United States Patent Application Publication	20250252653
Kind Code	A1
Publication Date	August 07, 2025
Inventor(s)	Cappiello; Will et al.

VISUALIZER FOR SWIMMING POOLS

Abstract

Swimming pools can be visualized according to some aspects described herein. In one example, a system can receive a user selection of a particular liner from among a plurality of liner options for a virtual swimming pool. The system can generate a virtual swimming pool having the particular liner. The virtual swimming pool can be a three-dimensional (3D) rendering of a swimming pool with the particular liner. The system can then output the virtual swimming pool for display on a display device.

Inventors: **Cappiello; Will (Latham, NY), Culp; Joel (Latham, NY), Mocha; Scott (Latham, NY), Reynolds; Scott (Latham, NY)**

Applicant: **Latham Pool Products, Inc.** (Latham, NY)

Family ID: **74418530**

Appl. No.: **19/186492**

Filed: **April 22, 2025**

Related U.S. Application Data

parent US continuation 18680587 20240531 parent-grant-document US 12307574 child US 19186492
parent US continuation 18131231 20230405 parent-grant-document US 12026821 child US 18680587
parent US continuation 17854006 20220630 parent-grant-document US 11645808 child US 18131231
parent US continuation 17139822 20201231 parent-grant-document US 11410373 child US 17854006
us-provisional-application US 62956256 20200101

Publication Classification

Int. Cl.: **G06T15/04** (20110101); **G06F30/13** (20200101); **G06F111/16** (20200101); **G06Q50/08** (20120101); **G06Q50/16** (20240101); **G06T15/20** (20110101)

U.S. Cl.:

CPC **G06T15/04** (20130101); **G06F30/13** (20200101); **G06Q50/08** (20130101); **G06Q50/16** (20130101); **G06T15/20** (20130101); G06F2111/16 (20200101)

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATIONS [0001] This is a continuation of U.S. application Ser. No. 18/680,587 filed on May 31, 2024, now allowed, which is a continuation of U.S. application Ser. No. 18/131,231 filed on Apr. 5, 2023, and patented on Jul. 2, 2024 as U.S. Pat. No. 12,026,821, which is a continuation of U.S. application Ser. No. 17/854,006 filed on Jun. 30, 2022, and patented on May 9, 2023 as U.S. Pat. No. 11,645,808, which is a continuation of U.S. application Ser. No. 17/139,822, filed on Dec. 31, 2020, and patented on Aug. 9, 2022 as U.S. Pat. No. 11,410,373, which claims the benefit of priority under 35 U.S.C. § 119 (e) to U.S. Provisional Patent Application No. 62/956,256, filed Jan. 1, 2020, the entirety of each of which is hereby incorporated by reference herein.

TECHNICAL FIELD
[0002] The present disclosure relates generally to swimming pools. More specifically, but not by way of limitation, this disclosure relates to a visualizer for swimming pools.
BACKGROUND
[0003] Customers generally take into account a variety of considerations when choosing a swimming pool (or “pool”) for their property. For example, customers often consider the pool's size, shape, material, and spatial positioning on the property. Customers may also take into account aesthetic considerations, such as the pool's liner pattern and color. Typically, pool manufacturers provide product catalogues that describe their pool products to aid the customer with these considerations. But visualizing what a pool will look like on the property of the customer still requires a great deal of imagination on the part of the customer.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
[0004] FIGS. **1**-**6** are screenshots of an example of a pool visualization application according to some aspects of the present disclosure.
[0005] FIG. **7** is a block diagram of an example of a computing device for providing a visualizer for swimming pools according to some aspects of the present disclosure.
[0006] FIG. **8** is a flow chart of an example of a process for providing a visualizer for swimming pools according to some aspects of the present disclosure.
[0007] FIG. **9** is a screenshot of an example of a pool visualization application according to some aspects of the present disclosure.
[0008] FIG. **10** is a block diagram of an example of a system for implementing a pool visualization application according to some aspects of the present disclosure.
DETAILED DESCRIPTION

