

SOFTWARE

2015.06.02

VERIFICATION

TEAM 4

[2nd Testing – System Test]

컴퓨터 공학부

200911393 박현규

201010768 최정한

201111339 김민우

Presenter → 201211389 함진아

CONTENTS

●-----	PART 1	Pairwise Testing -----	3
●-----	PART 2	Review -----	10
●-----	PART 3	Static Analysis -----	28
●-----	PART 4	Summary -----	61

PART 1

Pairwise Testing

PART 1 Pairwise Testing

Tool

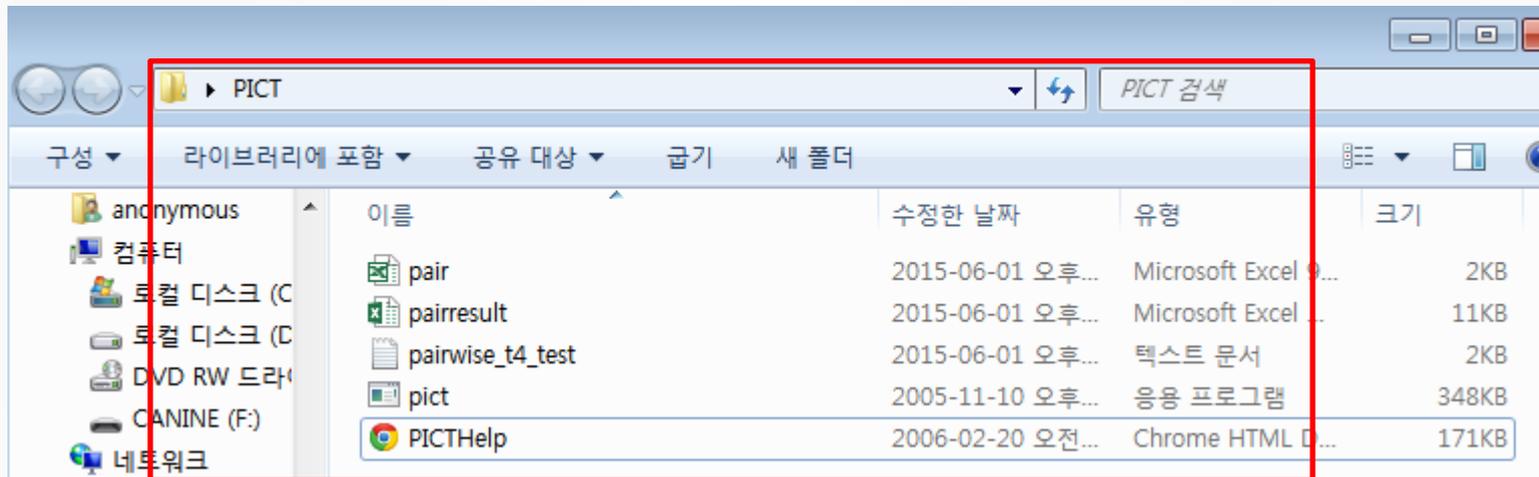
Available Tools

1.	CATS (Constrained Array Test System) *)	[Sherwood] Bell Labs.	
2.	OATS (Orthogonal Array Test System) *)	[Phadke] AT&T	
3.	AETG	Telecordia	Web-based, commercial
4.	IPO (PairTest) *)	[Tai/Lei]	
5.	TConfig	[Williams]	Java-applet
6.	TCG (Test Case Generator) *)	NASA	
7.	AllPairs	Satisfice	Perl script, free, GPL
8.	Pro-Test	SigmaZone	GUI, commercial
9.	CTS (Combinatorial Test Services)	IBM	Free for non-commercial use
10.	Jenny	[Jenkins]	Command-line, free, public-domain
11.	ReduceArray2	STSC, U.S. Air Force	Spreadsheet-based, free
12.	TestCover	Testcover.com	Web-based, commercial
13.	DDA *)	[Colburn/Cohen/Turban]	
14.	Test Vector Generator		GUI, free
15.	OA1	k sharp technology	
16.	CTE-XL	Berner & Mattner	GUI, free
17.	AllPairs	[McDowell]	Command-line, free
18.	Intelligent Test Case Handler (replaces CTS)	IBM	Free for non-commercial use
19.	CaseMaker	Díaz & Hilterscheid	GUI, commercial
20.	PICT	Microsoft Corp.	Command-line, free

PART 1 Pairwise Testing

How to Install PICT

[T4]Superman_Ver.2	2015-06-01 오후...	파일 폴더	
[T4]Superman_Ver.2	2015-06-01 오후...	압축(ZIP) 폴더	26,201KB
bitnami-redmine-3.0.3-0-windows-insta...	2015-05-26 오후...	응용 프로그램	170,051KB
pict33	2015-06-01 오후...	Windows Installer...	933KB



PART 1 Pairwise Testing

How to Use PICT

PICT는 cmd를 이용해서 해야함.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\anonymous>cd C:\Users\anonymous\Desktop\PICT

C:\Users\anonymous\Desktop\PICT>
  
```

```

2015-06-01 오후 08:53 <DIR>      ..
2015-06-01 오후 08:43           1,942 pair.xls
2015-06-01 오후 08:44          10,864 pairresult.xlsx
2015-06-01 오후 08:11           1,592 pairwise_t4_test.txt
2005-11-10 오후 11:04          356,352 pict.exe
2006-02-20 오전 11:12          174,480 PICTHelp.htm
          5개 파일              545,230 바이트
          2개 디렉터리      221,070,491,648 바이트 남음
  
```

```

C:\Users\anonymous\Desktop\PICT>pict pairwise_t4_test.txt > test.xls
Constraints Warning: All or no values satisfy relation [Parent-Button] in <"PBtn
SendMsg", "PBtnGoalSet", "PBtnChkGoal", "PBtnReset">
Constraints Warning: All or no values satisfy relation [Child-Button] in <"CMP",
"CMG", "CMD">
Constraints Warning: All or no values satisfy relation [Child-Practice-Button] i
n <"CPBtn1", "CPBtn2", "CPBtn3", "CPBtnSound", "CPBtnNext">
Constraints Warning: All or no values satisfy relation [Child-Game-Button] in <"
CGBtnSound", "CGBtnNext", "CGBtnExit", "CGWordEng">
Constraints Warning: All or no values satisfy relation [Child-Dic-Button] in <"C
DBtnSound", "CDBtnExit", "CDWordEng">
Constraints Warning: All or no values satisfy relation [Parent-Text] in <"PGoali
nt", "PRewardint">
  
```

pict [pictfilename] > [resultfilename]

*[resultfilename] option

1. .txt == 텍스트
2. .xls == 엑셀

PART 1 Pairwise Testing

Category Partitioning

Category	Kind	# Value	Error Constraints	Single Constraints	Property Constraints	
Account	Setting	100 Account Parent Password			[Property AP]	
		101 Account Child Name			[Property AC]	
		102 Account Parent Child Name			[Property APC]	
		103 Account Null	[Error]			
Mode Select	Mode-Button	200 Parent Mode			[if AP] [if APC] [Property PM]	
		201 Child Mode			[if AC] [if APC] [Property CM]	
Input-Button	Parent-Button	310 Parent.btnSendMsg			[if PM]	
		311 Parent.btnGoalSetting			[if PM] [Property PMG]	
		312 Parent.btnCheckGoal			[if PM]	
		313 Parent.btnReset			[if PM]	
	Child-Button	320 Child.btnPractice			[if CM] [Property CMP]	
		321 Child.btnGame			[if CM] [Property CMG]	
		322 Child.btnDictionary			[if CM] [Property CMD]	
	Child-Practice-Button	330 Child.Practice.btnWord[1]			[Single]	[if CMP]
		331 Child.Practice.btnWord[2]				[if CMP]
		332 Child.Practice.btnWord[3]				[if CMP]
		333 Child.Practice.btnSound			[Single]	[if CMP]
	Child-Game-Button	334 Child.Practice.btnNext			[Single]	[if CMP]
		340 Child.Game.btnSound				[if CMG]
		341 Child.Game.btnNext				[if CMG]
	Child-Dic-Button	342 Child.Game.btnExit				[if CMG]
350 Child.Dic.btnSound					[if CMD]	
351 Child.Dic.btnExit					[if CMD]	
Input-Text	Parent-Text	410 Parent.txtGoal_Null	[Error]		[if PMG]	
		411 Parent.txtGoal_Integer			[if PMG]	
		412 Parent.txtReward_Null	[Error]		[if PMG]	
	Child-Game-Text	413 Parent.txtReward_Integer				[if PMG]
		420 Child.Game.txtWord_Eng				[if CMG]
		421 Child.Game.txtWord_!Eng	[Error]			[if CMG]
	Child-Dic-Text	422 Child.Game.txtWord_Any	[Error]			[if CMG]
		430 Child.Dic.txtWord_Eng				[if CMD]
		431 Child.Dic.txtWord_!Eng	[Error]		[if CMD]	
		432 Child.Dic.txtWord_Any	[Error]		[if CMD]	
		Total	103680	4327	2102	

PART 1 Pairwise Testing

Apply Constraint

PICT와 partition category를
참고하여 적용한 constraint.

