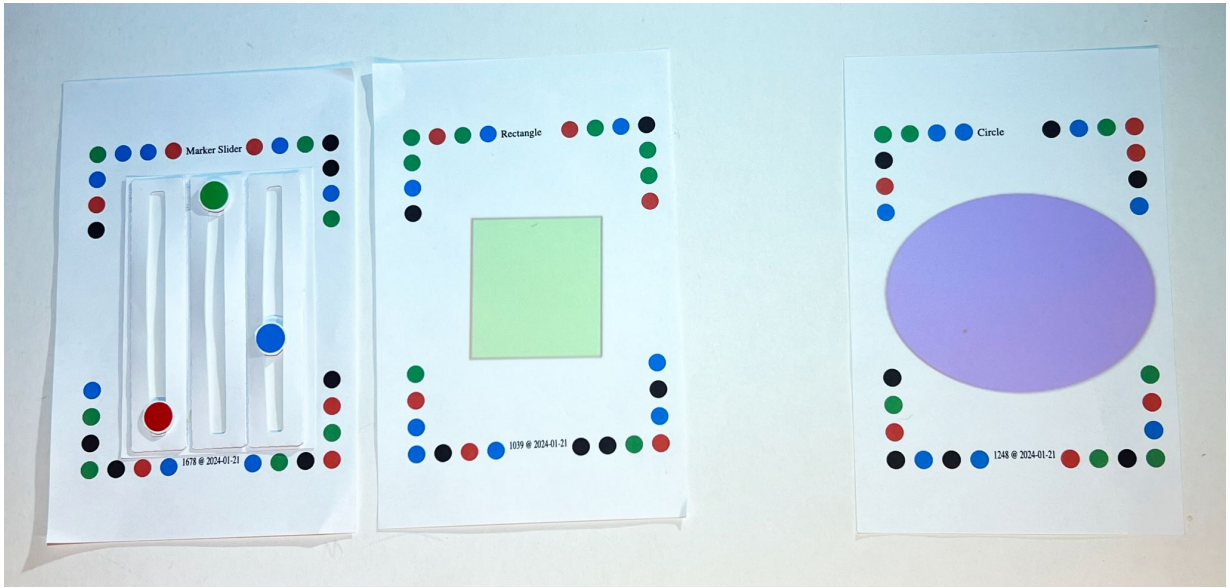
Paper Playground

Whiskers, sliders, and shapes

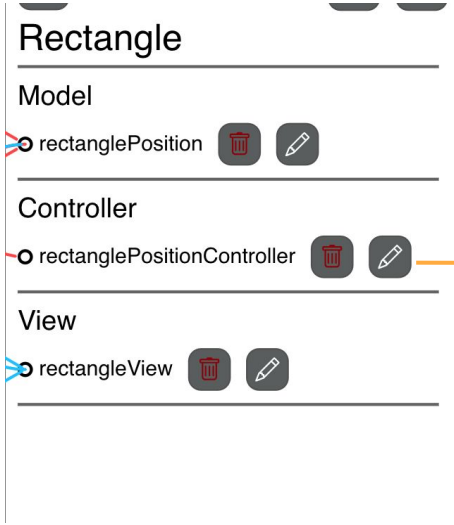
Conceptual overview

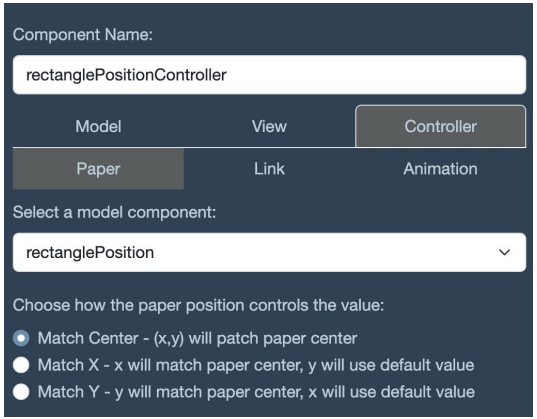
Supported by NSF Award #2119303
Inclusively-designing sensory extensions for STEM inquiry learning

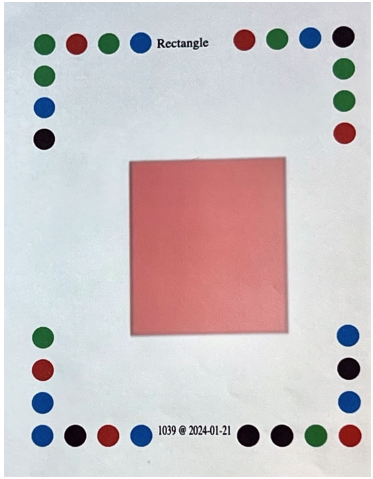
In this conceptual overview, we will use Paper Playground to create shapes and color controls using marker and whisker features. We will show the interplay between the software and physical paper aspects of the system.



