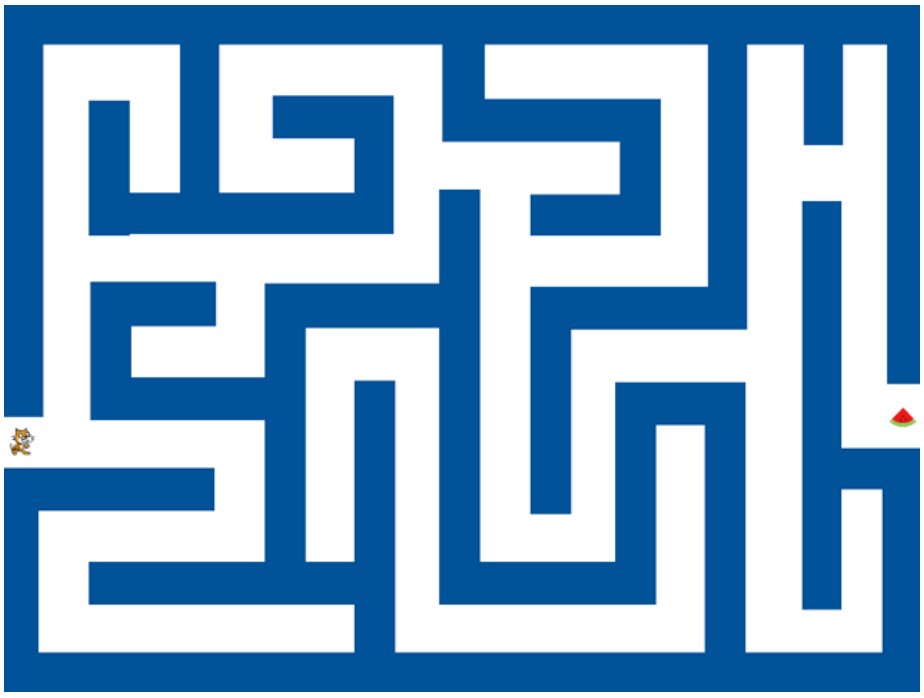
Maze Challenges

Making a maze

1. Use “vector mode” to draw a maze using only the rectangle tool. (Using the rectangle tool is a way to make sure all your lines are straight.)
    a. Use a known color.
    b. Draw a border around the screen.
    c. Draw the maze so that a 15x15 character, moving 5 steps at a time, can fit through all the passages (so this really means to leave a gap of 20 pixels though each passage and entryway).
    d. Do not include any “islands”. (All lines should be connected to the border.)

If you are not feeling creative, here is a maze to copy. Don’t worry if you don’t get it exactly right.



2. Hard question to think about: How could you write a program that draws a random maze? How could you make sure the mazes would be hard, but still solveable?

Using the keyboard to move through the maze

1. Write a "when up key pressed" handler. Move the character five steps at a time. How can you deal with running into a wall? Answer: try moving the five steps, and if you end up touching the wall color, move five steps back.
2. Write "down", "left", and "right" key handlers.
3. Make it so that when the green flag is clicked the character goes to the start of the maze.
4. Make it so that when the character reaches the goal, some kind of win message is displayed. You can do this in a few ways:
    a. Draw special color on the background image at the goal and check if the character is touching that color after each move.
    b. Place a sprite at the goal (like the watermelon in the picture above) and check if the character is touching that sprite after each move.
    c. Check if the character's coordinates are within certain values after each move.

Making the character move randomly through the maze

1. Make a forever loop that repeatedly picks a number from 1 to 4. If the number is 1, move the character up five pixels (but handle walls correctly). If the number is 2, move the character down. 3 means left. 4 means right.
2. Using the method above, the character doesn't get very far, because it only moves 5 pixels before picking another random

direction. Change it so that once the character picks a direction, it goes until it hits a wall, and only then does it pick another direction. It will help to use a variable to keep track of which direction the character is going.

Making the character solve the maze automatically

1. This is much more challenging, but here's a plan: Have the character walk so the wall is always at its feet. You will need to use a few variables to keep track of what's going on.