객체지향프로그래밍

Assignment 3

학과: 수학과

학번: 20140843

이름: 엄태강

POVIS ID: tkeom0114

Honor Code:나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

1. 개요

이번 과제에서는 C++을 사용하여 타자 연습게임을 작성하였다. 단어들의 종류가 여러가지이기 때문에 상속과 다형성을 이용해서 이번 과제를 구현하였다.

2.class 구성

이번 과제에서는 단어에 대한 정보를 가지는 word를 base class로 가지고 그 아래에 다음 그림과 같이 class hierarchy를 구성하였다.

가장 기본적인 단어인 word는 아래와 같이 구성되어 있다.

class Word{

private:

int line; Word가 위치한 행을 나타낸다(1~5사이의 값을 가진다).

int position; Word가 위치한 열을 나타낸다(1~5사이의 값을 가진다).

string word; Word가 어떤 단어인지 나타낸다.

bool isWord; object가 Word인지 derived class 인지 나타낸다(destructor에 사용됨)

static int hp; 사용자의 체력을 나타낸다. 초기값은 100이며 0이하이면 게임이 끝난다.

static int score; 사용자의 점수를 나타낸다. 초기값은 0이다.

protected:

static void setHP (int); 체력을 새로 설정한다.

static void setScore (int); 점수를 새로 설정한다.

public:

Word (string a,int b,int c): Word의 Constructor이다.

{word = a;line = b;position = c;isWord = true;}

virtual ~Word (void); Word의 Destructor이며 해당 Word가 사라질 때, 1점을 올린다.

virtual void printWord (void); word를 출력한다.

string getWord(void); Word의 단어를 return한다.

int getLine(void); Word가 위치한 행을 return한다.

int getPosition(void); Word가 위치하 열을 return한다.

void setIsWord(bool); Object가 만들어 질 때, Word가 아니면 false로 만든다.

virtual bool getNext(void); false를 return한다. (다른 class에서는 count가 0일 때 true를 return하는 함수가 됨)

virtual int getCount(void); 1을 return한다. (다른 class에서는 count를 return한다.)

static int getHP (void); 체력을 return한다.

static int getScore (void); 점수를 return한다.

};

회복 단어는 Word를 상속받아서 아래와 같은 class로 구현했다.

class RecoveryWord: public Word{

private:

public:

RecoveryWord (string a,int b,int c) : Word (a,b,c){setIsWord(false);}

RecoveryWord의 constructor이다. Word의 derived class이므로 isWord는 false로 만든다.

~RecoveryWord (void); RecoveryWord의 destructor로 체력과 점수를 각각 10, 1을 올리고 제거된다.

virtual void printWord (void); (word)를 출력한다.

};

폭탄 단어는 Word를 상속받아서 아래와 같은 class로 구현했다.

class BombWord: public Word{

private:

public:

BombWord (string a,int b,int c) : Word (a,b,c){setIsWord(false);}

BombWord의 constructor이다. Word의 derived class이므로 isWord는 false로 만든다.

~BombWord (void); BombWord의 destructor이다. 체력과 점수를 각각 20,2씩 감소시키고 제거된다.

virtual void printWord (void); <word>를 출력한다.

};

완전 회복 단어는 Word를 상속받아서 아래와 같은 class로 구현했다.

class CompleteRecoveryWord: public Word{

private:

int count; 초기값은 3이며 한 턴에 1씩 감소한다.

bool sucide; count가 다 되어서 사라지는 단어이면 true이다.

bool isComRecover; object가 derived class가 아닌 CompleteRecoveryWord이면 true이다.

public:

CompleteRecoveryWord (string a,int b,int c) : Word (a,b,c){setIsWord(false);count = 3;sucide = false;isComRecover = true;}

CompleteRecoveryWord의 Constructor이다.

