

# Introduction to Scratch

---

# Scartch

---

- Scratch is a free programmable toolkit that enable users to create their games, animated stories and interactive art and share their creations with one another over the internet.
- Fun and Educational

## **Usage of scratch allows you to:**

- Create solutions to problems using computers
- Study information
- 'Invent Algorithms
- Write programs to implement the algorithms
- Scratch is developed by the lifelong Kindergarten group at MIT Media Lab

# Scartch

---

<https://scratch.mit.edu>

**Motion**

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 0 y: 0
- glide 1 secs to random position
- glide 1 secs to x: 0 y: 0
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 0
- change y by 10
- set y to 0



Scratch workspace with a cat sprite and a small cat icon in the top right corner.

Stage area showing the cat sprite.

Sprite and Stage properties panel. Sprite: Sprite1, x: 0, y: 0, Size: 100, Direction: 90. Backdrops: 1.

**Motion**

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 0 y: 0
- glide 1 secs to random position
- glide 1 secs to x: 0 y: 0
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 0
- change y by 10
- set y to 0

The central workspace is a large grid area where code blocks are executed. A small Scratch cat sprite is positioned in the top right corner of the workspace. A large orange rounded rectangle with the text "Programming Area" is centered in the workspace, with a thick orange arrow pointing downwards towards the grid.

The stage area shows the Scratch cat sprite in the center. There are small window control icons (maximize, close) in the top right corner of the stage area.

Sprite: Sprite1

x: 0 y: 0

Show:

Size: 100 Direction: 90

Backdrops: 1

Scratch editor interface showing the 'Code' tab, 'Motion' category, and a central workspace. An orange callout box labeled 'Costume Design Shop' points to the 'Costumes' tab in the bottom right corner.

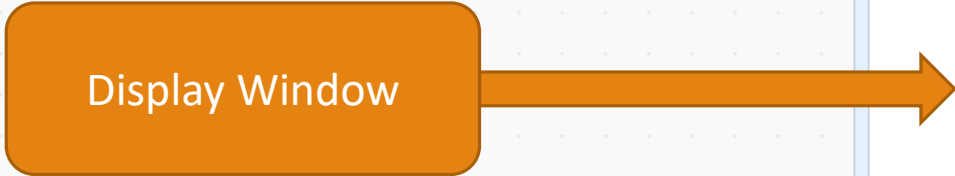
**Code Tab - Motion Category:**

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 0 y: 0
- glide 1 secs to random position
- glide 1 secs to x: 0 y: 0
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 0
- change y by 10
- set y to 0

**Stage Area:**

- Sprite: Sprite1
- Coordinates: x: 0, y: 0
- Show:
- Size: 100
- Direction: 90
- Backdrops: 1