We start with a standard Movable Rectangle from the templates list. By default, the Rectangle is created with a rectanglePosition, a rectanglePositionController, and a rectangleView.



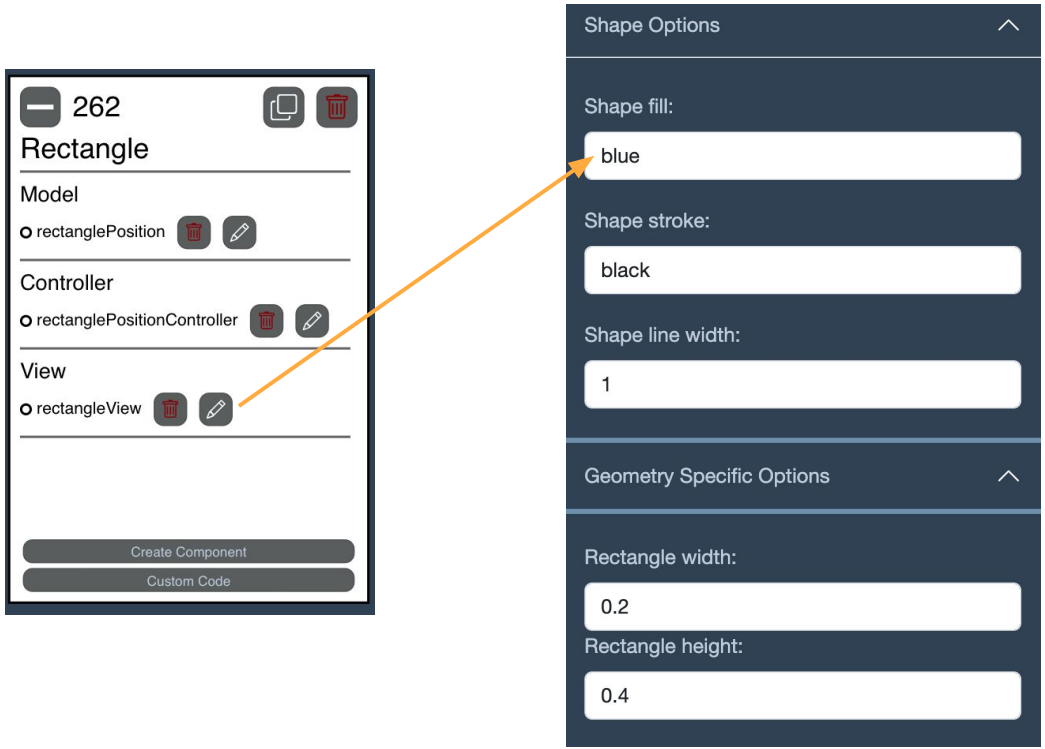




The projected rectangle moves with the paper.

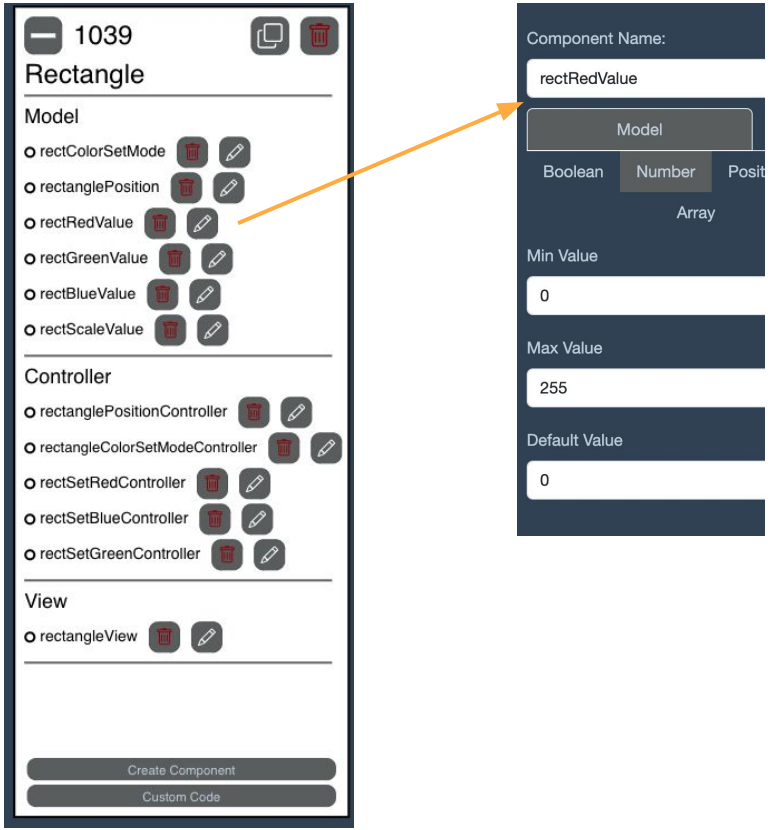
Clicking on the pencil icon of rectanglePositionController shows us that it controls the model component rectanglePosition using the center of the paper. In other words, the coordinates of the center of the paper are the x and y values of the rectanglePosition.

