

# BEAVER ACHIEVER

Hour of Code

Block-Based  
Programming



Copyright © 2020 by CodeMonkey Studios Ltd. All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

2345 Yale St., 1st floor  
Palo Alto, CA 94306  
info@codemonkey.com  
www.codemonkey.com



### Table of Contents

Introduction	4
Standards	5
Dam Builder Challenges	6
House Builder Challenges	17
Smoothie Party Challenges	21
Conclusion	28
Solutions	29
Reference Card	32

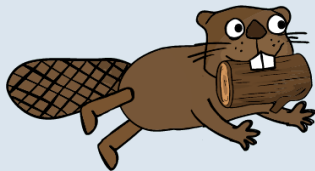
### Introduction

Welcome to Beaver Achiever! In this Hour of Code experience your students will use block-based coding to help the beaver fix different sections of a dam, finishing building a house, and even throw a smoothie party for its friends. In the process they will be introduced to a variety of programming concepts including sequencing, loops, and conditionals.





While the Hour of Code is designed to be self-paced, these notes will allow you to conduct the lesson in an more structured way if you wish or to simply coach and support the students as they work through the activity on their own

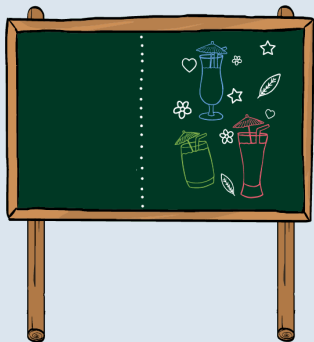
Please email us at [info@codemonkey.com](mailto:info@codemonkey.com) for any questions you may have along the way.

Have fun!  
The CodeMonkey Team







### CSTA Standards

-  Develop programs with sequences and simple loops, to express ideas or address a problem (CSTA.1A.AP-10)
-  Develop plans that describe a program's sequence of events, goals, and expected outcomes (CSTA.1A.AP-12)
-  Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops (CSTA.1A.AP-14)
-  Using correct terminology, describe steps taken and choices made during the iterative process of program development (CSTA.1A.AP-15)






### Challenge 1

-  This challenge introduces the goal of the first part of the game: helping the beaver complete the dam.
-  The challenge also introduces the block-based interface that is used to give instructions to the beaver.
-  As shown by the on-screen prompts, students will need to drag the drop block so it attaches below the when play block.
-  Once the two blocks are attached, the students need to click the play button to tell the beaver to go to work.



### Challenge 2

-  Have students start this challenge by trying the same solution that worked for challenge 1 – a single **drop** block attached to **when play**.
-  Of course that does not solve the challenge. Reassure students that it's okay to make mistakes. Sometimes we will even make mistakes on purpose to help us learn.
-  Reset the challenge and ask students what is different about this challenge compared to the last one.



### Challenge 2 (continued)








Challenge 1





Challenge 2

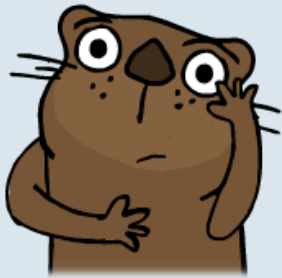
### Challenge 2 (continued)

-  If a student points out that the beaver is in a different position affirm that answer and ask what else is different.
-  Once a student has pointed out that challenge 2 has two missing logs ask the students what we need to tell the beaver to do to completely fix the dam.
-  If the students have trouble seeing the solution, you may need to show them that they can drag and attach additional  blocks to the stack.
-  Once everyone has solved it and enjoyed the animation, have them click the play button again and this time notice how each block is highlighted as the beaver carries out the instruction.






### Challenge 2 (continued)

-  Finish the challenge get students to try attaching a third drop block to the stack. Before they click the play button ask them what they think will happen.
-  The extra log causes the challenge to fail. Again emphasize that failing a challenge is not a bad thing. We have learned that the beaver needs to use the exact right number of logs to fix the dam.







### Challenge 3

-  In this challenge students will have to move the beaver to the left instead on the right and by two steps instead of one.
-  Students should try to complete this challenge on their own. If they have trouble, encourage them to add blocks one by one, clicking play each time to see what happens.
-  They can also use the “Need a Hint” button that displays if they fail to complete the challenge.