```

pairwise_t4_test - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Setting : AP, AC, APC
Mode-Button : PM, CM
Parent-Button : PBtnSendMsg, PBtnGoalSet, PBtnChkGoal, PBtnReset
Child-Button : CMP, CMG, CMD
Child-Practice-Button : CPBtn1, CPBtn2, CPBtn3, CPBtnSound, CPBtnNext
Child-Game-Button : CGBtnSound, CGBtnNext, CGBtnExit
Child-Dic-Button : CDBtnSound, CDBtnExit
Parent-Text : PGoalint, PRewardint
Child-Game-Text : CGWordEng
Child-Dic-Text : CDWordEng

IF [Setting] = "AP" THEN [Mode-Button] = "PM";
IF [Setting] = "APC" AND [Setting] = "AP" THEN [Mode-Button] = "PM";
IF [Setting] = "AC" THEN [Mode-Button] = "CM";
IF [Setting] = "APC" AND [Setting] = "AC" THEN [Mode-Button] = "CM";
IF [Mode-Button] = "PM" THEN [Parent-Button] in {"PBtnSendMsg", "PBtnGoalSet", "PBtnChkGoal", "PBtnReset"};
IF [Mode-Button] = "CM" THEN [Child-Button] in {"CMP", "CMG", "CMD"};
IF [Child-Button] = "CMP" THEN [Child-Practice-Button] in {"CPBtn1", "CPBtn2", "CPBtn3", "CPBtnSound", "CPBtnNext"};
IF [Child-Button] = "CMG" THEN [Child-Game-Button] in {"CGBtnSound", "CGBtnNext", "CGBtnExit", "CGWordEng"};
IF [Child-Button] = "CMD" THEN [Child-Dic-Button] in {"CDBtnSound", "CDBtnExit", "CDWordEng"};
IF [Parent-Button] = "PBtnGoalSet" THEN [Parent-Text] in {"PGoalint", "PRewardint"};
  
```

Category.

Constraint 적용.

PART 1 Pairwise Testing

PICT result

	A	B	C	D	E	F
1	Setting	Mode-Button	Parent-Button	Child-Button	Child-Practice-Button	Child-Game-
2	APC	CM	PBtnReset	CMD	CPBtnSound	CGBtnSounc
3	AP	PM	PBtnSendMsg	CMG	CPBtn3	CGBtnNext
4	AP	PM	PBtnReset	CMP	CPBtn2	CGBtnExit
5	AC	CM	PBtnChkGoal	CMG	CPBtn2	CGBtnExit
6	APC	PM	PBtnChkGoal	CMP	CPBtnNext	CGBtnSounc
7	AC	CM	PBtnSendMsg	CMP	CPBtnSound	CGBtnNext
8	AP	PM	PBtnReset	CMD	CPBtn3	CGBtnSounc
9	AC	CM	PBtnGoalSet	CMD	CPBtn1	CGBtnNext
10	APC	CM	PBtnSendMsg	CMD	CPBtnNext	CGBtnExit
11	APC	PM	PBtnGoalSet	CMG	CPBtnSound	CGBtnExit
12	APC	CM	PBtnChkGoal	CMD	CPBtn3	CGBtnNext
13	AC	CM	PBtnSendMsg	CMG	CPBtn2	CGBtnSounc

PART 2

Review

PART 2 Review

T4 : Specification Review

- 각 단계(Stage 1000, 2030, 2040)별 Spec 문서가 명확하지 않음.
 - 불분명한 표현이 사용
 - Stage마다 서술이 다름
 - 용어의 및 표현의 일관성 부족

PART 2 Review

T4 : Specification Review

- 불분명한 표현
 - Stage 1000

Use Case name	Print Message
Actor	Child
Description	Parent로부터 받은 message를 백그라운드에 출력



Use Case name	Print Message
Actor	Child
Description	Parent로부터 받은 Message를 GUI에 출력

- 백그라운드에 출력

Use Case name	Next Level
Actor	Child
Description	Child가 Wordtrain Game을 통해 얻은 점수가 일정 수준에 도달하면 다음 Level로 이동



Use Case name	Next Level
Actor	Child
Description	Exp가 50점의 배수일 때마다 Level이 증가

- 일정 수준

PART 2 Review

T4 : Specification Review

- 불분명한 표현
 - Stage 2040

Use Case	Parent Mode	
Actor	User	
Purpose	Parent Mode로 들어간다.	
Overview	Parent Mode로 들어가 프로그램에 관한 것을 설정한다.	
Type	Primary and Real	
Cross References	System functions: R1.1, R1.3 Use Case: Parent Mode, Account	Parent Mode
Pre-Requisites	N/A	User
Typical Courses of Events	(A): User / (S): System 1. (A) btnParentMode 클릭 2. (S) Parent.txt의 내용을 temp에 저장 3. (S) checkParentAccount()==true 확인 4. (S) checkPassword(txtPassword.getText(),temp)==true 확인 5. (S) Parent Mode GUI 생성	칸다. 가 프로그램에 관한 것을 설정한다.
Alternative Courses of Events	3. false면 return 후 account()	Primary
Exceptional Course of Events	4. 틀렸다면 상태유지	R1.1, R1.3
Typical Courses of Events		N/A
Alternative Courses of Events		stem le를 요청
Exceptional Course of Events		sword 설정 여부 검사
Typical Courses of Events		3. (S) Password를 입력하는 창을 띄움 4. (P) Password를 입력 후 확인 버튼을 누름 5. (S) Parent Mode로 입장
Alternative Courses of Events		2. 등록되어 있지 않다면 바로 Account로 이동
Exceptional Course of Events		4. 틀리면 입력한 Password가 옳지 않다는 Message 출력

- '상태유지'

PART 2 Review

T4 : Specification Review

- Stage 마다 다른 정의
 - Stage 2030 -> Stage 2040

Use Case	Goal Setting	Goal Setting
Actor	Parent	Parent
Purpose	Child에게 일정 목표 및 보상을 설정한다.	Child에게 일정 Exp값 및 Reward를 설정한다.
Overview	Child가 Wordtrain Game을 통해 얻어야 할 일정 Exp를 설정하고 이에 따른 보상을 설정한다.	Child가 Wordtrain Game을 통해 얻어야 할 일정 Exp를 설정하고 이에 따른 Reward를 설정한다.
Type	Primary	Primary
Cross References	R2.2	R2.2
Pre-Requisites	Parent Login을 성공해야 한다.	Parent Login을 성공
Typical Courses of Events	(P): Parent / (S): System 1. (P) Goal Setting을 요청 2. (S) 이미 설정한 Goal을 보여줌 3. (S) 새로운 Goal을 설정할 것인지 의사를 물어보는 이벤트 창 출력 4. (P) Goal로 설정할 Level 혹은 Exp와, 그에 따른 보상 설정	(P): Parent / (S): System 1. (P) Goal Setting을 요청 2. (S) 현재 Goal.txt의 내용을 보여줌 3. (P) Goal로 설정할 Exp와 그에 따른 Reward 설정
Alternative Courses of Events	N/A	N/A
Exceptional Course of Events	N/A	N/A

- 의사를 물어보는 이벤트창

Use Case	Goal Setting
Actor	Parent
Purpose	Child에게 일정 목표 및 보상 설정
Overview	Child가 Wordtrain Game을 통해 얻어야 할 일정 Exp를 설정하고 이에 따른 보상 설정
Type	Primary and Real
Cross References	System function: R2.2 Use case: Goal Setting
Pre-Requisites	Parent 객체 생성
Typical Courses of Events	(P): Parent / (S): System 1. (P) btnGoalSetting 클릭 2. (P) txtGoal, txtReward에 Goal, Reward 입력 3. (S) Superman.setGoal(txtGoal.getText())로 재설정 4. (S) Superman.setReeward(txtReward.getText())로 재설정 5. (S) Superman.txt에 Superman.goal, Superman.reward 저장
Alternative Courses of Events	N/A
Exceptional Course of Events	N/A

PART 2 Review

T4 : Specification Review

- Stage 마다 다른 정의
 - Stage 2030 -> Stage 2040

Use Case	Reset	Reset
Actor	Parent	Parent
Purpose	모든 정보를 초기화한다.	모든 정보를 초기화한다.
Overview	Parent를 포함한 Child의 모든 정보 초기화한다.	Parent를 포함한 Child의 모든 정보 초기화한다.
Type	Primary	Primary
Cross References	R2.4	R2.4
Pre-Requisites	Parent Login을 성공해야 한다.	Parent Login을 성공
Typical Courses of Events	(P): Parent / (S): System 1. (P) Reset 요청 2. (S) 실행 할 것인지 확인버튼 출력 3. (P) 확인버튼 클릭 4. (S) 모든 정보 초기화	(P): Parent / (S): System 1. (P) Reset 버튼 클릭 2. (S) 모든 정보 초기화
Alternative Courses of Events	3. 취소하면 Parent Mode로 돌아감	N/A
Exceptional Course of Events	N/A	N/A

- 확인버튼 출력

Use Case	Reset
Actor	Parent
Purpose	모든 정보를 초기화
Overview	Parent를 포함한 Child의 모든 정보 초기화
Type	Primary and Real
Cross References	System function: R2.4 Use Case: Reset
Pre-Requisites	Parent, Superman, Child 객체 생성
Typical Courses of Events	(P): Parent / (S): System 1. (P) btnReset 클릭 2. (S) Parent.reset() 실행 3. (S) Parent.txt, Goal.txt, Exp.txt, Message.txt, Reward.txt, Child.txt 초기화
Alternative Courses of Events	N/A
Exceptional Course of Events	N/A

PART 2 Review

T6 : Specification Review

- 서술 부족 및 불분명한 표현

Use Case 정의가 WordButton으로 되어있다.

2. AlphaButton	
Use Case	AlphaButton
Actor	User
Purpose	알파벳 검색 메뉴를 선택한다.
Overview	사용자가 영어 알파벳 검색 메뉴를 선택하면, 영어 알파벳 검색 화면을 출력한다.
Type	Primary
Cross Reference	R2
Pre-Requisites	프로그램 실행 후 초기화면
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : 사용자가 알파벳 검색 메뉴를 클릭한다. 2. (S) : 영어 알파벳 검색 화면을 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

→ stage 2030 v4에서 AlphaButton으로 수정 완료
 이후 작성되지 않은 specification 부분은 수정을 모두 확인하였음. 특이사항이 존재하는 경우만 작성

PART 2 Review

T6 : Specification Review

- 서술 부족 및 불분명한 표현

regist에서의 '게임을 완료한 상태'

11. regist	
Use Case	regist
Actor	User
Purpose	사용자의 기록을 랭킹에 등록한다.
Overview	사용자가 입력한 이름을 랭킹에 등록한다.
Type	Primary
Cross Reference	R4.2.3
Pre-Requisites	날말 퀴즈 게임을 완료한 상태
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : 사용자가 확인 버튼을 클릭한다. 2. (S) : Game 의 regist 를 호출한다. 3. (S) : Ranking 의 addRank 를 호출하여 score 를 기준으로 내림차순으로 정렬하여 랭킹 파일에 저장한다. 4. (S) : Controller 의 returnToMenu 를 호출한다. 5. (S) : MainScreen 패널로 전환한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

→ 게임을 완료한 상태가 게임을 다 맞춰야 완료인지, 아니면 하다 말고 종료를 눌러도 완료라고 인식하는지에 대한 의미가 불분명함.

PART 2 Review

T6 : Specification Review

- Stage 간 내용 차이 존재

