

Code Hunt Hint System

Daniel Perelman¹

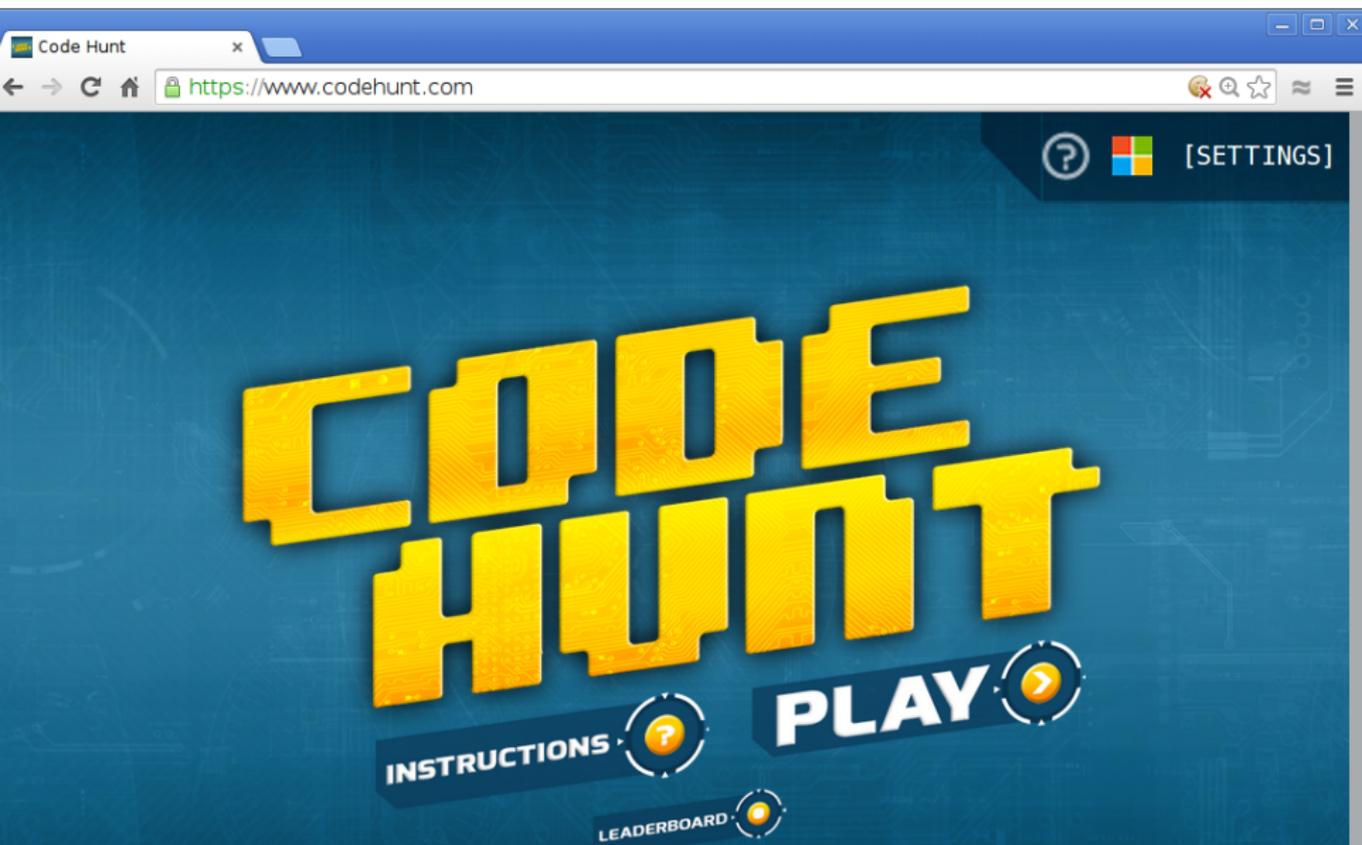
Sumit Gulwani² Dan Grossman¹ Judith Bishop²

¹University of Washington

²Microsoft Research

February 9, 2015

Code Hunt programming game



Code Hunt programming game

The screenshot shows a web browser window with the URL <https://www.codehunt.com>. The page title is "Code Hunt". The main content area is titled "SELECT SECTOR" and displays 15 numbered options arranged in a grid. Each option is represented by a colored square with a number and a topic name. The first six options (00-05) are highlighted in yellow and orange, while the remaining nine (06-14) are in dark blue. Each option has a circular arrow icon to its right, indicating a selection mechanism. The interface also includes a "CODE HUNT" logo, a hamburger menu, a question mark icon, a Windows logo, and a "[SETTINGS]" button in the top right corner.

