KEY PROJECTS

MIDDLE NAME

점심시간 팀이 개발한 단어 맞추기 게임입니다. 기존과 달리 가운데 글자를 추론하여 맞히는 것이 목표입니다.

팀원
김시온
최진석
이명균

MEETING

어떠한 게임을 만들것인가

점심메뉴

가운데 글자가 없는 점심메뉴를 출력
어떤 글자인지 예상해서 답을 맞히는 게임

어떠한 기능을 넣을 것인가

기능

난이도 선택 기능
점수 표현 기능
life 기능
combo 기능

파트 분배는 어떻게 할것인가

파트 분배

난이도 선택기능과
점수 기능
life 기능
점수에 따른 위치 이동 기능을 담당 하기로 함

INTRODUCTION

단어 제시

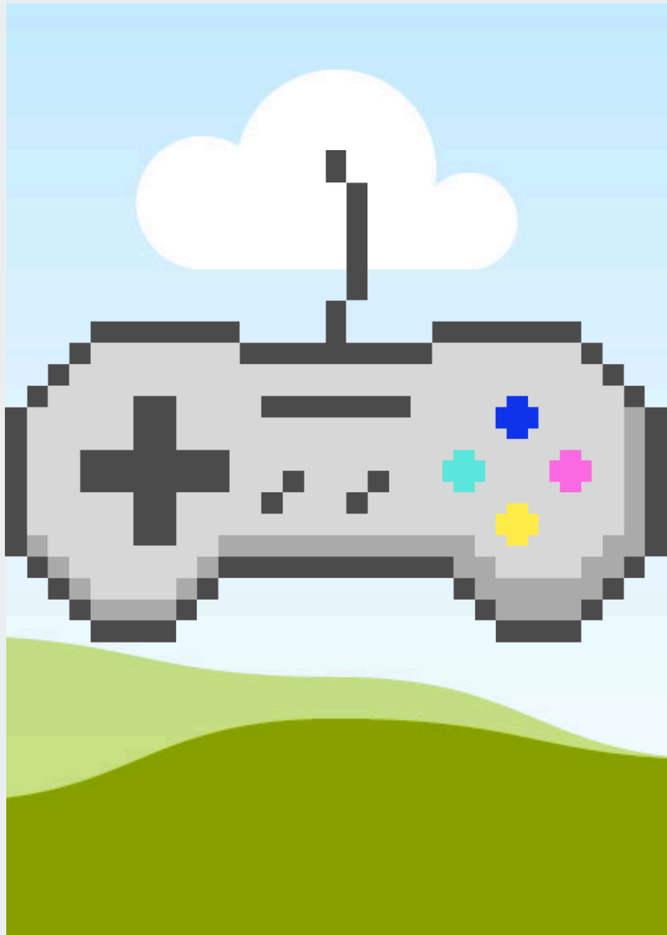
단어의 첫 글자와 마지막 글자가 주어집니다

가운데 글자 맞히기

플레이어는 가운데 글자를 추론해야 합니다

점수 획득

정답을 맞히면 10점을 획득합니다



FUNCTION

김시온

난이도 선택기능 구현

```
// do while문을 이용한 난이도 조절 반복문
do {
    System.out.println(yellow + "난이도" + exit + "를 선택하세요.");
    System.out.println("--------");
    System.out.println("|상| |하|");
    System.out.println("--------");
    c = in.nextLine();
} while (!c.equals("상") && !c.equals("하"));

//c 가 "상"이면 a[i]에 aA[i] 어레이의 값을 저장함. else 즉 "하"라면 a[i]에 tempa[i]를 저장함.
if (c.equals("상")) {
    for (int i = 0; i < a.length; i++) {
        a[i] = aA[i];
    }
} else {
    for (int i = 0; i < a.length; i++) {
        a[i] = tempa[i];
    }
}
```

- **do while 문**

do while 문을 사용해 일단 난이도 선택 화면을 한 번 실행시킨 후, 조건이 true가 아니면 do문을 반복해서 실행시키게 하였습니다.

원래는 for문으로 처리하려고 하였는데 do while문은 일단은 먼저 한번 실행시키고 나서 while 조건식에 조건을 설정 하는거라 훨신 더 직관적이고 깔끔하게 해결할수 있어서 좋았습니다.

- **if문**

사용자가 “상”을 입력하면 a[i] 배열에 난이도 상의 단어들이 있는 aA[i] 배열의 단어들이 저장되게 하였습니다.

else 즉 “하“일 경우에 a[i]에 tempa[i] 배열의 값이 저장되게 하였습니다.

FUNCTION

김시온

점수, 콤보 기능 및 life 기능

```
//정답시 효과, 오답시 효과등을 구현하기 위한 if문
else if (a[num].charAt(1) == b.charAt(0)) {
    System.out.println("");
    System.out.println(blue + "정답" + exit);
    System.out.println("");
    point += 10;
    cnt++;
} else {
    System.out.println("");
    System.out.println(red + "오답" + exit);
    System.out.println("");
    cnt = 0;
    life--;
    if (life <= 0) {
        System.out.println("[" + "Game Over" + "]");
        break;
    }

  }

if (cnt >= 3) {
    point += 5;
    System.out.println(cnt + "연속으로 정답을 맞혔습니다. (+5 점)");
}
```

if문 활용 정답시

입력한값 0번째 즉, 첫글자의 값과 a[num]charAt[1]의 두번째 글자가 같다면 정답이 출력 되게 하였고 정답시 point 에 10을 더하고 콤보 기능을 사용하기 위해서 cnt 에 +1이 되게 설정하였습니다.

if문 활용 오답시

false라면 오답이 출력되게 하였고 콤보 기능인 cnt는 초기화 하였습니다. 또한 점수는 추가 되지 않게 아무조건도 설정하지 않았습니다.
life는 기본 2로 설정 하였고 오답시 -1이 되게 하였습니다. 그리고 0이하가 되면 game over 창을 출력한 뒤 break로 for반복문을 나가게 하였습니다.
오답까지 처리 한뒤
cnt가 3이상이면 combo점수 5point가 계속 추가되게 하였고 안내문이 출력되게 하였습니다.

