

# CodeSpells: Spellcraft Lesson Plan

## Lesson Overview

Create and Experiment with Spells; Get Familiarized with Event-Driven Coding

### Lesson Summary

**DURATION:** 45-60 mins

#### Pre-Lesson Prep [teacher only]: (10-20 mins)

- [Setup computers with CodeSpells](#)

#### Getting Started: (15-20 mins)

- [Introduce the activity](#)
- [Introduce CodeSpells](#) *[Teachers: you might want to go over this by yourself before class]*
- [Using Other Blocks](#) (Optional)

#### Activity: (20-35 mins)

- [Have the students create one or two spells that work towards achieving the Goal](#)

#### Wrap-up: (10-15 mins)

- [Spell Showcase](#)

#### Assessment/Extended Learning: (2-5 mins)

- [Optional](#)

## Audience

This lesson plan is intended for use with learners who are already familiar with basic Computer Science concepts, such as Functions and Lists.

## Learning Objectives

By participating in this lesson, participants will:

- Become familiar with Event-Driven coding concepts

## Facilitation Guide

---

## Materials, Resources and Preparation

- A computer for each student or every two students
- A copy of CodeSpells installed on each of the computers
- (Optional) A projector for sharing student projects at the end of class



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

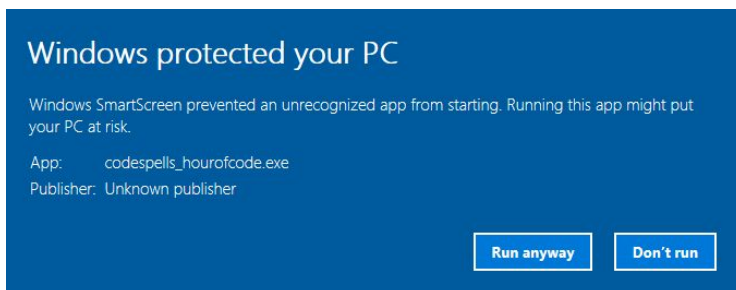
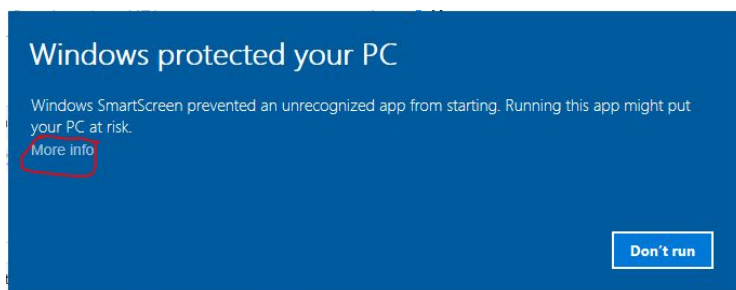
#HourofCode  
hourofcode.com

# CodeSpells: Spellcraft Lesson Plan

## Pre-Lesson Prep [teacher only]: (10-20 mins)

### Setup Computers with CodeSpells (10-20 minutes)

1. Download the Hour of Code version of CodeSpells found right [here](#). Make sure to do it on a Windows PC!
2. Click the downloaded file to access it, then double click on the new “CodeSpellsHourOfCode” folder and then again on “codespells\_hourofcode”. This will ask you what you want to do; click “Extract All” and wait for it to extract.
3. Once extracted, go through step 2 again with the new, extracted folder. This time, “codespells\_hourofcode” should have a nice visual icon that looks like a rock formation; if it does, you’ve done all the right steps! Double click on it, then click on “Run Anyway” if Windows asks you (you might have to click “More Info” in Windows 10 to see “Run Anyway”).



4. And you should be good to go!

## Getting Started (10-15 mins)

### Introduce the Activity

- Begin objectives of the class (1 minute)
- Give a crash-course in Event-Driven Programming (2 minutes)
- Use the flavor text below to help contextualize the lesson (1-2 minute)
- Explain the activity visually on a projector using the pictures below (1 minute)



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

#HourofCode  
hourofcode.com

# CodeSpells: Spellcraft Lesson Plan

## Introduce the activity (2-3 minutes)

Introduce the lesson by **explaining its context in Computer Science**, i.e. “Today we will be CodeSpells, a game where you create spells using code, to explore Event-Driven programming.”

