

PAN AND SELECTION GESTURE DETECTION

BACKGROUND

[0001] A current paradigm for navigating through various information contexts is windows based. A classic example of this is the web browser experience. A user might begin with a home page that occupies the entire browser space. The user might then select a hyperlink, whereupon a new window appears. However, the previous window either disappears or, in the case of exercising an option to open the new page in a new window, the previous window is fully, or at least partially, hidden.

[0002] The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practiced.

SUMMARY

[0003] At least some embodiments described herein relate to a user interface that has a canvas that is extendable in an extendable dimension and on which multiple selectable elements may be placed. In this context, when a selecting control is associated with a particular selectable element, and further user gestures are provided, a pan detection mechanism is configured to distinguish between an intent to pan the canvas and an intent to select the particular selectable element. In another embodiment, the pan detection mechanism is configured to distinguish between an intent to pan the canvas along its extendible direction (such as horizontally), and an intent to scroll a user interface element perpendicular (such as vertically) to the extendible direction of the canvas. In yet another embodiment, the pan detection mechanism is configured to distinguish between all three intents; that is 1) an intent to pan the canvas in its extendible direction, 2) an intent to scroll a user interface element perpendicular to the extendible direction of the canvas, and 3) an intent to select a user interface element.

[0004] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of

the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0006] Figure 1 abstractly illustrates an example computing system in which the principles described herein may be employed;

[0007] Figure 2 abstractly illustrates an environment in which a canvas which has several selectable user interface elements and is pannable in the extendible direction (e.g., horizontal) of the canvas, but several selectable elements are pannable (or scrollable) also in the direction orthogonal (e.g., vertical to the extendible direction) to the extendible direction of the canvas;

[0008] Figure 3 illustrates a supporting architecture for the mechanism for distinguishing between a panning and selection gestures in accordance with the principles described herein;

[0009] Figure 4 illustrates a much more detailed user interface that includes a canvas populated by a large number of selectable elements;

[0010] Figure 5 illustrates a user interface that is similar to the user interface of Figure 4, except that the arc associated with pointer movement is illustrated;

[0011] Figure 6 illustrates a user interface in which an example pan start region is illustrated with respect to the pan-start position; and

[0012] Figure 7 illustrates a flowchart of a method for permitting user interaction with a canvas that extends in an extendable dimension on the user interface, and which is be populated with selectable elements.

DETAILED DESCRIPTION

[0013] At least some embodiments described herein relate to a user interface that has a canvas that is extendable in an extendable dimension and on which multiple selectable elements may be placed. In this context, when a selecting control is associated with a particular selectable element, and further user gestures are provided, a pan detection mechanism is configured to distinguish between an intent to pan the canvas and an intent to select the particular selectable element. Alternatively or in addition, when a selecting control is associated with a particular orthogonally scrollable element that may be scrolled orthogonal to the extendible direction of the canvas, the pen detection mechanism may be

distinguish between an intent to pan the canvas and an intent to scroll the orthogonally scrollable element. Some introductory discussion of a computing system will be described with respect to Figure 1. Then, example user interfaces, methods and supporting architectures will be described with respect to subsequent figures.

5 **[0014]** Computing systems are now increasingly taking a wide variety of forms. Computing systems may, for example, be handheld devices, appliances, laptop computers, desktop computers, mainframes, distributed computing systems, or even devices that have not conventionally been considered a computing system. In this description and in the claims, the term “computing system” is defined broadly as including any device or system
10 (or combination thereof) that includes at least one physical and tangible processor, and a physical and tangible memory capable of having thereon computer-executable instructions that may be executed by the processor. The memory may take any form and may depend on the nature and form of the computing system. A computing system may be distributed over a network environment and may include multiple constituent computing systems.

15 **[0015]** As illustrated in Figure 1, in its most basic configuration, a computing system 100 typically includes at least one processing unit 102 and memory 104. The memory 104 may be physical system memory, which may be volatile, non-volatile, or some combination of the two. The term “memory” may also be used herein to refer to non-volatile mass storage such as physical storage media. If the computing system is
20 distributed, the processing, memory and/or storage capability may be distributed as well. As used herein, the term “executable module” or “executable component” can refer to software objects, routines, or methods that may be executed on the computing system. The different components, modules, engines, and services described herein may be implemented as objects or processes that execute on the computing system (e.g., as
25 separate threads).

