



-320개다.

영문이 몇개고 ?

전과자 목록 : 김성민 , 김가율 , 김성도
조 이름 : 전과자

짧은글 연습

긴글 연습

낙서장

타자 검정

타자 검정 결과

1 개발 환경

2 개발 일정

3 요구 사항

4 구성도

5 구현

6 미제 사건

7 수사 후기

LIST

| 전과자들의 개발 목표



**신속한
개발**

**팀원과의
의사소통**

**유지보수
유연함**

1

개발 환경



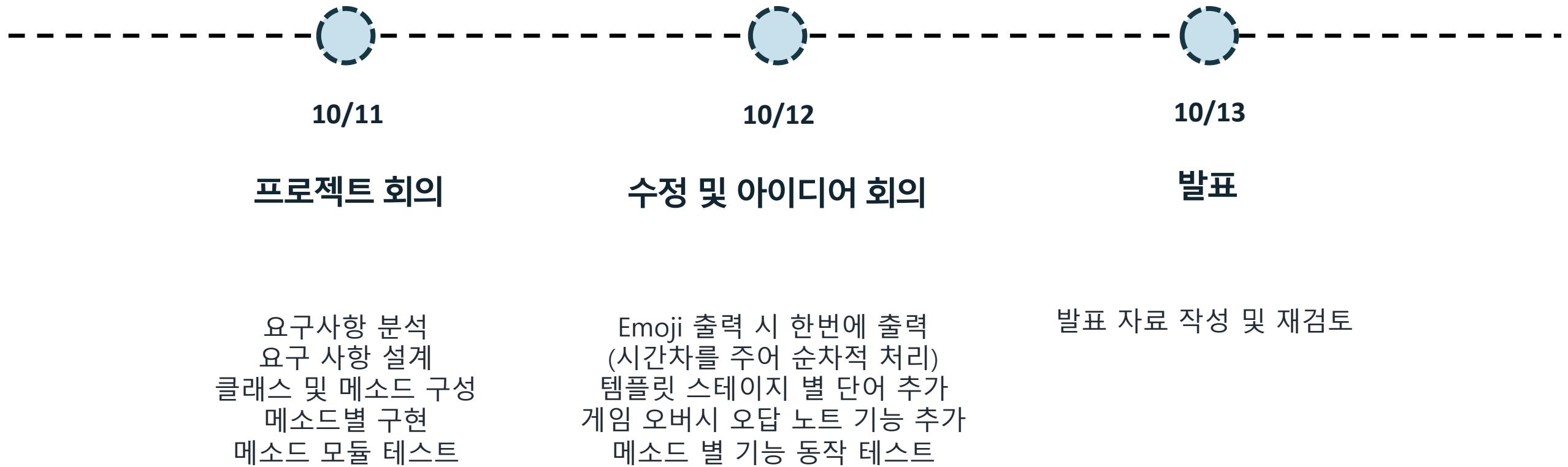
TOOL



2

개발 일정

Part 2 개발 일정



3

요구 사항

Part 3 요구 사항

List	내용	담당자
1	프로그램 시작 시 게임 규칙이 먼저 출력된다	김가율
2	Enter를 입력 받으면 게임이 시작 된다	김가율
3	처음 시작 할 때 갖고 있는 하트 갯수는 총 3개 이다.	김성민
4	Stage1 부터 시작하고 Stage는 총 3개로 구성된다	김성민
5	Stage별 문제가 랜덤으로 출력 된다	김성민

Part 3 요구 사항

List	내용	담당자
1	사용자에게 랜덤 문자열 출력을 보여준다	김성민
2	사용자는 출력된 단어에 따라 키보드로 입력한다	김성민
3	사용자가 입력한 문자는 대소문자를 구분 하지 않고 비교한다	김가율
4	정답일 경우 총 정답 개수와 연속 정답 갯수가 증가한다.	김성민
5	3번 연속 정답일 경우 다음 스테이지로 이동되며 이모지와 잔여 하트를 출력한다	김가율

