Ubuntu Gesture UI Guidelines

Version: 0.4

Author: John Lea

Contents

1. Introduction

This document details the base gesture primitives and behaviors that are used in the Ubuntu Touch shell and applications. This document serves as the reference that defines correct touch gesture usage, and what is specified here takes precedence over any contradictory documentation. Following this guide should help ensure that your application behaves consistently with other applications on the Ubuntu platform.

This document does not aim to provide an architectural overview of Unity or teach gesture based interaction design; rather it provides a pallet of functions for interaction designers and application developers use when developing for the Unity platform.

2. Underlying assumptions and rationale

The gestures and interaction conventions presented in this document conform to the following stipulations:

2.1 Biomechanical considerations

The standard biomechanical considerations that should be taken into account are:

1. Avoid outer positions (full extension of fingers)
2. It should be possible to perform the gesture with relaxed muscles
3. Relaxed neutral position is in the middle between outer positions
4. Avoid repetitive actions
5. Avoid forced holding of a static position beyond a brief pause to establish intent.

2.2 Gestural heuristics

To help meet the innate challenges of multi-touch development the following heuristics should be applied to gestural interactions:

2.2.1 Task frequency

The most commonly used functions should be mapped to the simplest gestures. The addition of more complex gestures must never degrade the usage of simple gestures.

2.2.2 Completeness

Simple gestures must be consistently supported in across the Ubuntu platform. Completeness takes precedence over features; simple gestures must be fully supported before new more advanced gestures are launched. Once a gesture is learnt by a user it should be applicable in all relevant contexts.

2.2.3 Responsive feedback

Immediate visual feedback must be provided in response to all gestural actions. Low latency is of critical importance. If an application cannot provide instant low latency feedback it should make use of system visual indication functions.

2.2.4 Physical mapping

Functions should map closely to the physical action implied by the gesture. For example rotating two fingers clockwise should rotate the object being interacted with clockwise.

2.2.5 Resonant

Similar outcomes should be achievable through similar gestures across the Unity platform. For example a translation or rotation gesture should generally have a translation or rotation consequence wherever it is used.

2.2.6 Easy to perform and remember

The basic gesture vocabulary should be minimal with complex actions performed through a sequence of simple, logical gestures.

2.2.7 Full simple gesture support

All gesture functions must be accessible via single touch. Compound (chain) gestures can provide a shortcut, but a user must be able to complete all gesture based user journeys using only simple non-compound gestures.

2.2.8 Complimentary gesture consideration

Where something can be manipulated through touch, the full set of complementary gestures should be considered and defined, if only to explicitly ignore such a gesture. For example if you can rotate a object right, consider and define explicitly which function is mapped to a left rotation.

2.2.9 Predictable

Gestures should not invoke outcomes that surprise the user.

