MOONHACK 2024

Sweeten the Crop

Can you grow the perfect sugarcane crop? Create a clicker game to give it your best try!

INTRODUCTION

What you will need

HARDWARE

A computer capable of running Scratch 3

SOFTWARE

Scratch 3:
either online
rpf.io/scratchon
or offline
rpf.io/scratchoff

What you will learn

- How to use **variables** to control a game
- How to use **branching statements**
- How to use **broadcast**

Additional notes for educators

Check out our blog post that talks about the science behind the project, integration links for teachers, and where to find more information.

Project Links

Starter project

https://scratch.mit.edu/projects/1046696958/

Completed project

https://scratch.mit.edu/projects/1047236490/

Moonhack Certificate

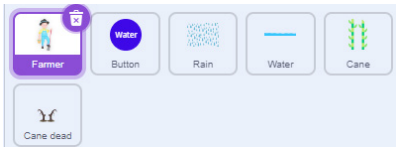
Finished the project? Download a certificate of completion here.

CANEGROWERS

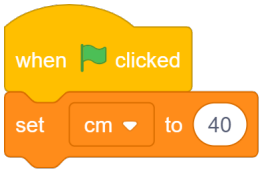
STEP 1 - SETTING UP THE VARIABLES

Start by opening the starter project - https://scratch.mit.edu/projects/1017283262. The variables in this project control how the game plays out. Let's get them set up.

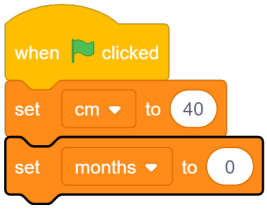
Select the farmer sprite



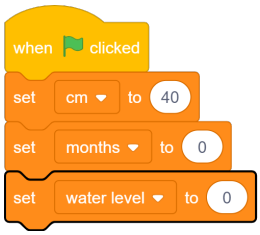
Start with the event when the green flag is clicked. Create a new variable called cm. This will be used to measure the height of the sugarcane as it grows. Start the height at 40cm.



Add another new variable called months. Sugarcane takes 12 months to grow and we need to track this in the game.

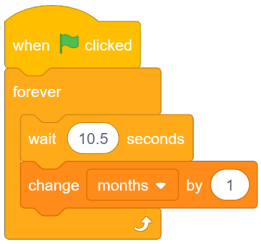


Add one more variable called water level. This is what will control most of the actions in the game, and what the player will interact with the most.



Set the conditions for the water level to always drop. This simulates the plants absorbing the water through the soil. Add a forever block. Inside add a wait 5 seconds block and then from the variables menu, change water level by -1.

We need the months to slowly change. Start a new algorithm with the green flag event. Add a forever block. Inside add a wait block, and then a change variables block. This will simulate the year passing.

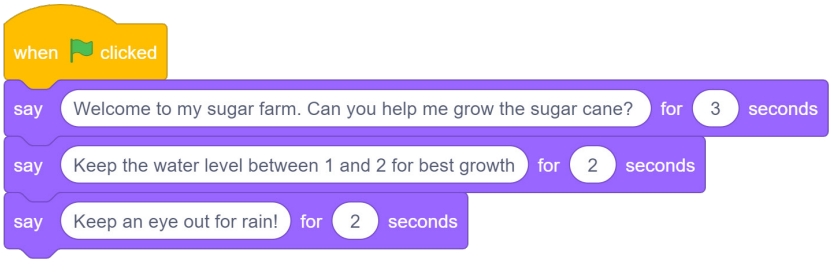


CANEGROWERS

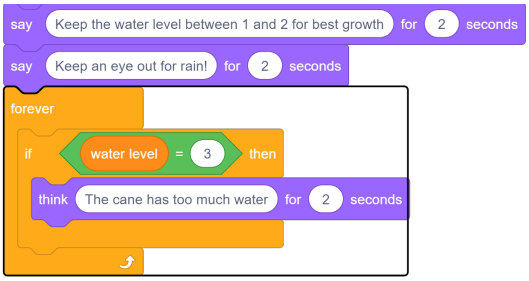
STEP 2 - INTRODUCING THE GAME

Let's get our sugarcane grower to introduce the game and give some information about how to play. Then we will set up some conditions of play.

