



# [2015SMA\_T3]

## 영단어 수집왕

Implementation & Unit Test

201011320 김용현  
201111360 손준익  
201111347 김태호

# INDEX.

---

---

## 1. Revise Plan

---

---

## 2. User Interface

---

---

## 3. Implement Focus

---

---

## 4. Unit Test

---

---

## 5. Demonstration

---

---

# 1

## Revise Plan

Revise Plan

### Revise Plan

#### Class Diagram

- Contents – Xpos:int, Ypos:int., getXPos(), getYPos(), setXPos(), setYPos() 삭제
- Alphabet – movePos(), setXPos(), playSound() 삭제
- Word – url2:String, playSound() 삭제
- Dictionary\_Controller - setDictionaryStatus(), getDictionaryStatus() 삭제
- Main\_Controller – db:Database 추가

#### Sequence Diagram

- 모든 Xpos, Ypos에 관한 function 삭제
- 조건문 추가(1개)

#### Database

- Web DB사용 -> 내장 DB사용(Sqlite)

---

# 2 User Interface

---



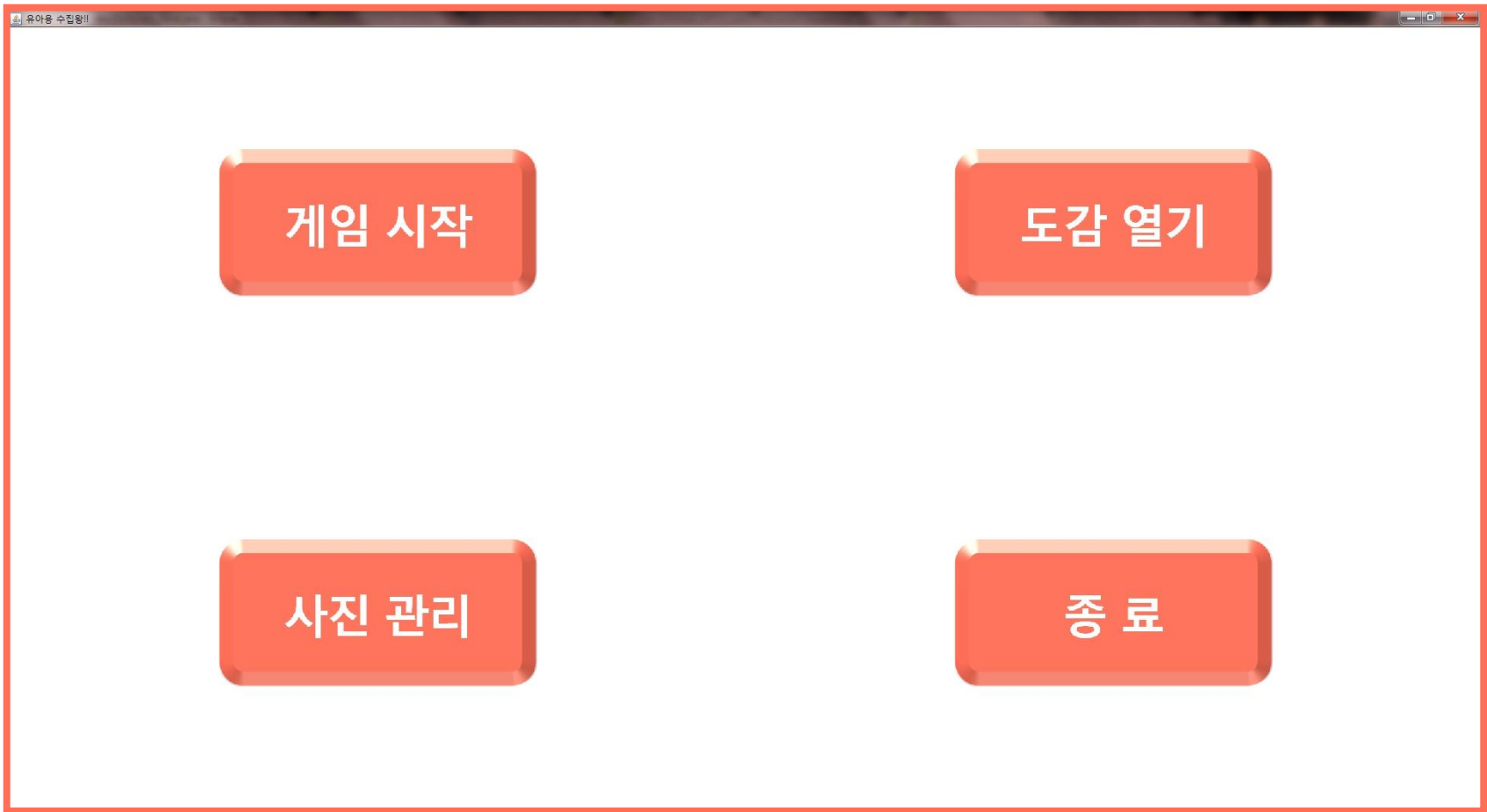
