



Alarm Customizing Digital Watch

2nd Cycle : Static Analysis

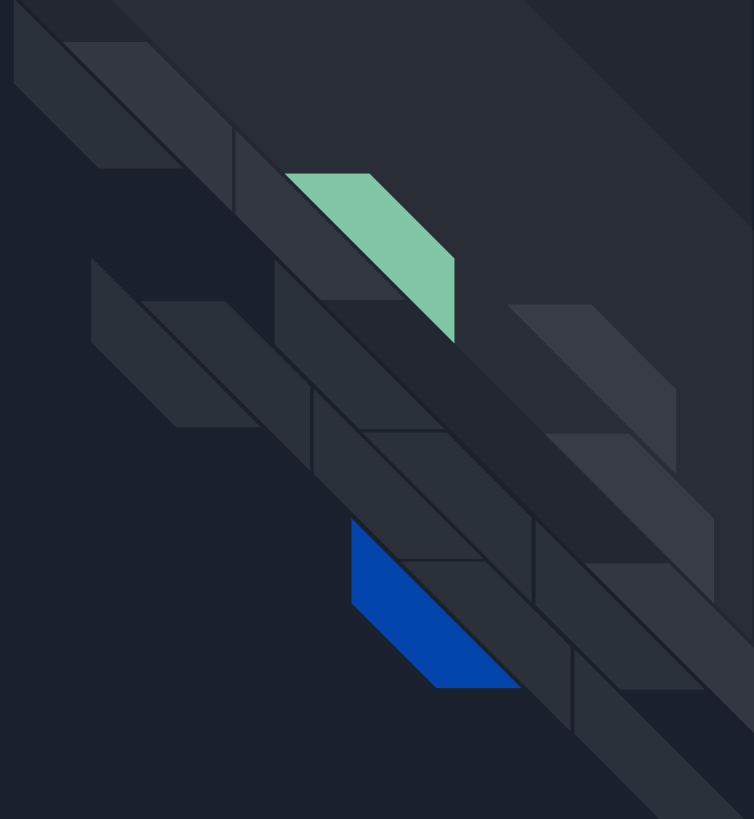
201511172 컴퓨터공학부 강민호
201511257 컴퓨터공학부 남관우
201511271 컴퓨터공학부 신윤섭
201810502 컴퓨터공학부 전현지

Index

001. Incomplete Part Completion

002. Specification Revision

003. Testing Revision



001. Incomplete Part Completion

1. 알람, 디데이터가 시간에 맞춰 울리는 기능 추가

Timekeeping.java

```
int alarmNum = system.alarm.getSize();
AlarmData[] alarmList = system.alarm.getAlarmList();
for (int i = 0; i < alarmNum; i++) {
    if (curTime.equals(alarmList[i].getTime())) { // 선택된 알람 인터벌이랑 블룸
        system.beepBuzzer(alarmList[i].getInterval(), alarmList[i].getVolume());
        break;
    }
}
```

현재시간과 알람을 비교하여, 알람이 울려야하는 시간이면 선택된 알람 interval, volume에 맞게 울림.

```
if(system.d_day.getD_day() != -1) {
    if(system.d_day.getD_dayDate().getCurrentDate().equals(curDate.getCurrentDate())) {
        system.startBorder();
        Date date = new Date();
        date.setDate( y: 0, m: 0, d: 0);
        system.d_day.setDate(date);
    }
}
```

현재날짜와 d-day를 비교하여, d-day 라면 화면 테두리가 깜빡임.

