

Software Modeling & Analysis

A REMIND WATCH

Project Team

Team 3

Team Members

1. 김우진 201511251
2. 박종엽 201511263
3. 이상현 201511279
4. 이정노 201511283

INDEX

Phase 2051. Implement Class & Method Definitions

Activity 2052. Implement windows

Activity 2055. Write Unit Test Code

Activity 2061. Unit Testing

Activity 2063. System Testing

Phase 2051. Implement Class & Method Definitions

Type	Class
Name	Pair
Purpose	객체 두개를 가지고 있기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Controller
Purpose	모든 컨트롤러에서 사용하는 함수들을 정의한다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Data
Purpose	컨트롤러별로 가지고 있어야 하는 데이터를 관리한다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	TimeData
Purpose	시간이 얼마나 흘렀는지에 관한 관리를 해주는 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	SelectFunction
Purpose	6개의 기능 중 사용할 4개의 기능을 사용하기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	DeltaTimeThread
Purpose	시간이 흐르는 단계에 맞춰 값을 갱신해주기 위한 스레드 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	CurrentTime
Purpose	현재 시간에 대한 값을 저장, 관리, 변경 해주기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Alarm
Purpose	알람에 대한 값을 저장, 관리, 변경 해주기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Timer
Purpose	타이머에 대한 값을 저장, 관리, 변경 해주기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Stopwatch
Purpose	스탑워치에 대한 값을 저장, 관리, 변경 해주기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Schedule
Purpose	스케줄에 대한 값을 저장, 관리, 변경 해주기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

Type	Class
Name	Expenditure
Purpose	지출액에 대한 값을 저장, 관리, 변경 해주기 위한 클래스이다.
Overview(class)	
CrossReference	
Exceptional Courses of Events	N/A

1) Controller

Type	Method
Name	moveCursor
Purpose	현재 커서를 이동시키는 함수이다.
CrossReference	
Input	Void
Output	Void
Abstract operation	1. 커서를 이동시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	increaseValue
Purpose	현재 커서를 변경시킨다..
CrossReference	
Input	Void
Output	Void
Abstract operation	<ol style="list-style-type: none"> 1. 커서 값을 변경하는 버튼을 누른다. 2. 현재 커서가 어딘지에 따라 값을 증가, 셋, 0 으로 바꿔주는 함수를 부른다.
Exceptional Courses of Events	N/A

Type	Method
Name	getData

Purpose 현재 데이터 값을 반환해주는 함수이다..

CrossReference

Input Void

Output HashMap<String, integer>

Abstract operation 1. 현재 데이터 값을 반환해준다.

Exceptional Courses of Events N/A

Type Method

Name saveValue

Purpose 현재 값을 저장하는 함수이다.

CrossReference

Input Void

Output void

Abstract operation 1. 현재 데이터 값을 저장한다.

Exceptional Courses of Events N/A

Type Method

Name setZero

Purpose 현재 커서의 값을 0으로 바꾸는 함수이다.

CrossReference

Input String, int

Output Int

Abstract operation 1. 현재 커서의 값을 0으로 반환한다.

Exceptional Courses of Events N/A

Type Method

Name SetUnsetValue

Purpose 현재 커서의 값을 on,off시키는 함수이다.

CrossReference

Input String, int

Output Int

Abstract operation 1. 현재 커서의 값을 on, off로 바꿔준다.

Exceptional Courses of Events N/A

Type Method

Name increaseElement

Purpose 현재 커서의 값을 증가시키는 함수이다.

CrossReference**Input** String, int**Output** Int**Abstract operation** 1. 현재 커서의 값을 1 증가 시킨다.**Exceptional Courses of Events** N/A**Type** Method**Name** increaseElement**Purpose** 현재 커서의 값을 증가시키는 함수이다.**CrossReference****Input** String, int**Output** Int**Abstract operation** 1. 현재 커서의 값을 1 증가 시킨다.**Exceptional Courses of Events** N/A**Type** Method**Name** saveData**Purpose** 현재 커서들의 값을 저장하는 함수이다.**CrossReference****Input** Void**Output** Void**Abstract operation** 1. 현재 커서들의 값을 data 객체 안에 저장한다.**Exceptional Courses of Events** N/A**Type** Method**Name** increaseElement**Purpose** 현재 커서의 값을 증가시키는 함수이다.**CrossReference****Input** String, int**Output** Int**Abstract operation** 1. 현재 커서의 값을 1 증가 시킨다.**Exceptional Courses of Events** N/A

2) Data

Type Method**Name** loadData

Purpose 데이터를 반환해주는 함수이다.

CrossReference

Input String, int

Output HashMap<String, Integer>

Abstract operation 1. 데이터를 반환해준다.

Exceptional Courses of Events N/A

Type Method

Name saveData

Purpose 데이터를 저장해주는 함수이다.

CrossReference

Input HashMap<String, Inter>,String, int

Output Boolean

Abstract operation 1. 현재 객체의 값을 저장한다.

Exceptional Courses of Events N/A

Type Method

Name Initialize_Data

Purpose 데이터를 초기화시키는 함수이다.

CrossReference

Input String

Output Boolean

Abstract operation 1. 데이터를 초기화 시키는 함수이다.

Exceptional Courses of Events N/A

Type Method

Name Delete_alarmData

Purpose 알람 데이터를 삭제하는 함수이다.

CrossReference

Input Int

Output Boolean

Abstract operation 1. Input 에 맞는 alarm 의 정보를 삭제한다.

Exceptional Courses of Events N/A

Type Method

Name Delete_alarmData

Purpose 알람 데이터를 삭제하는 함수이다.

CrossReference

Input Int

Output Boolean

Abstract operation 1. Input 에 맞는 alarm 의 정보를 삭제한다.

Exceptional Courses of Events N/A

3) SelectFunction

Type Method

Name checkFunctionCount

Purpose 선택된 함수가 4개인지 확인하기 위한 함수이다.

CrossReference

Input void

Output Boolean

Abstract operation 1. 선택된 함수가 4 개이면 true 를 반환 그 외의 경우 false 를 반환한다.

Exceptional Courses of Events N/A

Type Method

Name saveSelectFunctionData

Purpose 선택된 함수의 데이터를 데이터 객체에 저장한다.

CrossReference

Input void

Output Boolean

Abstract operation 1. 선택된 함수의 데이터를 데이터 객체에 저장한다.

Exceptional Courses of Events N/A

4) CurrentTime

Type Method

Name pauseTimeFlow

Purpose 현재 흐르고 있는 스레드를 멈추기 위한 함수이다.

CrossReference

Input void

Output Void

Abstract operation 1. 현재 시간이 흐르는 스레드를 정지시킨다.

Exceptional Courses of Events N/A

Type	Method
Name	contTimeFlow
Purpose	현재 멈춰있는 스레드를 시작하기 위한 함수이다.
CrossReference	
Input	Void
Output	Void
Abstract operation	1. 현재 멈춰있는 스레드를 돌린다.
Exceptional Courses of Events	N/A

5) Alarm

Type	Method
Name	increaseElement
Purpose	현재 커서의 값을 증가시킨다.
CrossReference	
Input	String, int
Output	Int
Abstract operation	1. 현재 커서의 값을 1 증가시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	getNextAlarmData
Purpose	다음 알람의 데이터를 불러온다.
CrossReference	
Input	Void
Output	Void
Abstract operation	1. 다음 알람의 데이터를 매핑시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	saveAlarmValue
Purpose	설정된 알람의 값을 저장하는 함수이다.
CrossReference	
Input	Void
Output	Void
Abstract operation	1. 설정된 알람의 값을 저장한다.

