



Integrating
Computing Across
the Curriculum

Scratch Lesson 3: Game Design with Scratch Lesson Framework



Scratch makes it easy to program your own interactive stories, games, and animations and share your creations on the web. As you create and share projects with Scratch, you learn to think creatively, reason systematically, and work collaboratively. Scratch is a project of the Lifelong Kindergarten group at the MIT Media Lab. It is available for free download from <http://scratch.mit.edu>.

Scratch functions:

- **Control:** When green flag clicked, when key pressed, forever if
- **Motion:** Move x steps, point in direction, go to (x,y)
- **Sensing:** Touching color?

Specific programming concepts¹:

- **Sequence:** To create a program in Scratch, you need to think systematically about the order of steps
- **Iteration (looping):** *forever* and *repeat* can be used for iteration (repeating a series of instructions)
- **Event handling:** *When key pressed* and *when Sprite clicked* are examples of event handling - responding events triggered by the user or another part of the program

Alabama Course of Study Technology Education Standards Third-Fifth Grade:

- **TC 1:** Use input and output devices of technology systems
- **TC 2:** Use various technology applications, including word processing and multimedia software
- **TC 9:** Use technology tools to organize, interpret, and display data
- **TC 10:** Use digital environments to collaborate and communicate
- **TC 11:** Use digital tools to analyze authentic problems (critical thinking, problem solving, and decision making)
- **TC 12:** Create a product using digital tools

Lesson Framework:

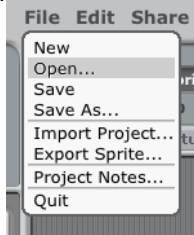
Step 1) Project Research

Have students research facts for their games. For example, students can create a maze game to highlight the life cycle of a butterfly. Or students can create a predator and prey food chain scoring game. Show the students the project PacMan as an example of a maze game.

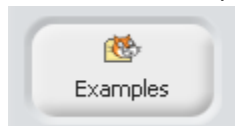
¹Programming concepts and skills supported in Scratch, <http://info.scratch.mit.edu/sites/infoscratch.media.mit.edu/files/file/ScratchProgrammingConcepts-v14.pdf>

Copyright 2011 g8Four, LLC. This work is licensed under Creative Commons Attribution-NonCommercial 3.0 License. Permission is granted to make use of this document for non-commercial educational purposes.

Click on File at the top of the screen in Scratch and select Open.



On the left side of the Open Project Box, click on Examples.



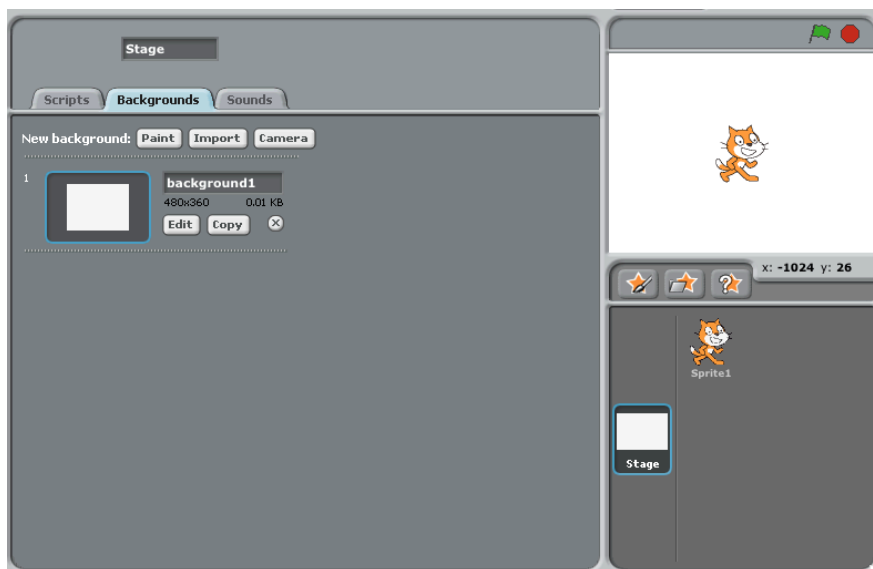
Open that last folder, Stories; and select the project PacMan to view.

You can ask you students to tell you what types of elements they saw in the project. Encourage them to keep these elements in mind as they will use some of them to create their own games.

Step 2) Project Building

The students will learn to create a maze game. First have the students create a background for their maze.

2) Click on the "Backgrounds" tab, then the "Paint" button.