[0009] Certain aspects and features of the present disclosure relate to visualizing a swimming pool on a property using a pool visualization application. In one particular example, a user may wish to install a swimming pool (or "pool") on his property, such as in his backyard or inside his home. To help visualize what the pool would look like, the user can execute a pool visualization application. The pool visualization application can walk the user through a series of steps, during which the user can select various visual characteristics associated with the pool. Examples of the visual characteristics can include a shape, liner, background setting, material, and orientation for the pool. The pool visualization application can then display a three-dimensional (3D) rendering of a virtual swimming pool, where the 3D rendering has the selected visual characteristics. This can aid the user with visualizing what the pool would look like, so that the user can more easily select an appropriate pool.

[0010] In some examples, the pool visualization application can be a website-based application. In some such examples, the virtual swimming pool can be generated within a website displayed in a website browser. At least some of the program code for the pool visualization application can be included in a plugin (e.g., a WordPress plugin), which can be executed by a server hosting the website or a client device accessing the website. In other examples, the pool visualization application can be a mobile application, such as a native application for iPhones® or Android® devices. A user can download the mobile application to a mobile device and execute the mobile application thereon.

[0011] These illustrative examples are given to introduce the reader to the general subject matter discussed here and are not intended to limit the scope of the disclosed concepts. The following sections describe various additional features and examples with reference to the drawings in which like numerals indicate like elements but, like the illustrative examples, should not be used to limit the present disclosure.

[0012] FIG. **1** is a screenshot of an example of a pool visualization application according to some aspects of the present disclosure. In this example, the pool visualization application is configured to generate a graphical user interface (GUI) with one or more pages that enable a user to select one or more visual characteristics associated with a virtual swimming pool. The pool visualization application can then display a rendering of the virtual swimming pool, where the rendering has the selected visual characteristic(s).

[0013] The screenshot of FIG. **1** depicts one exemplary GUI page of the pool visualization application. This GUI page includes a group of background settings **102***a-c* for a virtual swimming pool. In this example, the group of background settings **102***a-c* includes different types of homes, such as a modern ranch home, a traditional colonial home, and a contemporary estate. But other examples can include more, fewer, or different types of background settings. A user can select a particular background setting from among the group of background settings **102***a-c* to move on to a second GUI page.

[0014] One example of the second GUI page is depicted in FIG. **2**. This GUI page includes a group of shapes **202***a-c* for the virtual swimming pool. In this example, the group of shapes **202***a-c* includes a rectangle shape **202***a*, an L shape **202***b*, or a freeform shape **202***c*. But other examples can include more, fewer, or different types of shapes. A user can select a particular shape from among the group of shapes **202***a-c* and the pool visualization application will responsively update a visualization frame **204** to depict a virtual swimming pool having the particular shape within the context of the selected background setting. Once satisfied, the user can select the "continue" button to move on to a third GUI page.

[0015] One example of the third GUI page is depicted in FIG. **3**. This GUI page includes a group of liner options **302** for the virtual swimming pool. Each liner option can have a unique color palette, pattern, or both. The liner options depicted in FIG. **3** are not intended to be limiting, and other examples may have more, fewer, or different liner options. While the liner options are not listed in any particular order in FIG. **3**, other examples can have liner options organized by color palette,

pattern, pool water color, etc., and the user may be able to filter through the liner options based on those criteria.

[0016] A user can select a particular liner from among the liner options **302** and the pool visualization application will responsively update a visualization frame **304** (e.g., in real time) to depict at least a portion of a virtual swimming pool having the particular liner. The user can also select whether view the particular liner with or without the tiling (e.g., faux tiling) near the water's edge. For example, the user can select a tiling button to view the liner with the tiling, or deselect the tiling button to view the liner without the tiling. Once satisfied, the user can select the "SEE MY POOL!" button to move on to a fourth GUI page.