Part 3 요구 사항

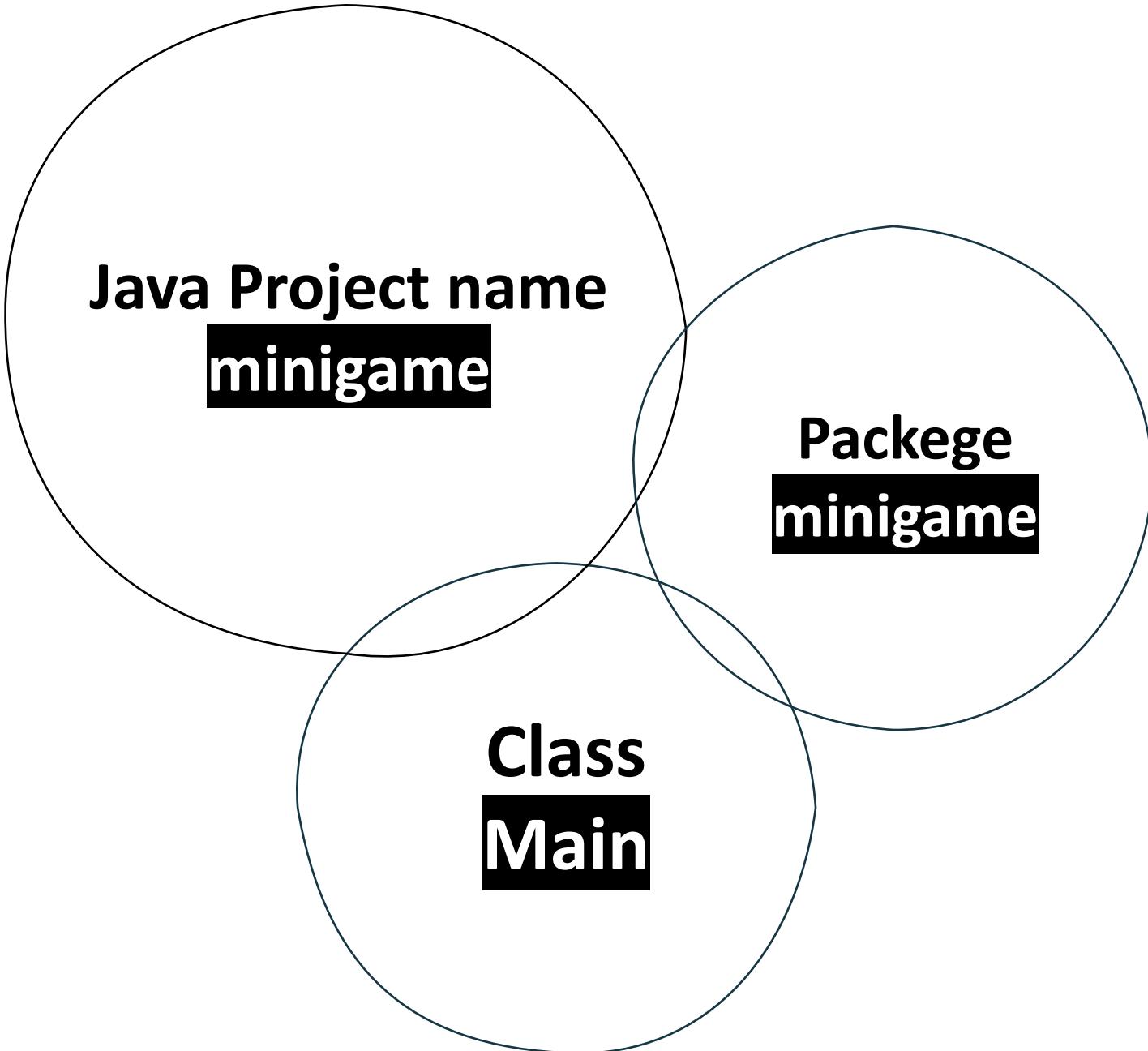
List	내용	담당자
11	Stage 이동 시 이동된 Stage에 맞는 문제를 랜덤으로 출력한다	김가율
12	오답인 경우 하트 3개중 1개가 차감되고 남은 하트 수가 표현된다	김성민
13	오답인 경우 연속 정답 카운트는 0 으로 초기화 되고 Stage는 1로 돌아간다	김성도
14	중간에 게임 종료를 희망 할 경우 exit를 입력 했을 때 즉시 종료된다	김성도
15	하트가 0 개 이하가 되었을때 게임이 종료된다	김성도