[0016] In the description that follows, embodiments are described with reference to acts that are performed by one or more computing systems. If such acts are implemented in software, one or more processors of the associated computing system that performs the act direct the operation of the computing system in response to having executed computer-executable instructions. For example, such computer-executable instructions may be embodied on one or more computer-readable media that form a computer program product. An example of such an operation involves the manipulation of data. The computer-executable instructions (and the manipulated data) may be stored in the memory 104 of the computing system 100. Computing system 100 may also contain
30

communication channels 108 that allow the computing system 100 to communicate with other message processors over, for example, network 110.

[0017] The computing system 100 also includes a display 112 on which a user interface, such as the user interfaces described herein, may be rendered. Such user interfaces may be generated in computer hardware or other computer-represented form prior to rendering. The presentation and/or rendering of such user interfaces may be performed by the computing system 100 by having the processing unit(s) 102 execute one or more computer-executable instructions that are embodied on one or more computer-readable media. Such computer-readable media may form all or a part of a computer program product.

[0018] Embodiments described herein may comprise or utilize a special purpose or general-purpose computer including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Embodiments described herein also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are physical storage media. Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the invention can comprise at least two distinctly different kinds of computer-readable media: computer storage media and transmission media.

[0019] Computer storage media includes RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other tangible medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0020] A “network” is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links which can be used to carry or desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

Combinations of the above should also be included within the scope of computer-readable media.

[0021] Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to computer storage media (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less volatile computer storage media at a computer system. Thus, it should be understood that computer storage media can be included in computer system components that also (or even primarily) utilize transmission media.

[0022] Computer-executable instructions comprise, for example, instructions and data which, when executed at a processor, cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0023] Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, pagers, routers, switches, and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0024] Figure 2 abstractly illustrates an environment 200 in which a portion 201A of a canvas 201 is displayed on a displayed area 210 of a display, and a portion 201B of the

canvas 201 is not displayed in the displayed area 210 of the display. As an example, the display might be, for instance, the display 112 of the computing system 100 of Figure 1. However, the display 200 might alternatively be a window within a display 112, so as not to occupy the entire area of the display 112. The canvas 201 extends in an extendable direction 221 along a single extendable dimension 222. For instance, in Figure 2, the extendable direction 221 is rightward, and the extendable dimension 222 is the horizontal dimension.

[0025] In some embodiments, the extendable dimension 222 of the canvas 201 may be configurable to be vertical or horizontal according to the tastes of the user. In other cases, the limitations of the display 210 or container might restrict extendable dimension 222 to the horizontal dimension, or to the vertical dimension. Also, the extendable direction 221 may be configurable. For instance, if the extendable dimension 222 is vertical, the user might configure the canvas to extend downward, or upward, according to the user's preference. If the extendable dimension 222 is horizontal (as is the case of Figure 2), the user might configure the canvas to extend leftward, or rightward (as is the case of Figure 2), according to the user's preference.

[0026] The canvas 201 is illustrated as including multiple selectable elements 202. For instance, the selectable elements 202 in this particular case are illustrated as including selectable elements 202A through 202F. Some of the selectable elements 202A and 202B are within the displayed area 201A of the canvas 201, and some of the selectable elements 202D through 202F are in the non-displayed area 201B of the canvas 201. One of the selectable elements 202C is illustrated as spanning a boundary between the displayed portion 201A and the undisplayed portion 201B of the canvas 201. Hereinafter, the portion of the canvas that contains elements (whether selectable or otherwise) will be referred to as the "populated portion" of the canvas 201. The selectable element might also include navigation controls.

[0027] When the populated portion of the canvas 201 is too large to display all at one time, the user interface may have a panning mechanism that permits the user to pan the canvas 201 along the extensible dimension. Two of the selectable elements 202B and 202E are illustrated as having a greater extent in a dimension orthogonal to the extendable direction (hereinafter an "orthogonal dimension"). In Figure 2, the orthogonal dimension is the vertical dimension represented by bi-directional arrow 223. When a particular element has a greater extent in the orthogonal dimension than the display is capable of

rendering at one time, the particular element itself may be pannable in the orthogonal dimension.