Clicking on the pencil icon of rectangleView shows many attributes of the shape, including its fill color.



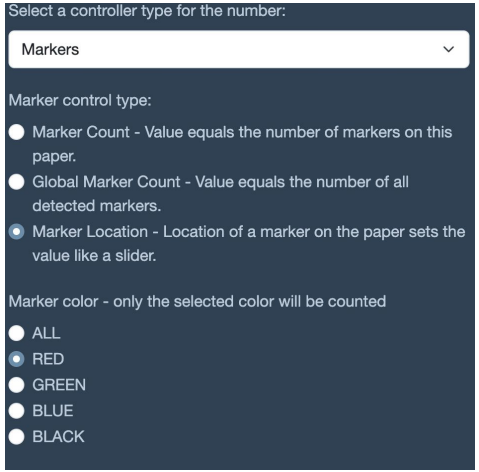
We will add some model components to Rectangle so that it is possible to pass it values for red, green, and blue to set its filll color.

Each of rectRedValue, rectBlueValue, rectGreenValue is a Model->Number with a value between 0 and 255, representing a value on the RGB scale.

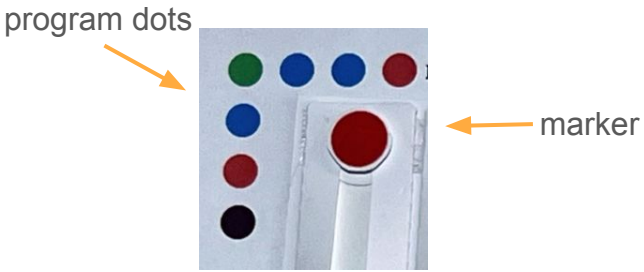


Markers

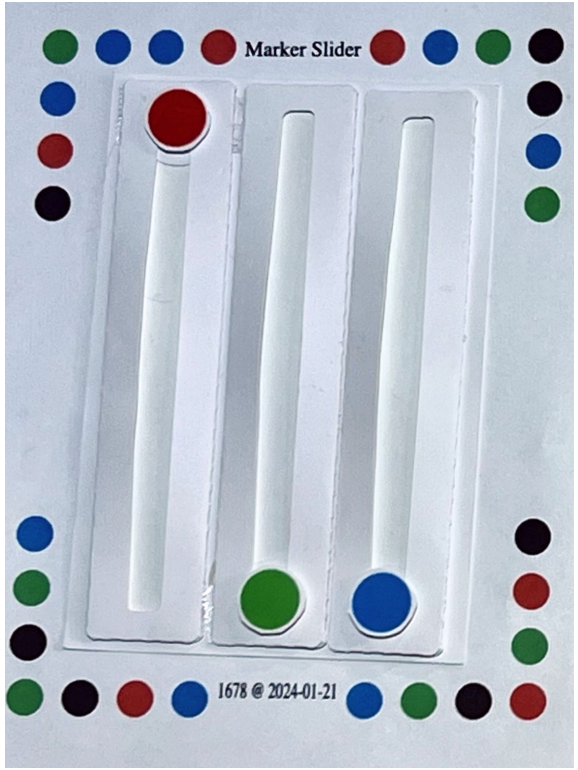
We will now build our own paper tool to control the color of the rectangle. We will use built-in controllers called Markers, which are located in the Controller->Paper interface. In the physical paper world, a marker is a filled circle in red, green, blue, or black. It must be larger than the dots used to define a paper program so that the system can distinguish the circle as a marker.



