June 2022

GENERATING ROUNDED CORNERS

Chet Haase

GENERATING ROUNDED CORNERS

ABSTRACT

A computing device (e.g., a smartphone, a laptop computer, a tablet computer, a smartwatch, etc.) may use a Bezier curve (e.g., a parametric curve) or a series of Bezier curves (i.e., a Bezier spline) to determine a path object (e.g., data or code specifying nodes or points that form a path) that traverses (e.g., coincides, corresponds in position to, etc.) the outer border edge of a corner of a graphical element. The shape of the Bezier curves may be based on a number of customizable parameters, including, but not limited to, a radius parameter, a circularity parameter, a smoothness parameter, and a roundness parameter. By leveraging one or more Bezier curves, the techniques allow for the generation of rounded corners that are smooth and continuous, which may improve the visual appearance of the graphical element. In addition, by providing customizable parameters, the techniques allow for greater flexibility in the design of rounded corners, which may also improve the visual appearance of the graphical element.

DESCRIPTION

FIG. 1 below is a conceptual diagram illustrating a system 100 that generates a graphical element with one or more rounded corners in accordance with techniques of this disclosure. As shown in FIG. 1, computing device 100 includes one or more presence-sensitive displays 102 (“display 102”), one or more processors 104, and one or more storage devices 106. As further shown in FIG. 1, storage devices 106 stores a rounded corner module 108 and a graphical user interface module 110 (“GUI module 110”), and one or more applications 112.

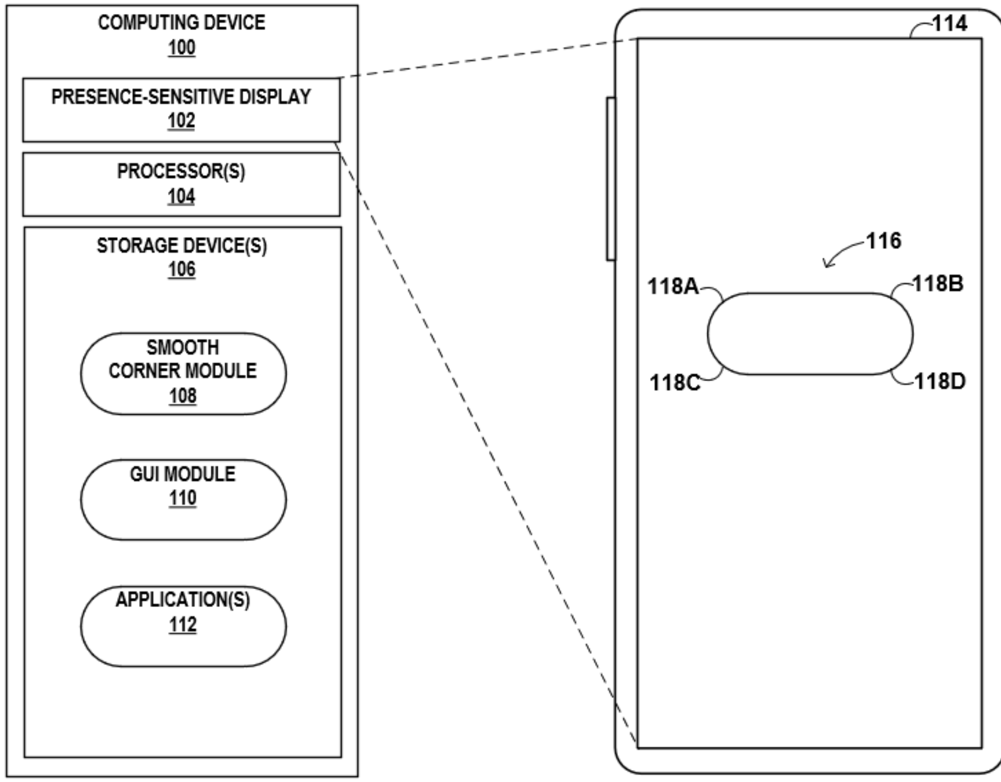

FIG. 1

Display 102 of computing device 100 may be a display that functions as an output device. For example, display 102 may function as an output (e.g., display) device using any of one or more display components, such as a liquid crystal display (LCD), dot matrix display, light emitting diode (LED) display, microLED display, organic light-emitting diode (OLED) display, e-ink, active-matrix organic light-emitting diode (AMOLED) display, or similar monochrome or color display capable of outputting visible information to a user of computing device 100. As further described here, GUI module 110 may obtain instructions from GUI module 110 and, according to those instructions, cause display 102 to output visible information, such as a graphical user interface 114 ("GUI 114"). GUI 114 may include one or more graphical elements, such as graphical element 116.