Part 3 요구 사항

List	내용	담당자
16	잔여 하트 소멸 시 이대로 종료할 것인지 오답 노트 확인 후 재도전 할건지 선택한다	김가율
17	재 도전 시 틀렸던 오답을 출력해주고 재 시작 한다	김성민
18	잔여 하트 소멸과 중간 탈출 시에도 총 정답 개수와 종료 시점의 Stage를 알려준다	김성민
19	이모지 출력 시 랜덤으로 색상 표현을 한다	김가율
20	이모지 출력 시 순차적으로 출력한다	김성도

4

구성도



Part 4 구성도

```
src
└ minigame
    choice
    └ Emoji.java
    └ Emoji
    └ StrPlot
    └ miniKeyGame
    └ Stage
    └ TrueAndFalse
    └ Main
```

Main & Class
클래스 구성

```
minigame.*;
java.util.Scanner;

static minigame.Emoji.RollBook;

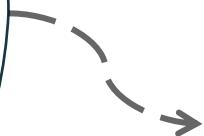
class Main {
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    RollBook();
    new miniKeyGame(in.nextLine());
```

Main 클래스에서는
RollBook (게임 규칙)을 먼저
호출하여 출력한 뒤
게임시작하는 생성자 생성

```
age new *
ic class miniKeyGame {
// 호출 생성자
1 usage new *
public miniKeyGame(String enter)
    miniKeyGame();
}
1 usage new *
public void miniKeyGame() {
    Scanner in = new Scanner(Sys
    Emoji Emo = new Emoji();
```

생성자 호출시 실행되는
miniKeyGame 메소드

Package **minigames**
Class
miniKeyGame



Class Choice

Method : endChoice
Method : isReGame

Class Stage

Method : whatStageNum
Method : whatStage

Class TrueAndFalse

Method : outWrongNote
Method : InPutWrong
Method : truelsFalse

Class Strplot

Method : Strplot

Class Emoji

Method : RollBook

Part 4 구성도

```
// 초기 생명 개수 선언  
int heart = 3;  
  
// 연속 정답 확인을 위한 변수  
int trueCnt = 0;  
  
// 오답노트에 입력될 배열 처리 cnt  
int falseCnt = 0;  
  
// 총 정답 개수  
int totCnt = 0;  
  
// 스테이지 처리를 위한 변수  
int stageNum = 1;
```

miniKeyGame에서
필요한 전역 변수

```
// 게임 시작을 알리는 하트이모지 출력  
StrPlot.Strplot(heartArr[heart]);  
  
while (flag) {  
    String outPuts = Stage.whatStage(stageNum);  
    System.out.println(" 문제 [ 대/소문자 관계없음 ] : " + outPuts);  
    System.out.print("User 입력 >> ");  
    String tmp = in.nextLine().toLowerCase();  
    // 강제종료 희망하는지 User 입력값 확인 : '참'일 경우 종료 문구 출력 및 종료  
    if (!choice.endChoice(tmp, totCnt, stageNum)) break;
```

게임 시작시 지급되는 하트3개 이모지 출력
while문을 사용한 반복문 시작
Stage별 랜덤 문제를 받기위해 Stage Class를 참조하여
whatStage 메소드를 사용합니다, 리턴되는 값을 문제로 사용합니다



miniKeyGame Class

```
// 강제종료 희망하는지 User 입력값 확인 : '참'일 경우 종료 문구 출력 및 종료  
if (!choice.endChoice(tmp, totCnt, stageNum)) break;
```



```
public static Boolean endChoice(String s, int totCnt, int stageNum){  
    Emoji Emo = new Emoji();  
    boolean answer = true;  
    if (s.equals(EXIT)) {  
        StrPlot.Strplot(Emo.gmExit);  
        System.out.println("\t\t\t\"총 정답 개수 :" + totCnt +  
                           "\n\t\t\t\"최종 스테이지 :" + Stage.whatStageNum(stageNum));  
        answer = false;  
    }  
    return answer;  
}
```

