

Chapter 4. Simulating an Impressionist Painting

Glen Bull, Jo Watts, and Alexis Kellam

The image on the left is a photograph of a window box with flowers. The image on the right is an impressionist painting of the flowerbox created through a technique known as pointillism. This method creates a painting by placing tiny dots on the canvas to create an impression of the original painting.

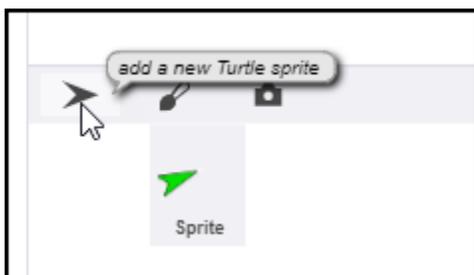


In the previous module, dots were placed across the stage to create a polka dot background. The same method can be used to create an impressionist painting.

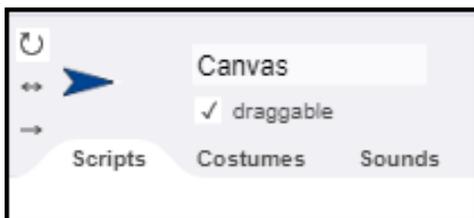
Topic 4.1 Using a Sprite as a Canvas

A photograph is imported into Snap!, where it becomes the costume of a sprite that serves as the canvas for the painting. This is the first module in which more than one sprite is used. One sprite will be used as the canvas for the painting. A second sprite will be used as a paint brush to create the painting on the canvas.

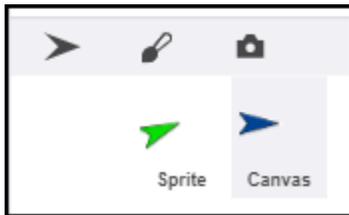
Begin by adding a new turtle sprite that will become the canvas.



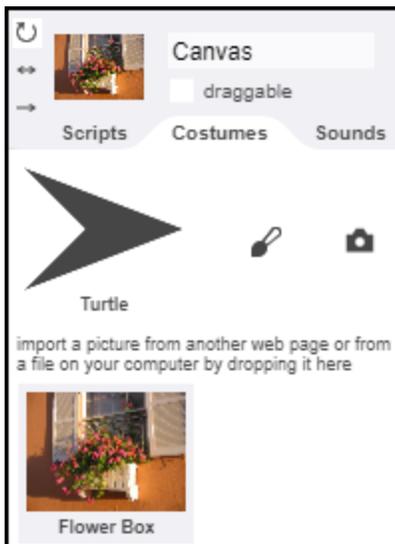
Once the second sprite is created, change its name to *Canvas*.



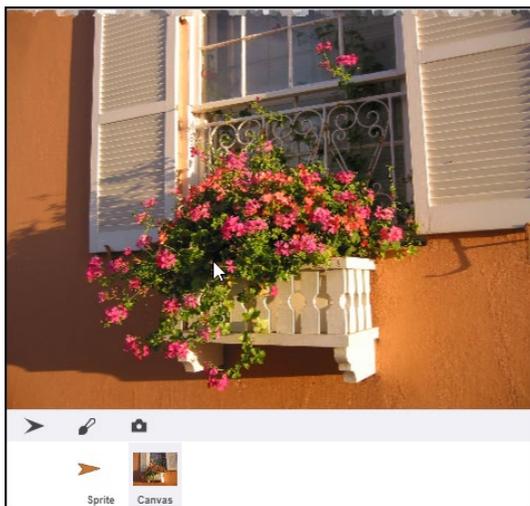
Once this is accomplished, two sprites will exist – the original sprite named “Sprite” and a second sprite named “Canvas.”



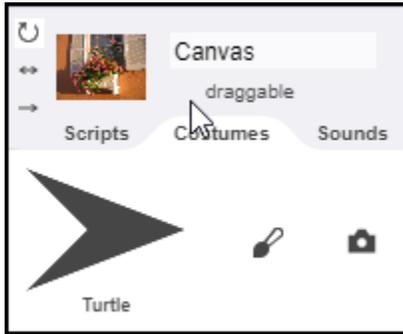
Select the *Costumes* tab of the *Canvas* sprite and import a photograph by dragging it into the space below the *Costumes* tab.



