

# Looks and sound- session 3

[https://scratch.mit.edu/projects/editor/?tip\\_bar=home](https://scratch.mit.edu/projects/editor/?tip_bar=home)

In this module, you'll:

- Create animations and image effects
- Learn how layers work in Scratch
- Play sound files and compose music
- Make complete animated scenes of your own

## The Looks Palette

OPEN PROJECT **Animation.sb2** FROM MODULE 3 <http://tinyurl.com/scratch-modules>  
Use File -> Upload from your computer to open it



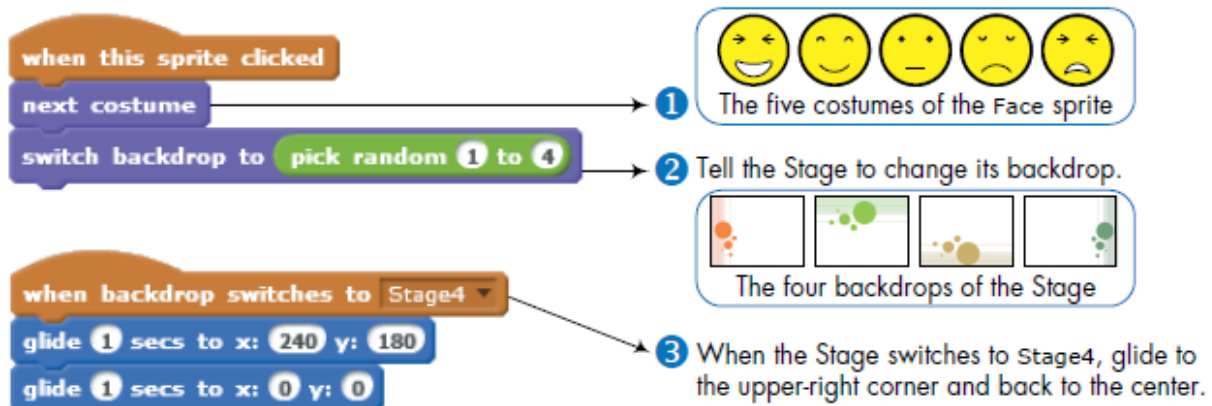
Experiment with different move and wait values

### next costume

You can draw images directly on the Stage using the pen commands, but costumes provide another powerful, and sometimes much easier, way to add graphics to your programs. This application contains one sprite with seven costumes along with one script. You can see the seven costumes in the Costumes tab and the script in the Scripts tab of the sprite. When you run the application by clicking the green flag, the stick figure will appear to walk on the Stage. The key to its motion is the **next costume** command, which tells the sprite to put on the next costume in its list. If the sprite is wearing the last costume in the list, it will roll over to its first costume.

### switch backdrop to

OPEN PROJECT **ClickOnFace.sb2**

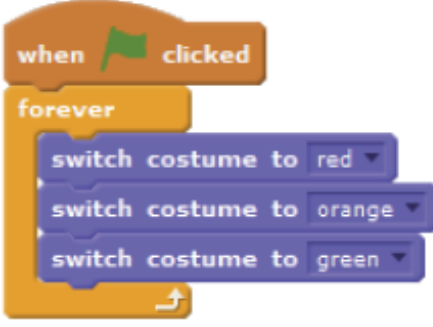



Experiment with different BACKDROPS and add your own sprite

If you want people to interact with a sprite, you could change its costume in response to a mouse click, as in the Click on Face application. This application contains a single sprite, named Face, which has the five costumes. It uses the **when this sprite clicked** block (from the Events palette) to tell the sprite when to switch costumes. The script also uses the **switch backdrop to** block to command the Stage to switch randomly to one of its four backdrops.

