(12 pts 30 minutes) Read Don Norman’s critique of gestural interfaces at:
Choose three of Nielsen’s principles for heuristic evaluation (from the lecture slides) and, for each, briefly explain why it would be either much more or much less likely to catch interface problems in gestural interfaces than in WIMP interfaces. Think about the differences between gestural and WIMP interfaces.

Reduce User Memory Load
This principle for heuristic evaluation basically states that the application should always give good indications of its functions rather than relying on the user to memorize where each function lies and how to execute it. Mostly this has to do with a certain level of visibility that each application needs in order to make its use easy and efficient for the user.

Gestural interfaces present a major problem in the use of this heuristic. The main issue lies in the fact that gestures cannot be easily indicated by the program without detailed instructions on each page which can consume a lot of space. As such most applications rely on the user’s ability to memorize gestures and be able to use them when needed. If this heuristic were to be applied normally on any gestural interface, many of these functions would inherently be labeled problematic as they are not always made obvious to the user. Furthermore, the heuristic would fail to properly identify gestures that impede the user’s ability to use the application. For example, Spawn Illuminati is an iPhone/iPod Touch application that allows the user to create abstract pictures by moving points/lines of light around the screen. In this application, different touches on the screen have different functions and the application only sometimes brings up a dialog box indicating what was just done.

Be Consistent
This principle states that within and even across applications elements should stay consistent and easily identifiable in order to allow the user to adapt to multiple functions. Tasks should be performed in similar manners, and the look and feel of an interface should be fairly consistent.

In gestural interfaces, this heuristic would be much less likely to catch problems because as it stands now, especially when looking at the variety of applications available in such things as the Apple App Store, each developer is allowed to define how to execute functions in their own ways. As such the user must learn each individual application and the new gestures each presents. Heuristic evaluations using this principle would then have little value because consistency is not always present. For example, on the iPhone/iPod Touch, the user can enlarge/shrink items using a pinch gesture but this is not always true as some applications choose to use +/- buttons in the place of that gesture. Here this particular principle would be unable to identify whether either is problematic in the given interface.

Prevent Errors
This principle states that applications need to give proper feedback to the user in order to prevent errors from being made. This includes giving appropriate messages when a user is about to do something in order to identify what state the application is in/what function is about to be executed. It also includes disabling certain functions when in different modes so as to diminish the likelihood of mistakes while performing a given function.

The lack of consistency in gestural interfaces would make it hard for this principle to identify ways in which to properly prevent errors. Many times, especially on the iPhone/iPod Touch, gestures execute functions without any sort of confirmation. For example, in my experience I have deleted several emails that I did not mean to because of a simple mis-touch of the screen. Because confirmation dialogs especially on a touch interface can make executing functions cumbersome it is difficult to find a good way of doing so. As such this principle falls short of identifying where it is appropriate to place dialog boxes or disable certain functions in order to prevent the user from making errors.

Far enough.