[0017] One example of the fourth GUI page is depicted in FIG. **4**. This GUI page includes depicts the virtual swimming pool **402** in a visualization frame **404**. The virtual swimming pool **402** has the selected shape and liner. The virtual swimming pool **402** is also depicted within the context of the selected background setting (e.g., in a virtual scene having the selected background setting). This can help a user visualize what a swimming pool having the selected visual characteristics may look like.

[0018] To get a better sense of what the swimming pool might look like from other angles, the pool visualization application can enable the user to select among a group of viewing angle options **406***a-d*. Each of the viewing angle options **406***a-d* depicts the virtual swimming pool **402** from a different perspective (e.g., within the selected background setting). One example of viewing angle option **406***a* is depicted in FIG. **5**, which shows the virtual swimming pool **402** from the perspective of a person positioned in the water and looking toward a home in the selected background setting. And one example of viewing angle option **406***b* is depicted in FIG. **9**, which shows a close-up view of the virtual swimming pool **402** near its entryway, to enable a user to more closely inspect how the selected liner might look on the entry stairs of a swimming pool. FIG. **9** also includes hotspots, each of which is designated by a cross-hatched circle (but any other suitable visual indicator can alternatively be used). A hotspot is a location with which a user can interact to obtain more information about a corresponding feature of the virtual swimming pool **402**. In the example shown in FIG. **9**, the user has hovered a cursor over one of the hotspots and the pool visualization application has responsively generated an information box providing more details ("High-quality faux tiling provides elegance and sophistication") about a faux tiling feature corresponding to the hotspot. Although only FIG. **9** depicts hotspots, any of the viewing angle options **406***a-d* can include one or more hotspots. Returning to FIG. **4**, viewing angle option **406***c* depicts the virtual swimming pool **402** from the perspective of a person positioned in the water and looking in a direction that is opposite to the home in the selected background setting. And viewing angle option **406***d* is the perspective currently shown in the visualization frame **404** of FIG. **4**. But other examples may have more, fewer, or different viewing angle options **406***a-d*. A user can select a particular viewing angle from among the group of viewing angle options **406***a-d* and the pool visualization application will responsively update the visualization frame **404** (e.g., in real time) to depict the virtual swimming pool from the selected viewing angle.

[0019] In some examples, the fourth GUI page may also include a mode selector **414** that enables a user to activate a day mode or a night mode. In day mode, the pool visualization application depicts the virtual swimming pool **402** at a selected viewing angle during the day time (e.g., with blue skies and sunshine). In night mode, the pool visualization application depicts the virtual swimming pool **402** at the selected viewing angle at night time. Switching between these modes can allow the user to view the virtual swimming pool **402** at different times of day, which can help in the decision-making process. In some examples, the pool visualization application can switch between the modes by accessing a group of pool images (e.g., generated beforehand), where the group of pool images can depict the virtual swimming pool at each viewing angle both during the day and at night. The pool visualization application can then obtain and render the appropriate pool image based on the selected viewing angle and the selected mode.

