Scratch Basic

Estimated duration: 1 hour 30 minutes

Age level: Lower secondary school; suitable for students 10 years old and above.

Learning objectives, skills and competencies: Students will familiarise with machines, algorithms, programming languages, instructions, events, conditionals, directions, Cartesian plane, coordinates and debugging.

By the end of this lesson, students will have learnt that:

- an algorithm is a process (recipe) to resolve a problem;
- Scratch is a visual programming environment;
- some instructions are only executed if triggered by an event (event programming);
- some instructions are executed one after the other (sequential programming);
- some instructions are executed only if a specific condition is satisfied (conditionals).

Activities and roles

Students discover the Scratch environment and create a game with a maze. The teacher provides instructions, monitors the class and provides assistance when necessary.

What do you need?

For the classroom:

- a video projector (or a smart board)
- (OPTIONAL) a USB key to save all projects

For each pair of students:

- a computer connected to the Internet or a computer on which the Scratch software is already installed (a Scratch icon on the Desktop is useful)
- a computer mouse

Learning space

School classroom

Activity description

Step 1: Intro (5 minutes)

The goal of this lesson is to create a maze game: (human) players will guide a (virtual) character through a maze to reach a treasure.

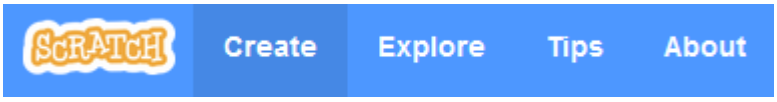
Show the class what the end result could look like by launching a Scratch project you will have created in advance. This will motivate the students.

Here is a simple version of the maze:

https://scratch.mit.edu/projects/240591995/

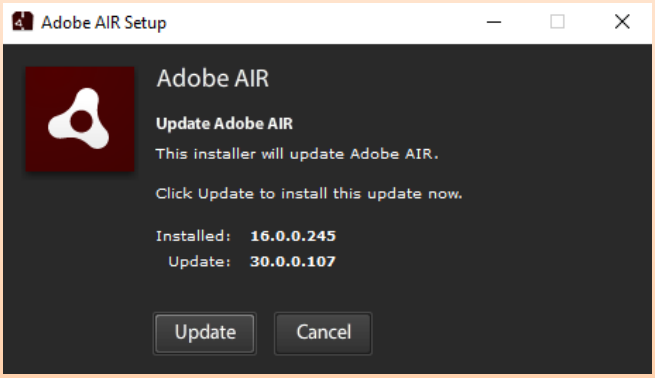
Step 2: Basic Instructions (10 minutes)

Invite students to connect to Scratch. If students are working online, they can find the Scratch website at https://scratch.mit.edu/ or by typing "Scratch" in a search engine (Google, for example). Next, they should click on the "Create" button.

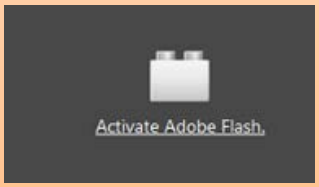


Tip

On Scratch's desktop version, students might be prompted to install updates. It's best to refuse and install the updates after class.



