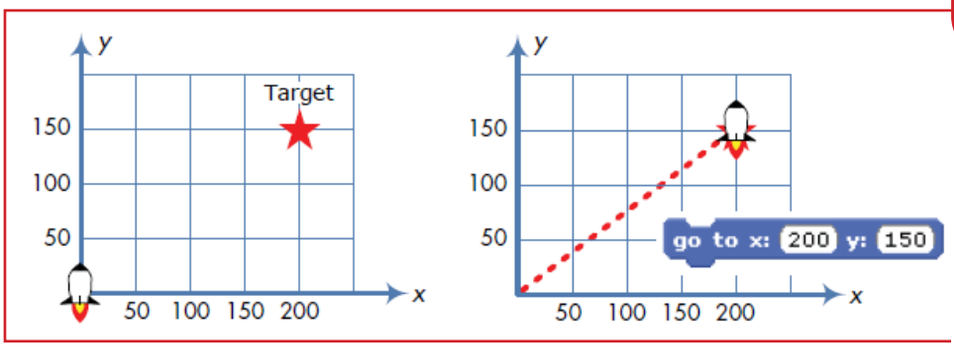
Motion and Drawing - session 2

In this module, you'll:
- Explore Scratch's motion and pen commands
- Animate sprites and move them around the Stage
- Draw artistic, geometric patterns and create games
- Learn sprite cloning

Using Motion Commands

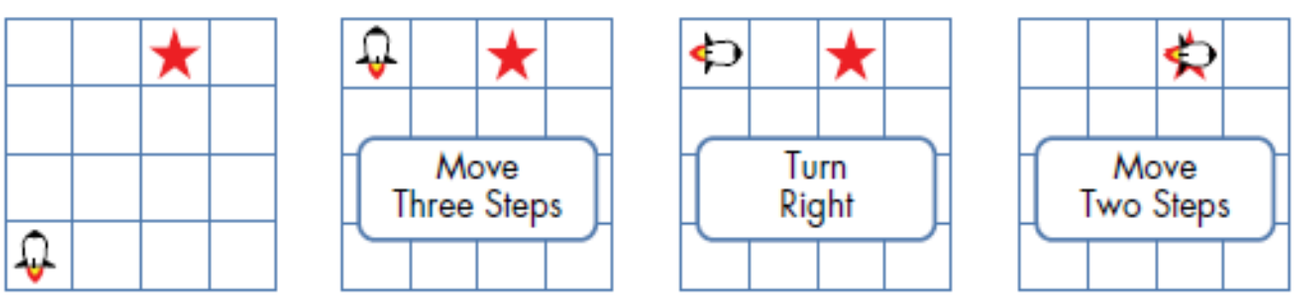
If you want to make games or other animated programs, you'll need to use blocks from the Motion palette to move sprites around.



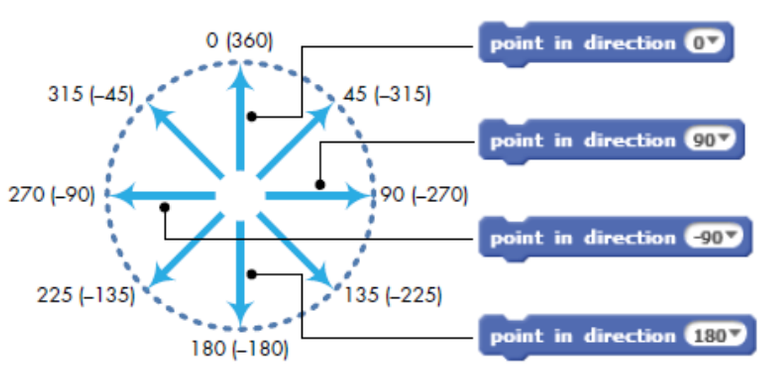
Absolute path

Let's say that you want to make the Rocket sprite hit the star-shaped Target sprite at position (200,150). The most obvious way to do this is to use the **go to** block. The x-coordinate tells the sprite how far to move horizontally across the Stage, whereas the y-coordinate tells it how far to move vertically. You can make the Rocket slow down by using the **glide** command instead.

