**2020년 2학기**

**프로그래밍과 문제해결**

**Assignment #2**

**담당교수: 윤은영**

**학번: 20200854**

**학과: 무은재학부**

**이름: 황찬기**

**POVIS ID: hcg0827**

**명예서약(Honor code)**

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

**Problem : 행맨 게임**

1. 문제 개요

이 프로그램은 행맨 게임을 수행하는 프로그램으로, 게임의 자세한 규칙은 아래와 같다.

* 매 라운드마다 컴퓨터가 단어장(txt 파일)에서 단어 하나를 불러오고 사용자가 이를 맞추는 게임이다.
* 매 시도마다 현재까지 맞춘 문자열(맞춘 문자는 해당 문자로, 맞추지 못한 문자는 ‘\_’로 표시됨), 사용한 문자 목록, 행맨의 목숨 수를 보여주고 문자를 입력받는다.
* 행맨의 목숨은 10개에서 시작하며, 단어(정답)과 입력한 문자를 비교하여 일치하는 것이 없으면 목숨을 하나 잃고, 일치하는 것이 있으면 출력 문자열에서 ‘\_’를 해당 문자로 바꾸어 출력한다.
* 이전에 입력한 문자를 다시 입력한 경우 오류 메시지가 뜨며 다시 입력하게 된다. (목숨 차감 없음)
* 10개의 목숨 이내에 단어를 맞히면 성공 메시지가 출력되며 라운드가 종료되고, 10개의 목숨을 모두 소모하면 실패 메시지가 출력되며 라운드가 종료된다.
* 매 라운드가 끝날 때마다 다음 라운드를 진행할 지 여부를 결정할 수 있다.

이 프로그램의 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.

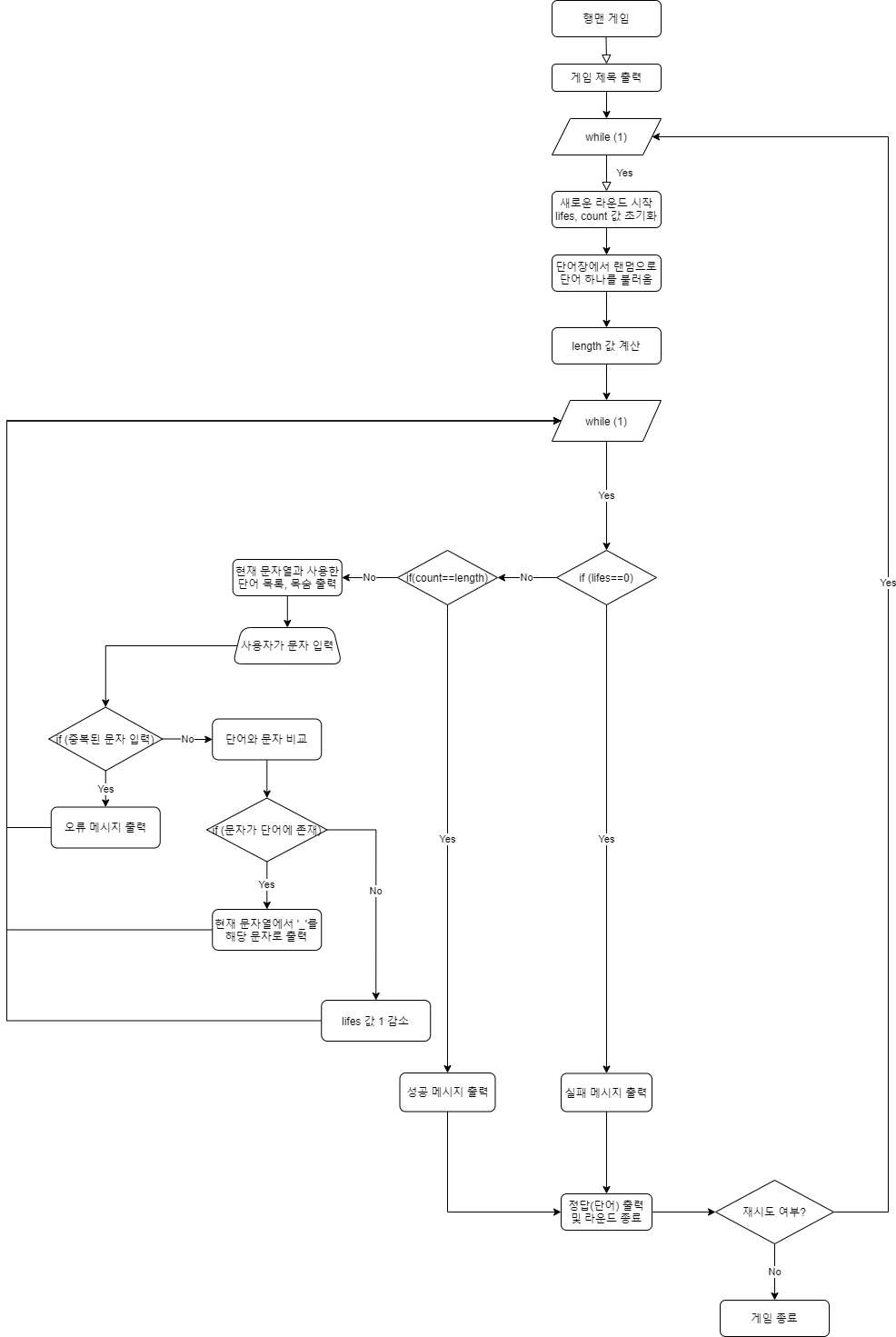
* 입력부 : 단어를 불러오고 사용자가 입력한 문자를 저장한다.
* 처리부 : 단어와 사용자가 입력한 문자를 비교하여 같은 지 다른 지를 표시한다.
* 출력부 : 현재 상황을 표시하는 문자열, 사용한 문자를 표시하는 문자열, 현재 목숨을 출력하고 게임이 끝나면 성공/실패 메시지를 출력한다.