choice Class .
endChoice Method
Boolean type을 반환



miniKeyGame Class

```
if (trueIsFalse(outPuts, tmp)) {  
    trueCnt++;  
    totCnt++;  
    if (trueCnt == 3) {  
        StrPlot.Strplot(Emo.stUP);  
        StrPlot.Strplot(heartArr[heart]);  
        trueCnt = 0;  
        stageNum++;
```



```
public static boolean trueIsFalse(String str1, String str2) {  
    //str1 : 문제 , str2 : User 입력문자  
    Boolean result = str1.toLowerCase().equals(str2);  
    if (result == true) {  
        System.out.println("\t\t\t정답입니다.");  
    } else if (result == false) {  
        System.out.println("\t\t\t오답 입니다");  
    }  
    return result;  
}
```

TrueAndFalse Class .
trueIsFalse Method
Boolean type을 반환

choice . isReGame Method

```

public static Boolean isReGame(String str, int totCnt, in
    boolean answer = true;
    Emoji Emo = new Emoji();
    String y = "Y";
    String n = "N";

    if (str.equals(y)) {
        System.out.println("◀");
        System.out.println("\t\t\t◀ 총 정답 개수 : " + totCn
        System.out.println("◀");
        StrPlot.Strplot(Emo.bye);
        answer = false;
    } else if (str.equals(n)) {
        System.out.println("◀");
        System.out.println("\t\t\t◀ 총 정답 개수 : " + totCn
        System.out.println("◀");
        outWrongNote();
        StrPlot.Strplot(Emo.gmStart);
    }
    return answer;
}

```

miniKeyGame Class

```

if (trueIsFalse(outPuts, tmp)) {
    trueCnt++;
    totCnt++;
    if (trueCnt == 3) {
        StrPlot.Strplot(Emo.stUP);
        StrPlot.Strplot(heartArr[heart]);
        trueCnt = 0;
        stageNum++;
    }
} // 오답일 경우
else { trueCnt = 0;
    stageNum = 1;
    falseCnt++;
    tnf.inPutWrong(outPuts, falseCnt);
    if (heart > 1) {
        heart--;
        StrPlot.Strplot(heartArr[heart]);
    } else if (heart <= 1) {
        StrPlot.Strplot(Emo.gmOVER);
        System.out.print(" 게임 종료 Y / 오답 노트 확인 후 재시작 N : ");
        flag = choice.isReGame(in.next().toUpperCase(), totCnt, stageNum);
        in.nextLine();
        stageNum = 1;
        totCnt = 0;
        trueCnt = 0;
        falseCnt = 0;
        heart = 3;
    }
}

```

StrPlot Class . Strplot Method

```

9 usages
public static void Strplot(String emj) {
    String[] color = {red, green, yellow, blue, purple, cyan, white};
    Random r = new Random();
    String colors = color[r.nextInt( bound: 5)];
    int timer = 0; // 다음 스트링줄이 나올때까지의 대기시간
    int l = 64; // 스트링 한줄의 길이
    int flag = 0; // stage
    String tmpS = "";
    for (int i = 1; i <= 8; i++) {
        while (flag == 0) {
            for (int j = l * (i - 1); j >= l * (i - 1) && j < l * i; j++) {
                tmpS += emj.charAt(j); // tmp에 i번째 문자 저장
            }
            System.out.println(colors + tmpS + exit); // out
            tmpS = ""; // Reset
            flag = 1; // loop end
            timer = 6 * 16777216; // next loop까지 대기시간
        }
        while (flag == 1) {
            if (timer == 0) {
                flag = 0; // loop end
            }
            timer--;
        }
    }
}