Then **explain what exactly [Event-Driven programming](#) is**, i.e. “Event-Driven programming is the coding paradigm that focuses on user actions to dictate when code is run. This means, unlike popular languages like Python and Java, there are no ‘main’ functions that need to be run to trigger our code. Instead, events such as mouse clicks, button presses, and in-program actions govern when and how your code is run. In CodeSpells, this concept translates to spells, each of which can be designed using code and have three different states at which your code can be triggered.”

After introducing the concept of Event-Driven programming, **explain the goal of the class**. “Using Event-Driven programming and CodeSpells, we’re going try to solve a problem. So pick up your magic staff, don your cape, and get out your spellbook, because it’s time to pick up some garbage - wizard style! While I was clearing out this floating island for my new wizard castle, I foolishly let four large boulders roll into a ditch I dug. Now, I’d like this ditch to one day be the wonderful swimming pool I always intended it to be and not just a boulder-holding reservoir; can you possibly help me clear all the boulders out and fill the ditch with water, all with one or two spells? I swear I’ll give you a pat on the back for your troubles and....oh, I guess I could also throw in everlasting life!”

Finally, **visually show the students the arena and their goal**, either through the pictures included below or through your own computer, and reiterating what they need to do. Your students will be brainstorming and designing a spell that will remove the boulders from a ditch that is incorporated in the CodeSpells arena shared above and replace them with water. The ditch is located at the southernmost end:



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

# CodeSpells: Spellcraft Lesson Plan



And the four boulders inside of it look like this:





# CodeSpells: Spellcraft Lesson Plan

After casting your spell, you should aim to get it to look like this:



All with only one or two spells cast **total**.

## Introduce CodeSpells

- Introduce the class to the CodeSpells controls (1-2 minutes)
- Go over the CodeSpells basics (1 minutes)
- Create your first spell (5 minutes)
- Explain the Earth blocks (5 minutes)

## **Introduce CodeSpells(10-15 minutes)**

In front of you and your students should be CodeSpells. While the controls are relatively easy to wrap your head around, make sure to go over them so each student is on the same page.

### Controls (1-2 minutes)

W,A,S,D: Move your wizard  
Mouse: Move your vision/cursor  
Spacebar: Jump  
Left Mouse Button (Hold): Charge Spell  
Left Mouse Button (Release): Cast/Create Spell  
F: Activate Spell  
Tab: Open Spell wheel



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

# CodeSpells: Spellcraft Lesson Plan

Escape: Open Spell library/options menu

Before you start, have your students mess around with a few of the different, pre-built spells. They can do this by hitting the “tab” key and then clicking on the spell they want to equip.

## CodeSpell Basics (2-3 minutes)

Explain to the students that there are a few different types of spells they can make.

- You can make **Earth Spells** if you want to interact with the environment and throw objects around.
- You can make **Fire Spells** if you want to explode and destroy things.
- You can make **Water Spells** if you want to spawn water and create currents.
- And you can make **Air Spells** if you want to get information about your surroundings and create gusts of wind.

Of course, they can combine all four types together to make their own unique type of spell!

Next, explain to them that each spell they cast, no matter what its type, creates an **Orb** that, by default, **launches** out in a straight line in front of them. These Orbs are the main way their spells will be controlled!