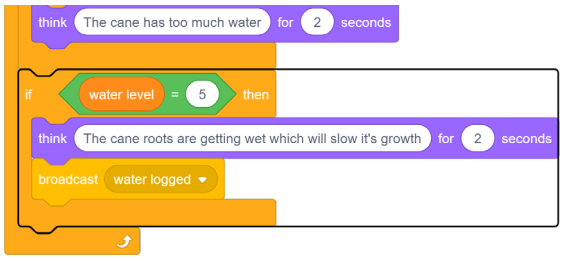
Start a new algorithm with the event when green flag is clicked. Add 3 say blocks. The first block will introduce the game. The second one will tell them how to play. The third one gives them a game hint.



Add a forever block. Our first condition is if the water level makes it to 3 the sugarcane has too much water. Add an if-then control block. Place in an equals operator with the water level variable on one side. Add a think block.

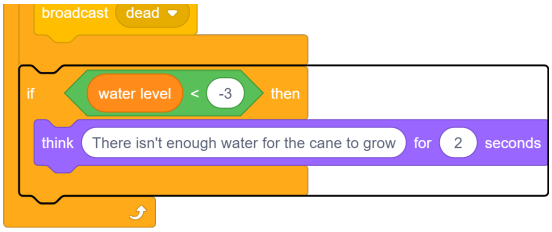


The second condition is if the water gets to 5 the sugarcane is water logged and will possibly die. Duplicate the previous if-then block. Change the water level to equal 5. Change the farmer's thinking. Then create a new broadcast. This will be used by the cane sprite.

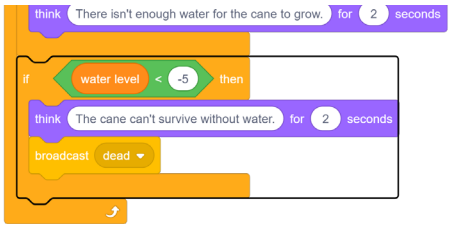


The next condition is when too much water kills the sugarcane. Duplicate the previous condition. Swap the equals operator for a larger than. Change the thought. Also create a new broadcast called dead.

The next condition is when there is not enough water. Duplicate the previous condition. Swap the larger than operator for a smaller than. Change the thought.



Create one last condition when there is no water and the sugarcane dies. Add the broadcast of dead to communicate with the sugarcane sprite.

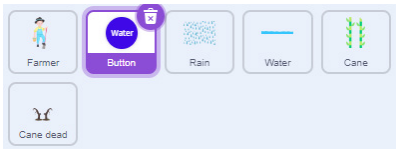


CANEGROWERS

STEP 3 - CLICKER

The clicker in this project is the water button. The player uses this to imitate watering the sugarcane when it is needed, after using soil moisture tests and weather data.

Select the button sprite



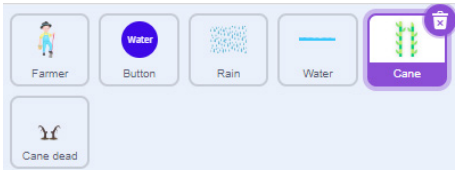
Start with the event when this sprite clicked. Create a new broadcast called water that will be used by the cane sprite. Add a variables block to change water by 1

Test your code. Your months variable should slowly change. When you click the water button it should increase your water level. If you wait the water should decrease.

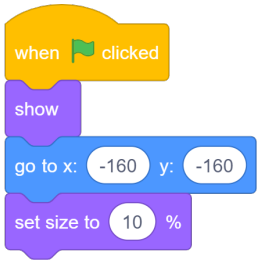
STEP 4 - THE SUGARCANE

The sugarcane is the hero of our game. Program the sugarcane so it knows how to grow and how to respond to some of the broadcasts that have been created.

