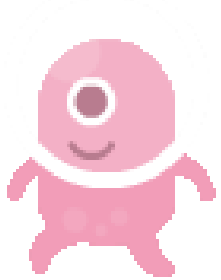
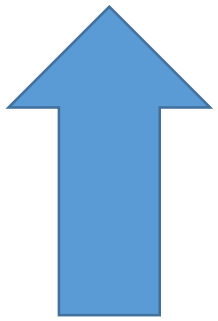
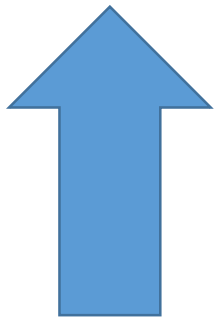
Sprites

Inside GameLab

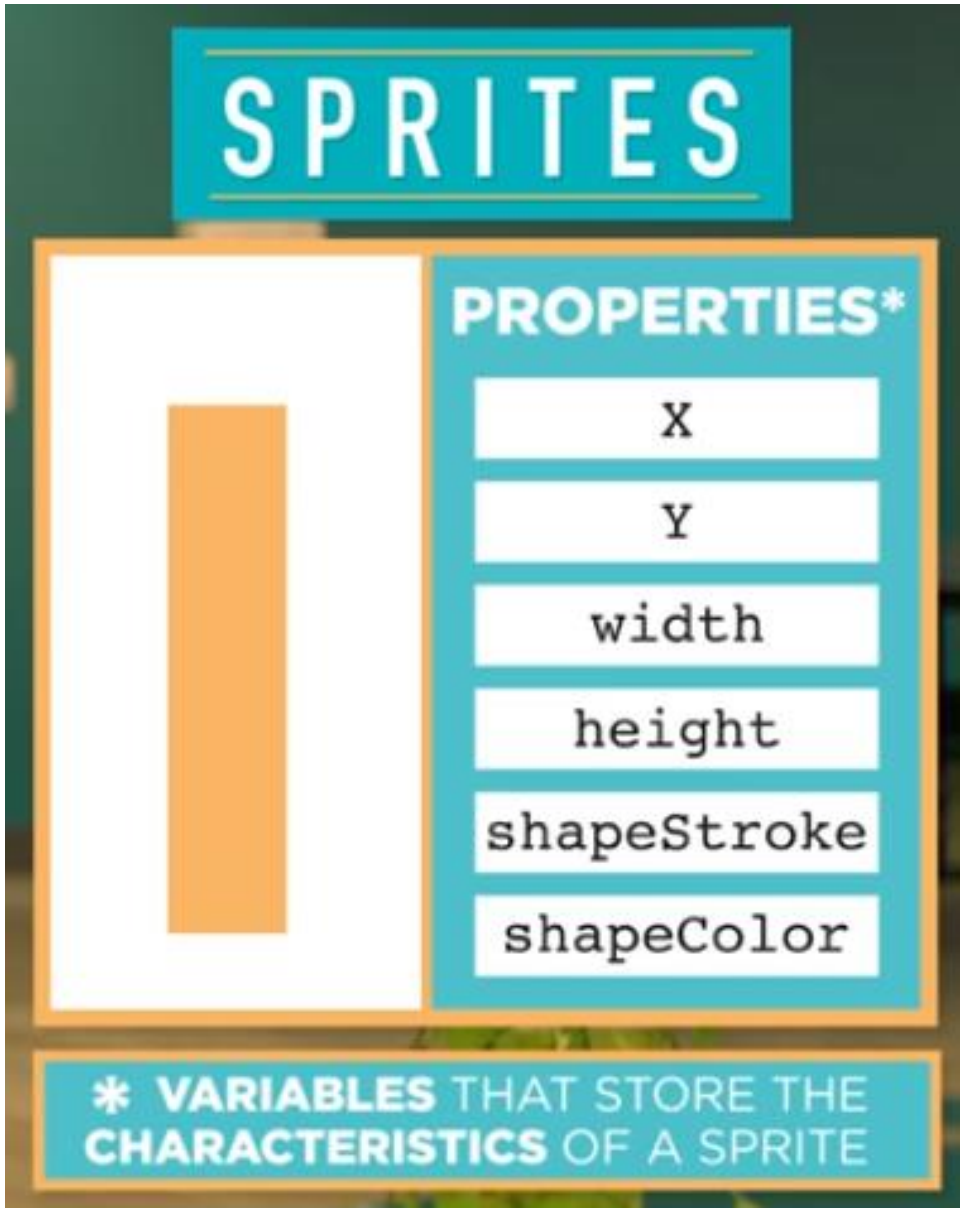






Previous Units

This Unit



A sprite is

(a)An animation

(b)Properties that determine how the animation is drawn.

Why are they useful?

By grouping all of the variables under one variable (the sprite name), they are much easier to track and manipulate.

Each aspect of the sprite is controlled in a different tab.

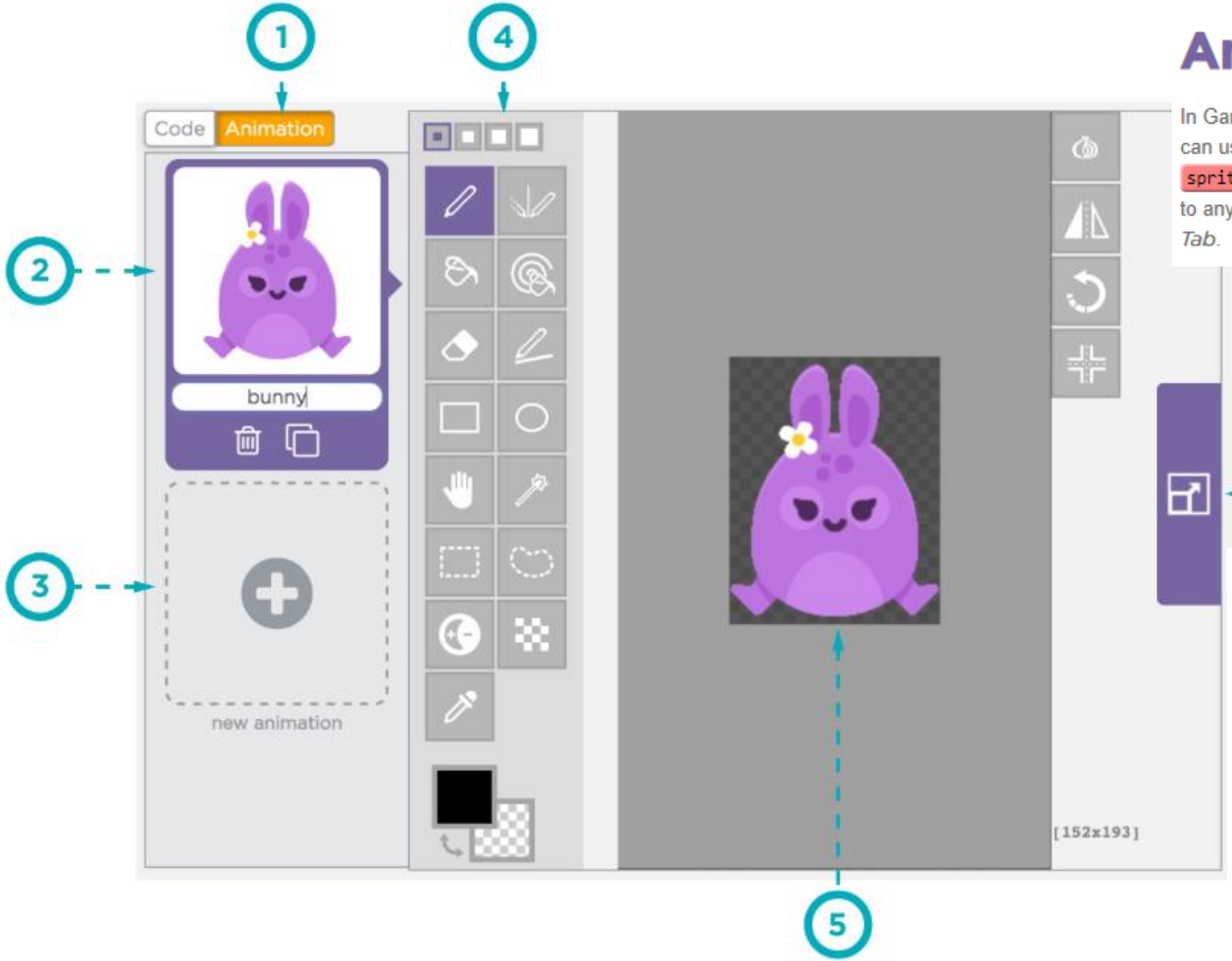
A sprite is

(a)An animation

(b)Properties that determine how the animation is drawn.