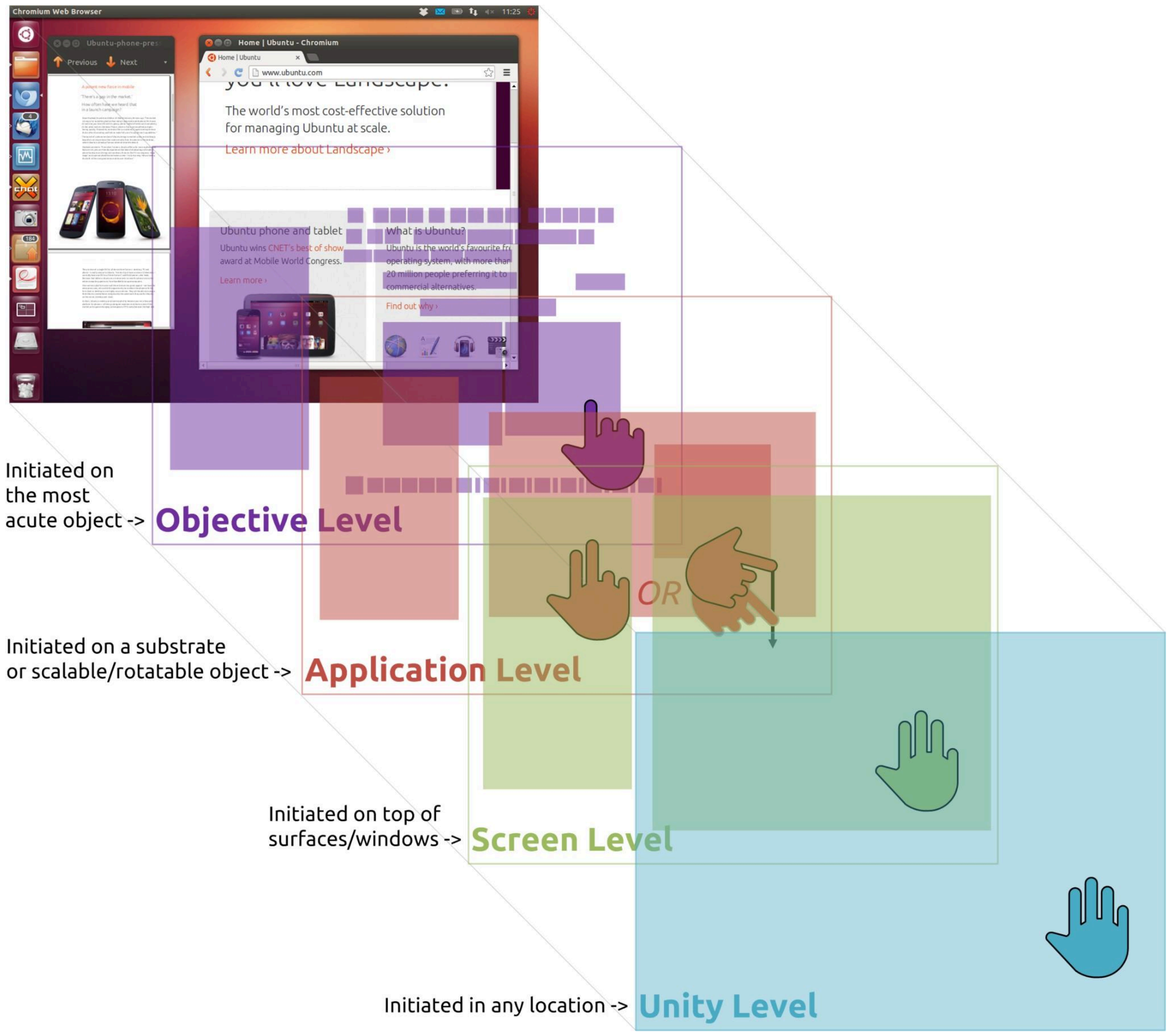
3. Interaction Levels

Gestures in Ubuntu may be aimed at an application, the surface manager (MIR), or the shell (Unity). We call this layering of where the touch input should be consumed the "interaction level". The interaction level entered is determined by both the number of fingers used for the initial touch (the gesture's *initiation*) and in the case of edge gestures by a set of ‘edge gesture characteristics’.

Level	Initial touch	Description
Objective	1 finger screen	Objective gestures always manipulate the most detailed interactive entity at the point pressed, for example a page in a ebook or a image in a word processor document. They are also perform mouse analogue interactions that users will expect based on their past experience of desktop point-click functions. One finger gestures are unique in the portfolio in that they can be used to specify exact points and arcs.
Application	2 fingers screen	Two-finger gestures are application specific. Any gesture or gesture sentence that is initiated with two fingers will be handled by an application. Generally, two-finger gestures are **spatial** in that they directly manipulate a plane: they might translate, rotate or scale a surface, or the viewport. Where the most specific object is the substrate (e.g. in applications where the primary function is to consume media), single finger gestures may also initiate spatial Application level interactions. For example dragging a single finger in a PDF viewer would move the document. Also fast one finger movements are often interpreted as application level interactions because the speed of movement implies a low granularity of target.
Screen	3 fingers screen	Screen gestures are about managing the use of screen space and focus. Typically these interactions focus on window/surface, tab and app management tasks.
Unity	1 finger edge 4 fingers screen	Either a 4 finger gesture at any location on the screen or a 1 finger edge drag. Accesses Unity shell functions that can overlay any working environment, for example revealing the Launcher, Dash, Indicators, or bottom toolbar. Note that once a Unity element is revealed, the user can interact inside the element using exactly the same gestures as used inside an application. For example, inside a Dash lens the user can drag & drop using the standard objective level gesture, or when inside the launcher they can scroll up and down with a single finger drag.