Select the cane sprite.

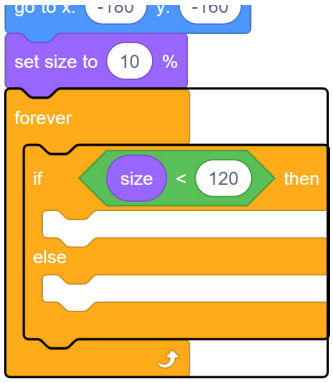


Start with the event when green flag is clicked. Add a show block so the sugarcane is on screen. Set its starting location with a motion block. And then add a set size block so it starts as a small plant.

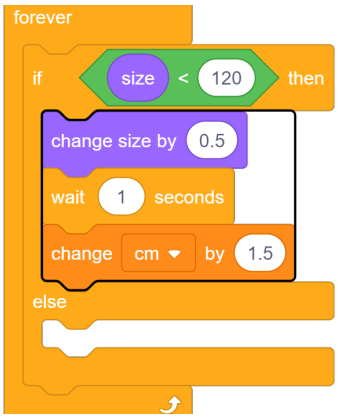


CANEGROWERS

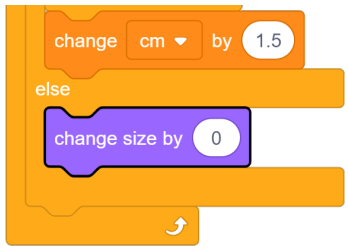
The sugarcane will be coded to continually grow, unless one of the conditions takes place (we made these on the farmer sprite). Add a forever control. Inside this place an if-then-else block. Create the condition if the size is less than 120.



120 is the maximum size for the sugarcane to grow. Until it reaches this we want the sugarcane to change size by 0.5. Then wait 1 second. Add a variable block to change cm by 1.5. The code here will continue until the size of the cane sprite reaches 120.

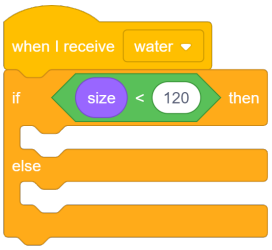


Once the sprite reaches 120% in size we want it to stop growing - this represents the plant's maturity. Add a change size block in the else section.

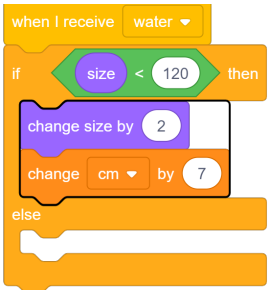


Test your game. Your cane sprite should slowly grow. You will see the sprite get bigger. Let's code the sugarcane to respond to the water button being pressed, as if the sugarcane grower is watering the crop.

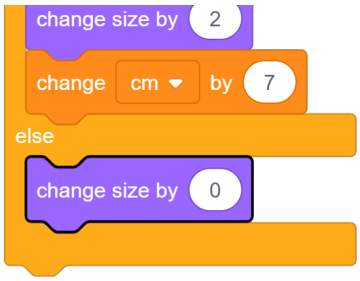
Start a new algorithm when I receive water. The code can be duplicated and adapted from the previous algorithm. Add an if-then-else block if the size is less than 120.



If the sprite is less than 120, add a block to change size by 2. Watering the plant will increase it's growth. Change the variable cm by 7 if the button is clicked.

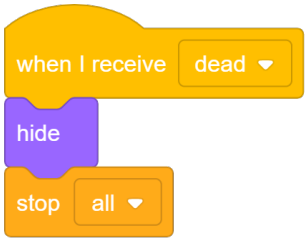


In the else section add a change size by 0 block. This will stop the sprite changing in size if it is over 120 in size.