Exceptional Courses of Events	N/A
--------------------------------------	-----

Type	Method
-------------	--------

Name	deleteAlarm
-------------	-------------

Purpose	현재 보고있는 알람을 삭제하는 함수이다.
----------------	------------------------

CrossReference	
-----------------------	--

Input	Void
--------------	------

Output	Void
---------------	------

Abstract operation	1. 현재 보고 있는 알람을 삭제한다.
---------------------------	-----------------------

Exceptional Courses of Events	N/A
--------------------------------------	-----

6) Timer

Type	Method
-------------	--------

Name	startTimer
-------------	------------

Purpose	타이머를 시작하기위한 함수이다.
----------------	-------------------

CrossReference	
-----------------------	--

Input	Void
--------------	------

Output	Void
---------------	------

Abstract operation	1. 타이머를 시작한다.
---------------------------	---------------

Exceptional Courses of Events	N/A
--------------------------------------	-----

Type	Method
-------------	--------

Name	resumeTimer
-------------	-------------

Purpose	일시정지 되어있는 타이머를 시작하기 위한 함수이다.
----------------	------------------------------

CrossReference	
-----------------------	--

Input	Void
--------------	------

Output	Void
---------------	------

Abstract operation	1. 일시정지 되어있는 타이머를 시작한다.
---------------------------	-------------------------

Exceptional Courses of Events	N/A
--------------------------------------	-----

Type	Method
-------------	--------

Name	pauseTimer
-------------	------------

Purpose	돌고있는 타이머를 정지하기위한 함수이다.
----------------	------------------------

CrossReference	
-----------------------	--

Input	Void
--------------	------

Output	Void
---------------	------

Abstract operation 1. 돌고 있는 타이머를 정지한다.

Exceptional Courses of Events N/A

Type Method

Name initTimer

Purpose 일시정지, 정지된 타이머를 원래 값으로 초기화한다.

CrossReference

Input Void

Output Void

Abstract operation 1. 타이머 데이터를 불러온다.

2. 불러온 데이터를 보여준다.

Exceptional Courses of Events N/A

Type Method

Name saveTimerValue

Purpose 설정하던 타이머 값을 데이터에 저장하는 함수이다.

CrossReference

Input Void

Output Void

Abstract operation 1. 설정하던 타이머 값을 저장한다.

Exceptional Courses of Events N/A

7) Stopwatch

Type Method

Name startStopwatch

Purpose 스톱워치를 시작하기 위한 함수이다.

CrossReference

Input Void

Output Void

Abstract operation 1. 스톱워치를 시작한다.

Exceptional Courses of Events N/A

Type Method

Name resumeStopwatch

Purpose 일시정지 되어있는 스톱워치를 시작하기 위한 함수이다.

CrossReference

Input	Void
Output	Void
Abstract operation	1. 일시정지 되어있는 스톱워치를 시작한다.
Exceptional Courses of Events	N/A

Type	Method
Name	pauseStopwatch
Purpose	돌고있는 스톱워치를 정지시키기 위한 함수이다.
CrossReference	
Input	Void
Output	Void
Abstract operation	1. 돌고있는 스톱워치를 정지시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	initStopwatch
Purpose	스톱워치를 0으로 초기화해주기 위한 함수이다.
CrossReference	
Input	Void
Output	Void
Abstract operation	1. 스톱워치를 0으로 초기화한다.
Exceptional Courses of Events	N/A

8) Schedule

Type	Method
Name	increaseElement
Purpose	현재 스케줄의 커서 값을 증가시키는 함수이다.
CrossReference	
Input	Void
Output	Int
Abstract operation	1. 현재 커서의 값을 1 증가시킨다.
Exceptional Courses of Events	N/A

Type	Method
Name	saveScheduleData
Purpose	현재 스케줄의 값을 저장하는 함수이다.

CrossReference**Input** Void**Output** Void**Abstract operation** 1. 현재 스케줄의 값을 저장한다.**Exceptional Courses of Events** N/A**Type** Method**Name** initializeScheduleValue**Purpose** 모든 저장된 스케줄을 초기화하기 위한 함수이다.**CrossReference****Input** Void**Output** Void**Abstract operation** 1. 모든 스케줄을 삭제한다.**Exceptional Courses of Events** N/A**Type** Method**Name** getScheduleData**Purpose** 가장 가까운 스케줄 값을 반환하기 위한 함수이다.**CrossReference****Input** Void**Output** Void**Abstract operation** 1. 가장 가까운 스케줄 값을 반환한다.**Exceptional Courses of Events** N/A

9) Expenditure

Type Method**Name** Expenditure**Purpose** 현재 커서의 값을 증가시키기 위한 함수이다.**CrossReference****Input** String, int**Output** int**Abstract operation** 1. 현재 커서의 값을 1 증가시킨다.**Exceptional Courses of Events** N/A**Type** Method**Name** DecimalToData

Purpose 각 자리수별로 데이터를 쪼개는 함수이다.

CrossReference

Input Void

Output Void

Abstract operation 1. 각 자리수 별로 데이터를 쪼개준다.

Exceptional Courses of Events N/A

Type Method

Name DataToDecimal

Purpose 자리수 별로 저장된 데이터를 10진수로 바꿔주는 함수이다.

CrossReference

Input Void

Output Void

Abstract operation 1. 자리수 별로 저장된 데이터를 10진수로 바꿔준다.

Exceptional Courses of Events N/A

Type Method

Name loadExpenditureValue

Purpose 지출액을 불러오는 함수이다.

CrossReference

Input Void

Output Void

Abstract operation 1. 지출액을 불러온다

Exceptional Courses of Events N/A

Type Method

Name saveExpenditureValue

Purpose 지출액을 저장하기위한 함수이다.

