

# 무 계산 시계

3rd Cycle

T2

201410935 조현종


201511903 박재영


201612368 이지우


201814122 이예인




# CONTENTS

 Specification Review

 System Testing

 Static Analysis

 OOAD Review





# Specification Review



# Specification Review

< Use case 3 / 5 / 13 >

Alternative Courses of Events에서의 Issue

[입력된 단위가 허용범위가 아닌 경우 최대 혹은 최소로 초기화 한다]

Stage	2041
Priority	High
Type	Document
Problem	<p><b>허용범위</b>가 어떻게 설정되어 있으며 어떤 값이 최대/최소인지 명시되지 않았고 어떤 상황에 최대/최소로 초기화하고 어떤 상황에 최소로 초기화되는지 명시되지 않음.</p>

Accept

<Use Case 3. Set Time>

=> 연도에 대해서 1990-2030의 범위 명시

Accept

<Use Case 5. Set Timer>

=> 23:59:59의 범위 표시

Reject

<Use Case 5. Set Timer>

=> 시, 분만 설정 가능하다고 Use Case에 명시  
시, 분은 허용범위가 명확하므로 Reject



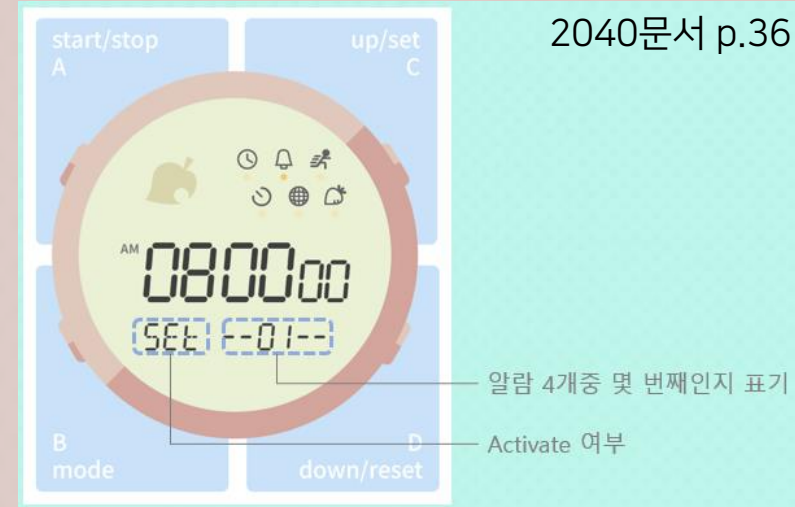
# Specification Review

< Use case 4 / 6 / 11 / 14 / 15 >  
 UI에서 표시되는 부분에 대한 설명 부재

Stage	2041
Priority	Low
Type	Document
Problem	<p>Sub Segment Display에          &lt;Use Case 4&gt; RUN          &lt;Use Case 6&gt; WAIT          &lt;Use Case 14&gt; SET          &lt;Use Case 15&gt; ---</p> <p>Main Segment Display에          &lt;Use Case 11&gt; 초기화된 시간</p> <p>이 표시되나 이에 대한 설명 없음.</p>

## Reject

Sub Segment Display의 경우, 현재 시계의 상태를 나타냄.  
 => Timer의 경우, 시작된 상태(RUN)와 정지된 상태(WAIT)  
 => Alarm의 경우, 활성화된 상태(SET)와 비활성화된 상태(---)  
 사용자의 이해를 돕기 위해 UI에 나타내는 것으로 2042부분에서 나타냄.



Main Segment Display의 경우,  
 <Use Case 11. Reset Stopwatch>에 0으로 초기화한다고 함.

# Specification Review

< Use case 20 / 22 / 23 >

무 값이 저장되는 시간대라는 표현의 모호함.

Stage	2041
Priority	High
Type	Document
Problem	시간대가 어떤 단위인지 명시되지 않음.

