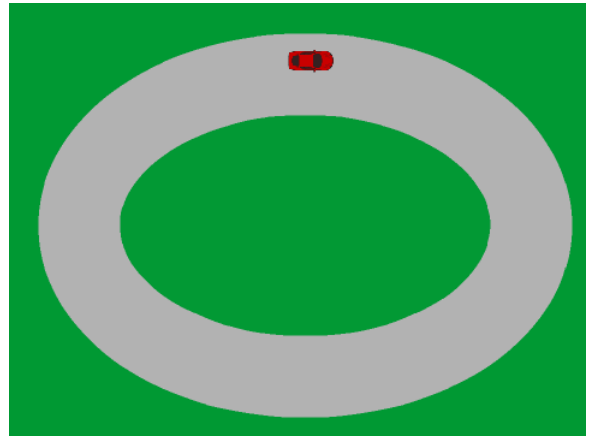


Driving Game

<http://scratch.mit.edu/projects/30428680/#editor>

Drive around a track in your fast red car!

The game uses an oval race track as the background. There is one sprite in the program, a red car. If you click on the red car sprite, you will see that there is already code for turning the car to the right when you press the right arrow key on the keyboard. Try it out. Click on the green flag to start the program, and then press the right arrow key. The car will turn. (Finished game: <http://scratch.mit.edu/projects/30356000/>)



1. Driving

EASY

Right now, the car can turn but it doesn't go anywhere. Change the program so that the car always moves forward slowly.

Now the car moves forward, and you can press the right arrow key to turn right. Can you drive around the race track?

Try changing the program so that when you press the left arrow key, the car turns to the left. Then you can turn in both directions.

Now change the program so that the car doesn't move forward all the time. Have it only move forward when you press the up arrow key.

2. No Shortcuts

MEDIUM

Right now, you don't have to stay on the road. You can drive over the grass as a shortcut. This is unfair. Can you make it so that the car moves forward more slowly if it is touching the green area?

Try It: Speed and Brakes

HARD

Your car only has one speed. It can move forward or it can stop. Let's change the program so that you can go different speeds.

First, create a variable called "speed" (only for this sprite). A variable is a number that you can change. Instead of the car always moving forward by the same number steps every time, you can change that number.

Then, in your code, whenever you move forward, replace the number of steps you move forward by the "speed" variable. If the "speed" variable contains a bigger number, then the car will move forward by a bigger amount, and the car will go faster. If the "speed" variable contains a smaller number, then the car will move forward by a smaller amount, and the car will move slower.

Change your program so that when you press the up arrow key, the car goes faster. Also change it so that when you press the down arrow key, the car goes slower.

Does the car still move slowly when you drive on the grass? What is the fastest you can drive around the track? Can you make the car drive backwards?

Try It: Different Tracks

EASY

The race track doesn't need to be an oval. You can change it. Click on the "stage" on the left of the screen. Then click on the backdrop tab. Try changing to a different race course background.

Try It: Multiplayer

MEDIUM

You can change your program so that two people can drive cars at the same time. Make a copy of the car and paint this new car to look different. Then change the controls for this new car so that it uses different keyboard buttons for moving. Now try racing against a friend to see who is fastest.

More Ideas

Can you draw your own race course? Remember that in your program, the green is for grass and makes your car go slowly. Can you keep track of the number of laps that you have raced? How about having cars with a limited amount of gas? What about making pit stops where cars can get more gas?

Useful blocks for this exercise

Motion

move steps

turn  degrees

Sensing

key pressed?

touching color

Data

set speed to

change speed by

speed

Events

when  clicked

Control

forever 

if then

Operators

and

<

Clay Shooting

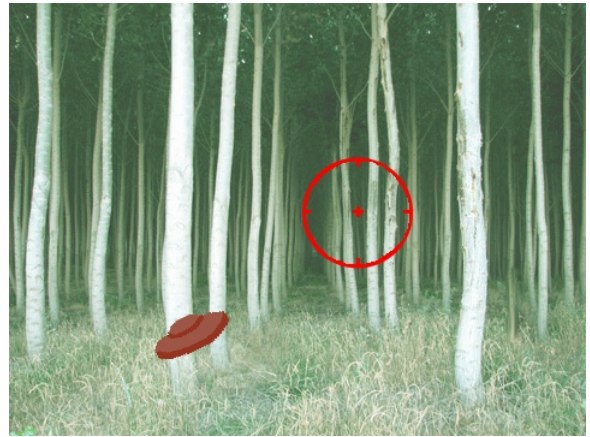
<http://scratch.mit.edu/projects/30428624/#editor>

Shoot at clay targets and see how many you can hit.