## Creating Your First Spell (5 minutes)

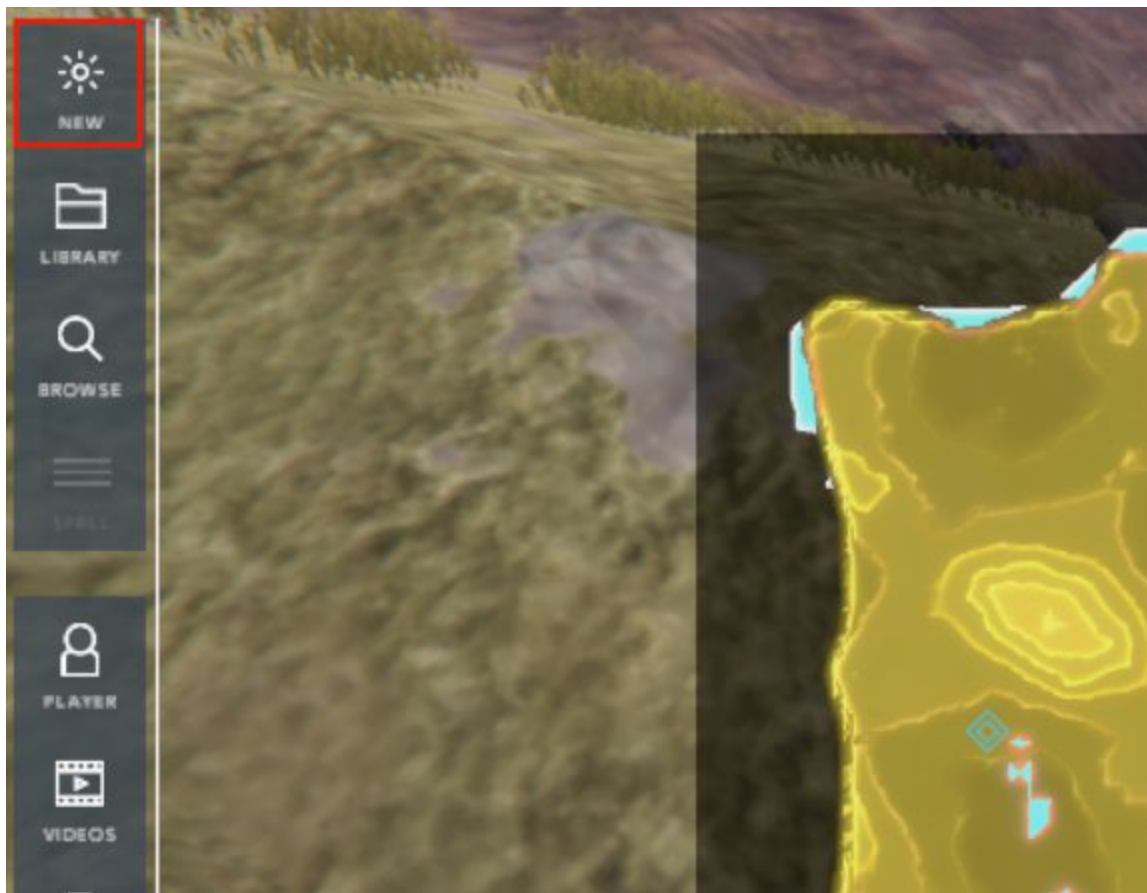
Now, how do we make a spell?



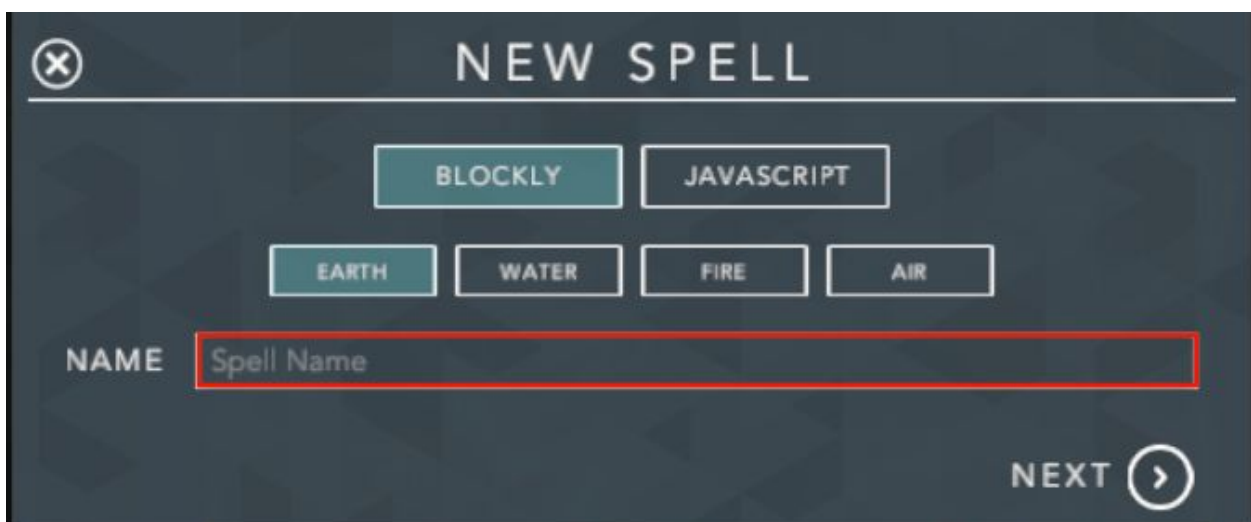
This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

# CodeSpells: Spellcraft Lesson Plan

Bring up the Spell interface by pressing the Escape key, then click “New”:



Here, you can name your Spell in the box here:



# CodeSpells: Spellcraft Lesson Plan

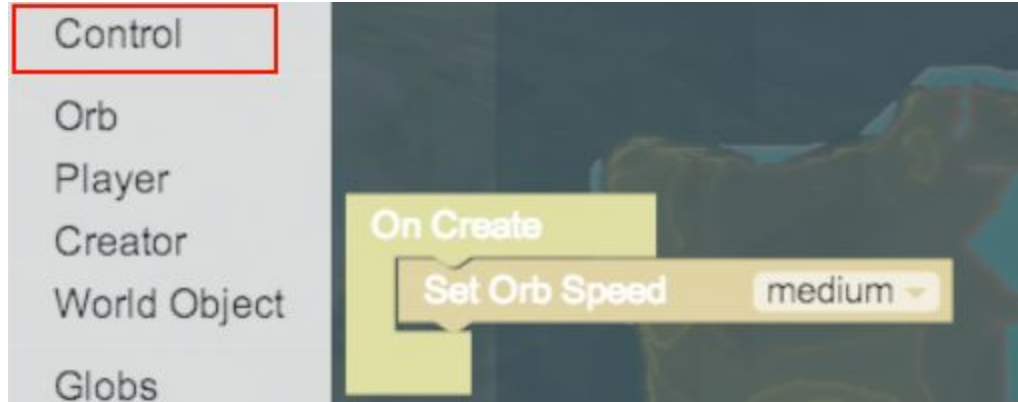
You can also select what type of spell you want to make. Don't worry; you can change it at any time once you've created it!

Now click the Next button and take a look at the new screen. At the top will be the name of your code on the left and its *mana cost* on the right:



The mana cost determines a few things, mostly how many high-cost actions you can add to your spell. Some blocks of code use “x number of y mana per second;” this will drain from your mana pool, so the more mana you have the longer it will run. You can change the values of your mana at any time by clicking on them, going as high as 200 or as low as 0.

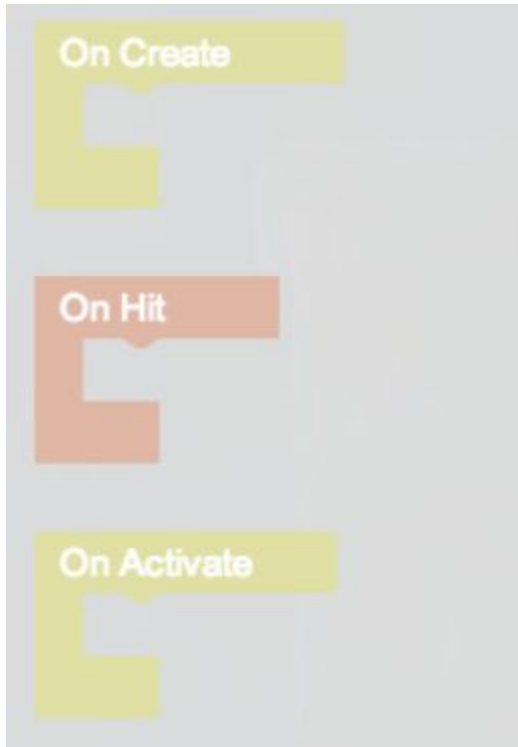
Below it CodeSpells will create some default code (that you can change at any time): an “OnCreate” block and an “OnHit” block, the first setting the Orb Speed to Medium and the second setting the speed to Stopped. “OnCreate” and “OnHit” are but two of a total three blocks that allow you to control what your spell does and when it does it; you can find the other one, “OnActivate”, by clicking on the “Control” tab. Explain that these are the **three triggering points in CodeSpells for Event-Driven programming**, as described earlier.





# CodeSpells: Spellcraft Lesson Plan

And:



Any code in “OnCreate” will be triggered *when your spell is launched*. For each spell, you’ll need to hold down on the left mouse button to charge it (the length of which depends on how much mana it costs) and release it by letting go. When you release the button and launch the spell, any of the code in “OnCreate” will trigger. Launching a spell will release an Orb, which is the physical object that then gets used in “OnHit” and “OnActivate”.

Any code in “OnHit” will be triggered *when your Orb collides with something*. This something can either be a static part of the environment, like the ground or a tree, or a dynamic “World Object”, which has physics and can be moved around using your spells.

Any code in “OnActivate” will be triggered *when you tap the “f” key after the spell has been launched*.

This allows you to run certain parts of your code multiple times, or activate a part of your code at a specific moment. All code activated will happen in the location of your Orb in physical space.