[0020] Some liners can alter the overall color hue of a swimming pool. For example, an installed liner may have properties that interact with the water and sunlight to generate an overall color hue that is perceptibly different from the color of the liner itself. In some examples, the pool visualization application can replicate this phenomenon to provide the user with a more realistic sense of what a swimming pool with the selected liner may look like in the real world. For example, the pool visualization application can determine a color hue for the virtual swimming pool based on water properties such as a depth property, a refraction property, and a reflection property; liner properties such as a color property, a material property, and a pattern property; sun properties such as a position property and a strength property; or any combination of these. The pool visualization application may receive values for some or all of these properties as user input (e.g., through the GUI). Additionally or alternatively, the pool visualization application can have default values for some or all of these properties. For example, the pool visualization application may be preprogrammed with default values for the water depth, refraction, and reflection properties. The pool visualization application may also be preprogrammed with default values for the sun properties. For example, the pool visualization may be preprogrammed with a first set of sun-property values for use in the day mode, whereby the sun is positioned at a first theoretical location in the sky and has a first theoretical strength. The pool visualization may also be preprogrammed with a second set of sun-property values for use in the night mode, whereby the sun is positioned at a second theoretical location in the sky and has a second theoretical strength. The pool visualization application can also have access to a database that correlates each liner option to its respective liner properties. The pool visualization application can access the database to determine the liner properties for the selected liner. After the values for the various properties have been determined, the pool visualization application can determine the color hue for the virtual swimming pool based on the property values, for example by using an algorithm. The pool visualization application can then render the virtual swimming pool with the determined color hue.
[0021] The fourth GUI page can also include other GUI objects, such as the menu button **408**. The menu button **408** can enable the user to modify the background setting, shape, and liner associated with the virtual swimming pool. For example, the pool visualization application can respond to a user interaction with the menu button **408** by presenting a menu having a first menu option through which a user can change the background setting for the virtual swimming pool **402**, a second menu option through which the user can change the shape of the virtual swimming pool **402**, and a third menu option through which the user can change the liner for the virtual swimming pool **402** (or choose whether the liner should be depicted with or without the tiling near the water's edge).
[0022] In some examples, the menu button **408** can also enable the user to find a supplier of pool products with the visual characteristics of the virtual swimming pool **402**. For example, the pool visualization application can detect a user interaction with a menu option for finding a dealer and responsively filter through a list of swimming pool suppliers to identify a group of suppliers that sell pool products with the selected visual characteristics of the virtual swimming pool **402**. The pool visualization application can then filter through the group of suppliers based on their proximity to the user (e.g., based on an address or zip code input by the user, or based on a GPS location of the handheld device **100**) to identify to a subset of suppliers within a predefined distance of the user. Of course, other examples can perform this filtering process in the opposite order—e.g., by first filtering by proximity and then filtering by product selection. Either way, the pool visualization application can identify appropriate suppliers and output the identified suppliers to the user.
[0023] In some examples, the fourth GUI page can also include an expand button **410**, which is selectable by a user to enlarge the visualization frame **404** (e.g., to full screen) depicting the virtual swimming pool **402**. This may help the user more easily see some of the finer details in the rendering of the virtual swimming pool **402**.
[0024] The fourth GUI page can additionally or alternatively include a change liner button **412**,

through which the user can change the liner for the virtual swimming pool **402**. This can serve as an alternative way in which the user can change the liner, which may be easier and faster than having to go through the menu associated with menu button **408**. For example, the pool visualization application can detect a user interaction with the change liner button **412** and responsively present an image carousel with a variety of liner options. An example of such an image carousel **602** is shown in FIG. **6**. The user can cycle through the various liner options **604** in the image carousel **602** and select a desired liner. The pool visualization application can receive the selected liner and responsively update the virtual swimming pool **402** in the visualization frame **404** to depict the selected liner. The user may also be able to interact with a tiling button to select whether the liner should be depicted with or without the tiling near the water's edge.

[0025] In some examples, the fourth GUI page can include a pool cover selector through which a user can select a pool cover for the virtual swimming pool **402**. For example, the user can interact with the pool cover selector and the pool visualization application can responsively output a group of pool cover options having varying sizes, shapes, textures, patterns, and colors. In some examples, the pool visualization application can select the group of pool cover options to present to the user based on one or more characteristics (e.g., the size and shape) of the virtual swimming pool, so that the pool cover options conform to those characteristics. A user can select a particular pool cover from among the group of pool cover options and the pool visualization application can responsively update the visualization frame **404** to depict the selected pool cover on the virtual swimming pool **402**. This can enable a user to view what the swimming pool would look like when covered, e.g., during winter months.

[0026] While FIGS. **1**-**6** depict a sequence of GUI pages in a particular order having particular GUI objects, these examples are not intended to be limiting. Other examples can involve more, fewer, different, or a different arrangement of GUI pages and GUI objects.

