



# 한판하시계

---

객체지향개발방법론 6팀

201611279 이동준

201611280 이동훈

201611298 정태민

201611266 성시진

# Bugs & Errors

A6 DWS

검색:  A6 DWS

+ 개요 작업내역 **일감** 소요 시간 Gantt 차트 달력 뉴스 문서 위키 파일 설정

## 일감

새 일감만들기 ...

검색조건

상태

진행중

검색조건 추가

옵션

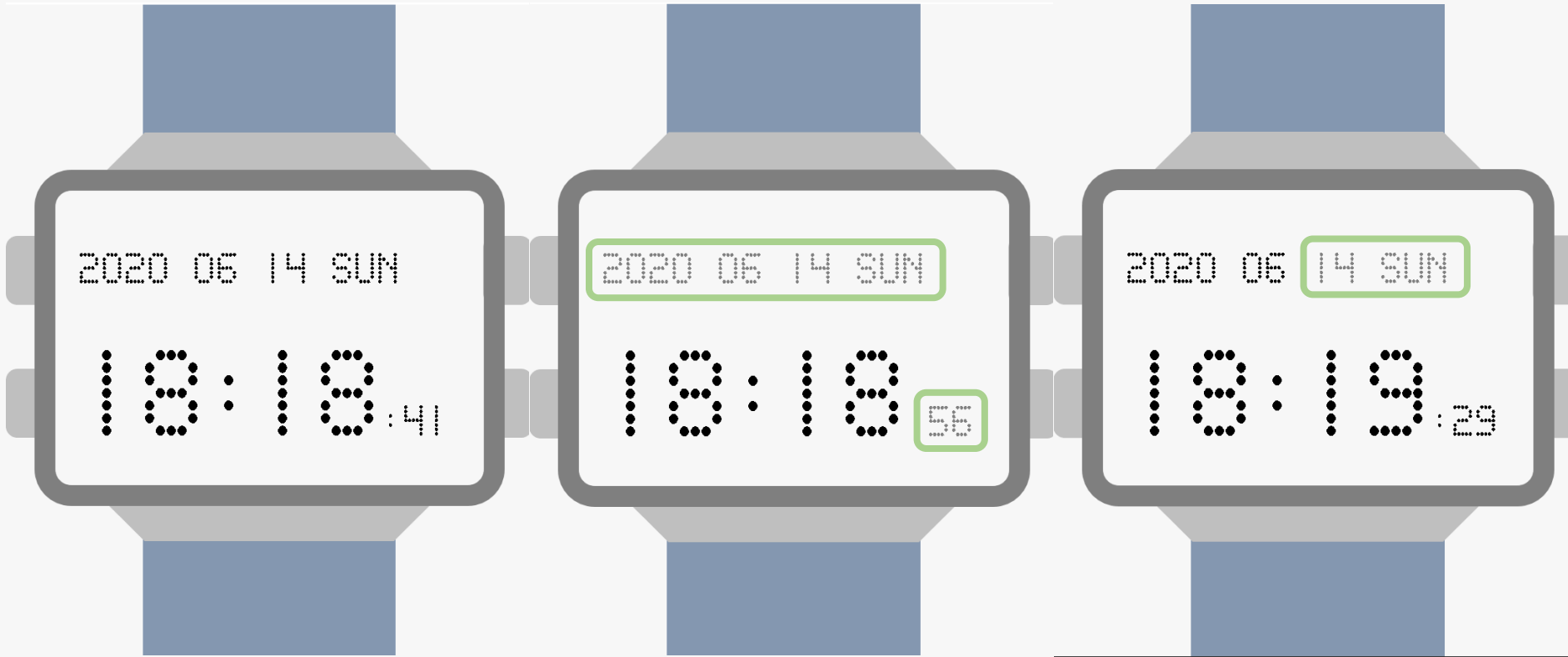
적용 지우기 저장

<input type="checkbox"/>	#	유형	상태	우선순위	제목	담당자	변경	
<input type="checkbox"/>	30	Bug	Resolved	Normal	[Game] 무작위 버튼을 눌렀을 때 에러가 나지 않는지 확인한다.	태민 정	2020/06/12 06:01	...
<input type="checkbox"/>	29	Bug	Resolved	Normal	[Alarm] 설정한 시간에 beep이 울리는지 확인한다.	동준 이	2020/06/13 15:27	...
<input type="checkbox"/>	28	Bug	Resolved	Normal	[Game] 지정한 버튼을 눌렀을 때만 게임이 시작되는지 확인한다.	태민 정	2020/06/12 06:03	...
<input type="checkbox"/>	27	Bug	Resolved	Normal	[Alarm] 알람 설정 시 시간단위가 깜빡이는지 확인한다.	태민 정	2020/06/13 14:51	...
<input type="checkbox"/>	26	Bug	Resolved	Normal	[WorldTime] Summertime를 적용할 필요가 없는 나라는 시간 변동이 있는지 검사한다.	시진 성	2020/06/13 06:47	...
<input type="checkbox"/>	25	Bug	Resolved	Normal	[Alarm] 무작위 버튼을 눌렀을 때 잘 작동하는지 확인한다.	태민 정	2020/06/12 06:06	...
<input type="checkbox"/>	24	Bug	Resolved	Normal	[Timer] 설정한 시간에 beep이 울리는지 확인한다.	태민 정	2020/06/13 05:55	...
<input type="checkbox"/>	23	Bug	Resolved	Normal	[Stopwatch] Lap Time이 되는지 확인한다.	태민 정	2020/06/13 07:52	...
<input type="checkbox"/>	22	Bug	Resolved	Normal	[Timer] 타이머 설정시 시간 단위가 깜빡이는지 확인한다	동준 이	2020/06/13 14:49	...
<input type="checkbox"/>	21	Bug	Resolved	Normal	[Timer] 무작위 버튼을 눌렀을 때 에러가 나지 않는지 확인한다.	태민 정	2020/06/12 06:09	...
<input type="checkbox"/>	20	Bug	Resolved	Normal	[Stopwatch] 모든 버튼을 눌렀을 때 잘 작동하는지 확인한다.	태민 정	2020/06/13 06:17	...
<input type="checkbox"/>	19	Bug	Resolved	Normal	[Timer] 모든 버튼을 눌렀을 때 잘 작동하는지 확인한다.	태민 정	2020/06/12 06:09	...
<input type="checkbox"/>	18	Bug	Resolved	Normal	[Timekeeping] 시간이 정상적으로 흘러가는지 확인한다.	동준 이	2020/06/13 15:26	...
<input type="checkbox"/>	17	Bug	Resolved	Normal	[Timekeeping] 시간 설정시 시간 단위가 깜빡이는지 확인한다.	동준 이	2020/06/13 07:19	...