[0028] In accordance with the principles described herein, after a selecting control (such as a mouse cursor) is associated with a selectable element (such as by pressing the left mouse button of the mouse while the mouse cursor is displayed over the selectable element), a pan detection mechanism distinguishes between a canvas pan gesture and an element selection gesture through subsequent interaction with the selectable element. For instance, in the canvas 201, perhaps the user may have the experience of being able to pan the canvas by dragging the canvas 201 along the extendable dimension 222. Yet, there are selectable elements 202 throughout the populated portion of the canvas 201. The pan detection mechanism allows the user to pan the canvas 201 by activating the selecting control on any portion of the canvas 201 even if that portion happens to be occupied by a selectable element. Thus, if the user wants to pan, the user need not be concerned about avoiding selection of elements. Instead, the user may provide a natural panning gesture on any portion of the canvas. On the other hand, if the user desires to select a particular selectable element, the user provides a selection gesture while the selecting control is associated with that particular selectable element.

[0029] Additionally, after a selecting control (such as a mouse cursor) is associated with an orthogonally scrollable element that is scrollable perpendicular to the extendible dimension of the canvas, the pan detection mechanism distinguishes between a canvas pan gesture and an element scroll gesture. For instance, in the canvas 201, perhaps the user may have the experience of being able to pan the canvas by dragging the canvas 201 along the extendable dimension 222. Yet, there are orthogonally scrollable elements 202B and 202E throughout the populated portion of the canvas 201. The pan detection mechanism distinguishes between an intent to pan the canvas 201 in the extendible dimension, and an intent to scroll the orthogonally scrollable element 202B or 202E associated with the selecting control. Furthermore, in this case, if the selecting control is associated with a selectable element within that orthogonally scrollable element, the panning control distinguishes between intents to pan the canvas in the extendible dimension, scroll the orthogonally scrollable element in the orthogonal dimension, or select the selectable element.

[0030] Figure 3 illustrates a supporting architecture 300 for a user interface 301. For instance, the user interface 301 is an example of the user interface 200 of Figure 2. The supporting architecture 300 is illustrated as including a selecting control 311 that is

capable of selecting any of the selectable elements within the canvas. A pan detection mechanism 312 may be used to distinguish between a pan gesture, an orthogonally scrollable element scroll gesture, and an element selection gesture. An activation module 321 may be used to respond to the selection gesture by selecting the appropriate selectable element. A pan module 322 may be used to pan the canvas in response to the pan detection mechanism 312 detecting a pan gesture and/or scroll the orthogonally scrollable element in response to the orthogonally scrollable element scroll gesture.

[0031] For instance, suppose the selecting control is over the selectable element 202A on the canvas 201 when the selecting control is activated. Depending on the movement of the selecting control during this activation, the user may provide a canvas pan gesture if the user desires to pan the canvas 201 along the extendable dimension 222, or the user may provide an element selection gesture if the user desires to select the selectable element 202A.

[0032] As mentioned above, there are even more possibilities for user intent if the selecting control is over the selectable element 202B on the canvas 201 when the selecting control is activated. In this case, depending on the movement of the selecting control during this activation, the user may additionally provide an orthogonal pan gesture (also called herein an orthogonal scroll gesture) if the user desires to pan the selectable element 202B in the orthogonal direction. This option is in addition to the possibility of a canvas pan gesture if the user desires to pan the canvas 201 along the extendable dimension 222, and an element selection gesture if the user desires to select the selectable element 202B.

[0033] Referring to Figure 3, the pan detection mechanism 312 may be used to distinguish between the element selection gesture, the canvas pan gesture, and the orthogonal pan gesture when the selecting control 311 is associated with a selectable element. If the pan detection mechanism 312 detects an element selection gesture, the activation module 321 may be used to respond to the selection gesture by selecting the appropriate selectable element. If the pan detection mechanism 312 detects a canvas pan gesture, the pan module 322 may be used to pan the canvas in the extendable dimension. If the pan detection mechanism 312 detects an orthogonal pan gesture, the pan module 322 may be used to pan the selectable element in the orthogonal dimension.

[0034] The pan detection mechanism 312 may operate in the same manner to detect the canvas pan gesture and the element selection gesture regardless of which of the selectable elements 202 are associated with the selecting control. Furthermore, the pan detection mechanism 312 may operate to pan the canvas in the extendable dimension

when the selecting control is activated and associated with an area of the canvas that is not associated with any of the selectable elements 202. Accordingly, the principles described herein allow the user to pan the canvas 201 through a simple drag operation without having to worry about avoiding selectable elements when grabbing hold of the canvas with the selecting control. Furthermore, when the user desires to select a selectable element, the user may use the same selecting control to do that also.

