touch sensitive display (not shown). The user presentation and interaction module **24** then collects coordinates of the touches on the display in response to user interaction **30** and passes the touch coordinates to the auto-resizing logic module **28**.

[**0015**] In accordance with the present invention, the auto-resizing logic module **28** analyzes the touch coordinates to determine how often the user has missed the input areas **12** and increases the size of the input areas **12** when the number of misses passes a predetermined threshold so that the input areas **12** will be easier to click on.

[**0016**] After the auto-resize logic module **28** detects a predetermined threshold of failed touches has been exceeded, then the auto-resize logic module **28** submits size changes to the user presentation and interaction module **24**, and the input areas **12** will be displayed larger than original as shown in **FIG. 3**.

[**0017**] By enlarging the input areas after the a predetermined threshold of failed touches has been exceeded by the user, the system aids the user in making more accurate selections, thereby making the touch screen application easier to use.

[**0018**] **FIG. 4** is a flowchart illustrating the process for dynamically increasing the size of touch screen application input areas in accordance with a preferred embodiment of the present invention. The process begins by displaying the GUI of a touch screen application with input areas in step **50**. In response to user interaction step **52**, it is determined in step **54** whether the touch landed within the coordinates of a clickable area. If so, then in step **56**, the selected input area **12** is activated and processed by the application **26**.

[**0019**] If it is found that the touch failed to land within a clickable area in step **54**, then in step **58**, it is determined whether the user's touch is within a predetermined range of the clickable area **16** (i.e., a near miss). If so, then in step **60**, no action is taken. The predetermined range could be a percentage or multiple of the narrowest dimension of the clickable area **16** beyond the clickable area **16** in each direction.

[**0020**] **FIG. 5** is a diagram illustrating a range boundary around a clickable area for determining whether a touch is a near miss. Assume that the clickable area **16** is X pixels wide by Y pixels high. Y is the narrowest dimension so the range boundary **80** surrounding the clickable area **16** is a box that is a multiple of Y in each direction beyond the clickable area **16**. In this example, the multiple is 1, so that the range boundary **80** will extend Y pixels past the left, right, top and bottom of the clickable area **16**. Therefore, the overall width for the range boundary **80** is X+2Y, and the overall height is 3Y. Touches landing outside the coordinates of the range boundary **80** will be labeled as misses. Touches landing inside the range boundary **80**, but outside the clickable area **16**, will be labeled as near misses. In the case where one range boundary **80** overlaps another range boundary **80**, and a near miss is within both range boundaries **80**, then the near miss is associated with both range boundaries **80**.

[**0021**] If the touch is determined to be a near miss in step **58**, then in step **62** it is determined whether a predetermined number of near misses have been associated with the clickable area **16**. If not, then no action is taken in step **64**. If yes, then in step **66** it is determined whether the number of near misses, or the miss rate, reaches a predetermined threshold, where the miss rate is defined as a percentage of touches falling outside of the clickable area **16** (but inside the range boundary). If no, then no action is taken in step **68**.

[**0022**] If the miss rate has reached the predetermined threshold, then in step **70**, at least one of the input areas **12** including (the viewable area **14** and the clickable area **16**), is enlarged in proportion to the average distance the missed touches are away from the clickable area **16**. For example, if the near misses are offset from the intended input area **12** by 15%, then the size of the input areas **12** may be increased by 15%. In a preferred embodiment, all of the input areas **12** are resized in response to repeated near misses, but in some applications **26**, it may preferable to resize only the intended input area **12**. In an alternative embodiment, all areas of a similar size or type to the intended input area may also be increased in size.

[**0023**] After the new dimensions for the input areas **12** have been calculated, the auto-resize logic module **28** submits the new dimensions to the user presentation and interaction module **24** for redisplay in step **72**, and the process continues as described.

[**0024**] The present invention has been described in accordance with the embodiments shown, and one of ordinary skill in the art will readily recognize that there could be variations to the embodiments, and any variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.

What is claimed is:

1. A method for increasing ease-of-use of a touch screen application, wherein the touch screen application includes a user interface that displays one or more touch screen input areas on a touch sensitive display, each of the touch screen input areas comprising a viewable area and a clickable area, wherein a user's touch of the clickable area activates the corresponding touch screen input area, the method comprising the steps of:

(a) collecting coordinates of the touches on the display in response to user interaction;

(b) analyzing the touch coordinates to determine how often the user has missed the input areas; and

(c) increasing the size of both the viewable area and the clickable area of at least one of the input areas when the number of missed touches reaches a predetermined threshold so that the input areas will be easier to touch.

2. The method of claim 1 wherein step (c) further includes the step of:

increasing the size of all the input areas.

3. The method of claim 2 wherein step (c) further includes the step of:

increasing the size of the input areas in proportion to an average distance the missed touches are away from the input areas.

4. The method of claim 3 wherein step (b) further includes the step of:

determining how often the user has missed each of the input areas.