# Bugs & Errors

#17



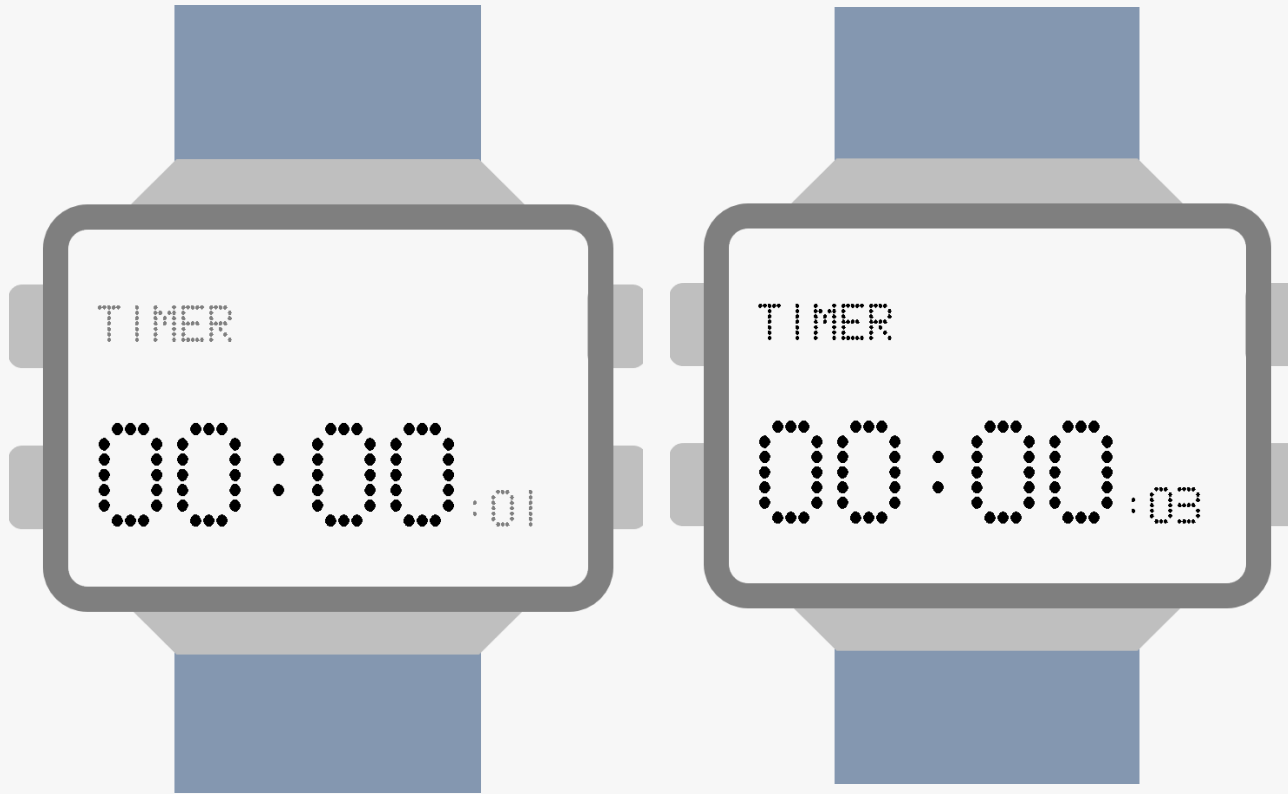
[TimeKeeping] 시간 설정시 시간 단위가 깜박이는지 확인한다

- Time Keeping 에서 시간 설정시에 시간 단위가 깜박이지 않는다



# Bugs & Errors

#22



[Timer] 타이머 설정시 시간 단위가 깜빡이는지 확인한다

- 시간단위 깜빡이지 않는다.

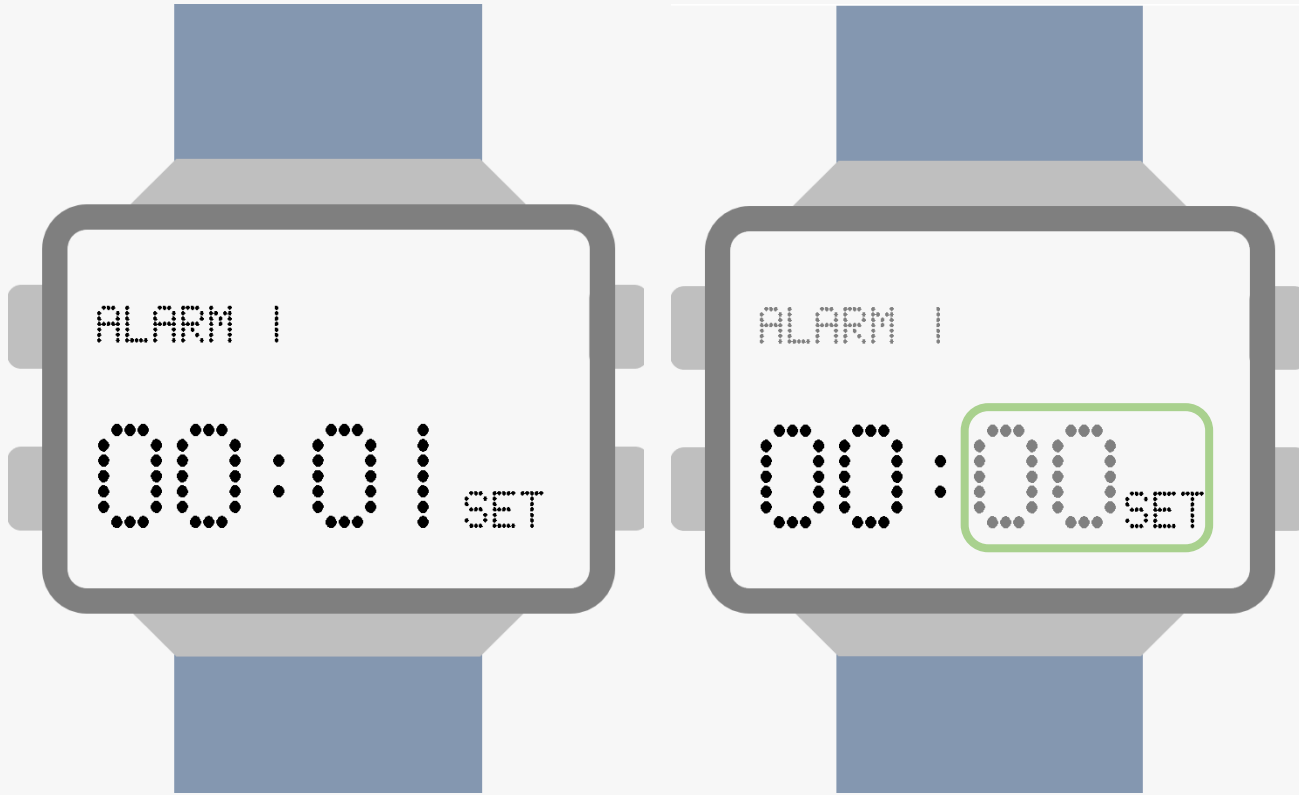
```
if (data[3].equals("X"))
    isSet = false;

if (isGray && !data[3].equals("X")){
    isSet = true;
    g.setColor(Color.GRAY);
}else
    g.setColor(Color.BLACK);

g.setFont(top);
if(curMode == 0){//time keeping mode **** ** ** *** 14
    g.drawString(data[0], x 100, y 388);
    if(isGray) {
        if (data[3].equals("6")) {
            g.drawString(data[0].substring(0, 5), x 100, y 388);
            g.setColor(Color.BLACK);
            g.drawString(String.format("%14s",data[0].substring(5, 14)), x 100, y 388);
        }
        else if (data[3].equals("5")) {
            g.drawString(String.format("%8s",data[0].substring(5, 8)), x 100, y 388);
            g.setColor(Color.BLACK);
            g.drawString(String.format("%5s",data[0].substring(0, 5)), x 100, y 388);
            g.drawString(String.format("%14s",data[0].substring(8, 14)), x 100, y 388);
        }
        else if (data[3].equals("4")) {
            g.drawString(String.format("%14s",data[0].substring(8, 14)), x 100, y 388);
            g.setColor(Color.BLACK);
            g.drawString(String.format("%8s",data[0].substring(0, 8)), x 100, y 388);
        }
    }
}
else // other mode
    g.drawString(data[0], x 100, y 388);
```

# Bugs & Errors

#27



[Alarm] 알람 설정 시 시간단위가 깜빡이는지 확인한다.