[0035] Figure 4 illustrates a much more detailed user interface 400 that includes a canvas populated by a large number of selectable elements. Only a few of the selectable elements are labeled as they are more relevant for the discussion herein. In Figure 4, the user has a selecting control in the form of a pointer 401. The pointer 401 is activated at the position illustrated in Figure 4 such that the pointer 401 is associated with the selectable control 402A at the beginning of either a pan or selection gesture (depending on movement of the pointer 401 thereafter while selected).

[0036] In one embodiment, the pan detection mechanism 312 disambiguates the element selection gesture (and interaction gesture as the parts can have hotspots) versus the canvas pan gesture by analyzing the pointer (e.g., mouse) events (e.g., mouse-down, mouse-up and mouse-move events). If the user mouses-down and then moves more than a certain number (X) of pixels (e.g. 4 pixels) then the pan detection mechanism detects a pan gesture, rather than an element selection gesture. If the user mouses-down, then mouses-up, without moving more than X pixels, the pan detection mechanism 312 identifies an element selection gesture. The region defined by the X pixels may be referred to as a “pan-start diamond”, an area that the user may interface with to express their intent, without them ever needing to cognitively recognize that they are expressing this intent.

[0037] For instance, if the selecting control stays within the pan start region until the selecting control is no longer activated, the pan detection mechanism identifies an element selection gesture. On the other hand, if the selecting control goes outside of the pan start region while the selecting control is still activated, the pan detection mechanism identifies at least one of (and potentially both of) the orthogonal pan gesture and the canvas pan gesture pans in accordance with movement of the selecting control. Upon the first exit of the selecting control outside of the pan start region while the selection control remains activated, a selection gesture is no longer possible. Furthermore, if the selecting control was activated over an orthogonally scrollable element, the pan selection mechanism distinguishes between a canvas pan gesture or an orthogonal scroll gesture based on where the selecting control exited the pan start region.

[0038] In order to accommodate users of different mouse-skill and touch experiences, the dimensions of the pan-start diamond may be adjusted by simply changing the width and height of the pan-start diamond. Further, the pan-start diamond may be adjusted to polish the core panning experience. When users pan the canvas, they often do not move
5 their mouse in a perfectly straight line. Rather, they pan in an arc. This arc is more exaggerated when panning left and right because the desired travel distance is often farther.

[0039] Figure 5 illustrates a user interface 500 that is similar to the user interface 400 of Figure 4, except that the arc associated with pointer movement is illustrated. For
10 instance, the pointer has moved continuously from its original position to position 501, 502, 503, 504 and 505 in succession, to create a drag motion 510 that includes an orthogonal dip at or near the pointer pan start position. If this arc is not accounted for, an intent to orthogonally scroll may be falsely inferred. Thus, an orthogonally scrollable element might orthogonally scroll slightly before the canvas begins to pan in the
15 extendable dimension. This might reduce the user's feeling of the quality of the experience.

[0040] In order to compensate for this user tendency, the pan detection mechanism 312 may use a pan start region that is a diamond that is elongated rather than square. In the illustrated case, the pan start region has a greater span in the orthogonal dimension
20 than in the extendable dimension of the canvas. More concretely, in the case of Figure 5, the pan start region is taller than it is wider. This means that the user has to pan a bit more orthogonally than they would in the extendable dimension before the pan detection mechanism detects an orthogonal pan gesture. In doing so, this greatly reduces chance that a user will see an orthogonal pan of a selectable element when the user really intended
25 a canvas scan in the extendable dimension.

[0041] Figure 6 illustrates a user interface 600 in which an example pan start region 601 is illustrated with respect to the pan-start position 401. In this embodiment, the horizontal half-width of the pan-start diamond region is 3 pixels, and the vertical half-height of the pan-start diamond region is 6 pixels. However, the 3 pixel and 6 pixel values
30 are for illustrative purposes only.

[0042] Figure 7 illustrates a flowchart of a method 700 for permitting user interaction with a canvas that extends in an extendable dimension on the user interface, and which is be populated with selectable elements. The method 700 is initiated upon detecting

activation of a selecting control (event 701). For instance, the user might activate a pointer (e.g., press the left mouse button).