Animation Tab

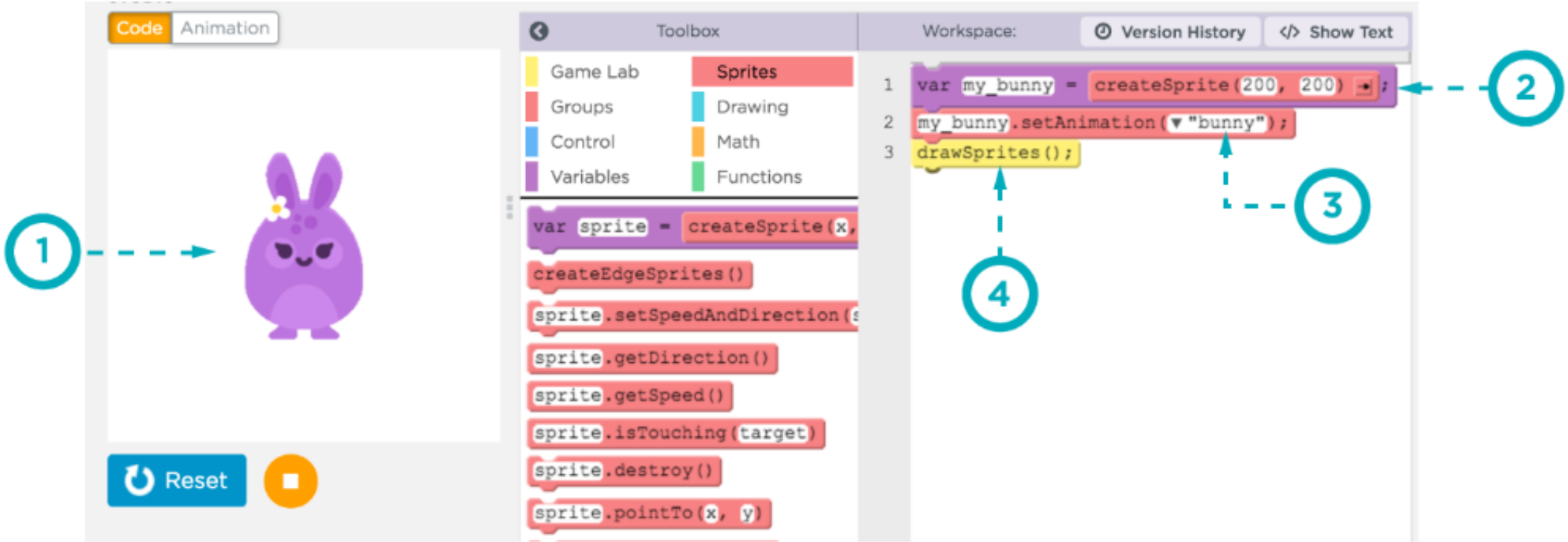
In Game Lab, *Animations* are drawings or other images that you can use to the change the look of your sprites. You can use the `sprite.setAnimation()` block to change your sprite's animations to any of the animations that you have added in the *Animation Tab*.



1. Use these buttons to switch between the animation tab and the code tab.
2. This column shows all of the animations you've created. Click on an animation from here to edit it.
3. Click the plus sign to add a new animation. You can either select an animation from the library, import an image from your computer, or draw one from scratch.
4. The drawing tools in this column allow you to draw or modify animations.
5. Draw or edit your image on this canvas.
6. Open up this drawer to change the overall size of your image or canvas.

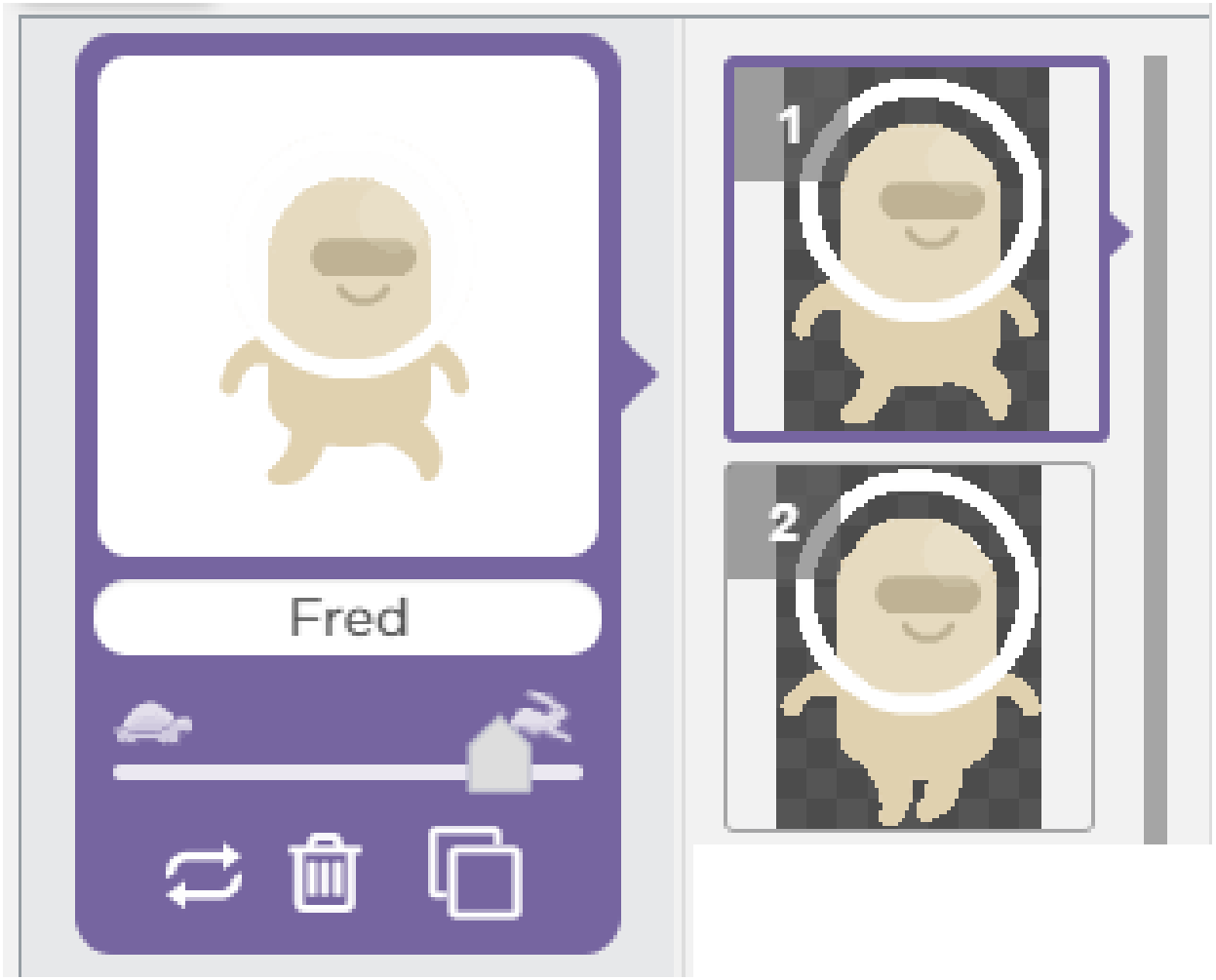
Creating a Sprite

You can create a new *sprite* using the `createSprite()` block.