- 시간단위가 깜빡이지 않는다.

```
if (isGray && data[3].equals("3")){
    isSet = true;
    g.setColor(Color.GRAY);
} else
    g.setColor(Color.BLACK);

g.setFont(main);
g.drawString(data[1].substring(0, 2), x: 95, y: 600);

g.setColor(Color.BLACK);
g.drawString(data[1].substring(2, 3), x: 265, y: 600); // :

if (isGray && data[3].equals("2")){
    isSet = true;
    g.setColor(Color.GRAY);
} else
    g.setColor(Color.BLACK);

g.drawString(data[1].substring(3, 5), x: 340, y: 600);

if (isGray && data[3].equals("1")){
    isSet = true;
    g.setColor(Color.GRAY);
} else
    g.setColor(Color.BLACK);

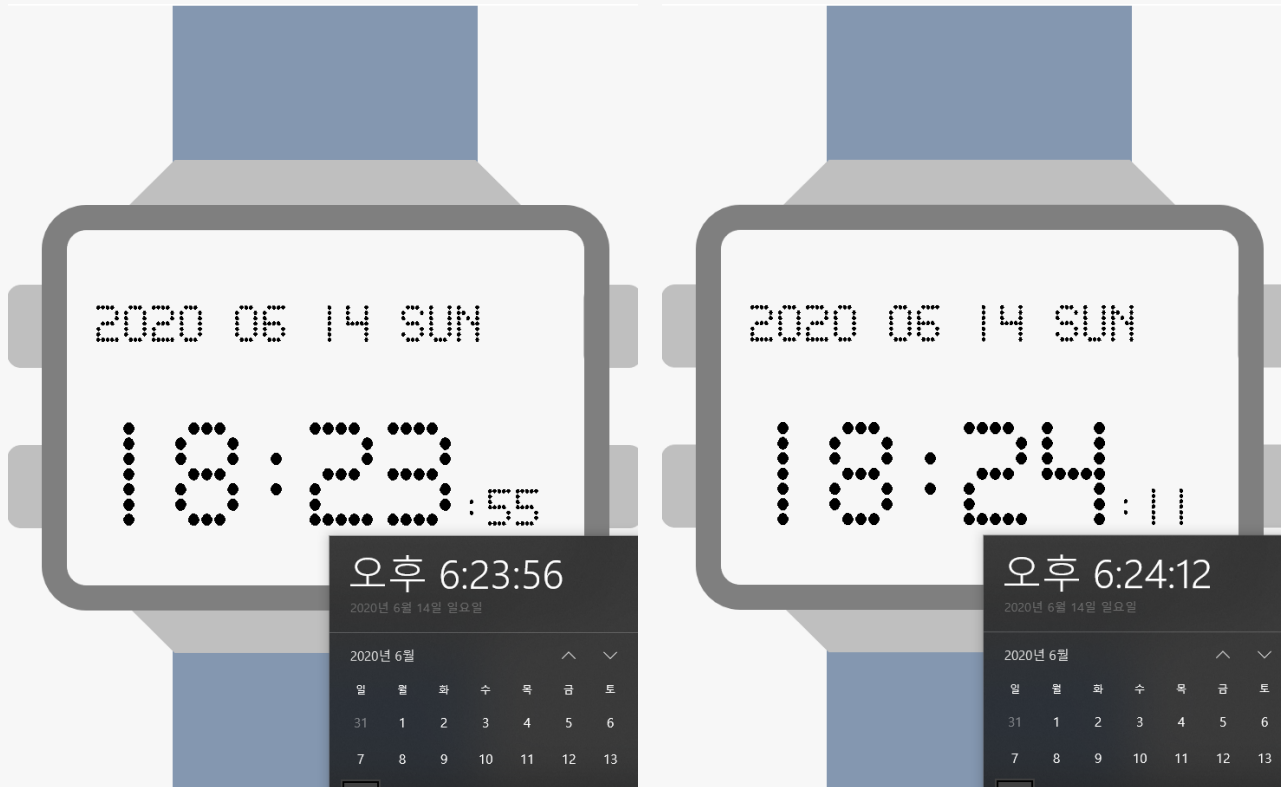
g.setFont(sub);
g.drawString(data[2], x: 520, y: 600);

if (count / 100 == 1) {
    isGray = !isGray;
    count = 0;
}
```



# Bugs & Errors

#18



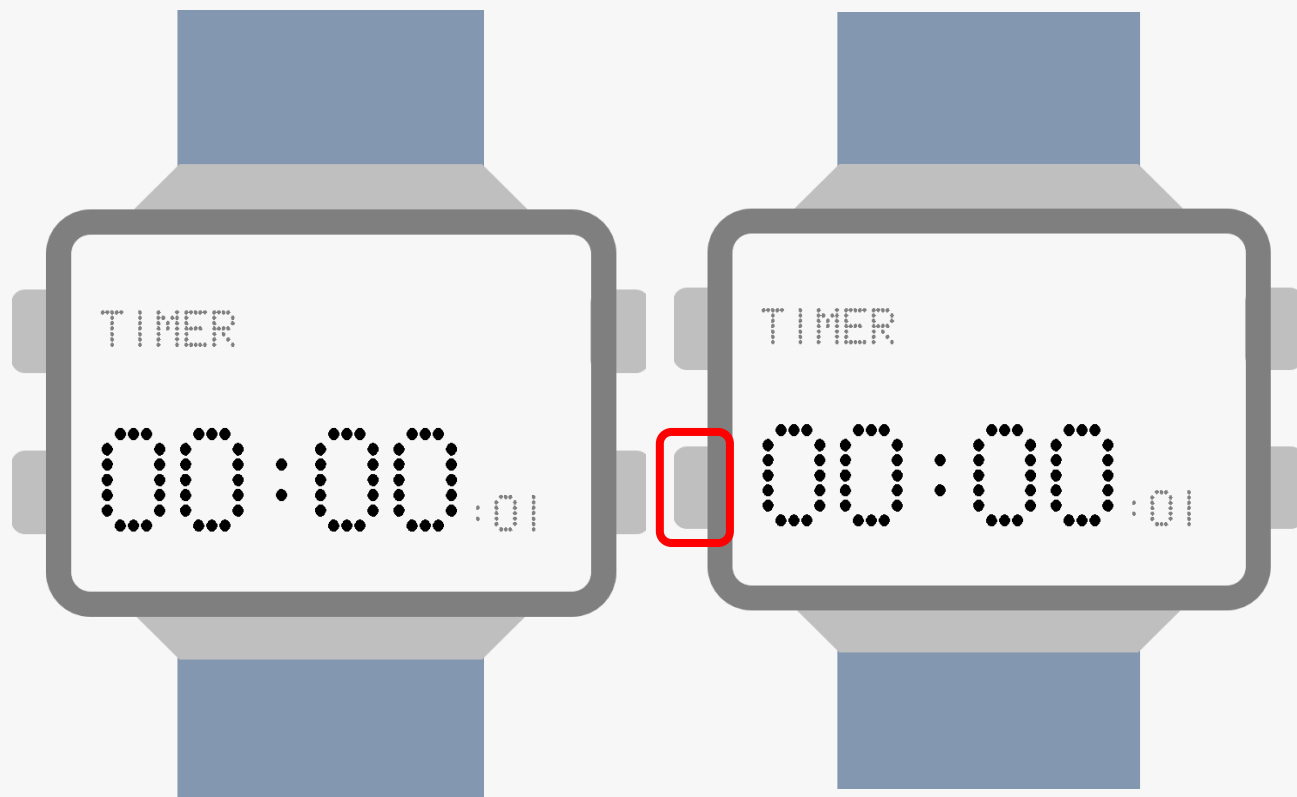
[TimeKeeping] 시간이 정상적으로 흘러가는지 확인한다

- Time Keeping 모드에서 시간이 1초에 2초씩 증가한다

```
public int updateTime() {  
    this.curTime.add(Calendar.MILLISECOND, amount: 10);  
}  
  
@Override  
public void run() {  
    while(true){  
        Watch.update();  
        repaint();  
        timeOut += 10; // plus 10 milliseconds  
        if (timeOut == 10000){ // TimeOut  
            timeOut = 0;  
            Watch.returnTimeMode();  
        }  
        try{  
            Thread.sleep( millis: 10);  
        }catch (InterruptedException e){  
            e.printStackTrace();  
        }  
    }  
}
```

# Bugs & Errors

#19

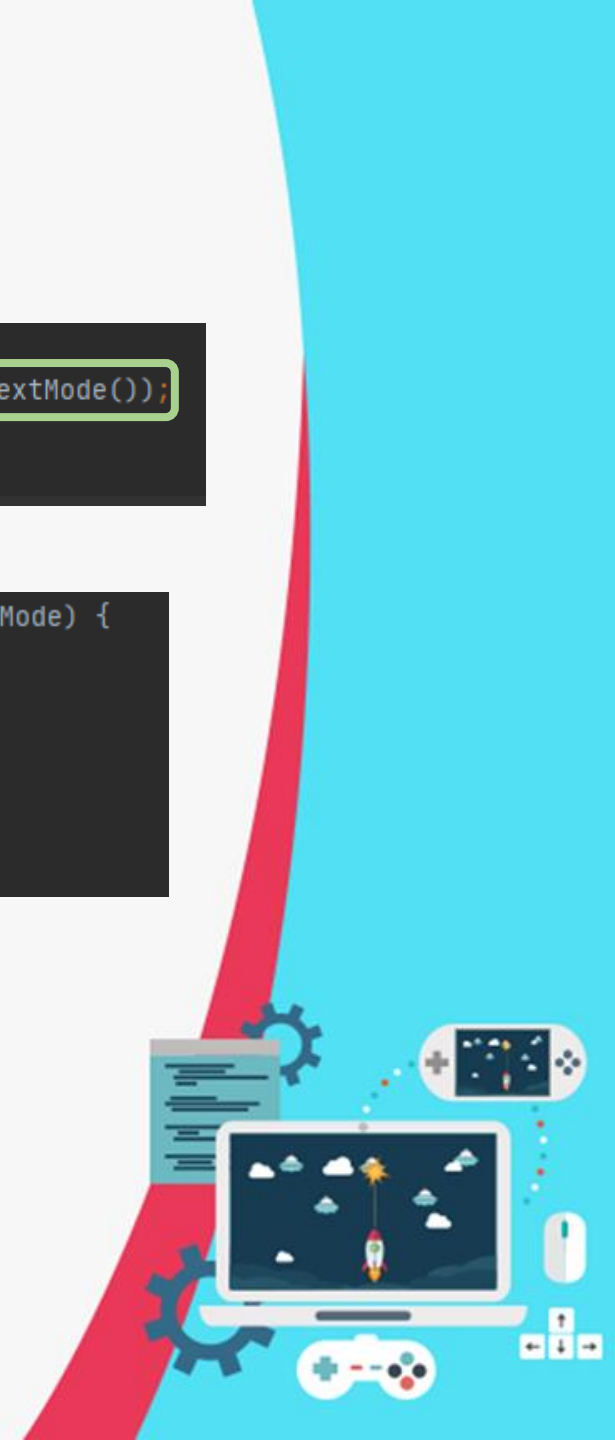


```
private void APressed(){  
    Watch.changeToMode(Watch.nextMode());  
    Watch.APressed();  
}
```

```
public void changeToMode(char Mode) {  
    if(game.gameState == 1)  
        return;  
    if(isSet)  
        return;  
    this.curMode = Mode;  
}
```