User  
Interface

---

# 2 User Interface

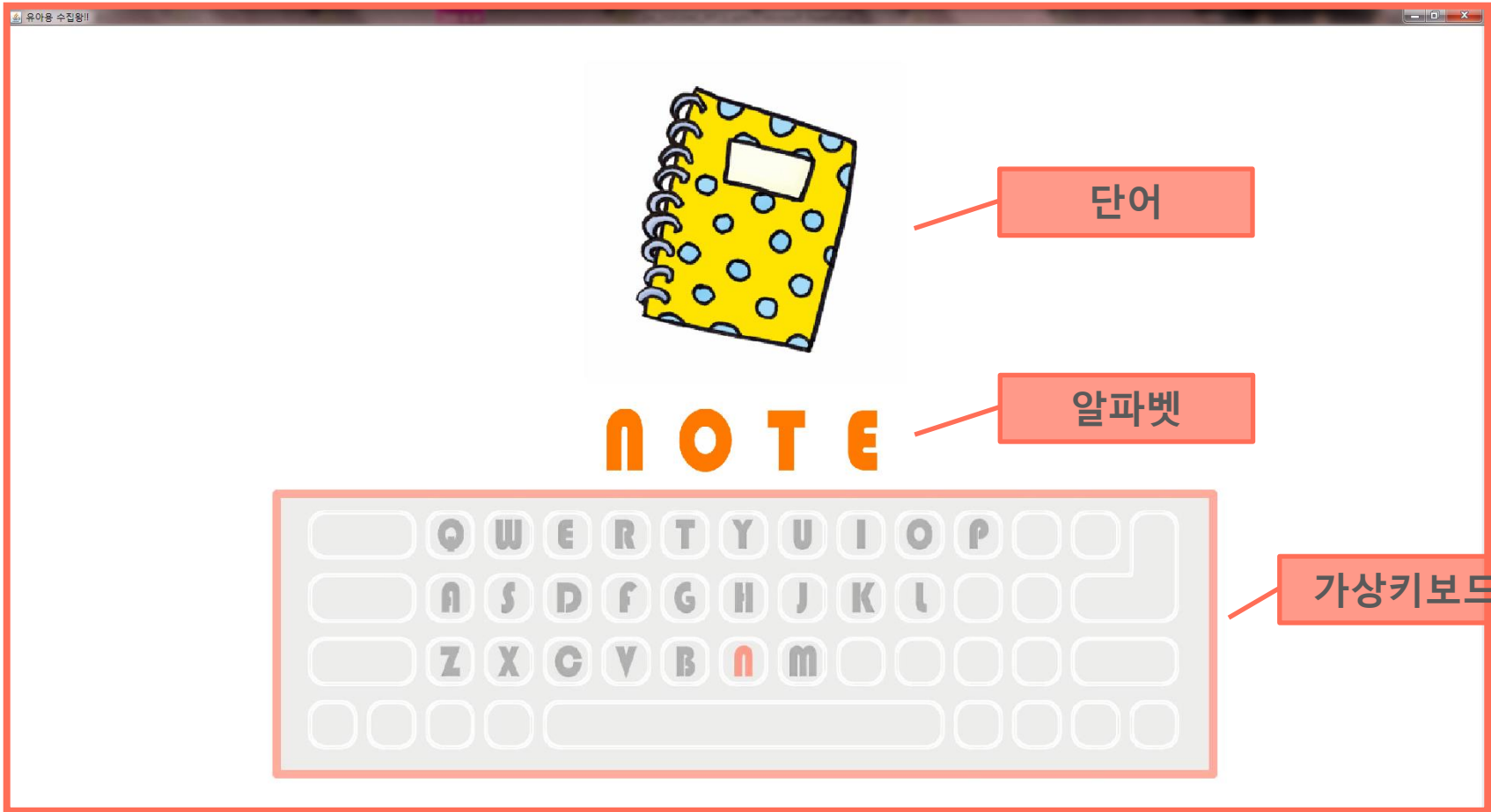
Main Menu

---



# 2 User Interface

게임 시작 - Play



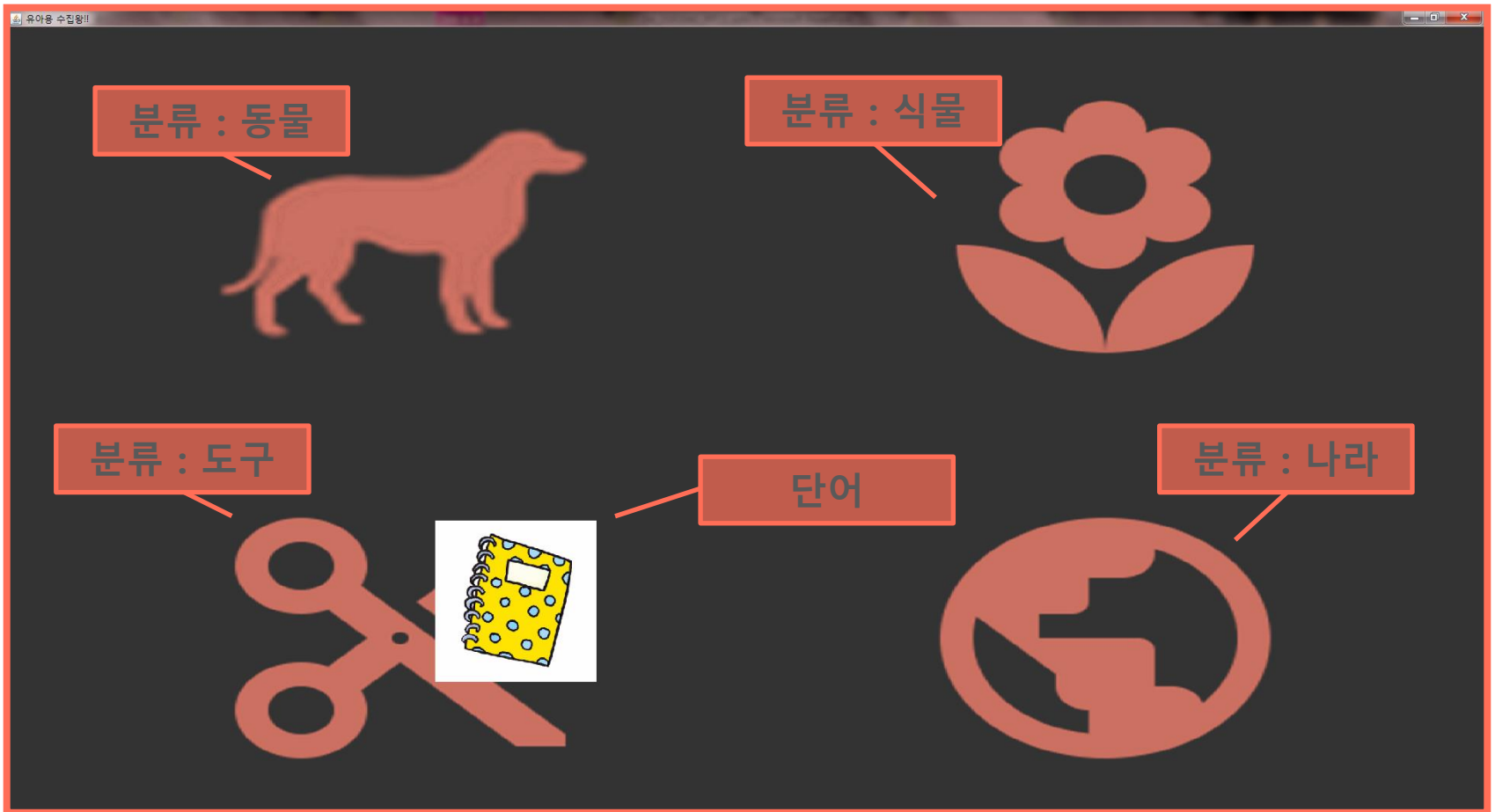
# 2 User Interface

게임 시작 - Display Reward Image



# 2 User Interface

게임 시작 - Category Word





# 2 User Interface

도감 열기 - 도감

검색창

발견한 단어

검색창

Enter

Animal

Plant

미발견

Tool

Nation

---

# 2 User Interface

도감 열기 - 검색

---



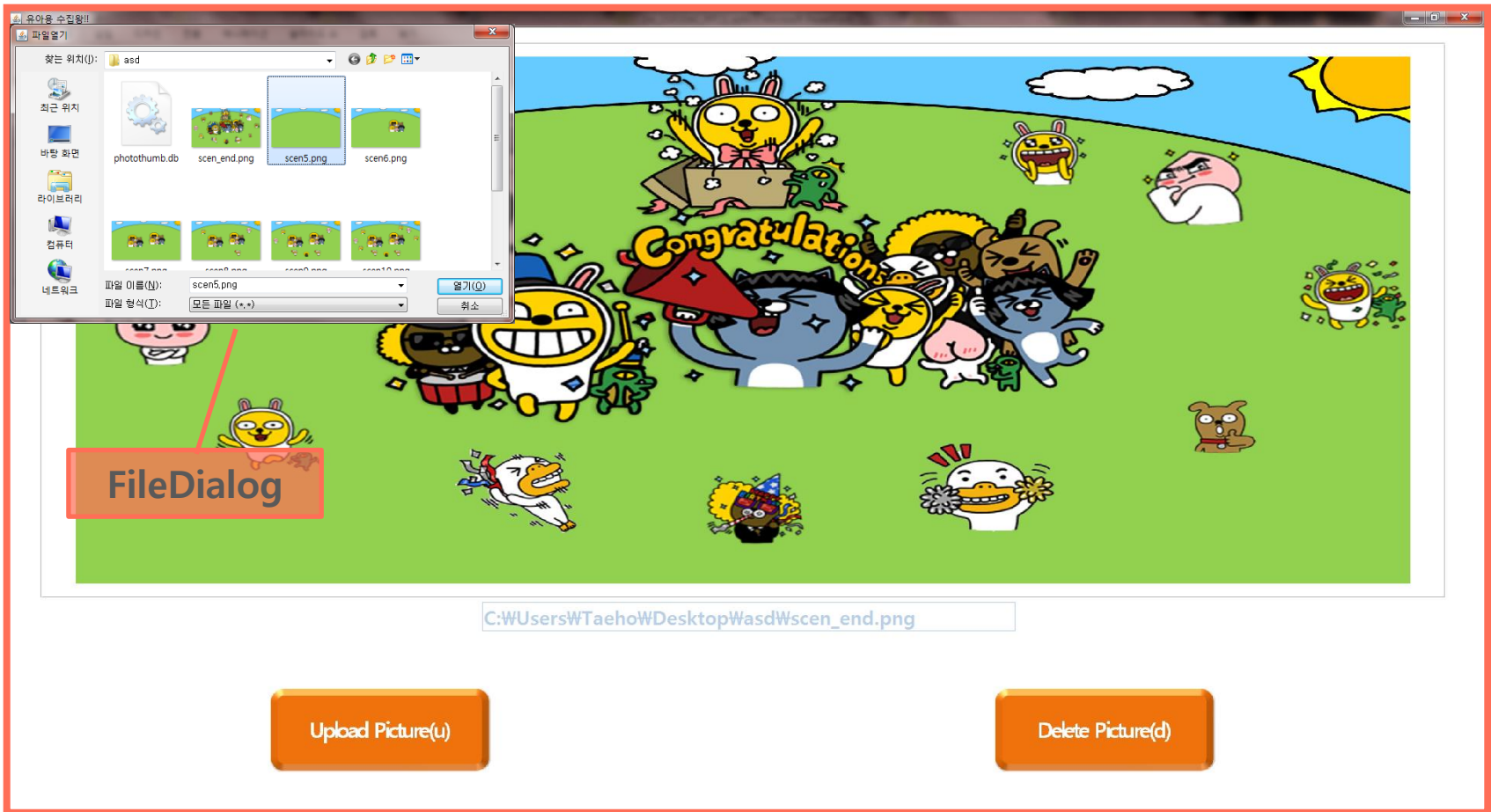
# 2 User Interface

사진 관리 - 사진



# 2 User Interface

사진 관리 - Upload Picture



# 2 User Interface

사진 관리 - Set Upload Picture



# 2 User Interface

사진 관리 - Delete Picture



---

# 3 Implement Focus

---



Implement  
Focus

# 3 Implement Focus

Display

## Comment

모든 해상도에서 최적화

1280x768



FOLDER



1600x900



IGUANA



800x600



GLUE



1366x768



