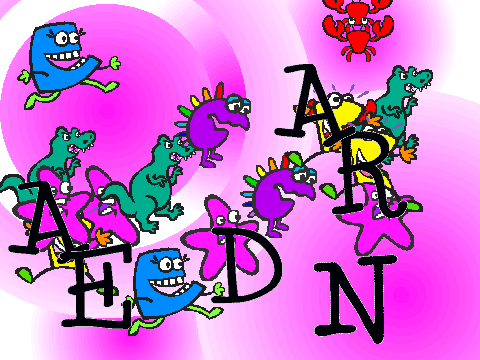
# Interactive Art Lesson 2: Random Gliding

In this lesson, we will make the letters of our name move around the Stage leaving behind a trail of Stamps. This is another example of an Interactive Art project, where the picture that is made is different every time the user runs your program.



You will learn about the (x, y) coordinate system, the forever loop, how to generate random numbers., and how to run initialization code when the Green Flag is clicked.

This lesson has five simple steps:

1. Change the background, or stage
2. When a letter is clicked, forever move to a new random location
3. Leave behind stamps as the letters move
4. Copy your scripts to all Letters Sprites in your name.
5. Set initial position of each letter

If you liked the word and costumes you animated in Lesson 1, you might want to start with that project and modify the scripts you wrote. That way, you won’t have to Import all of the Letters and Costumes again. If you choose this approach, just delete all of the old scripts (by dragging each script off to the left). Then, right away **Save As** your project to a new name (Lesson 2), so you don’t lose your first project!

## Change the background, or stage

In our previous lesson, we just used the white plain background that Scratch starts up with. It is very easy to change the background in Scratch. In Scratch, the background is actually called the Stage. It works a lot like a sprite because it can have costumes, scripts, and sounds too. To edit the stage, you need to first select it.

Click the **Backgrounds** tab (where the Costume Tab is normally) to see the backgrounds. You can draw your own background or import one that is already made.

We are going to start with an educational stage: one that shows the (x,y) coordinate system. Click the **Import** Button.

A window will pop up that will allow you to choose a background. Choose the **xy-grid**. You should see the new background in your world!

Start with just a single letter of your name. (We will often find that it is useful to write Scripts for just a single Sprite, experiment with the Scripts, and get them just right, before you copy those Scripts to other Sprites.)

Every Sprite has a position on the Stage. The vertical position (up and down) of each sprite is called **y**; the horizontal position (left and right) is called **x**. The current y position of a sprite is stored in a **variable** called **y position**; the current x position of a sprite is stored in a variable called **x position.**

You can find these variables under the **Motion** menu.



If you click on their boxes, their values will be shown in the World window.



Move your Letter Sprite around the Stage and watch the x position and y position variables change…

## Make letter move when clicked

Let us make our letter glide to a new position on the stage. We can do this with the **glide** command which is under the **Motion** menu. Glide takes 3 numbers as inputs (or parameters or arguments): the amount of time it should take to move the Sprite to the new position, the new x position, and the new y position.



To move the letter to the middle of the Stage, what x and y value should you specify?

To move the letter to the top of the Stage, what x and y values should you use? To the bottom? To the far left? To the far right? To the upper left corner?

Let us make our letter move to the upper left when the user clicks on it. How would you do that? (-240, 180)

It is a little boring having the letter always move to the same spot on the Stage every time you click on it. To make things much more interesting, we can have the letter move to a new random position each time. Use the **glide to random position** block for this. You may also want to slow down (or speed up!) the gliding by picking a different number of seconds.

Now our letter moves to a random spot on the stage each time we click on it!

Let us say we want the letter to keep moving after we just press it once? How can we do that? We can have the letter glide forever by using the **forever** loop under the **Control Menu.**

## Leave behind stamps

To make the behavior of your moving letter look more interesting, you can have it leave behind a trail of stamps as it moves. The **Stamp** block under **Pen** stamps the current costume of the Sprite wherever that Sprite is; when you move the Sprite, you can then see the Stamp that it left behind.

Where should you put the **Stamp** block in your scripts? Let’s leave behind a **stamp** after each **glide** in the **forever loop.**

**Optional:** If you want to make your art look a little more interesting, you can make the letter Sprite switch to a different costume before you leave a stamp of it behind. You will then want to switch back to the plain Letter costume at the end of the **forever** loop, before your Letter starts gliding again. We recommend using the blocks **switch to costume <name>** instead of **next costume** here so that you can keep track better of which costume is showing.

Or, maybe you want to switch the color of your letter each time through the loop. You can do this easily with the **Change color effect by 25** block under **Looks.**

## Copy to all letters of your name

We want all of the letters of your name to act like this. So, **copy** this script to all of the letter Sprites. After you have each of your letter sprites imported, you can just drag and drop to copy script blocks across them.

## Set initial position of letters

One problem you may have noticed is that you can’t get the letters of your name back in the right order very easily! It would be a pain if you had to use the mouse to drag every letter back into its right place when you start. A better way is for Scratch to put the letters in the right order for you. You will need to tell Scratch how to do this.