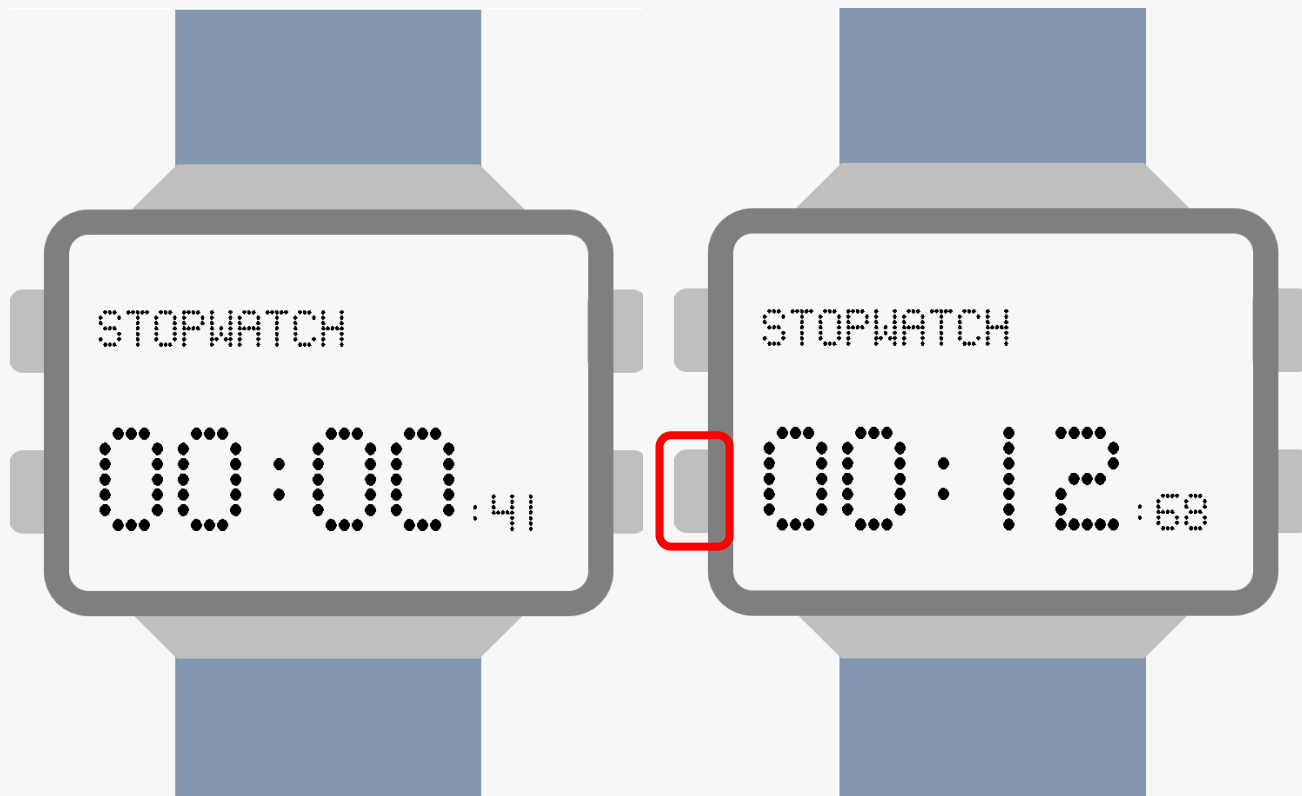
[Timer] 모든 버튼을 눌렀을 때 잘 작동하는지 확인한다.

- 설정 모드에서도 모드 변경이 이루어진다



# Bugs & Errors

#23

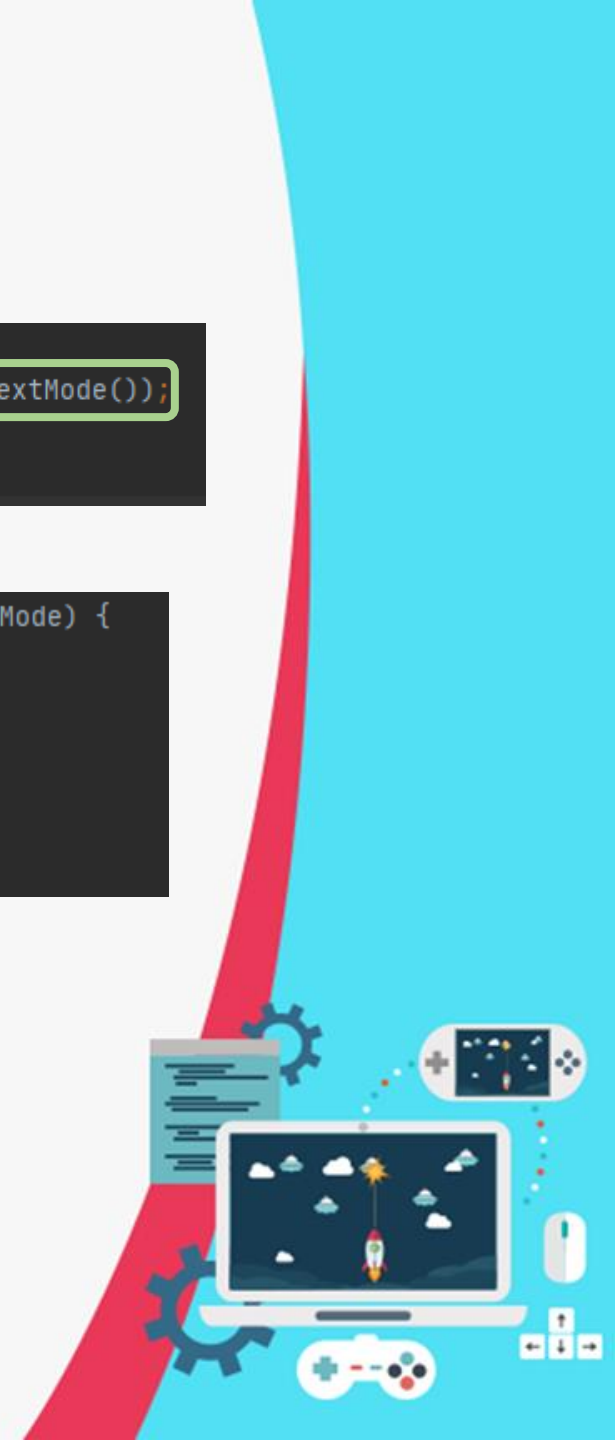


```
private void APressed(){  
    Watch.changeToMode(Watch.nextMode());  
    Watch.APressed();  
}
```

```
public void changeToMode(char Mode) {  
    if(game.gameState == 1)  
        return;  
    if(isSet)  
        return;  
    this.curMode = Mode;  
}
```

[Stopwatch] 모든 버튼을 눌렀을 때 잘 작동하는지 확인한다.