~CompleteRecoveryWord (void); CompleteRecoveryWord의 destructor이고 체력을 100으로 만들고 사라진다.

void minusCount (void); count를 1만큼 줄인다.

bool getSucide (void); sucide를 return한다.

void setIsComRecover (bool); derived class의 경우 isComRecover를 false로 만든다.

virtual int getCount (void); count를 return한다.

virtual void printWord (void); ((word(count))를 출력하고 count를 1만큼 줄인다.

virtual bool getNext(void); count가 0인 경우 destructor를 호출하고, true를 return한다.

};

죽음의 폭탄 단어는 CompleteRecoveryWord를 상속받아서 아래와 같은 class로 구현했다. 그 이유는 count와 sucide를 상속받기 위해서이다.

class DeathlyBombWord: public CompleteRecoveryWord{

public:

DeathlyBombWord (string a,int b,int c) : CompleteRecoveryWord (a,b,c){setIsWord(false);setIsComRecover(false);}

DeathlyBombWord의 constructor이며 isComRecover를 false로 만든다.

~DeathlyBombWord (void); DeathlyBombWord의 destructor이며 체력을 0으로 만든 후 제거된다.

virtual void printWord (void); <<word(count)>>를 출력하고 minusCount를 호출한다.

};

3.알고리즘

1) main

①모든 단어들을 모은 string[20] WordList와 생성될 수 있는 단어들의 list인 list<string> ableWordList, 화면에 있는 단어들의 list인 list <Word\*> wordlist, 이전 턴에 지워져서 다음턴에 바로 생성되면 안되는 단어들의 list인 list<string> erasedWord를 만든다.

②ableWordList에 wordList의 모든 단어들을 삽입한다.

③createWord를 반복 호출해 10개의 Word(와 derived objects)들을 만든다.

④printWords를 통해 단어들을 출력하고 입력을 받는다. 입력받은 단어는 printWords를 통해 return해서 erasedWord에 삽입한다.

⑤count가 0인 모든 단어들을 erasedWord에 삽입한다.

⑥wordList.remove\_if(countIsZero)를 통해 count가 0인 모든 word의 메모리 할당을 해제하고 wordList에서도 지운다.(countIsZero는 word()->get\_next()를 return하는 함수이다.)

⑦체력이 0이하인 경우 점수를 출력하고 게임을 종료한다. 아니면 ③으로 돌아간다.

2)단어 생성(void createWord (list <string> &ableWordList,list <Word\*> &wordList) )

①srand(time(0))로 rand함수의 seed를 준다.

②rand를 통해 새로 생길 단어를 ableWordList에서 고르고 ableWordList에서 제거한다.

③rand를 통해 새로 생길 단어의 line과 position을 정한다. 다른 단어가 쓰고 있는 위치면 다시 정한다.

④select의 값을 rand를 통해 0~9사이 중 하나로 정하고 그 값에 따라 생성되는 단어의 종류를 정한다.

⑤정해진 단어, 위치, object의 종류를 통해 새로운 object를 만든다.

⑥새로운 object를 wordList에 삽입하고 위치 순서대로 정렬한다.

3)단어 출력 및 입력(string printWords (list <string> &ableWordList,list <Word\*> &wordList))

①wordList의 단어들을 정해진 위치에 출력한다.