## TRY IT OUT

The file TrafficLight.sb2 contains one sprite that has three costumes (named red, orange, and green) and an incomplete script, as shown below. Complete the application by adding the necessary wait blocks to create a realistic traffic light animation.





red
orange
green

OPEN PROJECT [TrafficLight.sb2](#)

say / think





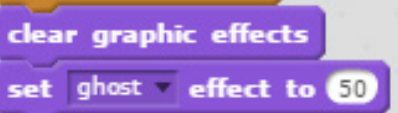



## TRY IT OUT

To see the Say and Think commands in action, open the file Argue.sb2 and run it. This application simulates an endless argument between two characters, as illustrated below. Study the scripts to understand how they use accurate timing to synchronize the actions of the two characters. Click on the backdrop and check the two other block commands from the look palette.









Replace the sprites with your own characters and make up your own story.

OPEN PROJECT [Argue.sb2](#)







 <p>1</p> <p>clear graphic effects</p>	 <p>2</p> <p>set color effect to 20</p>	 <p>3</p> <p>set fisheye effect to 50</p>
 <p>4</p> <p>set whirl effect to 200</p>	 <p>5</p> <p>set pixelate effect to 30</p>	 <p>6</p> <p>set mosaic effect to 10</p>
 <p>7</p> <p>set ghost effect to 70</p>	 <p>8</p> <p>set brightness effect to 80</p>	<p>clear graphic effects</p> <p>set brightness effect to 80</p> <ul style="list-style-type: none"> <li>color</li> <li>fisheye</li> <li>whirl</li> <li>pixelate</li> <li>mosaic</li> <li>brightness</li> <li>ghost</li> </ul>

Click the down arrow in the set effect to block to choose the effect you want from the drop-down menu. You can also use the change effect by command to adjust an effect instead of setting it directly

## size/visibility

```

when green flag clicked
  repeat (5)
    say aaaaa for 0.1 secs
    change size by 10
  say choooo
  repeat (5)
    change size by -10
    wait 0.1 secs
  say 
  
```

Say "aaaaa" for a short time and increase the sprite's size by 10. At the end of the loop, the sprite's size will have increased by 50%.

Reduce the sprite's size gradually to its original size.

Clear the speech bubble, so the sprite isn't saying anything.

change size by 10

show

hide

If you need to shrink or grow a sprite, the **set size to %** and **change size by** commands can help. The first sets a sprite's size to a percentage of its original size, and the second modifies a sprite's size by a specified amount relative to its current size. When you need to make a sprite appear or disappear, use the **show** block or the **hide** block, respectively.

## TRY IT OUT

Add a block to the end of the script in to have the Cat sprite finish its dramatic sneeze by vanishing afterward. Add another block to show the sprite at the beginning of the script.

## layers

OPEN PROJECT [Layers.sb2](#)

go to front  
go back 1 layers

### TRY IT OUT

The last two commands in the Looks palette affect the order in which sprites are drawn on the Stage. The Layers.sb2 application has four objects that move on the Stage. You can bring an object to the top by pressing the first letter of its color. Run the application to explore the effect of the go to front command.

## The Sound Palette

### playing audio files

play sound meow  
play sound meow until done  
stop all sounds

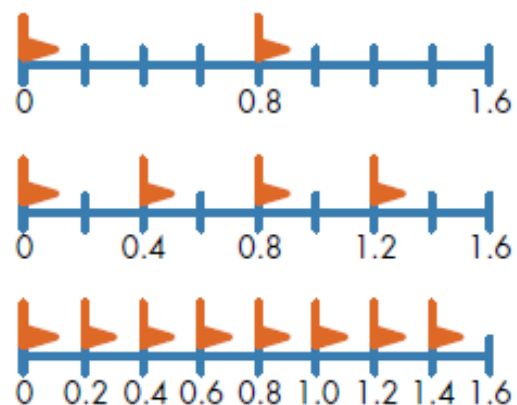
Two ways to create background music: Repeat the sound after it finishes (left) or start the sound over after playing it for a certain amount of time (right).