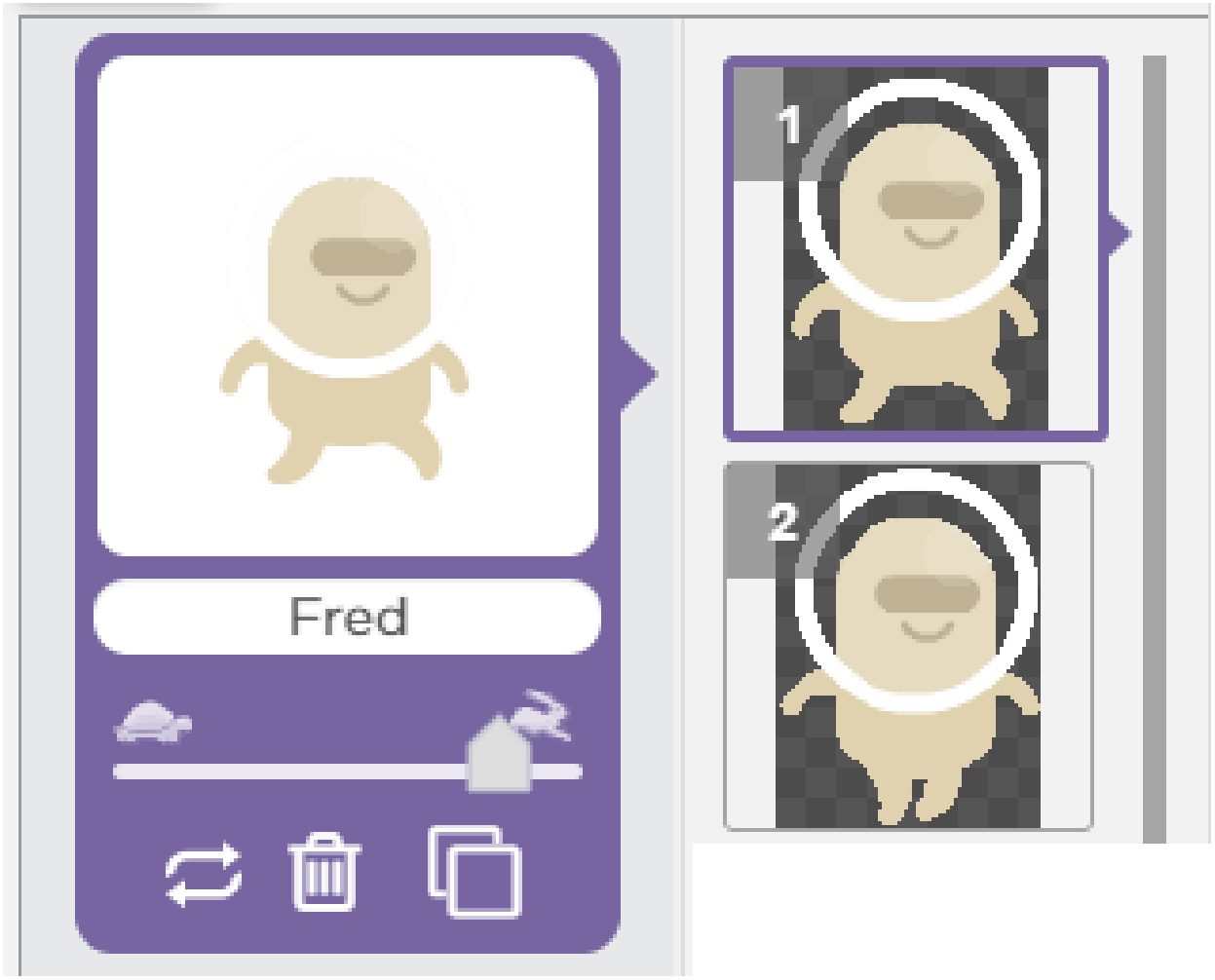
1. The *sprite* as drawn on screen. In this case, our *sprite* is located at (200, 200), the center of the screen, and has been assigned the animation "bunny"
2. The `createSprite()` block, which creates a new *sprite* at (200, 200) and assigns it to the variable label `my_bunny`. Note that just creating the *sprite* **doesn't** yet draw it on the screen. That will happen later.
3. The `sprite.setAnimation()` block assigns an animation (or image) to the *sprite*. In this case, we're using an animation named "bunny", which was added in the animation tab. Notice that instead of the default variable name `sprite`, we've updated this to `my_bunny.setAnimation()` so that it changes the animation of the `my_bunny` *sprite*.
4. Because *sprites* are just values stored as variables, they don't automatically get drawn on the screen. The `drawSprites()` block tells Game Lab to draw all of the *sprites* that have been created onto the screen.

This is an animation.



This is an animation.

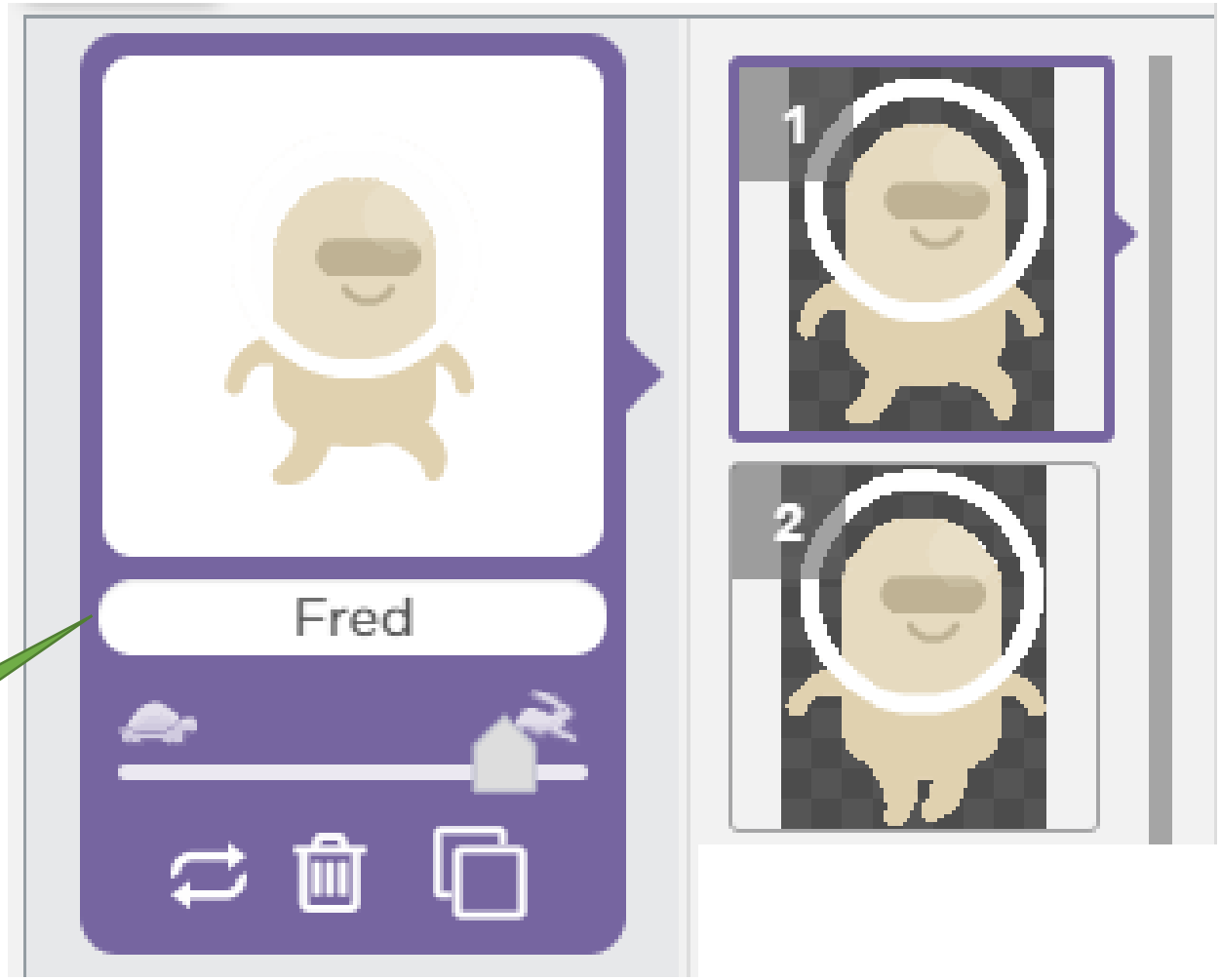
What is its name?



This is an animation.

What is its name?

