* 8/7
  + 180807でクラッシュを確認
    - if文内で＆をとっていないにも関わらず
    - 180731でも確認したため，これまでエラーを吐いていたと思わしき部分はクラッシュに関係ない（Arduinoは180725）
    - 180727でもエラーを確認（Arduinoは180725）  
      しかし，その後エラーは出ず
    - 180720ではエラーを確認せず（Arduinoは180719）
  + 結論：Arduino側が原因ではないか
    - モータの回転速度変更時に問題が起きてるのでは？  
      →Unityで180807，Arduinoで180719（モータの速度変更なし）でエラー出たので違うかも
  + 簡易の伸縮予測を実装
    - ある程度スムーズに動くようになった（階段をなぞっているような感覚ではなくなった…かも）
    - 振動はまだ感じるが，原因は位置修正を行おうとした結果震えているだけのように思える
    - クラッシュ原因はマジで意味不
* 8/8
  + 180807をコピー
  + Arduino180725使用時に，VR空間内でデバイスをめり込ませたまま動作させると画面がフリーズすることを確認（仮想物体表面まで先端を戻せば，再度動作）
* 8/13
  + デバイス先端・後端座標を用いて伸縮予測を行うように実装
    - しかし，うまく動作せず，if文などに問題ありと予想
  + Oculus Touchがかなりの頻度でフリーズしてほとんど確認できず
    - UnityとOculus Rift自体は動作し続ける
    - 本日だけでなく，180808のプログラムでも同様の問題を確認
  + 今日は一切フリーズせず
* 8/20
  + 過去のデバイス先端・後端座標を1フレーム目以降更新していなかったため，動作に問題があったことを確認・解決
* 8/21
  + Check\_PenetrateObject関数を作成
    - デバイスがオブジェクトにめり込んでいるかの判定を行う
  + Get\_HitPointLength関数を変更
    - 変更点としては，スライダを伸ばす始点と終点の2つの引数が必要となった
    - 以前の状態では，デバイス先端から床までの長さしか算出できなかったので，めり込み時に伸ばさなければならない正確な長さが算出できなかった
  + 現在の問題点
    - デバイスの先端・後端をもとに伸縮を予測しているため，予測位置におけるデバイス先端が仮想物体表面から少し浮いていても伸縮可能範囲内であればスライダが伸びてしまう  
      結果として，デバイスが仮想物体表面から浮いてしまい，うまくなぞることができなくなる
  + 解決策
    - 現段階では思いつかない
* 8/22
  + 昨日の解決策として，とりあえずRayと仮想物体表面，Rayと地面の接触点間の距離を伸縮する値とした
  + 180822\_2は仮想物体へのタグ付けで判定したVer
    - 気持ちクラッシュ回数が増えた気がする
  + あと，スライダが動かなかったとき，Arduino側のバグなのか，伸縮可能範囲を超えただけなのかわからないので，デバイスの色が変わるように実装
* 8/24
  + バグが発生したが，本日はモータシールドのマイナス側が点灯していないとき（モータが動作していない時）に発生を確認
  + HMDを被った状態でデバイスを動かすと，HMDを手で保持した状態よりも伸縮のブレが大きいような気がした
* 8/28
  + 変更点
    - デバイス先端・後端が一定以上動作した場合のみ，preTipD，preEndDが更新されるように変更
    - また，JITTERの値が大きすぎるとデバイス静止時のスライダの動作が大きくなりすぎると考え，小さめに設定
  + 結果として，HMDを被った際のブレは変化せず
    - HMDを被らずに制止させた場合も同様にひどく振動する場合を確認
    - また，クラッシュする回数が多かった気がする
* 8/30/18
  + 判明したこと
    - DeviceControlerクラスのGet\_HitPointLength関数が呼び出されていないのに勝手に動いて，データを送信する直前に距離を計算しなおしていたことが判明
  + 変更点
    - Get\_HitPointLength関数で数字を更新するのは，sendFlag = 0の場合のみに変更
      * この変更後，クラッシュが発生しないようになった（するときはする）
    - 3Dプリンタで作製したパーツを取り付けた
      * かなりスムーズにスライダが動くようになった
* 9/7
  + 判明したこと
    - VR空間自体が実世界と平行じゃなく，少し傾いている  
      （多分センサが原因かと）
    - HMDを被ったときフレームレートが6分の1ぐらいになっており，これがHMDを被ったときデバイスがガタガタする理由だと考えられる
      * これが解決しない限り運用は不可能
  + 変更点
    - CallibrationBaseクラスのUpdate関数内をOculus Touchのトリガーではなく，キーボード入力でBaseの位置を調整できるように変更
      * なぜかOculus Touchの入力が認識できないようになったので応急措置
    - デバイスの伸縮予測を過去の動きをそのまま使うのではなく，数フレーム前から加重移動平均で求めた値を予測値として利用
* 9/10
  + 変更点
    - 伸縮予測を行った際の次フレームの移動距離を1/15に変更
      * FpsがHMD起動時に約140fps→約90fpsに落ち，その結果フレーム間の移動距離が15倍ほどになっていたので変更を行った
      * 結果として，デバイスの振動自体は小さくなったが，それでも振動は感じるほど大きいので修正はまだ完了ではない
* 9/13
  + 気づいたこと
    - 伸縮中に押し付けて短くなった瞬間に画面全体がフリーズすることがある
    - クラッシュはほとんどしなくなった（本日1回のみ）
  + 判明したこと
    - UnityでFpsを固定するためには，Project Setting→Quality→VSync CountでDon't Sync（ディスプレイとの垂直同期なし）を選んだ上で，プロジェクトをビルドする必要あり
      * つまり，スクリプト上でfpsを固定しても，表示するディスプレイのリフレッシュレート（モニタ：150Hz，Oculus：90Hz）に固定される
    - Oculus Touchがよく空間内で固定されてしまう現象について
      * Oculus Touchの座標はOculus Riftを通じてPCに送信されるため，Oculus Riftがセンサに認識されていないと座標が更新されない  
        （傾きに関してはジャイロセンサを使用しているのか，座標が更新されなくても問題ない）
    - Get\_HitPoinLength関数は現状問題なく動いているが，少し変更するだけで正常に動作しない可能性あり
      * sendFlagの更新がsliderLength，nextLengthのどちらも計算するときに更新される可能性がある
      * sendFlagの更新を別の関数で行うか，別の場所で行うべき
  + 変更点
    - サブカメラの作成
      * PCモニタにOculusの映像をミラーリングするか，固定されたサブカメラからの映像を表示するか指定可能  
        （詳細はSubCameraに取り付けたCameraController.csを確認）
      * 動画撮影時などに利用
    - 閾値（JITTER）を10→20に変更（Unity，Arduinoどちらも）
      * 結果としてかなりスムーズに動くようになった
      * それでもガタガタする場合があることに関しては，VR空間内が現実と水平ではないため，常にデバイスがめり込んだ判定になっており，それを押し返そうと動くためだと思われる  
        ⇒早急に現実と仮想の机の位置合わせを正確に行う必要がある
* 9/14
  + 変更点
    - 仮想物体とデバイスの接触角度によって伸縮するか否かを判定
      * デバイスが傾きすぎている状態で仮想物体と接触しても，実空間では本来ならば反力は提示されないので，伸縮を行わないように変更