You can save audio files on your computer in many formats, but Scratch only recognizes two: WAV and MP3. There are three command blocks that allow you to use these sound files in your applications: **play sound**, **play sound until done**, and **stop all sounds**. Both of the first two blocks play a given sound. The play sound command lets the next command start before the sound finishes playing, while play sound until done won't move on to the next command until the end of the sound. The stop all sounds command immediately turns off any sound that is playing.

### playing drums and other sounds

OPEN PROJECT [BeatsDemo.sb2](#)

repeat 2  
play drum 1 for 0.8 beats  
repeat 4  
play drum 1 for 0.4 beats  
repeat 8  
play drum 1 for 0.2 beats



play drum 1 for 0.25 beats

choose from 18 drum sounds for a certain number of beats

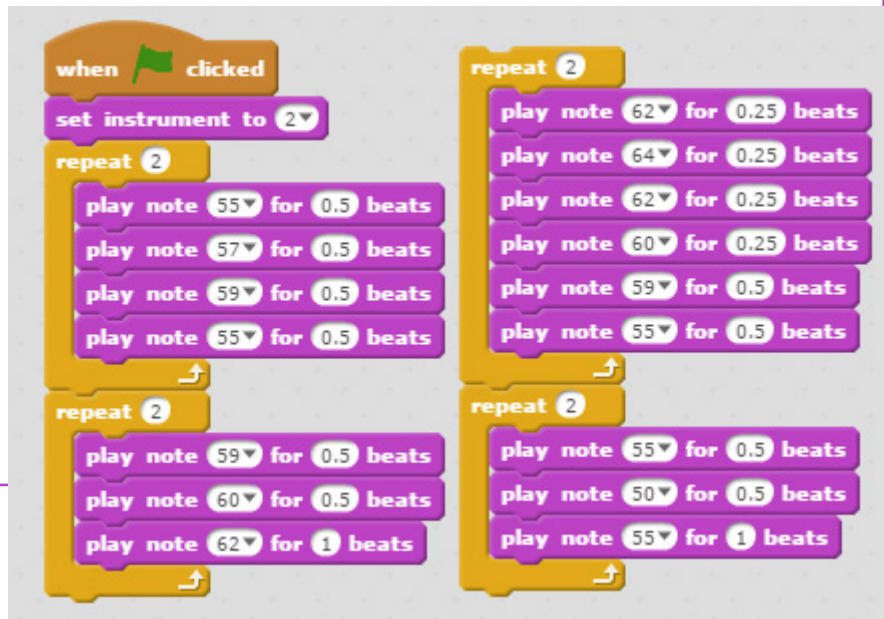
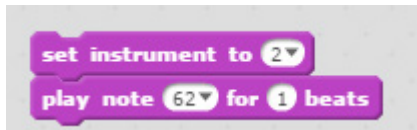




## composing music

OPEN PROJECT [FrereJacques.sb2](#)

Scratch also contains two commands that allow you to play musical notes and compose your own music. The **play note for beats** command plays the note you choose, from 0 to 127, for a number of beats you specify. The **set instrument to** block tells Scratch which instrument the note should sound like.



## controlling volume

OPEN PROJECT [VolumeDemo.sb2](#)

The **change volume by** block reduces or increases the volume by the number you enter.



### TRY IT OUT

The file VolumeDemo.sb2 simulates a cat walking into a forest. The application uses the change volume by command to make the cat's sound fade away as it goes deeper into the woods. Come up with some ideas to make this simulation more real and try to implement them.

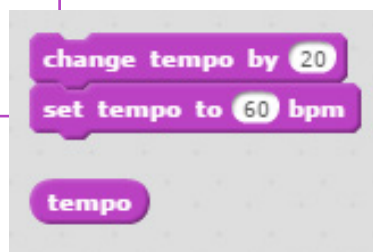
## setting the tempo

OPEN PROJECT [TempoDemo.sb2](#)

The last three blocks in the Sound palette are related to the tempo, or speed, at which drums and notes are played. Tempo is measured in beats per minute (bpm). The higher the tempo, the faster the notes and drums will play.

