내고 조합하여 새 닉네임을 만들어낸다.

- 설명

사용자가 입력한 닉네임을 받아서 새로운 닉네임을 만든다. 사용자가 선택한 스타일에 따라 해당하는 함수를 실행한다. 사용자가 1번 또는 3번을 골랐다면 함수 *NameMakerChar()를 실행하여 사용자의 닉네임 글자 수를 세고, 랜덤한 자리에 랜덤한 문자를 집어넣어 새로운 닉네임을 만든다. 사용자가 2번을 골랐다면 함수 *NameMakerStr()를 실행하여 result에 랜덤한 문자열을 복사하고, 사용자의 닉네임과 연결하여 새로운 닉네임을 만든다. 3개의 닉네임을 만들어서 출력하며, 이후 함수 FreeMemory()를 실행하여 할당된 메모리를 해제한다.

- 적용된 배운 내용

for문으로 닉네임을 총 3개 반복하여 만든다. If문을 사용하여 사용자의 선택에 따라 각 함수를 실행한다. 그리고 for문으로 3개의 닉네임을 모두 출력한다. 이때 if문을 사용하여 새 닉네임이 NULL이 아닌 경우에만 출력한다. 랜덤으로 난수를 만들기 위해서 라이브러리 <stdlib.h>와 <time.h>를 불러왔고, 문자열의 문자 수를 세는 strlen() 함수를 사용하기 위해 라이브러리 <string.h>를 불러왔다. 또한 구조체를 정의하여 사용자가 입력한 닉네임을 담을 수 있게 했다.

문자 요소로 새 닉네임을 만드는 함수 *NameMakerChar(char *nickname, char *name_deco)는 포인터 매개변수를 사용한다. rand() 함수를 사용하여 랜덤한 위치와 랜덤한 문자를 선택한다. malloc() 함수를 사용하여 result 변수에 메모리를 동적으로 할당한다. strlen() 함수를 사용하여 문자열의 크기를 잰다. strcpy_s() 함수를 사용하여 문자열을 복사한다. 이때 새 문자를 삽입하기 위해 for문을 사용하여 문자열을 한 칸씩 이동시킨다. 메모리가 잘 할당되었는지 확인하기 위해 if문을 사용했다.

문자열 요소로 새 닉네임을 만드는 함수 *NameMakerStr(char *nickname, char (*name_deco)[MAX_NAME_DECO_LENGTH], int rows)는 포인터 매개변수를 사용한다. rand() 함수를 사용하여 랜덤한 문자열을 선택한다. malloc() 함수를 사용하여 result 변수에 메모리를 동적으로 할당한다. strlen() 함수를 사용하여 문자열의 크기를 잰다. strcpy_s() 함수와 strcat_s() 함수를 사용하여 문자열을 복사하고 연결한다. 메모리가 잘 할당되었는지 확인하기 위해 if문을 사용했다.

할당된 메모리를 해제하는 함수 FreeMemory(char **new_nicknames)는 이중 포인터 매개변수를 사용한다. for문을 사용하여 할당된 메모리를 순회하고 free() 함수를 사용하여 할당된 메모리를 해제한다.

- 코드 스크린샷

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

// 원활한 유지보수를 위해 상수 정의
#define NUM_NICKNAMES 3
#define MAX_NICKNAME_LENGTH 20
#define MAX_NAME_STYLE_LENGTH 50
#define MAX_NAME_DECO_LENGTH 20
#define MAX_NEW_NICKNAME_LENGTH 30
#define NUM_NAME_STYLE 3
#define NUM_NAME_DECO2 18

// 함수 원형 선언
char *NameMakerChar(char *nickname, char *name_deco);
char *NameMakerStr(char *nickname, char *name_deco, int rows);
void FreeMemory(char **new_nicknames);
```

```c
    // 사용자의 선택에 따라 각 함수를 실행
    for (int i = 0; i < NUM_NICKNAMES; i++) {
        if (choice == 1) {
            new_nicknames[i] = NameMakerChar(nickname, name_deco1);
        } else if (choice == 2) {
            new_nicknames[i] = NameMakerStr(nickname, name_deco2, 18);
        } else if (choice == 3) {
            new_nicknames[i] = NameMakerChar(nickname, name_deco3);
        }
    }