2. 알고리즘

이 프로그램에 대한 알고리즘을 Pseudo Code로 나타내면 아래와 같다.

|  |  |
| --- | --- |
| Pseudo-algorithm for Numerical Baseball | |
|  | *// 프로그램에서 필요한 변수는 미리 선언한 것으로 가정* |
| 1 | Include essential headers <stdio.h>, <stdlib.h>, <time.h> |
| 2 | Define MAX 30 |
| 3 | Start repetitive 1(while) for round |
| 4 | Set lifes, length, count and strings |
| 5 | Get random word from txt. |
| 6 | Start repetitive 2(while) for lifes |
| 7 | Print strings(print\_word, use) and left lifes |
| 8 | Get user number(ans) |
| 9 | Compare the word and ans |
|  | Same : ‘-‘ changes into character(ans) & plus count |
|  | Not same : just put ‘-‘ |
| 10 | If lifes is 0, print fail message and end |
|  | Else if count is same with length(same with answer), print success message and end |
|  | Else, print strings and left lifes and go back to the start of repetitive 2 |
|  | (End of the repetitive 2) |
| 11 | Read character ‘conti’ |
| 12 | If ‘conti’ is ‘y’, start next round |
|  | In else siuation, quit the game |
|  | (End of the repetitive 1) |
| 13 | Print “Good bye.” |

이 Pseudo Code를 Flowchart로 표현하면 다음과 같다.



3. 프로그램 구조 및 설명

a) 게임 제목 출력

- 프로그램을 실행하면 가장 먼저 제목이 출력된다.

b) 새 라운드 시작 및 컴퓨터가 단어를 단어장에서 불러옴

- while문을 이용하여 라운드 반복을 위한 구조를 만들고, 단어장(txt 파일)에서 답이 될 단어 하나를 불러온다.

- 단어장을 불러오는데 실패하는 경우 오류 메시지를 출력하고 프로그램을 종료한다.

- 매 라운드를 시작할 때마다 행맨의 목숨(lifes)와 일치 개수(count) 값을 0으로 초기화하고, 불러온 단어를 이용하여 단어 길이(length)를 구한다.

c) 시도 반복을 위한 구조 생성

- while문을 이용하여 라운드 내에서 시도를 반복하기 위한 구조를 생성한다.

- lifes 값이 0이 될 경우 실패 메시지를 출력하고 라운드를 종료한다.

- count 값과 length 값이 같은 경우(단어와 현재 출력된 문자열이 같은 경우) 성공 메시지를 출력하고 라운드를 종료한다.

d) 입력 값 받기 및 단어와 문자 비교 후 출력

- 위의 경우에 해당하지 않을 경우 단어와 사용자가 입력한 문자를 비교한다. 이 때 입력받은 문자가 단어에 존재하면 출력 문자열에서 해당 문자를 출력하고, 이 외에 해당되지 않는 문자는 ‘\_’로 표시되어 출력된다.

- 입력받은 문자가 단어에 아예 존재하지 않는 경우 lifes 값을 1 감소시키고 돌아간다.

- 사용자가 입력한 문자는 또 다른 문자열에 저장되어 출력되고, 이전에 입력했던 문자를 다시입력할 경우 오류 메시지를 출력하고 돌아가서 다시 입력을 받는다.

e) 재시도 및 게임 종료 설정

- 매 라운드가 끝날 때마다 재시도 여부를 사용자의 문자를 받아 결정한다.