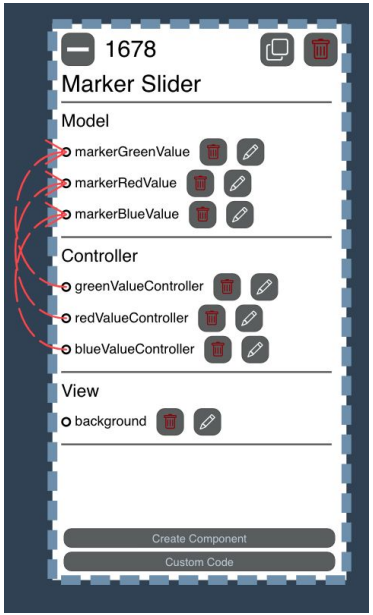
We can count markers (either locally on a single paper or globally across the entire playground) or we can use their locations as ways to control programs.

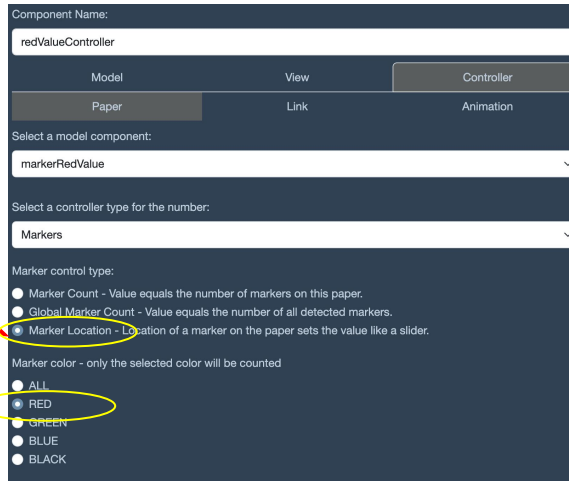


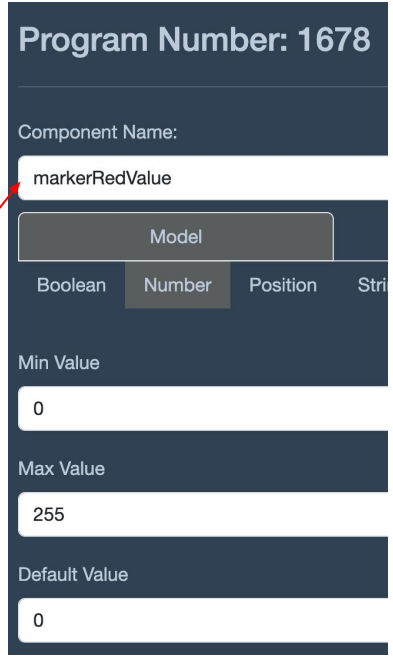
We will use marker locations to build the paper tool. This is a controller that we will call a Marker Slider. It consists of red, green, and blue markers sliding along tracks. In order to have it communicate with the rest of the objects, we need to place the tool on a program that reads the positions of its markers and sends it to the other programs.



How do we configure Marker Slider in creator?