    // 새로 만들어진 닉네임 3개를 순서대로 출력
    for (int i = 0; i < NUM_NICKNAMES; i++) {
        if (new_nicknames[i] != NULL) {
            printf("새 닉네임 %d: %s₩n", i + 1, new_nicknames[i]);
        }
    }

    FreeMemory(new_nicknames);  // 할당된 메모리 해제
```

```c
// 문자 요소로 새 닉네임을 만드는 함수
char *NameMakerChar(char *nickname, char *name_deco) {
    int name_len = strlen(nickname);
    int deco_len = strlen(name_deco);
    int position = rand() % name_len;
    int rand_deco = rand() % deco_len;
    int size = strlen(nickname) + 2;
    char *result = (char *)malloc(size);  // 동적 할당
    char new_nickname[MAX_NEW_NICKNAME_LENGTH];
    strcpy_s(new_nickname, MAX_NEW_NICKNAME_LENGTH, nickname);

    if (result == NULL) {
        return NULL;
    }

    // 문자열의 끝부터 시작하여 position까지 한칸씩 문자열 이동, 새 문자가 들어갈
    // 자리를 만들기 위함
    for (int i = name_len; i >= position; i--) {
        new_nickname[i + 1] = new_nickname[i];
    }

    // 랜덤한 자리에 랜덤한 문자를 삽입하고 result에 복사함
    new_nickname[position] = name_deco[rand_deco];
    strcpy_s(result, size, new_nickname);

    return result;
}
```

```c
// 문자열 요소로 새 닉네임을 만드는 함수
char *NameMakerStr(char *nickname, char (*name_deco)[MAX_NAME_DECO_LENGTH],
                   int rows) {
    int rand_deco = rand() % rows;
    int size = strlen(nickname) + strlen(name_deco[rand_deco]) + 1;
    char *result = (char *)malloc(size);  // 동적 할당

    if (result == NULL) {
        return NULL;
    }

    // 랜덤한 문자열을 복사하고 사용자의 닉네임과 붙임
    strcpy_s(result, size, name_deco[rand_deco]);
    strcat_s(result, size, nickname);

    return result;
}

// 할당된 메모리를 해제하는 함수
void FreeMemory(char **new_nicknames) {
    for (int i = 0; i < NUM_NICKNAMES; i++) {
        free(new_nicknames[i]);
        new_nicknames[i] = NULL;
    }
}
```

(3) 사용자의 마음에 들지 않을 경우 닉네임 재생성

- 입출력

사용자의 입력에 따라 프로그램을 종료할지 다시 반복할지 정해진다.

- 설명

사용자가 0을 입력하면 닉네임을 다시 만들 수 있고, 1을 입력하면 프로그램이 종료된다.

- 적용된 배운 내용

if문을 사용하여 사용자가 프로그램 종료를 위한 변수 end에 입력한 것을 판별하고 예외 처리한다. 그리고 while문을 사용하여 사용자가 올바른 입력을 할 때까지 버퍼를 제거한다. 이때 이 부분에 while문 무한루프를 사용하여 사용자가 올바른 입력을 할 때까지 질문을 반복한다. if문을 사용하여 사용자가 0을 입력했다면 다시 닉네임을 만들고, 사용자가 1을 입력했다면 main 전체에 걸려있는 while 무한루프를 빠져나와 프로그램을 종료한다.

- 코드 스크린샷

