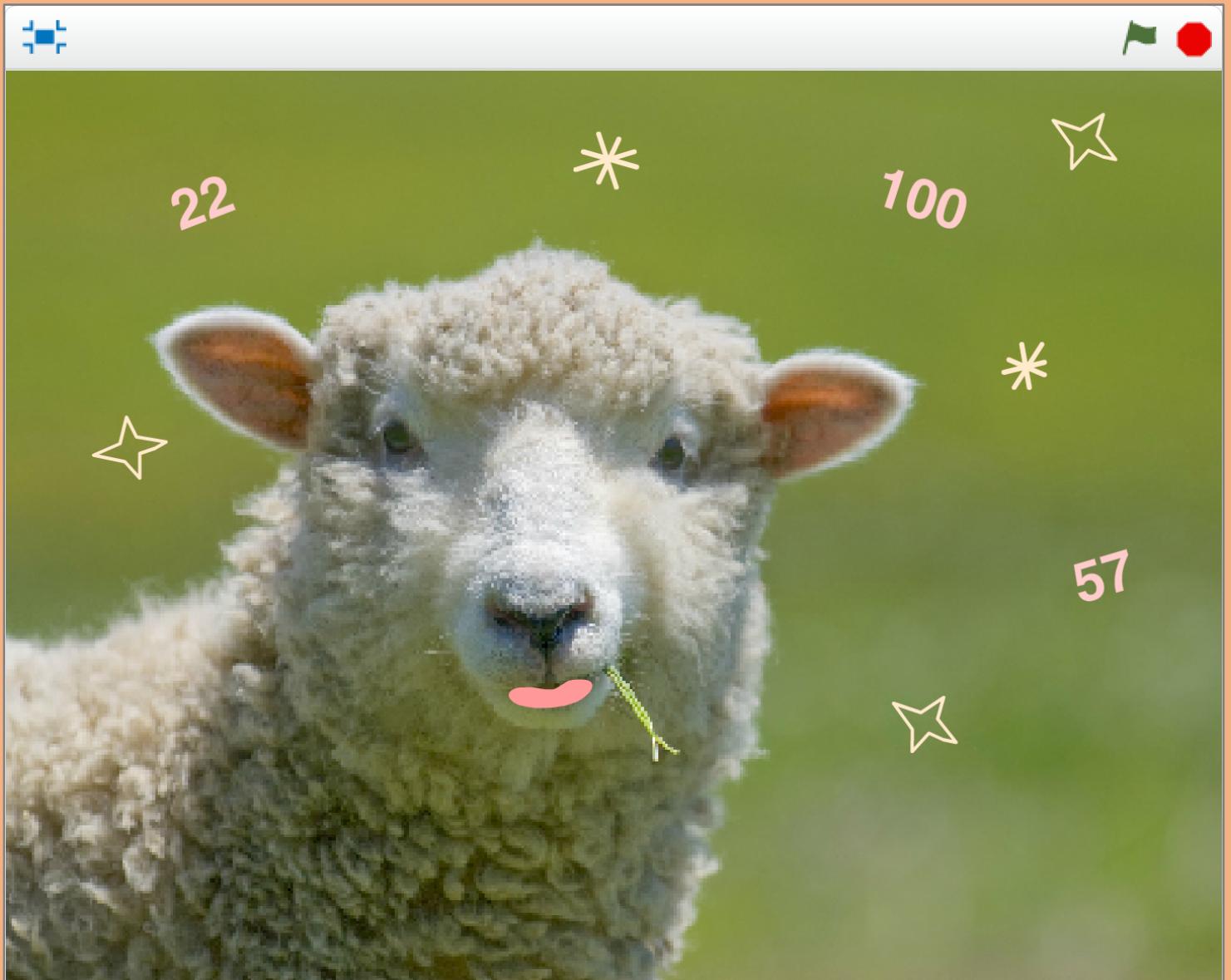# Mystic Sheep

Code It

Mystic Sheep is a number guess game based on a *binary search algorithm* and the *Law of Trichotomy*. Write a program in which the the computer picks a random number, then the player tries to guess it. Inform the player whether he is too low, too high, or correct; prompt him (if incorrect) to guess again.