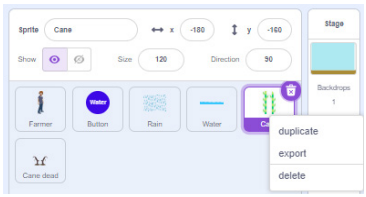
Test your game again, now with the button working. When you click the water button you should notice the sugarcane grow and the cm size jump quickly. Do the conditions start taking place if your water goes above 5? Let's add in what the sugarcane does if it dies, either from too much or too little water.

Start a new algorithm with the event when I receive dead. Add a hide block and a control of stop all. This will stop all scripts from running.

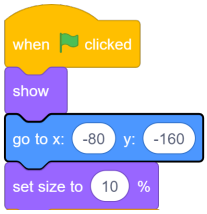


Now that we have the main code, let's 'plant' more sugarcane. There should be five plants in total.

Duplicate the cane sprite. Hold your mouse over the sprite and right click. elect duplicate.

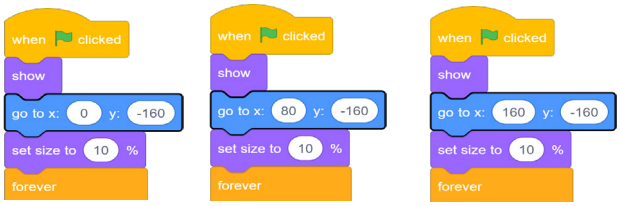


This duplicated sprite will have all of the same algorithms as the original sprite. The only input we need to change is the starting location.



Delete the change cm blocks from **both** algorithms.

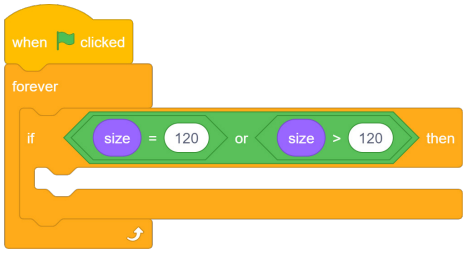
Make 3 more duplicates of the cane sprite so that you have 5 in total. Change the x coordinate by 80 each time.



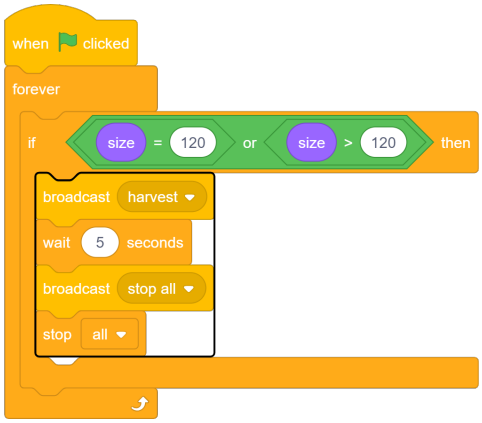
If you run the game now you should have 5 sugarcane plants that each grow together. The next step is to program the sugarcane to react to reaching harvesting size. We only need this code on one sugarcane sprite.

Select the first cane sprite.

To get to harvest we need the cane sprite to reach 120% in size. Start an algorithm with the green flag event. Add a forever block. Set the conditions if the size is equal to or more than 120.

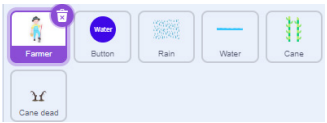


Add a new broadcast of harvest. Add a wait block. Add another broadcast of stop all then the control of stop all. The harvest broadcast will be used by our farmer. The stop all broadcast will be used by all sprites.

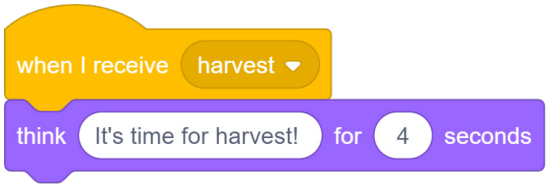


We need the sugarcane grower to take action when the broadcast of harvest is made.

Select the farmer sprite



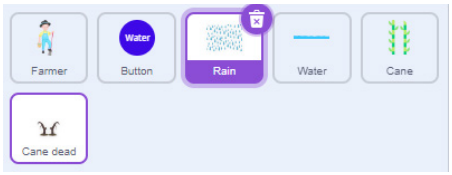
Start a new algorithm when I receive harvest. Add a think block.



