情報工学実験ⅠE1レポート

5班

19x3132 根本颯汰

10月8日(木) 実施

1, プログラムの仕様

　今回のE1の実験では、映像とコンピュータグラフィックス(CG)の双方を組み合わせた複合現実(Mixed Reality)を用いたものをプログラムで作成した。

　具体的に、ユーザのライブ映像にCGであるボールを映し出し、その停止した大きめのボールの外周をユーザの指先で擦るとそれに応じて同じ角度だけ回転(自転)するようなプログラムを作成した。

2, 仕組み(状態遷移図)

　事前に撮影した背景画像と、現在の背景画像を別々に保存し、この二つの画像を画素ごとに比較して色の異なる部分を抽出する背景差分という手法を用いて背景とユーザの動きの区別を行った。

ユーザとボール(今回用いるオブジェクト)の接触判定は上記の通りに行うので、ユーザとボールの重なり部分を調べた。ノイズの除去を行うため重なり部分の重心と重なり部分が100px以上の時に接触しているとみなした。

　ユーザの指先にボールが触れているかどうかをoverlappingでtrue/ falseを判定し、指先とボールが触れている状態のことをIN\_TOUCH、触れていない状態のことをOUT\_TOUCHとしたときの状態遷移図を以下の図１に示す。

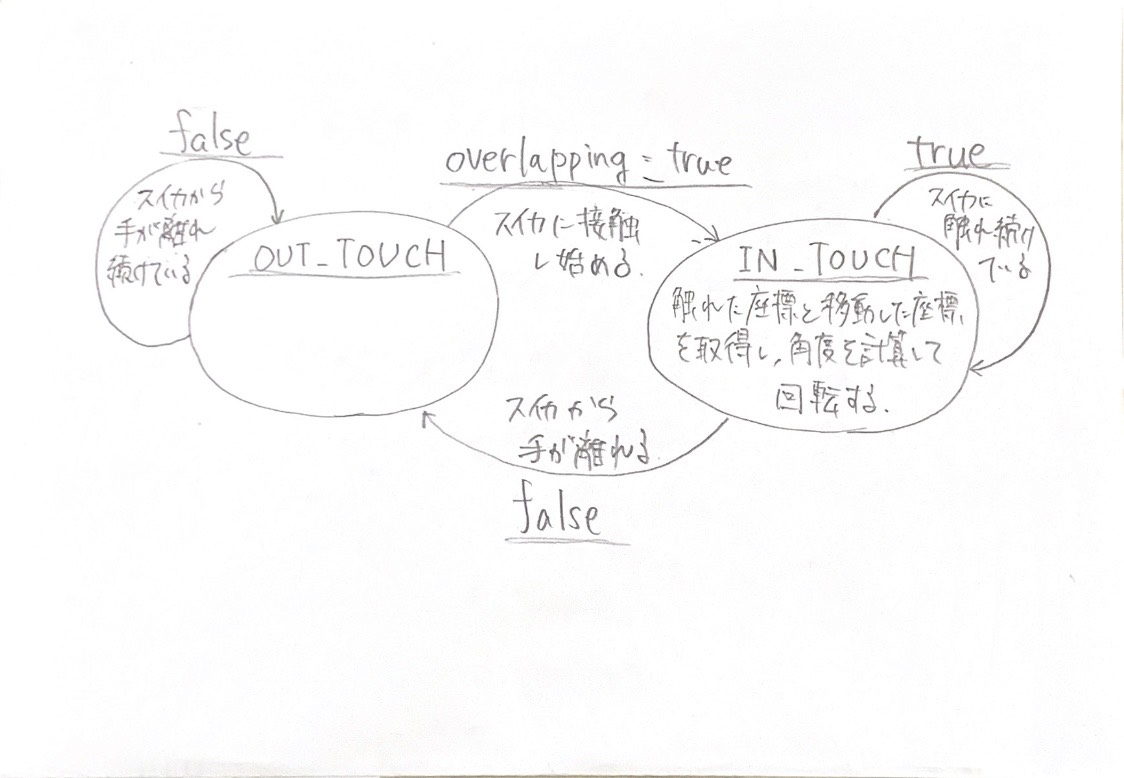


図１：ボールと指先の判定の状態遷移図

　初期状態をOUT\_TOUCHとしユーザとボールが重なったら(overlapping=ture)状態をIN\_TOUCHに移行しユーザとボールが離れたら、つまりユーザとボールが重なっていなかったら(overlapping=false)状態をOUT\_TOUCHに移行する。IN\_TOUCHのときにtrue、またOUT\_TOUCHのときにfalseの場合その状態を継続させた。

