

자바 프로그래밍 미니 프로젝트

타자게임

2024.08.23

정지용, 원진호

소개할 알고리즘

01 ——— 전체적인 코드 구조

02 ——— 게임 문자 중복없이 랜덤으로 섞기

03 ——— 입력값 일치여부 확인

04 ——— 결과값 산출

05 ——— 랭킹 지정

03 전체적인 게임 흐름

```
플레이할 인원수 설정(반복문을 실행할 횟수)
게임 난이도 선택 (조건문 사용)
(난이도 : 기본) {
    (사람 수 만큼 반복){
        사용자 id 입력
        (문제 수 만큼 반복){
            타자 게임 플레이
        }
    }
}
(난이도 : 심화) {
    (사람 수 만큼 반복){
        사용자 id 입력
        (문제 수 만큼 반복){
            타자 게임 플레이
        }
    }
}
점수, 타자속도, 정확도 계산
점수, 타자속도, 정확도를 환산해서 총점 계산
랭킹을 매겨서 총점이 높은 순서대로 통계 출력
```

04 기능소개 -게임 문자를 중복없이 섞기

```
String korWord = ""; // korArr에서 무작위로 받을 값
String signWord = ""; // signArr에서 무작위로 받을 값
String gameTxt = ""; // korWord + signWord
String randomGameTxt = "";
```

```
korWord = korArr[r.nextInt(korArr.length)];
signWord = signArr[r.nextInt(signArr.length)];
gameTxt = korWord + signWord;
// 게임 문자 랜덤으로 조합
```

korWword => 아지랑이
signWord => @~\$

```
for(int k = 0; k < gameTxt.length(); k++) {
```

아지랑이 + @~\$

```
char r_txt = gameTxt.charAt(r.nextInt(gameTxt.length())); // 랜덤으로 글자 가져오기
```

아, 지, 랑, 이, @, ~, \$

```
if(randomGameTxt.length() == 0) {
```

```
randomGameTxt += r_txt;
```

게임 문자열의 길이가 0이므로 반복문 실행 X
=> 임의로 첫 글자를 저장해 반복문을 실행

```
}
for(int l = 0; l < randomGameTxt.length(); l++) {
```

```
if(randomGameTxt.charAt(l) != r_txt) {
```

```
}else {
```

```
if(k != 0) {
```

```
k--;
```

```
}
```

```
break;
```

```
}
```

```
if(l == randomGameTxt.length() - 1) {
```

```
randomGameTxt += r_txt;
```

```
}
```

```
if(randomGameTxt.length() == gameTxt.length()) {
```

```
break;
```

```
}
```

```
}
```

```
}
```

기존에 합치기만 했던 문자열과
섞어서 나온 문자열의 길이가 같으면
종료

게임 문자열의 길이 만큼 반복문을 실행
랜덤으로 뽑아온 글자가 저장된 문자열 안에 있는지
한글자씩 비교해서 검사
=> 중복되지 않으면 게임 문자열 안에 추가
=> 중복되면 반복문을 빠져나와 랜덤 글자를 새로 가져와 다시 진행

05 기능소개 -입력값의 일치여부 확인

```
korWord = korArr[r.nextInt(korArr.length)]; // 랜덤으로 korArr에서 한글  
gameTxt = korWord; // 랜덤으로 받은 한글을 gameTxt에 저장한다.  
System.out.println(gameTxt); // 연습할 문제를 콘솔에 출력.
```

→ 랜덤으로 korArr배열에서 문자열 받아오기

```
String userTxt = in.nextLine(); // 값을 입력받을 변수
```

→ 플레이어로 부터 문자열을 입력 받기.

```
// 20240822_원진호_점수 계산하는 기능  
if(userTxt.equals(gameTxt)) { // 사용자가 입력한 문자랑 gameTxt 가 같다면  
    score += 20; // 정답시 현재 사용자가 이 문제에서 획득한 점수는 10점 입니다.  
}else {  
    missWord = ""; // 오타난 문자를 받기위해 저장공간을 비워준다.  
    score += 0; // 오답시 현재 사용자가 이 문제에서 획득한 점수는 10점 입니다.  
}
```

```
String missWord = "오타없음";
```

```
int missCnt = 0;
```