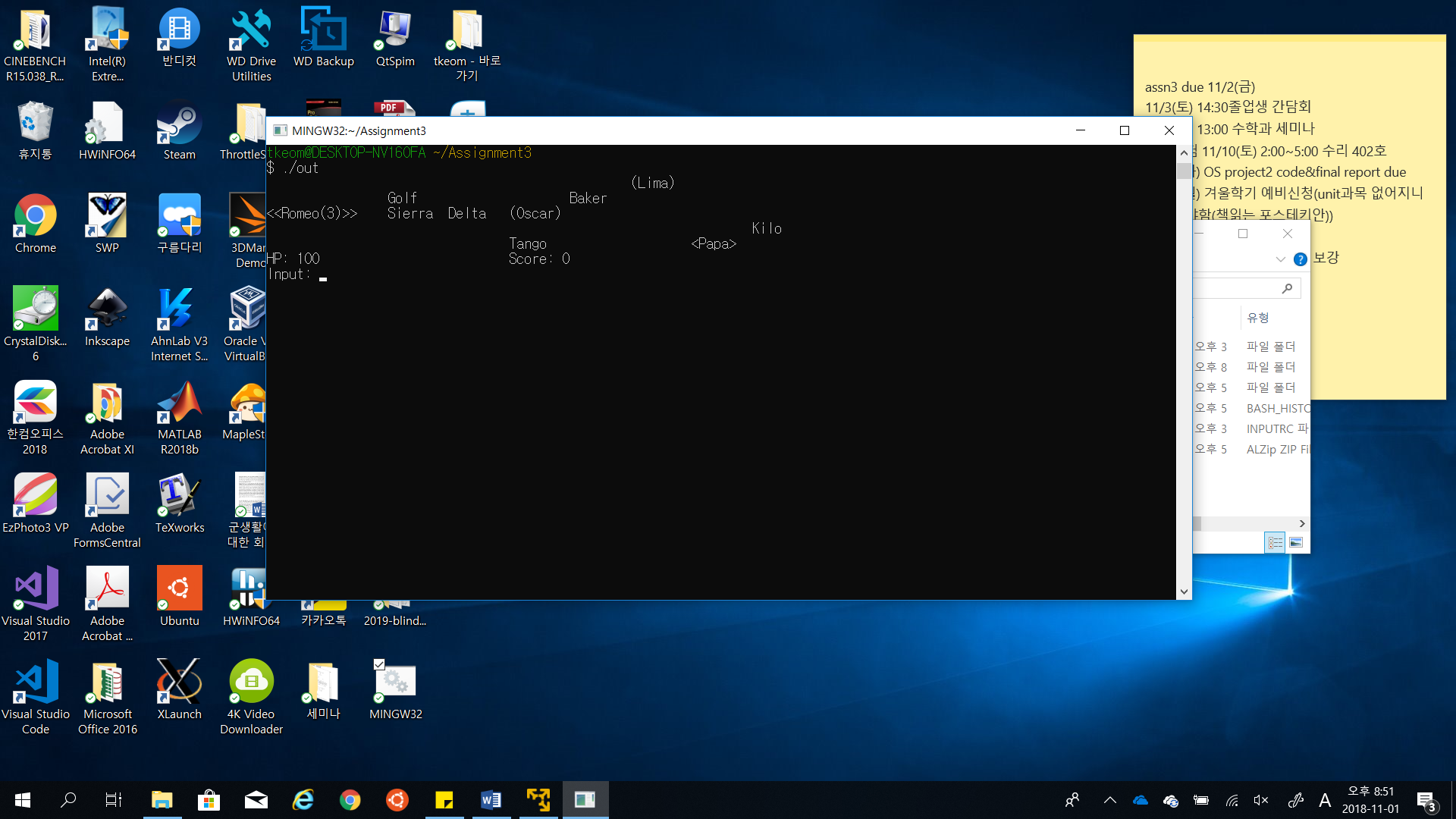
②단어를 입력 받는다. wordList에 없으면 다시 입력 받는다.

③그 단어에 해당하는 object를 찾아 할당 해제시키고 wordList에서 제거한다.