Accept

<b>22. Change Date</b>
User
Evident
Turnip Calculator Mode여야 한다.
(A) : Actor, (S) : System
1. (A) 사용자가 A버튼을 눌러 현재 표기되지 않은 다음 시간의 무 값을 요청한다.
2. (S) 다음 시간대의 입력되거나 계산된 무 값을 출력한다. (일요일은 하나의 무 값을 가지고 나머지 요일은 AM,PM에 각각 하나씩 무 값을 가진다)
<b>23. Alarm at High</b>
None
Hidden
무 값이 최고 값일 때의 시간대에 설정된 알람의 시간(AM-9:00, PM-12:00)이 현재 시간과 같아야 한다.

=> 시간대에 대해 정확히 명시

[ 시간대 : AM(0시-12시), PM(12시-24시) ]

[ 알람 시간 : AM 9시, PM 12시 ]



# System Testing



# System Testing

## Category-Partition Testing

Test Case = 168

Test Failed = 0

Test Success = 168

Test 성공률 = 100%

## Pair Wise Testing

Test Case = 84

Test Failed = 0

Test Success = 84

Test 성공률 = 100%

## Brute-Force Testing

Test Case = 6

Test Failed = 1

Test Success = 5

Test 성공률 = 83.3%



# Brute-Force Testing Issue

#	Test Case	Test Result
5	Timer가 시간이 0초가 되어 Beep가 울리기 시작한 후, 5초 이내로 다시 타이머를 Reset하고 Timer시작	Fail



```
if(!clip.isActive()) {  
    clip.setFramePosition(0);  
    clip.start();  
    try {  
        sleep(5000);  
    } catch (Exception e) {  
        e.printStackTrace();  
    };  
    clip.stop();  
}
```

기존에 sleep(5000)으로 5초간 Clip 재생 reqBeep()마다 오디오에 접근  
▶ sleep으로 인해 새로운 Beep 재생 안됨.

수정 후 setFramePosition과 start로 Clip 재생 (setFramePosition(0) : 재생위치를 처음으로)



# Static Analysis



# Static Analysis

Dir	Line	Rule	Severity	Review
Alarm.java	45	Code Smell	Minor	코드를 지울 때는 주석 처리가 아니라 완전히 지우는 것이 좋다.
Buzzer.java	-	Code Smell	Major	static 변수를 class 인스턴스로 접근 가능하면 위험하다.
Buzzer.java	-	Code Smell	Major	static 변수를 non-static 메소드에서 사용하면 위험하다.
Buzzer.java	33	Code Smell	Minor	try-catch문 뒤에 무의미한 세미 콜론이 붙어있다.
Controller.java	1	Code Smell	Minor	사용되지 않는 패키지 (java.lang.reflect.UndeclaredThrowableException)가 import 되어있다.
Controller.java	405	Bug	Major	404라인에서 할당되었던 alarm[] 변수의 길이 곧장 새로 할당되었다.
GUI.java	94 ~97	Code Smell	Minor	new Integer(n) 같은 생성자 메소드를 사용하는 것 보다 valueOf를 사용하는 것이 메모리 낭비도 적고 더 깔끔하다.
GUI.java	108 ~112	Code Smell	Minor	modeIndicator부터 segPath2까지의 멤버 변수들은 하나의 메소드 안에서만 할당되고 쓰이는 변수이므로 지역 변수로 바꾸는 것이 좋다.
GUI.java	157	Code Smell	Minor	catch문이 비어있는 것은 권장되지 않는다.
TurnipCalc.java	6	Code Smell	Minor	switch문에 default case가 없는 것은 권장되지 않는다.
TurnipCalc.java	63	Security Hotspot	Minor	Random보다 안전한 SecureRandom과 getInstance메소드를 사용하길 권장한다.



Dir	TurnipCalc.java
Review	Random보다 안전한 SecureRandom과 SecureRandom이 가진 getInstance 메소드를 사용하는 것이 권장된다.

**Reject**

무 예측 값 계산 과정에서  
현실성 있는 값 계산을 위해 Random값을 더해줌.