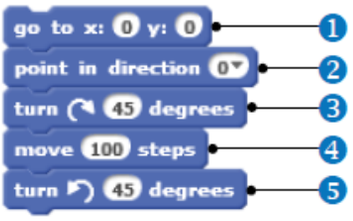
Relative Motion

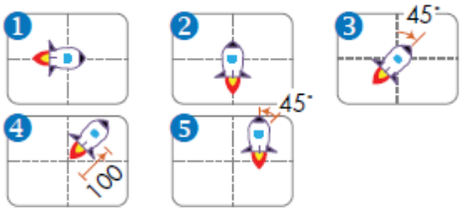


Now consider the grid depicted above which shows another Rocket sprite and target. You can't see the coordinates this time, so you don't know the sprites' exact position. If you had to tell the Rocket how to hit the target, you might say: "**Move three steps** then **turn right,** then **move two steps**." You can turn a sprite toward a particular direction (or heading) with the **point in direction** command



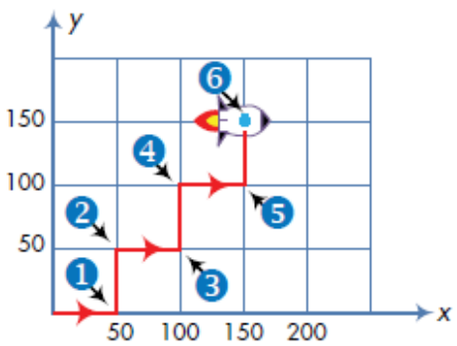
In Scratch, 0 is up, 90 is right, 180 is down, and –90 is left.



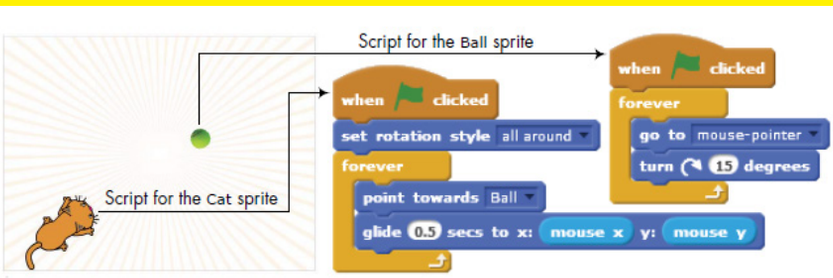


TRY IT OUT

Sample scripts that illustrate using variety of motion commands



Sometimes you might only want to move your sprite horizontal-lyn or vertically from its current position, and that's where the **change x by** and **change y by** blocks come in.



Programming a cat to run after a tennis ball using motion commands

Pen Commands and Easy Draw

The motion commands you used in the previous section allow you to move the sprite to any point on the Stage. Now wouldn't it be nice to see the actual path your sprite travels? Scratch's pen can help. Each sprite has an invisible pen, which can be either up or down. If the pen is down, the sprite will draw as it moves. Otherwise, the sprite moves without leaving any trace. The commands in the **Pen** palette allow you to control the pen's size, color, and shade.

TRY IT OUT

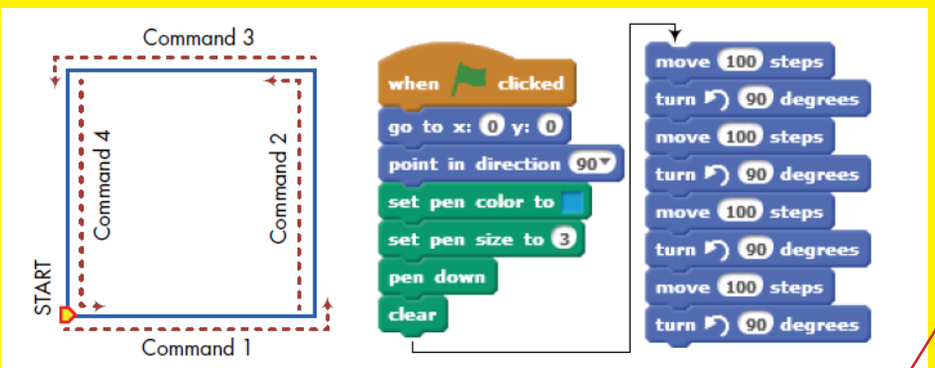
Open the Tips window in Scratch, click the house icon, and click Pen for a brief description of each Pen command. The scripts below demonstrate most of those commands. Re-create these scripts, run them, and describe the output of each. Don't forget to set the sprite's pen down before running these scripts. (You can find the repeat block in the Control palette.)

