Coyote Coding Camp: Dunk Contest

Part 1: Take off! Code a jump.

Overview

The Spurs Coyote is a lovable mascot who loves to dunk. In this activity, you will use Scratch code to make a dunk contest game and get to play as The Spurs Coyote! This first part will be focused on making the Coyote jump and return to an initial position.

Materials

- Computer, Chromebook, or tablet with keyboard
- Internet connection
- Scratch Account

Let's Get Started

Join Scratch by signing up for an account so that you can easily save your work and also share it with us when you're finished. Please make sure you have permission from an adult caregiver.

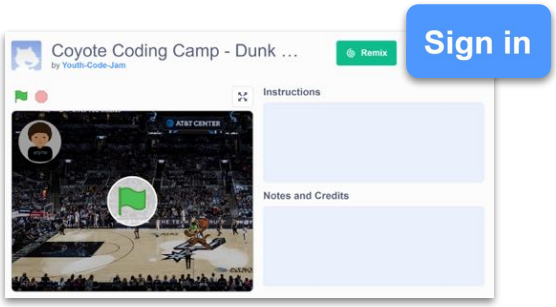
Guide for joining scratch:
https://spursgive.org/scratch-account-setup_scratch-offline-english-1

Project Setup

1

Open a web browser and type **https://bit.ly/SGdunkContest** into the address bar. Make sure you're signed in! If you don't see the green remix button, sign in to your Scratch account using the Sign in button in the top-right of the page.

https://bit.ly/SGdunkContest



2

Once you are signed in, you will see the green **remix button**. Click on that to make a copy of the starter project on your account.

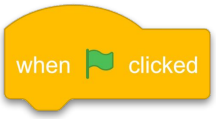
SPURS GIVE × YOUTH CODE JAM TX

Let's Code!

1 Set up starting values.

a Click on the thumbnail of the Coyote in the sprites pane to make sure you are editing code for the correct sprite.

Start by adding a **when green flag clicked** event block to the workspace.

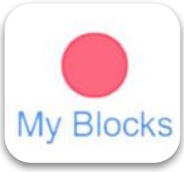


b Add some code to set up starting values:

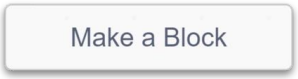
- **Set points to (0)** - Add a set variable block to start the game off with 0 points. Use the dropdown arrow to select **points** from the list of variables.
- **Go to x: (130) y: (-65)** - go to a starting position. Make sure the x and y values match the values you see here.
- **Point in direction (90)** - start off facing the correct direction.

2 Make a block to define what the Coyote will do when jumping.

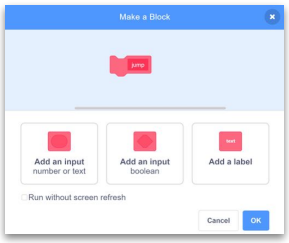
a Click on the **My Blocks** palette.



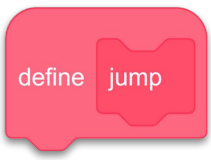
b Click on the **Make a Block** button.



c **Name** your block. Something descriptive like 'jump' works best. Press OK when you're done naming.



d You should now have a **define jump** block on your workspace. Next, we'll attach a procedure of actions for the Coyote that will happen (in order!) when you use the jump block in your game.

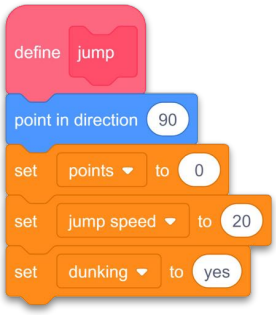


3 Make the procedure for the jump.

a

Attach some blocks to set up the jump:

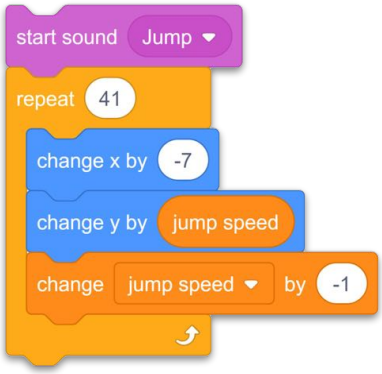
- **Point in direction (90)** - start off facing the correct direction.
- **Set points to (0)** - each dunk will start with 0 points.
- **Set jump speed to (20)** - stores a changing number that will
  decide vertical movement.
- **Set dunking to (yes)** - puts the Coyote into a state of dunking.



b

Attach some blocks to the end of the previous step to make the actual jump happen:

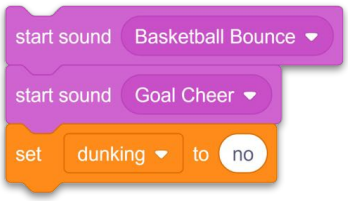
- **Start sound (Jump)** - boing!
- **Repeat (41)** - repeat some code.
- **Change x by (-7)** - horizontal movement (don't forget the minus!).
- **Change y by (jump speed)** - vertical movement (add the jump speed variable instead of a number).
- **Change jump speed by (-1)** - this block simulates gravity!