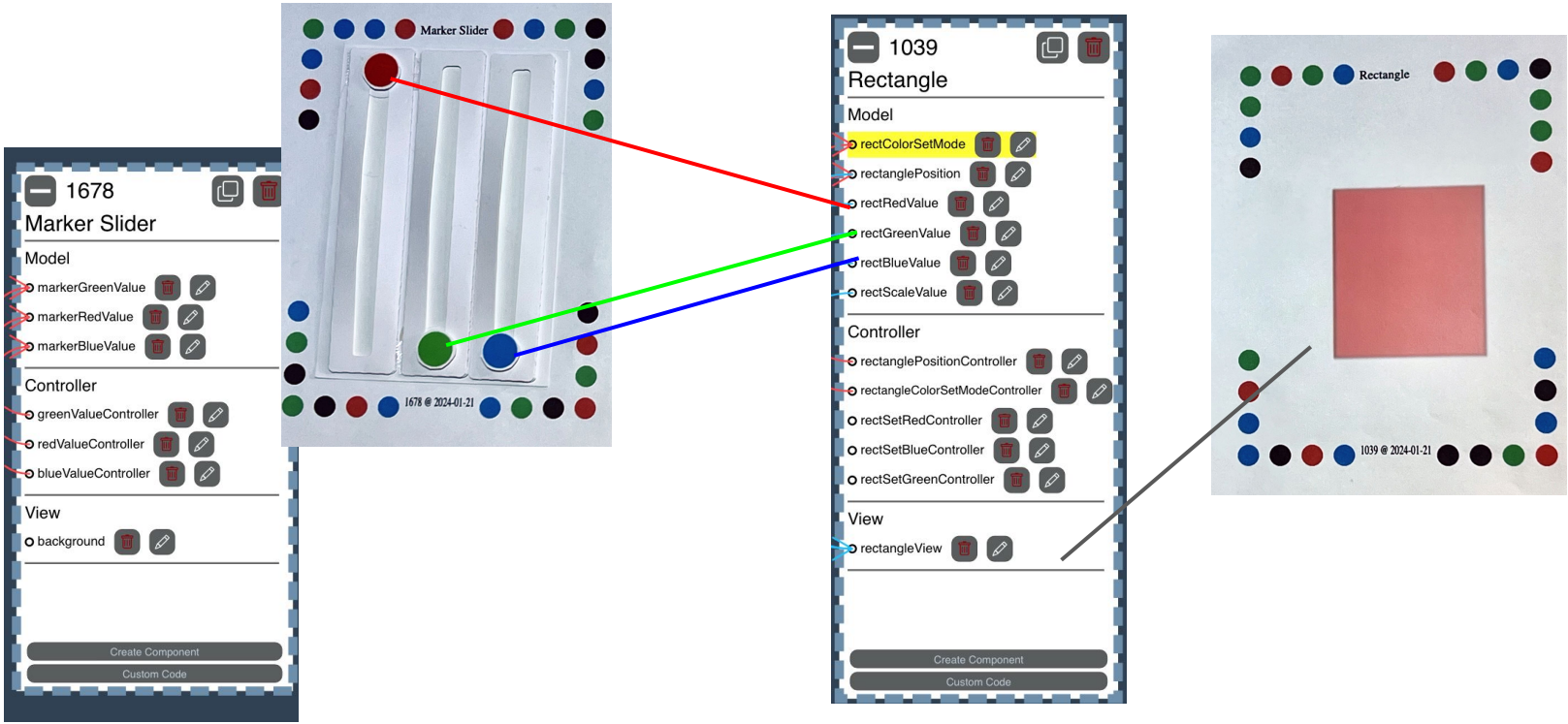




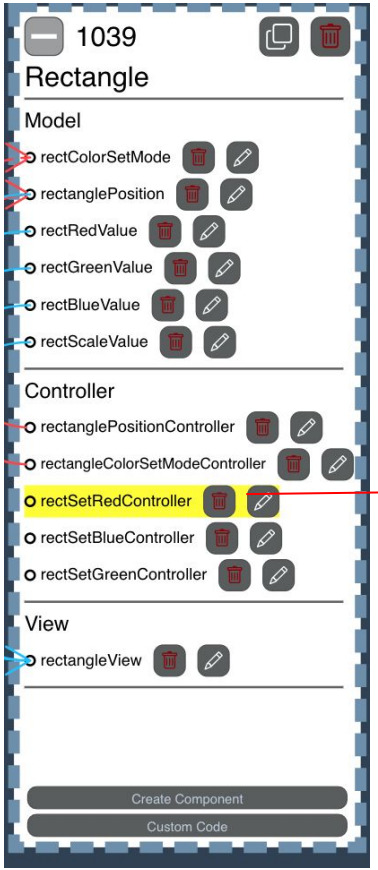


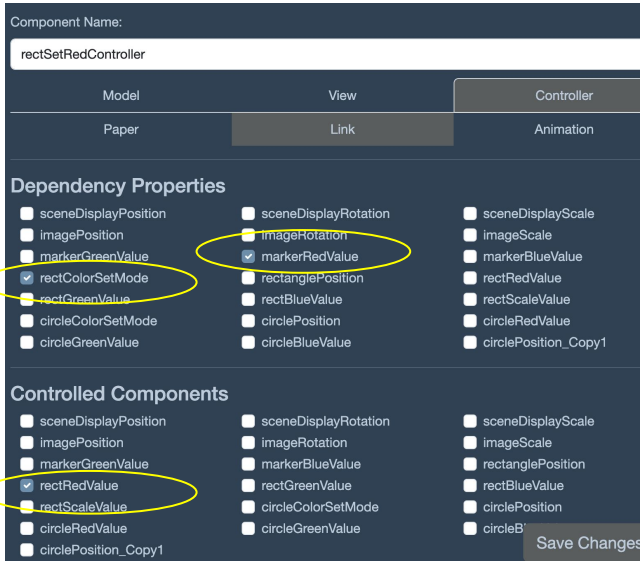
Marker Slider is a program that tracks the value of three markers: one red, one green, and one blue. Each color value is parameterized within the range specified in the model, specified by the y-position of the physical marker on the piece of paper. For example, a red marker at the bottom of the paper sets the markerRedValue to 0, while a red marker at the top of the paper sets the markerRedValue to 255.