→ .equals()메소드를 사용해서 입력값의 일치여부 판단.

```
String[] korArr = {  
    "연못", "강물", "해변", "고요", "바위", "은하수", "산길", "낙엽", "그림자", "사랑",  
    "파랑", "노래", "봄바람", "눈꽃", "별빛", "바람길", "햇빛", "하얀", "푸른", "가슴",  
    "꿈", "비밀", "가르침", "기억", "연기", "파도", "은빛", "물결", "바다", "나룻방",  
    "꽃", "구름", "길", "여름", "초록", "흰", "노을", "나비", "산책", "아침",  
    "땀", "별", "하늘", "가을", "겨울", "물", "여름", "밤하늘", "꿈", "기다림",  
    "추억", "소리", "시간", "달", "하루", "웃음", "들", "새", "작은", "낮잠",  
    "바람", "눈", "희망", "봄꽃", "고양이", "나무", "햇살", "미소", "초원", "바다새",  
    "은하", "무지개", "고요함", "등산", "별님", "달님", "바람개비", "푸르름", "햇빛", "구름사이",  
    "여름", "파도소리", "바다빛", "산등성", "고갯길", "아지랑이", "낙조", "별뿔별", "해돋이", "산새",  
    "들인", "꽃향기", "술속", "들담", "해오라기", "비바람", "가르수", "석양", "고향", "산들바람",  
    "햇살밭", "꽃길", "고드름", "이슬", "바람꽃", "산길", "은빛", "모래사장", "물방울", "새소리",  
    "강가", "별무리", "구름다리", "달빛사이", "은하수길", "고요한밤", "하얀구름", "푸른하늘", "햇살길", "아침햇살",  
    "이른아침", "구름산", "바람소리", "달그림자", "물안개", "술속길", "바람길", "고요한산", "별나라", "강둑",  
    "작은섬", "바람산", "고요한술", "햇살속", "별빛길", "나무술", "바다길", "푸른산", "파란하늘", "구름바람",  
    "고요한강", "별빛바람", "산속길", "달빛산", "바다속", "은빛파도", "구름사이길", "별빛속", "바람길", "산속물결",  
    "고요한바다", "햇빛술", "푸른초원", "바람술", "구름술", "달빛길", "바람길사이", "햇살속길", "산속고요", "강물소리",  
    "푸른바람", "고요한바람", "별빛사이길", "구름사이바람", "바다속길", "햇빛속바람", "고요한길", "은빛길", "푸른술길", "별빛고요",  
    "달빛고요", "산속길사이", "구름속바람", "햇살고요", "별빛속길", "달빛속길", "고요한은빛길", "바람속고요", "산속바람길", "푸른바람길"  
};
```

06 기능소개 -결과값 산출하기

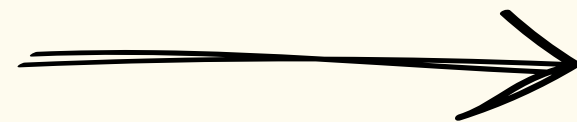
```
long beforeTime = System.currentTimeMillis();
String userTxt = in.nextLine(); // 값을 입력받을 변수
long afterTime = System.currentTimeMillis();
long secDiffTime = (afterTime - beforeTime)/1000;
questionCnt++;
accSpeed += secDiffTime;
accTxt += userTxt;
if(userTxt.equals(gameTxt)) {
    score += 20;
}else {
    missWord = "";
    score += 0;
}
for(int k = 0; k < gameTxt.length(); k++) {
    if(userTxt.charAt(k) != gameTxt.charAt(k)) {
        missCnt++;
        missWord += userTxt.charAt(k) + " ";
    }
}
```



입력값을 받기전과 후로 시간값을 받아서
타자속도를 구하기 위해
두 값의 차이를 계속 누적



100점 만점 기준으로 20점 5문제 제출
글자가 모두 맞으면 20점 부여
글자가 하나라도 틀리면 0점 부여



글자를 각각 비교해서 일치하지 않으면
오타를 누적하고 오타 글자를 출력

```
userSpeed[i] = Math.round((accSpeed / questionCnt) * 10) / 10.0; // 소수점 1자리까지
userCorrect[i] = Math.round(((1 - (double) missCnt / accTxt.length()) * 100) * 10) / 10.0; // 소수점 1자리까지
userPoint[i] = score; // 점수
for(int i = 0; i < userCnt; i++) {
    userTotalScore[i] = Math.round(((userPoint[i] * 0.5) + userSpeedScore[i] + userCorrectScore[i]) * 100) / 100.0;
}
```