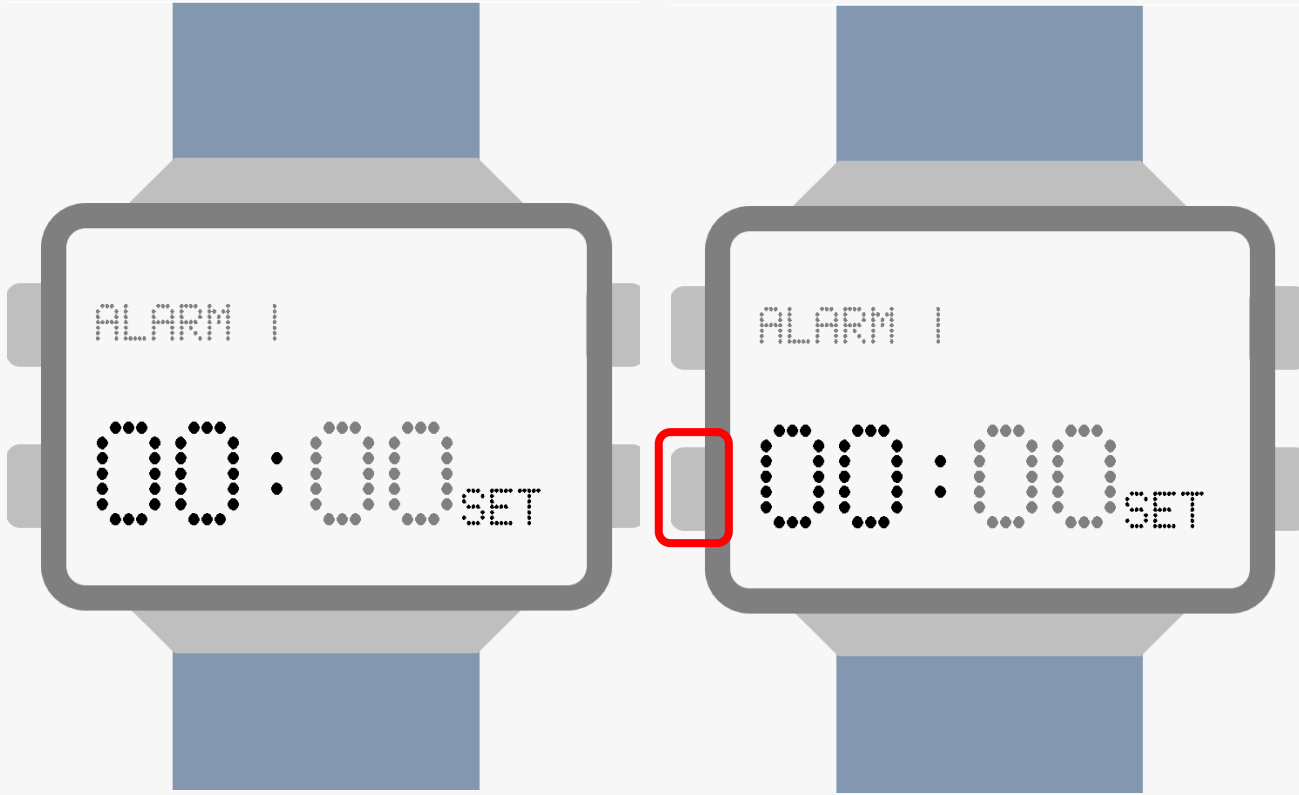
- stopwatch가 흘러가는 중에도 모드 변경이 된다.





# Bugs & Errors

#25

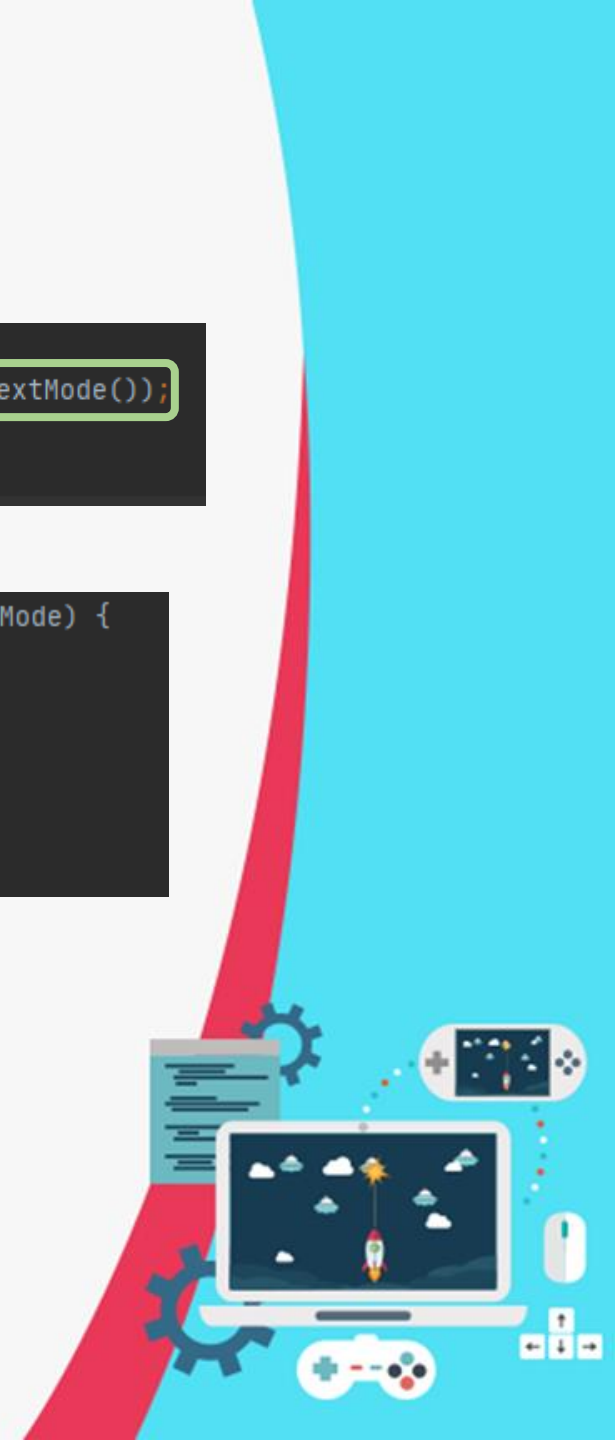


```
private void APressed(){  
    Watch.changeToMode(Watch.nextMode());  
    Watch.APressed();  
}
```

```
public void changeToMode(char Mode) {  
    if(game.gameState == 1)  
        return;  
    if(isSet)  
        return;  
    this.curMode = Mode;  
}
```

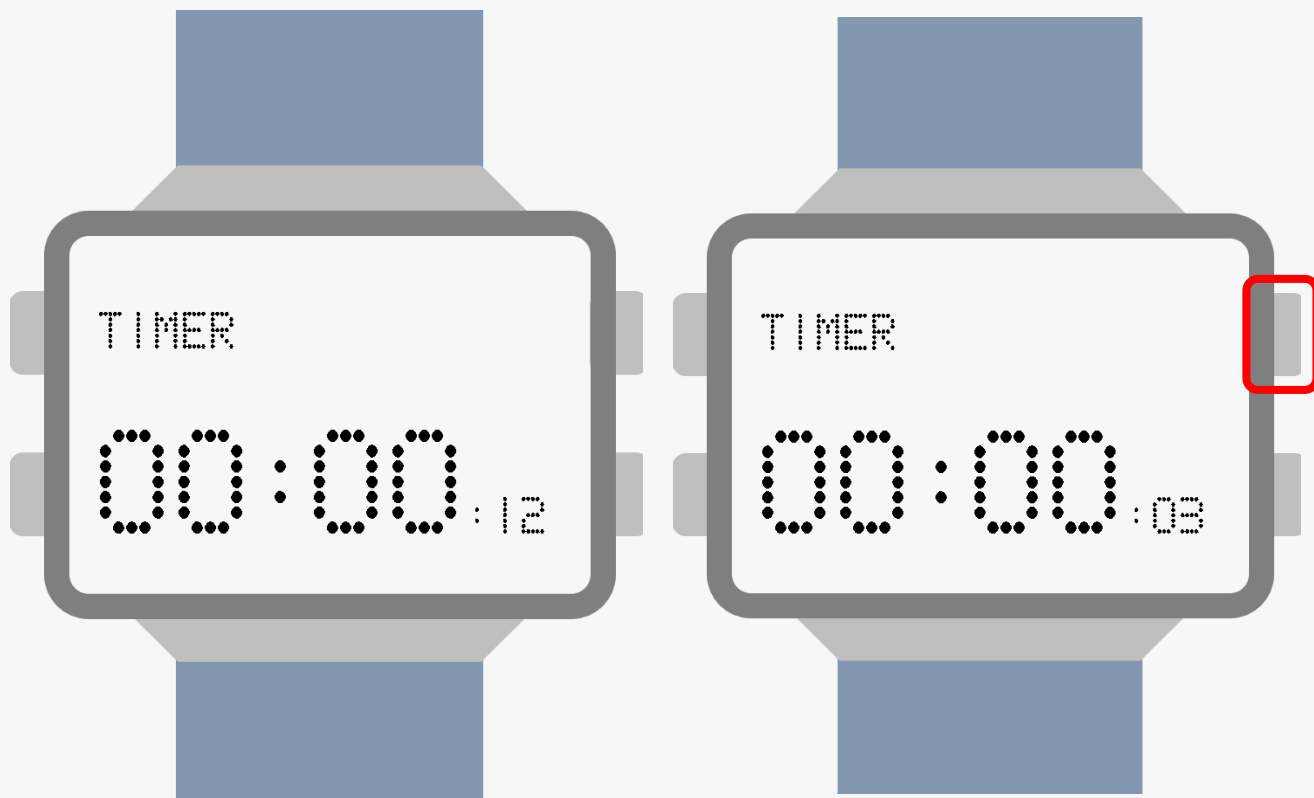
[Alarm] 무작위 버튼을 눌렀을 때 잘 작동하는지 확인한다.