- y가 입력되는 경우 새로운 라운드를 시작하고, n이 입력되는 경우 게임을 종료한다.

4. 프로그램 실행 예제

|  |  |
| --- | --- |
|  |  |
| 답을 맞춘 경우 | 답을 틀린 경우 |
|  |  |
| 다음 라운드로 넘어가는 경우 | 중복된 단어를 입력한 경우 |

**Problem : 삼목 게임**

1. 문제 개요

이 프로그램은 삼목 게임(TIC-TAC-TOE Game)을 수행하는 프로그램으로, 게임의 자세한 규칙은 아래와 같다.

* 매 라운드마다 3x3의 판에서 먼저 한 줄(가로, 세로, 대각선)을 차지하면 승리하는 게임이다.
* 매 시도마다 컴퓨터가 놓은 수의 위치 출력/사용자가 놓을 수를 입력 받아 출력하고, 현재까지의 상황을 표시한 삼목 판을 출력한다.
* 컴퓨터와 사용자의 시작 순서는 랜덤으로 정해지며, 먼저 시작하는 쪽이 O, 나중에 놓는 쪽이 X로 표시된다.
* 이전에 입력한 위치를 다시 입력한 경우 오류 메시지가 뜨며 다시 입력하게 된다.
* 컴퓨터가 먼저 한 줄을 차지하면 패배, 사용자가 먼저 한 줄을 차지하면 승리한다. 9칸이 모두 채워졌지만 승패가 결정되지 않는 경우 무승부이다.
* 매 라운드가 끝날 때마다 다음 라운드를 진행할 지 여부를 결정할 수 있다.

이 프로그램의 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.

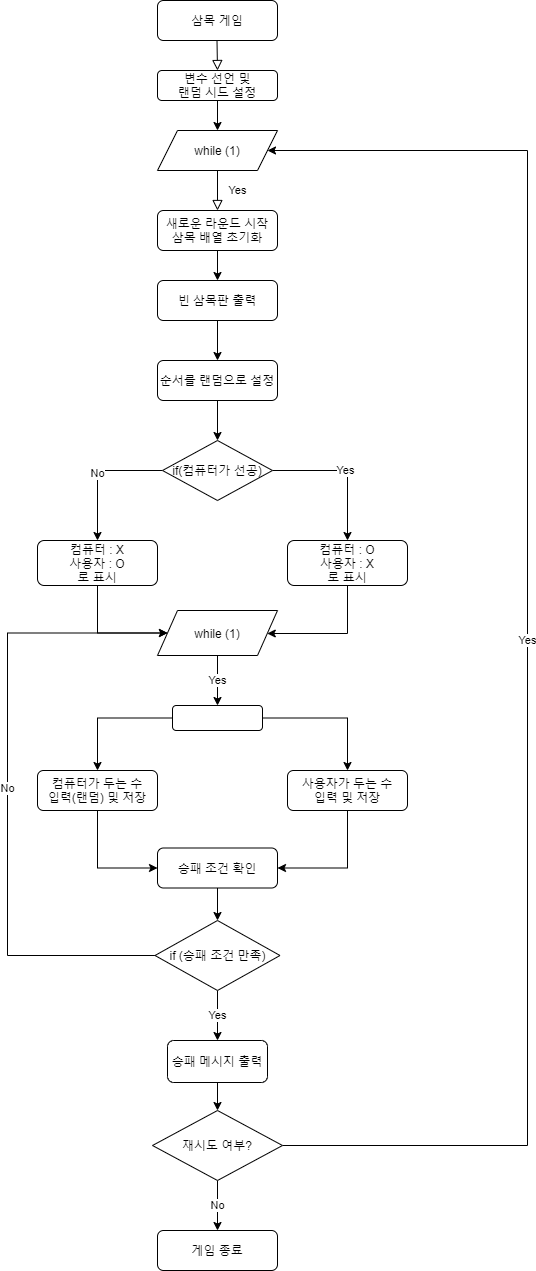
* 입력부 : 컴퓨터/사용자가 입력한 위치에 수를 놓고 배열을 저장한다.
* 처리부 : 승리/패배/무승부 조건에 일치하는 지 확인한다.
* 출력부 : 컴퓨터/사용자가 놓은 수의 위치와 현재 상황을 표시하는 삼목 판을 출력하고, 게임이 끝나면 승리/패배/무승부 메시지를 출력한다.