4.실행방법

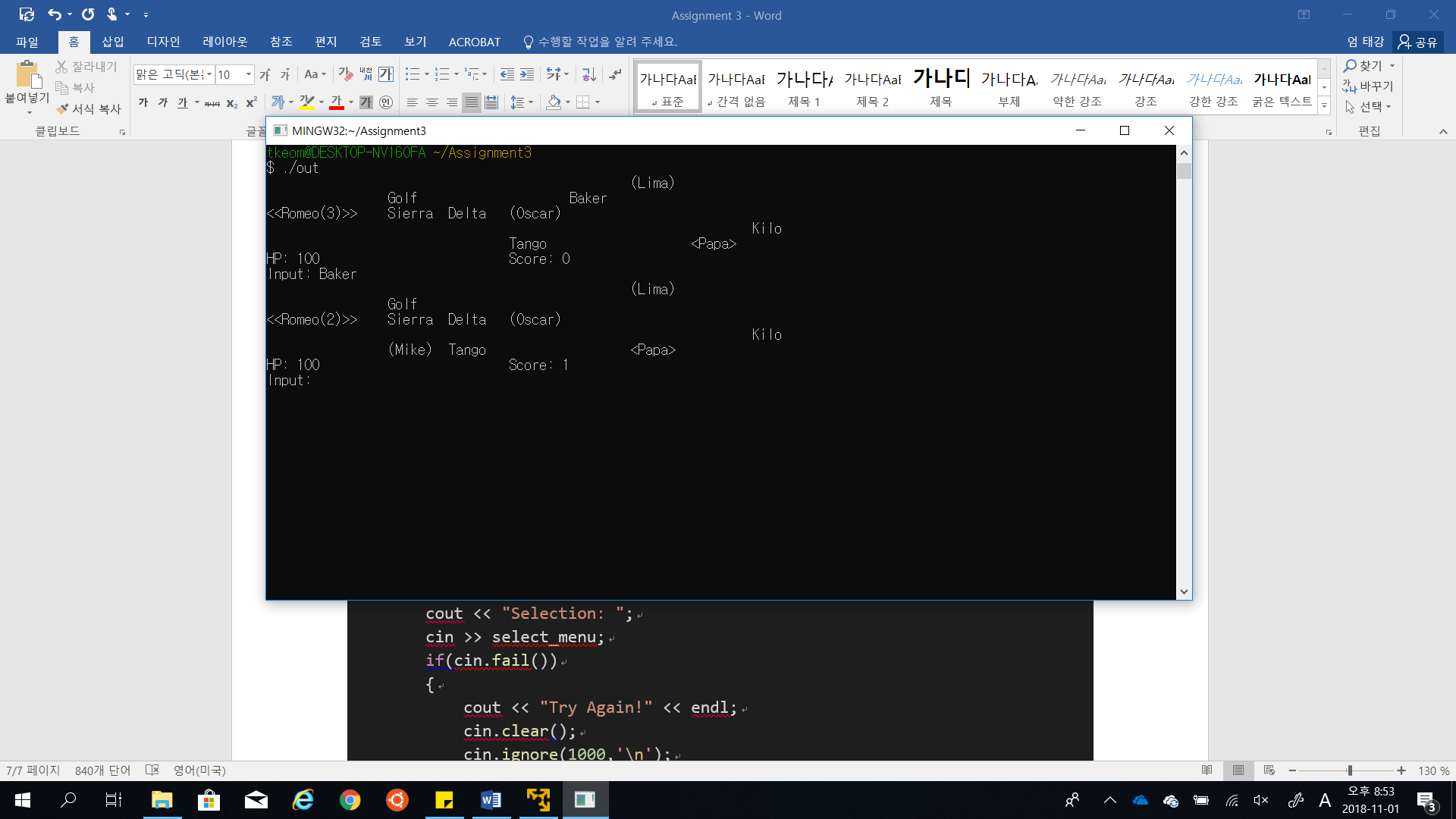
make 후 out파일을 아래와 같이 단어가 출력되고 화면에 보이는 단어를 입력해 게임을 시작하면

된다.