3.1 Interaction level initiation

Different gestures initiated on the same position on the screen can be passed through to either the app, surface/window management or the shell depending on the charastics of the gesture. For example a 3 finger drag on the tablet initiates a ‘screen level’ gesture that enables the user to move a surface between the main and side stages. In this example a one or two finger gesture performed in the same location would be passed through to the app. Another example is on the phone where a gesture performed on the edge of the screen can be passed to either the shell (Unity level) or to the app (Objective level) depending on the charastics of the gesture.


Initiated on the most acute object -> Objective Level
Initiated on a substrate or scalable/rotatable object -> Application Level
OR
Initiated on top of surfaces/windows -> Screen Level
Initiated in any location -> Unity Level

3.1 Edge drag gestures

These are Unity level gestures that allow the user to reveal and interact with shell elements like the Launcher. To reduce the number of false positives, edge drag gestures are only triggered when a user drags their finger over the edge of the screen and the drag **does not** have following characteristics:

- If the initial velocity is zero or close to zero an edge gesture is not triggered. The gesture is passed through to the app. This is because the intended gesture is probably a tap or a hold

- If the initial velocity is high an edge gesture is not triggered. The gesture is passed through to the app. This is because the intended gesture is probably a flick

- If the initial direction is not close to perpendicular to the edge an edge gesture is not triggered. This is because the intended gesture may be a vertical drag.

- If at the gesture's initiation the user has more than one finger touching the screen an edge gesture is not triggered. This is because the intended gesture may be a pinch, spread or rotate.

3.2 Compound gestures

Gestures can be joined together sequentially and in parallel to form compound gestures. The interaction level of compound gestures is set at the moment of initiation. For example, the user might initiate a three finger gesture, remove one finger but continue with the movement as a two finger gesture and finally remove another finger. In this example the compound gesture will be interpreted at the screen level of interaction because it was initiated with three fingers. Another example is that a the user may initiate with an edge gesture, and then continue the gesture to perform a complex operation. Compound gestures are broken into the following phases:

1. **Initiation** - The initial touch contact.

2. **Continuation** - One or more gestures applied sequentially. If a continuation gesture involves all fingers leaving the screen (for example when a tap is part of the compound gesture) the fingers must return to the screen within a certain interval otherwise the compound gesture is terminated.

3. **Termination** - The gesture is ended when all fingers being lifted from surface for longer than a specified interval.

3.3 Greedy windows or views

Normally applications receive only objective and application level gestures and up to two finger direct multi-touch event feeds from Unity. However if an application needs to receive multi-touch events using more than two fingers it can enter the 'greedy' application state. Applications cannot launch themselves into 'greedy' state. To make an application 'greedy' the user must perform the greedy trigger gesture (a three-finger hold) on top of the application surface. Performing the same gesture a second time on top of the application surface exits the 'greedy' state for that specific application.

3.3.1 Normal behavior

By default screen and unity level gestures can be made on top of application windows, for example a four finger tap on top of an application window will bring up the Dash. Applications are normally only able to capture objective and application level gestures and up to two finger multi-touch event feeds.

3.3.2 Greedy windows or views

Some applications need to capture all multi-touch input and override screen and unity level gestures within the scope of the application window, for example a finger painting app. The user can tell the system to direct all gestures to such a "greedy" application - either to a single surface, all surfaces controlled by the application, or portions of a surface. When greedy windows exist, screen and unity level gestures need to be initiated outside of those areas. The greedy behavior is optional; it should only be used by applications that can derive significant benefit from greater than two finger multi-touch interactions. Screen and unity level gestures cannot be initiated within a greedy area.