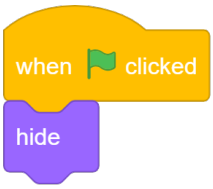
We also want to see all of the sprites respond to the stop all broadcast. Add this algorithm to any sprite that will need to stop acting at the end of the game. This should be the farmer sprite, button sprite and the rain sprite.

STEP 5 - RAIN

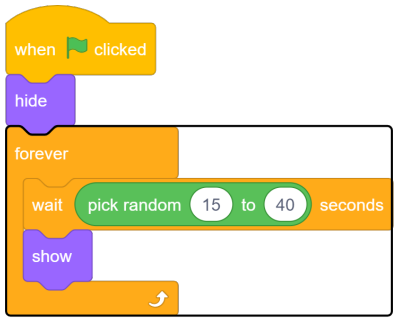
Select the rain sprite. This sprite covers the whole screen. It will show at random intervals and effect the water level.



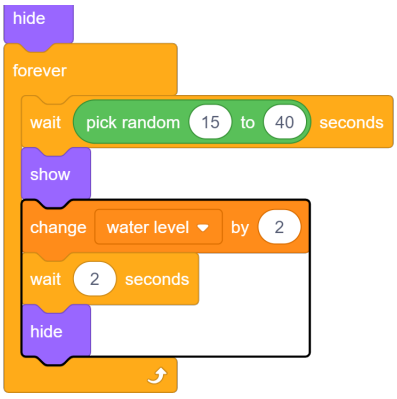
Start with the flag event. Add a hide block.



Add a forever block. Add a wait control block with a pick random operator. This will allow a random time for the rain to appear. Add a show block.



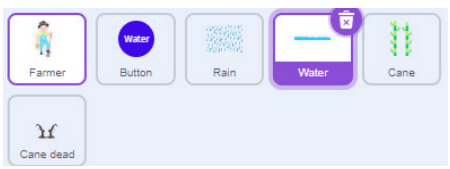
Add a change water level block. The rain brings a lot of water. Add a wait control block and finish with a hide block, as if the rain has stopped.



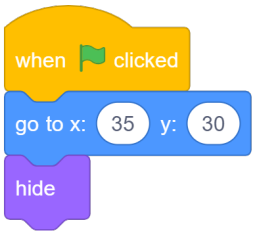
If you test your game now you will have the rain show at random times and it will impact the water level, perhaps even triggering the sugarcane grower to react.

STEP 6 - TOO MUCH WATER

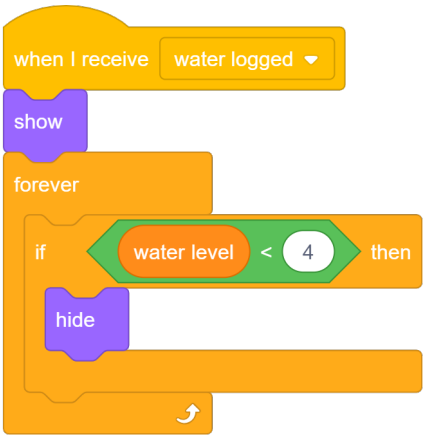
Select the water sprite. This sprite will show when there is too much water in the soil.



The sprite needs to be hidden at the beginning. Start with the event when flag clicked. Add a motion block to set its starting position and a hide block.



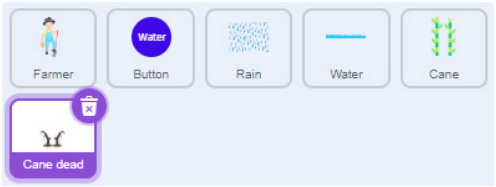
The sprite needs to respond to the water-logged broadcast. Start with the event water-logged and add a show block. When the water level is less than 4 we need the sprite to hide, as this means the water is receding. Add a forever control. If the water level variable is less than 4 then it needs to hide.