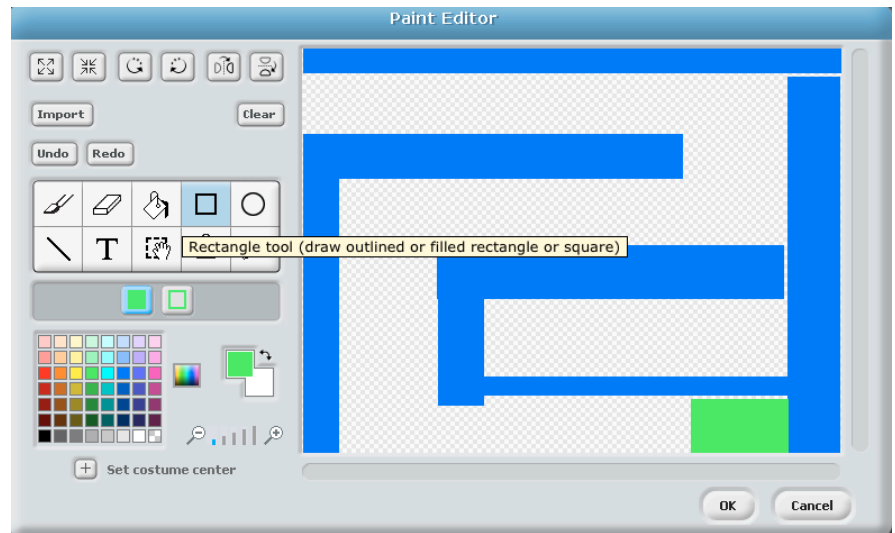
1) Click on the stage thumbnail.

3) First select a color for the maze wall. This color will be important later in the project building.

Select the *Rectangle Tool* and draw several rectangles to create the walls of your maze.

Place a different color rectangle at the end of your maze to represent the finish line.

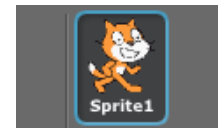
Press OK when you are done.



4) Next select a sprite to be the main character in your game by clicking. Click on the star with a paintbrush icon to paint a new Sprite for the first project element. This will open another Paint Editor window, similar to painting a background. The students can also choose to use a Sprite from file by selecting the star with a folder icon.



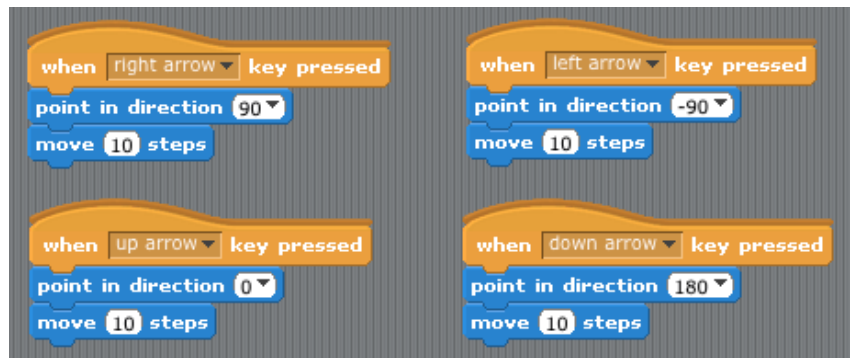
5) Click on the newly created Sprite thumbnail underneath the stage to access the scripts of this Sprite.



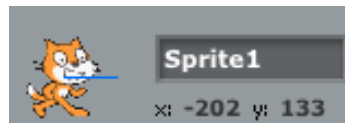
7) If the Sprite is too large for the path of the created maze, select the *Shrink Sprite* tool and click on the Sprite until it is the right size.



8) Create the following stacks of code. This code programs the Sprite to move in the direction of each arrow key.



9) Now we need to program the starting position of the Sprite. Place the Sprite at the beginning of the maze. The x- and y-coordinates of this location of the Sprite can be found about the Script. Using these coordinates, build the following blocks.



10) Next build this set of blocks. This code programs the Sprite to go back to the starting place if it runs into the wall of the maze. The touching color block is found under sensing. Click on the box in this block to select the color of the walls of the maze to send your Sprite back to the starting place. The mouse pointer will turn into an eyedropper when the box is clicked. Use the eye dropper to select the color of the maze walls on the stage. The box will change to that color.

11) Finally, we need to program the Sprite when it reaches the finish line. Build this code! The color of the finish line should be the color selected for this “touching color” block. The Sprite will say “You won!” when it reaches finish line. You can add other blocks to make the project unique!



12) When you are finished, save your project.