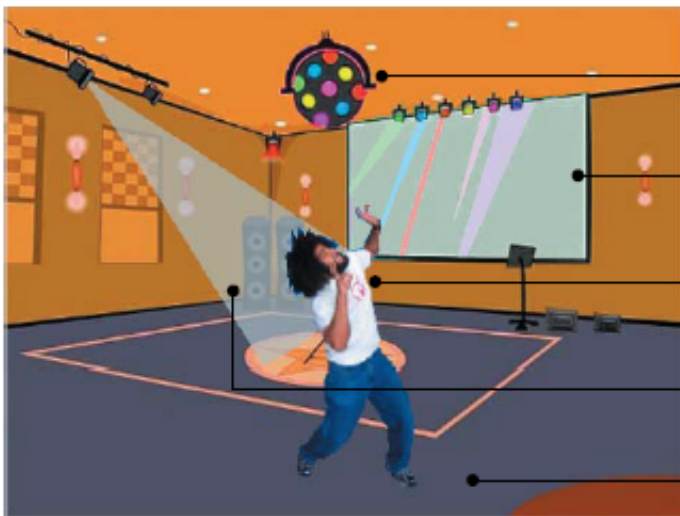
### TRY IT OUT

Open the file TempoDemo.sb2 and run it to see the set tempo to bpm and change tempo by commands in action



# PROJECT 1

## DANCING ON THE STAGE



The Ball sprite changes its color like a disco ball.

The Board sprite also changes color to simulate the spotlights.

The Dancer sprite will dance on the Stage to music.

The SpotLight sprite will follow the dancer as he moves around.

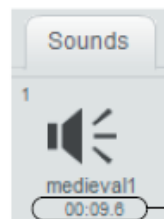
Stage background

### CHANGING BACKDROP

**Step1:** Open new file (from the File menu), delete Sprite1

**Step2:** Change backdrop to party room from the Indoors category. Delete the default white backdrop in the backdrops tab.

**Step 3:** Add the following blocks to the stage (import Medieval1 sound from Module 3 folder)



Duration of the audio clip (9.6 seconds)

### ADDING SPRITES:

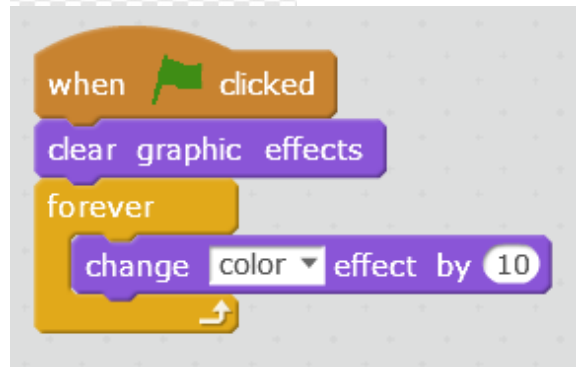
**Step 4:** Add 'Dancer' sprite from people category



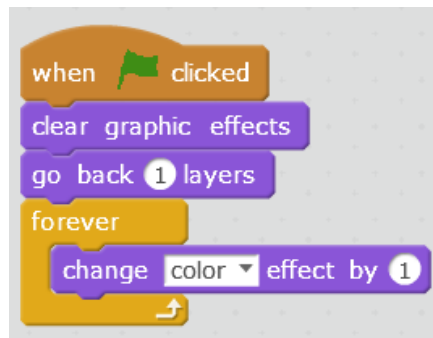
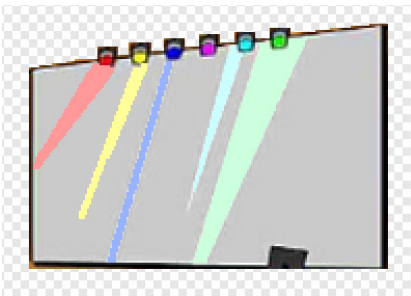
The Dancer moves 20 steps to the right, changes its costume, moves 20 steps to the left, and changes its costume again. These steps are repeated forever to make him look like he's really dancing. The script also changes the fisheye effect slightly with every step for some variety. Click the green flag to test this new addition to the program. You should hear the background music and see the Dancer moving left and right on the Stage.