MONKEY



1920x1080



CAT



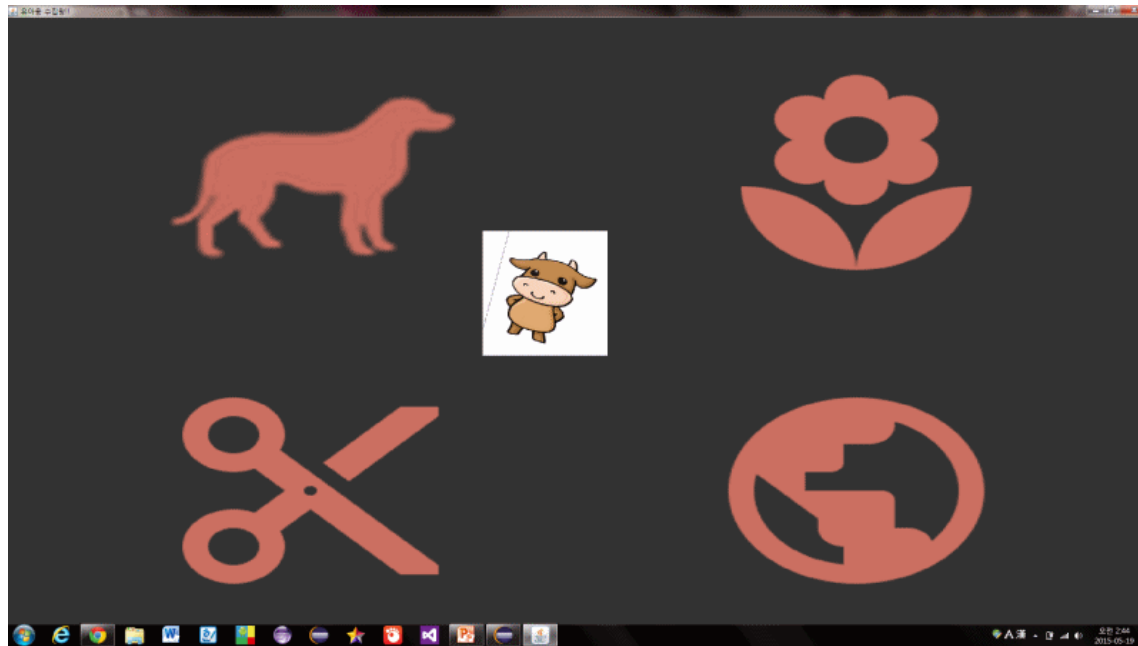


# 3 Implement Focus

Display

## Comment

이미지 이동




# 3 Implement Focus

Education

## Comment

000 = 유아용 수집왕

Animal	Plant
	
	
	
Tool	Nation
	
	
	

---

# 3 Implement Focus

Program

---

## Comment

프로그램 이식성

SQL Lite를 사용한 이식성 높은 Database 구성  
Java Swing Component 기반 GUI  
모든 해상도 지원

---

# 4 Unit Test

---



Unit Test

# 4 Unit Test

Total Test Case & Method

Class	Method_Name	Class	Method_Name
Main_Controller	quit()	Database	DBConnect()
Game_Controller	PressAlphabet()	Database	SelectGameWord()
Game_Controller	CheckKeyboardImage()	Database	SelectGameAlphabet()
Game_Controller	CheckWord()	Database	UpdateWord()
Game_Controller	GameStart()	Database	InitializeDictionary()
Game_Controller	SelectWord()	Database	SearchCorrectWord()
