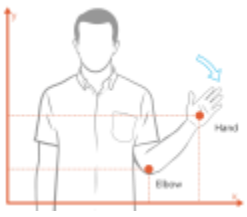
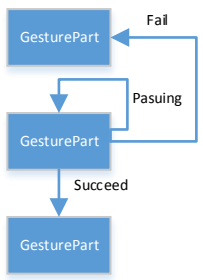
GestureLib v1.0

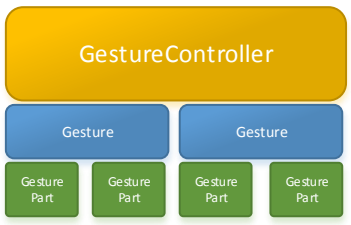
1.Gesture Segments:



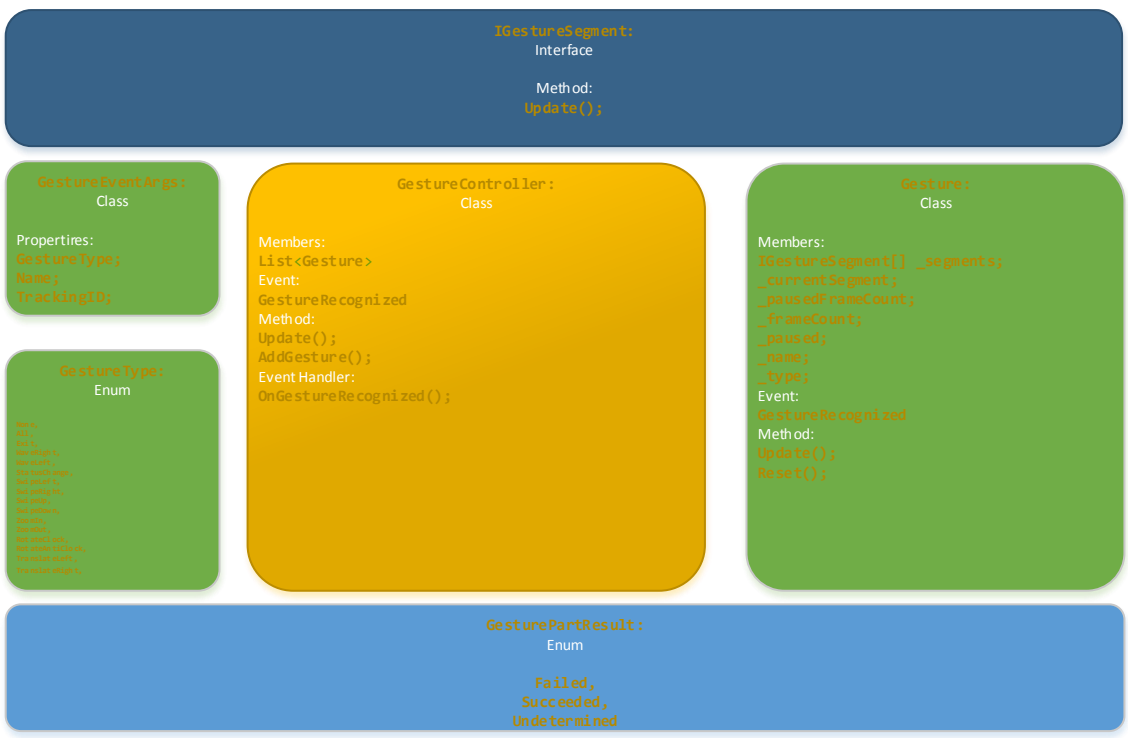
2.GestureLib Solution:



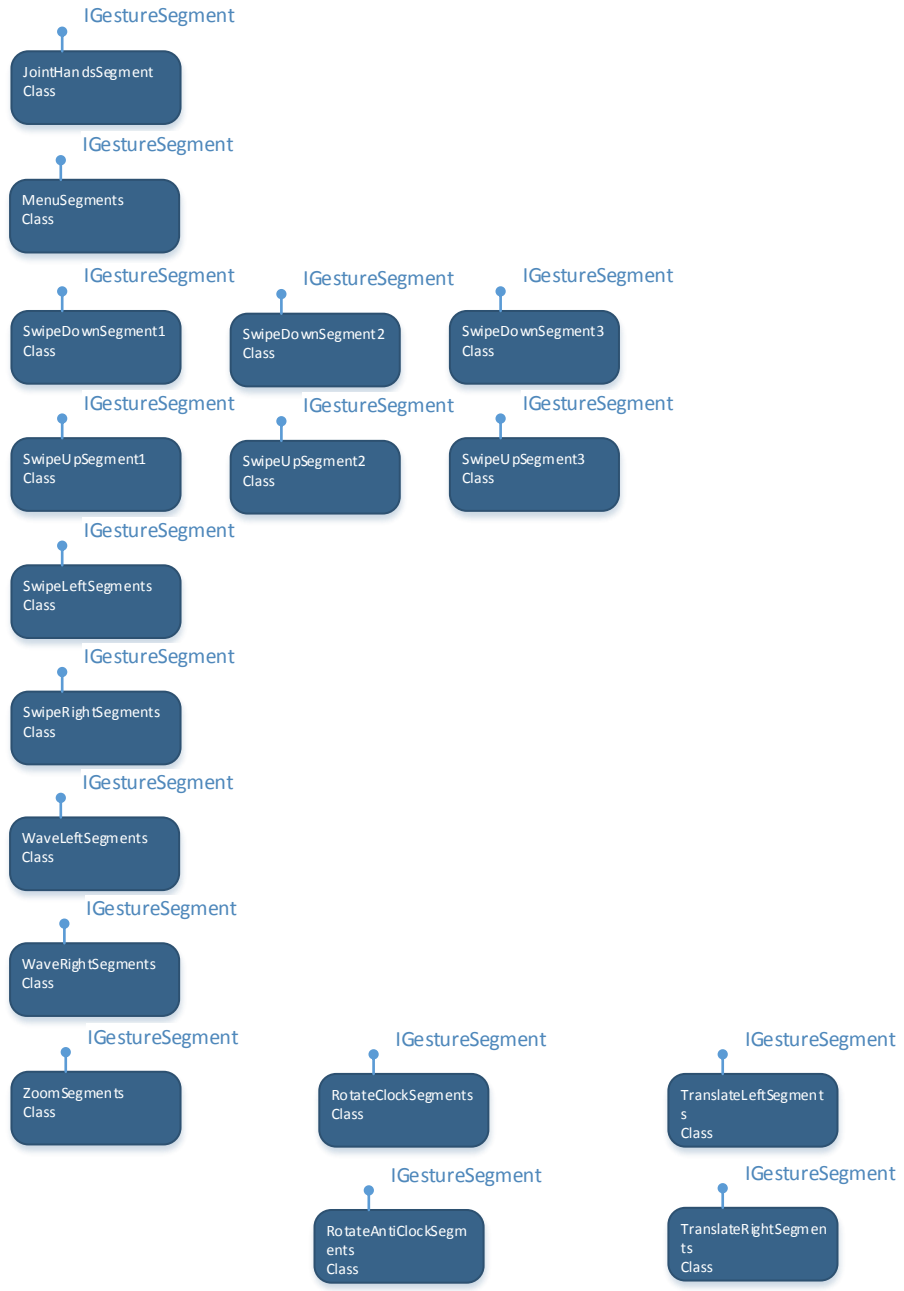
3.GestureLib Dataflow:



4.GestureLib Framework:



5.The Gestures:



IGestureSegment: 接口类，定义手势片段的识别接口
IGestureSegment接口定义手势片段的识别，仅包含一个Update方法，通过输入当前骨骼数据skeleton，返GesturePartResult类型

Gesture：类，手势识别的核心类定义，主要处理连续的手势片段的匹配
Gesture由手势片段组成，在其构造函数中通过手势类型GestureType和手势片段IGestureSegment[ ]_segments来定义手势，并根据不断输入的骨骼帧数据进行匹配
其中：
IGestureSegment[ ] _segments; //为存放一个手势的所有片段，***要区别于GestureController中的List<Gesture> _gestures***
int _currentSegment; //为当前匹配手势的正在检查的片段，即手势片段的序号
int _pausedFrameCount; //等待骨骼帧数，默认为10,
int _framecount; //等待帧的计数器
int _paused; //是否等待的状态指示
Update(Skeleton skeleton); //Gesture类中的核心方法，该方法在Kinect骨骼数据实践中被GestureController驱动调用，该方法参照集合IGestureSegments[ ] _segments, 逐帧进行对比，全部匹配后，触发事件GestureRecognized

GestureEventArgs：类， 手势识别的传参类
GestureEventArgs类是手势识别事件的传参类，包括GestureType Type； Name； TrackingID三个properties。

GestureController：类，手势识别的核心引擎，用于驱动多个手势识别的处理和调度
GestureLib的核心引擎，用于驱动多个手势识别，由于GestureController是GestureLib中的控制引擎，所以必须在应用程序的初始化部分定义，并注册手势识别事件。
在GestureController的构造函数中对各个手势片段进行实例化，并通过手势片段的组装来注册单个 手势，手势片段的顺序不同可以组合成不同手势。
主程序的OnGestureRecognized手势识别成功事件中，根据GestureEventArgs的事件类型来处理各个手势。
在主应用程序的骨骼跟踪事件中，调用_gestureController.Update(skeleton);来进行若干手势的识别。

GestureType：枚举，定义全部能够识别的手势类型
GesturePartResult：枚举， 定义每一帧手势片段的识别结果

How to used this framework:
1.独立实现一个IGestureSegment的新接口
2.在核心类GestureController的构造函数中通过AddGesture方法实现对这一手势的识别
该架构之所以能够如此简单的自由的添加gesture主要是采用了架构设计的主要思想之一：数据流和控制流成功剥离