Create a background by importing any animal image from the web. Also import a .wav sound that the animal makes. Then create a mouth or lips sprite. In Data, make a hidden variable called secret number.

Code the lips: *say* to the player the range of numbers for guessing, for example, 1 to 100. Use *question* to obtain a guess input from the player. In a *forever* loop, make three *if-then* conditionals comparing the guess with the secret number and reporting the result. When the correct number is input, *play* the sound and execute a *stop all.*
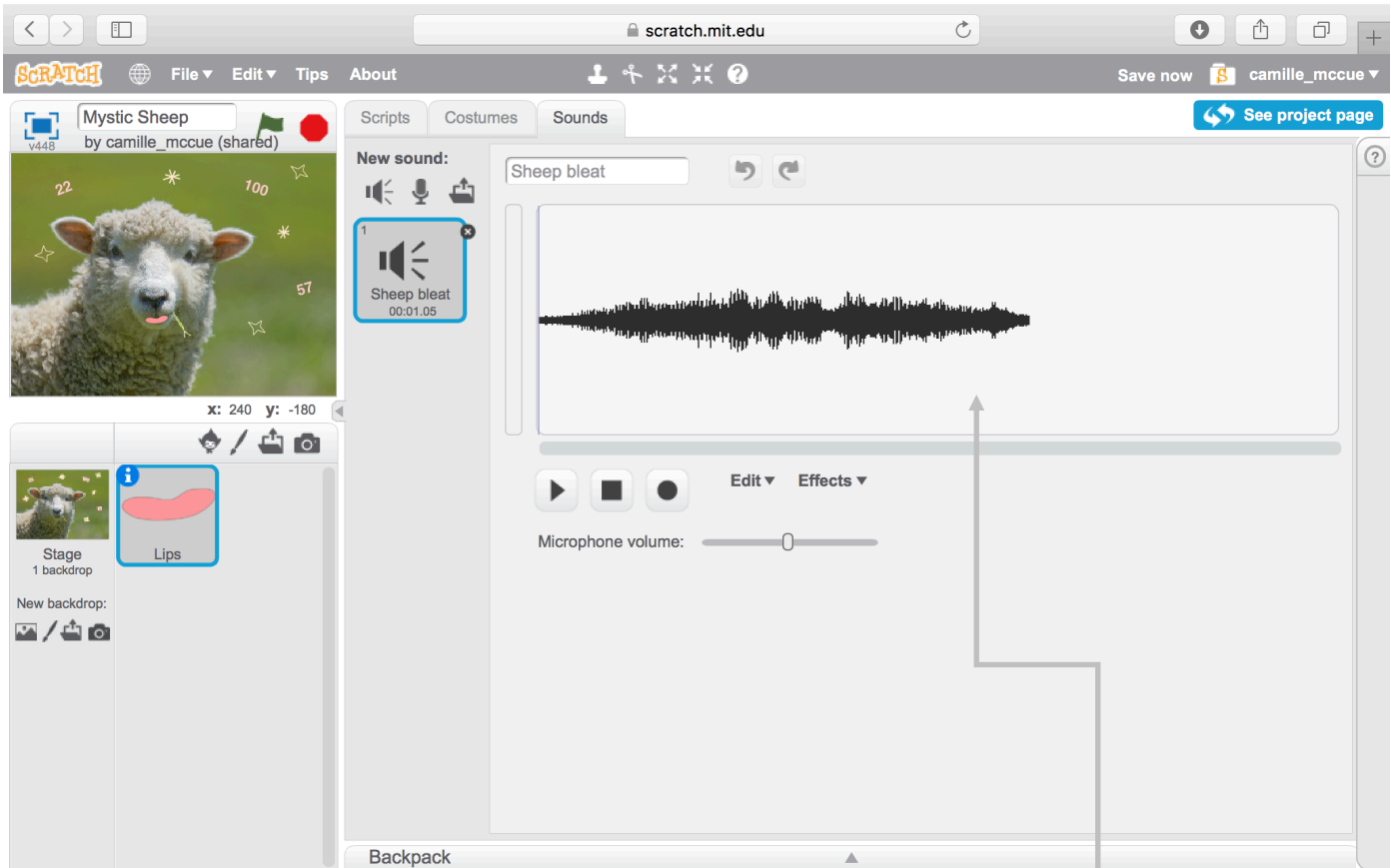
Play It

