

C++

행맨 게임

진척 보고서 #1

제출일자: 11월 17일

제출자명: 김강문

제출자학번: 233958

1. 프로젝트 목표

1) 배경 및 필요성

C++ 수업시간에 배운 내용을 바탕으로 게임을 구현하며 조건문, 반복문, 함수화 등 프로그래밍의 이해와 수준을 발전시키기 위함. 사람들이 많이 알고 하기 쉬운 게임을 구현해 혼자 혹은 여러 명에서 즐길 수 있는 게임인 행맨 게임을 구현하려 한다.

2) 프로젝트 목표

혼자 혹은 여러 명에서 행맨 게임을 플레이 할 수 있는 프로그램을 만드는 것

3) 차별점

기존의 행맨 게임과는 다르게 혼자서도 플레이 할 수 있도록 난이도 설정을 넣어 미리 적어놓은 단어 혹은 상대방이 입력한 단어를 맞추도록 한다.

2. 기능 계획

1) 기능 1 (난이도 선택)

- 난이도 선택에 따른 단어를 출력한다.

(1) 세부 기능 1 (상대방이 입력한 단어 맞추기)

- 난이도 설정을 통해 상대방이 입력한 단어를 맞출 수 있도록 한다.

(1) 세부 기능 2 (미리 입력한 단어 맞추기)

- 난이도 설정을 통해 미리 입력해 놓은 단어를 맞출 수 있도록 한다.

2) 기능 2 (남은 기회와 단어 맞추기)

- 남은 기회와 맞춘 문자를 보여준다. 남은 기회가 0이되면 실패 메시지 출력과 함께 프로그램이 종료되고, 단어를 다 맞추면 성공 메시지 출력과 함께 프로그램이 종료된다.

3. 진척사항

1) 기능 구현

(1) 게임 방법 설정 및 난이도 설정

- 입출력

int setting	//설정을 위한 변수
int difficulty	//난이도 선택을 위한 변수
int maxAttempts	//최대 횟수 저장을 위한 변수
string word	//단어 저장을 위한 변수
string getRandomWord()	//임의의 단어를 반환하는 함수
vector<string> words	//단어를 저장해 놓은 벡터

- 설명

요약: 설정 선택에 따른 동작과 안내 메시지 출력, 난이도 선택에 따른 동작과 안내 메시지 출력.

프로그램 시작 시 설정 안내가 출력되고 setting변수에 숫자를 입력 받는다. 설정에 1을 입력하면 getRandomWord()함수를 통해 word에 랜덤으로 단어를 저장하고 그에 따른 안내문이 출력과 반복문을 탈출한다. 2를 입력하면 안내문이 출력되고 단어를 입력할

설정을 끝낸 후 난이도 설정 안내가 출력되고 difficulty변수에 숫자를 입력 받는다. 1, 2, 3 입력 시 각각 쉬움, 보통, 어려움의 난이도가 선택되며 그에 따른 도전 횟수가 정해지고 안내문이 출력과함께 반복문을 탈출한다. 그 외 입력 시 안내문과 함께 다시 입력 받는다.

: 반복문을 이용해 무한루프를 만들어 일정 조건을 해당할 때만 탈출할 수 있도록 구성했다. 조건문을 통해 조건에 따라 설정과 난이도를 선택하도록 만들었다. 함수를 사용해 메인 함수를 줄이고 미리 저장해 놓은 단어를 랜덤으로 반환하도록 했다. 벡터에 맞출 단어들을 저장해 두었다.

[illegible]

```

54 // 난이도 설정
55 while (true) {
56     // 난이도 안내
57     cout << "\n난이도 선택 : 1.쉬움(기회: 8) Easy 2.보통(기회: 7) 3.어려움(기회: 6) : ";
58     cin >> difficulty;
59
60     if (difficulty == 1) {
61         maxAttempts = 8; // 쉬움
62         cout << "쉬움을 선택했습니다. 기회는 8번 입니다." << endl;
63         break;
64     } else if (difficulty == 2) {
65         maxAttempts = 7; // 보통
66         cout << "보통을 선택했습니다. 기회는 7번 입니다." << endl;
67         break;
68     } else if (difficulty == 3) {
69         maxAttempts = 6; // 어려움
70         cout << "어려움을 선택했습니다. 기회는 6번 입니다." << endl;
71         break;
72     } else {
73         cout << "1, 2, 3 중 하나를 입력해주세요.\n" << endl;
74     }
75 }