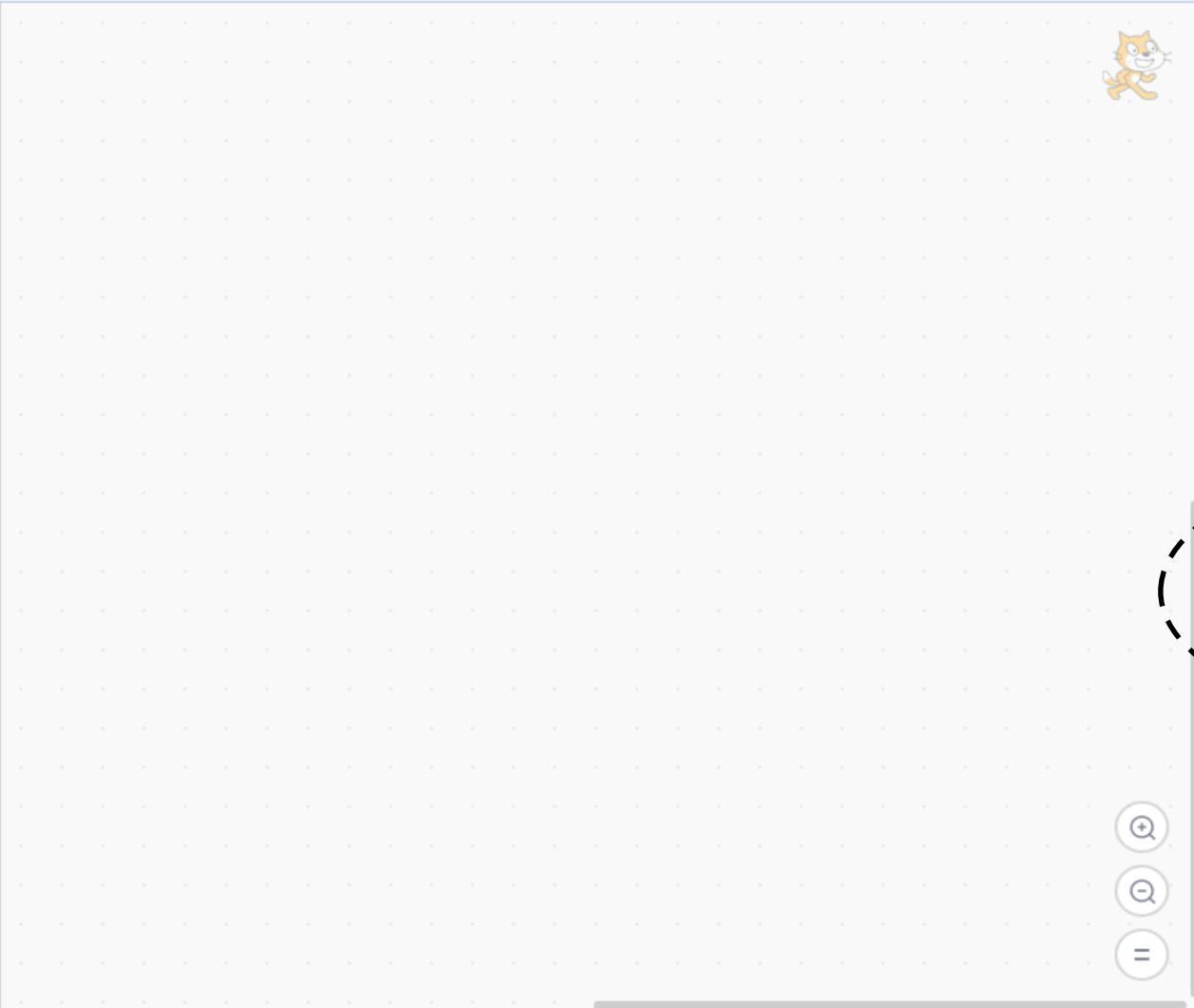
Scratch editor interface showing the 'Code' tab. The left sidebar contains various block categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The main workspace is a grid with a small Scratch cat sprite in the top right corner. An orange callout box with the text 'Display Window' and an arrow points to the right-hand side of the editor, which contains the 'Sprite' and 'Stage' panels. The 'Sprite' panel shows 'Sprite1' with a size of 100 and a direction of 90 degrees. The 'Stage' panel shows a single backdrop.



Code | Costumes | Sounds

**Motion**

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 0 y: 0
- glide 1 secs to random position
- glide 1 secs to x: 0 y: 0
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 0
- change y by 10
- set y to 0



Stage properties for Sprite1:

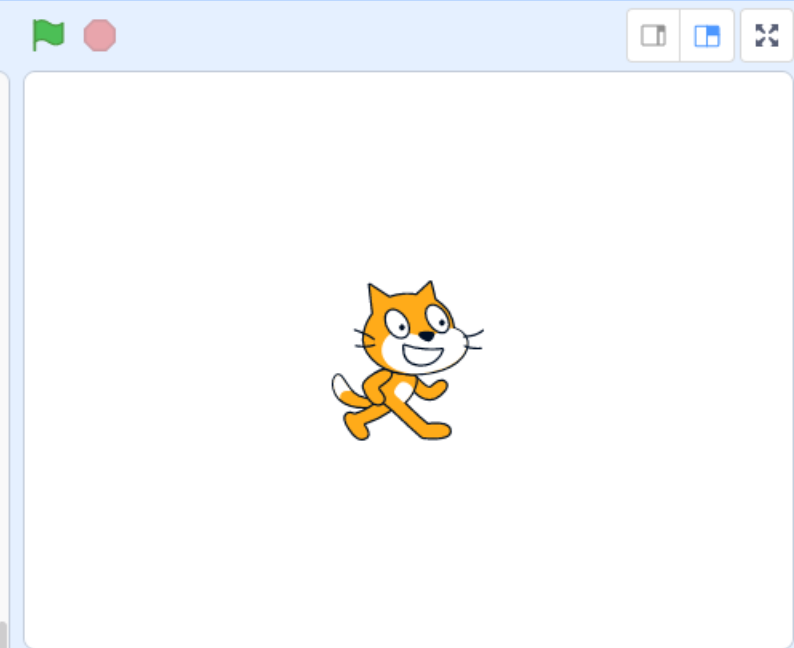
- Sprite: Sprite1
- x: 0, y: 0
- Show:  Show,  Hide
- Size: 100
- Direction: 90
- Backdrops: 1

