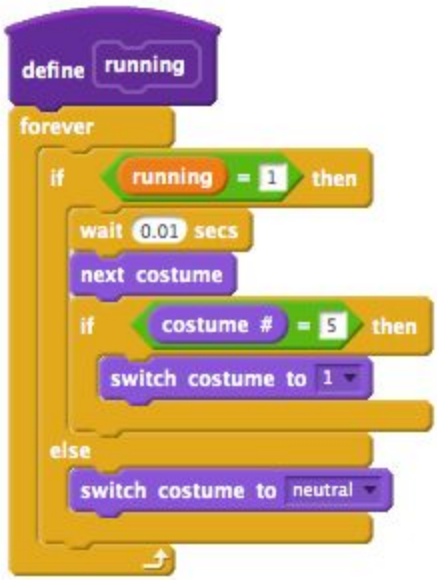
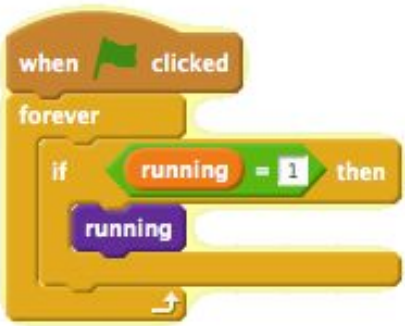
Jumping

Programming game characters

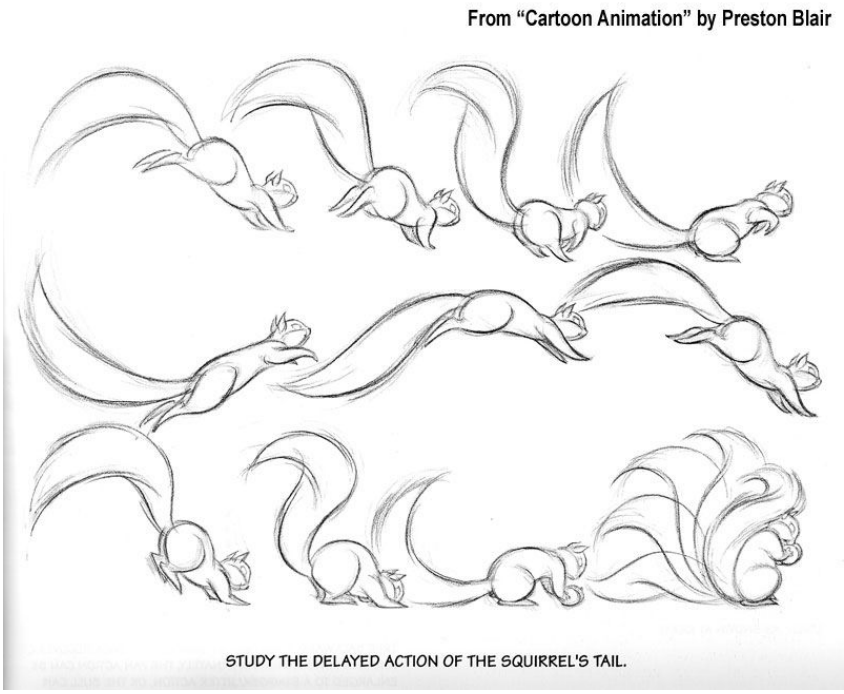
In our last lesson we got our character to walk using 4 costumes and some code.
To cycle between costumes when arrow keys were pressed.

Your code should look like this after last weeks lesson.
(In my I created a resting pose called “neutral” as costume number 5, I set it so that the squirrel would be in this costume when not moving, but the walk cycle would not include it)





So How do we make a smooth jump?

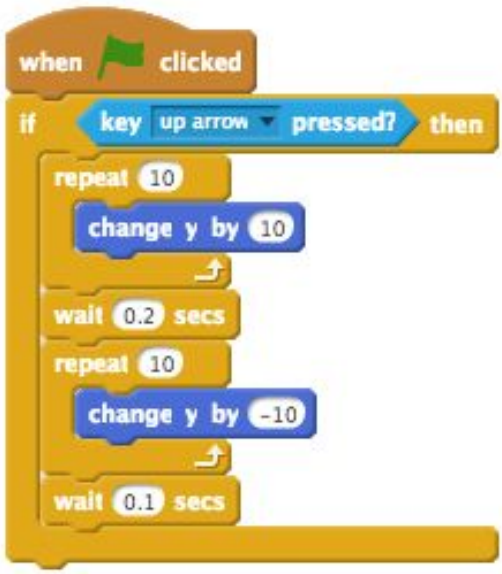


Let's work on this basic function. So that when the arrow key is pressed our squirrel will jump in a smooth arc.

Remember this picture?

When anything jumps or bounces, it goes up, then slows down at the top of the arc, before changing directions and coming down again.

If we only use a *change X by* block, then the sprite will suddenly jump to that location. and we can't use Glide because we don't know the exact point it will need to go to.
So we will use code to change the Y value a little bit at a time.



Here is code to make the jump work.

1. We are using an If statement to check if the up arrow key is pressed.

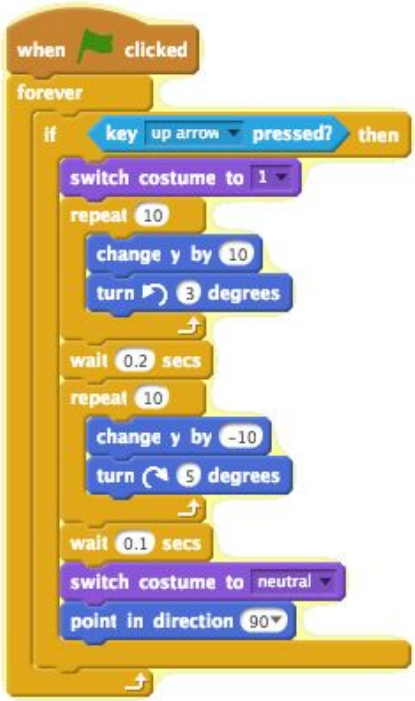
2. We will use a *repeat* look to change the Y by a little bit , 10 times. So instead of saying |change Y by 100" we are saying "change Y by 10" 10 times, the result is that we can see the character moving up

3. We wait for a 0.2 seconds, that is the top of the jump arc.
4. We make another repeat where the character moves downwards again. and we add a brief pause.