When you load the program, it will have a forest background. There are two sprites: a red frisbee called the “clay” and a red crosshair.

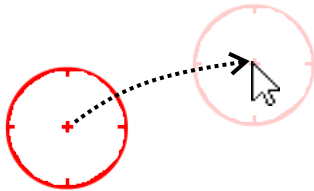
(Finished game: <http://scratch.mit.edu/projects/30374820/>)

* The art for the forest background is from Scratch and is licensed under CC BY-SA 2.0. Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>



1. Move the Crosshair

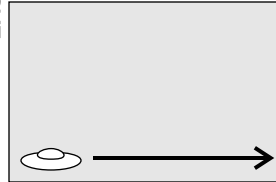
EASY



In the game, you will use the mouse to show where you want to shoot. Since the crosshair shows where you are aiming, the crosshair should always be under the mouse. Write a program for the crosshair sprite so that it always moves to where the mouse pointer is.

2. Flying Clay

EASY



When the game starts, the clay should fly across the screen so you can shoot it. Write a program so that when the green flag is clicked, the clay will start in the lower left of the screen. Use the position $x=-200$ and $y=-125$ for the lower-left of the screen. Then, have the clay move across the screen.

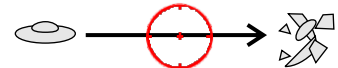
3. Shooting

MEDIUM

Right now, when you shoot the clay, nothing happens. Let's change that. If you click on the costumes tab of the clay sprite, you will notice that there are two costumes: the normal clay, and clay fragments.

Change the program for the clay so that if you shoot it, it will break up into fragments. When the clay moves, it should check if the mouse button is being pressed down and the mouse is touching the clay. If that happens, it means the clay was hit. If it gets hit, the clay should change its costume to the “fragments” costume to show it was hit.

When the game starts, the clay hasn't been hit yet, so the clay should set its costume to the normal “clay” costume when the green flag is clicked.



4. Play Again?

MEDIUM

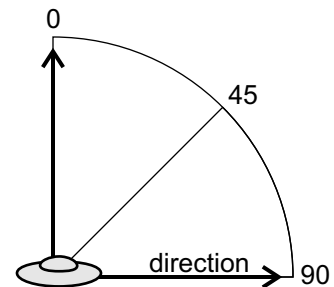
Now, when the game starts, the clay moves across the stage and then stops at the side. Change the program so that when the clay touches the edge of the screen, it goes back to the start.

Don't forget to reset the costume of the clay when it goes back to the start as well.

5. Different Directions

MEDIUM

Right now the clay always moves in the same direction. Change the program so that before it starts moving, it points in a different direction. Since 90 is right and 0 is up, you should pick a random number between 0 and 90 for the direction.



6. No Cheating

MEDIUM

Since the clay always starts at the lower-left of the stage, players can cheat and just always aim in the lower-left. One way to stop this is to hide the clay and wait for a few seconds before it moves. The amount of time it waits should be different each time. That way, players can't be sure when the clay will appear.

More Ideas

Can you keep track of how many clays were hit and how many clays were missed? Can you add sound effects to the game?

Useful blocks for this exercise

Motion

move steps

point in direction

go to

turn degrees

go to x: y:

Events

when  clicked

Looks

switch costume to

show

hide

Control

forever 

if then

wait secs

repeat until 

Sensing

mouse down?

touching ?