The Power of Repeat

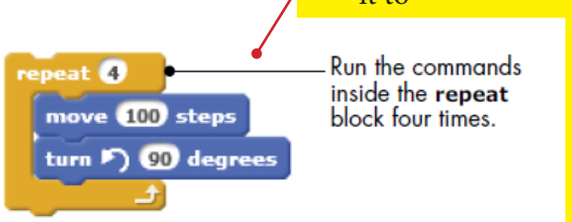
Our programs have been relatively simple so far, but as you start writing longer scripts, you'll often need to replicate the same stack of blocks several times in a row. Duplicating scripts can make your program longer, harder to understand, and tougher to experiment with. If you need to change one number, for example, you'll have to make the same change in each copy of the block. The repeat command from the Control palette can help you avoid this problem.

TRY IT OUT

Let's say that you want to draw the square shown on the left below. You could command the sprite to follow these repetitive instructions:

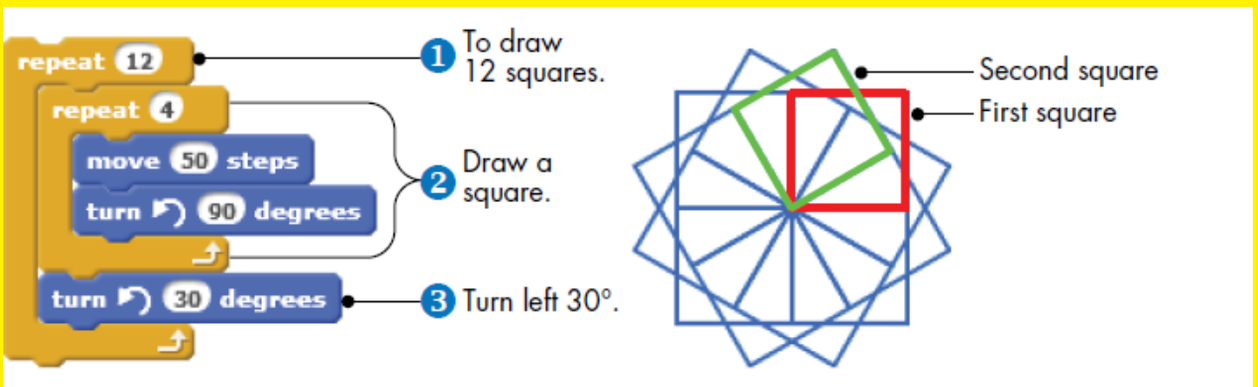


In contrast, we can avoid using the same two blocks over and over with the repeat block, which runs the commands inside it as many times as you tell it to

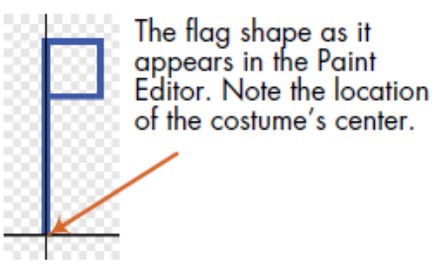


Rotated Squares

You can create amazing art by repeating a pattern in a certain sequence.



The stamp command allows you to create complex geometric patterns with ease.