2030, 2040 단계에는 랭킹에 관한 서술이 없음.

문제	정의되지 기능에 대한 요구사항 필요
원인	1000번대에 정의하지 않은 기능이 추가됨 Regist Use case가 1000번대 version3 문서를 잘못 제출해서 추가 안되어 있었음.
대응	1000번대 문서 새로운 version 제출

→ 불필요하다고 판단되어 1000에서 삭제한 것 확인. 대신, Regist 형태로 랭킹에 관한 서술이 2030, 2040에 나타나고 있으며 이에 대한 내용이 1000에 추가된 수정이 확인됨.

PART 2 Review

T6 : Specification Review

- Stage 간 내용 차이 존재

Time에 대해 1000번대에 정의한 기능과 2030에서 정의한 기능이 다름.

Use Case	13. Time
Actor	System
Description	날말 퀴즈 게임에 제한 시간을 설정해주고, 남은 시간을 출력한다. 제한시간을 초과하면 실패로 돌아가게 한다.

→ 제한 시간과 남은 시간에 대한 명시 추가 확인 완료.

13. Time

Use Case	Time
Actor	System
Purpose	게임의 남은 시간을 표시한다.
Overview	타이머에 따라 남은 시간이 갱신되어 표시된다.
Type	Primary
Cross Reference	R4.2.5
Pre-Requisites	날말 퀴즈 게임이 진행중인 상태.
Typical Courses of Events	(S) : System 1. (S) : 날말 퀴즈 게임의 제한 시간을 5분으로 설정한다. 2. (S) : 게임의 남은 시간을 계산한다. 3. (S) : 남은 시간을 화면에 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. 남은 시간이 0일 경우 게임을 종료하고 날말 퀴즈 게임의 주제선택 메뉴로 돌아간다

→ 그러나 1000번대에서의 '실패로 돌아가게 한다'가 2030번대에서의 '게임이 종료되고 주제선택 메뉴로 돌아간다'는 것인지에 대한 정의가 불분명함.

PART 2 Review

T6 : Specification Review

- 오/탈자

Stage 1000

Use Case Name	2. AlphaBotton
Actors	User
Description	영어 알파벳 검색 기능을 선택한다.

Use Case Name	4. WordButton
Actors	User
Description	영어 단어 검색 기능을 선택한다.

Use Case Name	6. GameBotton
Actors	User
Description	낱말 퀴즈 게임 기능을 선택한다.

→ Button이 아닌 Botton으로 오탈자 발견

PART 2 Review

T4 : Category Partition Testing Review

103	아무것도 설정하지 않는다.					Pass
1						
100.200.310	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, SEND 메시지버튼을 누른다.					Pass
100.200.312	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, 체크 골 버튼을 누른다.					Pass
100.200.313	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, 리셋 버튼을 누른다.					Pass
100.200.311	부모님 비밀번호 설정위, 부모님 모드 버튼을 누른다음, 골 세팅 버튼을 누른다.					Pass
100.200.311.410	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, 골 세팅 버튼을 누른다음, 골에 아무입력도 하지 않는다.				버튼을 누른 뒤, apple을 입력한 뒤, Sound 버튼을 누른다.	Pass
100.200.311.411	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, 골 세팅 버튼을 누른다음, 골에 100을 입력한다.				버튼을 누른 뒤, apple을 입력한 뒤, Exit 버튼을 누른다.	Pass
100.200.311.412	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, 골 세팅 버튼을 누른다음, 보상에 아무입력도 하지 않는다.				버튼을 누른 뒤, 사과를 입력한 뒤, Sound 버튼을 누른다.	Pass
100.200.311.413	부모님 비밀번호 설정위, 부모님 모드 버튼 누른다음, 골 세팅 버튼을 누른다음, 보상에 1000을 입력한다.				버튼을 누른 뒤, 사과를 입력한 뒤, Exit 버튼을 누른다.	Pass
102.200.310	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, SEND 메시지버튼을 누른다.				버튼을 누른 뒤, 456asdfl를 입력한 뒤, Sound 버튼을 누른다.	Pass
102.200.312	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, Check Goal 버튼을 누른다.				버튼을 누른 뒤, 456asdfl를 입력한 뒤, Exit 버튼을 누른다.	Pass
102.200.313	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, Reset 버튼을 누른다.				, 사전 버튼을 누른 뒤, apple을 입력한 뒤, Sound 버튼을 누른다.	Pass
102.200.311	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, Goal 세팅 버튼을 누른다.				, 사전 버튼을 누른 뒤, apple을 입력한 뒤, Exit 버튼을 누른다.	Pass
102.200.311.410	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, Goal 세팅 버튼을 누른뒤, 아무것도 입력하지 않는다.				, 사전 버튼을 누른 뒤, 사과를 입력한 뒤, Sound 버튼을 누른다.	Pass
102.200.311.411	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, Goal 세팅 버튼을 누른뒤, 100을 입력한다.				, 사전 버튼을 누른 뒤, 사과를 입력한 뒤, Exit 버튼을 누른다.	Pass
102.200.311.412	부모와 자식의 이름을 설정위, 부모 모드 버튼을 누른다음, Goal 세팅 버튼을 누른뒤, 아무것도 입력하지 않는다.				, 사전 버튼을 누른 뒤, 456asdfl를 입력한 뒤, Sound 버튼을 누른다.	Pass
102.201.320.333	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 연습 버튼을 누른 뒤, 사운드 버튼을 누른다.				, 사전 버튼을 누른 뒤, 456asdfl를 입력한 뒤, Exit 버튼을 누른다.	Pass
102.201.320.334	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 연습 버튼을 누른 뒤, Next 버튼을 누른다.					Pass
6						
101.201.321.420.340	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 apple을 입력하고, 사운드 버튼을 누른다.				버튼을 누른뒤, txtWord에 사과를 입력하고, 사운드 버튼을 누른다.	Pass
101.201.321.420.341	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 apple을 입력하고, Next 버튼을 누른다.				버튼을 누른뒤, txtWord에 사과를 입력하고, Next 버튼을 누른다.	Pass
101.201.321.420.342	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 apple을 입력하고, Exit 버튼을 누른다.				버튼을 누른뒤, txtWord에 456asdfl를 입력하고, Sound 버튼을 누른다.	Pass
101.201.321.421.340	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 사과를 입력하고, 사운드 버튼을 누른다.				버튼을 누른뒤, txtWord에 456asdfl를 입력하고, Next 버튼을 누른다.	Pass
101.201.321.421.341	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 사과를 입력하고, Next 버튼을 누른다.				음, 게임 버튼을 누른 뒤, 사과를 입력한뒤, Sound 버튼을 누른다.	Pass
101.201.321.421.342	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 사과를 입력하고, Exit 버튼을 누른다.				음, 게임 버튼을 누른 뒤, 사과를 입력한뒤, Next 버튼을 누른다.	Pass
101.201.321.422.340	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 456asdfl를 입력하고, Sound 버튼을 누른다.				음, 게임 버튼을 누른 뒤, 456asdfl를 입력한뒤, Sound 버튼을 누른다.	Pass
101.201.321.422.341	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 456asdfl를 입력하고, Next 버튼을 누른다.				버튼을 누른 뒤, 사과를 입력한 뒤, Sound 버튼을 누른다.	Pass
101.201.321.422.342	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른뒤, txtWord에 456asdfl를 입력하고, Exit 버튼을 누른다.				버튼을 누른 뒤, 456asdfl를 입력한 뒤, Sound 버튼을 누른다.	Pass
102.201.321.420.340	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, apple을 입력한뒤, Sound 버튼을 누른다.				음, 사전 버튼을 누른 뒤, 사과를 입력한 뒤, Sound 버튼을 누른다.	Pass
102.201.321.420.341	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, apple을 입력한뒤, Next 버튼을 누른다.				이 사전 버튼을 누른 뒤, 456asdfl를 입력한 뒤, Sound 버튼을 누른다.	Pass
102.201.321.420.342	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, apple을 입력한뒤, Exit 버튼을 누른다.					Pass
102.201.321.421.340	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, 사과를 입력한뒤, Sound 버튼을 누른다.					Pass
102.201.321.421.341	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, 사과를 입력한뒤, Next 버튼을 누른다.					Pass
102.201.321.421.342	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, 사과를 입력한뒤, Exit 버튼을 누른다.					Pass
102.201.321.422.340	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, 456asdfl를 입력한뒤, Sound 버튼을 누른다.					Pass
102.201.321.422.341	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, 456asdfl를 입력한뒤, Next 버튼을 누른다.					Pass
102.201.321.422.342	부모와 자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 게임 버튼을 누른 뒤, 456asdfl를 입력한뒤, Exit 버튼을 누른다.					Pass
18						
101.201.322.430.350	자식의 이름을 설정위, 자식 모드 버튼을 누른다음, 사전 버튼을 누른 뒤, apple을 입력한 뒤, Sound 버튼을 누른다.					Pass

PART 2 Review

T6 : Category Partition Testing Review

1	100.200.210	알파벳 찾기를 누르고, A~Z까지 알파벳을 입력하고, 찾기 버튼을 누른다.			Pass
2	100.200.211	알파벳 찾기를 누르고, A~Z까지 알파벳을 입력하고, 초기 메뉴로 버튼을 누른다.			Pass
3	100.201.210	알파벳 찾기를 누르고, 1을 입력하고, 찾기 버튼을 누른다.			Pass
4	100.201.211	알파벳 찾기를 누르고, 1을 입력하고, 초기 메뉴로 버튼을 누른다.			Pass
5	101.300.310	단어 검색을 누르고, 데이터베이스에 있는 영어 단어를 입력하고, 찾기 버튼을 누른다.			Pass
6	101.300.311	단어 검색을 누르고, 데이터베이스에 있는 영어 단어를 입력하고, 초기 메뉴로 버튼을 누른다.			Pass
7	101.301.310	단어 검색을 누르고, 데이터베이스에 없는 영어 단어를 입력하고, 찾기 버튼을 누른다.			Pass
8	101.301.311	단어 검색을 누르고, 데이터베이스에 없는 영어 단어를 입력하고, 초기 메뉴로 버튼을 누른다.			Pass
9	101.302.310	단어 검색을 누르고, 123을 입력하고, 찾기 버튼을 누른다.			Pass
10	101.302.311	단어 검색을 누르고, 123을 입력하고, 초기 메뉴로 버튼을 누른다.			Pass
11	102.400.410	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누른다.			Pass
12	102.400.410.420.430	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하고, 힌트 보기 버튼을 누른다.			Pass
13	102.400.410.420.431	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하고, 정답 체크 버튼을 누른다.			Fail
14	102.400.410.420.432	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하고, 게임 종료 버튼을 누른다.			Pass
15	102.400.410.420.433	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하고, 초기 메뉴로 버튼을 누른다.			Pass
16	102.400.410.421.430	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하지 않고, 힌트 보기 버튼을 누른다.			Pass