c

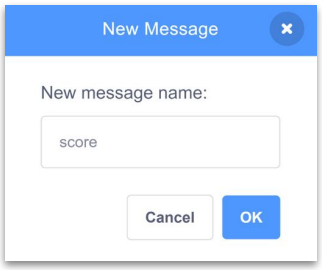
Attach some blocks to the end of the previous step to end the jump:

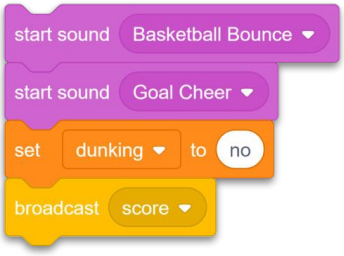
- **Start sound (Basketball Bounce)**
- **Start sound (Goal Cheer)** - woo hoo!
- **Set dunking to (no)** - exit the dunking state.



d

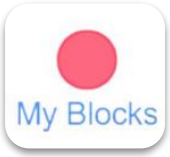
Attach a **broadcast message** block to the end of the code from the previous step. Use the dropdown arrow to select 'New message.' Type the word **score** and click 'OK.'



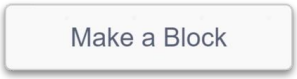


4 Make a block to define what the Coyote will do when returning to the original position.

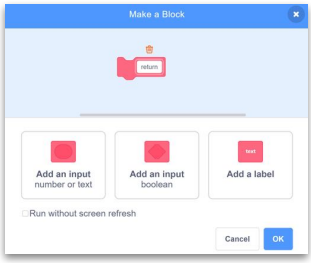
a Click on the **My Blocks** palette.



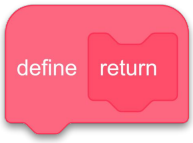
b Click on the **Make a Block** button.



c **Name** your block. Something descriptive like 'return' works best.Press OK when you're done naming.



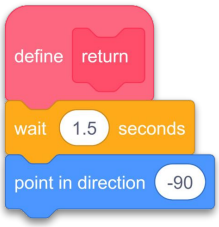
d You should now have a **define return** block on your workspace. Next, we'll attach a procedure of actions for the Coyote that will happen (in order!) when you use the return block in your game.



5 Make the procedure to return to the original position.

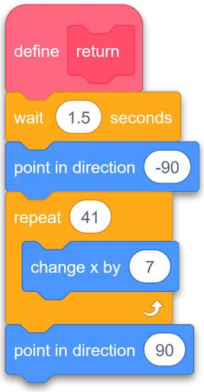
a Add some blocks to start the return:

- **Wait (1.5) seconds** - give the Coyote a short break.
- **Point in direction (-90)** - start off facing the correct direction.

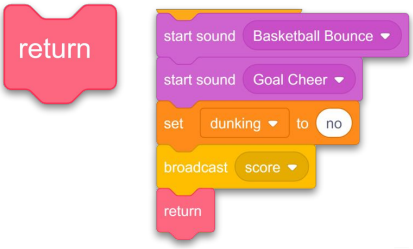


b Attach some blocks to make the Coyote move:

- **Repeat (41)** - repeat some code.
- **Change x by (7)** - horizontal movement.
- **Point in direction (90)** - end facing the correct direction.



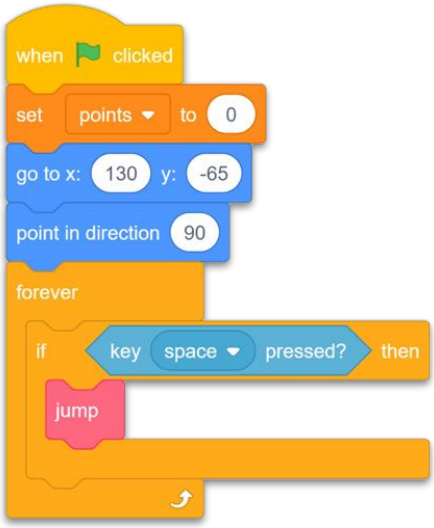
c Attach the **return** block from the My Blocks palette to the end of the code you wrote in step 3 (after the broadcast block).



6 Put it all together!

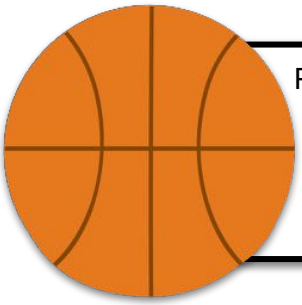
Attach some code to the code you wrote in step 1 to make your jump and return happen when you press a key on the keyboard:

- **Forever** - repeat some code while the program is running.
- **If < > then** - ask the computer a question. If the answer is true, then run the code contained in the if block.
- **< key space pressed >** - *senses* if the spacebar is being pressed. Returns a true or false value.
- **Jump** - "call" the code that lives in the jump block that you created. You can find this block in the My Blocks palette.



7

Give yourself a pat on the back! You should now have a working jump when you press the spacebar (don't forget to click the green flag to start the game). In the next parts we will add some tricks and scoring.



Post a screen capture of your project, or a picture of you coding and use **#spursgivecoding**

SPURS GIVE × YOUTH CODE JAM TX