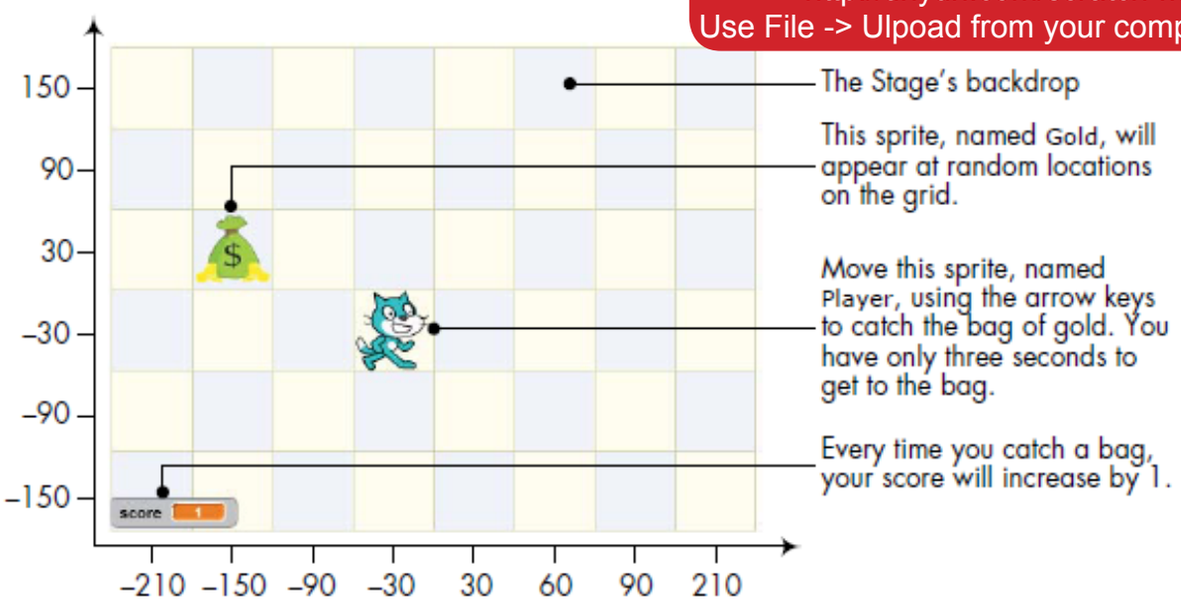


The flag shape as it appears in the Paint Editor. Note the location of the costume's center.