3, 実現方法

このコードはVisual Studio 2017で行った。また、事前にball.xを用意した。

Meshを継承したTouchableクラスを用意した。そのクラスの中にボールと指先が重なりの部分とその重心を引数にし、接触判定をtrue/falseで返すcheckOverlappingとボールと指先の重なり部分を引数に接触判定に応じたボールの動作を行うreactの２つのメンバ関数を用意する。他に直近の接触判定の結果を記憶するOUT\_TOUCH, IN\_TOUCHの列挙型とその列挙型の状態を保持する変数(state)をメンバ変数として用意した。

まず、背景差分を行う関数subtract\_maskを用意し、そこでは記憶した背景と現在カメラに写っている背景を引数にARSC::diffで双方の背景の差分を取得し、その結果をARSC::monochromeでモノクロ化(RGBを明度のみに変換)して、最後にその結果をthreshold関数で閾値と比較して閾値未満の場合0、閾値以上の場合画素値の最大値に変換するつまり白黒の二値化を行った。これにより背景を黒、ユーザの部分を白と区別することを可能にした。

次に、上記の通り定義したcheckOverlappingではボールと指先が重なりの部分とその重心を引数にし、ARSC::andでボールと指先の重なり部分の中から閾値を超える部分のみで各画素ごとにand演算を行い接触部分を白くし、接触判定のtrue/falseを返す。また、その演算結果と先の背景差分から画素の位置の平均値と個数を取得し、重なり部分の重心をARSC::getCGで計算した。

最後に接触判定を行い、その結果に応じてボールを回転(自転)させる関数reactでは、図１の状態遷移図の通りIN\_TOUCHに状態が変化した瞬間のボールとユーザの重なった部分の重心のx, y座標をそれぞれpre\_gx, pre\_gyにいれ、その間に変化した現在のx, y座標がgx, gyに入っているので、それらのatanの差分を計算した。(atan2((pre\_gy - c.y), (pre\_gx - c.x)) - atan2((gy - c.y), (gx - c.x)))その計算結果をthetaとしその分をSetRotationZ、Z軸回転(自転)した。また、今のgx, gyにおける接触判定がfalseのとき空気抵抗、慣性を計算し(air\_res \*= 0.93f)、air\_resをSetRotationZさせる。そして、先に入れたpre\_gx, pre\_gyとの接触判定がtrueのときIN\_TOUCH、falseのときOUT\_TOUCHに状態が変化する。一方OUT\_TOUCHの状態の際、今のgx, gyにおける接触判定がfalseの場合慣性の計算をし続け(air\_res \*= 0.93f)、SetRotationZをさせる。

4, 実行結果

　ユーザがボールに触れた方向に合わせてボールが回転(自転)した。また、回転しているときにユーザがボールから離れると慣性のようにしばらくゆっくり同じ方向に回転し、やがて止まった。しかし、時折同じように回転させても勢いよく回転したり、ゆっくり回転したりと回転数が異なる場合が多かった。

5, 吟味

　今回のプログラムでは接触時の座標とそのあとのユーザの移動後の座標から計算した角度の差を用いてボールの回転を行った。しかし、実行結果にもあるようにときに勢いよく回転するときがあったので、回転速度に上限を設けたり、ボールを左上に設けた分カメラの画角ぎりぎりでユーザとボールが接触した際の判定が鈍くなっていたりしたので、より中央よりにボールを設置すると改善されるのではないかと考えた。