Processors 104 may implement functionality and/or execute instructions associated with computing device 100. Examples of processors 104 include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device. In some examples, processors 104 may represent a system on a chip (SoC) that includes an integrated circuit for implementing one or more of the above referenced examples of processors 104, along with supporting memory and/or storage, and possibly various interfaces, modems, etc. as a single package.

Storage devices 106 may include one or more computer-readable storage media. For example, storage devices 106 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by computing device 100. In some examples, storage devices 106 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard disks, optical discs, solid state discs, and/or the like. In other examples, in place of, or in addition to the non-volatile storage elements, storage devices 106 may include one or more so-called “temporary” memory devices, meaning that a primary purpose of these devices may not be long-term data storage. For example, the devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random-access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), etc.

In general, applications 112, in conjunction with GUI module 110, may display GUI 114. GUI 114 may include graphical element 116, which may in turn include one or more corners, such as corners 118A-118D (“corners 118”). In some examples, applications 112 may round corners 118 (resulting in rounded, or smooth, corners). For instance, applications 112 may round a corner of an outer border edge of graphical element 116 based on a border radius parameter. Rounding a corner based on one border radius parameter may result in a circular corner, and rounding a corner based on two border radius parameters may result in an elliptical corner. In some cases, corners that are neither circular nor elliptical but still round may be desirable, but generating such corners may not be possible using the border radius parameter. Thus, conventional techniques for generating rounded corners may be limiting and even unsuitable in those cases.

In accordance with techniques of this disclosure, rounded corner module 108 may round corners based on one or more Bezier curves. For example, rounded corner module 108 may use one or more Bezier curves to determine a path object (e.g., data or code specifying nodes or points that form a path) that traverses (e.g., coincides, corresponds in position to, etc.) the outer border edge of a corner (e.g., corner 118A) of graphical element 116. The shape of the Bezier curves may be based on a number of customizable parameters, including, but not limited to, a radius parameter, a circularity parameter, a smoothness parameter, and a roundness parameter. By leveraging Bezier curves, the techniques allow for the generation of rounded corners that are smooth and continuous, which may improve the visual appearance of graphical element 116. In addition, by providing customizable parameters, the techniques allow for greater flexibility in the design of rounded corners, which may also improve the visual appearance of graphical element 116.

As noted above, rounded corner module 108 may round corners based on one or more Bezier curves. As used here, a Bezier curve refers to a parametric curve defined by a set of control points P_0 through P_n, where n is called the order of the curve (n = 1 for linear, 2 for quadratic, 3 for cubic, etc.). The first and last control points of a Bezier curve may be the endpoints of the Bezier curve and are referred to here as anchor points. In examples where rounded corner module 108 rounds corners based on a series of multiple Bezier curves, the multiple Bezier curves may form a Bezier spline.

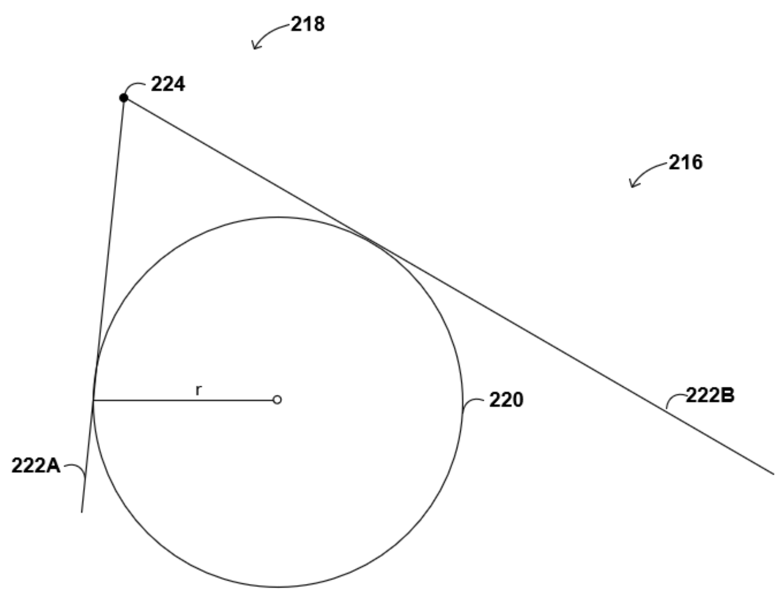

FIG. 2

FIG. 2 is a conceptual diagram of a corner 218 of a graphical element 216. Rounded corner module 108 may round corner 218 based on inputted parameters. For example, rounded corner module 108 may use a radius parameter to construct a circle 220 having a radius *r* equal to the radius value and a perimeter tangential to (at least two) outer border edges of graphical element 216, such as edges 222A-222B (“edges 222”). Circle 220 may serve as a starting point for the rounding of corner 218 (i.e., rounded corner module 108 may perform additional steps to round corner 218).