Try this out. Did your character make a jump?
What did you maybe forget?

How about a **forever loop** around the if statement?

Now to make the jump a little more natural.

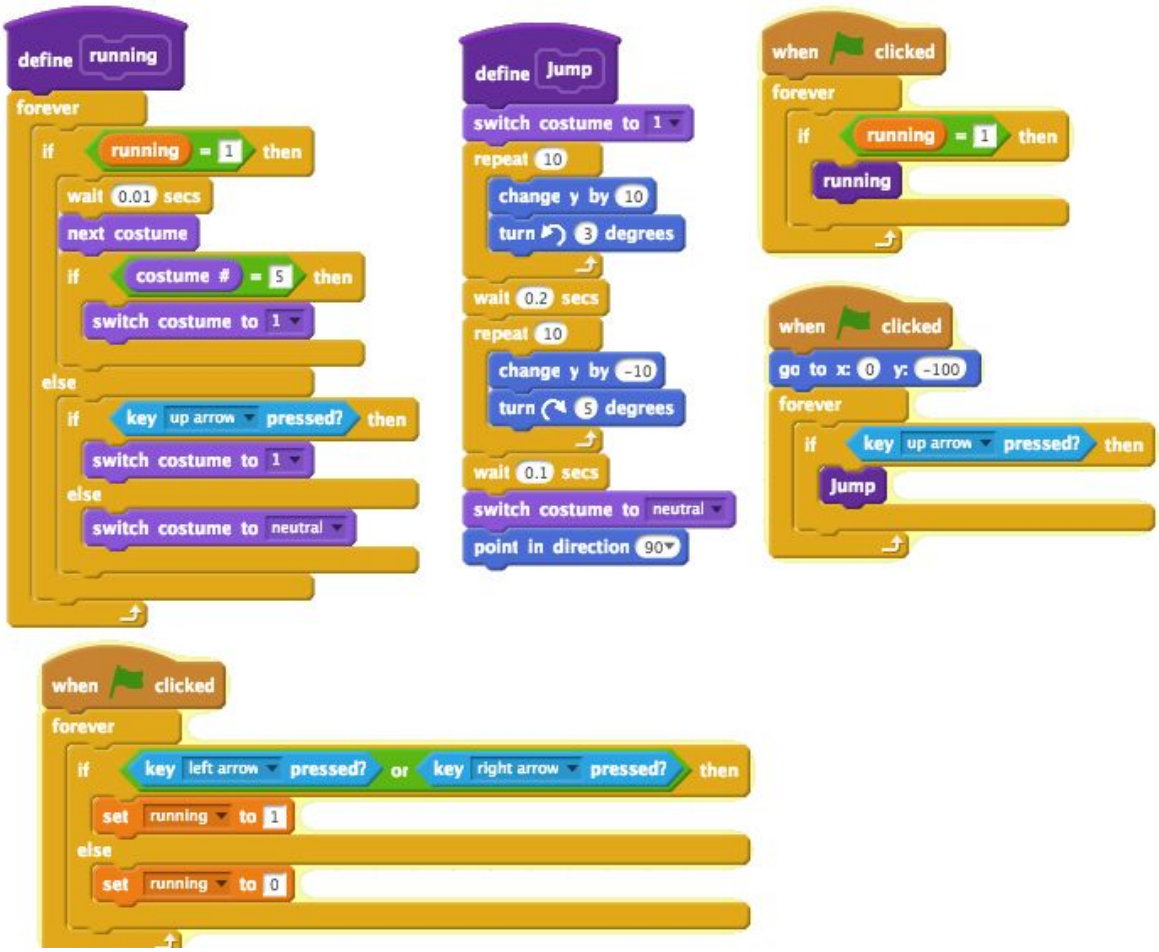


So whats going on here?

1. I included that pesky forever look to make the the code is always listening for the up arrow.
2. I've added costume changes, to make sure our squirrel is moving forward when jumping. and changes back to a neutral pose at the end of the jump.
3. I have added *directions.* We want our characters to lean with the jump, leaning back to jump up, for forward to jump down. So to start we tell the character to lean back, like the Y value, we make a small change because it will be repeated 10 times.
4. Coming down, we need to move the opposite way, twice as much. So when my character leaned back 3 degrees, they will lean forward, 6 degrees.
5. At the end, I made sure to switch back to my neutral costume, and make sure the character is pointing straight forward again.

If I am going to be changing directions, I will need to switch my rotation style to left-right. And the angles will not work. If this is the case you can edit your sprites costumes and create a special jump cycle where the costume is pointed the right way.

So now to tie all the code together



What are the changes?

1. I moved all of the code the the jump into a custom block
2. I made a new “when clicked” block to call the jump function. It resets the position of the character at the start. Then, in a forever loop, it listens for an up arrow, when it hears one, it runs the jump function.
3. I put a new “if else” in the running function. the costume was getting stuck on neutral when the character was not running, so I had to check for an up arrow when the character is not running, so that the right costume will be selected.

So how is it all coming together for you? Can you make this code even better?

Your next step would be to make new costumes to animate the way your character jumps.
and then we can get the background, and character moving.

Here are some examples of jump animations that you might base the new costumes on.