Operators

pick random to

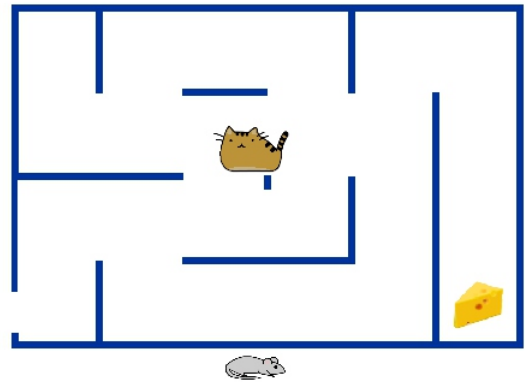
and

Cat and Mouse

<http://scratch.mit.edu/projects/30428582/#editor>

In this game, you are a mouse who needs to get through a maze to get to the cheese. Don't get caught by the cat though!

The program you are given has a blank white background. There are three sprites: the mouse, a cat, and some cheese.
(Finished game: <http://scratch.mit.edu/projects/30373584/>)



1. The Mouse

EASY

Let's start by writing a program to make the mouse move. We'll control the mouse by having it chase after the mouse pointer. Do this by having it turn towards the mouse pointer and then moving forwards. Can you now get the mouse to move around the screen?

2. Cheese

MEDIUM-EASY

The mouse should be able to eat the cheese. Do this by making the cheese disappear when the mouse touches it. Write a program for the cheese sprite. When it is touched by the mouse, it should hide itself. When the green flag is clicked, the cheese should reappear.

3. The Cat

MEDIUM

Write a program for the cat so that it moves back and forth across the screen. It should keep moving in one direction. When it reaches the edge of the stage, the cat should bounce and move in the other direction.

4. Caught!

MEDIUM

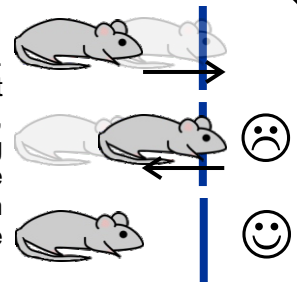
Change the program so that when the mouse touches the cat, the game will end and the program will stop.

5. Walls

MEDIUM

Let's make the game more complicated by adding a maze. Click on the stage, and change the backdrop to the maze backdrop. There's a problem though. The mouse can run through the walls!

We need to use something called collision detection. Every time the mouse moves forward, it should check if it is touching one of the walls. Since all the walls are blue, you only need to check if the mouse is touching something blue. If it is touching something blue, the mouse should move backwards to its starting position where it isn't touching a wall. Make sure you move the mouse away from the walls before trying your changes.



6. The Cat and Walls

MEDIUM

The cat is still ignoring the walls when it moves back and forth. Change the program to be like the program for the mouse. If the cat moves and touches a wall, it should move backwards, back to where it started.

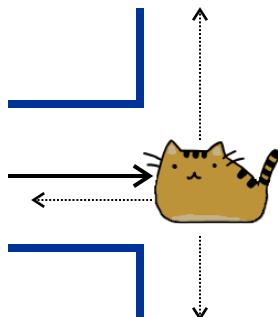
Now when you run the program, the cat will move in a straight line and stop as soon as it hits a wall. The cat keeps trying to move forward, but there is a wall in the way, so it will just stop there. Change the program so that when the cat touches a wall, it will move back to its starting position and then also turn 180 degrees. This means that the cat will turn around to face the other direction. What happens when the cat touches a wall now?

7. Walking the Maze

MEDIUM-HARD

We're now going to make the cat smart enough to walk through the maze itself. In the maze, you can only go up, down, left, and right. If the cat walks around and hits a wall or a corner, it can turn up, down, left, or right. If it does this, it will keep walking through the maze.

Change the program for the cat so that if it bumps into wall, it will turn 90, 180, or 270 degrees. With this change, the cat will wander through the maze by itself.



Try It: Less Jumpy

MEDIUM

When the mouse gets near the mouse pointer, it starts moving back and forth. This happens because when it moves toward the mouse pointer, it goes too far and ends up on the other side of the mouse pointer. Then it turns around and moves again, going back to the start. Can you stop this from happening?