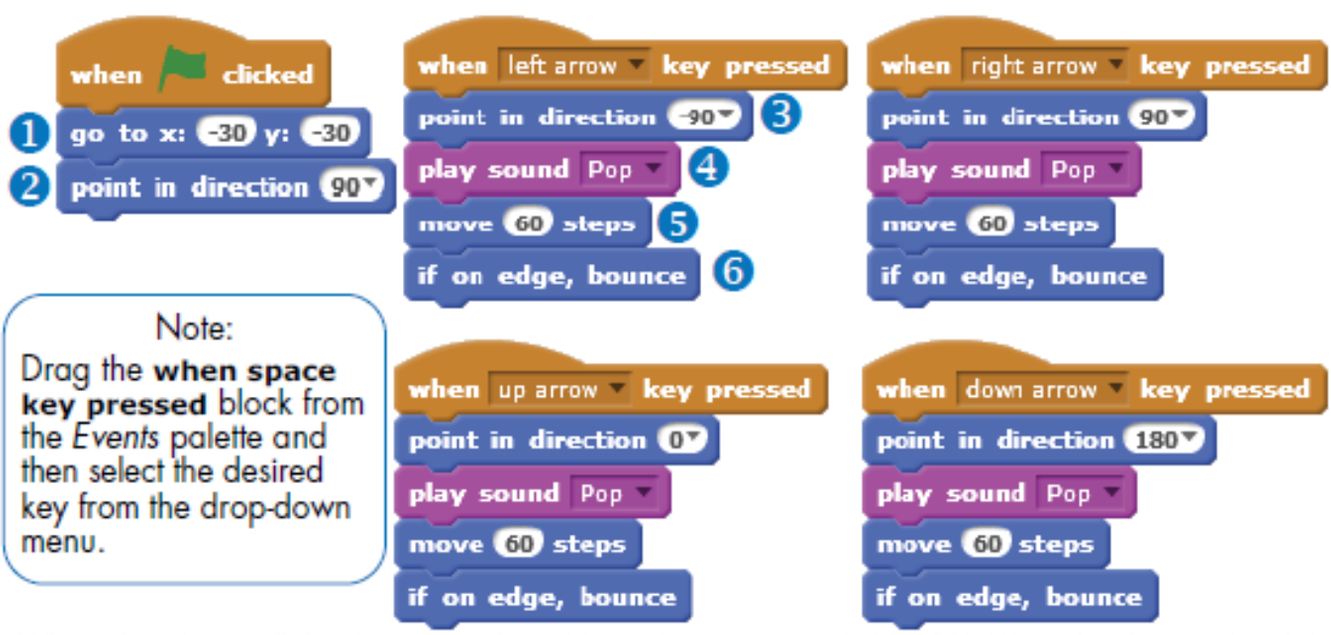
Scratch Project - GET THE MONEY

Our first application is a simple game in which the player needs to move the sprite by using the keyboard arrows to collect as many bags of gold as possible. As illustrated below, the bag of gold appears at a random location on the grid. If the player doesn't grab the bag in three seconds, it moves somewhere else.

OPEN PROJECT **MONEY** FROM MODULE 2
found under following link:
http://tinyurl.com/scratch-modules
Use File -> Ulpoad from your computer to open it



PLAYER SPRITE:

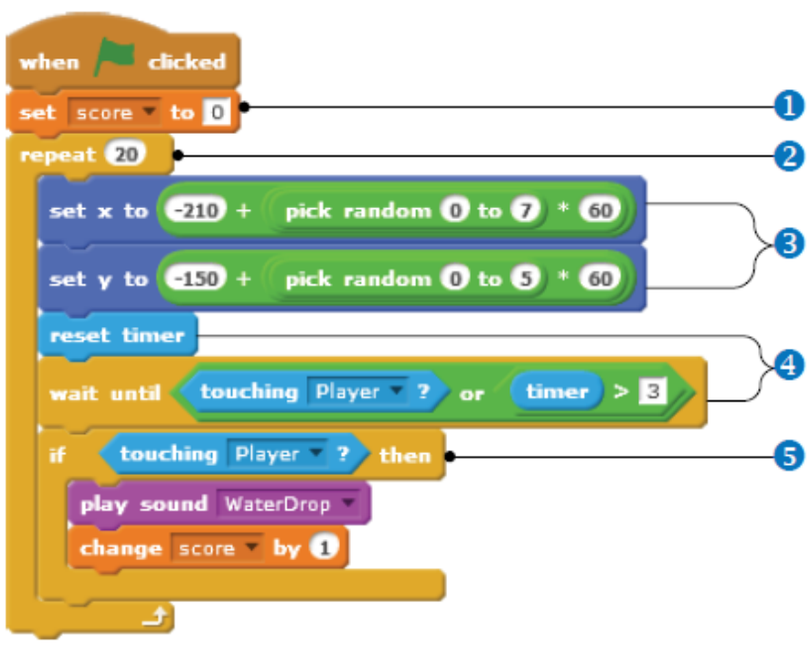


Note:
Drag the **when space key pressed** block from the *Events* palette and then select the desired key from the drop-down menu.

When the player clicks the green flag, this sprite moves to (–30,–30) (1) and points to the right (2). The other four scripts respond to the arrow keys. When an arrow key is pressed, the corresponding script changes the sprite's direction w, plays a short sound (using the play sound block (4) from the Sound palette), and moves the sprite 60 steps (5). The sprite bounces off the Stage's edge (6) if needed. Because 60 steps correspond to 1 square on the stage's backdrop grid, each time you press an arrow key, the Player sprite moves 1 square

GOLD SPRITE:

Since the game just started and we don't have any bags yet, we set score to 0 (1). Next, we start a loop that will repeat 20 times (2) to show a total of 20 bags to the player. (If you don't want 20 bags, feel free to use your favorite number instead.) Each time the loop runs, the bag of gold will appear at some random location (3), give the player some time to grab it (4), and increment score if the player is successful (5). We set the bag's x-position by generating a random number between 0 and 7, multiplying it by 60, and adding the result to −210. After appearing at a random location, the bag of gold will give the player three seconds to grab it. (You can change this duration to make the game harder or easier to play.) To track the time, the script first resets Scratch's built-in timer to 0. It then waits until either the player grabs the bag by touching it or the timer exceeds three seconds. When either condition happens, the **wait until** block will let the script move on to execute the **if/then** block. Blocks inside the if/then block will only run if the condition you specify in the header of the if/then block is true