Where rounded corner module 108 positions circle 220 with respect to graphical element 216 may be constrained by the radius value and the angle edges 222 form at a vertex 224 of corner 216. In general, a larger radius value and smaller angle may result in the center of circle 220 being further from vertex 224. Conversely, a smaller radius value and larger angle may result in the center of circle 220 being closer to vertex 224.

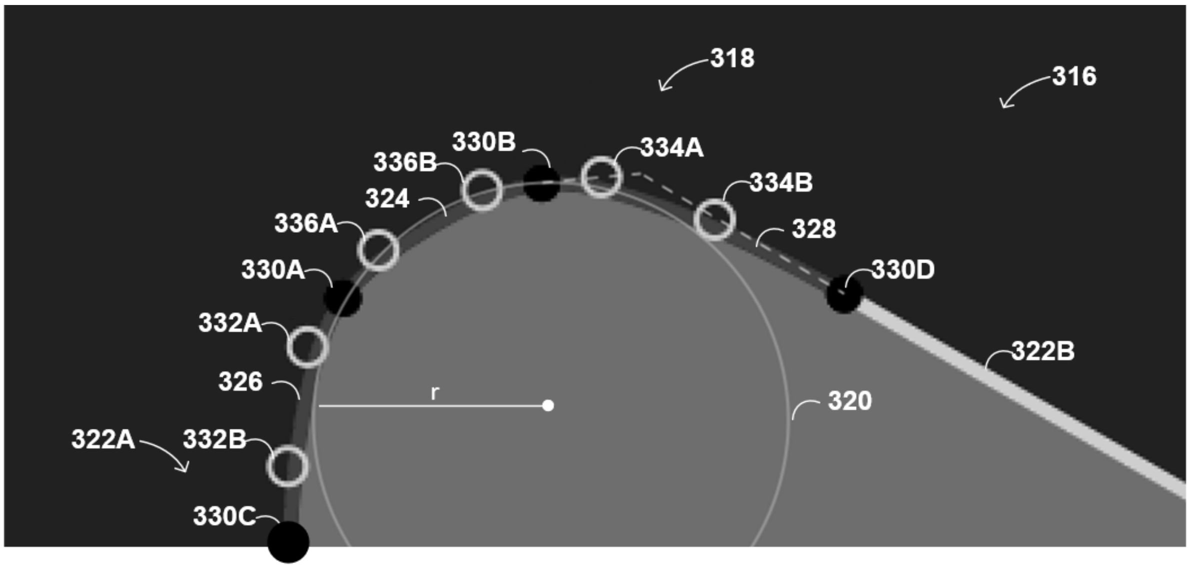

FIG. 3

FIG. 3 is a conceptual diagram of a corner 318 of a graphical element 316. As shown in FIG. 3, rounded corner module 108 may determine a path object including a series of multiple Bezier curves (i.e., a Bezier spline), where the path object defines the rounding geometry for corner 318. In the example of FIG. 3, the path object includes three cubic Bezier curves: a middle Bezier curve 324, a first flanking Bezier curve 326, and a second flanking Bezier curve 328. However, a path object may include more or fewer Bezier curves. For instance, a path object may include 1, 2, 4, etc., Bezier curves. Furthermore, the order of the curve of a Bezier curve may be larger or smaller than 3. For instance, the order of the curve may be 2, 4, etc. Thus, examples of path objects and Bezier curves other than the one illustrated in FIG. 3 are contemplated by this disclosure.

First flanking Bezier curve 326 may be on a first side (e.g., a left side) of middle Bezier curve 324, and second flanking Bezier curve 328 may be on a second side (e.g., a right side) of middle Bezier curve 324. In the example of FIG. 3, middle Bezier curve 324, first flanking Bezier curve 326, and second flanking Bezier curve 328 are cubic Bezier curves. As a result, middle Bezier curve 324, first flanking Bezier curve 326, and second flanking Bezier curve 328 may each have four control points, two of which are anchor points (i.e., control points that are endpoints) and two of which are intermediate control points. Middle Bezier curve 324 may approximately (e.g., substantially, generally, closely) traverse an arc of circle 320. First flanking Bezier curve 326 and second flanking Bezier curve 328 may be symmetrical and on opposite sides of middle Bezier curve 324.

Because middle Bezier curve 324, first flanking Bezier curve 326, and second flanking Bezier curve 328 combine to form a (continuous) Bezier spline, first flanking Bezier curve 326 and second flanking Bezier curve 328 may each share an anchor point with middle Bezier curve 324. For example, as shown in FIG. 3, middle Bezier curve 324 and first flanking Bezier curve 326 share an anchor point 330A, and middle Bezier curve 324 and second flanking Bezier curve 328 share an anchor point 330B. Similarly, because first flanking Bezier curve 326 and second flanking Bezier curve 328 extend (in a smooth and continuous fashion) from edges 322A-322B ("edges 322"), first flanking Bezier curve 326 and second flanking Bezier curve 328 may each share an anchor point with one of edges 322. For example, as shown in FIG. 3, first flanking Bezier curve 326 and edge 322A share an anchor point 330C, and second flanking Bezier curve 328 and edge 322B share an anchor point 330D.