```
97      int end;  // 프로그램 종료를 위한 변수
98
99      while (1) {
100       printf(
101           "₩n새 닉네임이 마음에 들지 않는 경우 0을 입력하여 닉네임을 처음부터 "
102           "다시 만들 수 "
103           "있습니다.₩n"
104           "새 닉네임이 마음에 든다면 1을 입력하여 프로그램을 "
105           "종료하세요.₩n₩n");
106       printf("0 또는 1을 선택하세요: ");
107
108       // 사용자가 프로그램을 종료할지 선택함, 사용자가 0, 1이 아닌 다른 것을
109       // 입력할 시 예외 처리
110       if (scanf_s("%d", &end) != 1 || (end != 0 && end != 1)) {
111         printf("₩n잘못된 선택입니다. 0 또는 1을 선택해주세요.₩n");
112         while (getchar() != '₩n')
113           ;
114       } else
115         break;
116     }
117
118     // 사용자가 0을 입력했다면 다시 전체 반복, 1을 입력했다면 프로그램 종료
119     if (end == 0) {
120       printf("₩n");
121       getchar();
122       continue;
123     } else if (end == 1)
124       break;
125   }
126
127   return 0;
```

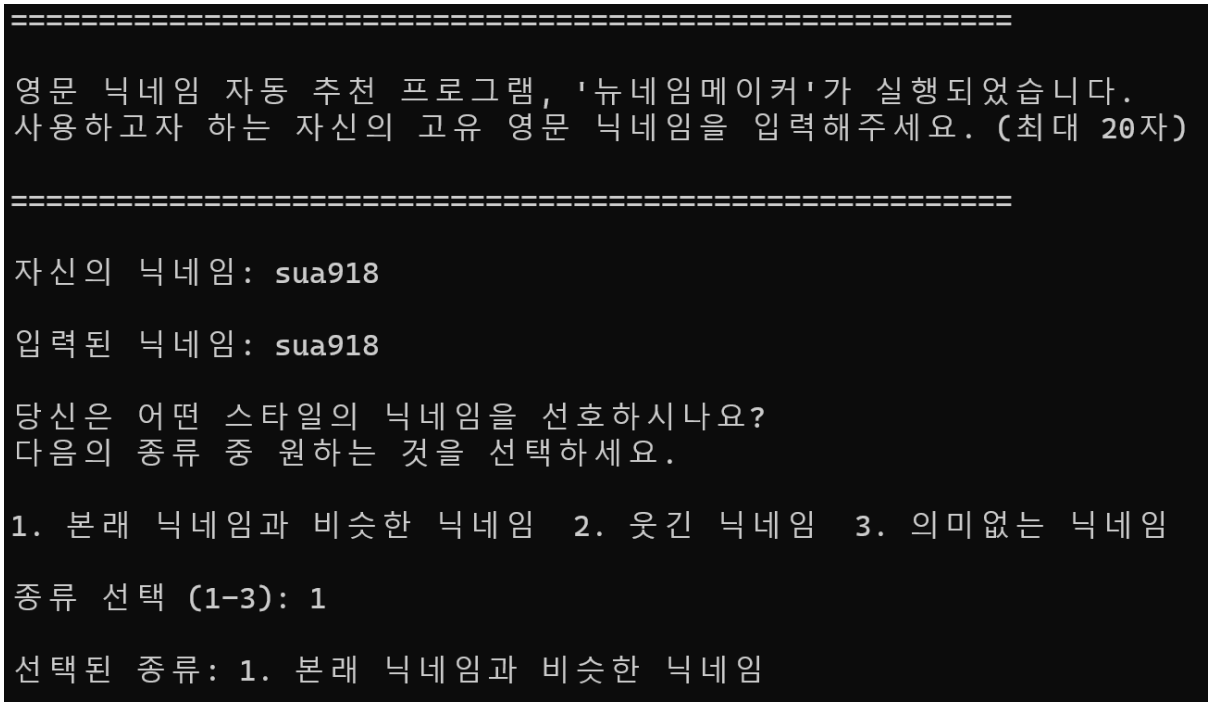
4. 테스트 결과

(1) 사용자의 닉네임과 취향 수집

- 설명

사용자가 입력한 닉네임과 메뉴가 잘 입력되는지 확인하기 위함.

- 테스트 결과 스크린샷



(2) 사용자의 새 닉네임 생성

- 설명

새로운 닉네임이 잘 만들어지는지 확인하기 위함.

- 테스트 결과 스크린샷

```
선택된 종류: 1. 본래 닉네임과 비슷한 닉네임

새 닉네임 1: sua91`8
새 닉네임 2: sua9.18
새 닉네임 3: sua9,18
```

```
선택된 종류: 2. 웃긴 닉네임

새 닉네임 1: Amusing_sua918
새 닉네임 2: Comical_sua918
새 닉네임 3: Pigheaded_sua918
```

```
선택된 종류: 3. 의미없는 닉네임

새 닉네임 1: sua91a8
새 닉네임 2: suua918
새 닉네임 3: sua9H18
```

(3) 사용자의 마음에 들지 않을 경우 닉네임 재생성

- 설명

사용자의 입력에 따라 프로그램 반복과 종료가 잘 수행되는지 확인하기 위함.

- 테스트 결과 스크린샷

```
새 닉네임이 마음에 들지 않는 경우 0을 입력하여 닉네임을 처음부터 다시 만들 수 있습니다.
새 닉네임이 마음에 든다면 1을 입력하여 프로그램을 종료하세요.

