

1. 通信の方法

実装しましたプログラムでは、以下の画像のように 1 つのデバイスが接続すると次のデバイスの接続を待つといった処理を繰り返すようになっています。



2. プログラムの変更点について

I. jamsketch.groovy について

jamsketch 側からの通信方法については特に変更しておりませんので今までの 1 人用 JamGesture と同じ記述以下のように for ループで繰り返すだけの変更になっています。

プログラム 32 行目より

```
if (MOTION_CONTROLLER != null) {

    def ctrl = Class.forName(MOTION_CONTROLLER).newInstance()

    ctrl.setTarget(this)

    for(;user_number > 0;user_number--){

        ctrl.init()

        ctrl.start();

    }

}
```

18 行目にあります、user_number という変数の値を変更することによって接続するデバイスの数を制御しています。現在は、プログラム上で接続台数を記述していますが最終的にはプログラム実行時にユーザに接続台数を入力してもらうように変更する予定です。

II. Rfcommserver.java について

こちらのプログラムについてはデバイス接続部分については JamGesture 実装時から大きな変更は加えていませんが、プログラム 101 行目~155 行目におけるスレッドによる接続デバイスから受信したデータ処理の部分について、接続しているどのデバイスから受信したデータかを判断できるような変更を加えています。

125 行目における user_number という変数で何番目の接続デバイスかを管理することによってどのデバイスから受信したデータかを判断します。

- 問題点

プログラムの 132 行目における接続デバイスごとの受信データの処理に関して

```
if(user_number == 1){  
    tm.setTarget(0, tm.height() * (1.0 - p));  
    if (evt != TargetMover.NO_EVENT) {  
        tm.sendEvent(evt);  
    }  
}else{  
    /*2 人目以降のユーザの TargetMover に関する処理*/  
}
```

user_number の値によって if 文で処理を変更するように記述してあるのですが、この方法では事前にプログラム上で接続上限台数分の処理を書かなければならず将来的に多数デバイスでの使用を目指す場合非常に効率が悪い書き方なのですが、よい方法が思いつかなかったためこのような方法になってしまいましたすみません。