```

Emoji Class . RollBook Method

```

2 usages
public static void RollBook() {
    Emoji Emo = new Emoji();
    StrPlot.Strplot(Emo.gmStart);
    String[] opStrArr = "-----"
        , "[-----]"
        , " GAME ROLL"
        , "[-----]"
        , " 亂 亂\|t||"
        , "( -·\·t||"
        , "/ つΦ"
        , " 1. 게임 시작 시 Enter를 눌러주세요"
        , " 2. 정답은 대/소 문자를 가리지 않습니다."
        , " 3. 처음 시작 시 하트3개 지급 , 오답 제출 시 하트 1개씩 소멸됩니다."
        , " 4. 하트3개 모두 소멸 시 게임 종료 / 재 시작시 처음부터 재시작 합니다"
        , " 5. 3번 연속 정답 시 다음 스테이지로 넘어갑니다 ( Stage 1 ~ 3 ) "
        , " 6. 스테이지 3번 부터는 띠어쓰기 & 공백 까지 같아야 정답입니다."
        , " 7. 오답 발생 시 스테이지는 처음부터 다시 시작합니다."
        , " 8. 즉시 종료 방법 : exit 입력시 프로그램은 즉시 종료됩니다."
        , "-----";
    // 다음 스트링줄이 나올때까지의 대기시간
    int timer = 0;
    // stage
    int flag = 0;

    for (int i = 0; i < opStrArr.length; i++) {
        while (flag == 0) {
            System.out.println(opStrArr[i]); // out
            flag = 1; // loop end
            timer = 6 * 16777216; // next loop까지 대기시간
        }
        while (flag == 1) {
            if (timer == 0) {
                flag = 0; // loop end
            }
            timer--;
        }
    }
}

```



5

구현

Part 5 구현

```
'#####:'##:: #: '# #####:'#####:'#####:'#####:'###:  
##....:: ##:: #::... #::..:: ##....:: #... #:::: ##:  
##::::: ##:: ##:: ##:: ##:: ##:: ##:: ##:: ##:  
#####:'## ## ##:: ##:: ##:: ##:: ##:: ##:: ##:  
##...:: ##. ##:: ##:: ##:: ##:: ##. ##:: ##:: ##:  
##::::: ##.. ##:: ##:: ##:: ##:: ##.. ##:: ##:: ##:  
#####:'##.. ##:: ##:: ##:: ##:: ##:: ##.. ##:: ##:  
.....::.....::.....::.....::.....::.....::.....:  
  
[GAME ROLL]  
  
ㅂ ㅂ ||  
( .ω. ||  
/ つΦ
```

1. 게임 시작 시 Enter를 눌러주세요
2. 정답은 대/소 문자를 가리지 않습니다
3. 처음 시작 시 하트3개 지금 , 오답 제출 시 하트 1개씩 소멸됩니다.
4. 하트3개 모두 소멸 시 게임 종료 / 재 시작시 처음부터 재시작 합니다
5. 3번 연속 정답 시 다음 스테이지로 넘어갑니다 (Stage 1 ~ 3)
6. 스테이지 3번 부터는 띄어쓰기 & 공백 까지 같아야 정답입니다.
7. 오답 발생 시 스테이지는 처음부터 다시 시작합니다.
8. 즉시 종료 방법 : exit 입력시 프로그램은 즉시 종료됩니다.

```
-----  
[GAME ROLL]  
  
ㅂ ㅂ ||  
( .ω. ||  
/ つΦ  
  
1. 게임 시작 시 Enter를 눌러주세요  
2. 정답은 대/소 문자를 가리지 않습니다  
3. 처음 시작 시 하트3개 지금 , 오답 제출 시 하트 1개씩 소멸됩니다.  
4. 하트3개 모두 소멸 시 게임 종료 / 재 시작시 처음부터 재시작 합니다  
5. 3번 연속 정답 시 다음 스테이지로 넘어갑니다 ( Stage 1 ~ 3 )  
6. 스테이지 3번 부터는 띄어쓰기 & 공백 까지 같아야 정답입니다.  
7. 오답 발생 시 스테이지는 처음부터 다시 시작합니다.  
8. 즉시 종료 방법 : exit 입력시 프로그램은 즉시 종료됩니다.  
  
-----  
  
.....::.....::.....::.....::.....::.....::.....:  
  
문제 [ 대/소문자 관계없음 ] : apple  
User 입력 >>
```

시작할때 나오는 화면 / Enter 입력 후 나오는 화면

Part 5 구현

오답 입니다

문제 [대/소문자 관계없음] : book
User 입력 >> book

정답입니다.