[0043] Next, the method 700 defines the pan start region (act 702). Essentially, the pan start region surrounds the area of the user interface where the user first activated the selection control in act 701. The exact shape of the pan start region will define whether or not the pan detection mechanism concludes that the user intended to select the element (if the selecting control stays within the pan start area while the selecting control is active), or intended to pan the canvas or an orthogonally scrollable element (if the selecting control exits the pan start area at any point while activated). Accordingly, the shape of the pan start area should be carefully selected to as to reduce false detections of intent. In some embodiments, the shape of the pan start area may vary by user.

[0044] Furthermore, perhaps the computing system can detect when it has wrongly detected intent for a given user, and adjust 1) the shape and/or size of the pan start area accordingly, and/or 2) the placement of the pan start area around the initial point that the activation control was initially activated. For instance, perhaps the pan start area is initially a diamond that has the initial point of activation at its center. However, with subsequent monitoring of user behavior, a month later, the system may have adjusted the pan start area to have a different shape (perhaps rounded slightly at the tips, or perhaps bulging slightly on one end), and may have perhaps shifted the pan start area rightward slightly with respect to the initial activation point. The computing system may thus experiment to improve prediction of user intent on a per-user basis.

[0045] In any case, while the selection control is active, the user may do any one of a number of things. The pan detection mechanism actively monitors whether the selection control exits the pan start region (decision block 703) and whether or not the selection control is deactivated (decision block 704). These two decisions may occur quite frequently and thus may be considered as a single consolidated decision. At any given time, if the selection control is still activated (“No” in decision block 704) and the selection control does not exit the pan start region (“No” in decision block 703), then there is no significant change in state of the method 700. However, if the selection control is deactivated (“Yes” in decision block 704) while the selection control still has not exited the pan start region (still “No” in decision block 703), the pan detection mechanism determines whether or not the selection control was activated at a selectable control within the user interface (decision block 705). If the selection control was activated at a selectable element (“Yes” in decision block 705), then the pan detection mechanism

causes the selectable element to be activated (act 706). Otherwise, if the selection control was activated at an area of the user interface that is not selectable (“No” in decision block 705), then the method 700 simply ends (act 707). For instance, there might have just been a stray mouse click.

5 [0046] On the other hand, if while the selection control is still activated (from act 701), and the pan detection mechanism detects that the selection control has exited the pan start region (“Yes” in decision block 703), then panning occurs (act 708). For instance, the canvas may be panned as directed by the selection control. As another more complicated example, if the user activated the selection mechanism at an orthogonally scrollable element, the pan detection mechanism may either pan the canvas or scroll the
10 orthogonally scrollable element (or perhaps both), depending on where the selection control exited the pan start region. This panning (act 708) continues as directed by any movement in the selection control, so long as the selection control continues to be activated (“No” in decision block 709). However, when the selection control is
15 deactivated (“Yes” in decision block 709), the method 700 ends.

[0047] Note that there is no need for keeping track of the pan start region once the pan detection mechanism determines that the selection control has exited the pan start region. A pan start region will thus be generated each time the selection control is activated. In some circumstances, the shape of the pan start region may differ according to the
20 circumstances. For instance, as described above, the pan start region may have a different shape, size, and position relative to the point of activation, depending on the user. However, the pan start region might also differ by any one or more of the following circumstances: 1) whether the canvas is already visualizing the whole populated portion of the canvas, 2) whether the canvas is already panned all the way to one end or the other of
25 the population portion of the canvas at one extent of the extendible dimension, 3) whether the selectable control was activated over a selectable element, 4) whether the selectable control was activated over an orthogonally scrollable element, and if so, the scrolled position of the orthogonally scrollable element in the orthogonal dimension, and so forth.

[0048] In one example, the selecting of a selectable control results in an additional
30 selectable control being created. For instance, selecting a selectable control might create a new user interface element that might be added to the populated portion of the canvas in the extendable direction, and which itself might include one or more selectable elements. Such further selectable elements in the new element might be further selected to add yet more user interface elements to the populated portion of the canvas.