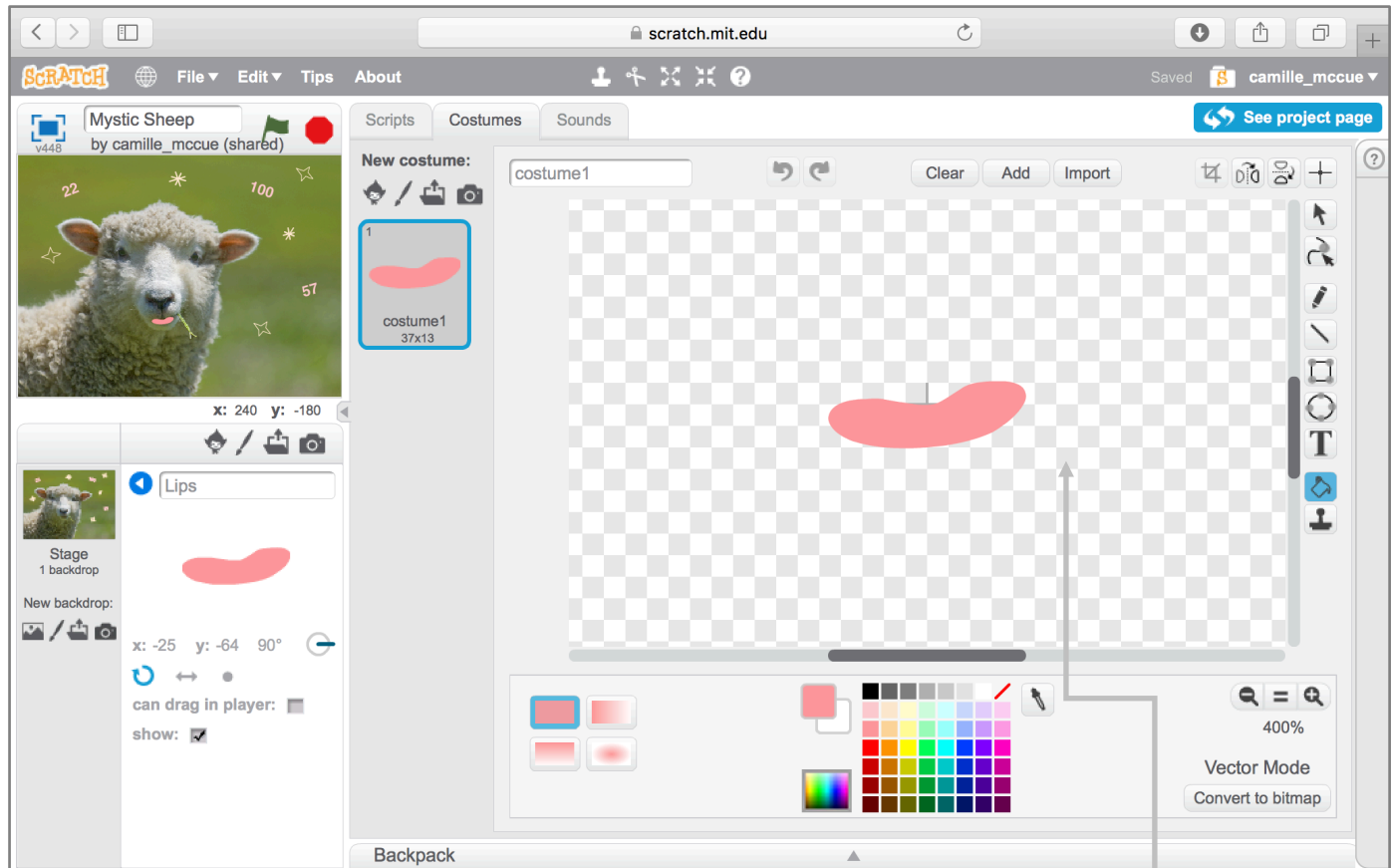Press the animal lips to start the action.