CrossReference

Input Void

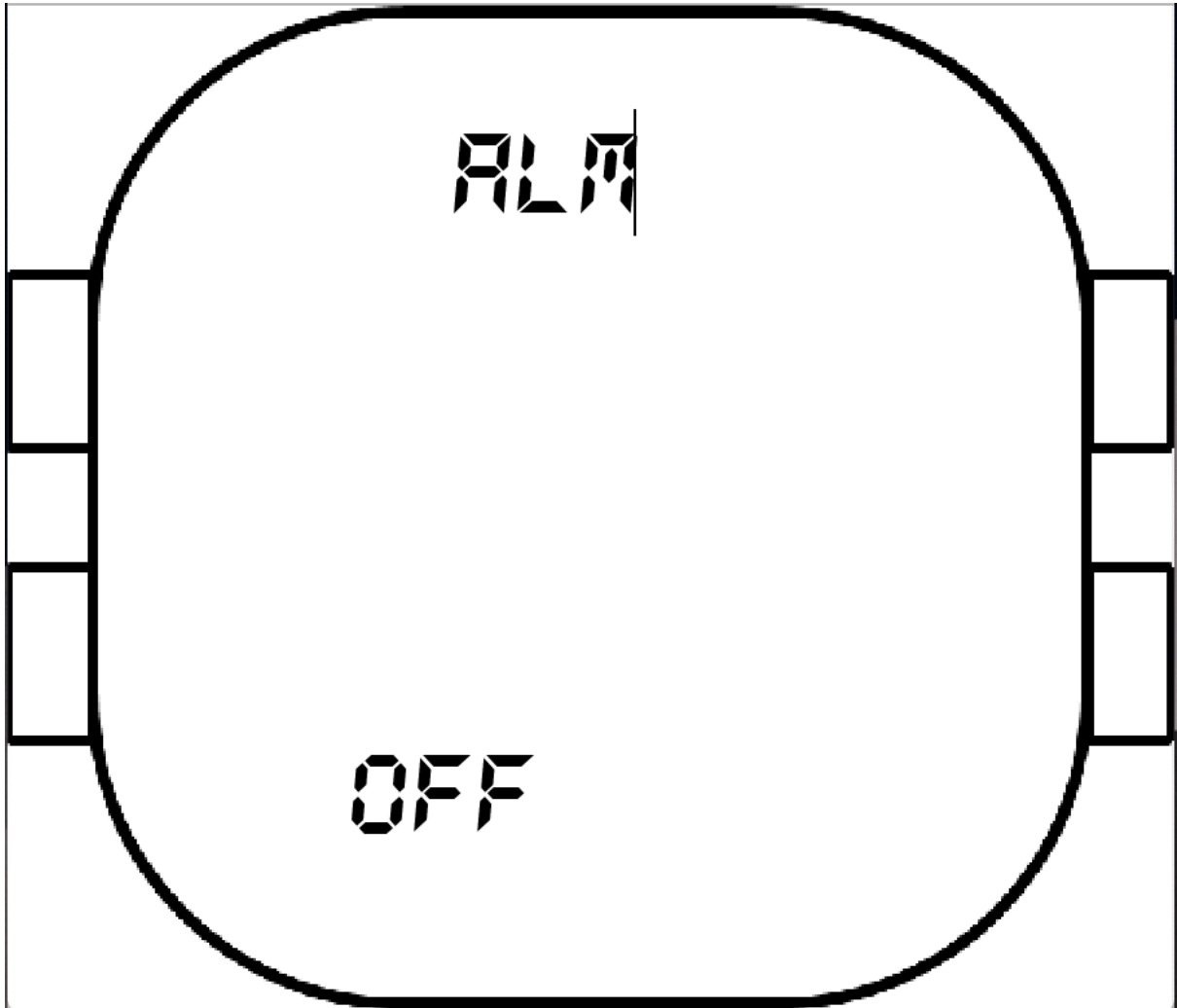
Output Void

Abstract operation 1. 지출액을 저장한다.