Bottom right icons: Add sprite, Add backdrop



**Motion**

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 0 y: 0
- glide 1 secs to random position
- glide 1 secs to x: 0 y: 0
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 0
- change y by 10
- set y to 0



Sprite: Sprite1 x: 0 y: 0

Show:   Size: 100 Direction: 90

Backdrops: 1

Choose a Sprite

# Sample Exercises on Scratch

---

## Create a new Scratch project

Let's start by creating a new project.

- Visit <https://scratch.mit.edu/>

# Step 1: Add a scrolling background

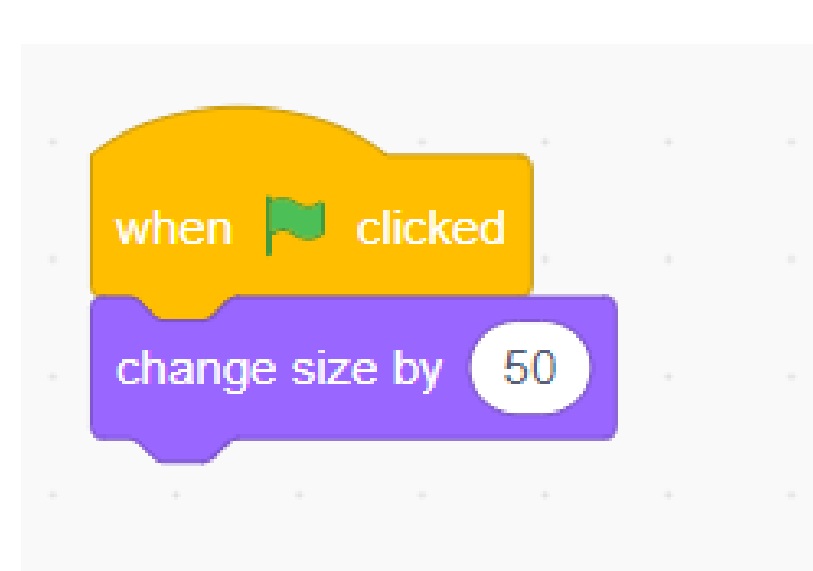
---

To give the effect of moving through space, we created a simple scrolling background.

- Start by deleting the Scratch cat sprite that appears in every new project. Click on the cat sprite and the trash can icon.
- Click the “Choose a Backdrop” icon and add the “Galaxy” backdrop.

The stars will move while the backdrop stays in place.