Stage – Background
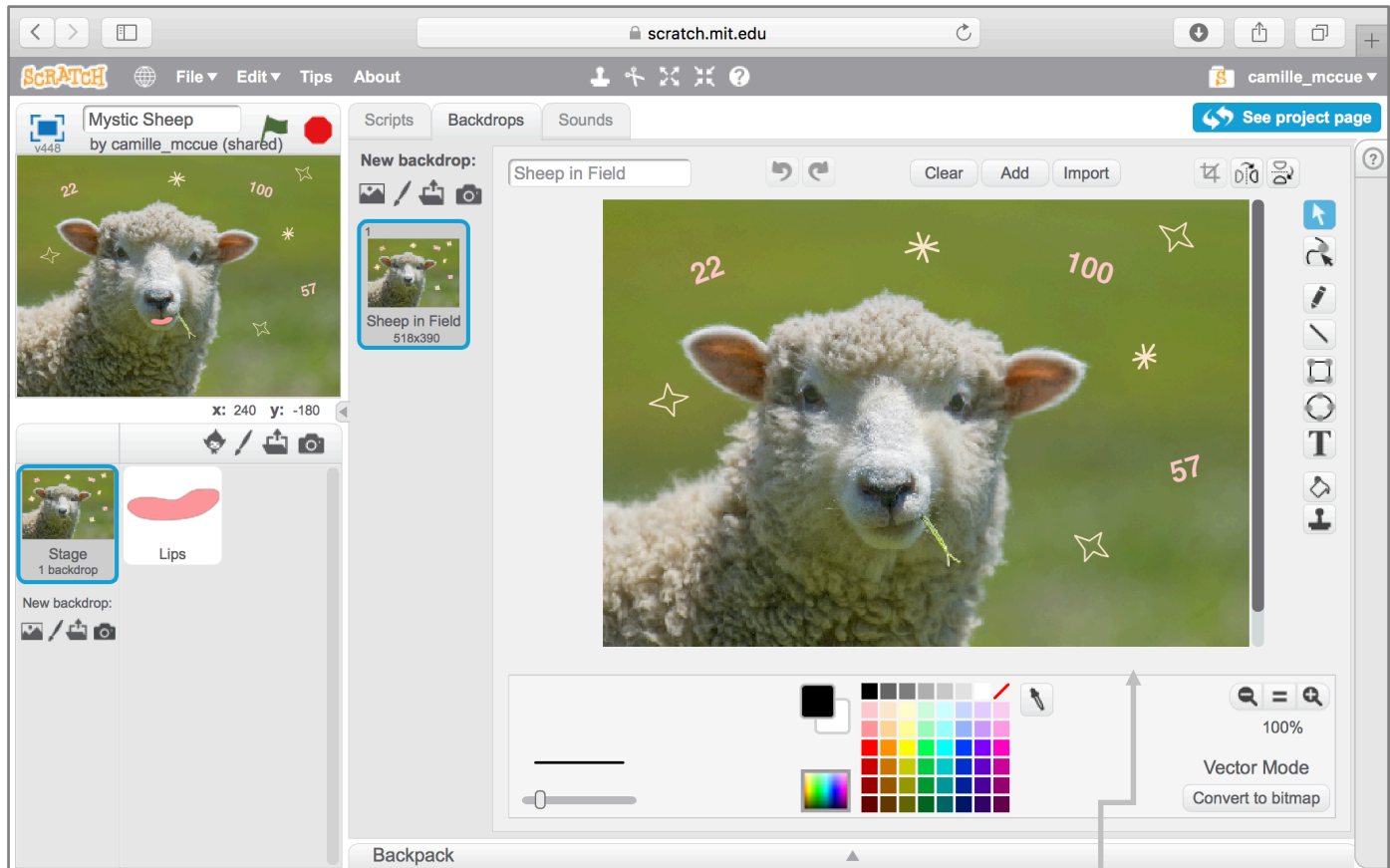
Find and import an image from the web;
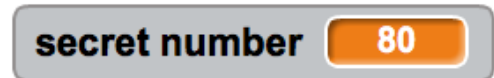If desired, add embellishments
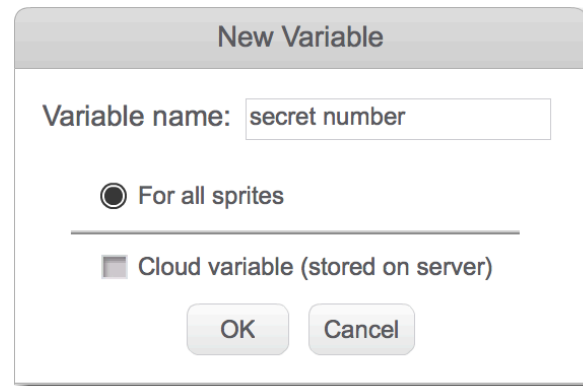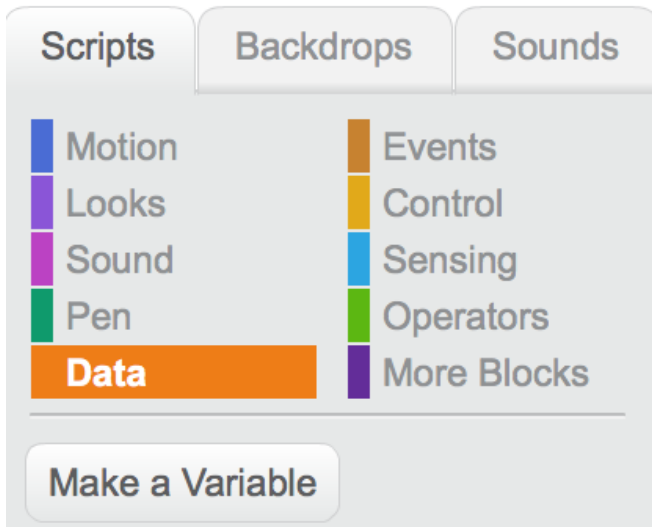
Lips Sprite – Sound