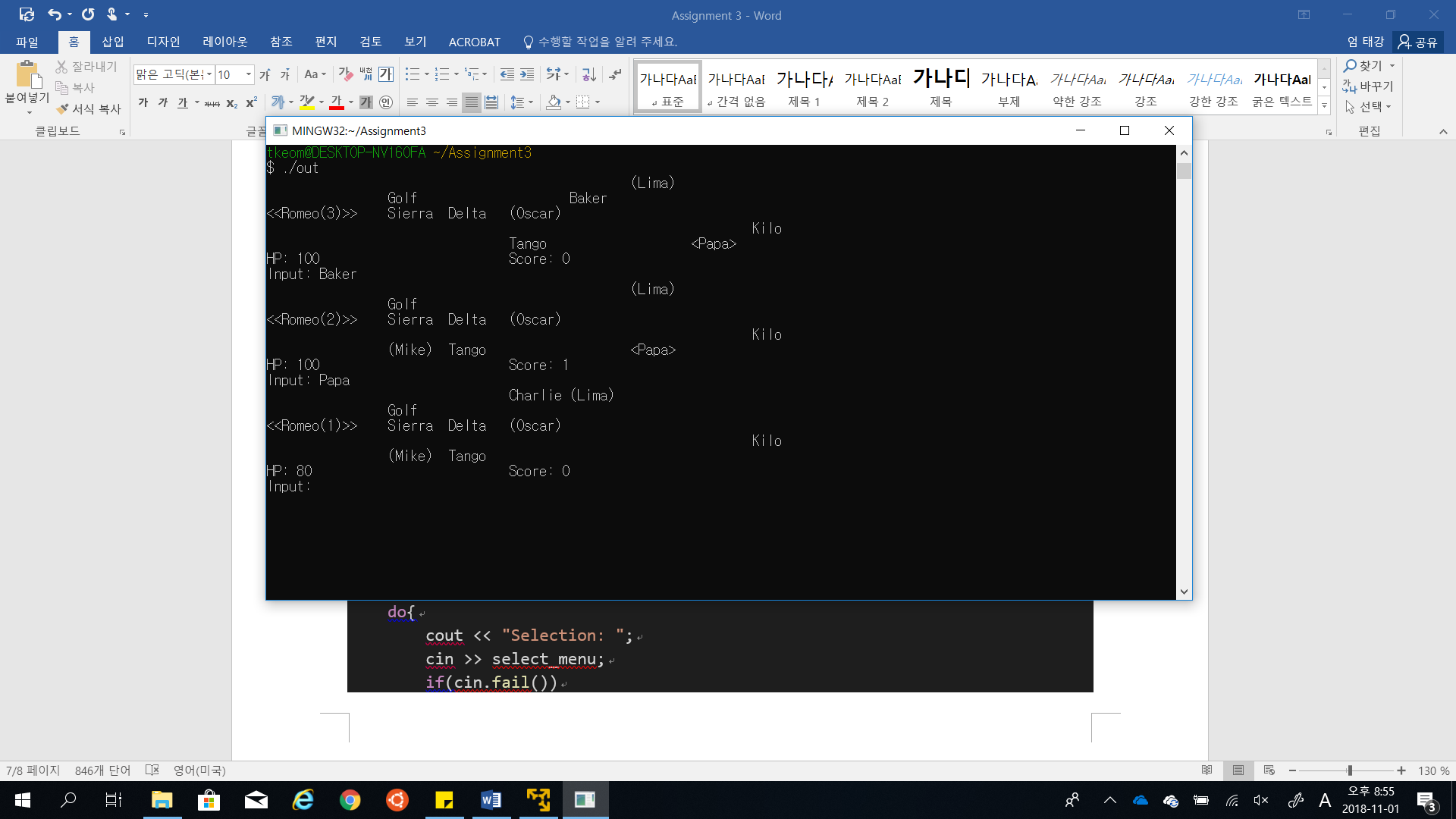


5.실행예시