Useful blocks for this exercise

Motion

move steps

point towards

turn degrees

point in direction

move steps

if on edge, bounce

Looks

show

hide

Sensing

touching color

touching ?

distance to

Events

when  clicked

Control

if then

forever

stop

Operators

>

or

pick random to *

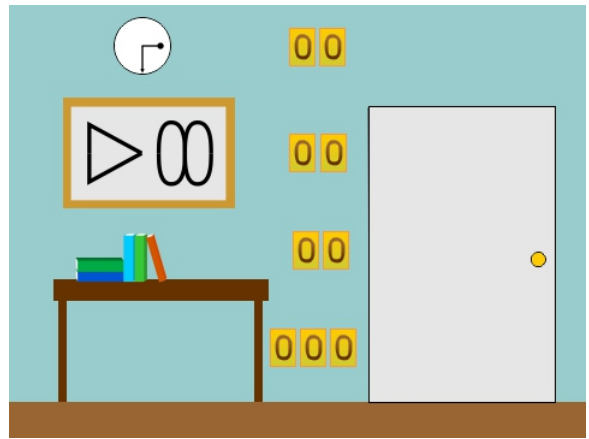
Room Escape

<http://scratch.mit.edu/projects/30428650/#editor>

Can you solve the puzzles and find the secret codes to escape the room? Room Escape games are about solving puzzles and exploring to figure out how to escape a room.

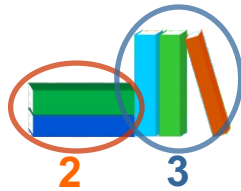
The program has four main sprites—the clock, the painting, the books, and the door—one for each puzzle. When you solve a puzzle, the sprite should disappear, revealing a secret underneath. There are also nine optional sprites for the combination numbers.

(Finished game: <http://scratch.mit.edu/projects/30367666/>)



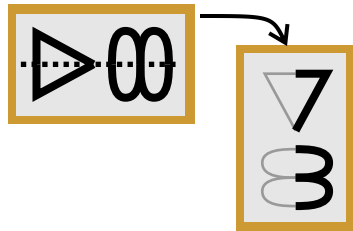
1. Books

Let's start with the book puzzle. The secret code for the books puzzle is 23. The reason is that there are two books lying sideways and three books standing up. Write a program for the books so that when they are clicked on, it will ask you to type in the secret code. If the answer is correct, you should hide the books.



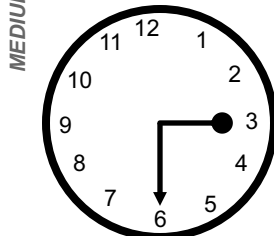
When the green flag is clicked to start the game, you should make sure that the books sprite is showing or people will see the secret code underneath.

2. Painting



When you draw a line through the middle of the painting, you will see the numbers 7 and 3. The secret code for the painting is 73. Write a program for the painting that will ask you for a code if you click on it. If you get the right answer, the painting should disappear.

3. Clock



The hands of the clock are pointing at the numbers 3 and 6. The secret code for the clock puzzle is 36. Write the program for the clock puzzle.

4. Door

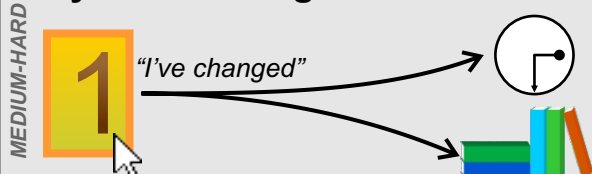
When you solve the books, painting, and clock puzzle, you will find the three secret numbers 8, 5, and 1. Write a program for the door that will ask you for the secret code if you click on it. If you get the right answer, the door should disappear.

Try It: Combination Lock

Instead of typing in the secret codes, let's use the combination locks to enter in the code instead. Click on one of the sprites for the combinations, and then click on the costumes tab. There should be one costume for each number. Write a program so that when the combination is clicked on, it should change to the next costume. When the green flag is clicked to start the game, it should start with the costume for the zero number.

Does the number on the combination change properly now? If so, copy your program to the other numbers of the combination lock. Don't forget that you can use the "Backpack" at the bottom of the screen to copy code between sprites.