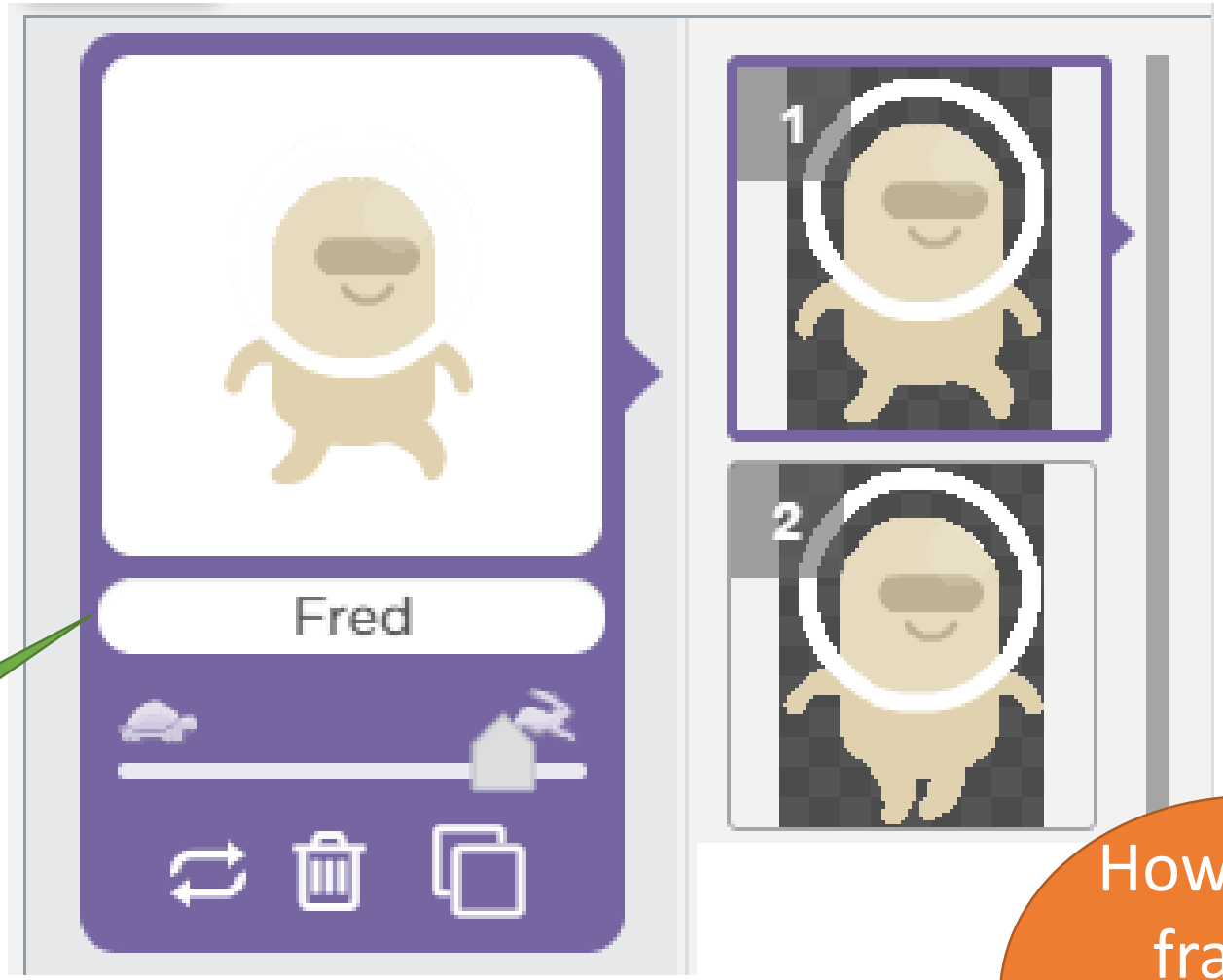
Fred



This is an animation.

What is its name?

Fred

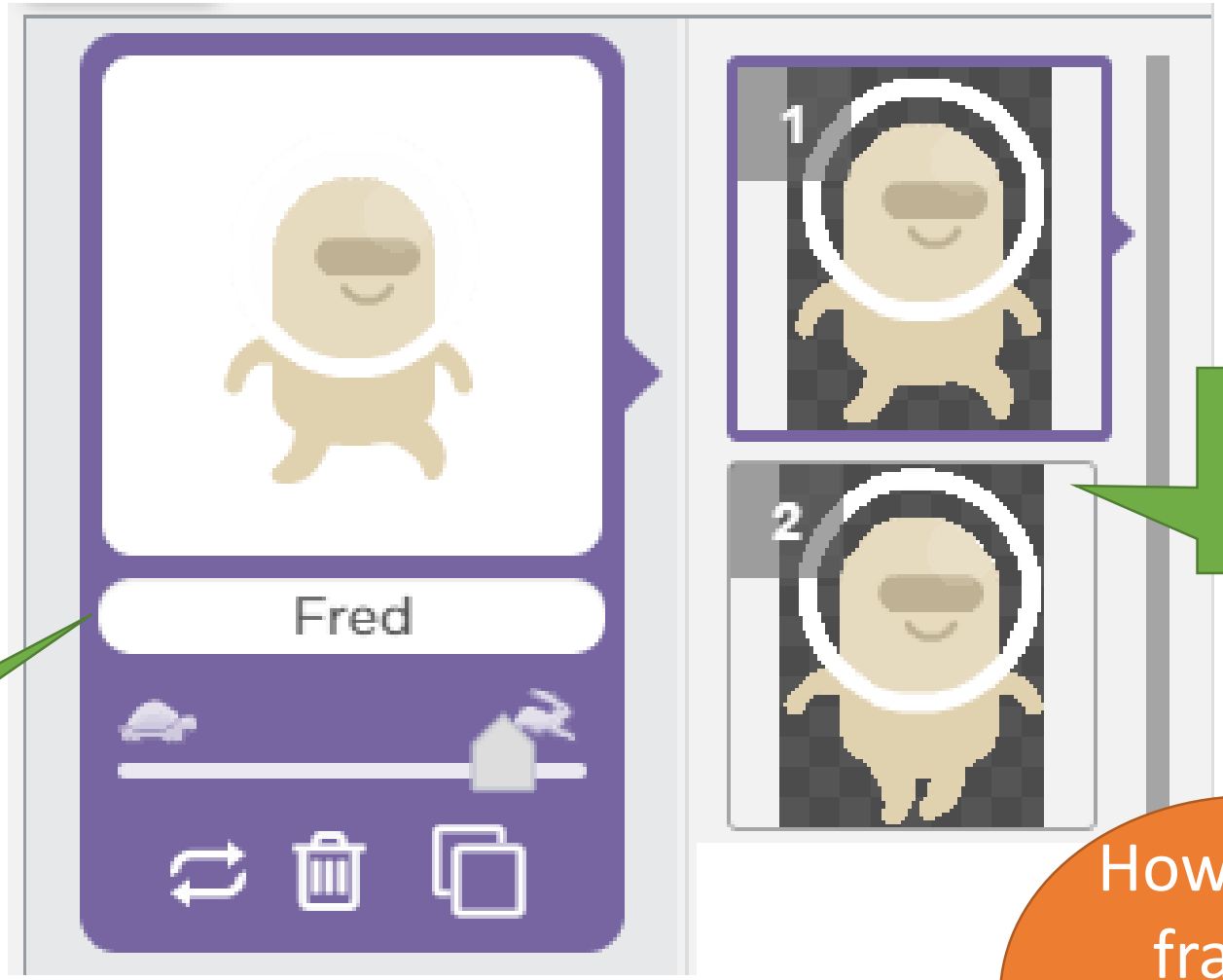


How many frames does it have?

This is an animation.

What is its name?