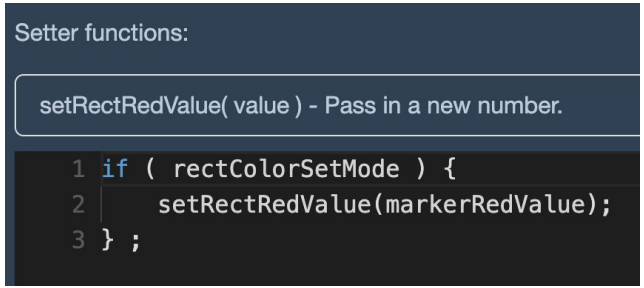
Conceptually, each marker represents a value from 0-255, depending on the position of the slider. In the photo above, the red slider is at 255, and the green and blue are both 0. The rectangle is then filled with the RGB color (255, 0, 0).



Here's a more detailed look:



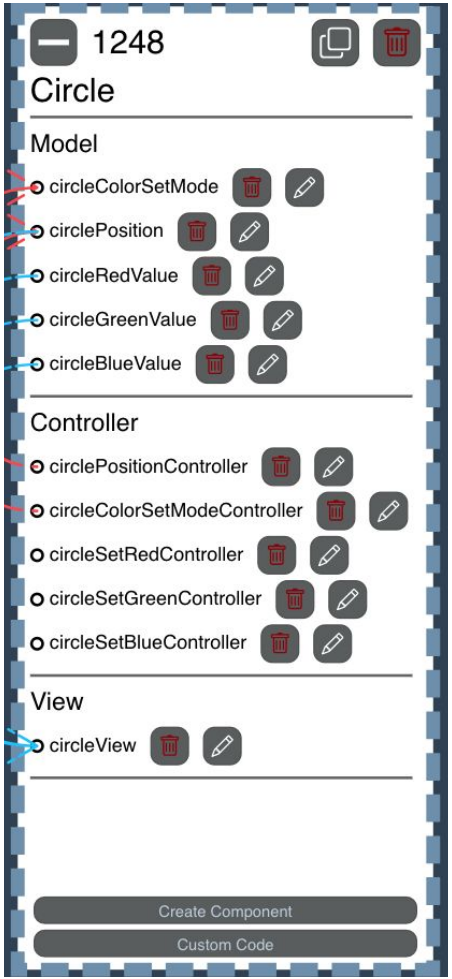


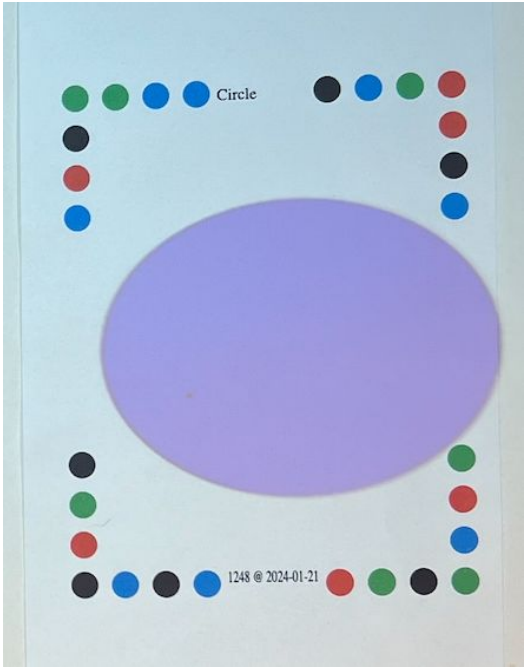


rectSetRedController is dependent upon model properties rectColorSetMode and markerRedValue, and it controls rectRedValue. [rectSetBlueController and rectSetGreenController behave in similar ways]

Then in the custom code box, we enter the following Javascript. It means, "If rectColorSetMode is true, then pass the markerRedValue (a number from the red marker in the color slider) to the controller that sets the rectRedValue".