17	102.400.410.421.431	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하지 않고, 정답 체크 버튼을 누른다.			Fail
18	102.400.410.421.432	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하지 않고, 게임 종료 버튼을 누른다.			Pass
19	102.400.410.421.433	게임 하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 답 문자열을 입력하지 않고, 초기 메뉴로 버튼을 누른다.			Pass
20	102.401.410	게임 하기를 누르고, 주제 선택을 하지 않고, 게임 시작 버튼을 누른다.			Pass
21	103.500	랭킹 보기를 누르고, 초기 메뉴로 버튼을 누른다.			Pass
22	104	종료를 누른다.			Pass

PART 2 Review

T4 : Brute Force Testing Report Review

#	테스트	기대
1	Account 계정 입력 없이 계속 버튼 클릭	하나만 뜸 (계정 1개라 했으니까)
2	계정 이미 만들어진 상태에서 Account 클릭	실행 안됨 (명세서에 account가 없을 때만 실행 된다고 명시됨)
3	goal에 숫자가 아닌 데이터 집어넣음	명세서에 안 써 있음
4	부모모드 계속 버튼 클릭	하나만 뜸
5	자식모드 계속 버튼 클릭	하나만 뜸
6	영어 단어 입력하고 바로 Sound 버튼 누르기	소리가 난다
7	child 모드가 떠 있는 상태에서 부모모드 창 띄워서 reset	게임 및 사전, 연습이 진행 안됨 (계정 없어졌으니까)
8	영어 단어 입력 란에 대문자 포함해서 영단어 입력	명세서에 안 써 있음
9	끝말잇기에서 단어 입력하고 next 누르기	새로운 단어 입력란 비었으면 좋겠음
10	끝말잇기에서 단어 맞추고 next 안 누르고 exit 누르기	맞춰오니 점수 반영 해달라 !!
11	부모모드에서 goal 설정 안하고 child 모드에서 끝말잇기 경험치 쌓고 부모모드에서 check goal	goal은 0이어도 경험치는 보여달라
12	부모모드에서 goal 설정하고 child 모드에서 끝말잇기 경험치 쌓고 부모모드에서 check goal	경험치 적용
13	like 단어 사전 검색	좋아하다 / ~와 같다 ... 라는 뜻 기대
14	메시지에 엄청 긴 값 집어넣음	들어간다
15	데이터베이스에 없는 단어 입력해보기	에러 메시지
16	꽤다 키면 Account 및 모든 정보 남아 있는지 확인	남아 있음
17	끝말잇기에 맞지 않는 단어 입력	에러 메시지
18	계정 설정 안하고 부모모드 / 자식모드 클릭	실행 안됨

실제 수행	테스트 결과	#
창이 계속 뜸	Pass	이제 계정에 부모와 자식을 모두 입력해야 Confirm가능
실행 됨	Pass	이제 계정이 있는 상태에서 다시 Account를 누르면, 새로운 계정을 만들거나, 아예 만들지 않는 경우로 바뀜
0으로 표시 / 에러 메시지 없음	Pass	숫자만 입력하도록 메세지창 뜨고 넘어가지 않음
계속 뜬다...	Pass	계속 클릭은 있을수 없다
계속 뜬다...	Pass	계속 클릭은 있을수 없다
안난다. enter를 쳐야만 인식한다 but 명세서에 그런 내용 없음	Pass	엔터 입력으로 문서를 고쳤다.
다 진행 잘 됨. child 모드를 다시 켜야만 reset이 적용 된다.	Pass	이제 한번에 한가지 모드로만 진행할수 있다.
대문자 인식 안됨. caps lock 켜있는 상태면 어떡함 ... ?	Pass	Apple이라고 해도 이제 통과된다. 섞어서 대문자를 넣어도 제대로 사진이 나옴
예전 단어 남아있음 ...	Pass	이제 예전단어는 남아있지 않다.
너 next를 누르지 않았으니 단어가 맞아도 점수 아중다 !!	Pass	점수가 오름다
child 모드에서는 경험치 올라가지만, 부모 모드에서는 반영 안됨	Fail	안보임.
바뀐 경험치 반영 안됨. 아예 프로그램 꺾다가 켜야만 적용 됨	Fail	경치 안쌓임.
뜻은 ~와 로 뜨고 사진은 어떤 애가 나뭇잎 보면서 슬퍼함 ??? 단어에 맞지 않는 뜻 존재하는듯 !!!	Fail	사진이 슬퍼하고있음...
들어간다	Pass	
에러 메시지	Pass	
남아 있음	Pass	
에러 메시지	Pass	
실행 안됨	Pass	

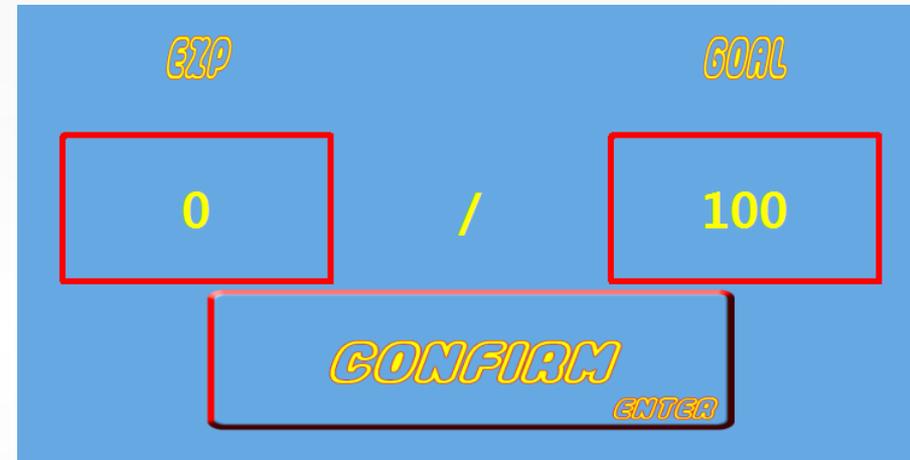
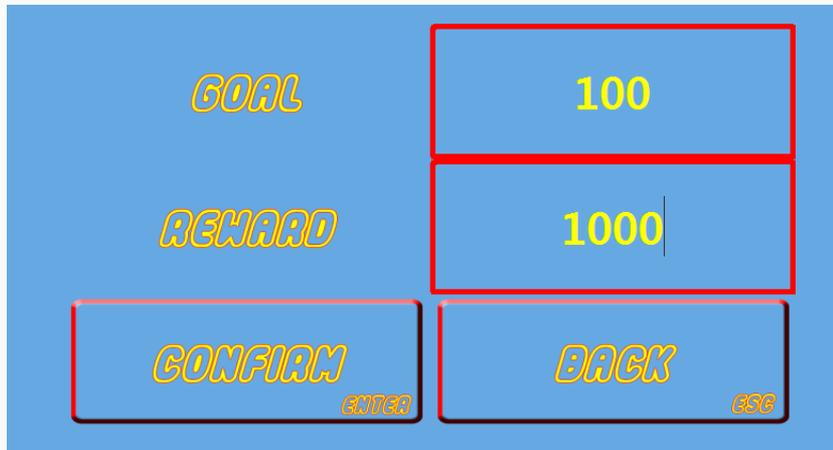
PART 2 Review

T4 : Brute Force Testing Report Review

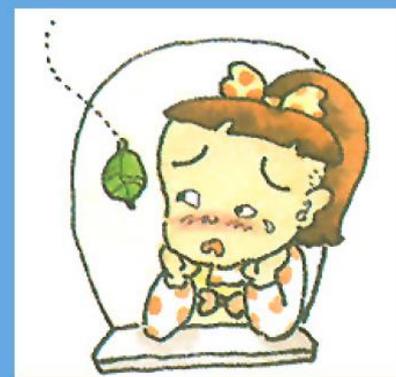
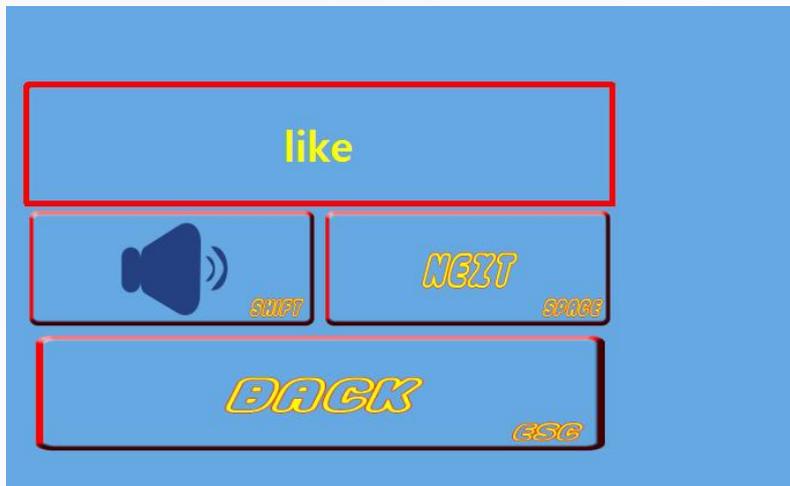


PART 2 Review

T4 : Brute Force Testing Report Review



#11, 12 : 여전히 경험치 반영이 되지 않는다는 점



좋아하다

#13 : 좋아하는 사진...?

PART 2 Review

T6 : Brute Force Testing Report Review

			기대
1	게임하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 알파벳 한 칸에 여러 알파벳을 입력한다.		한 칸에 하나의 알파벳만 입력 가능
2	게임하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 알파벳 한 칸에 맞는 알파벳과 공백을 넣는다.		한 칸에 하나의 알파벳만 입력 가능 or 공백을 넣어도 답으로 인식
3	게임하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 알파벳 한 칸에 맞는 정답을 넣는다.		답으로 인식하지 않는 것이 말이 될 것으로 생각
4	게임하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 알파벳 한 칸에 하나의 알파벳을 입력하고 방향키를 누른다.		다음 칸으로 이동
5	게임하기를 누르고, 주제 선택을 하고, 게임 시작 버튼을 누르고, 시간이 다 될 때까지 기다린다.		게임 끝
6	버전이 4개 존재. 버전 1에서는 정답 확인 기능이 제대로 작동함		버전 3_v07은 무엇... ?
7	정답을 맞추고 힌트 보기 버튼을 누른다.		사용자가 정답을 모를 경우 힌트 기능 을 사용해서 도움을 얻을 수 있다고 했 으니 답을 맞춘 경우에는 힌트가 필요 없다.
8	단어 검색을 누르고, 데이터베이스에 있는 영어 단어를 대문자를 포함해서 입력하고, 찾기 버튼을 누른다.		단어 찾아짐

버전이 여러 개 존재하는 문제로, 3_v07에서는 게임 시작 이후 정답 확인 버튼을 누르기만 하면 전부 랭킹 등록으로 연결되어 카테고리 파티셔닝에서 진행한 test 이외의 brute force test가 불가능함.

따라서 brute force testing report은 정답 확인 기능이 제대로 작동하는 1_v07을 기준으로 작성함. 버전 관리 및 버전 간의 기능 일치가 가장 먼저 해결해야 할 문제로 보임.

수행	
여러 개 입력 가능	Fail
틀린 답 처리	Fail
어떤 단어는 답으로 인식하지만, 어떤 단어는 같은 방식으로 답을 입력해도 답으로 처리하지 않음. 처리 방식이 일정하지 않아보임	Fail
다음 칸으로 이동하지 않고 이외의 칸으로 이동	Fail
게임 끝. 주제 선정 화면으로	Pass
버전 간에 기능이 일치하지 않는 모습이 보임	Fail
정답을 맞춰도 힌트를 준다. 문서 수정이나 프로그램 수정이 필요할 것으로 보인다.	Fail
단어 찾아짐	Pass

PART 2 Review

Redmine : 이슈 관리

✓ # ▾	유형	상태	우선순위	제목	담당자	변경
<input type="checkbox"/> 54	결함	신규	보통	[13] Brute force testing fail 대응		2015/05/26 11:10
<input type="checkbox"/> 53	결함	신규	보통	[12] Brute force testing fail 대응		2015/05/26 11:09
<input type="checkbox"/> 52	결함	신규	보통	[11] Brute force testing fail 대응		2015/05/26 11:08
<input type="checkbox"/> 51	결함	신규	보통	[10] Brute force testing fail 대응		2015/05/26 11:07
<input type="checkbox"/> 50	결함	신규	보통	[9] Brute force testing fail 대응		2015/05/26 11:06
<input type="checkbox"/> 49	결함	신규	보통	[8] Brute force testing fail 대응		2015/05/26 11:03
<input type="checkbox"/> 48	결함	신규	보통	[7] Brute force testing fail 대응		2015/05/26 11:02
<input type="checkbox"/> 47	결함	신규	보통	[6] Brute force testing fail 대응		2015/05/26 11:02
<input type="checkbox"/> 46	결함	신규	보통	[5] Brute force testing fail 대응		2015/05/26 11:00
<input type="checkbox"/> 45	결함	신규	보통	[4] Brute force testing fail 대응		2015/05/26 11:00
<input type="checkbox"/> 44	결함	신규	보통	[3] Brute force testing fail 대응		2015/05/26 10:57
<input type="checkbox"/> 43	결함	신규	보통	[2] Brute force testing fail 대응		2015/05/26 10:57
<input type="checkbox"/> 42	결함	신규	보통	[1] Brute force testing fail 대응		2015/05/26 10:59
<input type="checkbox"/> 41	결함	신규	보통	[102.201.322.432.350] test case fail 대응		2015/05/26 10:55
<input type="checkbox"/> 40	결함	신규	보통	[102.201.322.431.350] test case fail 대응		2015/05/26 10:53
<input type="checkbox"/> 39	결함	신규	보통	[101.201.322.432.350] test case fail 대응		2015/05/26 10:52
<input type="checkbox"/> 38	결함	신규	보통	[101.201.322.431.350] test case fail 대응		2015/05/26 10:51
<input type="checkbox"/> 37	결함	신규	보통	[102.201.321.422.341] test case fail 대응		2015/05/26 10:50
<input type="checkbox"/> 36	결함	신규	보통	[102.201.321.422.340] test case fail 대응		2015/05/26 10:48
<input type="checkbox"/> 35	결함	신규	보통	[102.201.321.421.341] test case fail 대응		2015/05/26 10:46
<input type="checkbox"/> 34	결함	신규	보통	[102.201.321.421.340] test case fail 대응		2015/05/26 10:44
<input type="checkbox"/> 33	결함	신규	보통	[101.201.321.422.341] test case fail 대응		2015/05/26 10:43
<input type="checkbox"/> 32	결함	신규	보통	[101.201.321.422.340] test case fail 대응		2015/05/26 10:42
<input type="checkbox"/> 31	결함	신규	보통	[101.201.321.421.341] test case fail 대응	준한 박	2015/05/26 10:45
<input type="checkbox"/> 30	결함	신규	보통	[101.201.321.421.340] test case fail 대응	준한 박	2015/05/26 10:44

PART 3

Static Analysis

PART 3 Static Analysis

Static Analysis Tool

- Checkstyle
- PMD
- findbugs
- Jdepend

모든 report는 xml 및 html 형태로 추출

Jenkins와의 연동을 통해, CI 서버에서도 dashboard 형태로 report 확인 가능

PART 3 Static Analysis

Checkstyle

- CheckStyle은 코딩 표준에 맞게 소스코드를 작성하도록 도와주는 도구이다.
- 개발자들이 잘 지키지는 않지만 중요한 것들을 지적해준다 (ex : private 설정)
- 협업시에 코딩스타일을 맞출 수 있고, 잠재적인 결함 발견도 가능하다. 팀의 코딩규칙을 문서로 정리해서 지키기로 하면, 규칙이 많아질 수록 어기기 쉽다.

- Checkstyle에서 기본적으로 제공하는 ruleset을 사용할 경우
- 14개의 카테고리 및 149개의 rule
 - 1667개의 warning 발생

PART 3 Static Analysis

Checkstyle

- Checkstyle에서 제공하는 ruleset의 일부를 활용
- 4개의 카테고리 및 40개의 rule
 - 141개의 warning 발생

Category	Total	Distribution
Annotation	1	■
Metrics	21	■
Naming	43	■
Whitespace	1602	■
Total	1667	

PART 3 Static Analysis

Checkstyle : Annotation

Checkstyle Warnings - Category Annotation

Summary

Total	High Priority	Normal Priority	Low Priority
1	0	1	0

Details

Details

[ControllerTest.java:25](#), AnnotationLocationCheck, Priority: Normal

Annotation 'ignore' have incorrect indentation level 4, expected level should be 1.

Check location of annotation on language elements. By default, Check enforce to locate annotations immediately after documentation block and before target element, annotation should be located on separate line from target element.

Example:

```
@Override @Nullable public String getNamelfPresent() { ... }
```

PART 3 Static Analysis

Checkstyle : Annotation

- ControllerTest.java:25 - 들여쓰기에 대한 지적

```
24  @Test
25  @Ignore("Not yet")
26  public void testChildMode() throws FileNotFoundException, URISyntaxException {
27      Controller cnt = new Controller();
28      cnt.childMode();
29      assertEquals("name", cnt.getTemp());
30      assertEquals(false, cnt.checkChildAccount("name"));
31      assertEquals(true, cnt.checkChildAccount("이□□빌□얏"));
32  }
--
```

PART 3 Static Analysis

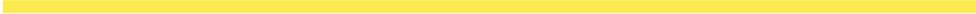
Checkstyle : Metrics

Checkstyle Warnings - Category Metrics

Summary

Total	High Priority	Normal Priority	Low Priority
21	0	<u>21</u>	0

Details

Folders	Files	Types	Warnings	Details
		Type	Total	Distribution
		ClassDataAbstractionCouplingCheck	11	
		ClassFanOutComplexityCheck	2	
		JavaNCSSCheck	8	
		Total	21	

PART 3 Static Analysis

Checkstyle : Metrics

- ClassDataAbstractionCouplingCheck
 - 클래스에서 사용된 레퍼런스 타입의 수를 계산한다.
 - 최대 7개까지가 적정선.

[Controller.java:15](#), ClassDataAbstractionCouplingCheck, Priority: Normal

Class Data Abstraction Coupling is 8 (max allowed is 7) classes [BufferedReader, BufferedWriter, FileReader, FileWriter, GUI_Account, GUI_Check_Password, GUI_Child_Mode, Superman].

This metric measures the number of instantiations of other classes within the given class. This type of coupling is not caused by inheritance or the object oriented paradigm. Generally speaking, any abstract data type with other abstract data types as members has data abstraction coupling; therefore, if a class has a local variable that is an instantiation (object) of another class, there is data abstraction coupling. The higher the DAC, the more complex the data structure (classes) of the system.

[GUI_Wordtrain_Game.java:37](#), ClassDataAbstractionCouplingCheck, Priority: Normal

Class Data Abstraction Coupling is 16 (max allowed is 7) classes [BufferedReader, EmptyBorder, File, FileWriter, Font, GUI_Child_Mode, ImageIcon, JButton, JFrame, JLabel, JPanel, JTextField, Random, Superman, Voca, Wordtrain].

This metric measures the number of instantiations of other classes within the given class. This type of coupling is not caused by inheritance or the object oriented paradigm. Generally speaking, any abstract data type with other abstract data types as members has data abstraction coupling; therefore, if a class has a local variable that is an instantiation (object) of another class, there is data abstraction coupling. The higher the DAC, the more complex the data structure (classes) of the system.

PART 3 Static Analysis

Checkstyle : Metrics

- ClassDataAbstractionCouplingCheck
 - Controller.java:15 – 8개의 참조 발견