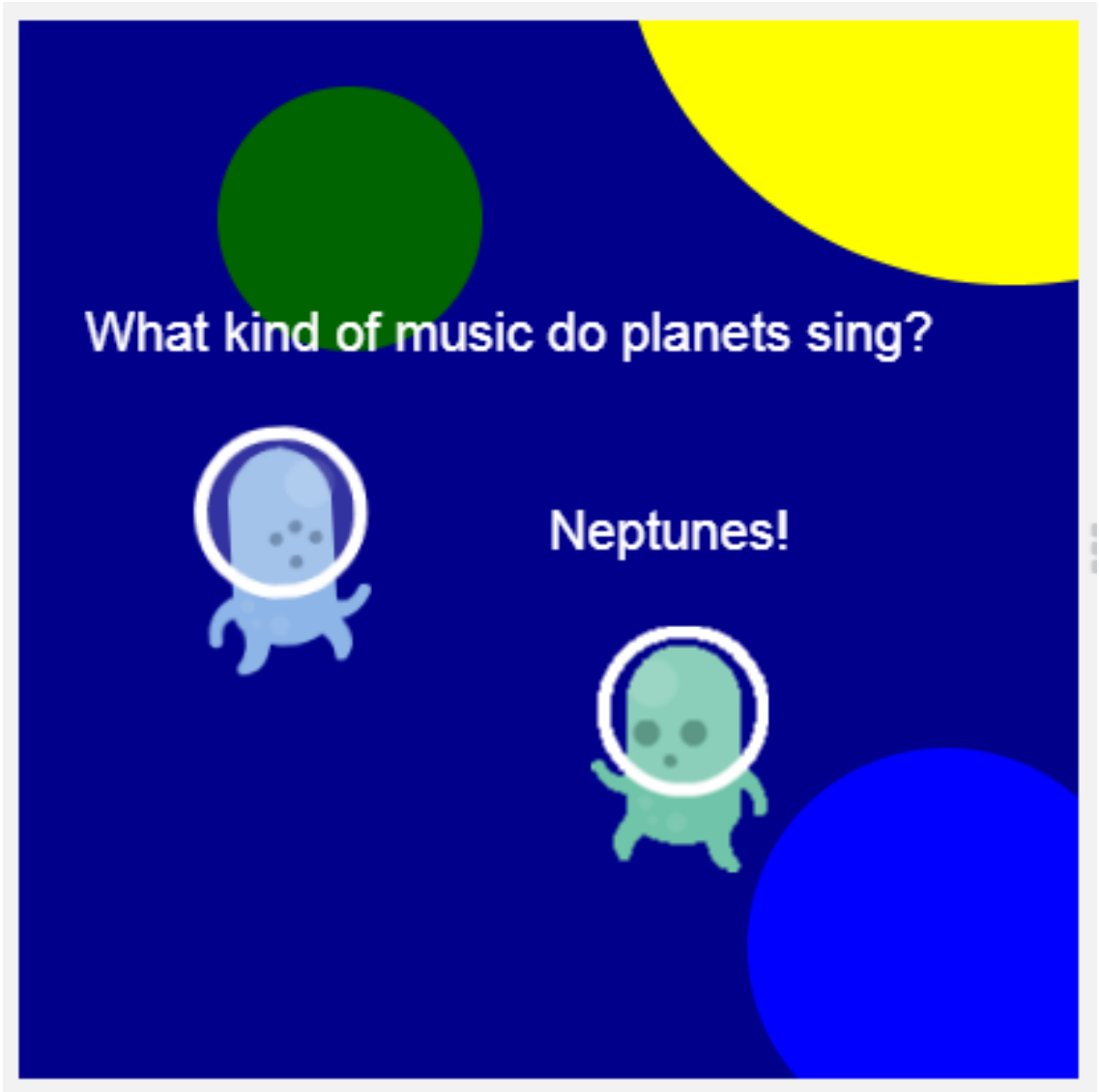
Fred



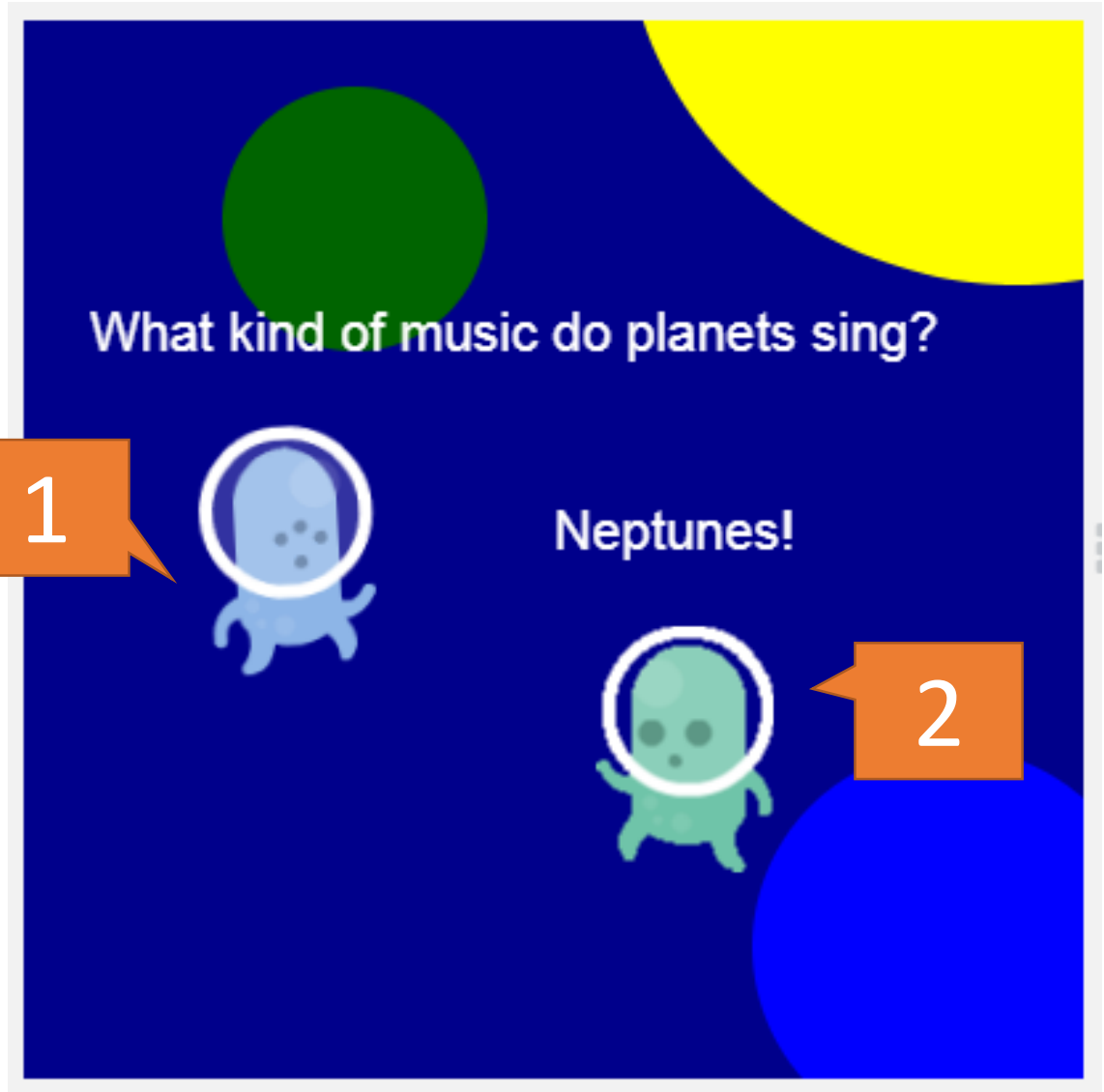
2

How many frames does it have?

How many animations would need to be set up to make this screen?

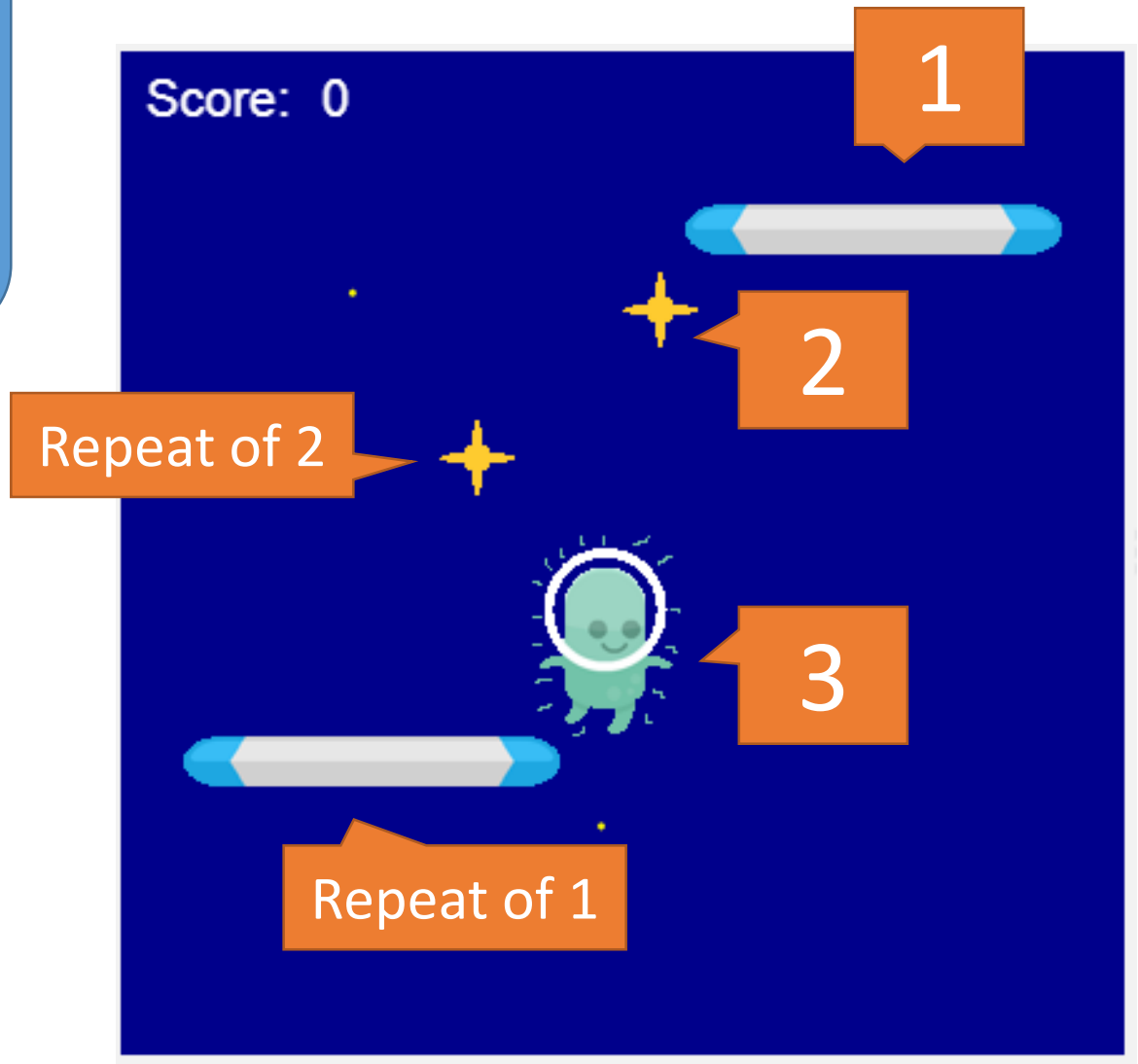


How many animations would need to be set up to make this screen?