Exceptional Courses of Events N/A

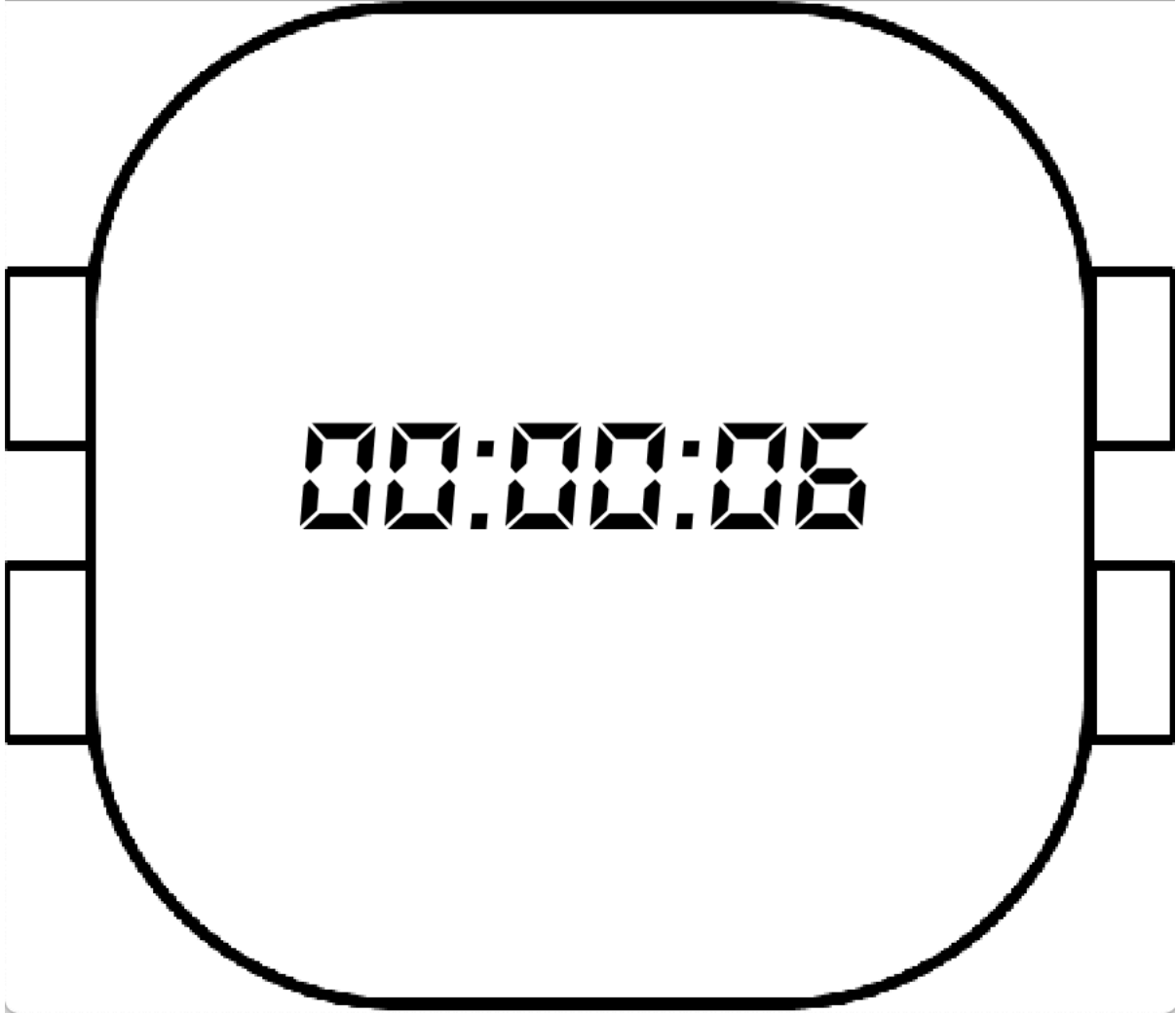
Activity 2052. Implement windows

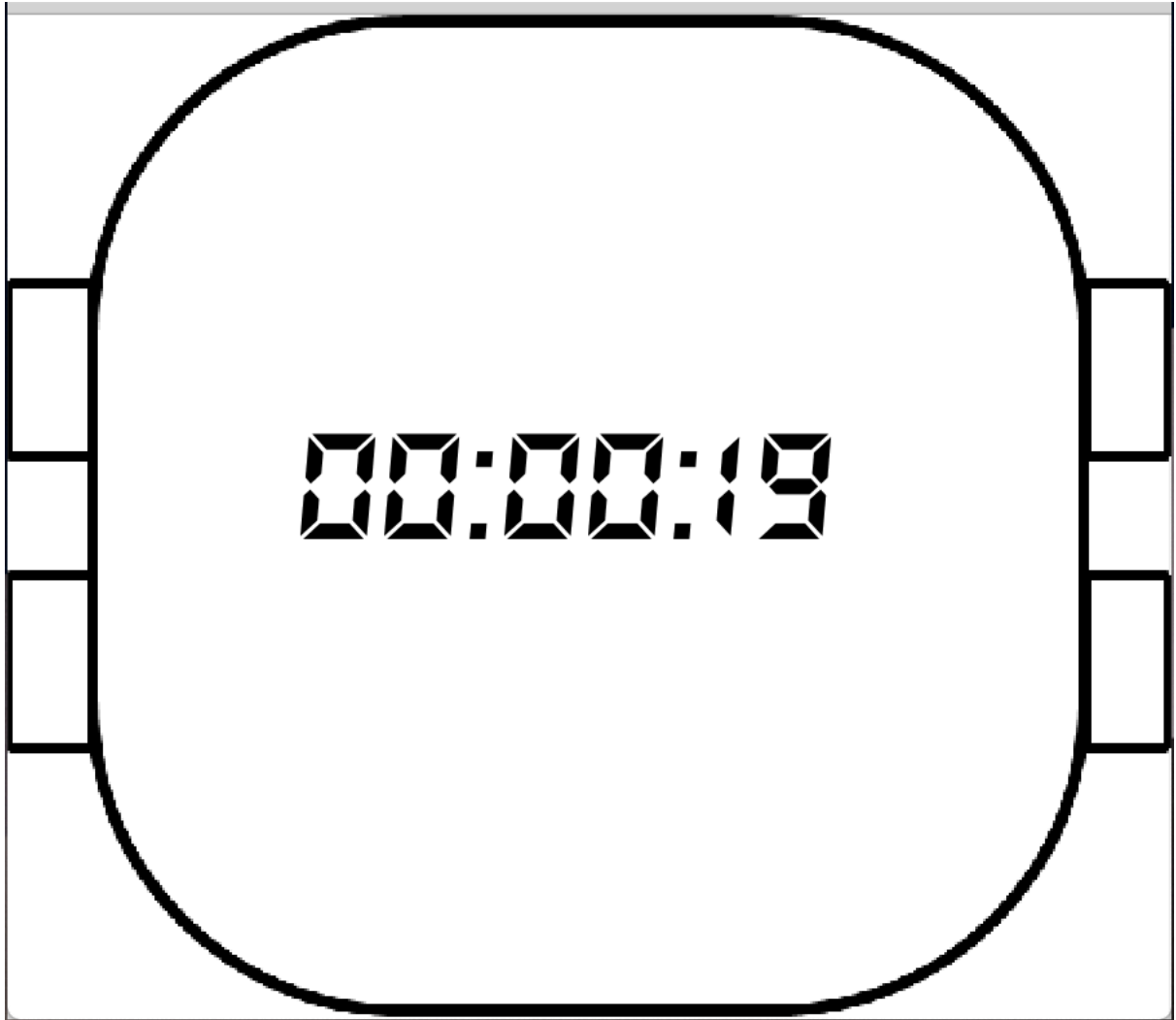
SelectFunction



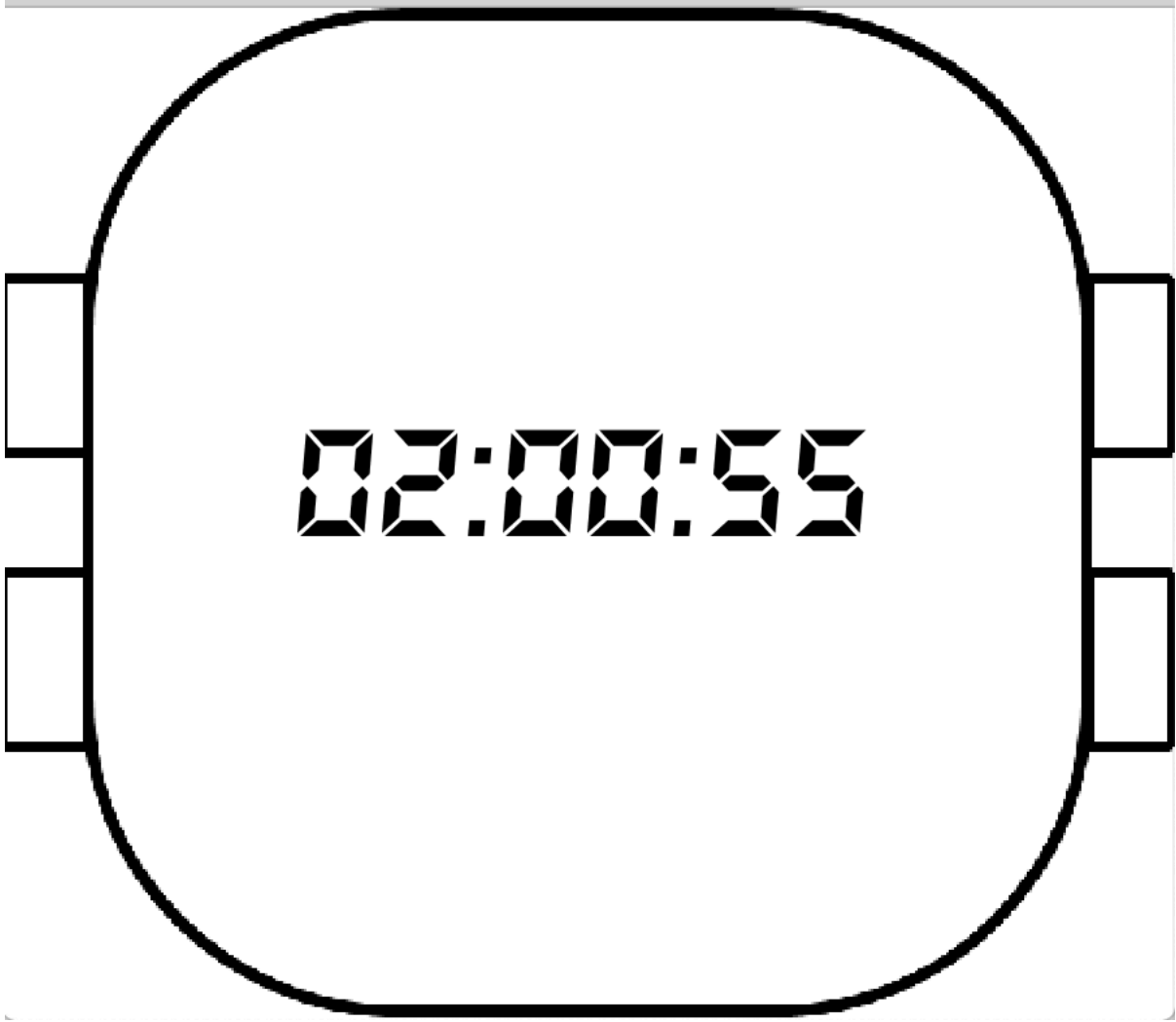
Stopwatch

Remind Watch

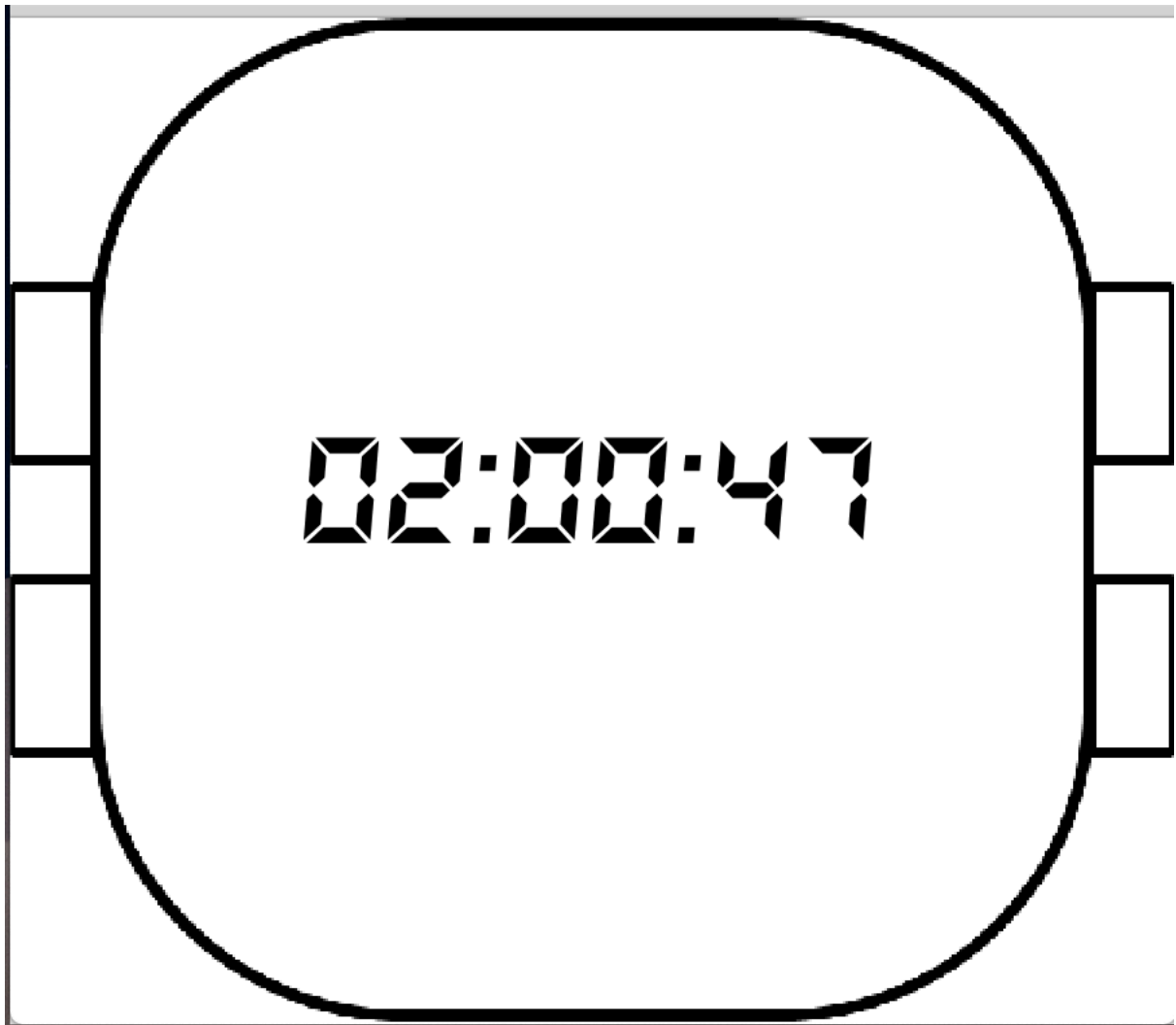




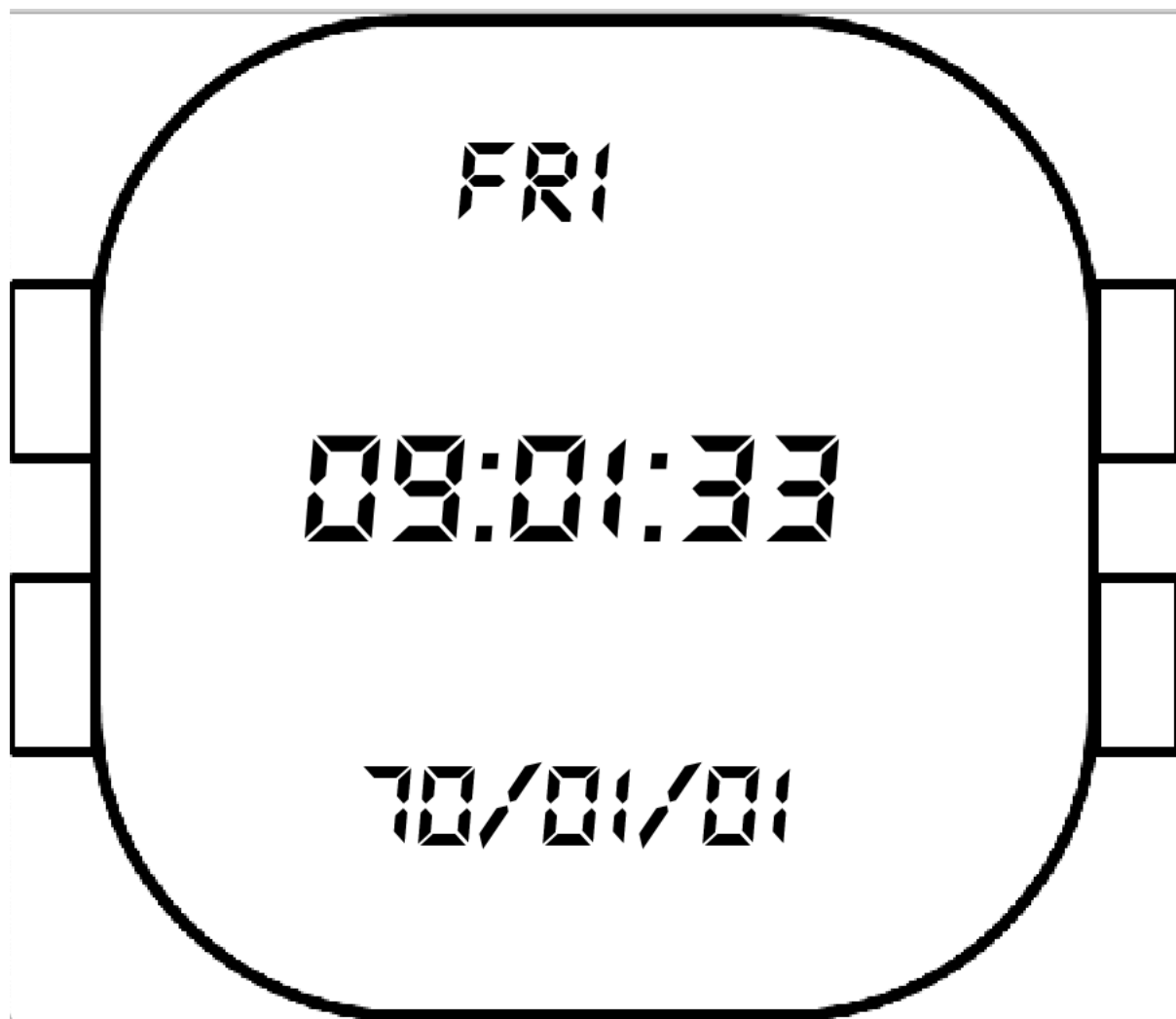
Timer



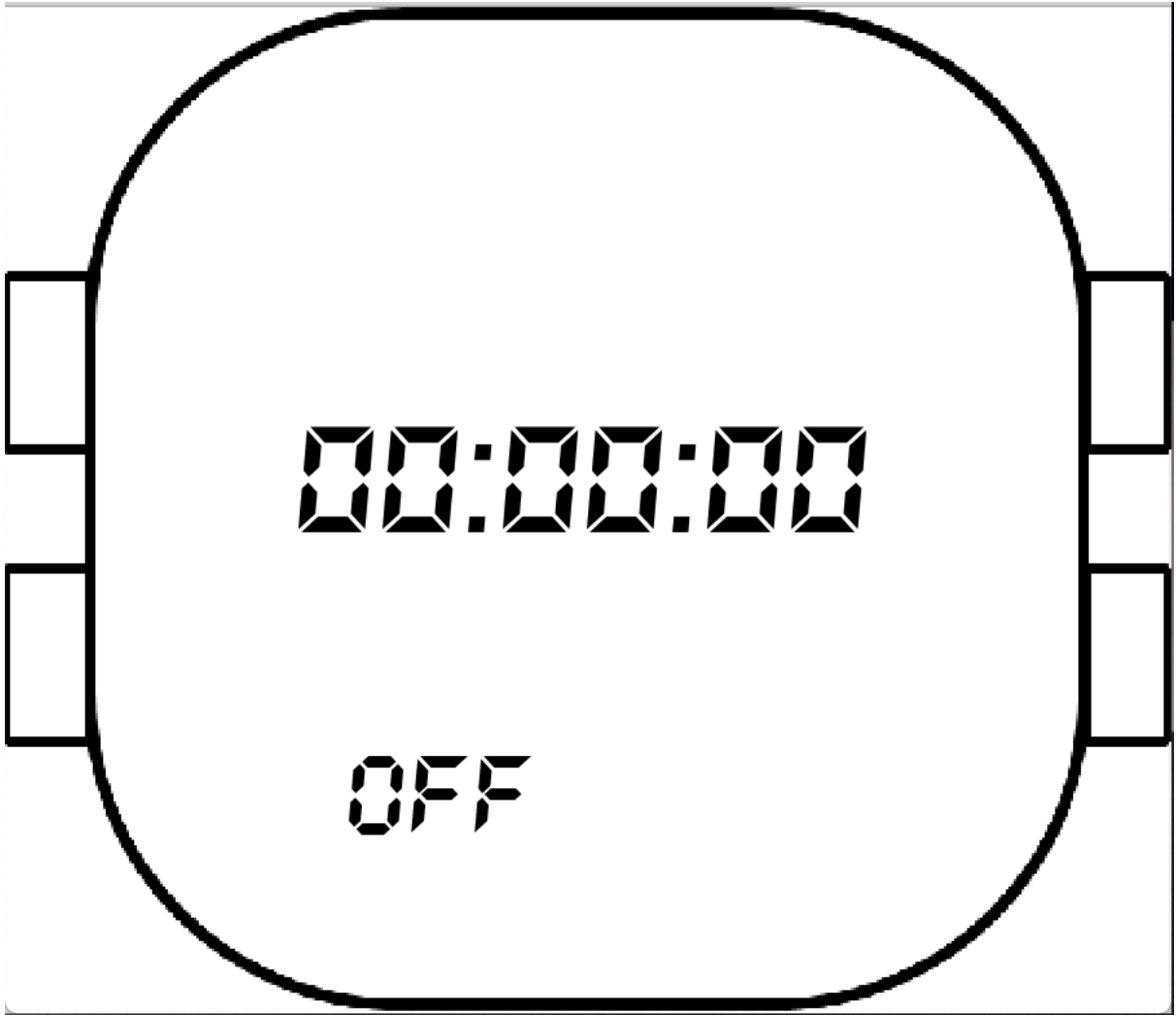
02:00:55



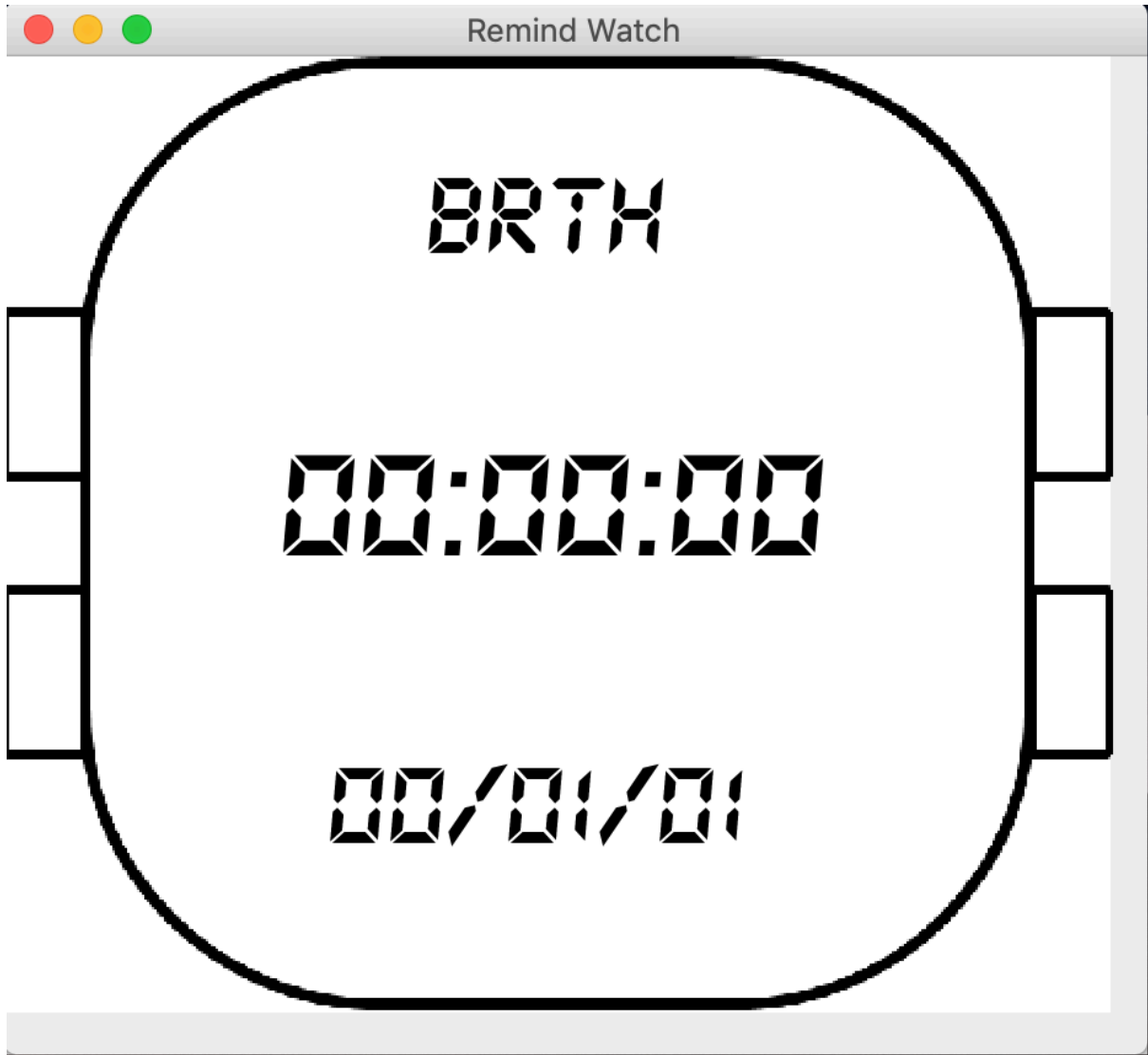
currentTime



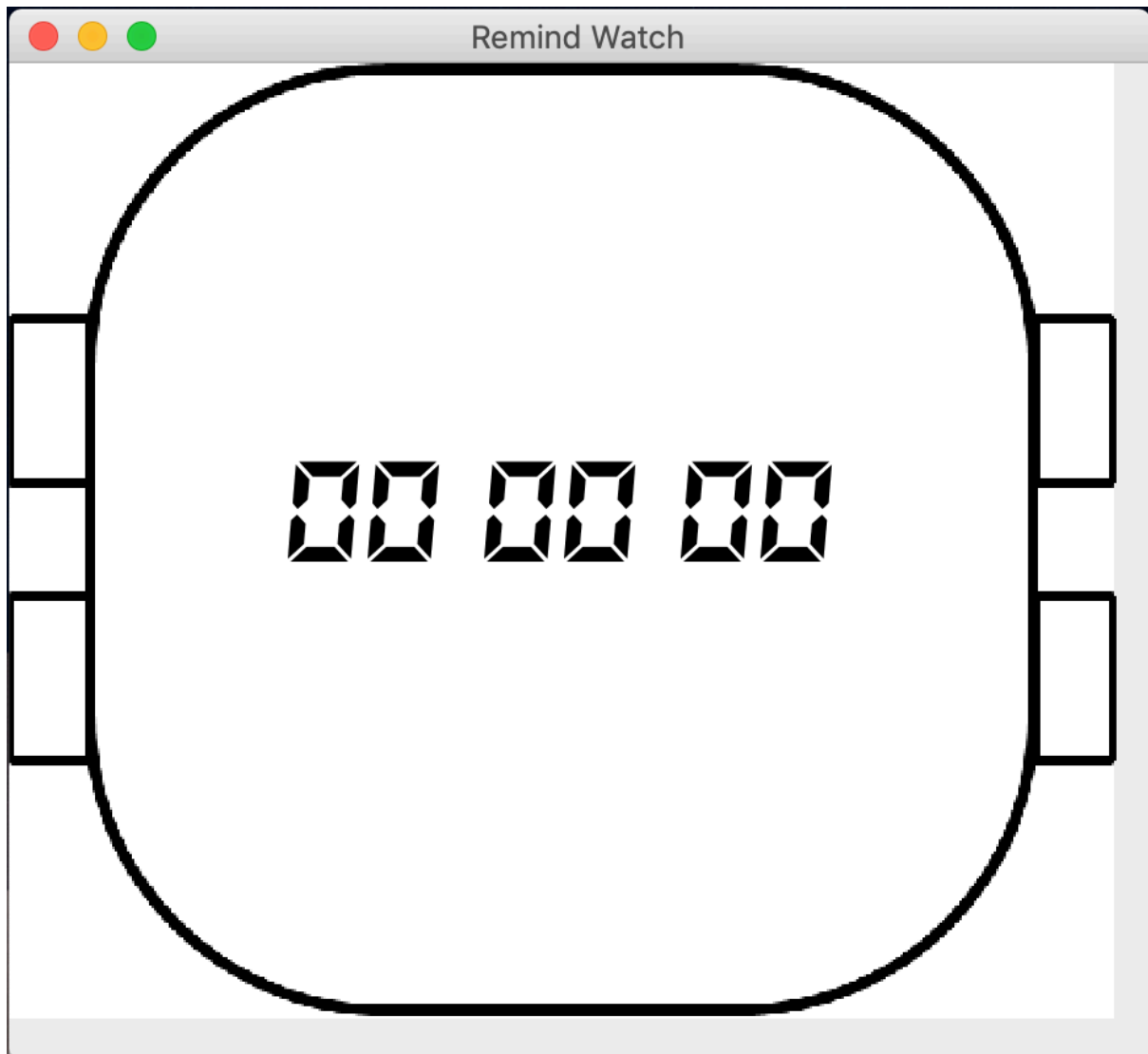