```

(2) 글자 맞추기

- 입출력

int wrongGuesses	//틀린 횟수를 위한 변수
vector<bool> guessed(word.length(), false);	//맞춘 상태 저장하기위한 벡터
vector<char> wrongLetters	//틀린 글자를 저장하기 위한 벡터
string word	//단어 저장을 위한 변수
int maxAttempts	//최대 횟수 저장을 위한 변수
char c	//틀린 글자 출력을 위한 변수
char guess	//글자 확인을 위한 변수
bool correctGuess	//글자를 맞췄는지 확인하기 위한 변수
bool allGuessed	//단어를 맞췄는지 확인하기 위한 변수

- 설명

기회가 남아있을 동안 글자를 맞출 수 있다. 매 시도마다 현재 상태(맞춘 글자와 못 맞

춘 글자), 남은 기회, 틀린 글자를 출력하고 입력 안내 후 글자를 입력 받는다. 입력 받은 글자가 입력 받았었던 글자인지 확인 후 맞으면 다시 입력 받고 아니면 글자가 단어에 들어가는지 확인한다. 틀렸을 경우 틀린 횟수를 증가시킨다. 모든 글자가 다 맞았을 경우 승리 메시지를 출력한다. 만약 기회를 모두 사용했다면 실패 메시지를 출력한다.

- 적용된 배운 내용 (반복문, 조건문, 함수, 포인터 등)

: 반복문을 이용해 틀린 횟수가 최대 횟수보다 작으면 단어를 맞출 수 있도록 했다. 조건문을 통해 이미 입력한 글자인지 확인, 입력한 글자가 단어에 있는지 확인하고 단어를 모두 맞췄을 경우와 모든 기회를 사용했을 경우에 따른 메시지를 출력하도록 구성했다. 함수와 포인터를 이용해 메인 함수를 줄이고 맞춘 글자와 못 맞춘 글자를 출력하도록 만들었다. 벡터를 사용해 글자의 맞춤 여부를 저장해 놓았다.

- 코드 스크린샷

```
15 // 현재 상태를 출력하는 함수
16 void display(const string& word, const vector<bool>& guessed) {
17     for (size_t i = 0; i < word.length(); ++i) {
18         if (guessed[i]) {
19             cout << word[i]; // 맞춘 글자 출력
20         } else {
21             cout << "_"; // 못 맞춘 글자는 밑줄 출력
22         }
23         cout << " ";
24     }
25     cout << endl;
26 }
```

```
77 int wrongGuesses = 0; // 틀린 횟수를 위한 변수
78 vector<bool> guessed(word.length(), false); // 맞춘 상태 저장하기 위한 벡터
79 vector<char> wrongLetters; // 틀린 글자를 저장하기 위한 벡터
80
81 cout << "\n행맨 게임을 시작합니다!" << endl;
```

```

83 // 맞추는 루프
84 while (wrongGuesses < maxAttempts) {
85     display(word, guessed); // 현재 상태 출력
86
87     // 남은 기회와 틀린 글자 출력
88     cout << "남은 기회 : " << maxAttempts - wrongGuesses << endl;
89     cout << "틀린 글자 : ";
90     for (char c : wrongLetters) {
91         cout << c << " ";
92     }
93     cout << endl;
94
95     // 입력 안내 및 입력 받기
96     char guess;
97     cout << "글자를 입력하세요: ";
98     cin >> guess;
99
100    // 이미 입력한 글자인지 확인
101    bool alreadyGuessed = false;
102    for (size_t i = 0; i < word.length(); ++i) {
103        if (word[i] == guess && guessed[i]) {
104            alreadyGuessed = true;
105            break;
106        }
107    }
108    for (char c : wrongLetters) {
109        if (c == guess) {
110            alreadyGuessed = true;
111            break;
112        }
113    }
114    if (alreadyGuessed) {
115        cout << "이미 입력한 글자입니다. 다시 입력하세요.\n" << endl;
116        continue;
117    }

```

```

119         bool correctGuess = false; //맞췄는지 확인하기 위한 변수
120
121         // 입력한 글자가 단어에 있는지 확인
122         for (size_t i = 0; i < word.length(); ++i) {
123             if (word[i] == guess && !guessed[i]) {
124                 guessed[i] = true;
125                 correctGuess = true;
126             }
127         }
128         if (!correctGuess) { // 틀렸을 경우
129             wrongGuesses++;
130             wrongLetters.push_back(guess);
131         }
132
133         // 모든 글자를 맞췄는지 확인
134         bool allGuessed = true;
135         for (bool letterGuessed : guessed) {
136             if (!letterGuessed) {
137                 allGuessed = false;
138                 break;
139             }
140         }
141
142         // 승리 메시지 출력
143         if (allGuessed) {
144             cout << "축하합니다. 단어 (" << word << ")를 맞췄습니다." << endl;
145             break;
146         }
147
148         cout << "\n";
149     }
150
151     // 패배 메시지 출력
152     if (wrongGuesses >= maxAttempts) {
153         cout << "기회를 모두 사용했습니다. 맞출 단어는 " << word << " 이었습니다." << endl;
154     }
155
156     return 0;
157 }