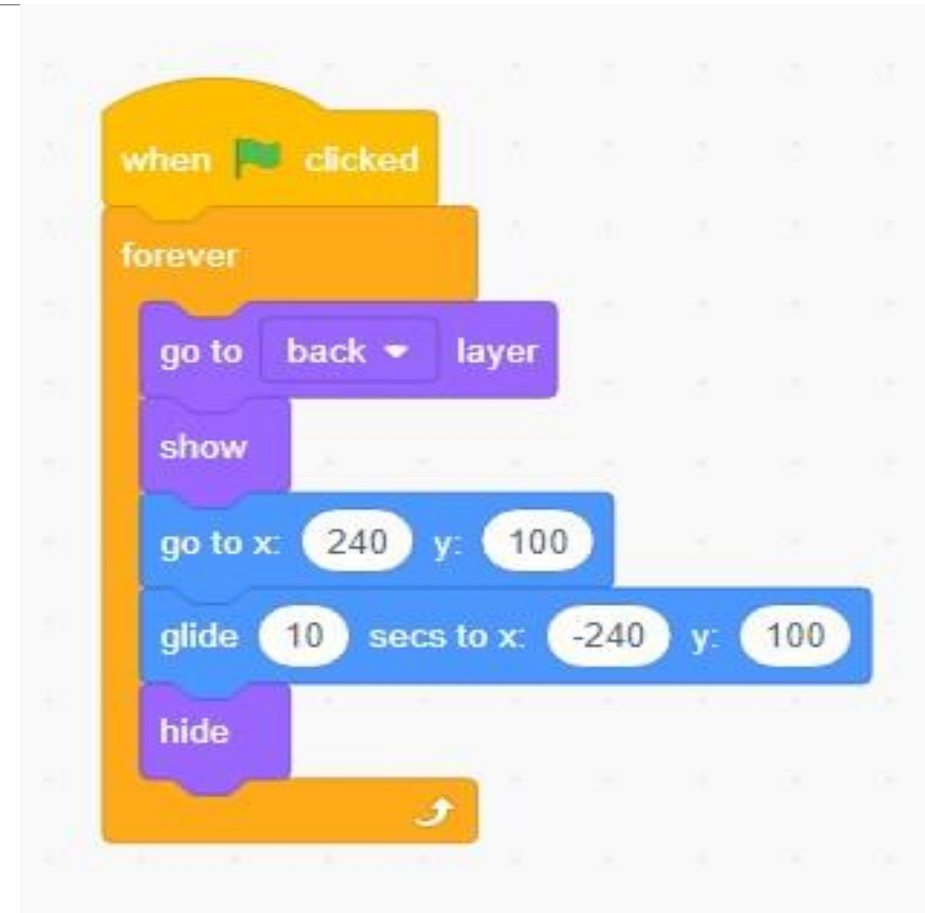
- Click upload sprite and upload the stars.png file
- Change the size of the Stars sprite to 50 and position it on the screen
- Add the “When Green Flag Clicked” block from the “Events” category



# Step 1: Add a scrolling background

- Use a “Forever” block with the “Go to Back Layer” block to make sure the stars show behind other sprites.
- Then, use a “show” block to make the stars visible
- Position them with the “go to” block and use the “glide” block to move it left for 10 seconds.
- “Hide” the sprite when it reaches the edge of the screen and the forever block will start this animation over again.

**Hint:** Add more images or use a different image than stars.png to customize your scrolling background.



# Step 2: Add flying cat and move it

---

Let's position our flying cat and add the logic to move it up and down with arrows.

- Select "Choose a Sprite", find the Cat flying sprite, and click to add it
- Use the "Go to" block to position the sprite at x= -150 y=0
- Add music with the "Forever" and "Play Sound Until Done" block. We're using the Dance Magic sound which you can add in the Sounds tab



# Step 2: Add flying cat and move it

---

Now, add logic to the arrow keys so our cat moves up and down when we click.

- Under events choose the “when key pressed” block and select “up arrow”
- Underneath use a “change y by” block and enter 5 to move up
- Go back to events, choose the “When key pressed” block and select “down arrow”
- Underneath use a “change y by” block and enter -5 to move down

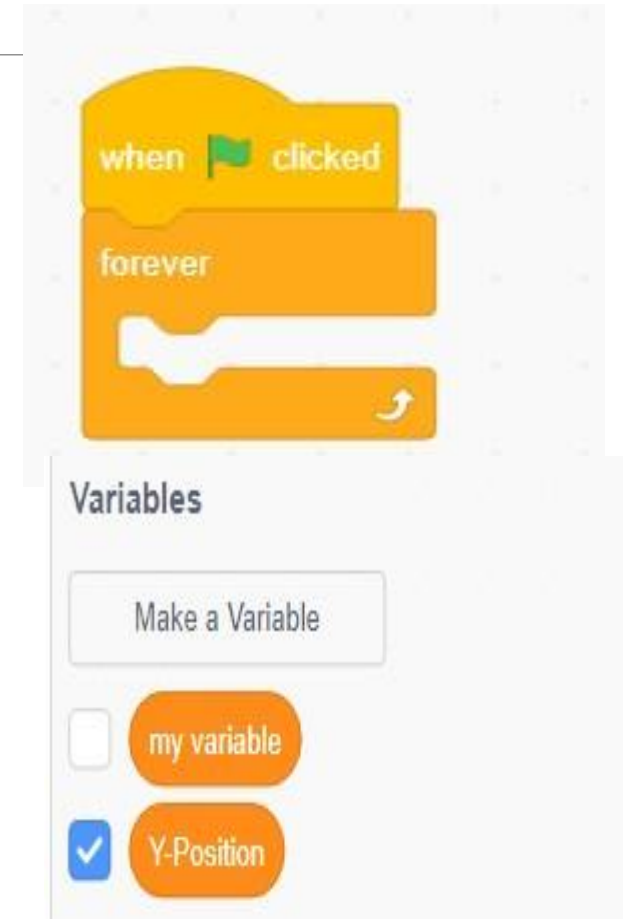
**Hint:** Change the music by adding more sound options in the “Sounds” tab. Then select them in the “play sound” block



# Step 3: Show random crystals and move them

We want random crystals to appear on the right side of the screen and move toward the left.