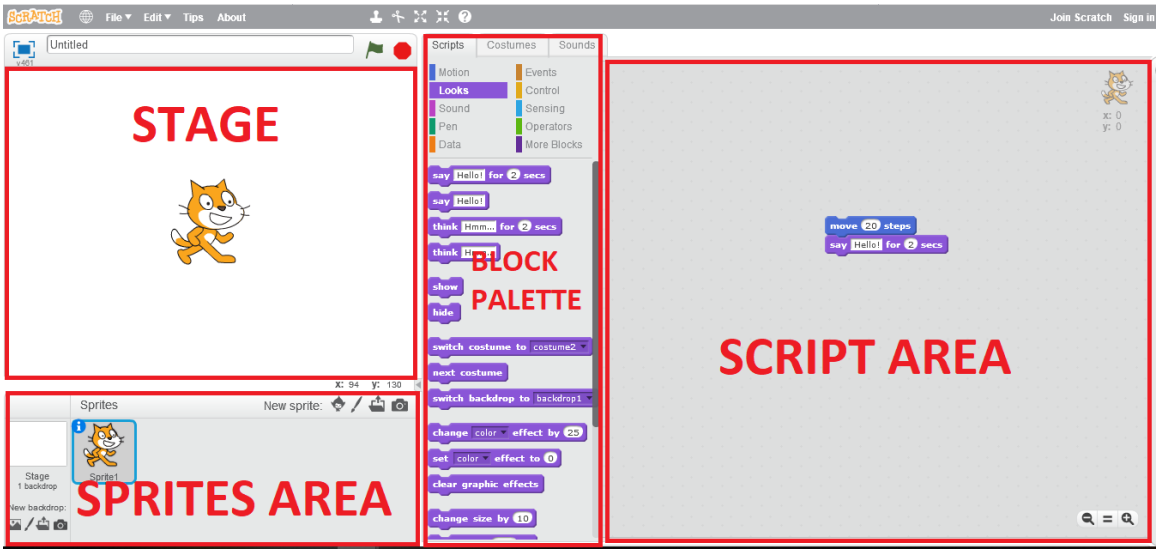
On the online version, students may be prompted to activate Adobe Flash:



They should do so by clicking on the link and then on the button "Authorize".

Show the main sections of the Scratch environment:

1. Stage (this is where the animation/game happens);
2. Sprites Area (characters or objects which are programmed);
3. Block Palette (set of instructions/blocks used to program the sprites);
4. Script Area (where the program is "written").

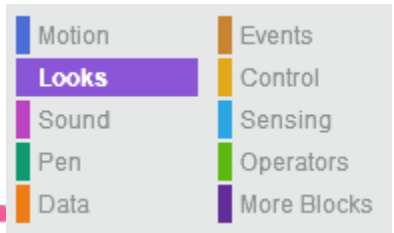


Next, show how to make the cat (the default sprite) move forward by dragging and dropping the “move 10 steps” instruction from the block palette to the script area.

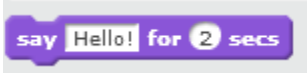
Click on the instruction. The cat moves forward of 10 steps (10 pixels).

If we want the cat to move 20 steps, we can replace “10” by “20”.

If we want to make the cat say something, we change the category of blocks to “Looks”,



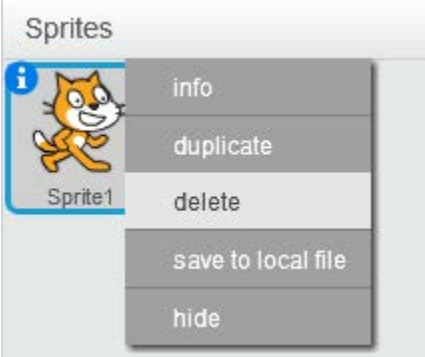
drag the instruction “say Hello! For 2 seconds” to the script area



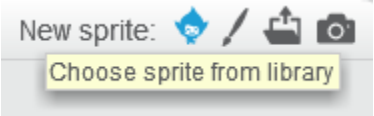
and “snap” it to the first block.

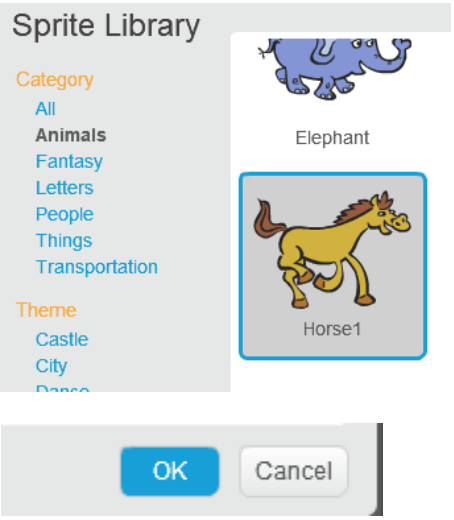
To delete a block, we drag the block back to the block palette.