001. Incomplete Part Completion

2. timeout 기능 구현

Timekeeping, stopwatch, timer, d_day, alarm, alarmcustom 6가지 기능 모두 timeout 구현 완료

```
public void startCheckTimeOut() {
    checkTimeOut = new Thread(() -> {
        while (true) {
            try {
                Thread.sleep( millis: 1000);
                if (java.lang.System.currentTimeMillis() - lastOperateTime >= TIMED_OUT) {
                    lastOperateTime = java.lang.System.currentTimeMillis();
                    switch (selectedFid) {
                        case 1:
                            if (timeKeeping.getMode() == 0 && mode == 1) {
                                this.cancel();
                                GUI.setView(GUI.timekeepingView);
                            } else if (timeKeeping.getMode() == 1 && mode == 0) {
                                timeKeeping.cancel();
                                GUI.timekeepingView.borderPanel.setVisible(false);
                            }
                            break;
                        case 2:
                            if (stopwatch.getMode() == 2) {
                                stopwatch.cancel();
                                GUI.stopwatchView.borderPanel.setVisible(false);
                            }
                            break;
                    }
                }
            }
        }
    });
}
```

```
case 3:
    if (timer.getMode() == 1) {
        timer.cancel();
        GUI.timerView.borderPanel.setVisible(false);
        String tmp = timer.getTimer().getCurrentTime();
        StringTokenizer st = new StringTokenizer(tmp, delim: " ");
        GUI.timerView.setHour(String.format("%02d", Integer.parseInt(st.nextToken())));
        GUI.timerView.setMinute(String.format("%02d", Integer.parseInt(st.nextToken())));
        GUI.timerView.setSecond(String.format("%02d", Integer.parseInt(st.nextToken())));
    }
    break;
case 4:
    if (d_day.getMode() == 1) {
        d_day.cancel();
        GUI.d_dayView.borderPanel.setVisible(false);
        if (d_day.getD_day() == -1) {
            GUI.d_dayView.setYear(" ");
            GUI.d_dayView.setMonth("NO");
            GUI.d_dayView.setDate("NE");
        } else {
            String curDate = d_day.getD_dayDate().getCurrentDate();
            StringTokenizer st = new StringTokenizer(curDate, delim: " ");
            GUI.d_dayView.setYear(String.format("%02d", Integer.parseInt(st.nextToken()) % 100));
            GUI.d_dayView.setMonth(String.format("%02d", Integer.parseInt(st.nextToken())));
            GUI.d_dayView.setDate(String.format("%02d", Integer.parseInt(st.nextToken())));
        }
    }
    break;
```

001. Incomplete Part Completion

2. timeout 기능 구현

Timekeeping, stopwatch, timer, d_day,
alarm, alarmcustom 6가지 기능 모두
timeout 구현 완료

```
        case 5:
            if (alarm.getMode() != 0) {
                alarm.cancel();
                GUI.alarmView.borderPanel.setVisible(false);
            }
            break;
        case 6:
            if (alarmCustom.getMode() != 0) {
                alarmCustom.cancel();
                GUI.alarmCustomView.borderPanel.setVisible(false);
            }
            break;
    }
    GUI.setView(GUI.timekeepingView);
    functionNumIdx = 0;
    selectedFid = 1;
}
} catch (InterruptedException e) {
    e.printStackTrace();
}
}
});
```

001. Incomplete Part Completion

3. unit test 수정

setTimeTest() → 경계값을 넘어가는지 확인, 알람과 d-day 동기화 되는지 확인

```
//경계값 넘어가는지 확인
timekeeping.requestTimeSettingMode();
for (int i = 0; i < 5; i++) {
    timekeeping.changeValue( diff: 5000);
    timekeeping.changeType();
}
timekeeping.changeValue( diff: 5000);
timekeeping.requestSave();
assert (time.toString().equals("23 59 59"));
assert (date.getCurrentDate().equals("2099 12 31"));
```

```
// 알람 동기화 되는지
system.alarm.requestAlarmSettingMode(); // mode = 1;

// 순서대로 시, 분, 초 입력
system.alarm.changeValue( diff: 2); // 시
system.alarm.changeType();
system.alarm.changeValue( diff: 2); // 분
system.alarm.changeType();
system.alarm.changeValue( diff: 2); // 초

//request Save
system.alarm.requestSave();

system.timeKeeping.setAlarmCnt(system.alarm.getSize());
assert (system.timeKeeping.getAlarmCnt() == 1);
```