타자 속도 => (누적 시간 값 / 문제 수)
정확도 => (오타 개수 / 입력한 글자 수)
점수 => 맞은 문제 수 만큼 20점 누적
총점 =>
점수 50%, 타자속도 20%, 정확도 30%
로 환산해서 100점 만점 기준으로 계산

07 기능소개 -랭킹 정하기

```
// 20240822_원진호_순위를 매기기 위한 반복문
int [] ranking = new int[userCnt];
double max_num = 0;
int max_index = 0;
int x = 0;
for(int i = 0; i < userCnt; i++) {
    if(userTotalScore[i] > max_num) {
        max_num = userTotalScore[i];
        max_index = i;
    }
    if(i == userCnt - 1) {
        ranking[x] = max_index;
        userTotalScore[max_index] = 0;
        max_num = 0;
        max_index = 0;
        x++;
        i = -1;
    }
    if(x == userCnt) {
        break;
    }
}
// 20240822_원진호_총점이 0이 되버려서 다시 원상복구 작업
for(int i = 0; i < userCnt; i++) {
    userTotalScore[i] = Math.round(((userPoint[i] * 0.5) + userSpeedScore[i] + userCorrectScore[i]) * 100) / 100.0;
}
// 결과
System.out.println("총점은 (점수 + 속도 + 정확도)의 환산값 입니다.");
System.out.println("이름   총점   점수   속도   정확도");
System.out.println("-----");
for(int i = 0; i < userCnt; i++) {
    System.out.println(userId[ranking[i]] + "   " + userTotalScore[ranking[i]] +
        "   " + userPoint[ranking[i]] + "   " + userSpeed[ranking[i]] +
        "   " + userCorrect[ranking[i]]);
}
```

반복문을 사용해서 첫번째 최댓값, 두번째 최댓값... 사용자 인원수 만큼의 최대값을 저장한다.

x는 랭킹 배열의 인덱스값이다.

이후 총점을 0에서 원상복구 하는 작업이 있어야 랭킹을 출력할때 총점값을 사용할 수 있게된다.

08 실행영상

난이도: 기본

3
난이도 선택:
1번 기본 / 2번 심화
1
1번째 ID를 입력해주세요
진호01
은하수길
은하수길
타이핑 시간 : 4초 / 오타 : 없음 / 현재 점수 : 20

음방송
음방

난이도: 심화

환영합니다. www.BANDICAM.COM
게임을 플레이 할 인원수를 설정해주세요. (숫자 입력)
3
난이도 선택:
1번 기본 / 2번 심화
2
1번째 ID를 입력해주세요

09 후기



문제 해결

랭킹순위를 정할때 , 인원수만큼의 최대값을 구하는 상황이 있었습니다. 저는 이문제를 해결하기 위해 팀원과 상의중 기존 최대값을 0으로 대체하고 무한루프를 사용해서 비교하는 방법을 알아내었습니다.



문제 해결

또한 제가 문자열 배열에서 인덱스 번호로 랜던 숫자를 사용 시 중복될 수 있다는 상황을 생각하지 못했는데, 이 또한 팀원과 상의 후 일게 됨.



많은 에러

기능을 추가하거나 수정하다보니 정말 많은 에러들을 마주했습니다. 덕분에 이제는 코드를 작성하다가 에러가 생겨도 머리가 하얘지거나 긴장하지 않고 무슨 에러가 일어났는지 파악 할 수 있습니다.



재밌는 고통

첫 미니프로젝트를 진행하면서 분업을 하는 것이 쉽지 않았고 코드가 원활하게 동작하지 않아서 답답하고 어지러웠습니다. 그치만 제작하는 동안에 시나리오 및 코드를 작성하는 것을 즐기면서 진행했다.

정원 조

감사합니다.