2. 알고리즘

이 프로그램에 대한 알고리즘을 Pseudo Code로 나타내면 아래와 같다.

|  |  |
| --- | --- |
| Pseudo-algorithm for Numerical Baseball | |
|  | *// 프로그램에서 필요한 변수는 미리 선언한 것으로 가정* |
| 1 | Include essential headers <stdio.h>, <stdlib.h>, <time.h> |
| 2 | Define MAX 3 |
| 3 | Start repetitive 1(while) for round |
| 4 | Initialize the array |
| 5 | Print empty board |
| 6 | Decide the order |
| 6 | Start repetitive 2(while) for lifes |
| 7 | Save and print computer/user input |
| 8 | Check the condition of result |
| 9 | Win : print message of win and exit round |
|  | Lose : print message of lose and exit round |
|  | Draw : print message of draw and exit round |
| 10 | Else, continue. |
|  | (End of the repetitive 2) |
| 11 | Input character ‘conti’ |
| 12 | If ‘conti’ is ‘y’, start next round |
|  | In else siuation, quit the game |
|  | (End of the repetitive 1) |
| 13 | Print “Good bye!” |

이 Pseudo Code를 Flowchart로 표현하면 다음과 같다.



3. 프로그램 구조 및 설명

a) 새 라운드 시작 및 빈 삼목 판 출력

- while문을 이용하여 라운드 반복을 위한 구조를 만들고, 초기화 된 빈 삼목 판을 출력한다.

- 컴퓨터/사용자의 순서를 정하고 이에 따라 O/X 문자를 배정한다.

b) 시도 반복을 위한 구조 생성

- while문을 이용하여 라운드 내에서 시도를 반복하기 위한 구조를 생성한다.

d) 입력 값 받기 및 현재 상황이 반영된 삼목 판 출력

- 컴퓨터/사용자가 두는 수를 입력 받아 표시한다.

- 승패 조건을 체크하여 해당되면 승리/패배/무승부 메시지를 출력하고 라운드를 종료한다.

- 가로, 세로, 대각선으로 한 줄을 먼저 완성하는 쪽이 승리하며, 9칸이 모두 채워져도 승패가 결정되지 않으면 무승부로 처리한다.

- 해당 사항이 없다면 다시 되돌아 가 또 다른 수를 받아 표시한다.

e) 재시도 및 게임 종료 설정

- 매 라운드가 끝날 때마다 재시도 여부를 사용자의 문자를 받아 결정한다.

- y가 입력되는 경우 새로운 라운드를 시작하고, n이 입력되는 경우 게임을 종료한다.

4. 프로그램 실행 예제

a) 초기 출력

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 사용자가 선공 | 컴퓨터가 선공 | 재시도 X | 재시도 O |

b) 승리한 경우

|  |  |
| --- | --- |
|  |  |
| 대각선 승리 | 가로 승리 |
|  |  |
| 세로 승리 | (중복 입력/잘못 입력한 경우) |

c) 패배/무승부인 경우

|  |  |
| --- | --- |
|  |  |
| 패배한 경우 | 무승부인 경우 |

5. 토론

* 행맨 게임에서 문자열을 비교하여 최종 결과를 출력할 때, 배열 전체를 비교하려고 하니 항상 다르다는 결과가 출력되어 각 문자열을 출력해보니 한 문자열에 쓰레기 값이 들어 있는 것을 확인할 수 있었다. 코드를 확인하니 그 이유는 포인터 변수를 잘못 사용하여서 이상한 값이 전달된 것이었다.
* 코드를 완성하고 직접 실행했을 때, 분명히 승리 조건이나 패배 조건을 만족하는 경우임에도 불구하고 계속 진행되다 무승부가 출력되는 오류가 발생했다. 이는 코드에서 if문, else if문을 사용하면서 우선순위를 잘못 설정하여 발생한 상황이었다.

6. 결론

- 이번 과제를 통해 1, 2차원 배열을 활용하여 여러 가지 게임을 만들어 볼 수 있었다. 또한 코드를 직접 짜고 실행시켜보며 헷갈렸던 포인터 변수, 역 참조 연산자의 적절한 사용법을 확인할 수 있었다. 또한 프로그램 실행 과정에서 발생하는 여러가지 오류를 코드를 하나하나 살펴보며 해결하며 코드를 조금 더 깔끔하게 작성해야 한다는 생각이 들었다.

7. 개선사항

- 이번 과제에서도 추가 기능을 도입하려 했으나 기존 동작을 코드로 짜는 과정에서도 오류가많이 발생했고, 이를 해결하며 새로운 기능을 도입하기 어려웠다. 나중에 이를 개인적으로 추가하고 변형하여 심심할 때 즐길 수 있을 정도로 만들어 보고 싶다.