```
015 public class Controller
```

```
016 {  
017 //Variables  
018     private Parent parent;
```

```
    BufferedReader in = new BufferedReader(new FileReader(getClass().getClassLoader().getResource  
    rent.txt").getPath().replace("%5c", "\\"));  
    temp=in.readLine();  
    if(checkParentAccount(temp)==false)  
    {  
        new GUI_Account(this);  
    }  
    else  
    {  
        new GUI_Check_Password(this,this.superman);  
    }  
}
```

PART 3 Static Analysis

Checkstyle : Metrics

- ClassFanOutComplexityCheck
 - 참조하여 사용하는 class의 개수
 - 최대 20개까지가 적정선.

Files Warnings Details

[GUI_Wordtrain_Game.java:37](#), ClassFanOutComplexityCheck, Priority: Normal

Class Fan-Out Complexity is 24 (max allowed is 20).

The number of other classes a given class relies on. Also the square of this has been shown to indicate the amount of maintenance required in functional programs (on a file basis) at least.

[GUI_Wordtrain_Practice.java:31](#), ClassFanOutComplexityCheck, Priority: Normal

Class Fan-Out Complexity is 21 (max allowed is 20).

The number of other classes a given class relies on. Also the square of this has been shown to indicate the amount of maintenance required in functional programs (on a file basis) at least.

PART 3 Static Analysis

Checkstyle : Naming

Checkstyle Warnings - Category Naming

Summary

Total	High Priority	Normal Priority	Low Priority
43	0	43	0

Details

Folders	Files	Types	Warnings	Details																		
<table><thead><tr><th>Type</th><th>Total</th><th>Distribution</th></tr></thead><tbody><tr><td>LocalVariableNameCheck</td><td>1</td><td></td></tr><tr><td>MemberNameCheck</td><td>5</td><td></td></tr><tr><td>PackageNameCheck</td><td>25</td><td></td></tr><tr><td>TypeNameCheck</td><td>12</td><td></td></tr><tr><td>Total</td><td>43</td><td></td></tr></tbody></table>					Type	Total	Distribution	LocalVariableNameCheck	1		MemberNameCheck	5		PackageNameCheck	25		TypeNameCheck	12		Total	43	
Type	Total	Distribution																				
LocalVariableNameCheck	1																					
MemberNameCheck	5																					
PackageNameCheck	25																					
TypeNameCheck	12																					
Total	43																					

PART 3 Static Analysis

Checkstyle : Naming

- LocalVariableNameCheck

→ Local 변수의 이름이 제시하는 format에 맞는가 ?

→ Naming의 전반적인 내용이 언더바 사용에 대한 지적

[WordtrainTest.java:38](#), LocalVariableNameCheck, Priority: Normal

Name 'voca_next' must match pattern `^[a-z][a-zA-Z0-9]*$`.

No description available. Please upgrade to latest checkstyle version.

```
public void testPrintWordList() throws IOException, URISyntaxException {  
    Wordtrain word = new Wordtrain();  
    Voca voca[] = new Voca[3];  
    Voca voca_next[] = new Voca[3];  
}
```

PART 3 Static Analysis

Checkstyle : Whitespace

Checkstyle Warnings - Category Whitespace

Summary

Total	High Priority	Normal Priority	Low Priority
1602	0	1602	0

Details

Folders	Files	Types	Warnings	Details
---------	-------	--------------	----------	---------

Type	Total	Distribution
EmptyLineSeparatorCheck	119	
FileTabCharacterCheck	25	
WhitespaceAfterCheck	84	
WhitespaceAroundCheck	1374	
Total	1602	

PART 3 Static Analysis

Checkstyle : Whitespace

- EmptyLineSeparatorCheck
 - 초기화 시에 공백이 필요, 혹은 잘못된 공백 사용
- FileTabCharacterCheck
 - 소스코드 내 tab 사용
- WhitespaceAfterCheck
 - 토큰 뒤에 공백이 필요
- WhitespaceAroundCheck
 - 토큰 앞뒤에 공백이 필요

```

015     private String name;
016     private int level;
017     private Superman superman;

```

Child.java:90, WhitespaceAfterCheck, Priority: Normal

',' is not followed by whitespace.

Checks that a token is followed by whitespace.