The photograph will become the new costume worn by the *Canvas* sprite. Drag the *Canvas* sprite to the center of the stage.



Once the *Canvas* sprite is centered on the stage, uncheck the “draggable” option so that the *Canvas* sprite cannot be moved from this position by accident.

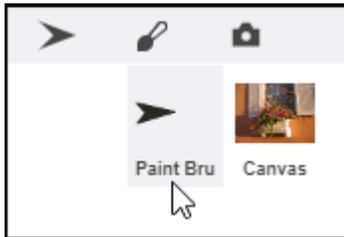


Exploration 4.1 Using a Sprite as a Canvas

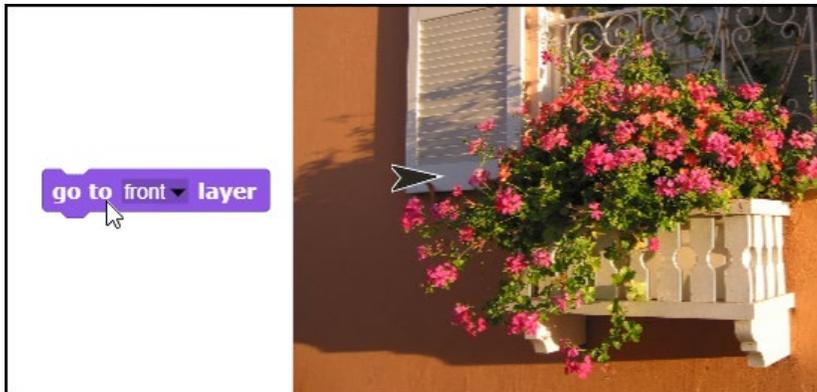
Select an existing photograph (or take a new photograph) and import it to use as the costume for the Canvas sprite. This photograph will be used as the basis for an impressionist painting.

Topic 4.2 Using a Sprite as a Paint Brush

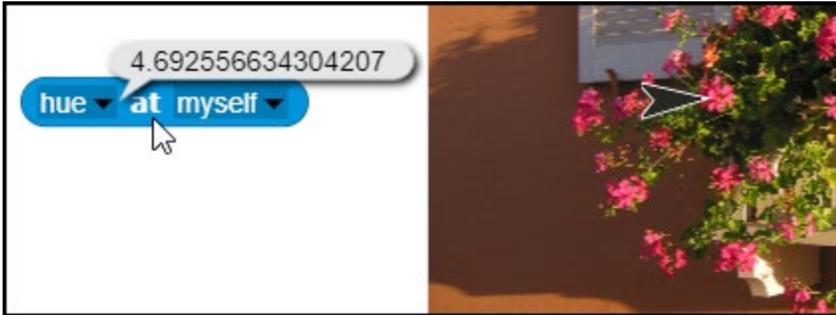
The original sprite (by default named “Sprite”) will be used as a paint brush. Rename the original sprite, using “Paint Brush” as its new name. When this step is completed, two sprites will be visible in the sprite corral beneath the stage: one named “Paint Brush” and a second named “Canvas.”