- 설정 화면에서도 모드 변경이 이루어짐.



# Bugs & Errors

#21



```
@Override
public void WPressed(boolean Longpress) { // c
    if(Longpress && !isSetTimer){
        isSetTimer = true;
        timerUnit = 0;
        return;
    }

    if (isSetTimer){
        isSetTimer = false;
        return;
    }
    if (isPaused)
        resetTimer();
}
```

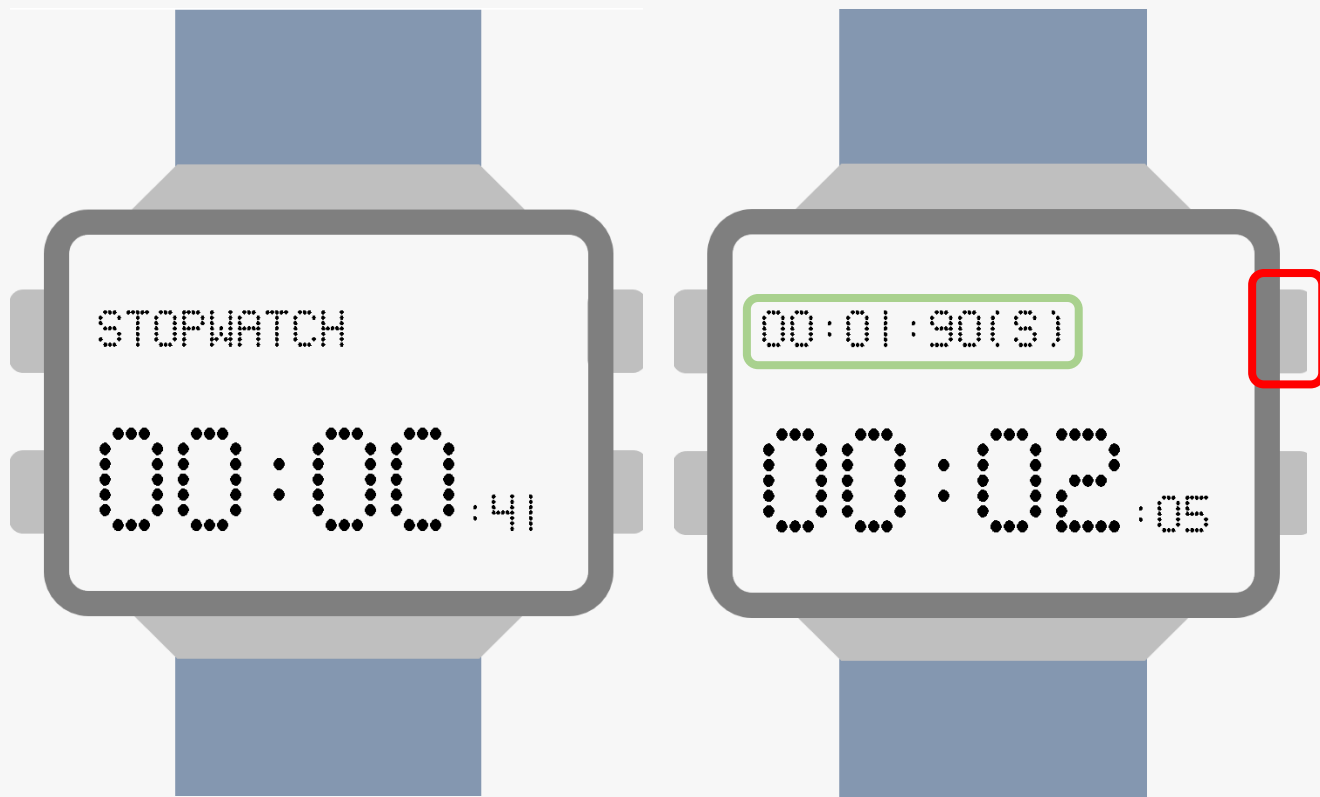
[Timer] 무작위 버튼을 눌렀을 때 에러가 나지 않는지 확인한다.

- Timer 시간이 흐르고 있을 때 reset 누르면 에러가 발생한다



# Bugs & Errors

#23



[Stopwatch] Lap Time이 되는지 확인한다.

- Lap time 기능이 작동하지 않는다.

```
public String[] requestStopwTime() {
    String[] stopwBufferArr = new String[4];
    Calendar tempStopwTime = (Calendar) stopwTime.clone();

    StringBuffer nameBuffer = new StringBuffer(); // stopwatch
    StringBuffer stopwBuffer = new StringBuffer(); // minute:sec
    StringBuffer csecBuffer = new StringBuffer(); // :csec

    nameBuffer.append("Stopwatch");
    stopwBufferArr[0] = nameBuffer.toString();

    stopwBuffer.append(tempStopwTime.get(Calendar.MINUTE) < 10 ? "0" : "");
    stopwBuffer.append(tempStopwTime.get(Calendar.MINUTE));
    stopwBuffer.append(":");
    stopwBuffer.append(tempStopwTime.get(Calendar.SECOND) < 10 ? "0" : "");
    stopwBuffer.append(tempStopwTime.get(Calendar.SECOND));

    stopwBufferArr[1] = stopwBuffer.toString();

    csecBuffer.append(":");
    csecBuffer.append(tempStopwTime.get(Calendar.MILLISECOND) < 100 ? "0" : "");
    csecBuffer.append(tempStopwTime.get(Calendar.MILLISECOND) / 10);

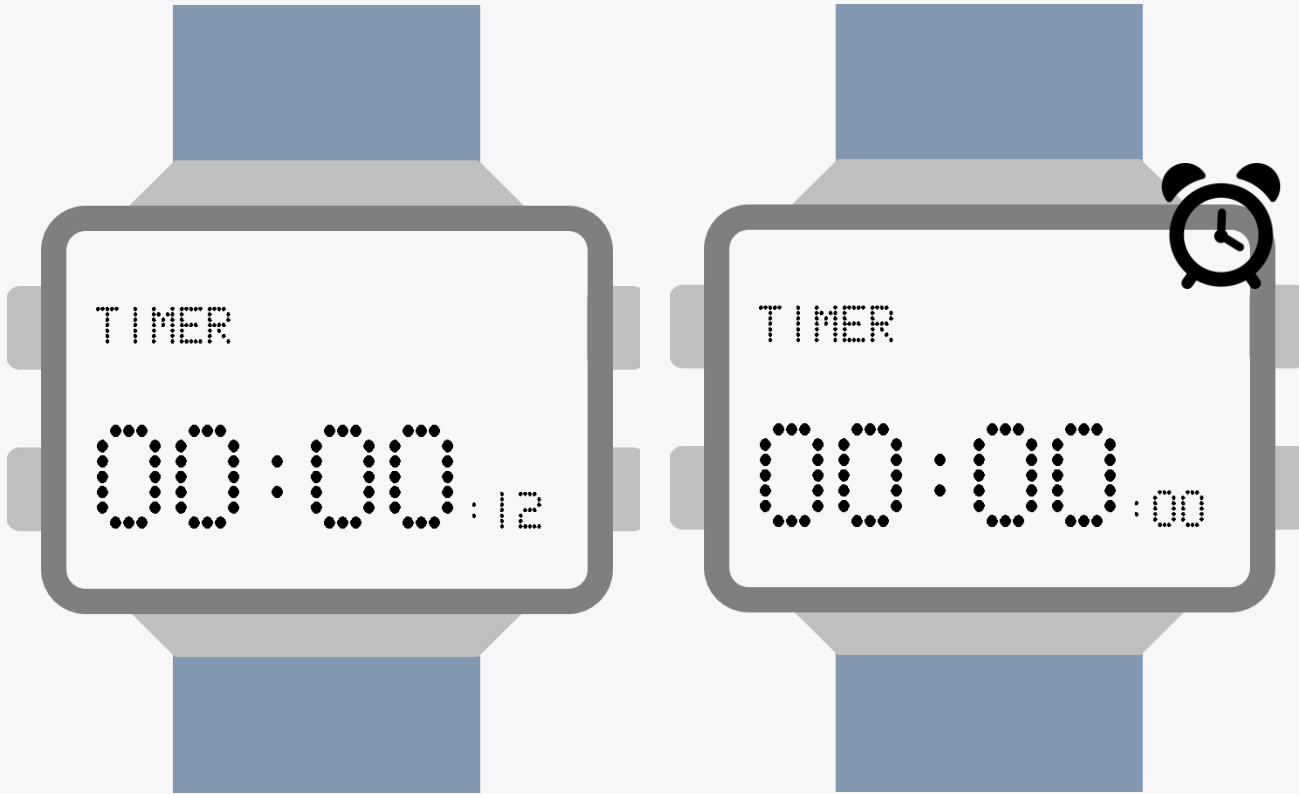
    stopwBufferArr[2] = csecBuffer.toString();

    stopwBufferArr[3] = "X";

    if(isSplit)
        stopwBufferArr[0] = requestSplit();
    return stopwBufferArr;
}
```