Show how to delete the cat sprite by clicking with the right mouse button on the cat in the Sprites Area and selecting “delete”.



Show how to add a new character by clicking on the “Choose sprite from library” button.





Once selected, the new character (sprite) will appear somewhere on the Stage. You can move the character around the Stage by clicking it and dragging it to the desired position.

Step 3: Sprites and Free Exploration (5 minutes)

Ask the students to pick:

1) a character which will go through the maze (we chose a horse);
2) a treasure (we chose a pair of glasses).

If they are new to Scratch, let them explore it on their own. They can test different categories of blocks.

Tips

You may want to restrict/monitor the use of the Sound category. With a large group, the sound can quickly become annoying especially when combined with loops...

Remind the students to exchange roles. They should each take turns at controlling the mouse and keyboard.

Step 4: Making the Character Move (15 minutes)

The player should be able to move around the game's character. Ask students how they would like to implement this functionality.

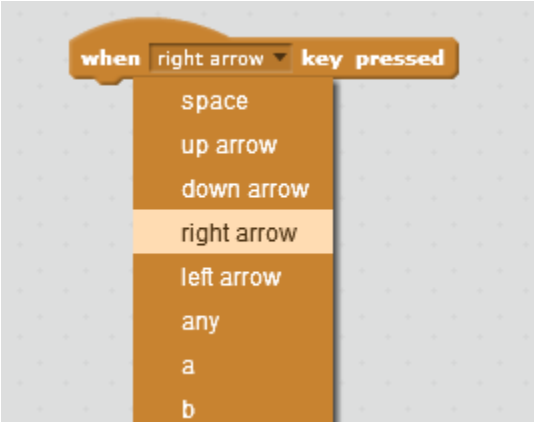
As we do not have video game controllers, we will use four keys on the keyboard: the up, down, left and right arrows.

When the player presses the right arrow, the character should move right.

Ask the students to "translate" this instruction into a Scratch script. You can give them a hint by indicating that they need a block from the Events category and one from the Motion category.

Students should come up with this script:

To do so, they must select the right key by clicking on the little triangle:



When the player presses the left arrow, the character should move left.

Here the simplest solution is to make the character go backwards:

Notice the *minus* 10 when moving to the left.

When the player presses the up arrow, the character should move up.

This said, what will happen when the player presses the up arrow?

This is where you can introduce the concept of Cartesian plane. As you can see at the top right

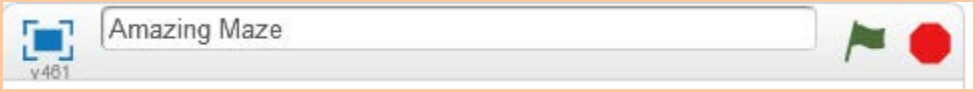
of the Script Area, each character has coordinates:

To move up or down, the value of the ***y*** coordinate has to change:

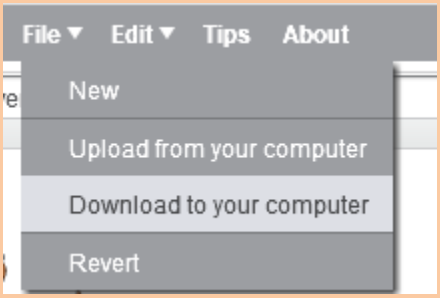
Step 5: Saving (5 minutes)

A. Scratch is used in the browser (online editor)

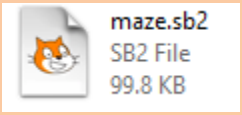
- Students should first give a name to their project:



- From the File menu, students can download the project to the computer:

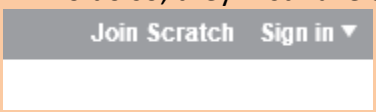


- This will create a SB2 File which can be saved on the desktop in any folder.