[0049] For instance, in Figure 4, the canvas might have originally included only a favorites area 410 that includes multiple selectable elements including selectable element 411. When the selectable element 411 is selected, a new user interface element 420 appears. The user interface element may be thought of as a slice (or blade) of the canvas that, within given dimensions in the extendible direction of the canvas, extends along all or substantially all of the canvas in the orthogonal dimension. For instance, if the extendible direction is horizontal as in the case of Figure 4, a blade is a vertical slice of the canvas. As well as being a selectable element itself, the user interface element 420 might further include selectable elements such as selectable element 421 (and 402A previously mentioned). When the selectable element 421 is selected, a further user interface element 430 (e.g., a blade) appears and as well as being a selectable element itself, might further include selectable elements such as selectable element 431. When the selectable element 431 is selected, a further user interface element 440 (e.g., a blade) appears and as well as being a selectable element itself, might further include selectable elements such as selectable element 441. This may continue to allow the user to engage in a journey building up a canvas of history showing the path taken to get to where the user is presently. Although the journey may begin in a manner that fits well in the displayed region, as the journey adds additional user interface elements, and the populated area of the canvas extends, the journey may quickly exceed the capability of the display to simultaneously visualize.

[0050] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

1. A computer program product comprising one or more computer-readable storage media having thereon computer-executable instructions that are structured such that, when executed by one or more processors of a computing system, cause the computing system to operate a user interface program that is configured to display a user interface on a display of the computing system, the user interface comprising:

a canvas that extends along an extendable dimension on the user interface, and which may be populated by a plurality of selectable elements;

a pan detection mechanism configured to distinguish between a canvas pan gesture and an element selection gesture when a selecting control is associated with a particular selectable element of the plurality of selectable elements,

the user interface program responsive to a canvas pan gesture by panning the canvas along the extendable dimension,

the user interface program responsive to an element selection gesture by selecting the particular selectable element then associated with the selecting control.

2. The computer program product in accordance with Claim 1, the particular selectable element being within an orthogonally pannable element placed within the canvas, the orthogonally pannable element being pannable in an orthogonal dimension that is orthogonal to the extendable dimension of the canvas.

3. The computer program product in accordance with Claim 1, wherein if the selecting control goes outside of the pan start region while the selecting control is still activated, the pan detection mechanism identifies both the orthogonal pan gesture and the canvas pan gesture.

4. The computer program product in accordance with Claim 1, the pan detection mechanism distinguishing between the canvas pan gesture and the element selection gesture using a pan start region that is defined with respect to a position on the canvas of the selecting control when the selecting control is activated,

such that if the selecting control stays within the pan start region until the selecting control is no longer activated, the pan detection mechanism identifies an element selection gesture, and

such that if the selecting control goes outside of the pan start region while the selecting control is still activated, the pan detection mechanism identifies the canvas pan gesture.

5. The computer program product in accordance with Claim 1, the pan detection mechanism operating in a same manner to detect the canvas pan gesture and the element selection gesture regardless of which of the plurality of selectable elements are associated with the selecting control.

6. The computer program product in accordance with Claim 1, the pan detection mechanism operating to pan the canvas when the selecting control is activated and associated with the canvas at an area that is not associated with any of the plurality of selectable elements.

7. A method for permitting user interaction with a canvas that extends in an extendable dimension on the user interface, and which is be populated with a plurality of selectable elements, the method comprising:

an act of detecting that a selecting control is associated with a particular selectable element of the plurality of selectable elements on the canvas; and

an act of distinguishing between a canvas pan gesture and an element selection gesture when the selecting control is still associated with the particular selectable element of the plurality of selectable elements.

8. The method in accordance with Claim 7, wherein the act of distinguishing results in an identification of a canvas pan gesture, the method further comprising:

an act of panning the canvas in the extendable dimension in accordance with movement of the canvas pan gesture in the extendable dimension.

9. The method in accordance with Claim 7, wherein the act of distinguishing results in an identification of an element selection gesture, the method further comprising:
an act of selecting the particular selectable element.

10. The method in accordance with Claim 9, further comprising the following in response to the act of selecting the particular selectable element:

an act of generating a first blade on the canvas, the first blade including multiple selectable elements, each of at least some of the multiple selectable elements having at least one corresponding blade such that if the corresponding element is selected, the corresponding blade appears on the canvas as a second blade on the canvas, wherein the second blade is contiguous with the first blade in the extendable direction of the canvas.