Download an animal sound and then import it; try a site such as www.animal-sounds.org

Lips Sprite – Costume

Use drawing tools to create an animal lips
Costume for a new sprite

Lips Sprite – Variable

Scripts | Backdrops | Sounds

- Motion
- Looks
- Sound
- Pen
- **Data**

- Events
- Control
- Sensing
- Operators
- More Blocks

Make a Variable

New Variable

Variable name: secret number

⦿ For all sprites

☐ Cloud variable (stored on server)

OK   Cancel

secret number `80`

Create and hide the variable, secret number; the computer will generate random values for this variable

Lips Sprite – Costume

Tip: Consider leaving the variable, secret number, visible while testing and debugging your code

Lips Sprite – Script Closeup

```
when this sprite clicked
set  secret number ▼  to  pick random (1) to (100)
say  I'M THINKING OF A NUMBER, 1 TO 100  for (2) secs
forever
    ask  GUESS !  and wait
    if  ( answer < secret number )  then
        say  TOO LOW  for (2) secs

    if  ( answer > secret number )  then
        say  TOO HIGH  for (2) secs

    if  ( answer = secret number )  then
        play sound  Sheep bleat ▼
        say  CORRECT  for (2) secs
        stop  all ▼
```

Extend It

What is the optimal guessing strategy for a player who is playing Mystic Sheep? If a player uses a "divide and conquer" (binary search) strategy as part of his search process, what is the maximum number of guesses he would have to employ in a worst case scenario? How does this number of guesses change if the range of possible values is 1 to 1000?

What other, related games can you create using the Law of Trichotomy? Consider the Price is Right game show in which players use similar guessing strategies (but are eliminated if they "go over," meaning they guess too high).

Try to create a new game using the concepts in Mystic Sheep (binary search algorithm or Law of Trichotomy) in a new format.