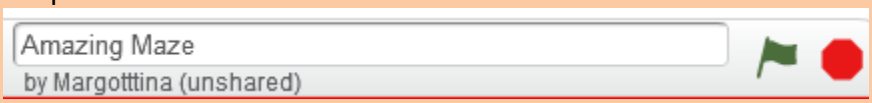


- Further changes to the project will **not** be saved automatically to the SB2 File. Students have to "overwrite" the file by downloading the file again (repeating the previous steps).

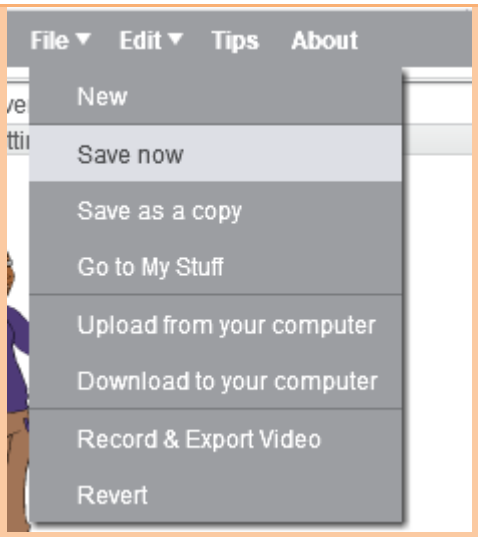
- Alternatively, students can save their work online (i.e. on the Scratch servers). To do so, they first have to either join Scratch or sign in.



- Once signed in, the project will automatically be saved in the student's online portfolio.

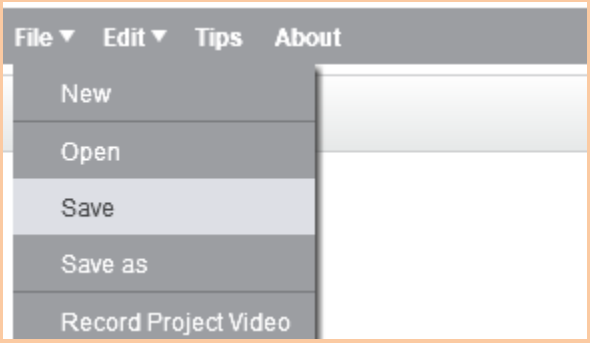


- At any point, if the « Saved » notification does not appear, students can manually save by clicking « Save now » in the File menu.


File
Edit
Tips
About
New
Save now
Save as a copy
Go to My Stuff
Upload from your computer
Download to your computer
Record & Export Video
Revert

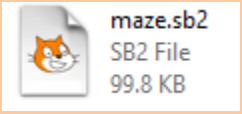
B. Scratch is installed locally (offline editor)

- Students can save by clicking "Save" in the File menu:



- Students will need to give a name to their project.

- This will create a SB2 File which can be saved on the desktop or any other folder.



Tips

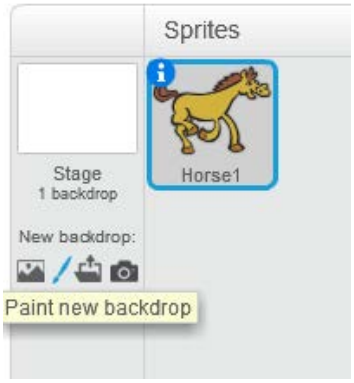
Joining Scratch can be a long and tedious process; all the more so with large groups. We suggest you request a Scratch Teacher Account to better manage your students' participation. See this page to learn more about Teacher Accounts.

In order for students to easily find their projects later on:

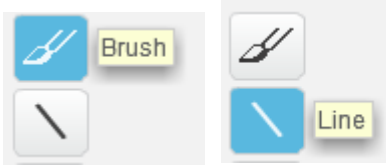
- Number the computers;
- Ask students to incorporate their names in the file names.

Step 6: Creation of Background (20 minutes)

Show how to change the background by clicking on the “Paint a new backdrop” button.