보안과 관련되어 사용되지 않기 때문에  
Random값이 예측 가능해도  
시스템 안정성과 관련하여 발생하는 문제 없음.



# OOAD Review



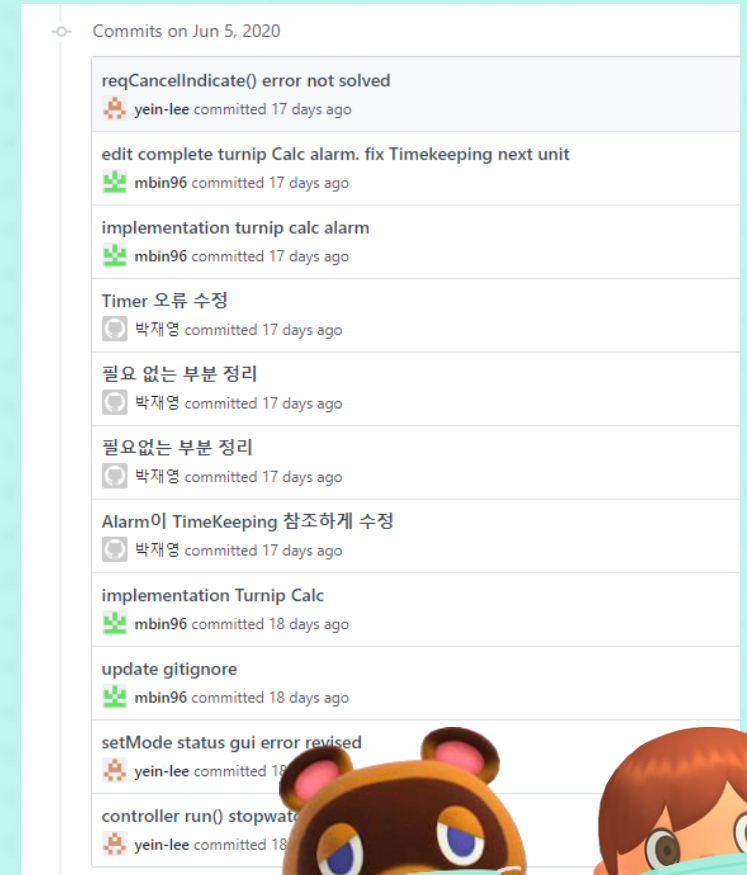
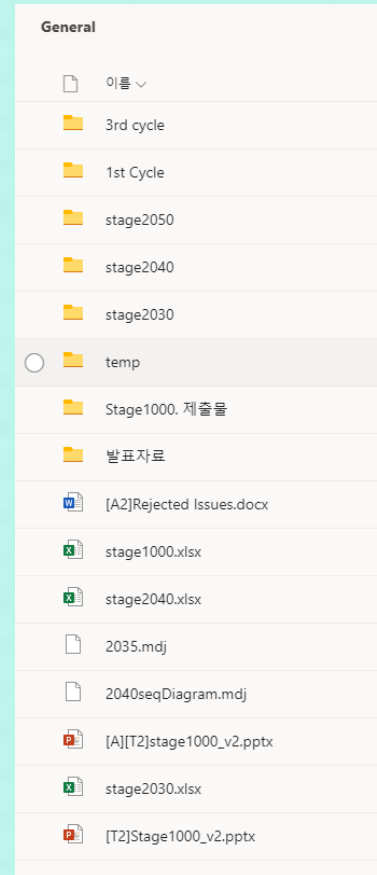