FUNCTION

김시온

life 및 특정조건시 특정위치 이동

```
/// 목숨을 스트링 ♥로 바꿔서 출력하기 위한 if문
String lifeS = "";

if (life == 2) {
    lifeS = "♥♥";

}else {
    lifeS = "♥♡";
}
```

if문 활용 num을 string으로 표현

life을 숫자로 설정을 해 놓았는데 그러면 게임같은 느낌이 없었기 때문에 ♥로 표현을 하게 설정 하였습니다.

```
//목숨이 0보다 작을경우 게임을끝내고 다시 난이도 선택 하는 창으로 돌아가기.
if(life <= 0) {
    System.out.println("");
    continue;
}
```

while문에서 continue활용

life가 0이면 메인for문에서 break로 반복문을 나오게 하였는데 그 다음에 게임을 끝내는게 아닌 다시 도전할수 있는 기회를 주고자 하였습니다. 그래서 처음 난이도 선택하는 do while문 위에 while문을 하나 만들었고 for문에서 나온 상태에서 continue로 다시 난이도 선택창으로 돌아가게 하였습니다.

REFACTORING

```
// do while문을 이용한 난이도 조절 반복문
do {
    System.out.println(yellow + "난이도" + exit + "를 선택하세요.");
    System.out.println("--------");
    System.out.println("|상| |하|");
    System.out.println("--------");
    c = in.nextLine();
} while (!c.equals("상") && !c.equals("하"));
```

원래는 for문으로 처리를 하였고 난이도 선택시 스트링으로 ‘상’ or ‘하’를 입력하는것이 아니라 숫자 ‘1’ or ‘2’를 입력하여 처리하였습니다.

if문으로 하였을때 좀더 지저분해 보였고 직관적이지 않은 것 같아 do while문으로 더 직관적이게 바꾸었습니다. nextint등을 사용하면 버퍼가 남아 있어 또 처리를 해야되기 때문에 nextline으로 한번에 깔끔하게 처리하고자 nextLine, 상, 하를 입력하게 하여 난이도를 선택하게 하였습니다.

REFACTORING

```
String b = in.nextLine();
```

```
가운데 글자를 맞혀보세요.

똠[ ]꿍

Exception in thread "main" java.lang.StringIndexOutO
```

```
if (b.equals("")) {

    System.out.println("");
    System.out.println(red + "오답" + exit);
    System.out.println("");
    cnt = 0;
    life--;
    if (life <= 0) {
        System.out.println("[" + "Game Over" + "]");
        break;
    }
}
```

문자가 출력된후

정답을 입력해야되는 상황에서 Enter키를 입력하면 오류가 나왔습니다.
왜 에러가 나는지 열심히 찾아봤는데 정답 입력시 charAt(); 으로 조건식을 설정 하였는데 입력한 값과 같으면 정답 처리를 하고 false이면 오답처리를 하게 하였습니다. 거기서 enter ""입력시의 조건이 처리가 안되어서 에러가 났다는 판단이 들었습니다.
그리하여 ""시 오답으로 처리하여 에러를 수정하였습니다.

REVIEW

프로젝트 후기
처음에는 머릿속에서 쉽게 구현될 것 같았지만, 막상 프로젝트를 시작하고 코드를 짜려니 예상보다 어려웠습니다. 예전에 배웠던 내용인데도 기억이 나지 않아서 많은 시간을 소비하며 고생했습니다.
특히, 간단한 미니 프로젝트였음에도 불구하고 발생한 여러 오류들로 당황하기도 했습니다.
처음으로 참여한 그룹 프로젝트였기 때문에, 파트 배분이나 서로 다른 의견을 조율하는 과정에서도 어려움이 있었습니다.
하지만 팀원들과 함께 "무엇을 만들까?" "어떤 기능을 추가하면 재미있을까?"를 고민하며 회의하는 과정은 매우 즐거웠습니다.
서로 모르는 부분을 도와주고, 부족한 점을 채워가며 프로젝트를 진행하는 동안 팀워크의 중요성을 깨달았고, 혼자 하는 것보다 훨씬 재미있게 임할 수 있었습니다.

예전에 배웠던 내용을 잘 기억하지 못해 고생했지만, 이번 프로젝트를 통해 복습할 수 있어 만족스러웠습니다.
오류로 인해 짜증이 나기도 했지만, 차분히 문제를 해결할 때 느꼈던 성취감은 매우 컸습니다.
마지막으로, 내가 생각만 하던 것을 직접 구현할 수 있어서 정말 기분이 좋았습니다.
물론, 더 많은 기능을 넣고 싶었지만, 시간과 실력의 한계로 인해 아쉬움이 남았습니다.
앞으로 더 실력을 키워서 다음 프로젝트에서는 더 잘해내고, 더 재미있는 프로젝트를 많이 해보고 싶습니다.

김시온

THANK YOU