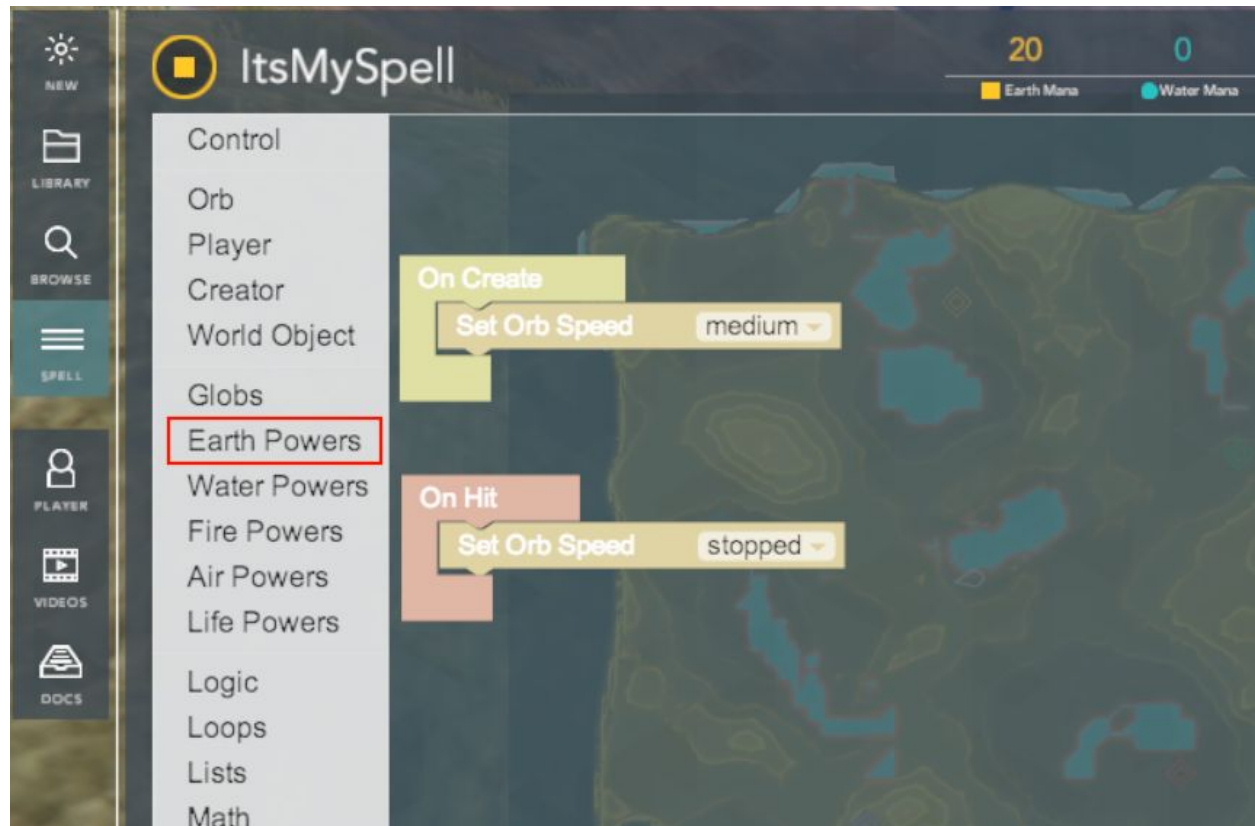
## Breaking Down the Earth Powers (5 minutes)

Now, with these in mind, run through the basic “Earth Power” blocks; these are the easiest, simplest, and most powerful blocks you can use.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

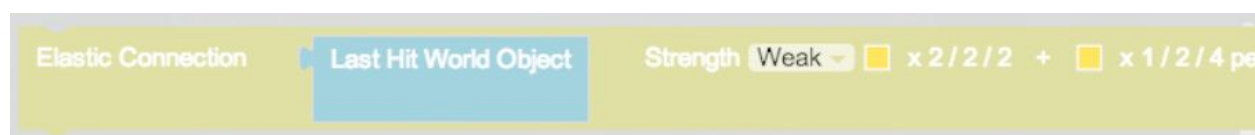
# CodeSpells: Spellcraft Lesson Plan



Looking inside, we've got a few choices. **Explain each block one at a time.**



This block will attach the Orb to the last dynamic object in the world that it came in contact with. Put this in your “OnHit” function, and it will attach as soon as your Orb hits something. The “two yellow squares per second” means that the spell will consume 2 Earth mana per second, so you’ll want to allocate enough mana at the top of the screen for your spell to work. Once attached, the Orb cannot be unattached until it runs out of mana.