001. Incomplete Part Completion

3. unit test 수정

controlStopwatchRecord() → 포인터가 경계값을 넘어가는지 확인

```
stopwatch.requestRecordCheckMode();
assertTrue( condition: stopwatch.getRecordPointer() == 0); // requestRecordCheckMode시 pointer가 0으로 초기화 되는지 확인

stopwatch.movePointer( diff: -1);
assertTrue( condition: stopwatch.getRecordPointer() == 0); // 0 아래로 내려가는지 확인

stopwatch.movePointer( diff: 1);
assertTrue( condition: stopwatch.getRecordPointer() == 1);

stopwatch.movePointer( diff: 1);
assertTrue( condition: stopwatch.getRecordPointer() == 2);

stopwatch.movePointer( diff: 1);
java.lang.System.out.println(stopwatch.getRecordPointer());
assertTrue( condition: stopwatch.getRecordPointer() == 2); // 기록이 3개이므로, pointer가 2 초과하여 증가하지 않는지 확인
```

001. Incomplete Part Completion

3. unit test 수정

setTimerTest() → 경계값을 넘어가는지 확인

```
// 경계값 확인
```

```
timer.requestTimerSettingMode();
for(int i = 0; i < 2; i++) {
    timer.changeValue( diff: 3000);
    timer.changeType();
}
timer.changeValue( diff: 3000);
timer.requestSave();

time = timer.getTimer();

timeStr = time.getCurrentTime();
String splitedTime2[] = timeStr.split( regex: " ");

assert(splitedTime2[0].equals("99"));
assert(splitedTime2[1].equals("59"));
assert(splitedTime2[2].equals("59"));
```

resetTimerTest() → pause된 상태에서도 reset이 되는지 확인

```
// pause 상태에서도 reset 되는지 확인
timer.requestTimerSettingMode();

timer.changeType();
timer.changeType();
timer.changeValue( diff: 5); // Timer를 5초로 Setting
timer.requestSave();

timer.requestStartTimer();

try {
    Thread.sleep( millis: 1100); // 1초가 흐르게 되네요
} catch(InterruptedException e) {
    java.lang.System.out.println(e.getMessage());
}

timer.requestPauseTimer();
time = timer.getTimer();

timeStr = time.getCurrentTime();
splitedTime = timeStr.split( regex: " ");

assert(splitedTime[2].equals("4"));

timer.requestResetTimer();

time = timer.getTimer();

timeStr = time.getCurrentTime();
splitedTime = timeStr.split( regex: " ");

assert(splitedTime[0].equals("0"));
assert(splitedTime[1].equals("0"));
assert(splitedTime[2].equals("0"));
```


001. Incomplete Part Completion

3. unit test 수정

setAlarmTest() → 경계값을 넘어가는지 확인

```
// ### 시, 분, 초 경계값 확인
// 순서대로 시, 분, 초 입력
alarm.changeValue( diff: -60); // 시
alarm.changeType();
alarm.changeValue( diff: -60); // 분
alarm.changeType();
alarm.changeValue( diff: -60); // 초
```

controlAlarmListTest()
→ 경계값을 넘어가는지 확인

```
assertEquals( expected: 0, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 1, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 2, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 3, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 4, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 5, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 6, alarm.getAlarmPointer());
alarm.movePointer( diff: 1);
assertEquals( expected: 6, alarm.getAlarmPointer()); // 경계값 걸리는지
```



002. Specification Revision

1. Stage 1000

1006		<p>Use case 의 description 이 더 상세하게 작성되어 있어야 한다. 해당 use case 의 구체적인 역할을 명시해야 한다.</p> <p>Ex)사용자가 기본 화면에서 시간과 날짜를 설정할 수 있다.</p> <p>설정 가능한 시간 단위는 시, 분, 초이고, 시는 0~23, 분과 초는 0~59 까지 바꿀 수 있다. 날짜는 YY/MM/DD 순으로 설정할 수 있다. 요일은 날짜가 바뀌는 것에 맞추어 자동으로 바뀌도록 한다.</p>
------	--	--

002. Specification Revision

1. Stage 1000

1006. Define Business Use Case - Describe Use Cases

Use Case	1. Set Time
Actors	User
Description	사용자가 기본 화면에서 현재 날짜, 요일, 시간을 설정한다. 설정 가능한 시간 단위는 시, 분, 초이고, 시는 0~23, 분과 초는 0~59까지 바꿀 수 있다. 날짜는 YY/MM/DD 순으로 설정할 수 있다. 요일은 날짜가 바뀌는 것에 맞추어 자동으로 바뀌도록 한다.
Use Case	2. Display Time
Actors	None
Description	현재 날짜, 요일, 시간을 TimeKeeping 화면에 보여준다.