The Green Flag is used whenever the user wants to start up your game or art project again. When the user presses the Green Flag, you want your letters to go to their right places. You can use the block **When Green Flag Clicked** under the **Events Menu** to start off a new set of blocks.

The block we want to execute when the user clicks the green flag is to **go to**  a certain (x, y) position. The go to block is a lot like the **glide** block, but it makes the sprite move there instantaneously instead of gradually. How can you figure out what (x, y) position you should specify as arguments?

One way is to fill in those numbers by hand using your knowledge of the x,y coordinate system. This is fine, but sometimes it can be a little difficult figuring out the exact right values.

An easier way is to let Scratch tell you. Try this out. You could try dragging a Sprite to a new position to see its x and y values. The problem is, when you try to drag the Sprite, the **When this Sprite clicked** script will activate! You also may find it difficult to figure out which letters are the actual Sprite and which are just stamps! So, we recommend splitting the **When this Sprite Clicked** block from the stack to de-activate it. And, **erase all** the stamps.

Now, move the Sprites to your desired initial positions. Scratch will update the parameters of the **go to** (and **glide)** blocks to the current (x, y) position of the Sprite!

One small final problem is that the stamps stick around when the user starts the Art project again. To clear the stamps, use the **erase all**  block, which is under the **Pen** Menu. The clear can put under any **When Green Flag Clicked** control block. (Optional: I like making this a script that is associated with the Stage since it controls something that happens to all of the Sprites.)

You should be all done now!

# Things to Remember

In Scratch, the background is called the Stage. The stage is a lot like a sprite because it can have costumes and scripts too. To edit the stage, select it and click the Background tab.

There are many ways to move sprites. Gliding is one way to make movement very smooth and natural. The **go to** block moves a Sprite to a new position instantaneously.

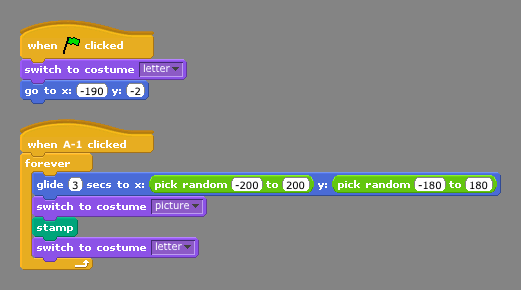
Variables are extremely useful. Variables hold information about what has happened. Two examples of variables are the x and the y position of a sprite; the x position shows where a sprite is left to right while the y position shows where it is up and down. Every sprite has its own x and y variable.

The **forever** control structure executes the blocks inside of the loop forever.

The **When Green Flag Clicked** block will be activated whenever the user clicks on the green flag to start up a new game. You should put a script after this block that initializes your game to its starting state or condition.

A useful trick to know is that Scratch fills in the values of the **go to** and **glide** blocks with the current (x, y) position of the Sprite when you click on the Movement Menu.

# Final Scripts



# Week 2 Homework

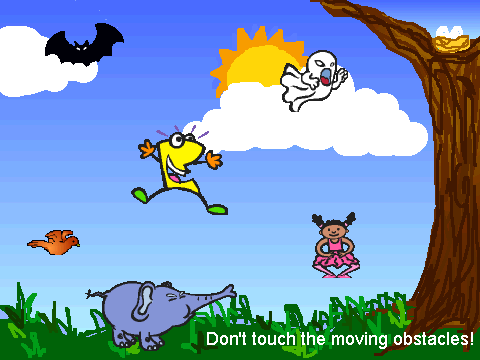
This week, you have two straight-forward assignments. You can do them in any order. You should save both projects onto your USB flash drive so that you can bring it to the Shorewood computers every week.

**Interactive Art Lesson 2**

One assignment is to complete Interactive Art Lesson 2. In this project, you make the letters of your name (or any word you choose) move around the stage leaving behind stamps. You can use any Stage (or background) that you like. You can either draw your own background or Import one included with Scratch (unfortunately, the Scratch installation on the XOs does not have any backgrounds other than the xy-grid, so you will need to paint your own).

You can combine this lesson with the previous one if you want: that is, while the letters of your name are moving around the stage they can also be changing costumes!

**Direct the Bird to its Nest**



On the USB flash drive, we have copied a Scratch program for you to experiment with.

Your task is to write code to direct the bird so it flies to its nest while avoiding all of the objects in between. Your script should belong to the bird Sprite and be activated when the user clicks the Green Flag. You can use the **glide** block or any other movement blocks you choose (for example, **move** or **turn).** You may find it helpful to press the space key while your program is running so you can see the xy grid on the Stage.

If you write code to move the bird to the nest, you will see the message **You Win!** If you write code that makes the bird touch one of the other Sprites, you will see the message **You Lose!**

**XO Help: Accessing USB flash drive**

To have Scratch open a Scratch file from a USB flash device, click on the **Open** button at the top of Scratch. In the diaglogue box, click **Computer** and then the "**media**" folder; in there, you should see the name of your USB device and you can find the scratch files in there. Follow similar steps to **Save As** a new Scratch program to the USB device.

To properly unmount the USB device, you need to use the Journal

activity; at the bottom of the Journal, you'll see an icon for the USB

device; if you hover the mouse over it, you'll see the option for

**unmount**.