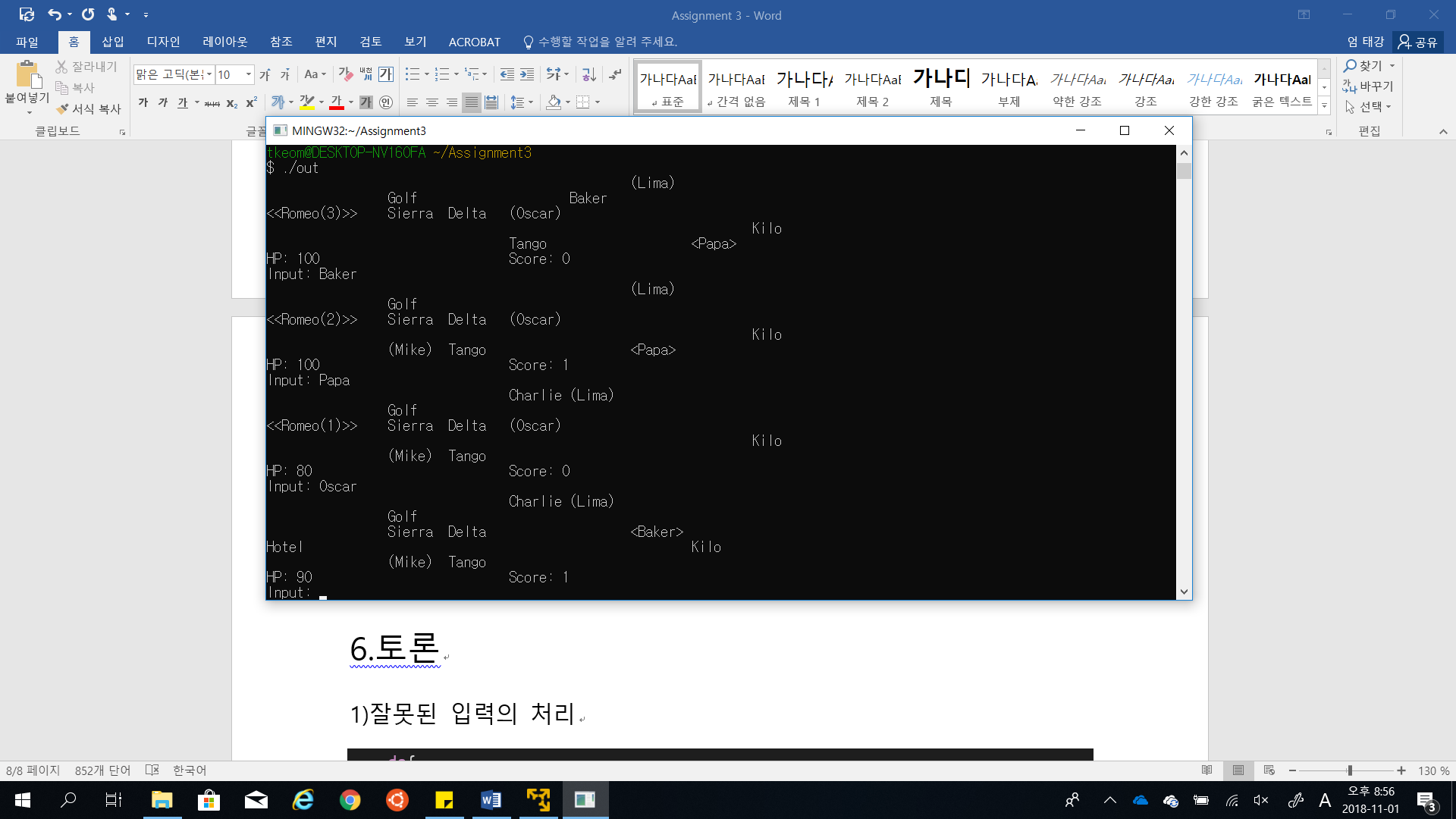
일반 단어를 입력하면 아래와 같이 된다.



