

マルチタッチ/マルチジェスチャ実装用ライブラリの開発

山脇 直樹[†] 飯田 聡[‡] 大杉 友哉[‡] 高見 友幸^{†‡}大阪電気通信大学大学院 総合情報学研究科 デジタルゲーム学専攻[†]大阪電気通信大学 総合情報学部 デジタルゲーム学科[‡]

1. 序論

近年のマルチタッチ入力デバイスの普及に伴い、多くのプログラミング言語でマルチタッチ処理が実装されるなど実用が進んでいる。しかし、これらの機能はハードウェア側での対応や独自の処理の構築を必要とする場合がほとんどである。そのため、センサやカメラといった従来と異なる入力デバイスを用いる場合にはプログラムのアルゴリズムを新たに構築する必要があった。そこで、この問題を解決するために ActionScript3.0 (以下、AS3)用ライブラリを開発した。このライブラリはプログラムの最初に特定の記述を加えるだけでプログラム全体をマルチタッチとマルチジェスチャ入力に対応させるものである[1][2]。プログラマがこれらの動作に対するアルゴリズムを考慮する必要がないため、マルチタッチアプリケーションのオーサリングツールとしても期待することができる。

2. 入力デバイス

入力デバイスとして測域センサ、赤外線カメラ、Android 端末の3つを用いた。それぞれマルチタッチによる複数の座標データを同時に送信できるという点で共通している。

測域センサでは近赤外線レーザビームを掃引することで物体の座標を検出している(図1)。赤外線カメラではレンズの周辺から赤外線を照射することでマーカの三次元座標、およびマーカの大きさと面積を取得している(図2)。また、Android 端末ではタッチ座標をソケット通信によりサーバへデータを送信するクライアントプログラムを導入している。

これらの入力デバイスから送信されてきた座標データを分析することでポインタデータを生成する仕組みとなっている。このポインタデータには座標データやこのポインタが指し示しているオブジェクトの情報が含まれている。

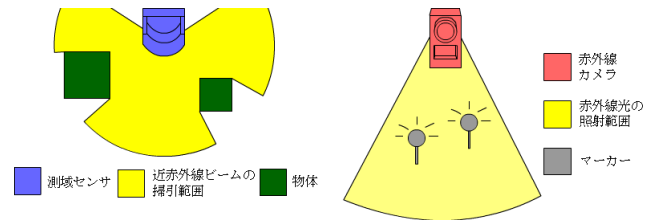


図1. 測域センサによる座標位置検出。

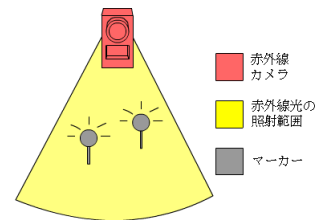


図2. 赤外線カメラによる座標位置検出。

3. ライブラリの実装

入力デバイスから送られるデータを随時分析し続けることでポインタデータの動きを見張り、マウスイベントを始めとするAS3内部のイベントを擬似的に発行している。導入に関してはプログラムのコンストラクタ中に1行とコールバック関数の15行程度を加えるだけである。その際、プログラムのアルゴリズムを変更する必要はなく、デバイスから送られてきた座標データに応じてライブラリが自動的に各種イベントを発行する。

本ライブラリが対応するイベントは MouseEvent, TouchEvent, そして TransformGestureEvent の3つである。また、従来のAS3プログラミングではマルチタッチとジェスチャ入力はどちらか片方だけの選択であるが本ライブラリにおいてマルチジェスチャを実装することで双方の同時利用を実現した。MouseEvent と TouchEvent は各ポインタデータの出現・移動・消去を見張ることでそれに対応したイベントを dispatchEvent 関数を用いて発行する仕組みである。

例えばポイントデータが新たに出現した場合、ポインタの指し示すオブジェクトに対してマウスダウン/タッチイベントを発行するようになっている(図3 a)。同様に、ポインタの座標が変化した場合にマウス/タッチムーブイベントを発行する(図3 b)。このとき、ポインタの指し示すオブジェクトが他のオブジェクトに入れ替わった場合、前のオブジェクトに対し状況に応じてマウス/タッチが外れたというイベントを発行する(図3 c)。その後、新たにポインタの指し示すオブジェクトに対してマウスオーバー/タッチが発生したというイベントを発行する(図3 d)。ポインタが消失するときには、マウスアップ/タッチエンドイベントを発行する(図3 e)。また、

Development of an AS3 Library for Multi-touch and Multi-gesture Applications.