Game_Controller	CheckAlphabet()	Database	SelectRewardImageURL()
Dictionary_Controller	SearchbyInitial()	Database	UpdateRewardImage()
Dictionary_Controller	SearchByFullWord()	Database	UpdateRewardToDefault()
Dictionary_Controller	Search()	Database	SelectKeyboardImage()
Dictionary_Controller	DictionaryOpen()	Database	SelectWordbyText()
Picture_Controller	DeletePicture()	Database	SelectRandomWordbyAlphabet()
Picture_Controller	ManagePicture()	Dictionary	AddWordArr()
Picture_Controller	SelectFileDirectory()	Word	SetAlphabetArr()
		Word	AppendAlphabet()

**Method : 27**

**Test Case : 61**

**Pass : 61**

**Fail : 0**

# 4 Unit Test

Game Controller

Runs: 15/15 ✖ Errors: 0 ✖ Failures: 0

controller.Game\_ControllerTest [Runner:

- testCheckWord1 (0.203 s)
- testCheckWord2 (0.016 s)
- testPressAlphabet1 (0.000 s)
- testPressAlphabet2 (0.016 s)
- testPressAlphabet (0.015 s)
- testCheckAlphabet1 (0.016 s)
- testCheckAlphabet2 (0.000 s)
- testCheckAlphabet3 (0.000 s)
- testCheckKeyboardImage1 (0.016 s)
- testCheckKeyboardImage2 (0.000 s)
- testCheckKeyboardImage (0.000 s)
- testCheckWord (0.015 s)
- testGameStart (0.016 s)
- testSelectWord (0.015 s)
- testCheckAlphabet (0.000 s)


```
@Test
public void testPressAlphabet() {
    Word word = new Word();
    word.setName("alligator");
    word.setLength(9);
    word.setAlphabetArr();
    gamecontrol.setWord(word);

    gamecontrol.pressAlphabet('a');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(0).getCorrect());
    gamecontrol.pressAlphabet('l');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(1).getCorrect());
    gamecontrol.pressAlphabet('l');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(2).getCorrect());
    gamecontrol.pressAlphabet('i');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(3).getCorrect());
    gamecontrol.pressAlphabet('g');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(4).getCorrect());
    gamecontrol.pressAlphabet('a');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(5).getCorrect());
    gamecontrol.pressAlphabet('t');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(6).getCorrect());
    gamecontrol.pressAlphabet('o');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(7).getCorrect());
    gamecontrol.pressAlphabet('r');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(8).getCorrect());
    assertTrue(gamecontrol.getWord().getCorrect());
    //fail("Not yet implemented");
}
```

