

C++프로그래밍및실습

단어스크램블게임

진척 보고서 #1

제출일자:2024.11.17

제출자명:윤동해

제출자학번:234129

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

단어 스크램블 게임은 단어의 철자를 무작위로 섞어 제시해 플레이어가 원래 단어를 맞추는 퍼즐 게임이다. 수업 시간에 학습한 내용과 그간 구현해보았던 '틱택토', '야구 게임', '머드 게임'을 참고해 간단한 단어 스크램블 게임을 구현할 수 있을 것 같아 주제로 선정하게 되었다. 이 게임은 언어 능력과 논리적 사고력을 향상시킬 수 있는 교육적 목적을 가진 게임으로, 이를 통해 영어 단어 어휘력 향상을 기대할 수 있다는 점에서 필요성을 가진다.

2) 프로젝트 목표

간단한 단어의 순서를 랜덤으로 제시해 플레이어의 언어 능력과 논리적 사고를 향상시키는 게임 프로그램을 만드는 것을 목표로 한다.

3) 차별점

기존 프로그램은 단어를 직접 입력 받아 프로그램이 뒤섞는데 이는 본인이 작성한 단어만 알면 쉽게 맞출 수 있기 때문에 랜덤하게 게임에서 다양한 단어 리스트를 통해 문제를 제시하며, 난이도 조절을 통해 기본 난이도 외에도, 쉬운 단어에서 어려운 단어로 점진적으로 상승하는 난이도 조절 기능을 제공하도록 한다. 점수 시스템을 추가해 직관적으로 점수를 확인할 수 있게 하며, 여유가 된다면 제한 시간 기능을 넣어 게임의 긴장감을 추가할 수 있도록 한다.

2. 기능 계획

1) 기능 1 기본 게임 플레이 (기본적인 게임 플레이에 필요한 기능들)

- 무작위 단어를 선택하고 이를 섞어 플레이어에게 제시한다.

(1) 세부기능 1 다양한 단어 테마 선택

- 플레이어가 게임 시작 시 원하는 주제를 선택하여 해당 주제의 단어들로만 게임을 진행할 수 있게 한다.

- 플레이어가 단어를 맞추면 점수를 획득하며, 틀리면 정답을 알려준다.

(2) 세부 기능 2 점수 시스템

- 맞출 때마다 점수를 획득하고, 틀릴 경우 감점하여 최종 점수를 누적한다.

2) 기능 2 단어 난이도 조절

- 짧은 단어에서 긴 단어로 점차 난이도가 상승하도록 설정한다.

3) 기능 3 제한 시간 추가

- 타이머를 사용하여 단어를 맞출 수 있는 제한 시간을 설정하고, 시간이 초과되면 점수가 차감되거나 게임이 종료된다.

4) 기능 4 최고 기록 저장

- 최고 점수 기록을 파일에 저장하여, 다음 게임 시작 시 플레이어에게 최고 점수를 보여준다.

3. 진척사항

1) 기본 게임 플레이 (기본적인 게임 플레이에 필요한 기능들)

- 무작위 단어를 선택하고 이를 섞어 플레이어에게 제시한다.

(1) 세부기능 1 다양한 단어 테마 선택

- 플레이어가 게임 시작 시 원하는 주제를 선택하여 해당 주제의 단어들로만 게임을 진행할 수 있게 한다.

- 입출력

- 입력: 주제를 선택할 때 1, 2, 3번을 입력 받는다.

섞인 단어가 출력되었을 때 정답을 입력 받는다.

- 출력: 주제를 선택하세요: 1. 과일 2. 동물 3. 직업 선택:

잘못된 입력입니다. 기본 주제(과일)로 진행합니다.

순서가 섞인 단어가 출력된다.

정답일 경우에 "정답입니다!"가 출력된다.

오답일 경우에 "틀렸습니다. 정답은 '<원래 단어>' 입니다."가 출력된다.

- 설명

- 단어를 섞는 함수 shuffleWord

- random_device rd: 난수 생성을 위한 장치를 초기화한다.
 - mt19937 generator(rd()): Mersenne Twister 난수 생성기를 초기화한다.
 - shuffle(word.begin(), word.end(), generator): 주어진 단어의 문자들을 무작위로 섞는다.

- 주제를 선택하고 단어를 반환하는 함수 chooseTheme

- 사용자가 주제를 선택하고 이에 맞는 단어 목록을 반환한다.
 - 잘못된 입력이 들어오면 기본 주제인 "과일" 목록을 반환한다.

- main 함수

- chooseTheme 함수를 호출하여 사용자가 선택한 주제의 단어 목록을 불러온다.
- 난수 생성기를 사용하여 단어 목록 중 하나를 무작위로 선택한다.
- 선택된 단어를 shuffleWord 함수를 통해 섞는다.
- 섞인 단어를 출력하고, 사용자가 원래 단어를 맞추도록 한다.
- 사용자의 입력과 원래 단어를 비교하여 정답 여부를 출력한다..