```

2) 테스트 결과

(1) 게임 방법 설정 및 난이도 설정

- 설명

1,2,3 사진은 설정 안내와 그에 따른 출력, 4,5,6,7 사진은 난이도 안내와 그에 따른 출력

- 테스트 결과 스크린샷

설정 : 1. 혼자(저장되어있는 단어가 랜덤으로 선택됩니다.) 2. 같이(상대방이 맞출 단어를 직접 입력합니다.) : 1 저장된 단어 중 하나를 랜덤으로 선택했습니다.	
설정 : 1. 혼자(저장되어있는 단어가 랜덤으로 선택됩니다.) 2. 같이(상대방이 맞출 단어를 직접 입력합니다.) : 2 상대방이 맞출 단어를 입력해주세요:program	
설정 : 1. 혼자(저장되어있는 단어가 랜덤으로 선택됩니다.) 2. 같이(상대방이 맞출 단어를 직접 입력합니다.) : 3 1 또는 2를 입력해주세요.	
난이도 선택 : 1.쉬움(기회: 8) Easy 2.보통(기회: 7) 3.어려움(기회: 6) : 1 쉬움을 선택했습니다. 기회는 8번 입니다.	
난이도 선택 : 1.쉬움(기회: 8) Easy 2.보통(기회: 7) 3.어려움(기회: 6) : 2 보통을 선택했습니다. 기회는 7번 입니다.	
난이도 선택 : 1.쉬움(기회: 8) Easy 2.보통(기회: 7) 3.어려움(기회: 6) : 3 어려움을 선택했습니다. 기회는 6번 입니다.	
난이도 선택 : 1.쉬움(기회: 8) Easy 2.보통(기회: 7) 3.어려움(기회: 6) : 4 1, 2, 3 중 하나를 입력해주세요.	

(2) 글자 맞추기

- 설명

1,2,3 사진은 단어 맞추는 루프의 출력, 4,5 사진은 각각 성공과 실패에 따른 출력, 6번은 이미 입력한 글자를 입력했을 경우의 출력이다.

- 테스트 결과 스크린샷

행맨 게임을 시작합니다! 남은 기회 : 6 틀린 글자 : 글자를 입력하세요: p		행맨 게임 시작안내와 현재 상태와 남은 기회, 틀린 글자 출력 후 입력 받기
p 남은 기회 : 6 틀린 글자 : 글자를 입력하세요: o		전에 입력이 맞췄을 경우 기회는 줄어들지 않고 맞춘 글자 출력과 못 맞춘 글자 등 출력, 입력 받기

<p>p o 남은 기회 : 5 틀린 글자 : i 글자를 입력하세요: u</p>	<p>전에 입력이 틀렸을 경우 기회가 줄어들거 맞춘 글자 출력과 못 맞춘 글자, 틀린 글 자 등 출력, 입력 받기</p>
축하합니다. 단어 (orange)를 맞췄습니다.	
기회를 모두 사용했습니다. 맞출 단어는 program 이었습니다.	
<p>a 남은 기회 : 6 틀린 글자 : 글자를 입력하세요: a 이미 입력한 글자입니다. 다시 입력하세요.</p>	

4. 계획 대비 변경 사항

1) 난이도 선택

- 이전: 난이도 선택에 따라 맞출 단어를 다르게 설정
- 이후: 설정과 난이도 선택에 따라 게임 방법과 기회 설정
- 사유: 알고 있는 영어 단어가 많지 않았고, 게임 방법과 난이도를 나누어 설정하고 싶었다. 하지만 게임 방법을 먼저 설정하면 난이도에 따라 단어를 저장시키기가 애매했다. 그래서 기회를 다르게하기로 결정했다.

5. 프로젝트 일정

업무		11/3	11/10	11/17	12/1
제안서 작성		완료			
기능1	세부기능1		완료		

	세부기능2		완료		
기능2				진행중	
보완할 기능					-----

업무		12/15	12/22
제안서 작성			
기능21	세부기능1		
	세부기능2		
기능2			
보완할 기능		----->	