Naoki YAMAWAKI, Satoshi IIDA,
Tomoya OHSUGI and Tomoyuki TAKAMI.
Faculty of Information Science and Arts, Osaka Elec-
tro-Communication University.

ポインタが出現したときと消失したときの位置が 100 ピクセル以内であれば、クリック/タップイベントを発行する(図 3 f)。

2 つのポインタが互いに近くの位置に出現した場合は TransformGestureEvent を発行する準備に入る。このとき、双方のポインタに専用のフラグが設けられ、その 2 つのポインタを繋ぐ架け橋となるデータが生成される。この架け橋データは 2 つのポインタ間の距離と角度から対応する TransformGestureEvent を発行する役割を持っている。直前のフレームからポインタ間の距離と角度の変化を計算する。変化した距離の差と現在の角度から scaleX と scaleY プロパティを設定し、対象のオブジェクトに対して拡大/縮小イベントを発行する(図 4 a)。同様に変化した角度から rotation プロパティを設定し、対象のオブジェクトに対して回転イベントを発行する(図 4 b)。

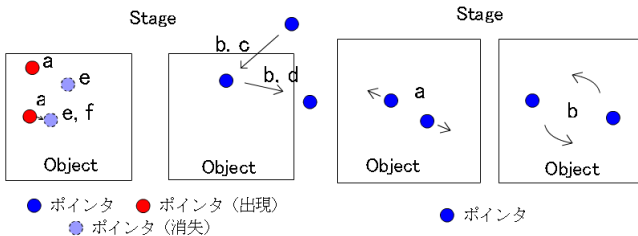


図 3. MouseEvent / TouchEvent の発行時の動作例。 図 4. TransformGestureEvent の発行時の動作例。

本ライブラリでは、計測データからどのようなポインティングデータを抽出するかを Rule クラスのプロパティを設定することによって容易に変更可能である。以下にその設定を行なうソースコード部分を示した。

```
var rule:DefaultRule = new DefaultRule();
rule.eraseIntervalTime = 1000;
rule.upIntervalTime = 100;
rule.moveMeasureValue = 2;
rule.updatePointerDistance = 100;
systemManager.bindProductionRule(stage, rule);
```

eraseIntervalTime プロパティは、ポインタオブジェクトが消失するまでの判定時間、updatePointerDistance プロパティは、ポインタオブジェクト更新時に利用する更新範囲距離、moveMeasureValue プロパティは、マウスイベントのムーブイベントを発行するための範囲距離、そして、upIntervalTime プロパティは、マウスイベントのアップ判定が発生するまでの判定時間である。

4. マルチタッチ

従来 AS3 でマルチタッチを実現するためには AIR ランタイムに搭載されている TouchEvent を用い

て対応するハードウェアによる入力を行うか、送られてきた座標データ列から個別に当たり判定を行うアルゴリズムを構築する必要があった。そこで、本ライブラリでは TouchEvent を用いたプログラムに対する互換性を保ちつつ、本来シングルタッチ用である MouseEvent を用いたプログラムもマルチタッチに対応させることができたようにした。このため、アプリケーションのマルチタッチ化に特別な技術を必要としなくてよい。

5. マルチジェスチャ

AS3 のジェスチャ入力は TransformGestureEvent を用いて行うが、同時に単一のオブジェクトに対してのみ可能である。そこで、本ライブラリではポインタデータの変化から TransformGestureEvent を発行することで複数のオブジェクトに対するジェスチャの同時入力を実現した。この実装に関してもプログラマは従来どおり TransformGestureEvent を対象のオブジェクトで addEventListener するだけで使用可能となっている(図 5)。



図 5. マルチジェスチャの動作。二人で同時にオブジェクトの操作を行っている。

6. まとめ

本ライブラリによって、シングルタッチ用に作られていたアプリケーションをそのままマルチタッチへと対応させることができる。このため、マルチタッチに特有の処理をプログラマが個別に考える必要がなくなり、経験の浅いプログラマでも容易にマルチタッチプログラミングを行うことが可能となる。また、本ライブラリはセンサ、カメラ、Android 端末といった様々なハードウェアデバイスに幅広く対応している。

参考文献

[1] T. Iwamoto, et. al., Development of Wall Amusements Utilizing Gesture Input, ICEC 2010, 499-501, 2010.
 [2] A. Karino, et. al., Development of Multi-touch Table Games with Real Objects, Proc. of NICOGRAPH International 2010 (CD-ROM), 2010.