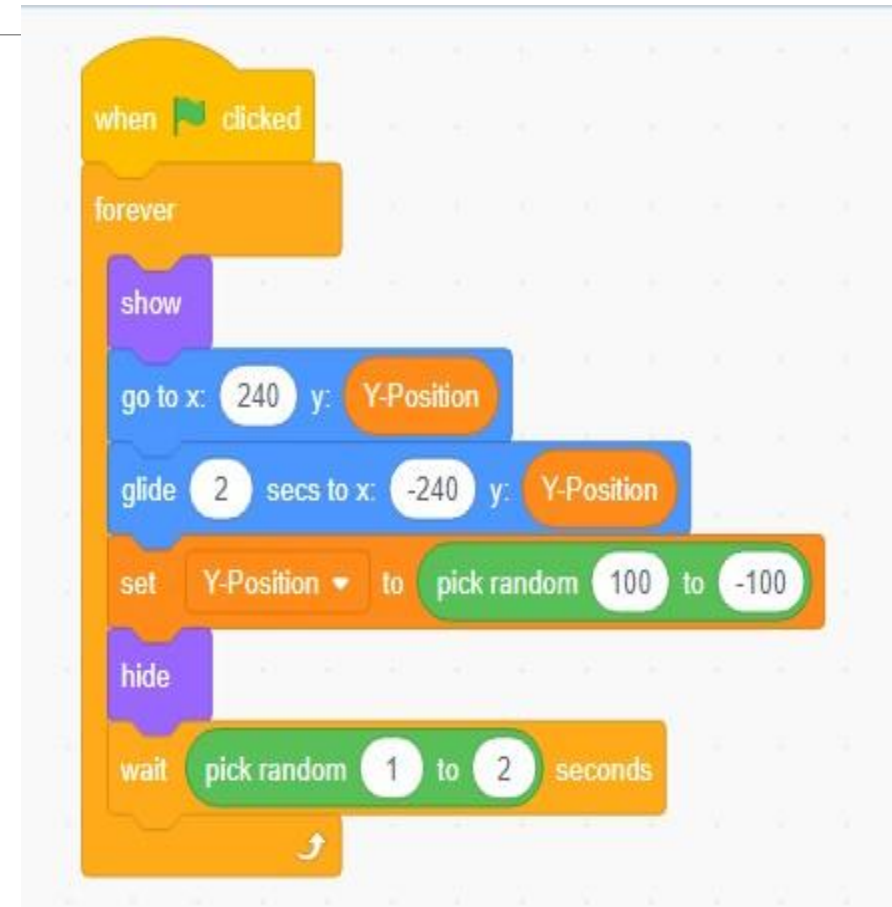
- Select “Choose a Sprite”, find the Crystal sprite, and click to add it
- Add the “When Green Flag Clicked” and a “Forever” block underneath it to repeat the movement of the crystals
- Go to the “variables” blocks and “Make a variable”, name it Y-Position, and uncheck it



# Step 3: Show random crystals and move them

- Inside the “forever” block “show” the crystal and set it to a random position on the right side using “go to x” with x=240 and y=y-position
- Then, glide 2 seconds to x=-240 and y=y-position
- Use “set” y-position to a random number between 100 and -100, so it shows up in a different position next time
- “Hide” the crystal then wait a random amount of time between 1 and 2 seconds to repeat the loop