Alarm



Schedule



Expenditure



Activity 2055. Write Unit Test Code

```
@Test
public void test_increaseElement_minutes_10_times_0_to_10() {
    Alarm alarm = new Alarm();
    alarm.moveCursor();alarm.moveCursor();
    for(int i = 0; i<10; i++)
        alarm.increaseValue();
    assert(alarm.getData().get("minutes").equals(10));
}
```



```

@Test
public void test_getNextAlarmData_firstAlarm_minutesValue_is_not_equal_to_secondAlarm() {
    Alarm alarm = new Alarm();
    int a, b;
    a = alarm.getData().get("minutes");
    alarm.getNextAlarmData();
    alarm.moveCursor();
    alarm.moveCursor();
    alarm.increaseValue();
    b = alarm.getData().get("minutes");

    assert(a!=b);
}

```

```

@Test
public void test_after_deleteAlarm_three_times_getNextAlarmData_and_compare_minutes() {
    Alarm alarm = new Alarm();
    alarm.moveCursor();alarm.moveCursor();
    alarm.increaseValue();

    alarm.deleteAlarm();
    alarm.getNextAlarmData();alarm.getNextAlarmData();alarm.getNextAlarmData();

    assert(alarm.getData().get("minutes") != 1);
}

```

```

@Test
public void test_increase_Expenditure_Value() {
    Expenditure expenditure = new Expenditure();

    for(int i = 0; i<9; i++)
        expenditure.increaseValue();

    assert(expenditure.data.get("0").equals(9));

    expenditure.increaseValue();

    assert(expenditure.data.get("0").equals(0));
}