0 또는 1을 선택하세요: 0

============================================================

영문 닉네임 자동 추천 프로그램, '뉴네임메이커'가 실행되었습니다.
사용하고자 하는 자신의 고유 영문 닉네임을 입력해주세요. (최대 20자)

============================================================
```

```
새 닉네임이 마음에 들지 않는 경우 0을 입력하여 닉네임을 처음부터 다시 만들 수 있습니다.
새 닉네임이 마음에 든다면 1을 입력하여 프로그램을 종료하세요.

0 또는 1을 선택하세요: 1

C:\Users\sssua\Desktop\대학교\2023_2학기\C프로그래밍및실습\C파일\기말과제_업무자동화\x64\Debug\기말과제.exe(프로세스 6652개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

5. 계획 대비 변경 사항

1) 기능 2와 3의 병합

- 이전

기능 2와 기능 3이 분리되어 있었다.

- 이후

기능 2와 기능 3을 병합하여 기능 2 하나로 만들었다.

- 사유

기존의 기능 2는 단순히 만들어진 닉네임들을 보여주는 것이기 때문에, 굳이 하나의 기능으로 분리할 내용이 아니라고 판단하였다. 또한 기능적으로 기능 2의 순서가 기능 3보다 앞에 있는 것이 맞지 않기도 하여 기능 3과 병합하였다.

2) 기존 기능 3(현재 기능 2)의 세부 기능 병합

- 이전

세부 기능 1과 세부 기능 2가 있었다.

- 이후

세부 기능 1의 내용을 세부 기능 2와 합했다.

- 사유

기존 기능 3의 세부 기능 1은 단순히 닉네임을 꾸밀 수식어가 들어 있는 배열에 불과하기 때문에, 세부 기능으로 분리될 필요가 없다고 생각하였다. 따라서 세부 기능 2의 내용에 추가하여 세부 기능을 한가지로 만들었다.

6. 느낀점

이번 프로젝트를 진행하면서, 많은 어려움이 있었다. 사실 처음에 프로젝트 주제를 정할 때부터 꽤 힘들었는데, 내가 C언어를 잘 모르는 상태에서 막무가내로 주제를 정했다가 구현할 수 없게 되면 어떡할 것인가의 걱정이 있었기 때문이다. 그렇기에 나는 맨 첫 과정이었던 제안서 작성 파트가 가장 골치 아팠던 거 같다. 그리고 프로젝트를 진행하면서는, 원래 한글 닉네임을 목적으로 기획하였는데 영어는 잘 되고 한글은 잘 안되는 경우, 프로그램이 잘 실행되다가 무언가를 추가했을 때 갑자기 문제를 일으키는 경우 등 참 다양했다. 분명 잘 돼야 하는데 예상치 못한 문제가 발생했을 땐 크게 당황했고, 사소한 오류들이 발생했을 땐 디버깅을 하며 풀어나갔던 것 같다. 그리고 프로젝트 과정에서 꽤 좋은 기분을 느꼈던 것은, 잘 구현이 되어 내가 생각한 대로의 결과를 얻어냈을 때였다. 내 생각이 그대로 실현된다는 것은 꽤나 쾌감을 주는 일이었다. 이런 사소한 것들에 힘입어 포기하지 않고 끝까지 완성할 수 있었다고 생각한다. 이렇게 체계적으로 무언가를 만드는 프로젝트는 처음 경험해 보았다. 이번 프로젝트의 경험은 나중에 고학년이 되어서 하게 될 프로젝트들의 양분이 될 수 있을 것이다. 과정은 힘들었지만 배운 점도 많아 즐거운 시간이었다.