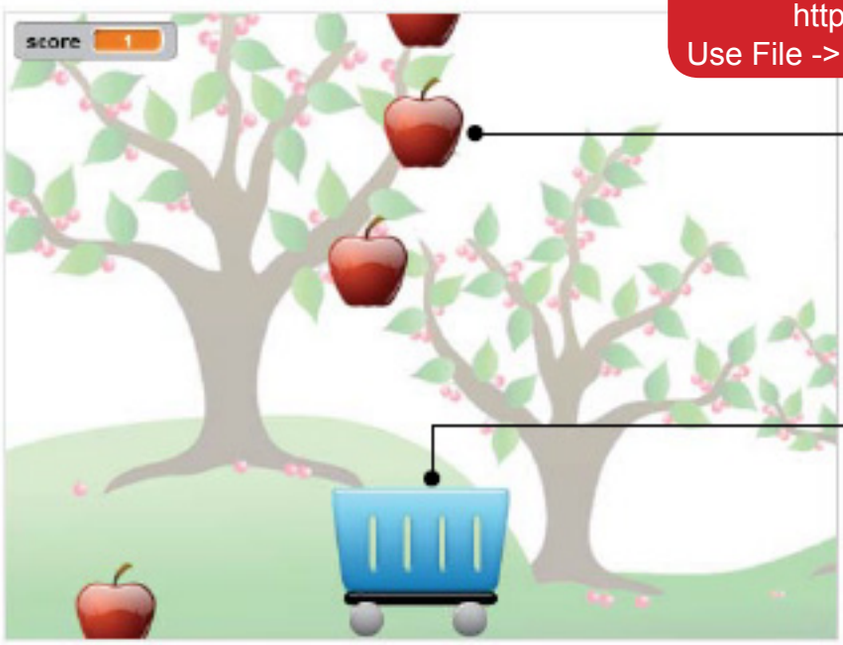


Like the Player script, this script also starts when the green flag is clicked. It moves the bag of gold around. It also tracks how many bags have been collected with a variable named score

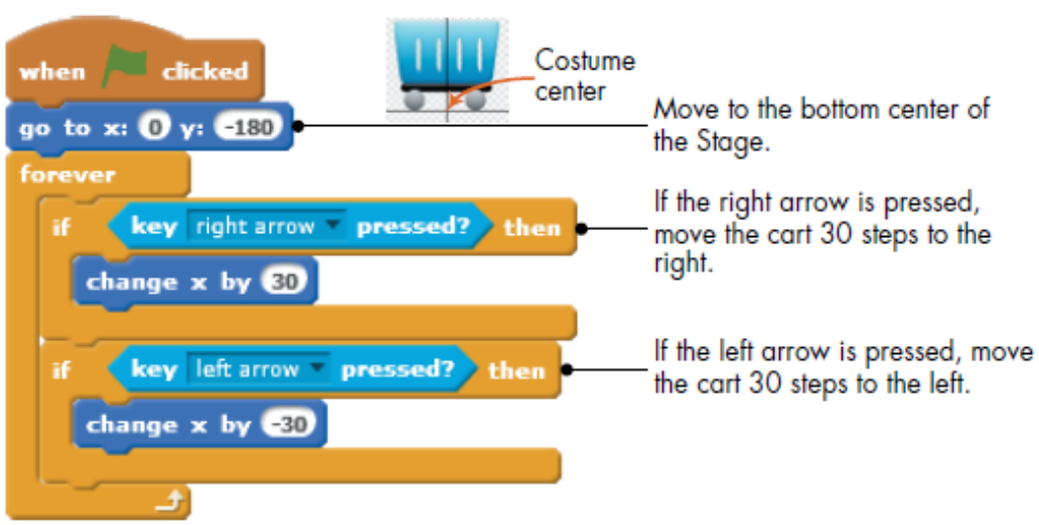
Scratch Project (Introduction to Cloning)- CATHING APPLES

In this game, apples appear at random horizontal positions at the top of the Stage at random times and fall to the ground. The player has to move the cart to catch the apples before they touch the ground, and each apple is worth 1 point.