### Step 5. Create Ball sprite

To create the Ball sprite, click the thumbnail of the Stage to select it and then select the Backdrops tab. Right-click the thumbnail of the party room backdrop and select save to local file from the pop-up menu. This brings up a dialog that allows you to save the backdrop image locally. Remember where you saved this image because you'll import it back in a moment. Click the Upload sprite from file button (above the Sprite List) and select the image you just saved. This creates a new sprite whose costume is the same as the backdrop image. Name this sprite Ball and edit its costume in the Paint Editor to remove everything but the colorful ball depicted in the image above.



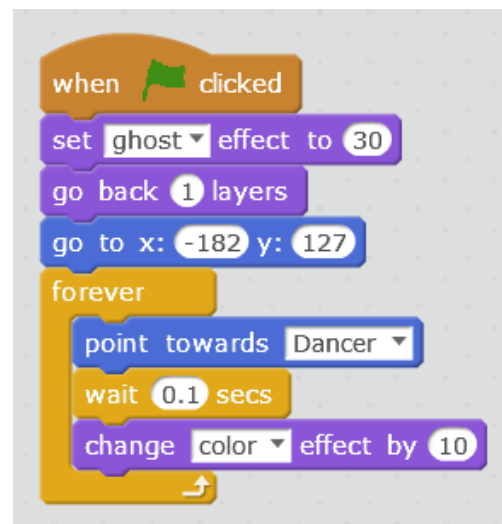
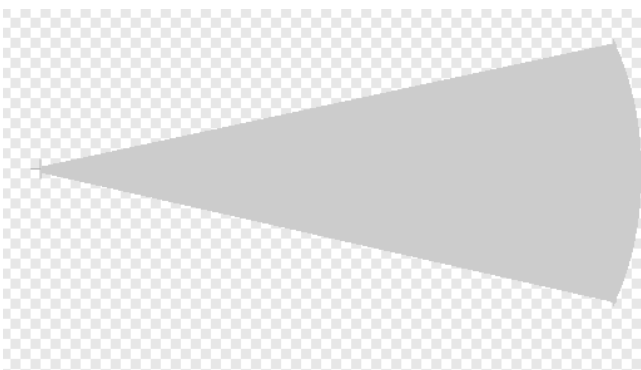
### Step 6. Create the Board sprite the same way you created the Ball sprite.



Because the Board sprite overlaps with the Dancer, the script sends the Board two layers to the back so the Dancer will always be in the front. You can do the same thing by selecting the Dancer sprite and clicking the go to front block from the Looks palette

### Step 7. Create SpotLight sprite

Use Paint New Sprite to paint the shape depicted below. The center of the image is at the tip of the cone shape, which represents a light beam



The script first sets the sprite's ghost effect to 30 to make it transparent so that it won't obscure the backdrop. The script then sends this sprite one layer back, which places the light beam behind the dancer. The sprite is then positioned so that the light beam appears to be emanating from the spotlight. You'll have to choose the x- and y-coordinates based on your drawing. After that, the script commands the light beam to follow the dancer (using the point towards command) and change its color forever.





## PROJECT 2

### FIREWORKS



Clones of the Rocket sprite will explode and produce colorful sparks that fall to the ground.

