**Hopscotch Lessons from Curriculum PDF**

Link to Web Site resources: <https://www.gethopscotch.com/resources>



Link to curriculum: <http://hopscotch-curriculum-files.s3.amazonaws.com/Hopscotch_Curriculum.pdf>

Cover:



What we think: this is a well written curriculum (83 pages) with a set of lessons that would be easy to follow. The videos are lengthy but are easy to follow. They can be watched by you in preparation for the lesson but also as the lesson progresses. Also, don’t forget your students can watch the videos on their own time by supplying them with the necessary links. Both the PDF of the lessons and the videos can be shown to your students and they can follow along with the instructions. You might wonder if teaching a program that is only available for iPhones and iPads is the way to go. We leave that decision up to you. At our school we did use SCRATCH with more of our grade levels because it has an Off-Line version and an On-Line version, and a simpler version Scratch JR for iPad.

The lessons are broken down into sections and include suggested time allotments.

These sections are: Big Idea; Skill Focus; Key Vocabulary; Transfer Goals; Materials; Teacher Brief; Lesson; Differentiation; and Reflection.

In the lessons note: authors offer suggestions such as what real world examples to use when explaining computer terms; suggestion if you want to break a lesson over several time periods where the natural breaks should be; suggestions to meet the needs of different students by adapting characteristics of the game; and questions at the end of the lesson to consolidate learning. The separation of instructions 1.1, 1.2, 1.3, and so on makes it easy to direct students to specific individual instructions and tends to minimize confusion.

GUIDE TO THE LESSONS

|  |  |  |  |
| --- | --- | --- | --- |
| Lesson Title | Pages in Curriculum Document | Videos | Link to Finished Program & short Description of Lesson |
| Lesson 1 Crossy Road | 11 - 22 | <http://hop.sc/CrossyRoadVideo> or<https://www.youtube.com/watch?v=lKpKfwchrpM>  | <http://hop.sc/crossyroadproject>Learn about: Events, Sequence, LoopsAim of Game is to cross a road and avoid cars |
|  | 1. Crossy RoadA simple game that introduces events, sequences, and loops, through helping a character navigate across a busy street.*Big Idea: If you can code, you can make things that you like and use, and that may not have existed before. Coding is a superpower!* |
| Lesson 2 Geometry Dash | 23-32 | <http://hops.sc/GeometryDashVideo> or | <http://hop.sc/geometrydashproject> Learn about: Bug; Debugging; Concurrency; Random; Range |
|  | 2. Geometry DashA single-button jumping game that focuses on drawing and animation, and increasing the complexity of loops and sequences, including concurrency, so that debugging is required.*Big Idea: Computers do only what you say, because they are not smart enough to figure out what you mean. Be specific!* |
| Lesson 3: Which Emoji are You? | 33-44 | <http://hop.sc/EmojiQuizVideo> or | <http://hop.sc/emojiquizproject> Learn about Value, Variable, ConditionalAim of quiz is to keep track of player’s answers & calculate their results |
|  | 3. Which Emoji are You?A customizable quiz that keeps track of your answers and computes a score or outcome using variables and conditionals.*Big Idea: If you know how to use individual blocks like conditionals and variables, you can put them together in powerful ways to build what you want. Little blocks build big programs!* |
| Lesson 4: Flappy Bird | 45-55 | <http://hop.sc/FlappyBirdVideo>  | <http://hop.sc/flappyproject> Learn about: functions/abilities, Physics Engine, Reverse Engineering |
|  | 4. Flappy BirdAn exercise in reverse engineering, where students are deeply familiar with the goal, and have to work backward to make it happen. Introduces the concept of a physics engine.*Big Idea: Coding means telling computers what to do, in a language they can understand. Computers speak numbers!* |
| Lesson 5: Subway Surfers | 56-66 | <http://hop.sc/SubwaySurferVideo>  | <http://hop.sc/subwayproject> Learn about: pair programming,  |
|  | 5. Subway SurfersA complex action game that requires multiple components and design decisions, perfect for introducing the paradigm of pair programming.*Big Idea: There is often more than one solution to a problem, and some solutions are better than others. There may be another way!* |
| Lesson 6: Can you Escape? | 67-76 | <http://hop.sc/EscapeTheRoomVideo>  | <http://hop.sc/escapetheroomproject> Learn about: logic, iteration, values, conditions, events, binary logic |
|  | 6. Can you Escape?An open-ended point-and-click adventure that connects the ideas of programming logic to real world logic.*Big Idea: The way to write good programs is to have ideas and make mistakes, over and over. This process is called iteration. Stick to it!* |
| Lesson 7: Game Design Workshop Lesson 8: Game Showcase | 77 |  |  |
|  | 7. Game Design Workshop (Optional)An opportunity to refine one of the games in lessons 1-6, or start over from scratch with an original idea. Watch Dr. Em’s advice on making games at http://hop.sc/1MwRIID8. Game Showcase (Optional)Share your games in a showcase with others, make a webpage or ad for your game, or write a review of someone else’s game. An opportunity to practice sharing and attribution, communication and using appropriate vocabulary, and evaluating the work of others. |

Longer videos were developed for Hour of Code. These how-to videos could be used for Junior Division or Intermediate Division students. They begin at downloading the app and go step by step through creating one game, so in theory they can be used with no experience with Hopscotch. Students with some experience from other Hour of Code activities or those who have been taught the six Hopscotch lessons above will breeze through these videos.

Hopscotch: Hour of Code Screencast! (2013) 25 minutes. Link: <https://www.youtube.com/watch?v=3f7_xNIcTGY>

Teacher’s Guide (14 pages) <http://hopscotch-curriculum-files.s3.amazonaws.com/Hopscotch-2015-HOC-Teacher-Notes.pdf>

This was made to use as an Hour of Code activity. We think it would be difficult for a whole class to work on at the same time if the video was being shown to the class. Individual students might need more or less time to work on each step and that might cause management problems with frequent stops and starts and replays, keeping every in the class moving ahead. If this is your situation you might consider paired programming. We have always found this to be a better situation especially in a whole class setting. First it reduces the number of devices needed and talking back and forth to a partner helps eliminate many programming bugs before they happen. It would be suitable if the video could be played to a small group of students and then each have a device of their own or paired programming.

 