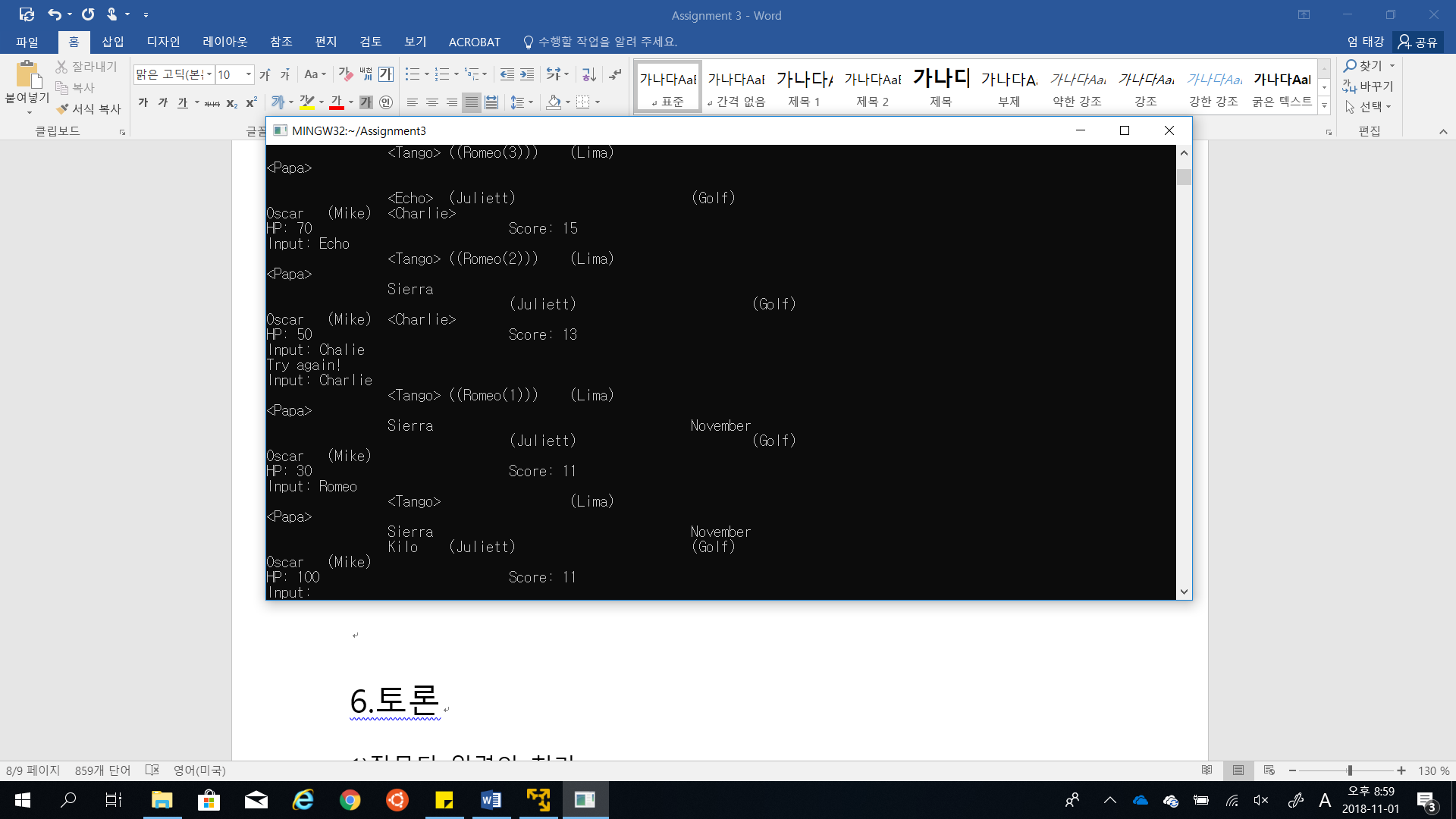
폭탄 단어를 입력하면 아래와 같이 된다.