Rounded corner module 108 may round corner 318 based on inputted parameters. For example, rounded corner module 108 may use a circularity parameter to adjust how circular corner 318 appears after rounding. For example, with reference to FIG. 3, adjusting the circularity parameter may cause the positions of anchor points 330A and 330B on circle 320 to change. For example, increasing the circularity parameter may cause the positions of anchor points 330A and 330B on circle 320 to move outwards (with respect to the vertex of corner 318), increasing the arc length of the arc that middle Bezier curve 324 traverses. This may result in a more circular path object and therefore a more circular appearance of corner 318. Conversely, decreasing the circularity parameter may cause the positions of anchor points 330A and 330B on circle 320 to move inwards (with respect to the vertex of corner 318), decreasing the arc length of the arc that middle Bezier curve 324 traverses. This may result in a less circular path object and therefore a less circular appearance of corner 318.

In another example, rounded corner module 108 may use a smoothness parameter to determine how gradually edges 322 transition into first flanking Bezier curve 326 and second flanking Bezier curve 328. For example, with reference to FIG. 3, adjusting the smoothness parameter may cause the positions of anchor points 330C and 330D on edges 322 to change. For example, increasing the smoothness parameter may cause the positions of anchor points 330C and 330D on edges 322 to move outwards (with respect to the vertex of corner 318), resulting in a more gradual (and lengthy) transition from edges 322 into first flanking Bezier curve 326 and second flanking Bezier curve 328. This may produce a smoother appearance of corner 318. Conversely, decreasing the smoothness parameter may cause the positions of anchor points 330C and 330D on edges 322 to move inwards (with respect to the vertex of corner 318), resulting in a less gradual (and lengthy) transition from edges 322 into first flanking Bezier curve 326 and second flanking Bezier curve 328. This may produce a more angular appearance of corner 318.

In yet another example, rounded corner module 108 may use a roundness parameter to determine how gradually middle Bezier curve 324 transitions into first flanking Bezier curve 326 and second flanking Bezier curve 328. For example, with reference to FIG. 3, adjusting the smoothness parameter may cause the positions of intermediate control points 332A and 332B of first Bezier curve 326 and the positions of intermediate control points 334A and 334B of second Bezier curve 328 to change. For example, increasing the roundness parameter may cause the positions of intermediate control points 332A and 332B to move further from each other, resulting in a more gradual (and lengthy) transition from middle Bezier curve 324 into first flanking Bezier curve 326. Similarly, increasing the roundness parameter may cause the positions of intermediate control points 334A and 334B to move further from each other, resulting in a more gradual (and lengthy) transition from middle Bezier curve 324 into second flanking Bezier curve 328. This may produce a round appearance of corner 318.

Conversely, decreasing the roundness parameter may cause the positions of intermediate control points 332A and 332B to move closer to each other, resulting in a less gradual (and lengthy) transition from middle Bezier curve 324 into first flanking Bezier curve 326. Similarly, decreasing the roundness parameter may cause the positions of intermediate control points 334A and 334B to move closer to each other, resulting in a less gradual (and lengthy) transition from middle Bezier curve 324 into second flanking Bezier curve 328. This may produce a less round appearance of corner 318.

In some examples, the set of positions of intermediate control point 332A may be on a line tangential to edge 322A. The set of positions of intermediate control point 332B may be on a line tangential to middle Bezier curve 324 at anchor point 332A. The set of positions of intermediate control point 334A may be on a line tangential to edge 322B (as illustrated by a dashed line in FIG. 3). The set of positions of intermediate control point 334B may be on a line tangential to middle Bezier curve 324 at anchor point 332B (as illustrated by a dashed line in FIG. 3). As for intermediate control points 336A and 336B of middle Bezier curve 324, rounded corner module 108 may position intermediate control points 336A and 336B on circle 320 such that middle Bezier curve 324 approximately traverses an arc of circle 320 (per the mathematical formula for a Bezier curve).

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2014/0362089A1. In another example, the techniques of this disclosure may be combined with the techniques described in Mathieu Jouhet, "Rounding polygon corners", Observable, March 31, 2021. In yet another example, the techniques of this disclosure may be combined with the techniques described in Bojan Kverh, "How to get rounded corner shapes in C++ using bezier curves and QPainter: A step-by-step guide", Toptal, February 7, 2022.