002. Specification Revision

2. Stage 2030

2031	14	Typical courses of events 의 설명이 모호하다.
------	----	---------------------------------------




Typical Courses of Events	(A):Actor, (S):System
	1. (A) : 기능 설정 화면을 요청한다.
	2. (S) : 기능 설정 화면으로 전환한다. 6개 기능(1. Timekeeping, 2. Stopwatch, 3. Timer, 4. D-day, 5. Alarm, 6. AlarmCustom) 중 현재 설정되어 있는 4개 기능의 숫자가 4개 구간에 순서대로 나타난다.
	3. (A) : 선택 버튼으로 4개 구간을 순서대로 이동하여 수를 설정할 구간을 선택한다.
	4. (S) : 사용자가 설정중인 구간을 표시한다.
	5. (A) : 구간마다 숫자를 지정한다.
	6. (A) : 모든 설정을 마치고 저장을 요청한다.
	7. (S) : 사용자의 변경 사항을 저장하고 TimeKeeping 화면으로 돌아간다.

002. Specification Revision

2. Stage 2030

2031	46	Timer, stopwatch 를 진행하고 있을 경우에는 작동하지 않는다. →Timer, stopwatch 가 설정이 완료되어 실행 중인 경우에는 작동하지 않는다.(표현의 모호함)
------	----	---



timeout은 어떤 상황에서도 발생하는 것으로 수정하였다.

2031. Define Essential Use case

Use Case	30.Time Out
Actor	None
Type	Hidden
Pre-Requisites	마지막 행동을 한 후 아무 동작 없이 10분이 지나야 한다. TimeKeeping 화면의 기본 모드가 아니어야 한다.
Typical Courses of Events	(S): System 1. (S) : TimeKeeping 화면의 기본 모드로 돌아간다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

003. Testing Revision

1. Set Time

년, 월, 일에 따른 요일이 계산되어 즉시 설정되는지 확인한다.

→바로 변경되는 것이 아니라 다른 화면으로 넘어갔다가 돌아와야 변경된다.

Fail

“즉시”라는 표현이 모호했다.
실제로는 1초 후에 갱신된다.
1초 후 부터는 정상적으로 작동한다.



sys09270883 Yesterday at 10:28 PM

이 부분도 어떻게 안됐었는지 궁금하네요

image.png ▾

년, 월, 일에 따른 요일이 계산되어 즉시 설정되는지 확인한다.
→바로 변경되는 것이 아니라 다른 화면으로 넘어갔다가 돌아와야 변경된다.

Fail

2 replies



Yoona Heo 11 hours ago

아 텀이 있었군요요 방금 한번 더 테스트해보니까 잘 됩니다



sys09270883 3 minutes ago

네 확인했습니다 감사합니다

003. Testing Revision

1. Set Time: 윤달 예외처리

각 월에 따른 일의 범주를 넘어서지 않는지 확인한다(ex. 2월 30일은 존재하지 않는다).

→2월을 제외한 나머지 월은 전부 맞다(2021년에는 2월 28일까지만 존재하지만, 2024년 2월에는 29일까지 존재한다. 윤달을 고려하지 않음).

Fail

timekeeping과 d-day에서 윤달을 예외처리함.

```
if(timeSettingValue[4] == 2) {
    int temp, year = timeSettingValue[3];
    int tmpNumOfDay;
    temp = year % 4;
    if (temp == 0) {
        temp = year % 100;
        if (temp == 0) {
            temp = year % 400;
            if (temp == 0) {
                tmpNumOfDay = 29;
            }
            else
                tmpNumOfDay = 28;
        }
        else {
            tmpNumOfDay = 29;
        }
    } else
        tmpNumOfDay = 28;
}
```


003. Testing Revision

2. Display Time: d-day 표시

설정된 현재 시각, 날짜, 요일, D-day, 알람 개수, 아이콘이 잘 표시되는지 확인한다.

