Creating a Dynamic Component

There are infinite things you can create dynamic components (DCs) to do, and it would take a separate book just to scratch the surface. But for Pro users interested in learning about what comprises a DC, work through this relatively simple exercise. We’ll create a DC in which you can change object materials, and animate one of the objects.

Once you learn some basics about attributes, behaviors, and functions, you can make changes to existing DC’s, or maybe venture out and create some of your own.

Preparation the Model

We’ll start with an existing set of objects, and make a DC from them.

1. In the Components window, search for “bouncer roskes.” For the model shown below, click its thumbnail to insert the model into your file.

   ![Components window](image)

   NOTE: The model “Dynamic Bouncing Ball” is the completed DC you’re about to create.

2. Do not explode the model. The DC will be the main, “first-level” component, which is always comprised of nested components.

3. Open the Outliner (Window / Outliner). The DC will be built from the “Bouncer” component, which consists of one ball, one stand, and two identical wall components.

   ![Outliner window](image)

4. This DC will offer users a choice of materials, and these materials are already in the model. Open the In Model collection of the Materials window to see them.

   ![Materials window](image)

5. Wondering where these materials are used? Turn over the stand component - they are hidden underneath.

   ![Model diagram](image)

   If you plan on offering users a choice of standard colors, you don’t have to embed these colors in your model, since colors can be specified by name or
number. But materials don’t have standard names, so they must be included in your model, with the names you want to use.

6. The materials in this model have their original names, such as “Metal_Corrogated_Shiny.” These names will be used when defining component attributes, so they should be simplified. Change the metal material name to “metal.”

7. Do the same for the other three materials, using the names “BlueMarble,” “PinkStone,” and “Wood.” (These names are not case sensitive.)

8. Now we’re ready to create some attributes. But first, select the component and open its Component Options. It’s not dynamic yet, and has no options or attributes.

Creating Material Attributes

Attributes are basically properties, either assigned to the DC as a whole, or to its sub-components. We need to create material attributes for three parts: ball, stand, and wall, as well as an attributes that define the “x” position of the ball.

Keep the component selected, and open the Component Attributes window (available on the component’s popup menu, or the icon, or the Window menu).

1. The overall DC needs three custom attributes defined - those for the three separate materials. These attributes will be referred to when creating functions for the sub-components. To add the first attribute, click “Add attribute” under “Bouncer.”

2. A list of standard attributes appears, but we’re defining a custom attribute that does not appear on the list. So ignore the list, and type “WallMat,” which appears in the attribute field. Then press Enter. This is the attribute for wall material.

3. The space to the right of “WallMat” is where you enter the attribute definition. This defines the initial material of the walls, which will be changeable by the user. Type the wood material, using the equal-sign and quote format shown below.

4. Press Enter, and you should see “wood” listed in the field.
(If you get an error message, check to make sure the format matches exactly the one above. Also, make sure that the material name is exactly “Wood,” with no spaces.)

5. Click “Add attribute” again, and enter the next attribute: “StandMat” (the material for the stand). Its definition should point to the metal material.

6. Add a third attribute, “BallMat” for the ball material. This one should point to the “pinkstone” material.

7. Now we can set the material behavior for each sub-component, using the attributes we just defined. We’ll start with a wall: click “Add attribute” under either “Wall.”

8. This time we’ll choose a standard attribute from the list: choose “Material,” which is listed under “Behaviors.”

9. “Material” now appears under the new “Behaviors” heading. The definition for this material contains a reference to the wall material attribute we defined under “Bouncer.” Enter the definition shown below, with an equal sign, but without quotation marks.

10. Press Enter, and “wood” should appear in the field, since that is WallMat’s defined material.

Wood should now appear on the walls themselves. (Because the “Wood” components are identical, the material attribute you defined for one wall is applied to both.)
11. Do the same for the “Ball” component: add a “Material” attribute, and point it to the “BallMat” attribute.

12. When you press Enter, this field should contain “pinkstone,” and the ball itself should be pink.

13. Do the same for the “Stand” component, which should point to “metal.”

Creating Ball Position and Animation

In addition to changing materials, the other feature of this DC will be animating the ball to move between the walls and stand. This involves changing of the ball’s position in the X (red) direction. So just as we defined each object’s initial material, we need to define the ball’s initial X position.

1. First, make sure you are working in inches. If you are working in a metric file and see “cm” on the ruler icon next to “Ball,” click the icon to toggle to “inch.” (This is because the values listed in later steps are in inches.)

2. Add another attribute under “Ball.”

3. We’ll use another standard attribute: choose X, which is located under “Position.”

4. “X” now appears under the new “Position” heading. The initial value of 34” is listed in the definition field, but grayed out.
34” is the position of the center of the ball, measured from the start of the component (i.e. measured from the back of one wall). SketchUp assumes you want to use this value, but it’s not “official” until you confirm it (or change it).