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

- 벡터, 랜덤 난수, switch, string, cout, 조건문 if, void

- 새롭게 공부한 내용

- #include <algorithm>, shuffleWord 함수는 따로 검색하여 알아보았다.

```
void shuffleWord(string &word) {
    random_device rd;           // 난수 생성기
    mt19937 generator(rd());    // Mersenne Twister 엔진
    shuffle(word.begin(), word.end(), generator); // std::shuffle 사용
}
```

random_device rd는 난수 생성기를 초기화한다.

mt19937 generator(rd())는 Mersenne Twister 알고리즘을 사용하는 난수 생성기다.

shuffle(word.begin(), word.end(), generator)은 shuffle 함수를 사용하여 word 문자열의 문자를 무작위로 섞는다. shuffle(시작, 끝, 난수 생성 엔진)

```
// 랜덤 단어 선택
random_device rd;
mt19937 generator(rd());
uniform_int_distribution<int> distribution(0, words.size() - 1);
string originalWord = words[distribution(generator)];
```

uniform_int_distribution 객체를 생성한다. 이 객체는 0부터 words.size() - 1 (단어 목록의 마지막 인덱스) 사이의 정수 중에서 균일하게 난수를 생성한다.(모든 단어 동일 확률)

distribution(generator)을 호출하면 generator를 사용해 지정된 범위 내에서 난수를 생성한다. 이 난수는 words 벡터의 인덱스로 사용되고, 그 인덱스에 있는 단어가 originalWord에 저장된다.

- 코드 스크린샷

```
c > interim_reports1.cpp > chooseTheme()
1  #include <iostream>
2  #include <vector>
3  #include <algorithm> // shuffle
4  #include <random>    // random_device, mt19937
5
6  using namespace std;
7
8  void shuffleWord(string &word) {
9      random_device rd;          // 난수 생성기
10     mt19937 generator(rd());    // Mersenne Twister 엔진
11     shuffle(word.begin(), word.end(), generator); // std::shuffle 사용
12 }
13
14 vector<string> chooseTheme() {
15     int choice;
16     cout << "주제를 선택하세요:\n1. 과일\n2. 동물\n3. 직업\n선택: ";
17     cin >> choice;
18
19     switch (choice) {
20         case 1: return {"apple", "banana", "mango", "grape", "melon", "orange", "pear", "apple"};
21         case 2: return {"lion", "tiger", "elephant", "monkey", "rabbit", "cat", "dog", "lion"};
22         case 3: return {"doctor", "engineer", "teacher", "nurse", "lawyer", "artist", "writer", "doctor"};
23         default:
24             cout << "잘못된 입력입니다. 기본 주제(과일)로 진행합니다.\n";
25             return {"apple", "banana", "cherry", "grape", "melon", "orange", "pear", "apple"};
26     }
27 }
28
29 int main() {
30     // 주제 선택 및 단어 로드
31     vector<string> words = chooseTheme();
32
33     // 랜덤 단어 선택
34     random_device rd;
35     mt19937 generator(rd());
36     uniform_int_distribution<int> distribution(0, words.size() - 1);
37     string originalWord = words[distribution(generator)];
38
39     // 단어 섞기
40     string scrambledWord = originalWord;
41     shuffleWord(scrambledWord);
42
43     cout << "섞인 단어: " << scrambledWord << "\n";
44     cout << "원래 단어를 맞춰보세요: ";
45
46     string playerGuess;
47     cin >> playerGuess;
48
49     if (playerGuess == originalWord) {
50         cout << "정답입니다!\n";
51     } else {
52         cout << "틀렸습니다. 정답은 '" << originalWord << "' 입니다.\n";
53     }
54
55     return 0;
56 }
```

2) 테스트 결과

(1) 테스트한 기능 이름

- 설명

세 가지 테마를 사용자가 선택하도록 해 선택할 시 리스트에 있던 단어의 철자가 랜덤으로 출력하게 하고, 잘못 선택할 시 1번 주제로 게임을 플레이하도록 한다. 사용자가 정답을 입력하면 정답인지 오답인지 출력한다. 다음 스크린샷은 모든 경우의 수를 캡처한 것이다.

- 테스트 결과 스크린샷

```
주제를 선택하세요:  
1. 과일  
2. 동물  
3. 직업  
선택: 1  
섞인 단어: enmlo  
원래 단어를 맞춰보세요: melon  
정답입니다!
```

```
주제를 선택하세요:  
1. 과일  
2. 동물  
3. 직업  
선택: 2  
섞인 단어: nmyoke  
원래 단어를 맞춰보세요: monkey  
정답입니다!
```

```
주제를 선택하세요:  
1. 과일  
2. 동물  
3. 직업  
선택: 3  
섞인 단어: gneneire  
원래 단어를 맞춰보세요: e  
틀렸습니다. 정답은 'engineer' 입니다.
```