# 4 Unit Test

Game Controller

Runs: 15/15 ✖ Errors: 0 ✖ Failures: 0



controller.Game\_ControllerTest [Runner:

- testCheckWord1 (0.203 s)
- testCheckWord2 (0.016 s)
- testPressAlphabet1 (0.000 s)
- testPressAlphabet2 (0.016 s)
- testPressAlphabet (0.015 s)
- testCheckAlphabet1 (0.016 s)
- testCheckAlphabet2 (0.000 s)
- testCheckAlphabet3 (0.000 s)
- testCheckKeyboardImage1 (0.016 s)
- testCheckKeyboardImage2 (0.000 s)
- testCheckKeyboardImage (0.000 s)
- testCheckWord (0.015 s)
- testGameStart (0.016 s)
- testSelectWord (0.015 s)
- testCheckAlphabet (0.000 s)


```
@Test
public void testPressAlphabet1() {
    Word word = new Word();
    word.setName("----");
    word.setLength(4);
    word.setAlphabetArr();
    gamecontrol.setWord(word);

    gamecontrol.pressAlphabet('-');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(0).getCorrect());
    gamecontrol.pressAlphabet('-');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(1).getCorrect());
    gamecontrol.pressAlphabet('-');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(2).getCorrect());
    gamecontrol.pressAlphabet('-');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(3).getCorrect());
    assertTrue(gamecontrol.getWord().getCorrect());
    //fail("Not yet implemented");
}
```

# 4 Unit Test

Game Controller

Runs: 15/15 ✖ Errors: 0 ✖ Failures: 0



controller.Game\_ControllerTest [Runner:

- testCheckWord1 (0.203 s)
- testCheckWord2 (0.016 s)
- testPressAlphabet1 (0.000 s)
- testPressAlphabet2 (0.016 s)
- testPressAlphabet (0.015 s)
- testCheckAlphabet1 (0.016 s)
- testCheckAlphabet2 (0.000 s)
- testCheckAlphabet3 (0.000 s)
- testCheckKeyboardImage1 (0.016 s)
- testCheckKeyboardImage2 (0.000 s)
- testCheckKeyboardImage (0.000 s)
- testCheckWord (0.015 s)
- testGameStart (0.016 s)
- testSelectWord (0.015 s)
- testCheckAlphabet (0.000 s)