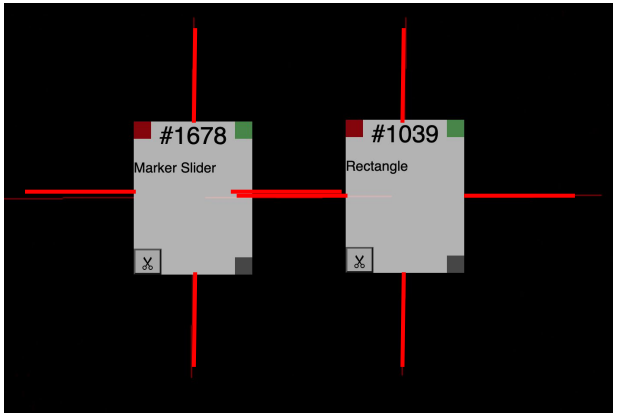
We can configure other shapes in much the same way. We've added the same types of attributes and controllers to a Circle.



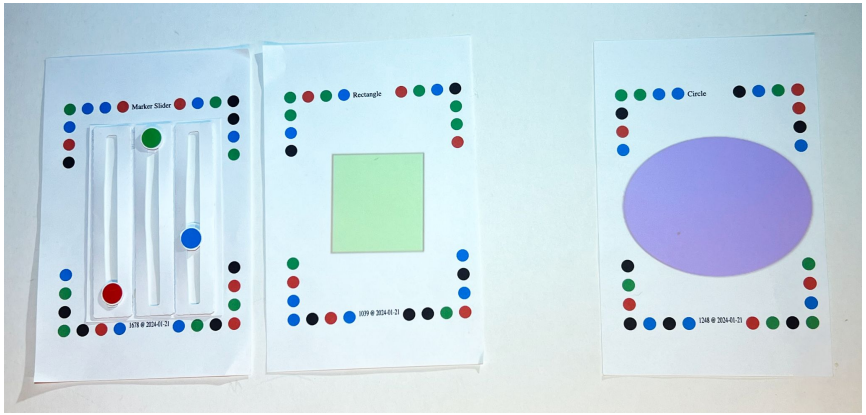


Whiskers

Now suppose we want our Marker Slider to be a *general* color-changing tool that can work on different shapes in the same play space. We need a way to tell a Marker Slider which shape we want to color. We will use a feature called “whiskers”.

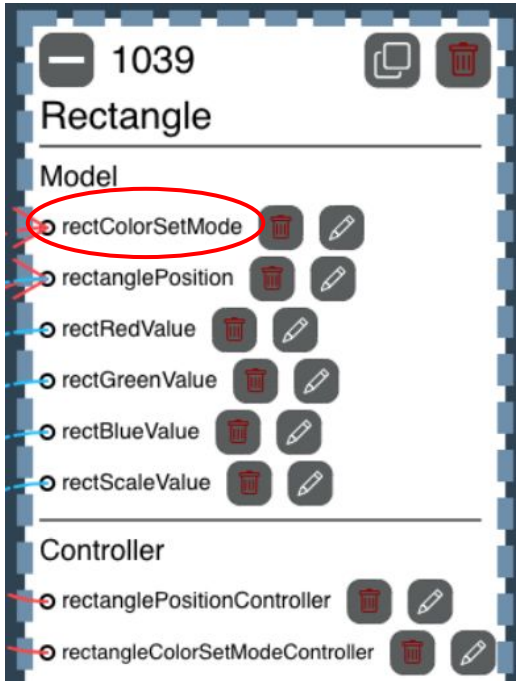


When whiskers are used as controllers, program behaviors trigger only when other specific programs are within proximity. This is a representation of the two pieces of paper showing whiskers.



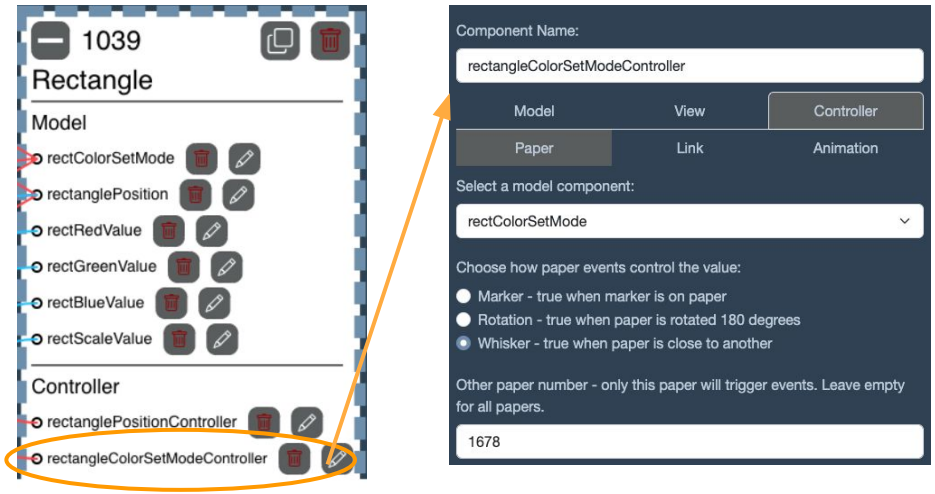
These are programs #1678 (Marker Slider) and #1039 (Rectangle) as physical pieces of paper. Marker Slider will change the color of Rectangle because they are within whisker distance of one another. But Marker Slider will not affect the color of Circle because it is not within whisker proximity of it.