When a user performs the 'greedy' trigger gesture on top of an application that supports greedy interactions, a visual effect will confirm that which windows have entered the greedy state. Performing the 'greedy' trigger gesture on top of an application that does not support a greedy state will produce no effect.

If an application has multiple surfaces it can choose which surfaces become greedy. Applications can also choose to only make a portion of the window greedy. Greedy applications continue to be greedy regardless of window focus, any greedy space that remains on screen will consume all gestures initiated there.

3.3.3 Gesture initiation

The locations where screen and unity level gestures can be initiated change when an application enters greedy state.

Gestures can be initiated...	Normal behavior	Greedy state
Objective level gestures	...within app window	...within greedy space
Application level gestures	...within app window	...within greedy space
Screen level gestures	...any location on screen	...outside greedy space
Unity level gestures	...edges + any location on screen	...edges + outside greedy space

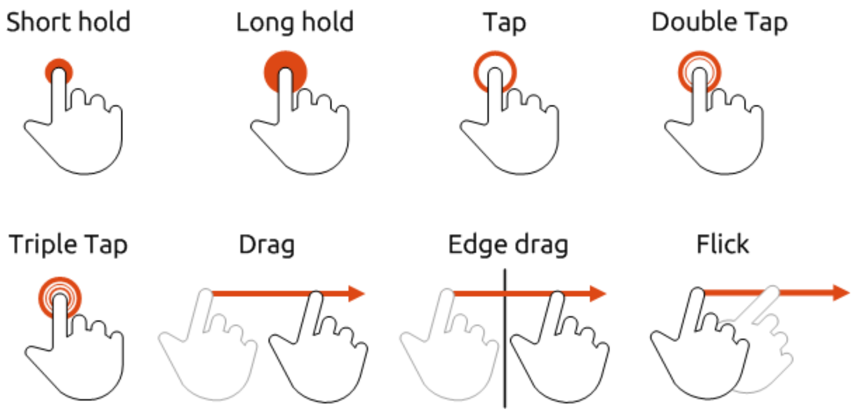
3.3.4 Protected ‘greedy mode’ on/off gesture

The ‘greedy mode’ on/off gesture (three finger long hold) is a protected gesture that will always function in any location, even when initiated over a greedy application. If this gesture is performed over a greedy application a message is passed back to the application to enable it to undo any actions it has performed as a result of the gesture. This is useful in the case of say a finger painting app where the user has performed a three finger long hold. As soon as the user touches the canvas with three fingers three spots are painted. After the user long holds and the protected gesture is detected, the app can then undo the three spots which were painted at the moment the protected gesture was initiated.