DOWNLOAD PROJECT **CATCH APPLES**
FROM MODULE 2 found under following link:
http://tinyurl.com/scratch-modules
Use File -> Ulpoad from your computer to open it

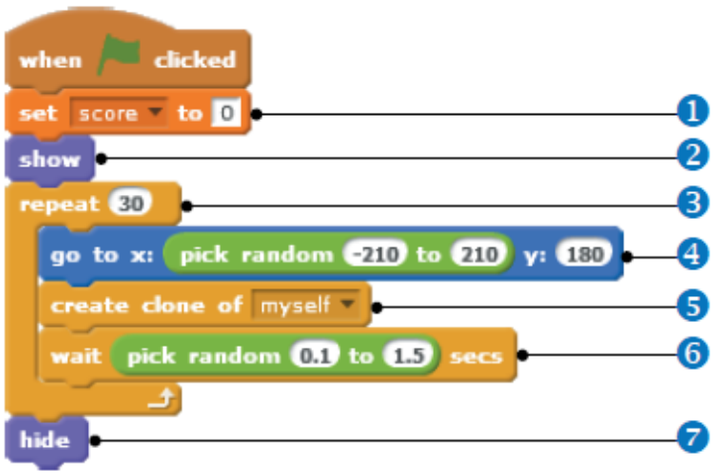


CART SPRITE:



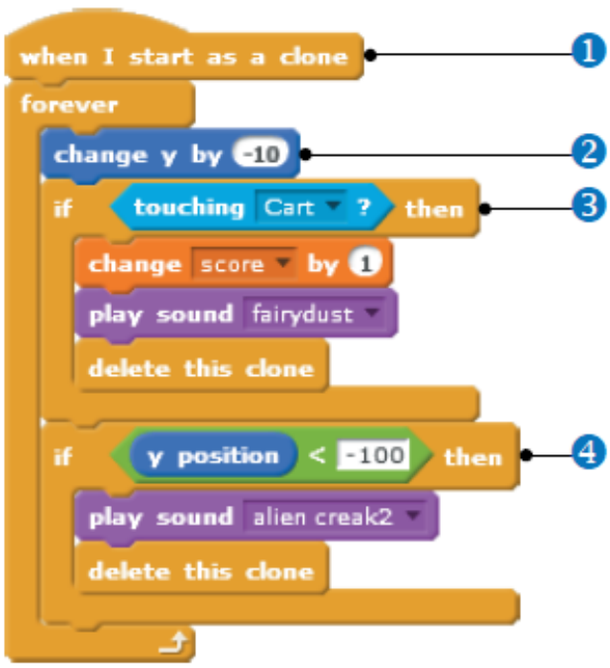
When the green flag is clicked, we move the cart to the bottom center of the Stage. The script then continuously checks the state of the right and left arrows and moves the cart accordingly. I picked the number 30 based on trial and error, so feel free to change it based on your own experimentation.

APPLE SPRITE:



Since we haven't caught any apples yet, the script sets the score variable to 0 (1). Next, it makes the sprite visible with the show block from the Looks palette (2). It then starts a repeat block that will loop for 30 times (3) to have 30 apples fall. During each pass of the loop, the Apple sprite will move to a random horizontal position at the top part of the Stage(4). It then calls the create clone of block (from the Control palette) to clone itself(5), waits for a short random time (6), and starts the next round of the repeat block. After completing the 30 rounds of the repeat block, the script hides the Apple sprite using the hide block (7) from the Looks palette.

If you run the game now by clicking the green flag, 30 apples will pop up randomly at the top of the Stage and stay there—because we haven't told the cloned apples what to do. This is where the next script for the Apple sprite comes in.