문제 [대/소문자 관계없음] : poverty
User 입력 >> pro

오답 입니다

정답으로 스테이지 이동시 출력, 오답일 시 나오는 출력

Part 5 구현

```
.....'#####:'##:'#####:'##:'#####:'##:'#####:'##:'#####:  
.....#####:'##: ##.... ##: ##: ##: ##....: ##: #####:  
.....#####:'##: ##: ##: ##: ##: ##: ##: ##: #####:  
..... ##: ##: ##: ##: ##: ##: ##: #####:'##:  
.....: ##: ##: ##: ##: ##: ##: ##: ##: ##: .....:  
.....#####:'##: ##: ##: ##: ##: ##: ##: ##: #####:  
..... ##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
.....: .....: .....: .....: .....: .....:  
  
게임 종료 Y / 오답 노트 확인 후 재시작 N : n  
  
◆ 총 정답 개수 : 9  
◆ 최종 스테이지 : Stage1  
  
오답 노트 >> slow and steady win the game, poverty, ,  
  
'#####:'##: ##: ##: #####:'##: ##: ##: #####:'##:  
##....: ##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
#####:'##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
##: ##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
##: ##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
##: ##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
#####:'##: ##: ##: ##: ##: ##: ##: ##: ##: #####:  
.....: .....: .....: .....: .....: .....:  
  
문제 [ 대/소문자 관계없음 ] : baseball  
User 입력 >>
```

재 시작시 나오는 문구

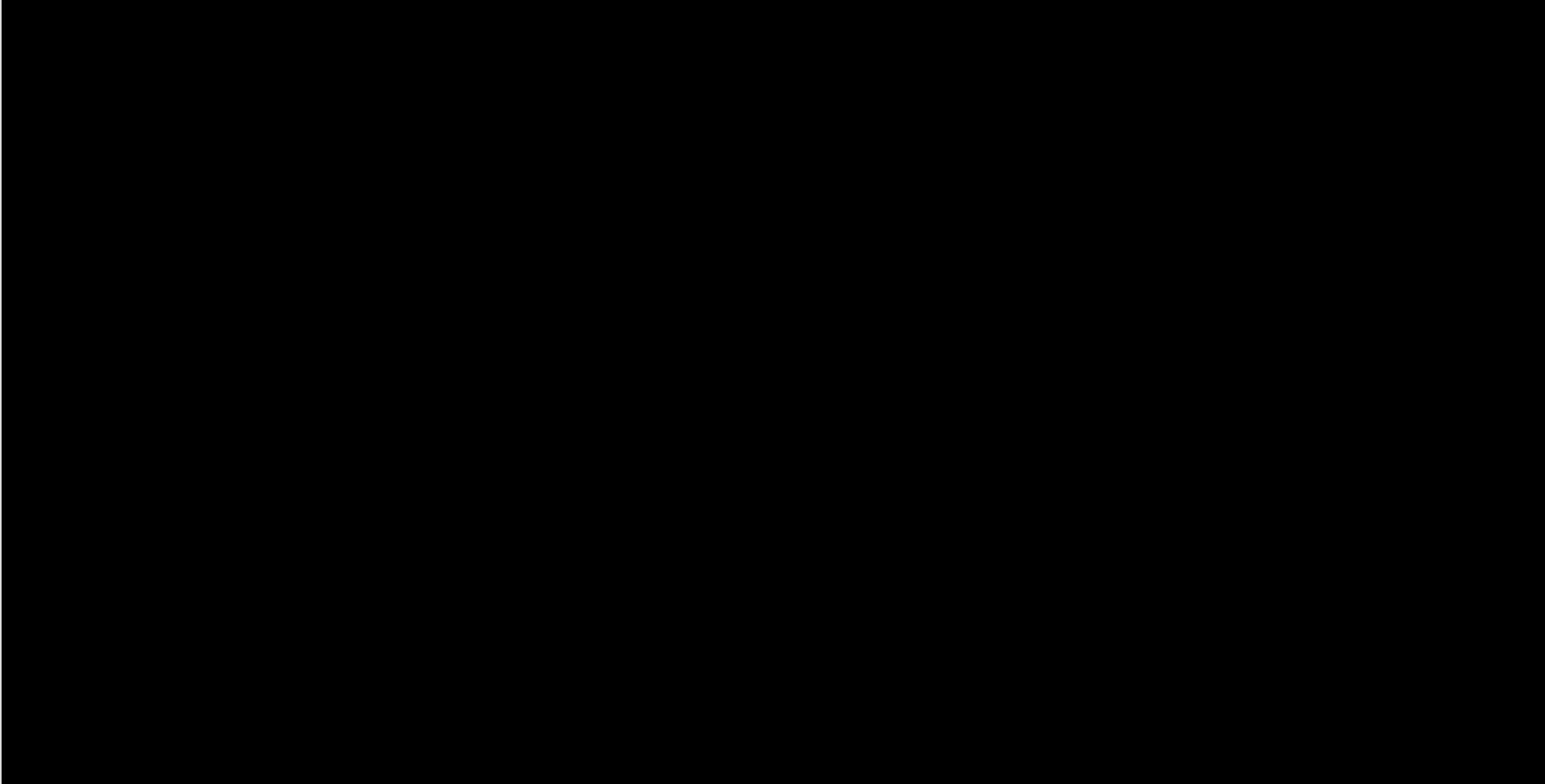
Part 5 구현



```
git add .
git commit -m "제작 1차기능"
git commit -m "제작 2차기능"
git commit -m "제작 3차기능"
git push
```