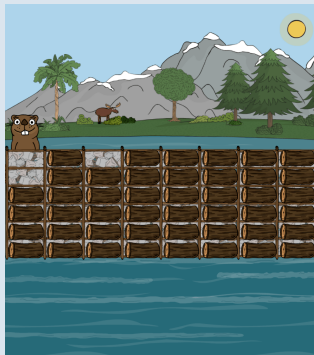
### Challenge 4

-  The tricky part of this challenge is that the beaver must move left and right.
-  Start with a `move ←` block. Play to show students where this leaves the beaver. Then ask students what blocks need to be added to complete the challenge.
-  After the class has solved the challenge once, ask if anyone can think of another way to do it.
-  If they need a hint, ask what would happen if we started with a `move →` block.




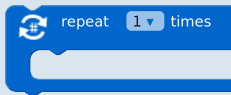
### Challenge 5


-  This is a challenge that students can attempt on their own.
-  Based on the previous challenge, some students may think that the beaver needs to move both right and left. Remind them that they do not have to use every block type every time.
-  Either before or after students complete the challenge, ask them to see what happens if the beaver moves to the left from the left edge of the dam.



### Challenge 6




-  This challenge introduces a block that allows us to repeat the same step or steps a certain number of times.



-  The block(s) for the steps to be repeated fit inside the C-shaped repeat.







### Challenge 6 (continued)

-  The on-screen hints will walk students through the solution: first attach a `repeat` to `when play`, then fit a drop inside it.
-  Tell students to watch how the blocks are highlighted when the program runs. They should notice that the `drop` block gets highlighted three times, alternating (very quickly) with the `repeat` block.
-  Explain that in coding an instruction that causes the same steps to repeat over and over again is called a **loop**







### Challenge 7

-  Start this challenge by asking students how many times the beaver needs to drop a log to complete this dam.
-  Since there is only one log to drop, it doesn't make sense to put a `drop` block inside a `repeat`.
-  Ask students what action does the beaver need to do multiple times to solve this challenge.
-  Remind students that since the drop does not need to be repeated, it attaches below the `repeat`, not inside it.









### Challenge 8

-  This is a new game with a new goal.
-  The beaver is trying to finish building a house.
-  In each challenge the objective will be to add one or more of planks to the back wall.
-  The planks come in three sizes. Point out that this time the missing plank is a large one, as indicated by the greenish-yellow outline.








### Challenge 8 (continued)

-  Let students follow the prompt to add the three blocks needed to solve this challenge.
-  Review the solution with students. They should understand that to complete the task the beaver has to  a piece of uncut wood and  it multiple times until it fits.
-  Before leaving this challenge, have students see what happens with just one  block, and then with three.







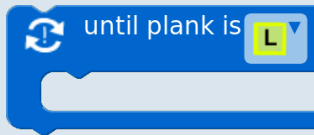
### Challenge 9

-  In this challenge students will learn about the `until plank is` block.
-  Let them follow the prompts to create the solution.
-  First comes a `pick up` block, then an `until plank is` block.
-  Since `until plank is` is C-shaped, we can put other blocks inside of it.
-  Ask students what action the beaver needs to do repeatedly until the board is large sized.






### Challenge 9 (continued)

-  Once students have figured out that `chop` goes inside of `until plank is`, let them run and test the solution.
-  Explain that `until plank is` is a special kind of block that causes the blocks inside them to repeat. Remind them that such a block is called a **loop**.
-  Unlike a `repeat` block, `until plank is` large doesn't repeat the blocks inside a certain number of times.
-  Instead it repeats the steps *until* the plank is the right length to fit in a large space.







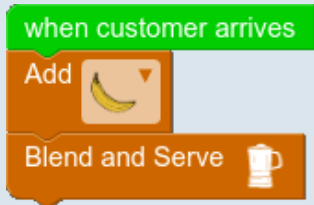
### Challenge 10

-  This is another new game.
-  The object of the game is to prepare smoothies for the beaver's animal friends using their favorite ingredients.
-  The board tells us that in this challenge the only visitor will be a monkey who likes a smoothie with bananas.







### Challenge 10 (continued)


-  As shown by the prompts, the students should drag the **add** block and attach it to the **when customer arrives** block.
-  Then drag the **blend and serve** block and attach it below the **add** block.
-  Now students can click on the play button and see a fun animation of the beaver serving a smoothie to the monkey.
-  The solution to every challenge in this game will involve first **add** ing the right items to the blender and will always end with **blend and serve**.



### Challenge II



-  This challenge is almost exactly like the previous one.
-  Point out to students that the chalkboard is different.
-  This time the beaver will be serving a rabbit who likes carrot smoothies.
-  Also ask students to notice that the **add** block already has the carrot selected.











-  Students should be able to follow the model of the previous challenge to complete this one. Provide help as needed.