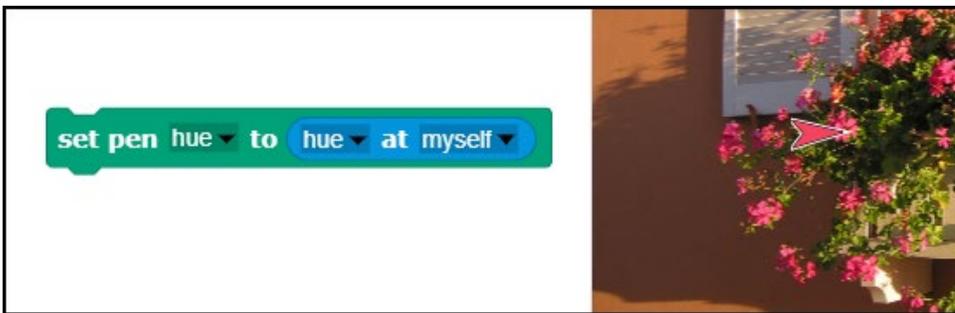
The *Canvas* sprite may cover up the *Paint Brush* sprite on the stage. If that is the case, the *Paint Brush* sprite will not be visible. If the *Paint Brush* sprite is not visible, use the **Go to Front Layer** code block to bring the *Paint Brush* sprite to the front layer so that it is visible. (Note: The *Paint Brush* sprite must be selected in the *Sprite Corral* when this command is executed.)



When the *Paint Brush* sprite is in the front layer, the **Hue at Myself** reporter block can be used to identify the color of the canvas beneath the *Paint Brush* sprite.



The **Hue at Myself** reporter block can be combined with the **Set Pen Hue** block to change the color of the *Paint Brush* sprite to the color beneath the *Paint Brush* sprite.



This method can be extended to include *saturation* and *brightness* as well as the hue.



This will ensure that the *Paint Brush* sprite reflects the exact shade and color of the canvas beneath it. Once this code has been verified and tested, it can be encapsulated in the form of a custom code block, **Reflect Color**.



The **Reflect Color** custom code block causes the *Paint Brush* sprite to reflect the color of the canvas beneath it.

Essential Knowledge

CS Principle 3.13 Developing Procedures

Assigning a name to a group of instructions to create a procedure is one form of abstraction. Once a code sequence is tested and encapsulated in the form of a procedure, the programmer no longer needs to remember the specific details of the procedure, enabling the programmer to focus on other program details.

Application of Essential Knowledge

In the example above, a custom code block named **Reflect Color** replaced a longer sequence of instructions. Replacing an entire sequence of instructions with a named procedure yields several benefits. First, it makes the top level of the program more concise and easier to read. Second, if meaningful names are used, it makes the program easier to understand.

Exploration 4.2 Using a Sprite as a Paint Brush

Create a **Reflect Color** custom block that causes the *Paint Brush* sprite to reflect the color of the underlying *Canvas* sprite. Drag the *Paint Brush* sprite around the *Canvas* and observe the changes in its color when the **Reflect Color** code block is executed.

Topic 4.3 Painting the Canvas

After the *Paint Brush* and *Canvas* sprites have been created, the **Reflect Color** block can be combined with the **Random Position** custom block to move the paint brush to a random position on the stage, reflecting the color beneath the paint brush at the new position. The *Paint Brush* sprite is then used to place a dot at that position.

The **Go to Front Layer** block is involved prior to initiation of the **Repeat** block to ensure that the *Paint Brush* sprite is in the top layer (in front of the *Canvas* sprite) when the process is begun.



Once the code is created and tested, it can be used to create a custom **Paint Canvas** block. The **Paint Canvas** block can be repeated as many times as desired to create the desired impression of the underlying photograph.

Application of Essential Knowledge

In this section, the main **Paint Canvas** procedure calls the **Go to Random Position**, **Reflect Color** and **Dot** code blocks to paint the canvas.

Exploration 4.3 Painting the Canvas

Create a **Paint Canvas** custom block that causes the *Paint Brush* sprite to create a pointillist painting. Explore the effect of invoking the **Paint Canvas** block with different dot sizes.

Topic 4.4 Further Explorations in Simulation of Impressionist Paintings

The **Paint Canvas** block created with the **Dot** block is just a starting point for exploration of impressionist effects. Just as the **Dot** block was replaced with the **Line** block in the previous module to create a confetti background, the **Dot** block can be replaced with the **Line** block in **Paint Canvas** to create a different effect.

Other possibilities include substitution of a starburst in place of a line or a dot. These substitutions can be used to create effects such as the impression of looking at the photograph through frosted glass. Explore the range of effects that can be achieved through different combinations and extensions.