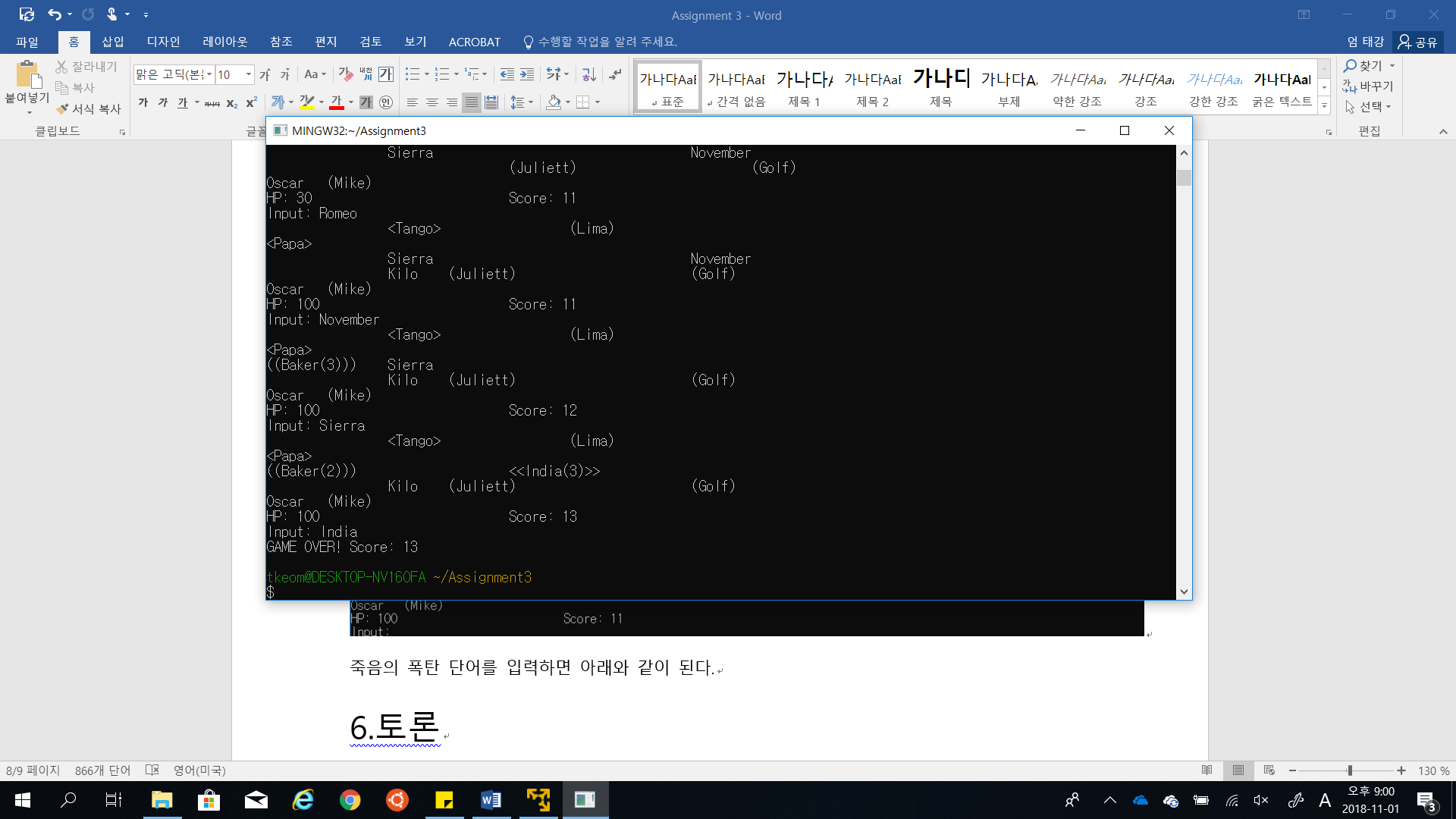
회복 단어를 입력하면 아래와 같이 된다.



완전 회복 단어를 입력하면 아래와 같이 된다.



죽음의 폭탄 단어를 입력하면 아래와 같이 된다.



6.토론

1)polymorphism을 이번 assignment에서 사용한 이유

사실 이번 과제에서는 상속 개념만 사용하면 되는 과제였다. 하지만 굳이 배운지 얼마 안된 polymorphism을 사용한 이유는 그렇게 하면 모든 단어들을 한 list에서 관리할 수 있기 때문이다. running time에 type을 판별하게 하면 한 list에서 object의 type에 따라 print가 다르게 수행되기 때문에 모든 종류의 단어들을 print할 수 있다. 그리고 count를 확인하는 경우에도 Word에 virtual 함수를 만들어 count가 없더라도 실행되는데 아무 문제가 없도록 만들 수 있었다.

7.결론

이번 과제에서는 C++을 사용하여 단어 입력 게임을 구현했다. 이번 프로젝트에서는 특히 객체지향 프로그래밍의 상속과 다형성을 이용해 프로그램을 작성해 보았다. 다형성 개념은 사용하라는 말은 없었지만 그 개념을 사용해서 훨씬 더 편하게 이번 과제를 수행할 수 있었다.