→D-Day 가 설정되지 않아 표시되지 않는다.

Fail

d-day가 설정되어 있지 않으면 “000”으로 표시하게 설정함.



Display Time

시간, 날짜, 요일, 알람 아이콘, 알람 개수, D-day(설정이 되어있지 않으면 “000”)를 표시한다



003. Testing Revision

14. Display Stopwatch Record: 리스트 출력

기록이 최신순으로 출력되는지 확인한다.

→가장 오래된 것부터 출력된다.

Fail

기록을 오래된 순으로 출력하게 수정함.



Display Stopwatch Record

스톱워치 기록 리스트(최대 3개)를 오래된 순으로 보여준다.

003. Testing Revision

15. Control Stopwatch Record: 리스트 조회

기록이 존재하지 않을 경우 기록 확인 모드로 전환되지 않는지 확인한다.

→select button 을 2 초간 누르고 있으면 기록 확인 모드로 전환된다.

Fail

→ 기록이 존재하지 않을 경우 기록 확인을 못하게 예외처리함.

```
public void requestRecordCheckMode() {  
    if(getSize() == 0) return;  
    changeMode(2);  
}
```

003. Testing Revision

18. Beep Alarm

Beep Alarm	설정된 시간에 알람이 울리는지 확인한다. →알람이 울리지 않아서 확인할 수 없다.	Fail
	알람이 커스텀된 볼륨과 인터벌에 맞게 울리는지 확인한다. →알람이 울리지 않아서 확인할 수 없다.	Fail

1. 알람이 울리는 기능을 추가하였다.

1. 커스텀 기능도 정상적으로 작동한다.

```
// Alarm
if (system.alarm == null)
    return;

int alarmNum = system.alarm.getSize();
AlarmData[] alarmList = system.alarm.getAlarmList();
for (int i = 0; i < alarmNum; i++) {
    if (currentTime.equals(alarmList[i].getTime())) { // 선택된 알람 인터벌이랑 볼륨
        system.beep buzzer(alarmList[i].getInterval(), alarmList[i].getVolume());
        break;
    }
}
```



003. Testing Revision 19. Stop Alarm

Stop Alarm Buzzer	알람이 설정된 시간에 맞추어 버저가 울릴 때 어떤 버튼을 조작하더라도 알람이 멈추는지 확인한다. →알람이 울리지 않아서 확인할 수 없다.	Fail
----------------------	---	------



알람 기능을 구현하지 않았어서 Fail 된 Test.
구현 완료 했으며, 아무 버튼을 누르면 정상적으로 종료된다.



003. Testing Revision

26. Control Alarm Custom List

Control Alarm Custom List	알람 선택 모드가 specification 에 맞게 동작하는지 확인한다. →모드 진입 후 프로그램이 간혹 멈춘다. 멈추지 않을 때는 잘 동작한다.	Fail
	알람이 존재하지 않으면 알람 선택 모드로 진입하지 않는지 확인한다.	Pass



해당 상황을 경험하지 못해서 수정하지 못했다.

003. Testing Revision

30. Timeout

Time Out	TimeOut 이 발생했을 때 specification 에 맞게 TimeKeeping 화면의 기본 모드로 되돌아 가는지 확인한다. →설정을 변경 중이었을 경우 포인터만 사라지고 해당 화면에서 그냥 멈춰있다.	Fail
	정확히 아무런 동작을 하지 않은 지 10 분이 지났을 때 작동하는지 확인한다.	Fail
	입력하던 정보가 저장되지 않았는지 확인한다.	Pass

1. timeout 기능이 미구현이었는데 timekeeping 화면에 돌아가는 것까지 추가 구현하였다.
2. 10분이 지나면 정상적으로 작용한다.



003. Testing Revision

30. Timeout

1. timeout 코드에 timekeepingview로 돌아가는 코드를 추가하여 해결하였다.
2. 해당 뷰에서의 사용했던 변수들을 적절하게 초기화하였다.

```
GUI.setView(GUI.timekeepingView);  
functionNumIdx = 0;  
selectedFid = 1;
```




QnA