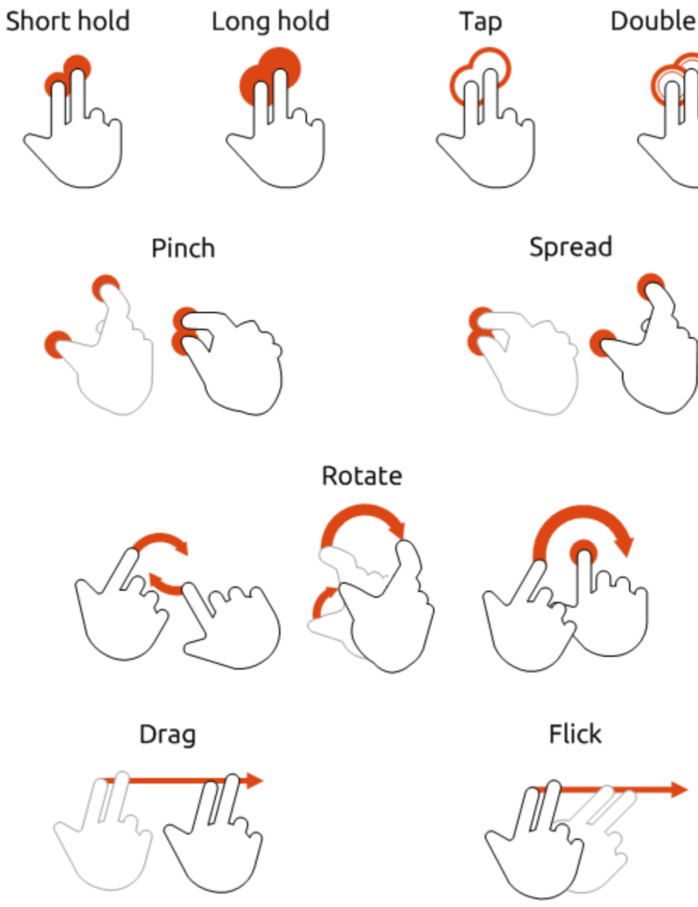
4. Gesture primitives quick reference

This section details the physical actions from which Unity gestures are composed.

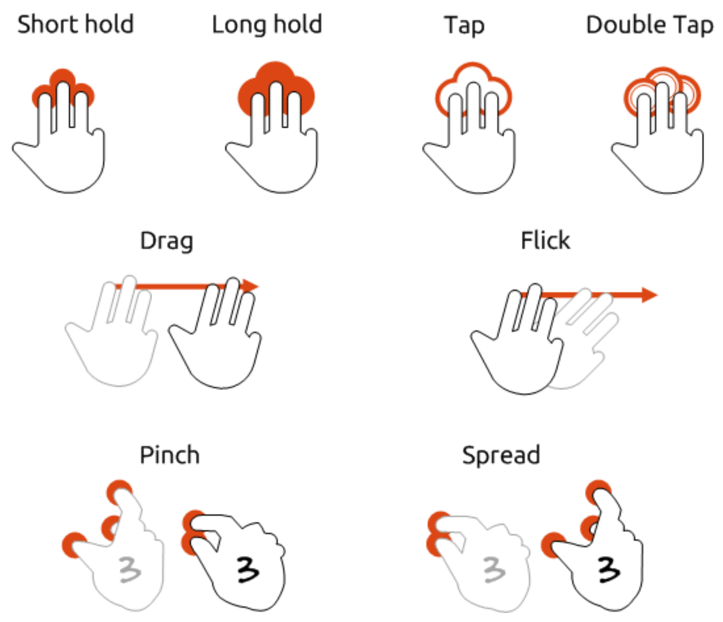
4.1 Single finger



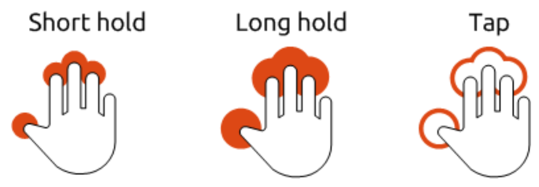
4.2 Two fingers



4.3 Three fingers



4.4 Four fingers



4.5 Minimum dimensions of touchable areas

A touch target should be no smaller than 9mm or 34px. The minimum spacing between visual elements should be 2mm or 8px. However the touch targets can be directly adjacent. The visual size of a touch UI element can be 50% to 100% of the touch target size.

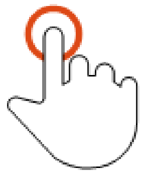
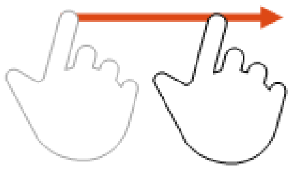
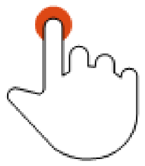
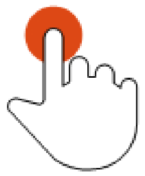
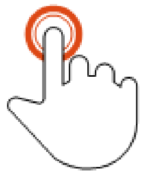
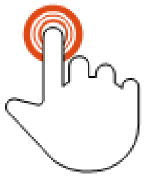
4.6 Affiliated gestures

When mapping a gesture to a function, care must be taken to map all the affiliated gestures to their respective functions. For example if a drag gesture moves a list downwards, the flick gesture should perhaps enable the user move downwards through the list faster.