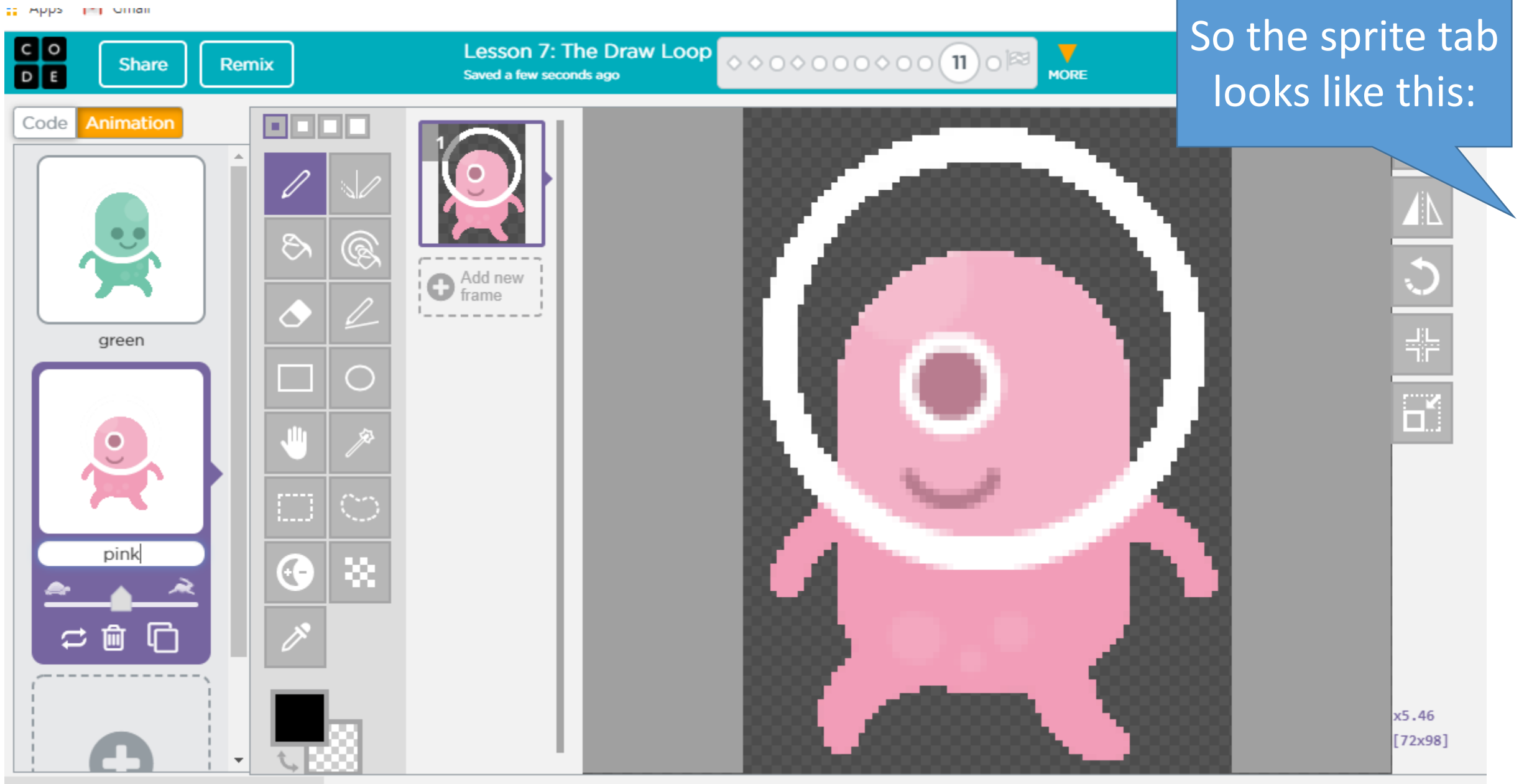
How many animations would need to be set up to make this screen?

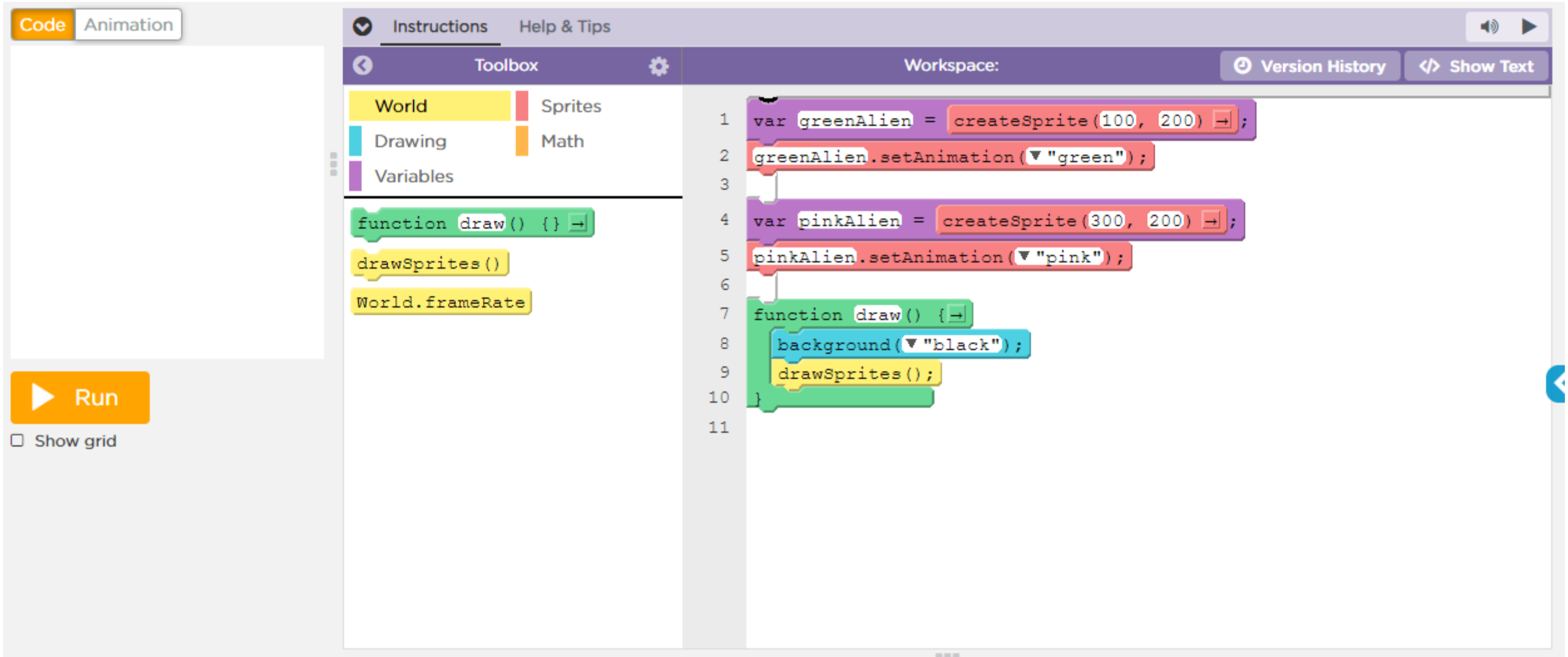
How many animations would need to be set up to make this screen?