[0027] As alluded to above, the GUI can be output on a computing device at least in part by executing the pool visualization application. One example of such a computing device **700** is shown in FIG. **7**. In some examples, the computing device **700** can be a server, a mobile device (e.g., a mobile phone, tablet, or laptop computer), or a desktop computer.

[0028] The computing device **700** includes a processor **702** communicatively coupled to a memory **704** by a bus **706**. The processor **702** can include one processor or multiple processors. Non-limiting examples of the processor **702** include a Field-Programmable Gate Array (FPGA), an application-specific integrated circuit (ASIC), a microprocessor, etc. The processor **702** can execute instructions **708** stored in the memory **704** to perform operations. In some examples, the instructions **708** can include processor-specific instructions generated by a compiler or an interpreter from code written in any suitable computer-programming language, such as C, C++, C #, etc.

[0029] The memory **704** can include one memory device or multiple memory devices. The memory **704** can be non-volatile and may include any type of memory device that retains stored information when powered off. Non-limiting examples of the memory **704** include electrically erasable and programmable read-only memory (EEPROM), flash memory, or any other type of non-volatile memory. At least some of the memory device includes a non-transitory computer-readable medium from which the processor **702** can read instructions **708**. A non-transitory computer-readable medium can include electronic, optical, magnetic, or other storage devices capable of providing the processor **702** with the instructions **708** or other program code. Non-limiting examples of a non-transitory computer-readable medium include magnetic disk(s), memory chip(s), ROM, random-access memory (RAM), an ASIC, a configured processor, optical storage, or any other medium from which a computer processor can read the instructions **708**.

[0030] The instructions **708** can include a pool visualization application **718**. The pool visualization application **718** may be executable by the processor **702** for causing the processor **702** to perform any of the functionality described herein. The pool visualization application **718** may also be

executable by the processor **702** to perform more functionality, less functionality, or different functionality than described herein.

[0031] The computing device **700** also includes input components. One example of an input component is the user input device **710**, which may include one user input device or multiple user input devices. Examples of such user input devices can include a mouse, a keyboard, a touchpad, and a touch-screen. Another example of an input component is the sensor **712**, which may include one sensor or multiple sensors. Examples of such sensors can include a global positioning system (GPS) unit, a gyroscope, an accelerometer, an inclinometer, and a camera.

[0032] The computing device **700** further includes output components. One example of an output component is the display device **714**, which may include a liquid crystal display (LCD) or a light-emitting diode (LED) display. The computing device **700** may also include audio output components, like a speaker, and haptic output components, like a haptic actuator. But for simplicity, the audio and haptic output components are not shown in FIG. **7**.

[0033] While FIG. **7** depicts the components (e.g., processor **702**, display device **714**, sensor **712**, etc.) internal to a single housing, in other examples the components may be distributed and in wired or wireless communication with one another. Additionally, although FIG. **7** depicts a certain number and arrangement of components, this is for illustrative purposes and not intended to be limiting. Other examples can include more components, fewer components, different components, or a different arrangement of the components shown in FIG. **7**.

[0034] FIG. **8** is a flow chart of an example of a process for providing a visualizer for swimming pools according to some aspects of the present disclosure. While FIG. **8** depicts a certain sequence of steps for illustrative purposes, other examples can involve more steps, fewer steps, different steps, or a different order of the steps depicted in FIG. **8**. The steps of FIG. **8** are described below with reference to components of FIG. **7** described above.

[0035] In block **802**, a processor **702** receives a user selection of one or more visual characteristics associated with a virtual swimming pool. Examples of the visual characteristics can include a background setting, a liner, a shape, a viewing angle, an orientation, a material, or any combination of these. For example, the processor **702** can output thumbnail images of a plurality of liner options in a GUI on a display device **714**. The processor **702** can then detect a user selection of a thumbnail image corresponding to a particular liner among the plurality of liner options.