```

088     public void wordtrainPractice(Controller ctr) throws URISyntaxException
089     {
090         new GUI_Wordtrain_Practice(ctr, this.superman);
091     }

```

PART 3 Static Analysis

Checkstyle : T6

CheckStyle Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
1201	0	0

Summary

Total	High Priority	Normal Priority	Low Priority
1201	0	1201	0

Details

Files	Categories	Types	Warnings	Details
	Category	Total	Distribution	
	Metrics	11		
	Naming	11		
	Whitespace	1179		
	Total	1201		

PART 3 Static Analysis

PMD

- Program May Dependable의 약자
 - 정해진 규칙에 따라 소스코드를 검사해서 위반사항을 찾아낼 수 있다.
 - Eclipse -plugin으로 사용할 수 있으며, 별도로 커맨드라인에서 사용 가능하다.
 - 분석 대상은 java, javaScript, XML 등이 있다.
-
- PMD에서 제공하는 카테고리로부터, 분석 대상 프로그램 언어인 Java에 맞는 카테고리 선정

PART 3 Static Analysis

PMD

- 카테고리의 큰 틀 16가지

Basic (rulesets/basic.xml)

Naming (rulesets/naming.xml)

Unused code (rulesets/unusedcode.xml)

Design (rulesets/design.xml)

Import statements (rulesets/imports.xml)

JUnit tests (rulesets/junit.xml)

Strings (rulesets/string.xml)

Braces (rulesets/braces.xml)

Code size (rulesets/codesize.xml)

Javabeans (rulesets/javabeans.xml)

Finalizers

Clone (rulesets/clone.xml)

Coupling (rulesets/coupling.xml)

Strict exceptions (rulesets/strictexception.xml)

Controversial (rulesets/controversial.xml)

Logging (rulesets/logging-java.xml)

PART 3 Static Analysis

PMD

PMD Result

Warnings Trend

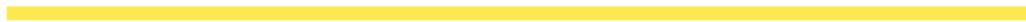
All Warnings	New Warnings	Fixed Warnings
14	1	149

Summary

Total	High Priority	Normal Priority	Low Priority
14	0	14	0

Details

Packages	Files	Categories	Types	Warnings	Details	New	Fixed
----------	-------	------------	-------	----------	---------	-----	-------

Category	Total	Distribution
Braces	1	
Code Size	6	
String and StringBuffer	7	
Total	14	

PART 3 Static Analysis

PMD : Basic

- 대부분의 개발자들이 동의하는 규칙: catch 블록들은 비어있어서는 안되고, equals()를 오버라이딩 할 때 마다 hashCode()를 오버라이드한다.
 - UnconditionalIfStatement
 - CollapsibleIfStatements
 - MisplacedNullCheck
 -
- T4, T6 모두 basic rule에서는 warning 발견되지 않음

PART 3 Static Analysis

PMD : Braces

- for, if, while, else 문장이 괄호를 사용하는지 여부 검사.
- Wordtrain.java:96 – if/else 문에서 괄호 사용 안함

[Wordtrain.java:96](#), IfElseStmtsMustUseBraces, Priority: Normal

Avoid using if...else statements without curly braces.

Avoid using if..else statements without using surrounding braces. If the code formatting or indentation is lost then it becomes difficult to separate the code being controlled from the rest.

```
// this is OK  
if (foo) x++;
```

```
// but this is not  
if (foo)  
    x = x+1;  
else  
    x = x-1;
```

```
if(str.charAt(0) == temp.charAt(temp.length()-1))  
{  
    return true;  
}  
else  
    return false;
```

PART 3 Static Analysis

PMD : Code Size

- 과도하게 긴 메소드, 너무 많은 메소드를 가진 클래스, 리팩토링에 대한 유사한 후보들을 위한 테스트.

[GUI_Wordtrain_Game.java:37](#), CyclomaticComplexity, Priority: Normal

The class 'GUI_Wordtrain_Game' has a Cyclomatic Complexity of 4 (Highest = 10).

Complexity directly affects maintenance costs is determined by the number of decision points in a method plus one for the method entry. The decision points include 'if', 'while', 'for', and 'case labels' calls. Generally, numbers ranging from 1-4 denote low complexity, 5-7 denote moderate complexity, 8-10 denote high complexity, and 11+ is very high complexity.

```
public class Foo { // This has a Cyclomatic Complexity = 12
1   public void example() {
2       if (a == b) {
3           if (a1 == b1) {
4               fiddle();
5           } else if a2 == b2) {
6               fiddle();
7           } else {
8               fiddle();
9           }
10          } else if (c == d) {
11              while (c == d) {
12                  fiddle();
13              }
14          } else if (e == f) {
15              for (int n = 0; n < h; n++) {
```

Decision point의 복잡성

→ 최대 10개

PART 3 Static Analysis

PMD : String

- 스트링 관련 작업을 할 때 발생하는 일반적인 문제들 규명. 스트링 리터럴 중복, String 구조체 호출, String 객체에 toString() 호출하기 등.

[Controller.java:66](#), AvoidDuplicateLiterals, Priority: Normal

The String literal "%5c" appears 8 times in this file; the first occurrence is on line 66.

Code containing duplicate String literals can usually be improved by declaring the String as a constant field.