# Bugs & Errors

#24



[Timer]설정된 시간에 beep이 울리는지 확인한다.

- beep이 울리지 않는다.

```
public class Buzzer {
    private AudioClip beepSound;

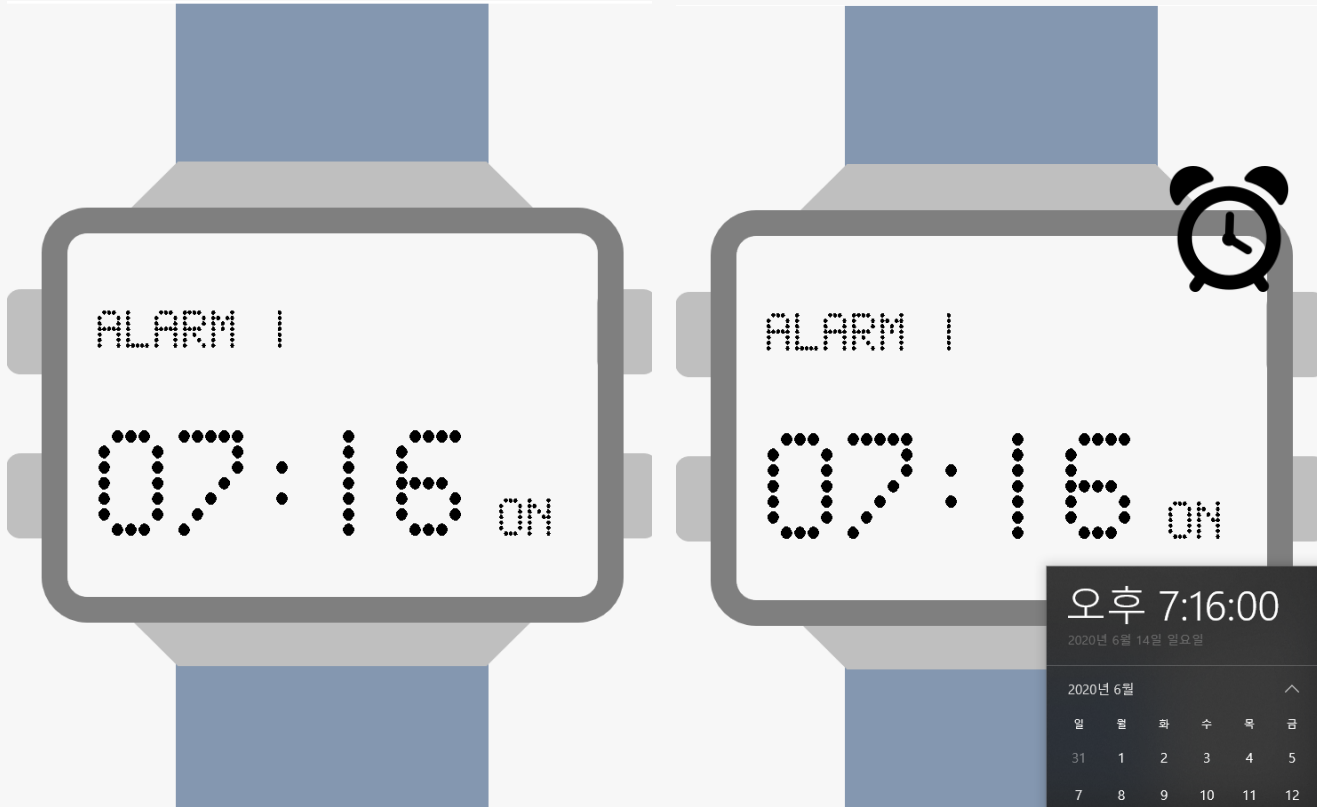
    public Buzzer() {
        try {
            beepSound = Applet.newAudioClip(
                Resource.class.getClassLoader().
                    getResource("resources/beep.wav"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void beep() { beepSound.play(); }
}
```



# Bugs & Errors

#29



```
public class Buzzer {  
    private AudioClip beepSound;  
  
    public Buzzer() {  
        try {  
            beepSound = Applet.newAudioClip(  
                Resource.class.getClassLoader().  
                    getResource("resources/beep.wav"));  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public void beep() { beepSound.play(); }  
}
```

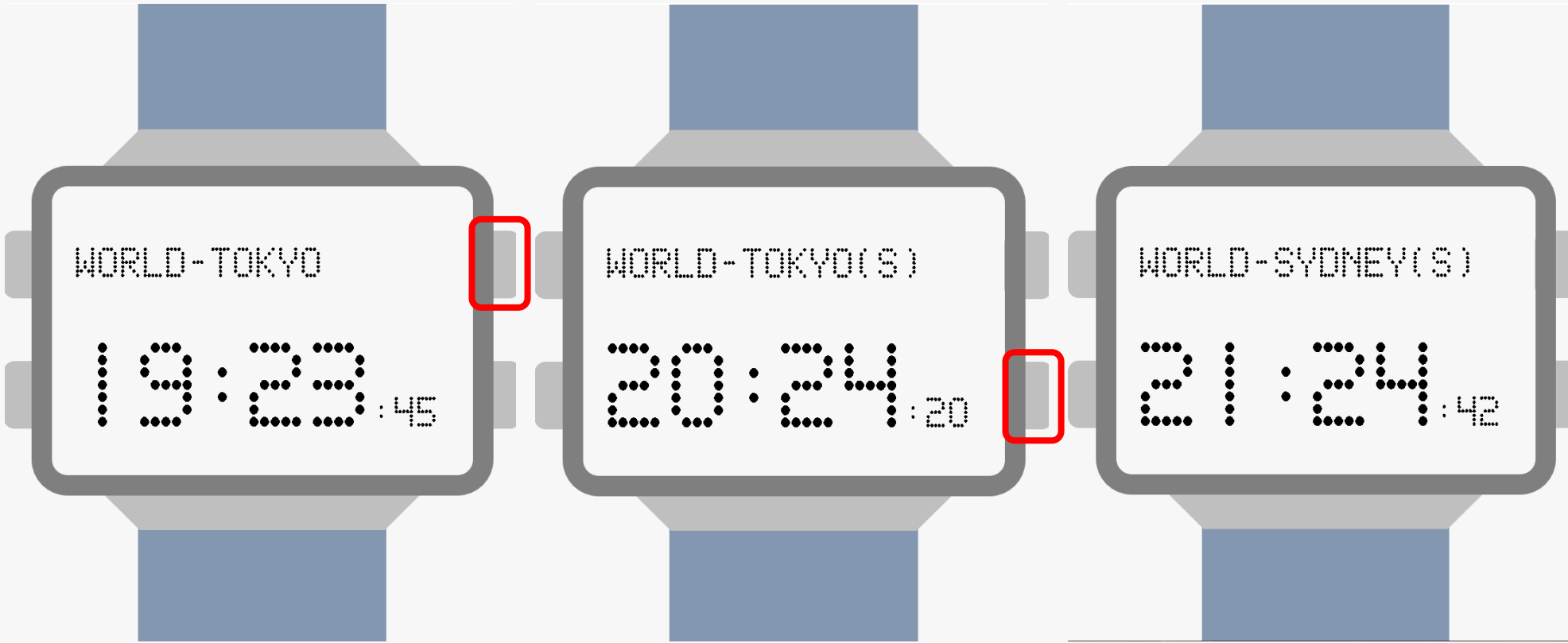
[Alarm] 설정한 시간에 beep이 울리는지 확인한다.