[0036] In block **804**, the processor **702** generates a virtual swimming pool based on the one or more visual characteristics selected by the user. The virtual swimming pool is a three-dimensional (3D) rendering of a swimming pool. The processor **702** can generate the virtual swimming pool using any number of combination of techniques. Two example techniques are described below.

[0037] One exemplary technique can involve the processor **702** identifying a particular pool image from among a group of pool images based on the one or more visual characteristics selected by the user. For example, a group of pool images may have been generated beforehand using a 3D rendering tool, such as V-Ray. Each of the pool images can have a unique combination of visual characteristics, such as a unique combination of a background setting, a liner, a viewing angle, a pool shape, or any combination of these. For instance, if there are 3 options for background settings, 4 viewing angle options for each background setting, 20 liner options, and 3 options for pool shape, then there may be $3\times4\times20\times3=720$ total images in the group, to cover every possible combination of these options. The processor **702** can determine which particular pool image has the one or more visual characteristics selected by the user, access a repository in which the group of pool images are stored, and retrieve the particular pool image from the repository. The processor **702** can then generate the virtual swimming pool using the particular pool image.

[0038] Another exemplary technique can involve the processor **702** using a layered approach, in which the processor **702** can generate the virtual swimming pool by overlaying images on top of one another. For example, the processor **702** can identify a background image having the background setting and viewing angle selected by the user, access a repository of background

images, and retrieve the identified background image. The processor **702** can also identify a pool image having the shape and viewing angle selected by the user, access a repository of pool images, and retrieve the identified pool image. The processor **702** can further identify a liner image having the liner and viewing angle selected by the user, access a repository of liner images, and retrieve the identified liner image. The background, pool, and liner repositories can be a single repository or different repositories. The processor **702** can then overlay the pool image overtop of the background image, and overlay the liner image overtop of the pool image. This can yield a virtual scene with a virtual swimming pool, whereby the virtual scene and virtual swimming pool have the one or more visual characteristics selected by the user. This technique may require the generation of significantly fewer images upfront than the first technique discussed above. For example, if there are 3 options for background settings, 4 viewing angle options for each background setting, 20 liner options, and 3 options for pool shape, then there may only need to be (3×4)+(20×4)+(3×4)=104 total images to cover every possible combination of these options.

[0039] In some examples, the second technique discussed above may also result in faster updating than the first technique discussed above. For example, after the virtual swimming pool is initially displayed, the user may decide that he does not like a visual characteristic that he has previously selected. So, the user may provide user input to modify (e.g., update) a visual characteristic associated with the virtual swimming pool, such as the liner. The processor **702** can receive the user input and responsively modify the visual characteristic by changing one of the image layers (e.g., by switching the liner image for another liner image) rather than the entire image, which can be significantly faster than updating an entire image.

[0040] In block **806**, the processor **702** outputs the virtual swimming pool for display on a display device, such as on display device **714**. Outputting the virtual swimming pool for display can involve transmitting one or more electronic communications (e.g., over the Internet) configured for causing the virtual swimming pool to be displayed on the display device. For example, the processor **702** can be part of a server in communication with a client device. The server can transmit one or more electronic communications over the Internet to the client device. The one or more electronic communications can be configured for causing the virtual swimming pool to be displayed on a display device of the client device. As another example, the processor **702** can generate and transmit one or more display signals to the display **714** for causing the display to output the virtual swimming pool.

[0041] FIG. **10** is a block diagram of an example of a system **1000** for implementing a pool visualization application according to some aspects of the present disclosure. The system **1000** includes a server **1002** that may include some or all of the components discussed above with respect to FIG. **7**, such as the processor **702** and the memory **704**. The server **1002** is in communication with a client device **1004** via a network **1020**, such as the Internet. Examples of the client device **1004** can include a desktop computer or a mobile device, such as a laptop computer, a tablet, or mobile phone. The server **1002** can include at least part of a pool visualization application, such as pool visualization application **718**, for implementing some aspects described herein.