Try It: Checking the Combination



Now you need a way to check if the combinations are correct. First, the books, painting, clock, and door need to know when the combinations are being changed. Change the programs for combination numbers so that they broadcast a message when they are changed. A broadcast sends a message to all the other sprites telling them that something has happened. Remember that you can use the Backpack to copy code.

Now, you can check the combination codes. Change the programs for the sprites for each puzzle. When one of them receives a message that the combinations have changed, it should check if the combination code is correct. If it is correct, the sprite should hide itself.

More Ideas

Try changing the puzzle sprites to have different answers. Or try making entirely new puzzles for the game!

Useful blocks for this exercise

Events

when  clicked

when this sprite is clicked

when I receive

broadcast

Looks

switch costume to

next costume

Sensing

ask and wait

costume # of

answer

Operators

and

=

Control

if then

HARD

Moon Landing

<http://scratch.mit.edu/projects/30428484/#editor>

Can you land the moon lander in this simulation of the low gravity of the moon? Your space ship is fragile. If you touch the rocks or if you land too fast, your ship will fall apart.

When you open this program, you will see a space background and one sprite: the moon lander. In this program, you will write code for simulating what it is like to move in space and what gravity on the moon is like.

(Finished game: <http://scratch.mit.edu/projects/30358794/>)



1. Basic Movement

MEDIUM

Start by writing a program that let's you move the space ship in a simple way. Some code has already been written for moving the ship to the left and right. The spaceship uses "x" to show where it is to the left and right. It uses "y" to show where it is up and down. Finish writing the code for moving the ship up and down.

2. Crashing

MEDIUM

Click on the costumes tab of the lunar lander sprite. Look at the different costumes for the sprite. Change your program so that if the ship touches the rocks it will explode. Since all the rocks are grey, your program should check if the ship is touching something grey, and if it is, it should switch to the "explosion" costume and stop the program. Also, when the green flag is pressed to start the game, you should switch to the starting "lunar_lander" costume.

3. Space Movement

HARD

In space, there is no air. That means that when you move in space, there is no air to get in your way and slow you down. If you start moving in one direction, you will keep moving in that direction. You can use variables to show what it's like to move in space. Let's start with moving left and right.

Create a variable called "x speed" for the sprite that keeps track of how fast the ship is moving left or right. During the game, the "x" position of the ship should be regularly changed by the "x speed" that the ship is moving at. This way, if you start moving at a speed of 1 step to the right, the ship will keep moving 1 step to the right again and again.

In space, when you use thrusters, your ship keeps moving even after the thrusters stop. Change your program so that when you press the left and right arrow keys, the "x speed" changes instead of directly moving the ship. That way, the ship will keep moving even when your thrusters are off, just like in real life.

4. Up and Down

HARD

Now that you've written code to move the space ship left and right using space movement, do the same for moving up and down. Use a variable called "y speed" to change the "y" position of the ship.

5. Moon Gravity

HARD

The moon has very weak gravity that pulls things to the ground. Gravity makes things fall to the ground faster and faster until they hit something. Add gravity to your program by having the "y speed" of your ship slowly change by -0.1.

6. Landing

HARD

Change your program so that if the lander touches the yellow landing pad, the game ends.

If you land on the landing pad while moving too fast, the ship will fall apart. Add code to your program so that if your speed is too fast when you touch the landing pad, your ship will explode.

Try It: Fuel

HARD

Space ships have only a small amount of fuel. Astronauts need to land their ship before running out of fuel. Make a variable called "fuel" for your program. When your game starts, your ship should start with 100 fuel. Every time you use your thrusters, you should lose fuel. If you run out of fuel, then your thrusters shouldn't work any more.

Try It: Thrusters

MEDIUM-HARD

The sprite for the spaceship has different costumes showing different thrusters being used. Change your program so that when you press the arrow keys, the correct thruster is shown.

Try It: New Maps

MEDIUM

If you click on the stage, you can go to the backdrops tab to change the terrain on the moon. Can you make your own moon surface?

Useful blocks for this exercise

Motion

move steps

turn  degrees

go to x: y:

change x by

change y by

Looks

switch costume to

Data

set speed to

change speed by

Sensing

touching color

key pressed?

Events

when  clicked

Control

wait secs

forever 

if then 

stop all

Operators

and

>