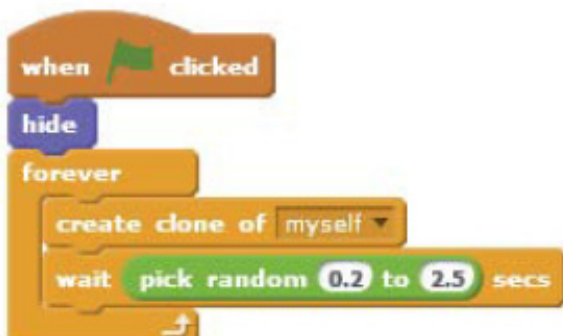
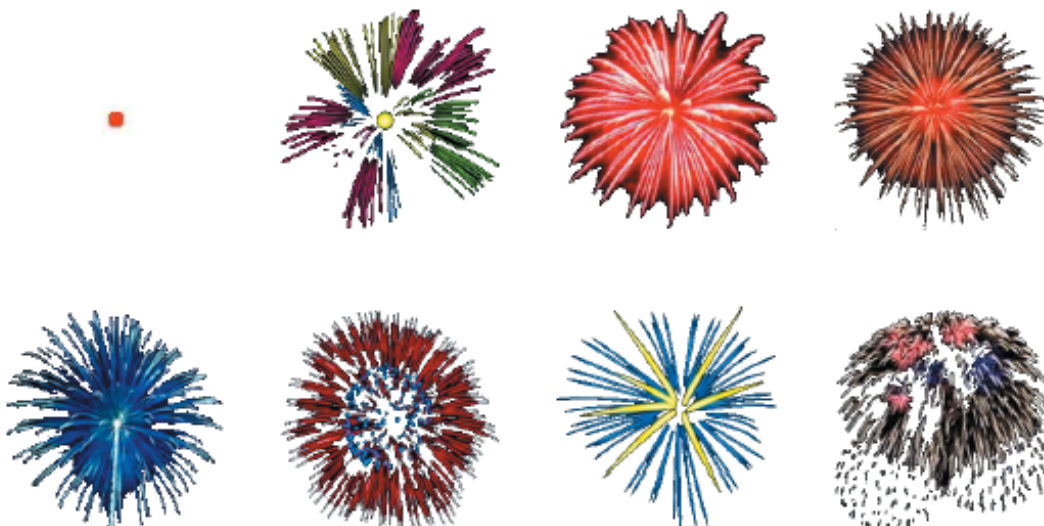
The City sprite

In this project, you'll make a simple firework animation that floods the sky with colorful sparks. The firework rockets will explode at random times, producing sparks that fall as though acted upon by gravity and fade slowly with time,

