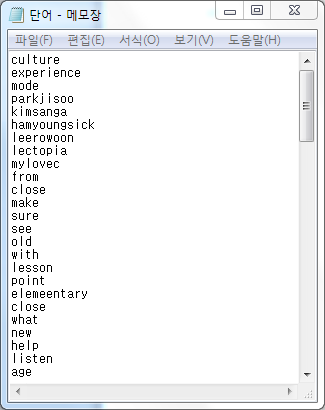
|  |  |
| --- | --- |
| **wordGame** | |
| int | **wordGame()**  전달인자 없음  리턴값은 score  단어를 생성빈도와 떨어지는 속도  각 함수들 호출  입력한 단어를 stringcheck() 함수로 비교  floorUp의 리턴값이 1이면 게임종료 |
| void | **wordOk(Word\* w)**  전달인자로 Word 구조체배열의 주소를 받는다.  0~99번 방중에 한개의 방을 랜덤으로 아직 단어가 출력되지 않은 것들 중에서  (ok == 0) 출력여부(ok)를 출력함(1)으로, x, y축 좌표를 초기화 |
| int | **floorUp(int \*floor,Word \*w)**  전달인자로 바닥값의 주소와 Word 구조체배열의 주소를 받는다.  리턴값은 바닥값이 2(y축)일 경우 게임종료, 0이면 종료아님  0번방부터 99번방까지 화면에 출력중인 단어(ok==1)중에 바닥값과 y축기 같으면  잔상을 지우고 출력여부(ok)를 출력안함(0)으로 바꾸고 바닥값을 감소한후 floorDisplay()호출 |
| void | **floorDisplay(int floor)**  전달인자로는 바닥값을 받는다.  바닥구분선을 출력하고 바닥값에 따라 메시지를 출력한다. |
| void | **floorFrame(void)**  전달인자 없음.  단어입력 칸과 점수칸을 출력한다. |
| void | **stringcheck(Word \*w, char \*cp, int \*score)**  전달인자로 Word 구조체배열의 주소와, 입력한 단어의 주소와 점수의 주소를 받는다.  0번방부터 99번방까지 모두검사  화면에 출력중인 단어(ok==1)중에 사용자가 입력한 단어와 일치하면 잔상을 지워주고 점수를 올려준뒤 출력여부(ok)를 출력안함(0)으로 초기화한다. |
| void | **yPlus(Word\* w)**  0번방부터 99번방까지 화면 출력중인 단어(ok==1)의 y축을 1씩 증가 |
| void | **wordDisplay(Word\* w)** 0번방부터 99번방까지 화면 출력중인 단어(ok==1)만 잔상을 지우고 출력하기 |
| void | **wordLoad(Word\* w)**  전달인자로 Word 구조체배열의 주소를 받는다.  단어.txt에서 한줄씩 읽어와서 구조체배열에 저장하고 화면에출력여부(ok)를  출력안함(0)으로, 단어의 길이(len)를 초기화 한다. |
| void | **beginDisplay(void)**  전달인자 없음.  Signal 구조체배열 선언(방 100개)  랜덤으로 0번방부터 99번방까지 화면 출력여부(ok)가 출력안함(0)인 단어를  ok=1, ch=알파벳랜덤으로, x,y축, 거리, 속도 초기화  알파벳과 타자GAME을 출력 |

파일설계



구조체

struct Word

{

char word[100];

int x,y; //좌표

int len; //단어길이

int ok; //화면에 출력여부

};

struct Signal

{

int ok; //화면에 출력여부

char ch; //출력할 문자

int x,y; //좌표

int distance; //이동할 거리

int nFrame; //속도

int nStay; //속도에 대한 카운트

};