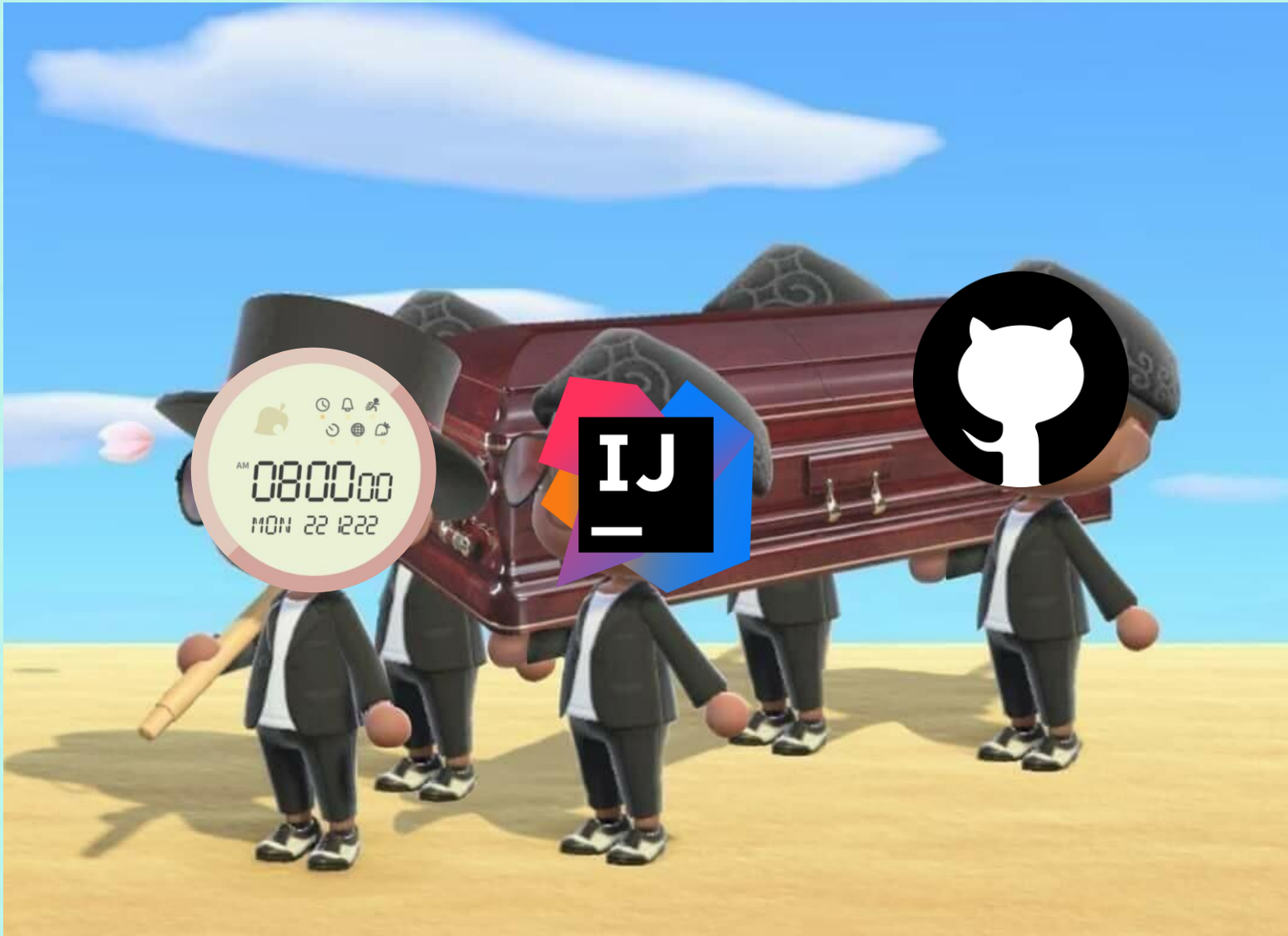
## 온라인 미팅

- Microsoft Teams를 이용한 공유 문서 작업
- 실시간 통화를 통한 회의(4월 28일부터 6월 22일까지 56일간 1시간 반씩 32통화 + Zoom)
- 2번의 오프라인 미팅

## CTIP 환경

- Git Commit 100회
- Master, dev 브랜치 분리를 통한 리비전관리의 효율화
- Notion을 통해 실시간 이슈 관리
- SonarQube를 통한 static analysis 경험







**감사합니다!**