#### CHANGING BACKDROP

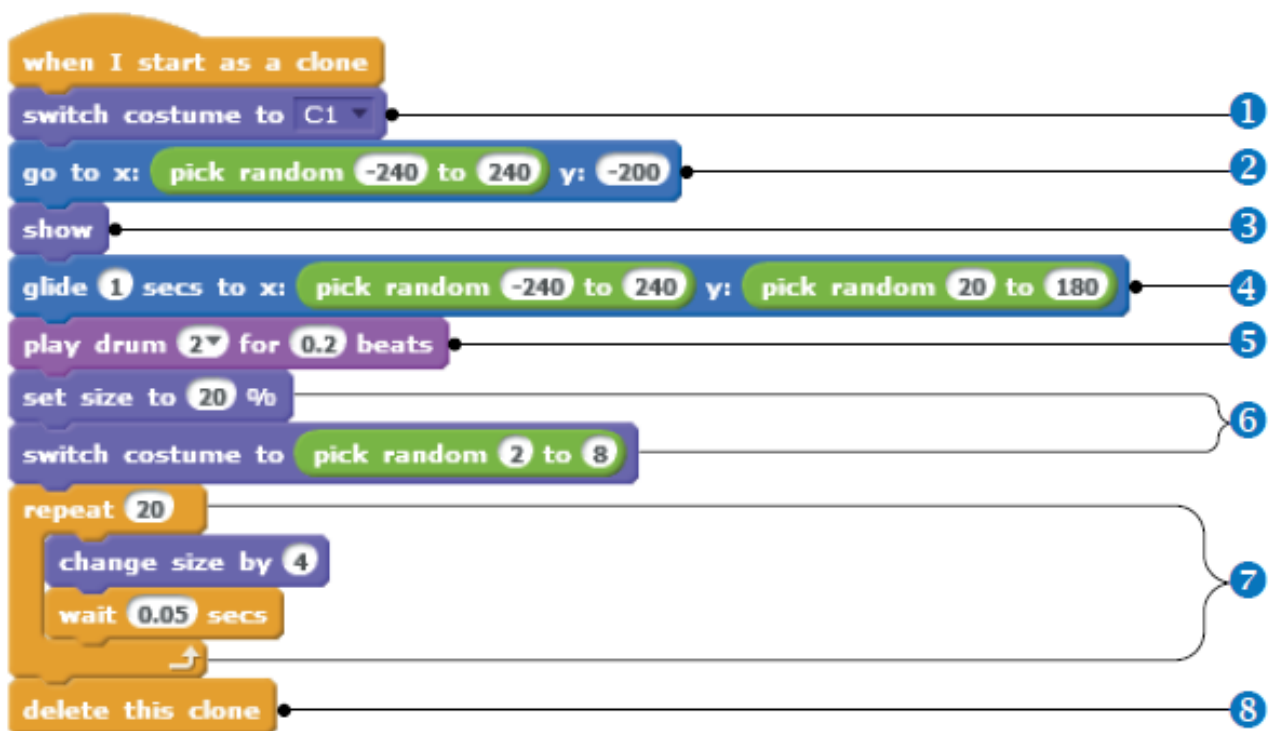
**Step1:** Start by opening the file Fireworks\_NoCode.sb2, which contains the initial setup of the application without any scripts the application contains two sprites: the City sprite and the Rocket sprite. The City sprite shows an image of tall buildings that you can animate in any way you like. The Rocket sprite will continuously create clones that explode in the dark sky, producing the fireworks.

**Step2:** Add Scripts to the rocket sprite. The picture below shows 8 costumes of the rocket sprite.



The script runs when the user clicks the green flag. After hiding the Rocket sprite, it starts a forever loop to create clones of itself at random times. Since the clones inherit the visibility state of the Rocket sprite, all created clones will be hidden at first. We now need to tell the cloned rockets what to do. This script is shown below





The cloned rocket starts by putting on its first costume (1) (the small red dot). It then moves to a random horizontal position at the bottom of the Stage (2), shows itself (3), and glides to a random position (4) in the upper part of the Stage (somewhere above the buildings). This part of the script simulates the launch of the rocket, and if you run it, you will see a red dot moving from the ground to the sky. When the dot reaches its final point in the sky, it explodes due to instructions in the second part of the script. First, the clone plays a short drum sound (5) (to simulate the sound of an explosion). Fireworks explosions start small and expand, so the clone sets its initial size to 20% and picks one of its other costumes randomly (6). It then starts a repeat loop (7) to grow the firework. Every pass through the loop, the clone increases its size by 4. At the end of the loop, the clone deletes itself (8).

## CHALLENGE TASKS

1. Open the application [Zebra.sb2](#), shown on the right. The application contains a single sprite (the Zebra), which has three costumes. Write a script that makes the Zebra move across the Stage and switch among its costumes to create the illusion of running.



2. Open the application [Joke.sb2](#). Finish the scripts for the Boy and the Girl sprites to tell any joke you want.

3. Open the application [ChangingHat.sb2](#). The hat in this application is a sprite that has five costumes. Create a script to switch the Hat's costume when it is clicked. Then create a game in which the player dresses characters by clicking on different pieces of clothing

3. Open the application [Wolf.sb2](#), shown on the right. When you click the green flag, the Wolf will play the Wolf-Howl sound, which takes about 4 seconds. Create a script that changes the Wolf's costumes in sync with the sound. (Hint: Insert a wait block with an appropriate time delay after each costume switch.)