**Hint:** Select a different sprite or draw your own sprite to customize your collection item.



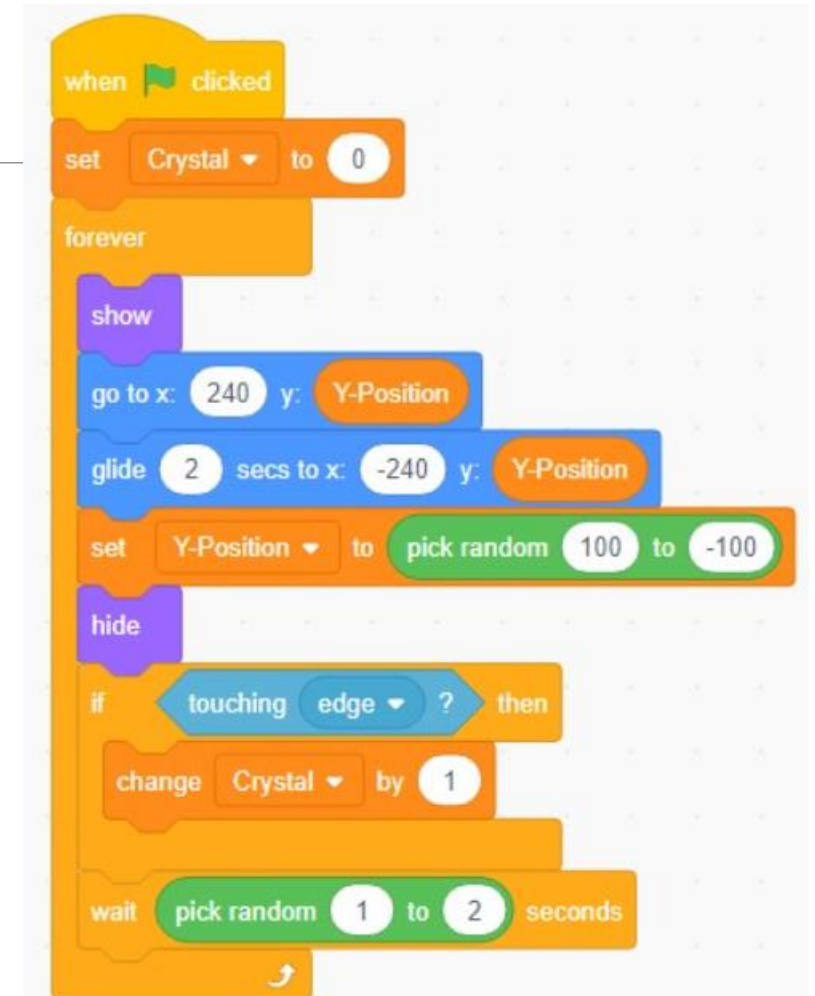


# Step 4: Add up losing crystals

When a crystal flies by us and we don't catch it, we want to count it.

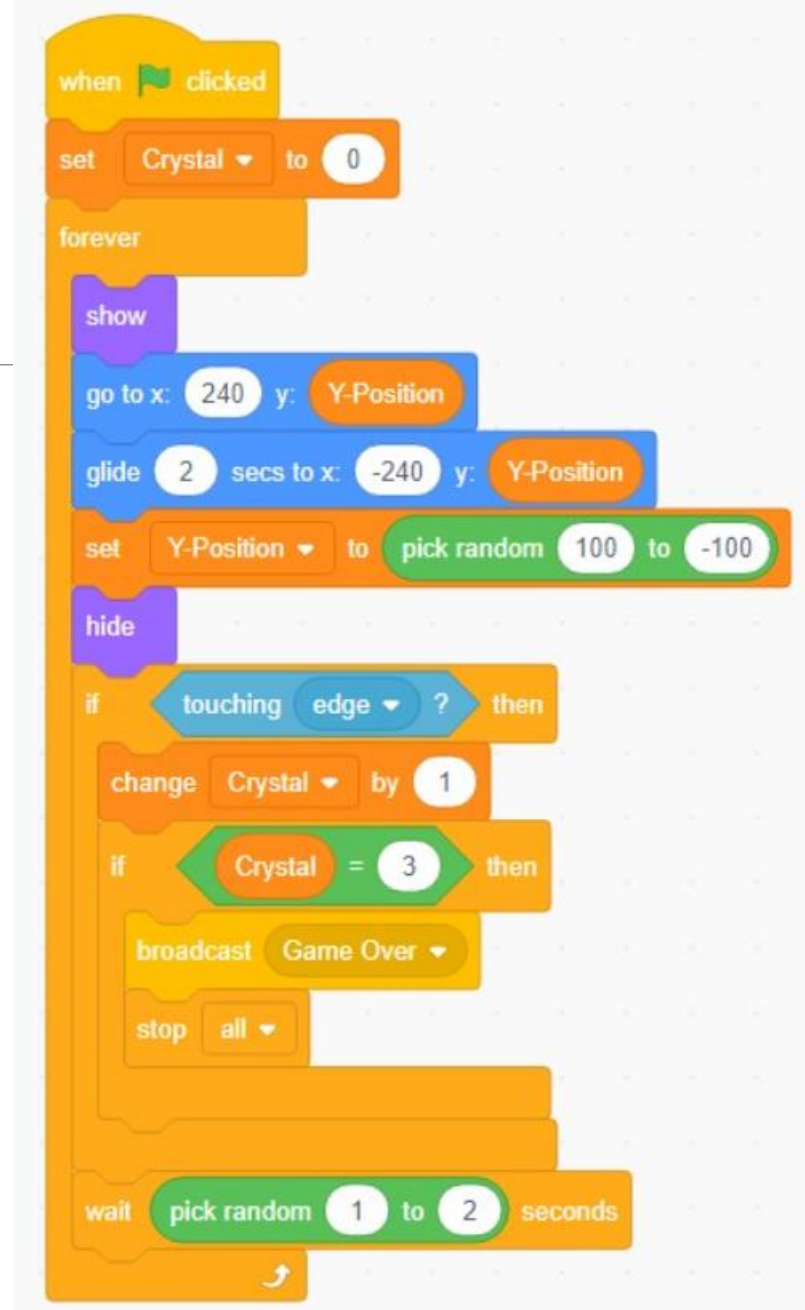
When we reach 3 crystals, the game is over. Let's add on to the code you started in Step 4.