[0042] In some examples, the pool visualization application can include client-side code **1006** and server-side code **1008** that can work together in a client-server architecture to implement some or all of the functionality described herein. The client-side code **1006** can be program code configured to be delivered to and run by the client device **1004**. Examples of the program code can include hypertext markup language (HTML) code, Javascript code, JQuery code, Cascading Style Sheet (CSS) code, C++ code, Java code, Objective-C code, Python code, or any combination of these. In the example of FIG. **10**, the client-side code **1006** is transmitted to the client device **1004** as part of a webpage **1014**. The client-side code **1004** is run (e.g., executed or interpreted) by a website browser **1010** of the client device **1004** for rendering the webpage **1014** with an interactive GUI **1018** for visualizing a swimming pool on a property. An example of such a GUI **10818** is the GUI

described above with reference to FIGS. **1**-**6**. But in other examples, the client-side code **1006** may be a standalone software application configured to execute outside the context of a website browser **1010**. Any suitable implementation of client-side code **1006** may be used to generate an interactive GUI **1018** for implementing some or all of the functionality described herein.
[0043] The server-side code **1008** can include program code configured to be run by the server **1002**. Examples of the program code can include PHP code, Python code, C++ code, Java code, or any combination of these. The server-side code **1008** can be configured to interact with the client-side code **1006** to generate the GUI **1018**.
[0044] In some examples, a user **1012** of the client device **1004** can interact with the GUI **1018** by supplying input to the client device **1004**. The client-side code **1006** can receive the user input and responsively transmit requests **1016** (e.g., REST requests) to the server-side code **1008** of the server **1008**. The requests **1016** can be configured for effectuating some or all of the functionality described herein. For example, the client-side code **1006** can receive user selections of a liner, background setting, and shape for the virtual swimming pool and responsively transmit requests **1016** indicating these selections to the server-side code **1008**. The server-side code **1008** can receive such requests **1016** and responsively perform one or more operations based on the requests **1016**. For example, the server-side code **1008** can generate or update the GUI **1018** to depict a virtual scene with a virtual swimming pool in accordance with the user selections.
[0045] Although FIG. **10** depicts a certain number and arrangement of components, other examples can include more components, fewer components, different components, or a different arrangement of the components shown in FIG. **10**. For example, multiple servers or services (e.g., microservices or serverless functions) may work together to collectively implement some or all of the server-side functionality associated with the server-side code **1008** in the client-server architecture described above.
[0046] The foregoing description of certain examples, including illustrated examples, has been presented only for the purpose of illustration and description and is not intended to be exhaustive or to limit the disclosure to the precise forms disclosed. Numerous modifications, adaptations, and uses thereof will be apparent to those skilled in the art without departing from the scope of the disclosure. For instance, examples described herein can be combined together to yield still further examples.

Claims

1. A system comprising: one or more processors; and one or more memories storing instructions that are executable by the one or more processors for causing the one or more processors to perform operations including: providing a webpage to a client device of a user, wherein the webpage includes a graphical user interface through which the user can select a pool shape from a set of pool shapes and a pool liner from a set of pool liners; receiving, via the graphical user interface in the webpage, selections from the user of a particular pool shape in the set of pool shapes and a particular pool liner in the set of pool liners; identifying a pre-rendered pool image, from a stored set of pre-rendered pool images, depicting the particular pool shape and the particular pool liner selected by the user; and outputting the identified pre-rendered pool image in the webpage, wherein the pre-rendered pool image depicts a swimming pool that has the particular pool shape and the particular pool liner selected by the user.
2. The system of claim 1, wherein the webpage includes a JavaScript application configured to generate the graphical user interface on the client device.
3. The system of claim 1, wherein the operation of providing the webpage to the client device comprises: transmitting client-side code to the client device, the client-side code being configured to be executed by a website browser of the client device to generate the webpage.
4. The system of claim 3, further comprising server-side code that is configured to cooperate with

the client-side code to provide real-time interactivity to the graphical user interface.