- beep이 울리지 않는다.



# Bugs & Errors

#26



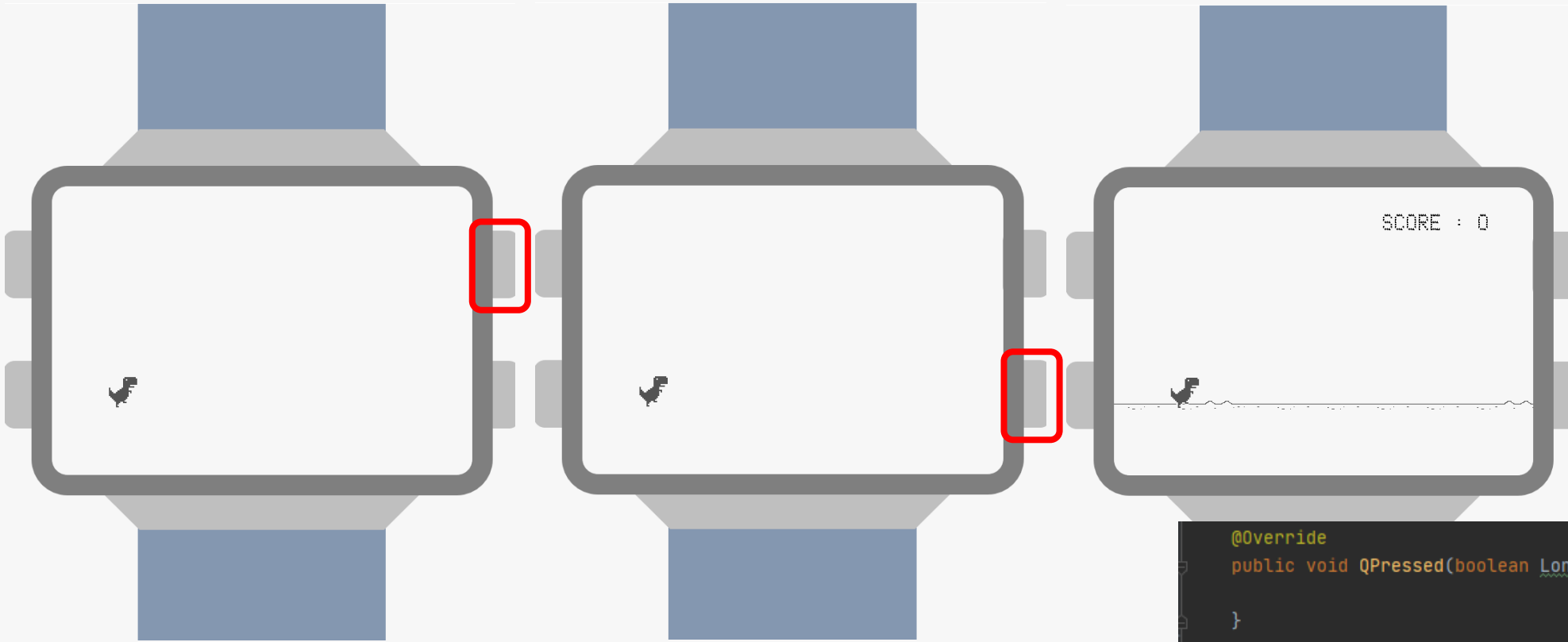
[WorldTime] Summertime을 적용할 필요가 없는 나라는 시간 변동이 있는지 검사한다.

- Summer time을 적용할 필요가 없는 나라도 1시간이 추가된다.



# Bugs & Errors

#28



[Game] 지정한 버튼을 눌렀을 때만 게임이 시작되는지 확인한다.

- start 외 버튼을 눌러도 게임이 시작되고 공룡이 점프한다.

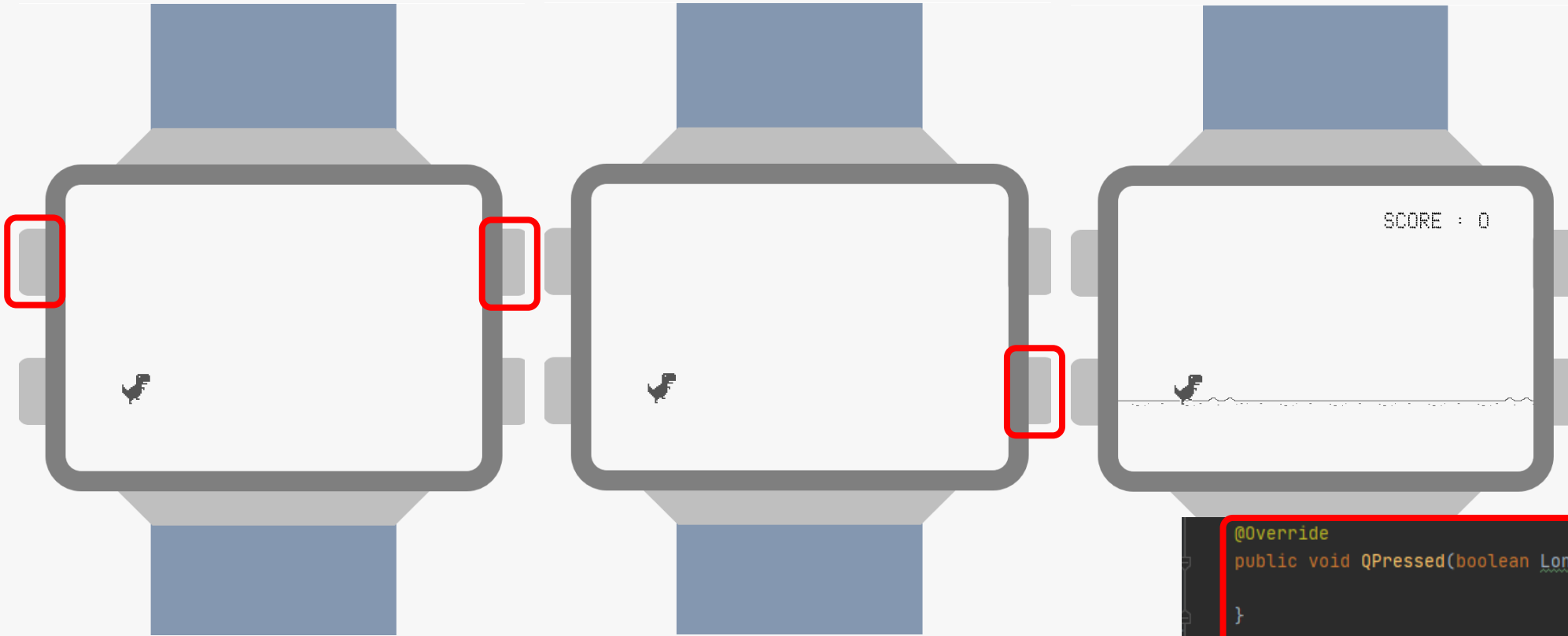
```
@Override
public void QPressed(boolean Longpress) {
}

@Override
public void WPressed(boolean Longpress) {
}

@Override
public void SPressed(boolean Longpress) { buttonPressed(); }
```

# Bugs & Errors

#30



[Game] 무작위 버튼을 눌렀을 때 에러가 나지 않는지 확인한다.

여러 버튼을 연속적으로 누르면 간헐적으로 에러가 발생한다.

```
@Override
public void QPressed(boolean Longpress) {
}

@Override
public void WPressed(boolean Longpress) {
}

@Override
public void SPressed(boolean Longpress) { buttonPressed(); }
```



# Q&A



**THANK YOU!**