5. To set 34” as the initial position of the ball, click in the definition field and add an equal sign before the 34.

6. Press Enter, and the 34” is no longer grayed out.

7. We now need to add another behavior for the ball, one that tells the ball what to do when it is clicked. Add another attribute for the ball, and choose “onClick.”

8. “onClick” is added to the ball’s “Behaviors” list. Its definition needs a specific syntax which is not so easy to enter manually. So we’ll copy it from a predefined list. Click inside the “onClick” definition field.

9. Then click the Functions tab at the top of the Component Attributes window.

10. To see the list of functions, click the drop-down arrow below the Functions tab. The syntax we want is for “ANIMATE,” listed under “onClick Functions.”
11. Once the ANIMATE syntax is listed at the top of the Functions tab, click Insert.

The syntax now appears next to “onClick,” under “Ball.”

12. Edit this syntax as shown below.

The “X” defines the attribute that will be animated (the ball’s X position), and the numerical values define the ball’s position each time the ball is clicked:

• The ball starts at 34” (atop the stand)
• It then moves back 32” to stop at the 2” position (the front of one wall).
• It then returns to the 34” position (back to its stand).
• It then moves forward 32” to stop at the 66” position (the front of the other wall).
• With continuous clicks, the ball starts back at 34” and scrolls through the values again.

13. To test your animation, activate Interact and click the ball.

It should move back and forth between the walls and the stand.

Creating Material Behavior

When the user uses Interact on the ball it moves, and when the user click on the wall or stand, we want its material to change. So we need a different type of “onClick” behavior for these objects.

1. Start with the wall: add an “onClick” attribute. Then place your cursor inside its definition field.
2. For the definition syntax, find the SET function, listed under “onClick.”

3. Insert this syntax into the definition field.

4. The attribute is the wall material, entered below as “Bouncer!WallMat.” Then each material name comes next, each within quotes and separate by commas. Type this carefully!

5. After pressing Enter, test out the walls by using Interact to click on them. Both walls should scroll through all four materials.

6. We want the same behavior applied to the stand. But we don’t need to type this syntax out again; we can copy the one we already entered for the wall. Select the wall’s material behavior syntax and copy it (Ctrl+C, Cmd+C).

7. Add an “onClick” attribute to the stand, and paste in the syntax (Ctrl+V, Cmd+V).

8. Make the one necessary change to the material attribute:
9. Now you should be able to interact with the stand to change its materials.

Adding User Options
So far we can use Interact on each object for different purposes. But there are no options the user can define (of course, not all DC’s have options). So we’ll add options for the user to choose materials for the ball, and to specify colors for the walls and stand (in case they don’t like the provided materials).

1. We’ll start with the ball - its “onClick” behavior moves the ball, and we need a way for the user to set its material. This option will set the “BallMat” attribute, so locate this attribute under “Bouncer.” Place your cursor in the definition field, and click the Details arrow.

2. By default, all attributes are hidden from the user, so you need to define the ones you want the user to see. Choose User can select from a list, and for its Display Label, enter the instructions you want the user to see for choosing the ball material.

3. Since the user needs to see a list of available materials, click Add option, and enter the material names and values. The definition under Values must match the material names exactly.

4. Click Apply when finished. Then open the Component Options, and you should see your instructions and drop-down list.
5. For the wall and stand, we can be more generous, and allow the user to choose his own color. Click the Details arrow for the “StandMat” attribute.

6. This time, choose Users can edit as a textbox, and write a label that lets users know they can enter a color.

7. Click Apply, then add the same function for “WallMat.”

Now the Component Options shows three fields: one for choosing the ball material, and two for entering wall and stand colors. Colors can be entered by name, as shown below, or by RGB values (such as 230,55,140), or by hexadecimal values (such as FF8040).

8. If you want to go back to the preset materials for the walls and stand, just use Interact to scroll through the list.
Finishing the Dynamic Component

A well-made DC has a detailed description that informs the user about everything the DC can do.

1. First, the DC could use a name better than “Bouncer.” Add a “Name” attribute under “Bouncer.”

2. Assign this much better name: “Bouncing Ball.”

3. Add a “Description” attribute that explains the available options, and what happens when using Interact.

Tip: The description is HTML-based, so you can add tags and even hyperlinks if you like.

Now the Component Options window is much more informative.
Sharing a Dynamic Component

Interesting in uploading DC’s to the 3D Warehouse?

In your current file, your DC is already a component. So if you upload the entire file in its current state, and another user inserts your model directly into his file, it will have to be exploded. The DC will then become a first-level component, and it will function just fine. However, most people don’t want to have to explode a DC in order to use it.

Sharing a the dynamic component can also be done like sharing a regular component. Just right-click on it and choose Share Component.