5. The system of claim 4, wherein the client-side code is configured to transmit real-time requests to the one or more processors executing the server-side code in response to user interactions with the graphical user interface to provide the real-time interactivity.

6. The system of claim 1, wherein the operations further comprise: prior to receiving the selections from the user, storing the set of pre-rendered pool images in a datastore that is remote from the client device.

7. The system of claim 1, wherein the graphical user interface shows a plurality of images of pool liners and a plurality of images of pool shapes, the plurality of images of pool liners corresponding to the set of pool liners, and the plurality of images of pool shapes corresponding to the set of pool shapes.

8. A method, comprising: providing, by a processor, a webpage to a client device of a user, wherein the webpage includes a graphical user interface through which the user can select a pool shape from a set of pool shapes and a pool liner from a set of pool liners; receiving, by the processor and via the graphical user interface in the webpage, selections from the user of a particular pool shape in the set of pool shapes and a particular pool liner in the set of pool liners; identifying, by the processor, a pre-rendered pool image, from a stored set of pre-rendered pool images, depicting the particular pool shape and the particular pool liner selected by the user; and outputting, by the processor, the identified pre-rendered pool image in the webpage, wherein the pre-rendered pool image depicts a swimming pool that has the particular pool shape and the particular pool liner selected by the user.

9. The method of claim 8, wherein the webpage includes a JavaScript application configured to generate the graphical user interface on the client device.

10. The method of claim 8, wherein the operation of providing the webpage to the client device comprises: transmitting client-side code to the client device, the client-side code being configured to be executed by a website browser of the client device to generate the webpage.

11. The method of claim 10, further comprising: executing server-side code that cooperates with the client-side code to provide real-time interactivity to the graphical user interface.

12. The method of claim 11, further comprising: receiving real-time requests from the client-side code, the real-time requests being generated by the client device in response to user interactions with the graphical user interface to provide the real-time interactivity.

13. The method of claim 8, further comprising: prior to receiving the selections from the user, storing the set of pre-rendered pool images in a datastore that is remote from the client device.

14. The method of claim 8, wherein the graphical user interface shows a plurality of images of pool liners and a plurality of images of pool shapes, the plurality of images of pool liners corresponding to the set of pool liners, and the plurality of images of pool shapes corresponding to the set of pool shapes.

15. A non-transitory computer-readable medium comprising program code that is executable by one or more processors for causing the one or more processors to perform operations including: providing a webpage to a client device of a user, wherein the webpage includes a graphical user interface through which the user can select a pool shape from a set of pool shapes and a pool liner from a set of pool liners; receiving, via the graphical user interface in the webpage, selections from the user of a particular pool shape in the set of pool shapes and a particular pool liner in the set of pool liners; identifying a pre-rendered pool image, from a stored set of pre-rendered pool images, depicting the particular pool shape and the particular pool liner selected by the user; and outputting the identified pre-rendered pool image in the webpage, wherein the pre-rendered pool image depicts a swimming pool that has the particular pool shape and the particular pool liner selected by the user.

16. The non-transitory computer-readable medium of claim 15, wherein the webpage includes a JavaScript application configured to generate the graphical user interface on the client device.

17. The non-transitory computer-readable medium of claim 15, wherein the operation of providing the webpage to the client device comprises: transmitting client-side code to the client device, the client-side code being configured to be executed by a website browser of the client device to generate the webpage.

18. The non-transitory computer-readable medium of claim 17, wherein the operations further comprise: executing server-side code configured to cooperate with the client-side code to provide real-time interactivity to the graphical user interface.

19. The non-transitory computer-readable medium of claim 18, wherein the operations further comprise: receiving real-time requests from the client-side code, the real-time requests being generated by the client device in response to user interactions with the graphical user interface to provide the real-time interactivity.

20. The non-transitory computer-readable medium of claim 15, wherein the operations further comprise: prior to receiving the selections from the user, storing the set of pre-rendered pool images in a datastore that is remote from the client device.