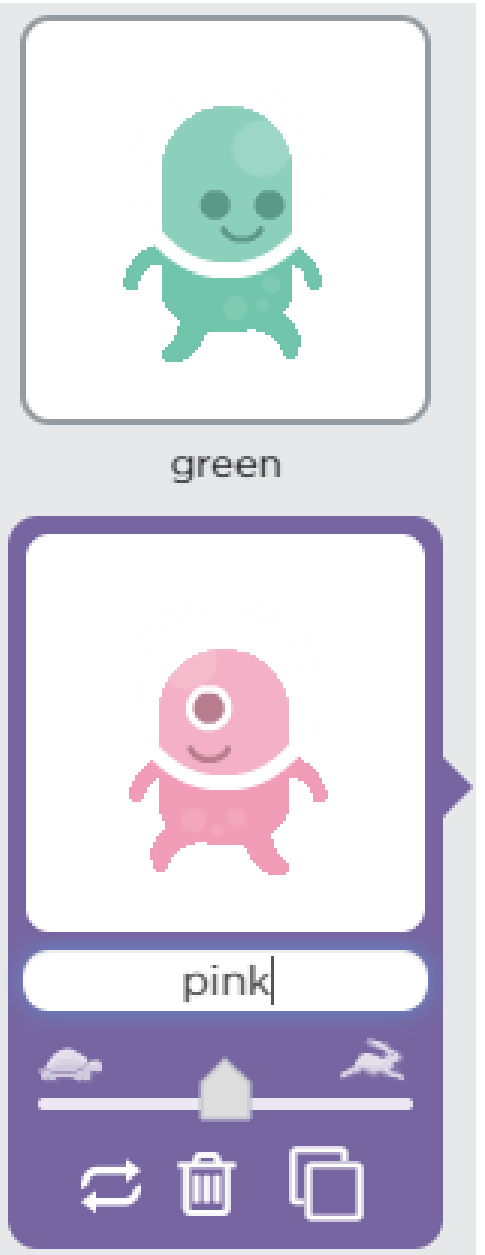
Let’s say we want to make this:

How many sprites?


Share
Remix
Lesson 7: The Draw Loop
Saved a few seconds ago
11
MORE
Code
Animation
green
pink
Add new frame
So the sprite tab looks like this:
x5.46
[72x98]


Code
Animation
Run
Show grid
Instructions
Help & Tips
Toolbox
Workspace:
Version History
Show Text
World
Sprites
Drawing
Math
Variables
function draw() {}
drawSprites()
World.frameRate
1 var greenAlien = createSprite(100, 200);
2 greenAlien.setAnimation("green");
3
4 var pinkAlien = createSprite(300, 200);
5 pinkAlien.setAnimation("pink");
6
7 function draw() {
8 background("black");
9 drawSprites();
10 }
11

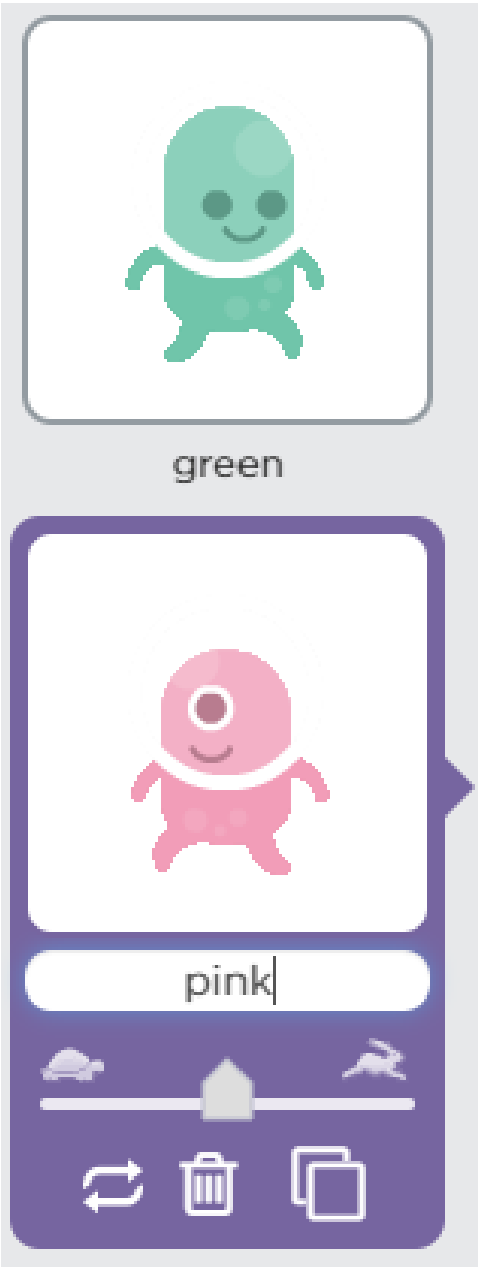
```
1  var greenAlien = createSprite(100, 200);
2  greenAlien.setAnimation("green");
3
4  var pinkAlien = createSprite(300, 200);
5  pinkAlien.setAnimation("pink");
6
7  function draw() {
8    background("black");
9    drawSprites();
10 }
11
```



```
1  var greenAlien = createSprite(100, 200);
2  greenAlien.setAnimation("green");
3
4  var pinkA          teSprite(300, 200
5  pinkAlien.s        ion("pink");
6
7  function draw() {
8    background("black");
9    drawSprites();
10 }
11
```

Sprite Name

Animation Name



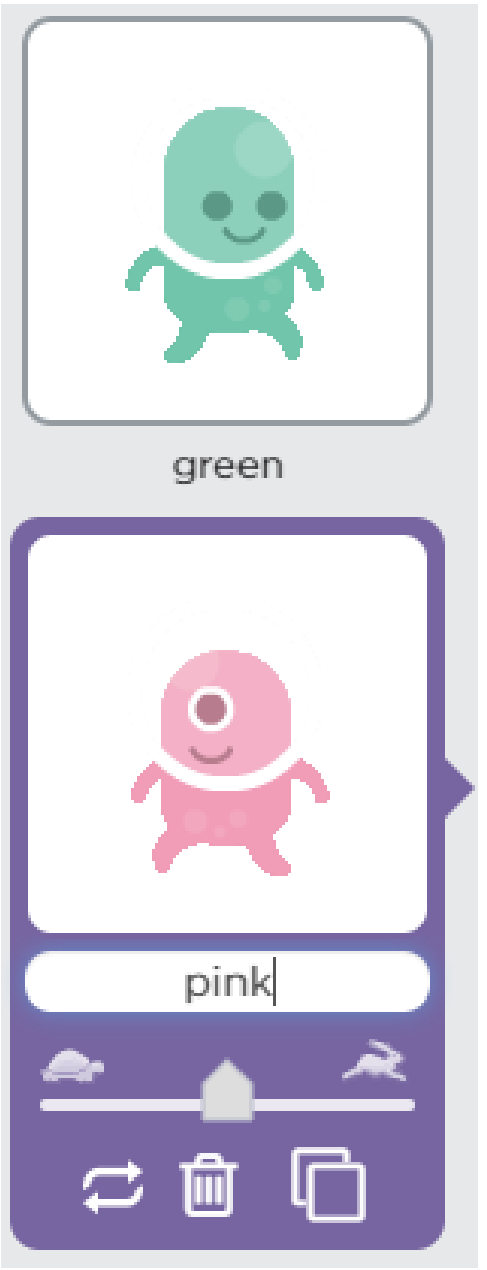
```
1  var greenAlien = createSprite(100, 200);
2  greenAlien.setAnimation("green");
3
4  var pinkA          teSprite(300, 200
5  pinkAlien.s          ion("pink");
6
7  function draw() {
8    background("black");
9    drawSprites();
10 }
11
```

Sprite Name

Animation Name

Background First

Sprites on top



Sprite Properties

If you think of a *sprite* as a collection of values that represents an object in the real world, then the *properties* of a sprite are like its attributes. Some of the most common sprite properties include:

`sprite.x`

`sprite.y`

`sprite.rotation`

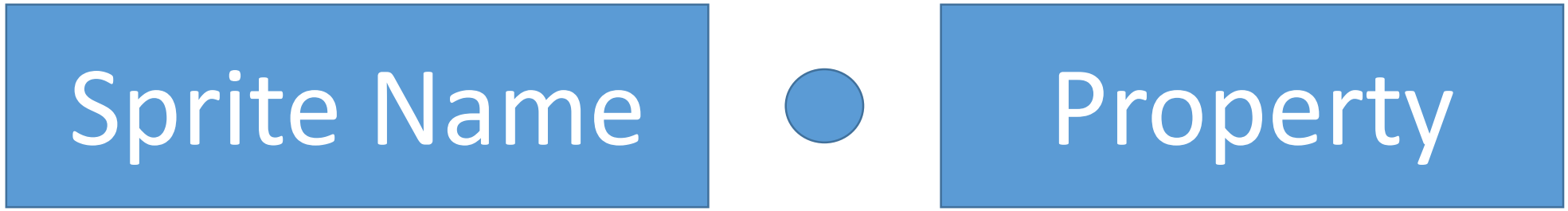
`sprite.scale`

`sprite.visible`

Dot Notation

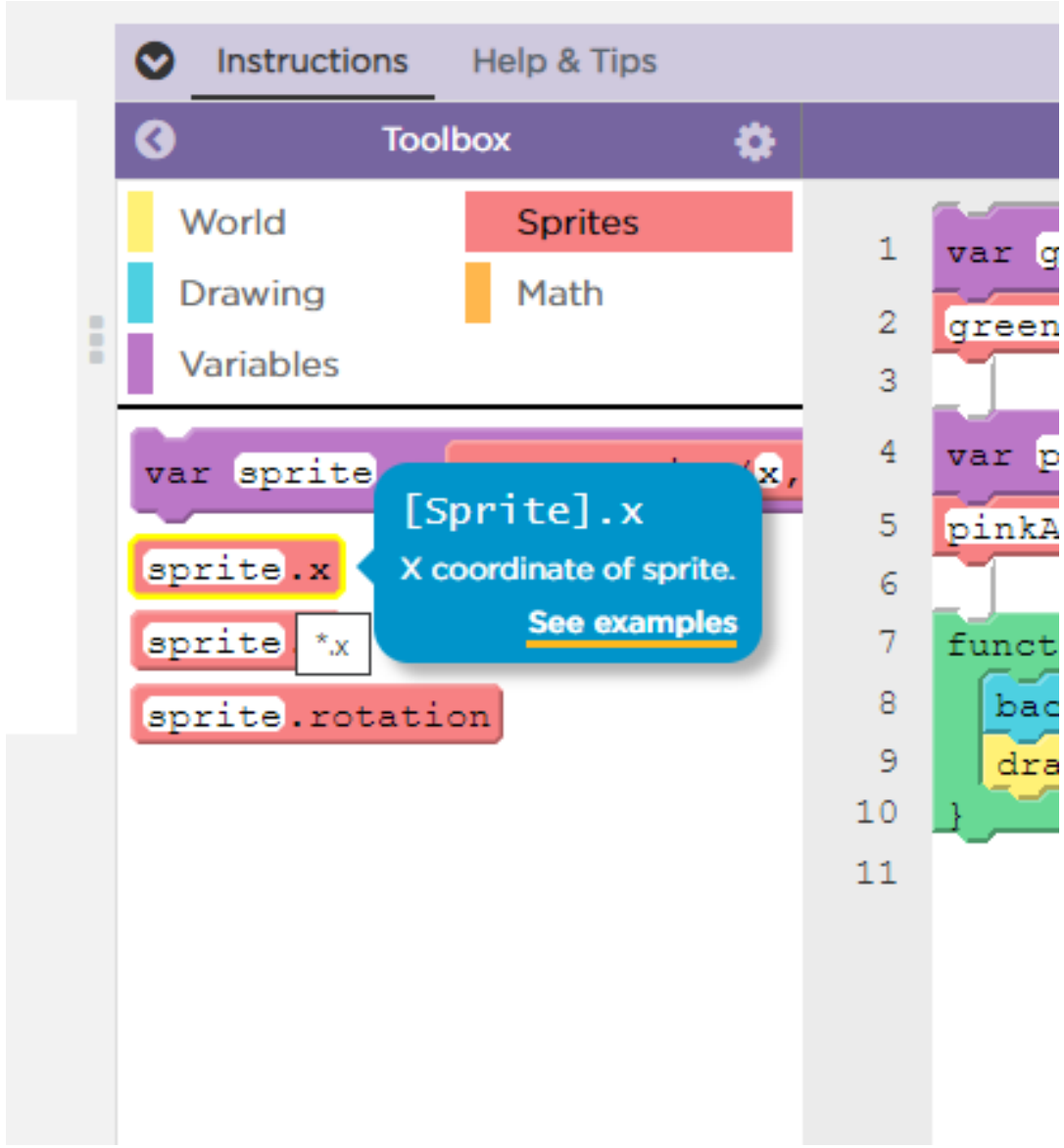
Notice that all of the examples above follow a common pattern of **sprite label . sprite property**. We call this format *dot notation*. The first part will *always* be unique to the sprite that you want to modify, but and the second part will *always* be one of the properties common to all sprites.

Dot Notation:



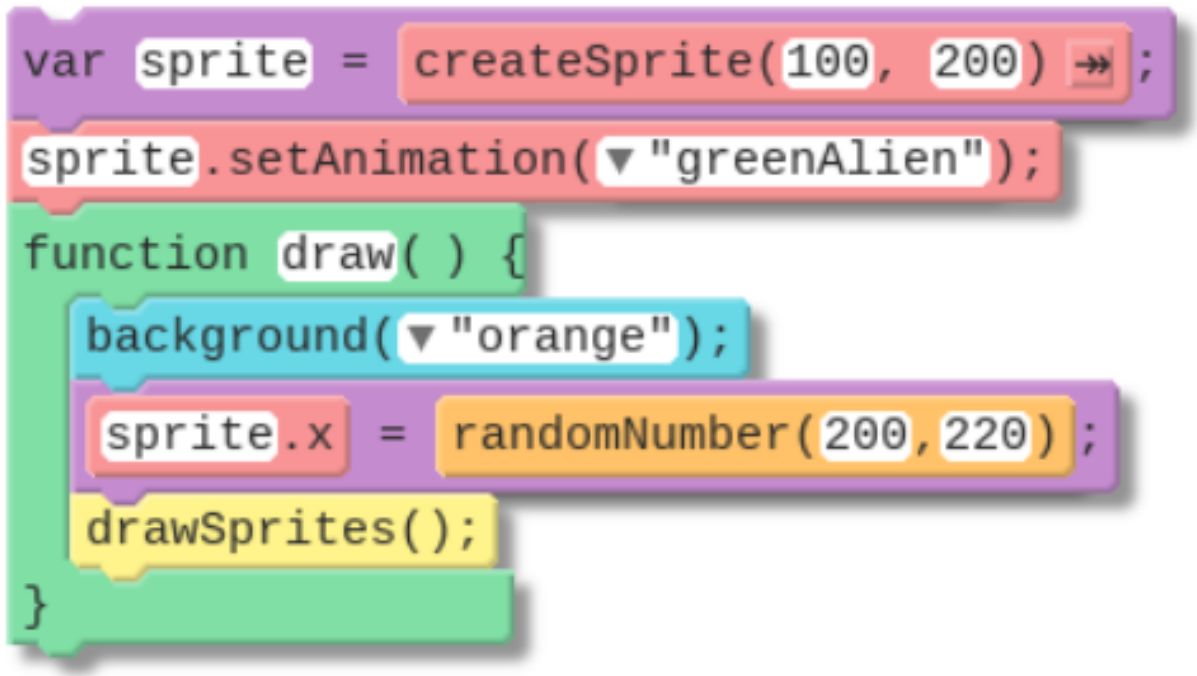
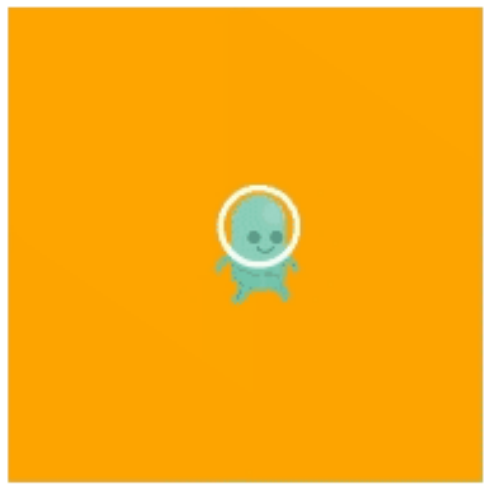

```
greenAlien.setAnimation("green");
```

Keeps the code organized.



If you mouse over the items in the toolbox shows you what they mean.

Draw Loop with Sprites

Code	Animation
	

```
var sprite = createSprite(100, 200);
sprite.setAnimation("greenAlien");
function draw( ) {
  background("orange");
  sprite.x = randomNumber(200,220);
  drawSprites();
}
```

This program creates a sprite and sets its animation outside the draw loop. Then it repeatedly gives the sprite a new x location and redraws the white background and the sprite to make it move.