- Go to the “variables” blocks and “Make a variable”, name it Crystal, and uncheck it
- Under “when green flag clicked” set the Crystal variable to 0
- Create an “if” block with the condition “touching edge”.
- Inside this “if” block, change the crystal variable by 1. Add it under the “hide” block. Now each time it touches the edge, we add one.



# Step 4: Add up losing crystals

- Add another “if” block inside the first “if” block. Use the condition “crystal = 3”.
- Then we will broadcast a new message named “Game Over”
- Add a “stop all” block to stop the game once the number of crystals reaches 3.
- **Hint:** You can give different names to your variables, just make sure you are using the right variable in your code.



# Step 5: Increase score for gaining crystals

---

Every time our flying cat touches a crystal, we want to increase our score. In addition to increasing the score, we will also decrease the count of the crystal variable. We do this because even though we are hiding the crystal, it still touches the edge each time.

- Go to the “variables” blocks and “Make a variable”, name it Score. Keep it checked so it shows on your game screen.
- Add the “When Green Flag Clicked” and set the Score to 0
- Next, add a “Forever” block underneath



# Step 5: Increase score for gaining crystals

Inside the forever block, we will add to the game score every time our cat touches a crystal.

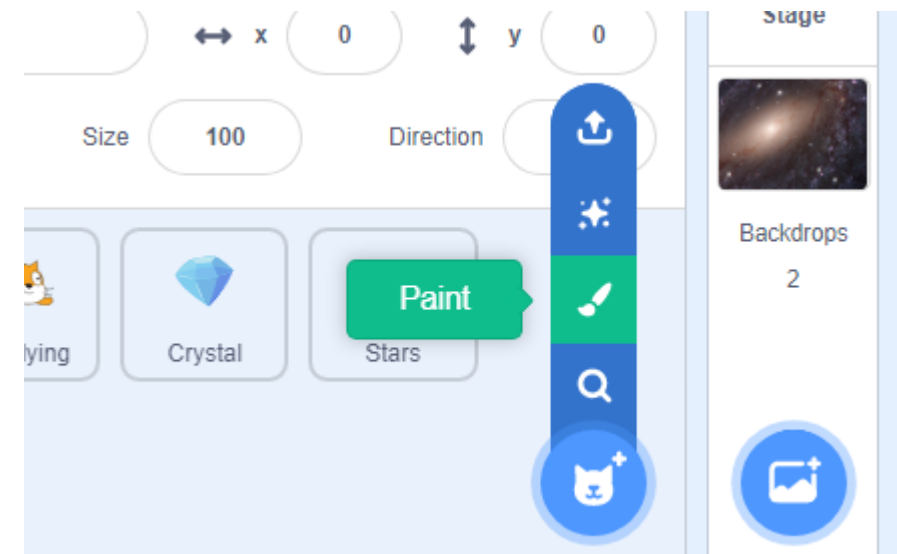
- Use an “if” block with the condition “touching cat flying” sprite
- Add the “start sound” block and select “magic spell” for a sound effect
- Then change the score variable by 1 and the crystal variable by -1
- Hide the crystal once it’s collected



# Step 6: Show “Game Over” screen

Your game is almost complete! Let the players know when the game is over by showing a game over screen. This will trigger when you’ve missed three crystals.