```
private void bar() {
    buz("Howdy");
    buz("Howdy");
    buz("Howdy");
    buz("Howdy");
}
private void buz(String x) {}
```

PART 3 Static Analysis

PMD : Unused Code

- 결코 읽히지 않은 프라이빗 필드와 로컬 변수, 접근할 수 없는 문장, 결코 호출되지 않는 프라이빗 메소드 등을 찾기.

Type	Total	Distribution
UnusedLocalVariable	1	
UnusedPrivateField	5	
Total	6	

[Game.java:22](#), UnusedPrivateField, Priority: Normal

Avoid unused private fields such as 'name'.

Detects when a private field is declared and/or assigned a value, but not used.

PART 3 Static Analysis

FindBugs

- 자바프로그램 분석도구로, 메릴랜드 대학에서 개발했다.
- data flow, control flow 분석등을 이용한다.
- 버그 패턴을 자동으로 찾아서 알려준다 (ex : race condition, null Pointer exception 등)
- source code(*.java 파일) 보다는 byte code(*.class 파일)를 기반으로 분석을 진행한다.

PART 3 Static Analysis

FindBugs

FindBugs Result

Warnings Trend

All Warnings	New this build	Fixed Warnings
24	0	0

Summary

Total	High Priority	Normal Priority	Low Priority
24	4	20	0

Details

Files	Categories	Types	Warnings	Details	High	Normal
		Type	Total	Distribution		
		DM_DEFAULT_ENCODING	3			
		EQ_COMPARETO_USE_OBJECT_EQUALS	1			
		NP_ALWAYS_NULL_EXCEPTION	1			
		NP_NULL_ON_SOME_PATH_EXCEPTION	1			
		OBL_UNSATISFIED_OBLIGATION_EXCEPTION_EDGE	1			
		REC_CATCH_EXCEPTION	1			
		SBSC_USE_STRINGBUFFER_CONCATENATION	1			
		SF_SWITCH_NO_DEFAULT	2			
		URF_UNREAD_FIELD	6			
		UR_UNINIT_READ	3			
		UUF_UNUSED_FIELD	1			
		UWF_UNWRITTEN_FIELD	3			
		Total	24			

PART 3 Static Analysis

FindBugs

- DM_DEFAULT_ENCODING : byte에서 string으로 변환하는 작업 탐지.
- EQ_COMPARETO_USE_OBJECT_EQUALS : string에 대한 compareTo 사용 (priority queue)
- NP_ALWAYS_NULL_EXCEPTION : 항상 null인 pointer를 사용 (false warning 가능성 높다)
- NP_NULL_ON_SOME_PATH_EXCEPTION : null일 가능성이 있는 pointer를 사용 (")
- OBL_UNSATISFIED_OBLIGATION_EXCEPTION_EDGE : stream clean up 경고
- REC_CATCH_EXCEPTION : 발생하지 않는 exception에 대한 try-catch 처리
- SBSC_USE_STRINGBUFFER_CONCATENATION : 반복문 내에서 string 사용 (퍼포먼스 저하)
- SF_SWITCH_NO_DEFAULT : default가 없는 switch문
- URF_UNREAD_FIELD : never read area. 지우는 것 추천
- UR_UNINIT_READ : uninitialized read
- UUF_UNUSED_FIELD : 사용되지 않는 field
- UWF_UNWRITTEN_FIELD

PART 3 Static Analysis

JDepend

- 패키지 의존성과 관련된 설계 품질을 관리/분석 할 수 있다.
- 클래스 파일들을 읽어서 분석을 한다.
- 각 패키지별로 의존성 측정이 가능하며, 수치화, 그래프로 표현이 가능하다.
- 개발과정 중에도 아키텍처 품질을 평가할 수 있다.
- XML 형식으로도 저장 가능하다.

PART 3 Static Analysis

JDepend

Package	TC	CC	AC	Ca	Ce	A	I	D	V
<u>Classes</u>	14	14	0	1	2	0.0%	67.0%	33.0%	1
<u>GUI</u>	12	12	0	1	1	0.0%	50.0%	50.0%	1

※ 결과가 가지는 의미

Package	TC	CC	AC	Ca	Ce	A	I	D	V
<u>project</u>	47	47	0	0	1	0.0%	100.0%	0.0%	1

CC	Concrete Class
AC	Abstract Class 추상클래스나 인터페이스의 개수. 확정성의 척도
Ca	Afferent Couplings 현재 패키지에 종속성을 갖는 패키지 개수.
Ce	Efferent Couplings 현재 패키지가 종속하고 있는 패키지 개수.
A	추상화 정도. 0은 완전 구체적 패키지. 1은 추상적 패키지.
D	Main Sequence로 부터의 거리. Main Sequence란 추상적이면서 안정적이거나, 완전 구체적이면서 불안정한 패키지. 0일수록 가깝고 1일수록 멀다.

PART 3 Static Analysis

JDepend

Term	Description
Number of Classes	The number of concrete and abstract classes (and interfaces) in the package is an indicator of the extensibility of the package.
Afferent Couplings	The number of other packages that depend upon classes within the package is an indicator of the package's responsibility.
Efferent Couplings	The number of other packages that the classes in the package depend upon is an indicator of the package's independence.
Abstractness	The ratio of the number of abstract classes (and interfaces) in the analyzed package to the total number of classes in the analyzed package. The range for this metric is 0 to 1, with A=0 indicating a completely concrete package and A=1 indicating a completely abstract package.
Instability	The ratio of efferent coupling (Ce) to total coupling ($Ce / (Ce + Ca)$). This metric is an indicator of the package's resilience to change. The range for this metric is 0 to 1, with I=0 indicating a completely stable package and I=1 indicating a completely instable package.
Distance	The perpendicular distance of a package from the idealized line $A + I = 1$. This metric is an indicator of the package's balance between abstractness and stability. A package squarely on the main sequence is optimally balanced with respect to its abstractness and stability. Ideal packages are either completely abstract and stable ($x=0, y=1$) or completely concrete and instable ($x=1, y=0$). The range for this metric is 0 to 1, with D=0 indicating a package that is coincident with the main sequence and D=1 indicating a package that is as far from the main sequence as possible.
Cycles	Packages participating in a package dependency cycle are in a deadly embrace with respect to reusability and their release cycle. Package dependency cycles can be easily identified by reviewing the textual reports of dependency cycles. Once these dependency cycles have been identified with JDepend, they can be broken by employing various object-oriented techniques.

PART 3 Static Analysis

JDepend

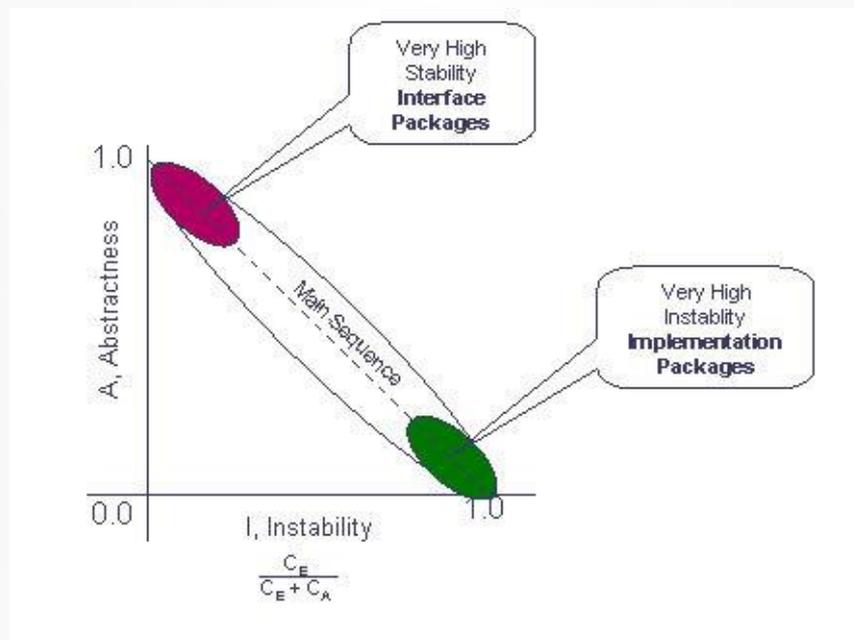
Afferent Couplings	Efferent Couplings	Abstractness	Instability	Distance
0	1	0.0%	100.0%	0.0%
Abstract Classes	Concrete Classes	Used by Packages	Uses Packages	
<i>None</i>	project.AlphaPanel project.AlphaPanel\$1 project.AlphaPanel\$2 project.AlphaPanel\$3 project.AlphaPanel\$4 project.Controller project.FindWord project.FindWordTest project.Game project.GamePanel project.GamePanel\$1 project.GamePanel\$2 project.GamePanel\$3 project.GamePanel\$4 project.GamePanel\$5 project.GamePanel\$6 project.GameTest project.Main project.MainPanel project.MainPanel\$1 project.MainPanel\$2 project.MainPanel\$3 project.MainPanel\$4 project.MainPanel\$5	<i>None</i>	org.junit	

PART 3 Static Analysis

JDepend

※ 결과가 가지는 의미

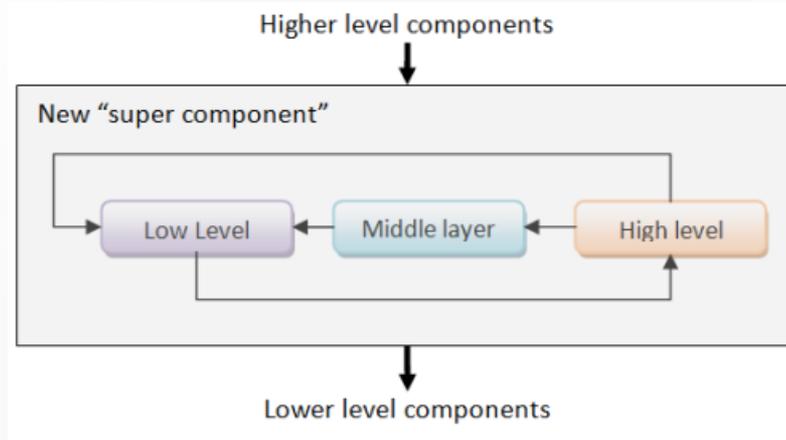
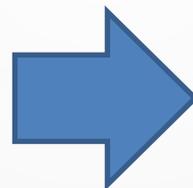
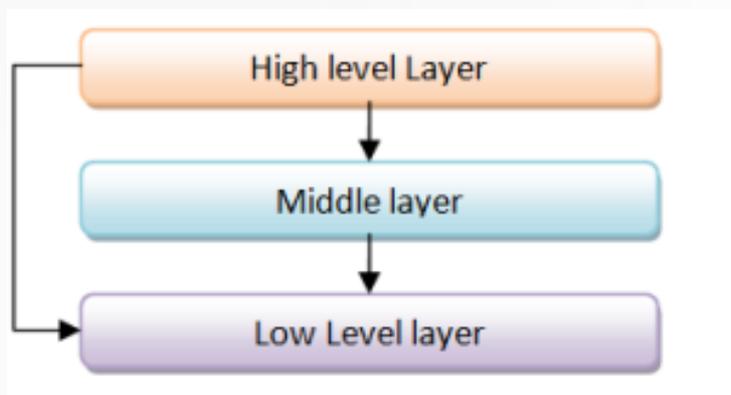
- D 값이 낮을 수록 인터페이스 기반의 개발. Coupling이 적음
- 1에 가까울수록 다른 클래스들이 의존 해야 될 클래스이고
- 0에 가까울수록 다른 클래스에 의존성을 가져야 한다.



PART 3 Static Analysis

JDepend

- 큰 규모의 개발을 쉽게 하기 위해서 덩어리 기반으로 쪼개어 개발을 하게 된다.
- 쪼개어 개발함으로써 개발과 유지보수에 도움이 된다.
- Dependency 구조가 꼬여서 cyclic 구조가 생기게 되면, 쪼개놓은 것이 하나로 합쳐지는 것이 된다.



Cycles

[[summary](#)] [[packages](#)] [[cycles](#)] [[explanations](#)]

There are no cyclic dependencies.

Package	Package Dependencies
Classes	GUI Classes
GUI	Classes GUI

PART 5

Summary

PART 5 Summary

Total Test Result

Static Analysis Tool	Team4	Team6
Checkstyle	1667 warnings	1201 warnings
PMD	14 warnings	43 warnings
Findbugs	29 warnings	24 warnings
JDepend	Dependency average 40%	Dependency average 0%

THANK YOU
ANY QUESTIONS?