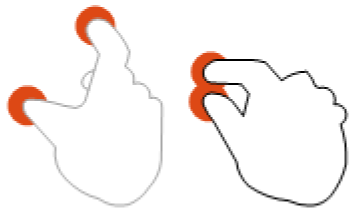
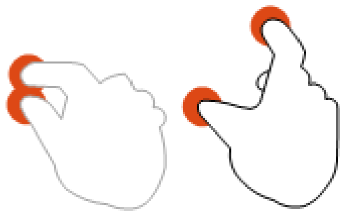
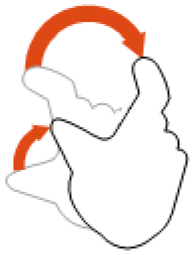
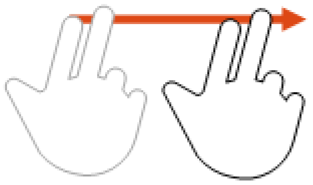
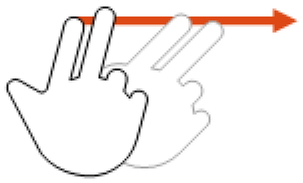
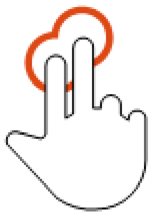
5. Initial gesture usage

The actions performed by initial gestures should be consistent across Ubuntu and Ubuntu applications. Initial gestures are also the entry point (Initiation) for compound gestures.

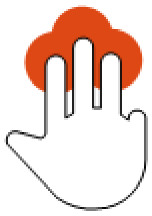
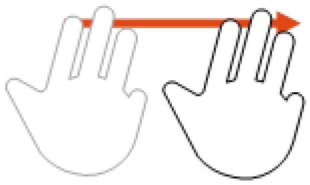
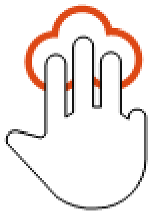
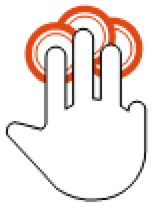
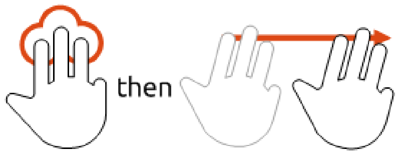
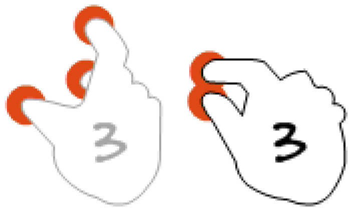
5.1 Objective Gestures

Gesture	Typical action	Visual Representation
Single finger tap	Select item	
Single finger drag	Move most specific item	
Single finger flick	Move the substrate	
Single finger short hold	Open Quicklist or context menu	
Single finger long hold	Enter a 'edit' mode e.g. re-order icons on the launcher	
Single finger double tap	Select a whole word or item.	
Single finger triple tap	Select paragraph or closely related set of items.	

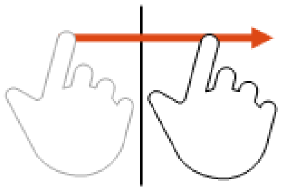
5.2 Application Gestures

Gesture	Typical action	Visual Representation
Two finger pinch	Contract the most specific item	
Two finger spread	Expand the most specific item	
Two finger rotate	Rotate the most specific item	
Two finger drag	Move the substrate	
Two finger flick	Move the substrate	
Two finger tap	Zoom in/out toggle	

5.3 Screen Gestures

Gesture	Typical action	Visual Representation
Three finger long hold (protected gesture)	Make application, surface or window gesture greedy	
Three fingers drag any direction	Moves surface between main and side stage (tablet) Moves window (desktop)	
Three finger tap	Opens the HUD (focused on the surface the gesture is performed on)	
Three finger double tap	Switch directly to previous window (Alt-tab release)	
Three finger tap then drag	Open Alt-tab view with focus switched to the previous window	
Three fingers pinch	Trigger spread exposing all windows/surfaces.	