### Challenge 12

-  In this challenge the beaver has to serve the monkey and the rabbit.
-  Ask students to try the three solutions below. What happens?

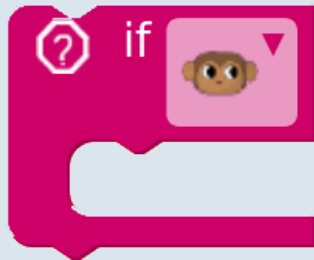
<p>when customer arrives</p> <p>Add </p> <p>Blend and Serve </p>	<p>when customer arrives</p> <p>Add </p> <p>Blend and Serve </p>	<p>when customer arrives</p> <p>Add </p> <p>Add </p> <p>Blend and Serve </p>
--	--	---

-  None of these makes every body happy. What can we do?







### Challenge 12 (continued)

- 🦫 The beaver needs to be able add different ingredients depending on which customer is there.
- 🦫 That's where the **if** block comes in.
- 🦫 **if** is a block that other blocks can fit inside. When the computer comes to an if block, it executes the blocks inside only if the condition answer to the block's **condition** is yes.
- 🦫 A condition is a question with a yes or no answer.
- 🦫 For the if block shown here, the condition is "Is the customer a monkey?"






### Challenge 12 (continued)

-  As shown by the on-screen hint, students should drag the **if** monkey block and attach it to the **when customer arrives** block.
-  Ask what should go inside the **if** block. What should the beaver do if the customer is a monkey?
-  That takes care of the monkey. What about if the customer is a rabbit? That requires a separate **if** rabbit block attached below – not inside – the **if** monkey.
-  Of course an **add** block with the rabbit's favorite food needs to go inside the **if** rabbit.



## Challenge 12 (continued)

-  Don't forget the **blend and serve** block. That goes after the second **if** block.
-  When you press play to start, make sure your students pay attention to how and when the blocks are highlighted.
-  When the customer is a monkey, the **if** rabbit block will highlight but not the **add** carrot that is inside it. That shows that the condition is "Is the customer a rabbit?" is being checked but because the answer is no the code inside the block is skipped.



### Conclusion

Thank you for playing the Beaver Achiever Hour of Code. We hope that you and your students had fun helping the beaver complete its chores and celebrate with friends.

This activity is only a sample of the full Beaver Achiever course. Check it out (along with our other courses and games) at [codemonkey.com](http://codemonkey.com).

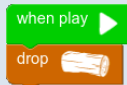
See you next time. Happy coding!

The CodeMonkey Team



### Solutions

Challenge 1



Challenge 2



Challenge 3



Challenge 4



### Solutions (continued)

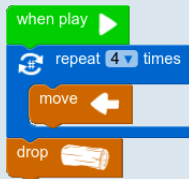
Challenge 5



Challenge 6



Challenge 7



Challenge 8

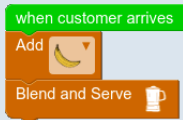


### Solutions (continued)

Challenge 9



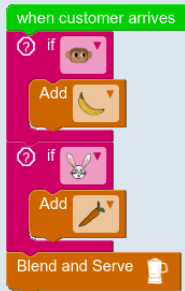
Challenge 10



Challenge 11



Challenge 12



## Reference Card



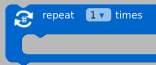
Tells the beaver to move one column to the right. Causes an error if the beaver is already on the right edge of the dam.



Tells the beaver to move one column to the left. Causes an error if the beaver is already on the left edge of the dam.



Tells the beaver to drop a log to fill in an empty space in the current column of the dam. Causes an error if the column is already full.



Tells the beaver to repeat all of the steps placed inside the block the number of times chosen on the dropdown.

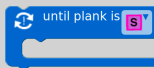
## Reference Card



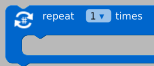
Tells the beaver to pick up an uncut piece of wood to prepare to cut it to the right size.



Tells the beaver to gnaw the piece of wood it is holding, making it shorter. It takes more chops to cut shorter planks.



Tells the beaver to repeat the steps inside the block until the plank is the length chosen on the dropdown.

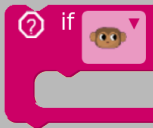


Tells the beaver to repeat all of the steps placed inside the block the number of times chosen on the dropdown.

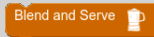
## Reference Card



Tells the beaver to add the ingredient chosen on the dropdown to the blender.



Tells the beaver to only execute the steps placed inside the block if the current customer is the animal chosen on the dropdown.



Tells the beaver to blend all the ingredients in the blender and serve the smoothie. Should always be the last block in the program.