```

```
@Test
public void test_save_and_load_ExpenditureValue() {
    Expenditure expenditure = new Expenditure();
    for(int i = 0; i<9; i++)
        expenditure.increaseValue();

    expenditure.saveExpenditureValue();
    expenditure.increaseValue();

    assert(expenditure.data.get("0").equals(0));

    expenditure.loadExpenditureValue();

    assert(expenditure.data.get("0").equals(9));
}
```

```
@Test
public void test_increase_Schedule_value_59_plus_1() {
    Schedule schedule = new Schedule( msTimestamp: 0);

    for(int i = 0; i<59; i++)
        schedule.increaseValue();

    assert(schedule.data.get("minutes").equals(59));

    schedule.increaseValue();

    assert(schedule.data.get("minutes").equals(0));
}
```

```
@Test
public void test_Schedule_moveCursor(){
    Schedule schedule = new Schedule( msTimestamp: 0);

    schedule.moveCursor();
    schedule.increaseValue();

    assert(schedule.data.get("hours").equals(10));
}
```

```
@Test
public void test_Schedule_moveCursor(){
    Schedule schedule = new Schedule( msTimestamp: 0);

    schedule.moveCursor();
    schedule.increaseValue();

    assert(schedule.data.get("hours").equals(10));
}
```

```
@Test
public void saveScheduleValue() {
    Schedule schedule = new Schedule( msTimestamp: 0);

    for(int i = 0; i<5; i++)
        schedule.increaseValue();

    schedule.saveScheduleValue();

    assert(schedule.getData().get("minutes").equals(5));

    schedule.increaseValue();

    assert(schedule.dataKeep.loadData( name: "SCHEDULE", num: 0).get("minutes").equals(5));
}
```

```
@Test
public void startStopwatch() {
    Stopwatch stp = new Stopwatch();

    for(int i=0; i<20; i++)
        stp.increaseValue();
    stp.startStopwatch();
    assert(stp.getTimestamp() < 20);
}
```

```
@Test
public void pauseStopwatch() {
    Stopwatch stp = new Stopwatch();

    for(int i=0; i<20; i++)
        stp.increaseValue();
    stp.startStopwatch();

    try {
        Thread.sleep( millis: 1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    stp.pauseStopwatch();
    assert(stp.getTimestamp() > 17);
}
```

```
@Test
public void startTimer() {
    Timer timer = new Timer();

    for(int i=0; i<20; i++)
        timer.increaseValue();
    timer.startTimer();
    assert(timer.getTimestamp() < 20);
}
```

```
@Test
public void pauseTimer() {
    Timer timer = new Timer();

    for(int i=0; i<20; i++)
        timer.increaseValue();
    timer.startTimer();

    try {
        Thread.sleep( millis: 1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    timer.pauseTimer();
    assert(timer.getTimestamp() < 3);
}
```

```
@Test
public void test_print_CurrentTime_Value(){
    CurrentTime currentTime = new CurrentTime( msTimestamp: 0);
    currentTime.printCalendar();
    assert(currentTime.getData().get("year").equals(1970));
}
```

```

@Test
public void test_change_currentTime_Value(){
    CurrentTime currentTime = new CurrentTime( msTimestamp: 0);
    assert(currentTime.getData().get("minutes").equals(0));

    currentTime.moveCursor();
    currentTime.increaseValue();
    currentTime.increaseValue();
    assert(currentTime.getData().get("minutes").equals(2));
}

```

```

@Test
public void test_does_currentTime_work_per_second() {
    CurrentTime currentTime = new CurrentTime( msTimestamp: 0);

    currentTime.setData();
    currentTime.contTimeFlow();
    for(int i=0; i<4; i++) {
        try {
            Thread.sleep( millis: 1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    System.out.println(currentTime.getCurrentTimestamp());
}
}

```

Activity 2061. Unit Testing

▼	✘ AlarmTest (Controller)	12 ms
	✘ test_after_deleteAlarm_three_times_c	12 ms
	✔ test_increaseElement_minutes_10_tim	0 ms
	✔ test_getNextAlarmData_firstAlarm_mi	0 ms

▼	✔ ExpenditureTest (Controller)	5 ms
	✔ test_save_and_load_ExpenditureValue	5 ms
	✔ test_increase_Expenditure_Value	0 ms

✘	ScheduleTest (Controller)	73 ms
✔	test_increase_Schedule_value_59_plt	60 ms
✔	test_Schedule_moveCursor	0 ms
✘	saveScheduleValue	13 ms

▼	✔	StopwatchTest (Controller)	1 s 74 ms
	✔	pauseStopwatch	1 s 71 ms
	✔	startStopwatch	3 ms

▼	✔	TimerTest (Controller)	1 s 45 ms
	✔	startTimer	44 ms
	✔	pauseTimer	1 s 1 ms

✔	CurrentTimeTest (Controller)	61 ms	"C:\Program Files\Java\jdk-10.0.1\bin
✔	test_print_CurrentTime_Value	61 ms	CurrentTime 1970 / 0 / 1 9 : 0 : 0 Process finished with exit code 0

▼	✔	CurrentTimeTest (Controller)	4 s 71 ms	"C:\Program Files\Java\j
	✔	test_does_currentTime_work_per_	4 s 71 ms	997 1988 2992 3990

Activity 2063. System Testing

TEST NUMBER	REF #	TEST CASE	DESCRIPTION	USE CASE	P/F
1	r 1.1	디스플레이 전환 시험	디스플레이가 전환되는지 test	Switch Display	Fail
2	r 2.1	버튼 기능 전환 시험	각 기능이 변할 때 버튼의 기능이 변하는지 test	Change Button's function	Fail
3	r 3.1	현재 시간 표시 시험	현재 시간이 화면에 표시되는지 test	Current time	Fail
4	r 3.2	현재 시간 변경 시험	현재 시간 수정이 이루어지는지 test	Current time	Fail
5	R 3.3	현재 시간 갱신 시험	현재 시간이 1초 단위로 바뀌는지 test	Current time	Fail
6	r 4.1	타이머 설정 시험	타이머 설정이 이루어지는지 test	Timer	Pass
7	r 4.2	타이머 초기화 시험	타이머가 초기화 되는지 test	Timer	Pass
8	r 4.3	타이머 시간 시작 시험	타이머가 현재 타이머 시간으로부터 시작되는지 test	Timer	Pass
9	r 4.4	타이머 일시정지 시험	타이머가 일시정지 되는지 test	Timer	Pass
10	r 5.1	스톱워치 동작 시험	스톱워치가 현재 스톱워치 시간으로부터 동작되는지 test	stopwatch	Pass
11	r 5.2	스톱워치 일시정지 시험	스톱워치가 일시 정지 되는지 test	Stopwatch	Pass
12	r 5.3	스톱워치 초기화 시험	스톱워치가 초기화가 이루어지는지 test	Stopwatch	Pass
13	R 6.1	알람 설정 시험	알람 설정이 이루어지는지 test	Alarm	Pass
14	r 6.2	알람 설정 삭제 시험	알람 설정이 삭제되는지 test	Alarm	Fail
15	R 6.3	알람 작동 시험	알람이 작동하는지 test	Alarm	Pass
16	R 7.1	버저 시험	버저가 울리는지 test	Buzz	Fail
17	R 7.2	버저 중지 시험	버저가 중지되는지 test	Buzz	Fail
18	R 8.1	스케줄 등록 시험	스케줄이 등록되는지 test	Schedule	Fail
19	r 8.2	스케줄 삭제 시험	스케줄이 삭제되는지 test	Schedule	Pass
20	R 8.3	스케줄 초기화 시험	모든 스케줄이 삭제되는지 test	Schedule	Pass
21	R 8.4	스케줄 알람 시험	스케줄 시간에 알람이 작동하는지 test	Schedule	Pass
22	r 9.1	지출액 추가 시험	지출액이 추가 되는지 test	Expenditure	Pass
23	r 9.2	지출액 초기화 시험	이번달 지출액이 0원으로 초기화 되는지 test	Expenditure	Fail
24	r 9.3	지출액 누적 시험	누적된 지출액이 보이는지 test	Expenditure	Fail
25	r 10.1	데이터 저장 시험	데이터를 저장하는지 test	Data	Fail
26	R 10.2	데이터 로드 시험	데이터를 불러오는지 test	Data	Fail
27	R 10.3	데이터 삭제 시험	데이터가 삭제되는지 test	Data	Fail
28	R 11.1	기능 4개 선택 시험	기능 4개를 선택되는지 test	Select Function	Pass

Activity 2067. Design Traceability Analysis