시작부터 종료 , 재시작 가능 시연

Part 5 구현



오답시 기능 시연
exit 입력 시 종료 시연

6

추가 기능 continue ..

- 2~3명의 플레이어들이 한 팀을 만든다
- 팀별 스테이지 클리어 속도전을 한다
- 같이 대결할 수 있는 기능

- 타자 게임이 좋다면 좋아요 기능
- 게임이 마음에 든다면 친구에게 초대하는 기능

- 오답 노트 실행 시 틀린 영단어를 다시 작성하는 기능
- 오답 노트 까지 모두 정답이 되어야 게임이 종료되게 하는 기능

- 타이머를 이용하여 평균 타자 속도를 계산하고 출력 기능
- 오답율과 정답율 평균 내어 키보드 입력 시 출력해주는 기능

7

후기

Review

이번 프로젝트를 통해 코딩을 따라가기에는 아직
부족한 부분이 많다는 것을 느꼈습니다.

그래도 저희 팀원들은 부족한 저를 포기하지 않고 함께 회의를 하며,
각자가 잘할 수 있는 부분들을
업무 분장하여 무사히 이 프로젝트를 마칠 수 있었습니다.
다음 프로젝트에는 팀원들에게 도움이 될 수 있게
앞으로 수업을 더 열심히 듣고, 잘 쫓아갈 수 있도록
노력하겠습니다

Review

프로젝트에서 제가 맡은 업무는 텍스트 순차적(시간차를 둬서 한 줄씩)
출력과 전체적 코드 오류 검토 및 수정, 비복원 추출 기능 추가 등을 맡았습니다.
처음에는 구현하는데 어려움을 느꼈지만 수업 시간에 배운 내용을 토대로
배운 것들을 활용하여 충분히 구현할 수 있었습니다.

Class를 따로 분리해서 메소드들을 만들었는데 가율님이 잘 가르쳐 주셔서 활용할 수 있었습니다.
이런 프로젝트를 할 때마다 팀원들과의 의사 소통도 늘고,
제가 작성한 코드와 팀원들이 작성한 코드를 비교를 해가며 작업을 해보니,
이런 프로젝트 덕분에 실력이 느는 제 자신을 발견할 수 있었습니다.
앞으로의 프로젝트도 지금처럼 주어진 업무에 최선을 다하도록 하겠습니다.

Review

프로젝트를 진행하면서 중요하게 생각했던 부분이 문제 해결을 위한 팀원들과 의사소통이었습니다.

부족한 부분을 성민님과 성도님이 채워주시고

서로 역할 분담을 짜임새 있게 잘 되어 손발이 맞는 팀이었습니다.

수업에서 아직 진도를 나가지 않았던 부분인(클래스 및 메서드, 생성자)에서
선생님의 당부 말씀이 걸려 고민을 하였지만, 스스로 코드를 작성하다 보니

유지 보수의 용이성과 가독성을 생각하게 되었습니다.

자료를 찾아가며 사용해 봤지만 아직 부족하다는 생각이 많이 듭니다.

앞으로 공부를 더 열심히 하여 복잡도 및 가독성,

유지 보수 용이성에 적합한 코드를 작성하도록 노력하겠습니다

THANK YOU .