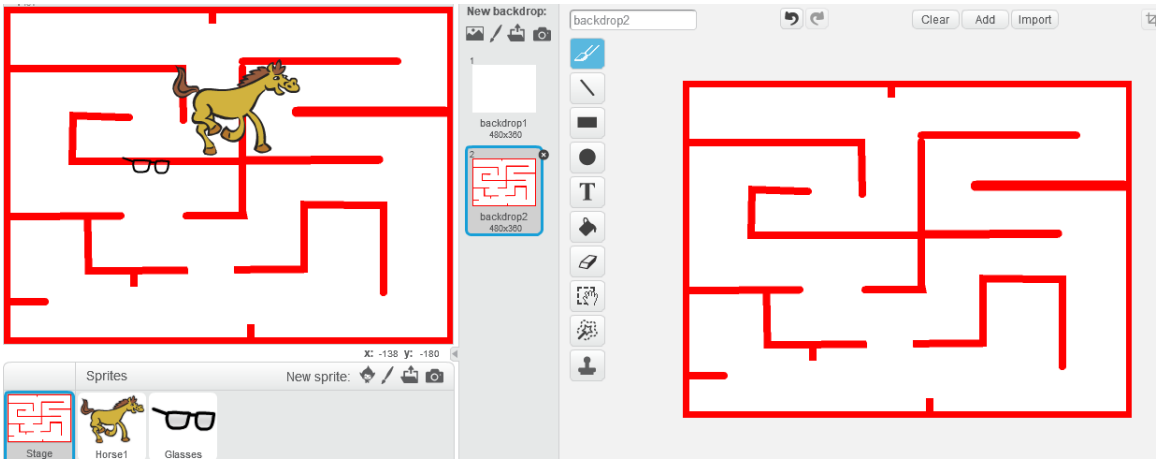
Choose either the Brush or Straight line tool from the icons on the left of the window.



Select a color (preferably a bright one!) and make the line width thicker by dragging the bar at the bottom of the screen.

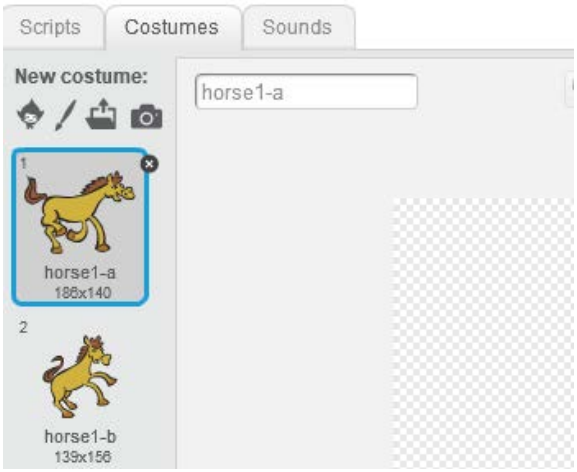
Time to draw the maze!

Your character may be too big to move between the walls.