This block creates an “Elastic Connection” (which has the same basic properties as a bungee cord) between the Orb and the last dynamic “World Object” it came in contact with. Much like the “attach” block, this works especially well inside the “OnHit” function. You can also choose which “Strength” you want to connection to be, between “Weak,” “Medium”, and “Strong”, each



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

#HourofCode  
hourofcode.com

# CodeSpells: Spellcraft Lesson Plan

with higher mana costs. You'll want the connection to be stronger for larger, heavier objects; "Weak" will work just fine with a small rock, but you'll need the connection to be "Strong" in order to move a big boulder.

## Break Elastic Connection

This block will sever the Elastic Connection between the Orb and the object, if there is one. If not, it does nothing. It works well in the "OnActivate" function, letting you drop whatever object your Orb is dragging along.



This block will teleport the player to wherever the Orb is currently located...for the high cost of 50 Earth mana. Still, it's a small price to pay for the ability to move across huge spaces in the blink of an eye. And here's the kicker: you can replace "Player World Object" with *another* blue block with a tail to create some devious ways to move objects around.

## Using Other Blocks (This is mostly for reference)

With these basics in mind, you can encourage students to incorporate basic programming concepts -- like Loops, Lists, and If-Statements -- into their spells. You can find their blocks under the Power tabs:

Logic	Logic contains all the If-Statements and booleans
Loops	Loops contains all the different types of loops
Lists	Lists contains blocks that allow you to create, add to, and select from lists
Math	Math contains number blocks and various math operations
Vector3	Vector3 handles all three-dimensional operations and is very complex
Debug	Debug allows you to send messages or add Javascript code to your spell
Text	Text contains all the things you need to send the Debug messages
Variables	Variables allows you to create and set new variables and use old ones
Functions	



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

# CodeSpells: Spellcraft Lesson Plan

Functions allows you to create new functions and use old ones

Above the Power tabs are a few more Spell-specific concepts that they can bring into their spell.

Control	Control houses the three “On___” blocks and a few Event blocks
Orb	Orb allows you to change the settings of your Orb
Player	Player allows you to get properties of the Player <sup>1</sup>
Creator	Creator allows you to get properties of the Spell’s Creator <sup>2</sup>
World Object	World Object allows you to get and compare what your Orb ran into
Globs	Globs are spherical balls of elemental energy that have unique uses

## Activity (20-40 mins)

Have the students create one or two spells that work towards achieving the Goal

The Goal, once again, is to **remove the boulders from a ditch that is incorporated in the CodeSpells arena shared above and replace them with water.**

### When your students come across difficulties

It’s okay to respond:

- “I don’t know. Let’s figure this out together.”
- “Try changing some of your code and see what happens!”
- “Learning to program is like learning a new language; you won’t be fluent right away.”

### What to do if a student finishes early?

- If the student completed the goal using 2 spells, have them attempt to refine and improve it down to only 1 spell
- Have them try to incorporate different types of blocks, like Loops or Variables
- If they’ve done it in 1 spell, have them work on creating a brand new spell and exploring the different blocks

---

<sup>1</sup> The Player means you, the person playing CodeSpells

<sup>2</sup> In the full version of CodeSpells, the spell’s Creator may be different from you, the Player, in a game of multiplayer





# CodeSpells: Spellcraft Lesson Plan

## Wrap-Up (5-10 mins)

### Spell Showcase

- In the last ten minutes of class, have the students showcase the spells that they've made and the solutions they've found. You can do this either by having each student visit another's computer, or by sharing each spell to a central computer and then displaying them on a projector.

## Assessment/Extended Learning (2-5 mins)

### Optional

Time permitting, challenge your students to reflect on the day's activities and continue their learning.

- Think about new, other spells that could be made, as well as new problems that could be solved!

## Beyond one hour

There are many ways to go Beyond an Hour of Code:

- Have them play CodeSpells at home and try to make new spells
- Have them explore spells that other users have shared in CodeSpells
- Have them go to the CodeSpells forum to explore new spell ideas
- Challenge the other students to a one-on-one spell battle, with the goal of creating the most destructive or deadly spells!
- Brainstorm how problems in the real world could be solved using spells and the coding interface of CodeSpells

## Questions?

Just e-mail us at [codespells@thoughtstem.com](mailto:codespells@thoughtstem.com)! Make sure to include "Spellcraft Lesson Plan" in the title!



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

#HourofCode  
hourofcode.com