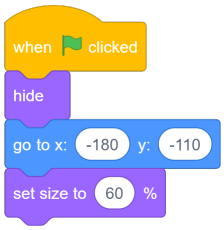
Run your game and press the water block to force the water level up high. Does the water sprite respond?

STEP 7 - DEAD SUGARCANE

Select the dead cane sprite. When the sugarcane is deprived of water (the level drops to -5) the ground is in drought and the sugarcane dies.



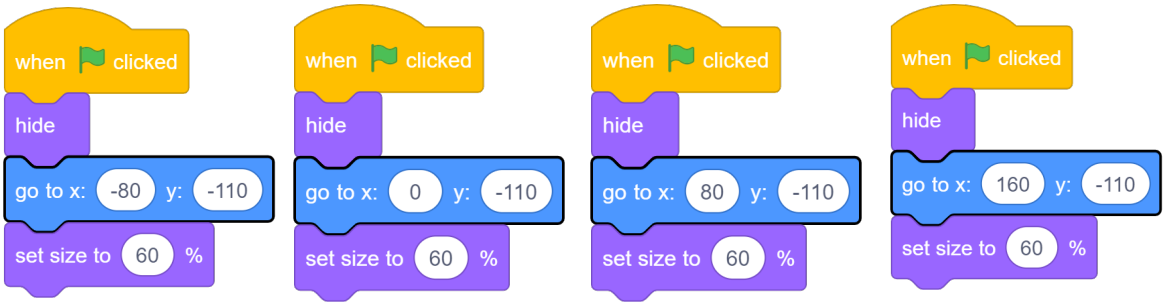
Start with the green flag event and a hide block. Add a motion block to set its starting location and a looks block to set the size to 60%.



Start an algorithm with the event when I receive dead. Add a show block.

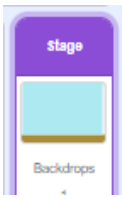
CANEGROWERS

Now duplicate this sprite 4 times, so you have 5 dead cane sprites. The only thing to change on each one is the starting position.

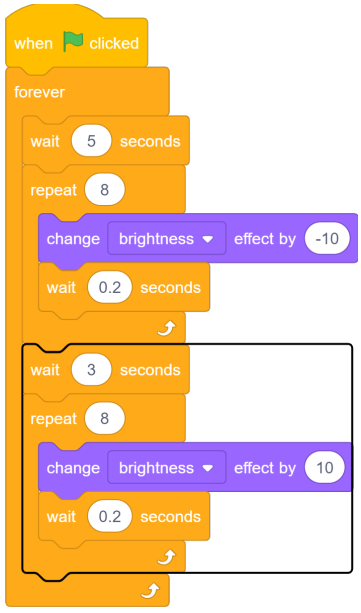


STEP 8 - DAY AND NIGHT (OPTIONAL)

Select the stage. Let's code the background to get brighter, and then darker, representing day and night.



Start with the flag event and a forever block. The day starts with a 5 second wait. Add in a repeat block that changes the brightness effect by -10 (getting darker) and a short wait in-between. Repeat the previous blocks, this time changing the brightness by 10 (getting lighter)



We also want the day and night to finish when the sugarcane is done. Add in when I receive stop all to stop all scripts.

Your game is now complete! Can you grow a successful harvest?

Challenges:

Next steps

What else can you do to adapt the game? Is there a next level? Perhaps the cane beetle comes in and attacks the crop and a solution is needed.

Information

This game reflects the way that sugarcane growers use weather knowledge and soil moisture data to control the amount of water that they use on their crops. How can you add information to the project so that the player learns this information along the way?

Harvesting the crop

After 12 months the crop is grown and ready for harvest. Can you add harvesting to the game? What would this look like?

Congratulations! You're a Moonhack changemaker. You can try one of our other projects or try one of the challenges above.

Don't forget to talk to an adult about registering your participation and downloading your certificate at moonhack.com