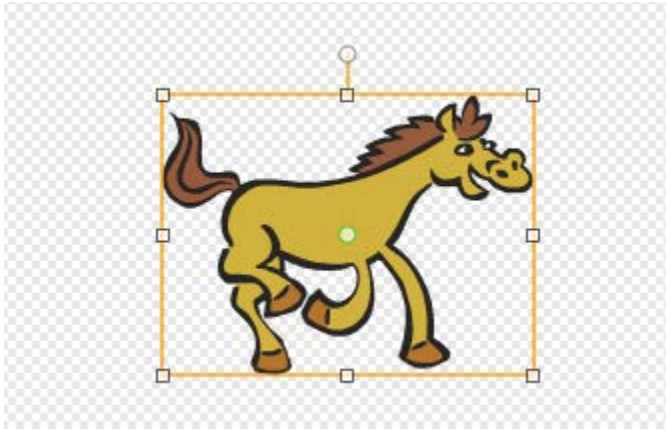


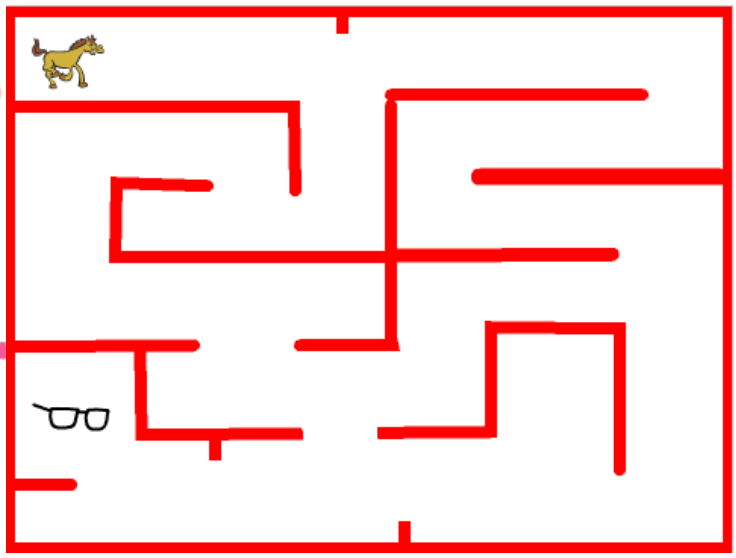
No problem, you can resize it, by clicking on it in the Sprites Area:

Make sure the “Costumes” tab is selected.



Click on the character and a yellow box will appear around it. You can resize by dragging in the outer little squares.





Step 7: Full Walls (20 minutes)

Now, our character can go through walls… not the most efficient maze!

If the character touches a red wall, the character should move back.
In order to recognize the walls, the character needs to "sense" them. In Scratch, a sprite may sense whether it is touching a particular color – red, for instance – present in the background or another sprite. Unsurprisingly, you will find the color-sensing block in the Sensing category.

The character will backtrack only ***if*** a wall is touched. This is a conditional statement. " If" blocks can be found in the Control category.

Students should come up with something like this:

This would also work:

In programming, there are often several ways of achieving the same result. That is ok!

The same logic can be applied to movements towards the left, top and bottom.

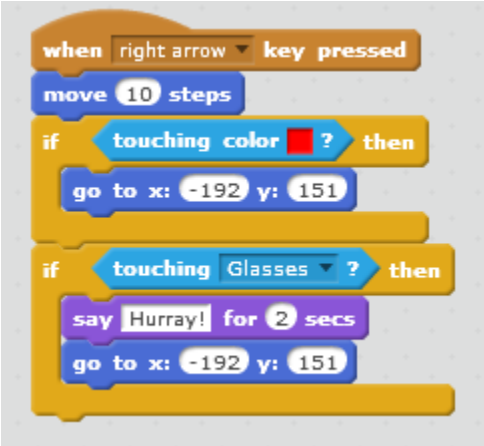
Step 8: Winning the Game (10 minutes)

When the character finds the treasure, it should scream of joy!

Going further…

Students may come up with different ideas for their game. For instance, when the character touches a wall, it is "thrown" back at the beginning of the maze.

What about playing again? The same strategy can be used:



As a final step, you can make students realize that a lot of the script is repeated. For instance, those two scripts are identical, except for the second instruction:

Instead of repeating the same set of instructions for each direction (right, left, up, down) could

we not use *one* set of instructions to be applied during the *whole game*? Of course!

If you want to see how this can be done, please consult the "Upper Secondary" version of this lesson plan.

Have fun teaching Scratch!

Notes

- Take at least an hour or two to get familiarized with Scratch prior to teaching this lesson. You can follow our video tutorials. You will see, it is lots of fun, even for adults!
- Take a few moments before the lesson to make sure all computers are connected to the Internet (if necessary) or have Scratch installed.
- This lesson is designed for students aged **10 years old and above** who can manipulate a keyboard and a mouse. We assume they have never worked with Scratch.
- Pair programming works best. The ideal setup is to have 2 students per computer, alternating control over the keyboard and mouse every 10 minutes.
- Losing hours of work is extremely frustrating! Make sure students have a strategy to save their work. See **step 5**.

Name of author: Margo Tinawi