5.4 Unity Gestures

Gesture	Typical action	Visual Representation
Four finger tap	Open or close the Dash	
One finger edge drag	Reveal shell element e.g. Launcher	

6. Gesture testing criteria notes

6.1 ISO 9241-9 evaluation criteria

This evaluation involves asking a user to perform the same task using two different input devices. For the purposes of evaluating touch gestures we baseline the test against traditional keyboard and mouse interaction. The criteria are:

- Overall operation
- General comfort
- Accuracy
- Operation speed
- Operation smoothness
- Operation effort
- Actuation force
- Finger fatigue
- Wrist fatigue
- Arm fatigue
- Shoulder fatigue
- Neck fatigue

6.2 Synaptics gesture satisfaction criteria

This questionnaire asks users to rate their experience of performing a gesture based interaction on a scale of 1 to 9.

Please rate your experience using the [gesture name] gesture today:

Question	Criteria (1 to 9 scale)
Using the gesture was...	Very difficult - Very easy
Performing this gesture felt	Unnatural - Natural
Starting this gesture required	Too little motion - Right amount of motion - Too much motion
The action caused by this gesture was	Too slow - Just right - Too fast
Performing this gesture was	Not tiring - Tiring
I would use this gesture if it were available	Strongly disagree - Neither agree or disagree - Strongly agree

7. Glossary

Bounce Stop - When a moving object hits a boundary it can perform a bounce stop. The bounce stop produces an effect similar to hitting a dampened spring. The object continues past the boundary line but it's velocity is rapidly absorbed. After reaching zero velocity, it bounces back losing velocity and stopping exactly at the boundary line.

Compound Gestures - A number of distinct gestures performed sequentially and/or in parallel.

Compositional Analysis - Understanding the multi-touch input and translating it into gestures and compound gestures.

Drag - A user presses one or more finger(s) to the touch surface, and then moves these finger(s) in a direction. At least one finger remains in contact with the touch surface until the end of the gesture. In the moments prior to the release, the average velocity is under a certain value. This results in the moved object stopping exactly at the point of touch release.

Flick - A user makes contact with the touch surface, and then moves in a particular direction, losing contact with the surface in the process; unlike a drag, contact is broken after the movement. At least one finger remains in contact with the touch surface until the end of the gesture. In the moments prior to the release, the average velocity is above a certain value. Additionally there may be a decreasing pressure slope. The velocity of the gesture around the point of release determines the velocity imparted to the object. The object continues to move until the movement absorbed by resistance or until the object hits a stop point, triggering a hard or bounce stop.

Gesture Result Preview - A hint (preview) of what will happen if a user finalizes a particular gesture.

Gesture Result Range - A gesture that, through a range of motion, effects a naturally ordered range of results. For example a zoom gesture in a file manager could change the view along a continuum of file display options starting with a simple list view and ending with large thumbnails.

Hard Stop - When a moving object hits a boundary it can perform a hard stop. The hard stop instantly stops the object exactly at the boundary line.

Hold - A user makes contact with the touch surface and then maintains that contact point; no movement is associated with this gesture.

Initial touch - The first touch on the screen at the beginning of a gesture.

Parallel Gestures - Two or more hands making separate independent gestures to control independent functions.

Physics Environment - All gestures happen within a physics environment. The physics environment consists of object weights, resistance characteristics, hard or bounce stop attributes, etc...

Primary hand - The right hand for a right handed person.

Secondary hand - The left hand for a right handed person.

Shuttle style interaction - A interaction that mimics the behavior of a shuttle wheel or slider. Shuttle wheels are commonly found in audio/visual hardware. Shuttle wheels control speed and direction, with the speed determined by the distance the wheel or slider is moved from its starting point.

Simultaneous Gestures - Two distinct gestures performed at the same time with one hand (e.g. performing an expand gesture while dragging across a screen).

Specific item - The most acute (detailed) commonly accessed entity in a particular location.

Surface - Either a window in a windowed environment or a surface in a tiled environment

Touchpoint menu - A menu that is opened in response to a gesture at the location the gesture was performed. For example it could be opened after a single finger hold or tap-drag-release gesture.

Touch-stall - When a user touches a object that has momentum, the momentum instantly stops.