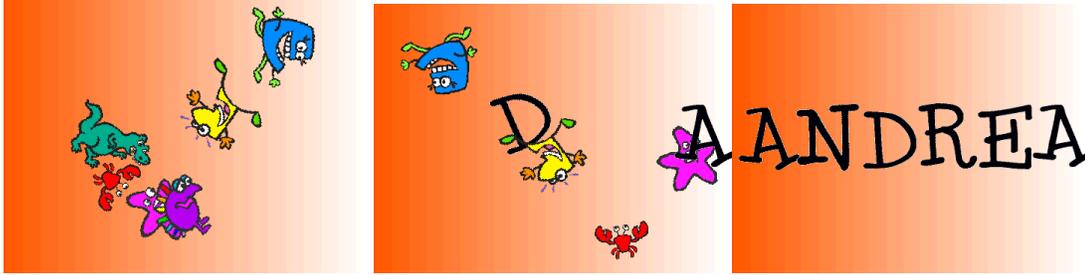


## Catch that Letter!

In this lesson,



You will learn about the conditions and the **repeat until** control structure in this lesson.

This lesson has four steps:

1. Make a sprite move around the stage
2. When the user clicks the sprite, switch the sprite to show a letter and glide to the correct location.
3. Copy your scripts to all the Letters Sprites in your name.
4. Play music.

### 1) *Make a Sprite move*

- Sprites to move around the stage.
- We are going to learn about some different **Motion** commands.
- First block is **move <num> steps**. Move takes one parameter:
  - What happens if you specify a negative number of steps?
- Second block is **turn <num> degrees**. What number makes the Sprite turn half the way around the circle? What number makes it turn all the way around? What do negative numbers do?
- Keep running the code; what happens? The Sprite moves off the Stage so that you can't see it! We want to keep the Sprite on the Stage. Scratch contains a command to make sure this happens: **if on edge, bounce**.
- Use a **forever loop** around these blocks. Under the **Control** commands, add the **When the green flag is clicked** block to the top of these scripts.
- What happens?

### 2) *Switch costumes when user catches Sprite*

The user is trying to catch your Sprite and click on it. When the user does this, you want the Sprite to **switch** to the Letter costume and **glide** to its correct position in your name.

- **When Sprite1 Clicked** start block. What happens if we put the **switch to costume** and **glide** commands in a new script that starts when the Sprite is Clicked?
- Place the **glide** and **switch to costume** blocks after the **forever** loop. Scratch won't even let us do that because the forever loop never ends!
- Change the **forever** block to a loop that will end when the user clicks on this Sprite.

- Use the **repeat until** block under the **Control Menu**. This structure repeats the blocks until the condition specified at the top is true. How can we find out when the user has clicked on the Sprite?
- Use the **Sensing** blocks. Let us try putting the block **touching mouse-pointer** as the condition (layer the scripts). Notice that this block has pointy edges just like the space in the **repeat until** block. The Sprite moves until the user puts their mouse pointer over it and then glides to the right place!
- We want the user to have to **Click** on the Sprite too. Let's try placing the **Sensing** block **mouse down** in the **repeat until** condition instead. Now what happens? The loop repeats until the user clicks the mouse, but the mouse might not even be on the Sprite when they do that! This isn't what we want either.
- We want the loop to repeat until the user has clicked the mouse **and** the mouse is touching the Sprite. We can test for both of those conditions using the **AND** block under the **Operators Menu**
  - Remember....you can layer the green **and** script and then add the two sensing scripts).
- When the user clicks on the Green Flag, what costume should we show? Add the code to switch to the correct costume initially.
- 
- you may have noticed that the letter is faced in an awkward direction when it goes to its position in your name. To point the letter the right direction use the block **point in direction 90 (right)**.

### 3) **Copy to all letters of your name**

We want all of the letters of your name to act like this. So, again, copy this script to all of the letter Sprites like you do for the previous lesson.

- To get this to work right, you do need to get one detail correct. You need all of the **costume names** to be identical across Sprites if you want the same code to work across all Sprites. When your code says **switch to costume picture** and **switch to costume letter** there will actually be a costume with that name to switch to. If you don't have the right costume name, your code won't work correctly: it won't switch to the right costume.

**Remember:** that you will need to put the correct **x** and **y** values in the **glide** block for every Sprite.

You might want to change the amount that different Sprites move or turn by to make them all behave a little differently and make the game harder.

### 4) **Play music.**

You can liven up your game by having it play music in the background. Which Sprite should be the one playing the music? Since the music is associated with the whole game and not a specific Letter, it makes sense to put the scripts playing the music with the **Stage**.

- Click on the **Sounds** Tab next to Scripts and Backgrounds. Then click on **Import**. You will have many options.

Let us activate that block when the **Green Flag is clicked**. What is the problem? We want the music to play **forever**.

## Things to Remember

There are many ways to move sprites. Gliding is one way to make movement very smooth and natural. Another option is the **move** block, which moves the Sprite in the direction it is currently facing. The **if on edge, bounce** block is very useful for detecting if a Sprite is on the edge of the Stage and then keeping the Sprite on the Stage.

The **repeat until** control structure executes the blocks inside of the loop until a condition is true. The condition we tested in today's game was if the user's mouse was touching the Sprite **and** the mouse was down (i.e., clicked). There are many interesting conditions under the **Sensing Menu**.

You can change the name of Costumes and Sprites. It is a good idea to pick a name that describes the Costume or Sprite in a meaningful way. If you copy Scripts that involve costume names across different Sprites, make sure that each Sprite uses the same costume names.

You can associate Scripts with the Stage. A logical place to put a Script that plays background music is with the Stage Scripts.