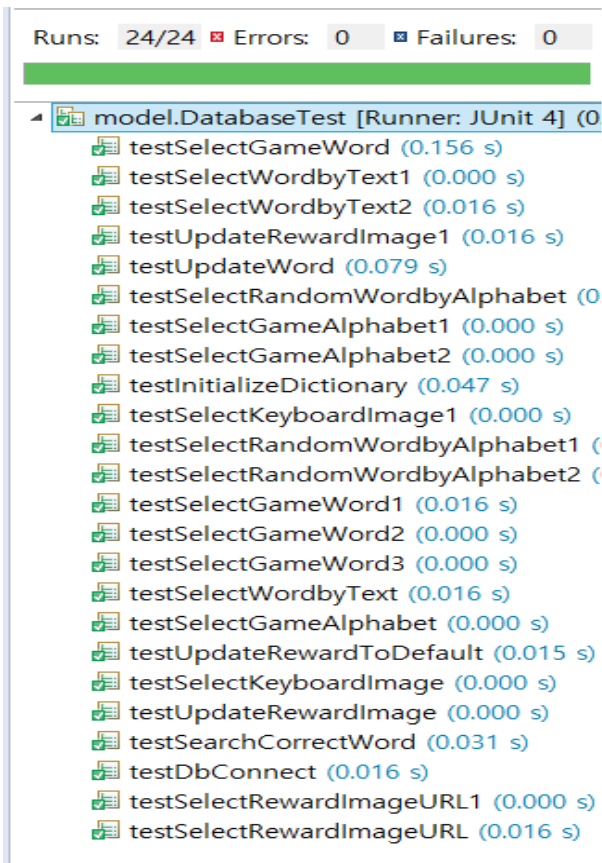
```
@Test
public void testPressAlphabet2() {
    Word word = new Word();
    word.setName("g01*zd%");
    word.setLength(4);
    word.setAlphabetArr();
    gamecontrol.setWord(word);

    gamecontrol.pressAlphabet('g');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(0).getCorrect());
    gamecontrol.pressAlphabet('0');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(1).getCorrect());
    gamecontrol.pressAlphabet('1');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(2).getCorrect());
    gamecontrol.pressAlphabet('*');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(3).getCorrect());
    gamecontrol.pressAlphabet('z');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(4).getCorrect());
    gamecontrol.pressAlphabet('d');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(5).getCorrect());
    gamecontrol.pressAlphabet('%');
    assertTrue(gamecontrol.getWord().getAlphabetArr().elementAt(6).getCorrect());
    assertTrue(gamecontrol.getWord().getCorrect());
    //fail("Not yet implemented");
}
```



# 4 Unit Test

Database



Runs: 24/24 Errors: 0 Failures: 0

- model.DatabaseTest [Runner: JUnit 4] (0)
  - testSelectGameWord (0.156 s)
  - testSelectWordbyText1 (0.000 s)
  - testSelectWordbyText2 (0.016 s)
  - testUpdateRewardImage1 (0.016 s)
  - testUpdateWord (0.079 s)
  - testSelectRandomWordbyAlphabet (0.000 s)
  - testSelectGameAlphabet1 (0.000 s)
  - testSelectGameAlphabet2 (0.000 s)
  - testInitializeDictionary (0.047 s)
  - testSelectKeyboardImage1 (0.000 s)
  - testSelectRandomWordbyAlphabet1 (0.000 s)
  - testSelectRandomWordbyAlphabet2 (0.000 s)
  - testSelectGameWord1 (0.016 s)
  - testSelectGameWord2 (0.000 s)
  - testSelectGameWord3 (0.000 s)
  - testSelectWordbyText (0.016 s)
  - testSelectGameAlphabet (0.000 s)
  - testUpdateRewardToDefault (0.015 s)
  - testSelectKeyboardImage (0.000 s)
  - testUpdateRewardImage (0.000 s)
  - testSearchCorrectWord (0.031 s)
  - testDbConnect (0.016 s)
  - testSelectRewardImageURL1 (0.000 s)
  - testSelectRewardImageURL (0.016 s)

```
@Test
public void testUpdateWord() {
    Dictionary_Controller dic_c = new Dictionary_Controller();
    Word word = new Word();
    word.setName("monkey");
    word.setCorrect(false);
    db.updateWord(word);
    dic_c.dictionaryOpen();
    for(Word w : dic_c.getDictionary().getWordArr()){
        if(w.getName()=="monkey")
        {
            fail("failed");
        }
    }
    word.setCorrect(true);
    db.updateWord(word);
    dic_c.dictionaryOpen();
    for(Word w : dic_c.getDictionary().getWordArr()){
        if(w.getName()=="monkey"){
            assertEquals("monkey",w.getName());
        }
    }
}
```

---

# 5 Demonstration

Demo

---



Demo



**감사합니다.**

[2015SMA\_T3]

---

---