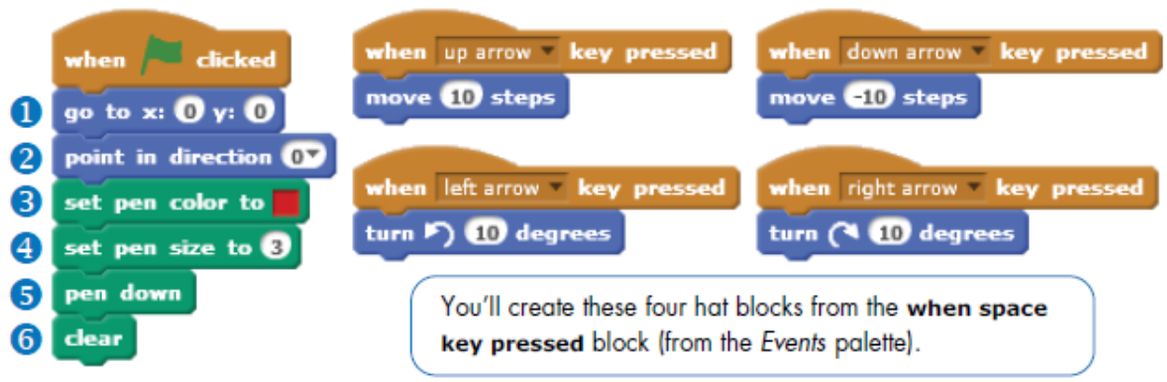


Thanks to the when I start as a clone block (1) (from the Control palette), each clone will execute the script shown in this figure. Each Apple moves down 10 steps (2) and checks whether it was caught or missed by the cart. If the clone detects that it is touching the cart (3), that means it was caught. Therefore, it increases the score, plays a sound, and deletes itself (because it has no more work to do). If the clone falls below the cart (4), then the player missed; in this case, the clone plays a different sound before deleting itself. If the clone is neither caught nor missed, then it's still falling, and the forever block goes around again.

TRY IT OUT

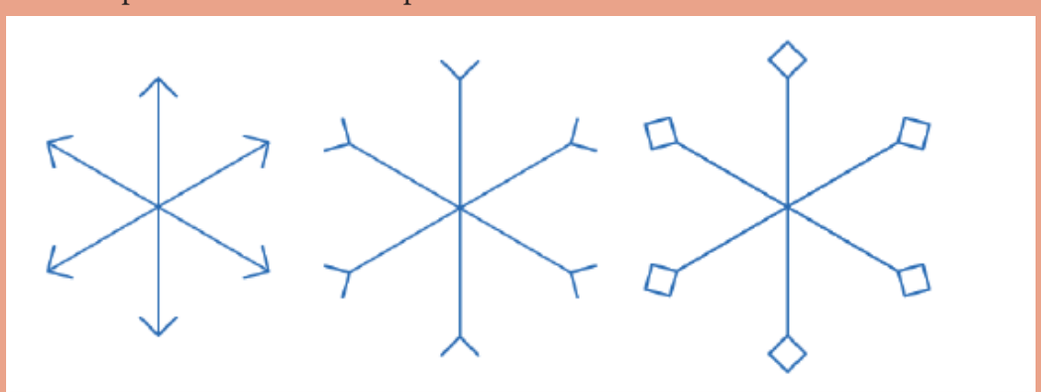
Scripts for the Easy Draw application

Start a new Scratch project. Replace the Cat's costume with something that clearly shows if the sprite is pointing left, right, up, or down. In the Costumes tab, click the Choose costume from library button and select an appropriate costume. Now, add the scripts shown below. You can create the four when key pressed blocks from the when space key pressed block in the Events palette. Just click the down arrow and choose the arrow key you need.



CHALLENGE TASKS

1. Write a script to connect each of the following sets of points in order and reveal the final shape:
   a. (30,20), (80,20), (80,30), (90,30), (90,80), (80,80), (80,90), (30,90), (30,80), (20,80), (20,30), (30,30), (30,20)
   b. (–10,10), (–30,10), (–30,70), (–70,70), (–70,30), (–60,30), (–60,60), (–40,60), (–40,10), (–90,10), (–90,90), (–10,90), (–10,10)
2. Write a script to draw each of the patterns shown below



3. Consider the following script and its output. Re-create the script, add the necessary pen setup commands, run it, and explain how it works.