How do we know when we want a shape to have its color set by Marker Slider?

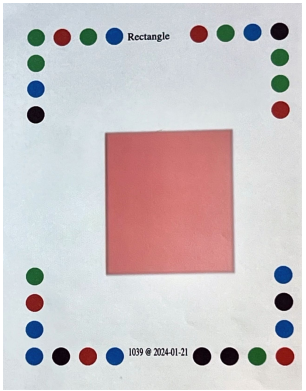


We will add a model component and call it rectColorSetMode, which is a Model->Boolean. By default it is false. We can only change model colors when rectColorSetMode is true.

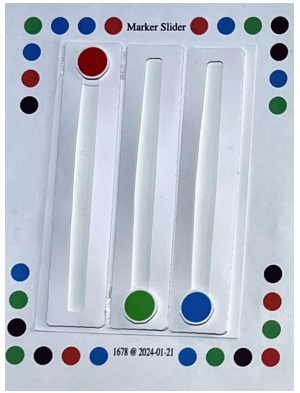
A more detailed look. We will add a controller and call it rectangleColorSetModeController. It controls the model component rectColorSetMode with a Whisker control. We enter #1678 as the only paper that will activate the controller. In other words, when paper #1678 (Marker Slider) is close enough to the rectangle (#1039) program, it sets rectColorSetMode to “true”. This means that we can now set the color of the rectangle. When we move the papers apart (out of whisker range), the rectColorSetMode automatically reverts to “false”. The rectangle will retain the most recently set color even when we move the Marker Slider out of whisker range.



This is what it looks like in Creator. Clicking on the pencil icon will show the component details.



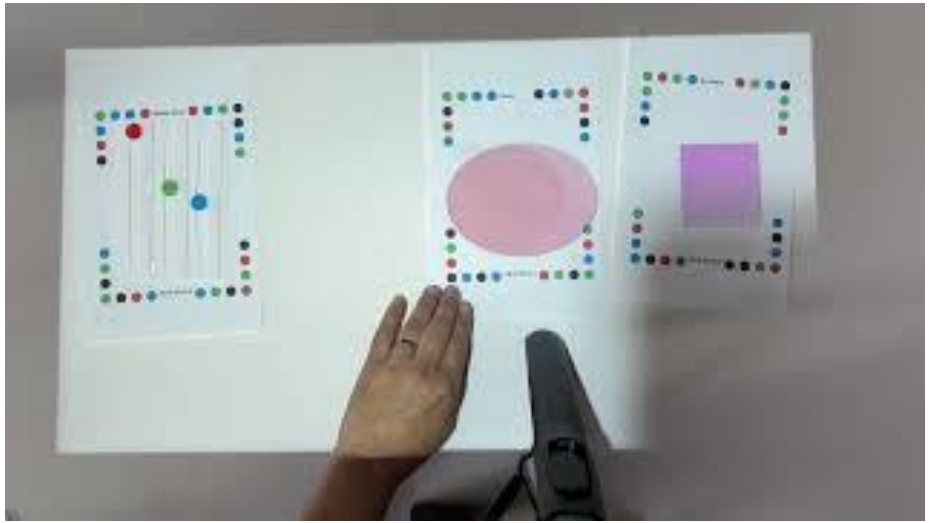
Program #1039



Program #1678

These are the physical pieces of paper. When #1678 is close enough to #1039, rectColorSetModeController sets rectColorSetMode to “true”. When we move them apart, rectColorSetMode automatically reverts to “false”.

Here is a demo of the finished scenario:



View here on YouTube.

Visit Paper Playground for downloads, documentation, and tutorials!