- Select “paint” under sprites and rename this sprite Game Over.
- In the costumes tab, select the rectangle tool and click and drag to draw a rectangle
- Then, use the type tool with font “marker” and add the message “Game Over”
- Adjust the message on your game screen



# Step 6: Show “Game Over” screen

---

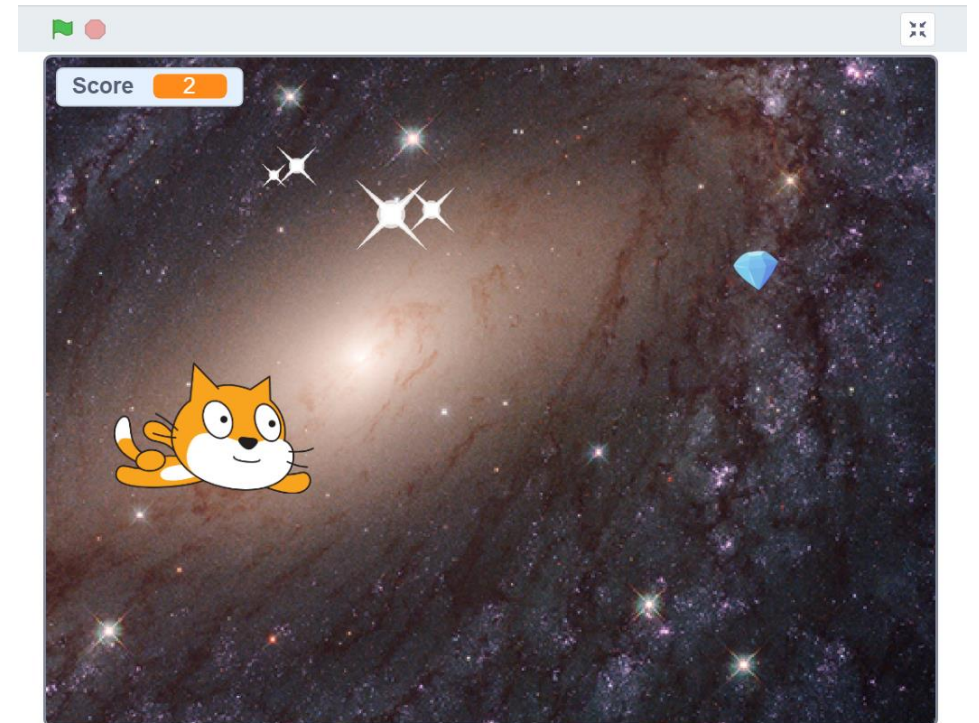
- Go back to the “Code” tab
- Add the “When Green Flag Clicked” and hide block. That means this message will not show when the game starts
- Add the “receive broadcast” block and select game over. This is the message we sent from the crystal sprite when the crystal variable reached 3.
- Then, add the “stop all” block to end the game.
- **Hint:** You can change the font, colors, and text for your Game Over message in the “Costumes” tab.



# Your Flying Space Cat game is complete!

---

Click the green flag and give it a try. If you want to let other people see your project and remix it, click the “Share” button at the top.



Games provide numerous opportunities to explore a variety of computational concepts and skills. Here are some blocks that are frequently useful in games.

### TOUCHING

See if two sprites are touching or if a sprite is touching a color



### VISIBILITY

Make a sprite appear or disappear



### RANDOM

Get a computer-generated number from within a specified range



### TIMING

Have the computer keep track of time for you



### STRINGS

Test, access, and change words and sentences



### VARIABLES

Store a number or string in a container to access later



### COMPARE

Compare values to help make decisions within your game



### KEY PRESS

Make a sprite respond when different keys are pressed





---

**THANK YOU**