Number	Topic	Color
00	TRAINING	Yellow/Orange
01	ARITHMETIC	Yellow/Orange
02	LOOPS	Yellow/Orange
03	LOOPS 2	Yellow/Orange
04	CONDITIONALS	Yellow/Orange
05	CONDITIONALS 2	Yellow/Orange
06	STRINGS	Dark Blue
07	STRINGS 2	Dark Blue
08	NESTED LOOPS	Dark Blue
09	1D ARRAYS	Dark Blue
10	JAGGED ARRAYS	Dark Blue
11	ARRAYS 2	Dark Blue
12	SEARCH SORT	Dark Blue
13	CYPHERS	Dark Blue
14	PUZZLES	Dark Blue

Code Hunt programming game

The screenshot shows a web browser window with the address bar displaying `https://www.codehunt.com`. The page title is "Code Hunt". The main content area has a dark blue background with a circuit-like pattern. At the top right, there is a navigation bar with the "CODE HUNT" logo, a hamburger menu icon, a grid icon, a question mark icon, a Windows logo, and a "[SETTINGS]" button. Below this, the text "SECTOR 00" is displayed in large white letters. Underneath, there are four yellow and orange gradient cards representing levels: "00.01 TUTORIAL", "00.02", "00.03", and "00.04". Each card has a progress indicator at the bottom consisting of four white dashes.

Code Hunt

<https://www.codehunt.com>

CODE HUNT

[SETTINGS]

SECTOR 00

00.01
TUTORIAL

00.02

00.03

00.04

4 / 65

Code Hunt programming game

Code Hunt

https://www.codehunt.com

LEVEL: 00.02 ▶

CODE HUNT [SETTINGS]

Discover the arithmetic operation applied to 'x'.

RESET LEVEL SWITCH TO C# Java

```
1  
2 public class Program {  
3     public static int Puzzle(int x) {  
4         return 0;  
5     }  
6 }
```

Code Hunt programming game

LEVEL: 00.02 ▶

CODE HUNT [SETTINGS]

Discover the arithmetic operation applied to 'x'.

```
1  
2 public class Program {  
3     public static int Puzzle(int x) {  
4         return 0;  
5     }  
6 }
```

RESET LEVEL SWITCH TO C# Java

	X	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
✖	0	1	0	Mismatch
✔	-1	0	0	

Code Hunt programming game

LEVEL: 00.02 ▶

CODE HUNT [SETTINGS]

Discover the arithmetic operation applied to 'x'.

```
1 public class Program {
2     public static int Puzzle(int x) {
3         return 1;
4     }
5 }
6 }
```

X	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
0	1	1	
1	2	1	Mismatch

Java

RESET LEVEL SWITCH TO C#

65

Code Hunt programming game

Code Hunt

LEVEL: 00.02 ▶

CODE HUNT

Java

Discover the arithmetic operation applied to 'x'.

CAPTURE CODE

```
1  
2 public class Program {  
3     public static int Puzzle(int x) {  
4         if(x == -1) {  
5             return 0;  
6         } else if(x == 0) {  
7             return 1;  
8         } else if(x == 1) {  
9             return 2;  
10        } else {  
11            return 0;
```

	X	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
✓	-1	0	0	
✓	0	1	1	
✓	1	2	2	
✗	2	3	0	Mismatch

RESET LEVEL SWITCH TO C#

65

Code Hunt programming game

LEVEL: 00.02 ▶

CODE HUNT [SETTINGS]

Discover the arithmetic operation applied to 'x'.

CAPTURE CODE

```
1 public class Program {
2     public static int Puzzle(int x) {
3         return x+1;
4     }
5 }
6 }
```

Java

RESET LEVEL SWITCH TO C#

X	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
0	1	1	✓

Code Hunt programming game

The screenshot shows a web browser window with the address bar displaying <https://www.codehunt.com>. The main content area features a dark grey background with a circuit board pattern. At the top, a yellow bar contains the text "You repaired and captured the code fragment." Below this, the text "SKILL RATING:" is followed by three yellow rectangular bars. Underneath, it says "you wrote elegant code!". The central focus is the text "TOTAL SCORE: 6" in large white font. At the bottom, there are three buttons: a blue "KEEP TRYING" button on the left, a yellow "NEXT" button on the right, and a central circular "LEVEL SELECT" button with a 3x3 grid of squares and directional arrows. On the left side of the browser window, a vertical sidebar shows a list of levels numbered 1 through 6. On the right side, there are navigation links for "[TINGS]", "Java", and "RIPTION".

Code Hunt

<https://www.codehunt.com>

LE

Disc

opera

1

2

3

4

5

6

[TINGS]

Java

RIPTION

You repaired and captured the code fragment.

SKILL RATING: [] [] []

you wrote elegant code!

TOTAL SCORE: 6

KEEP TRYING

LEVEL SELECT

NEXT

65

More difficult level

The screenshot shows a web browser window with the URL <https://www.codehunt.com>. The page title is "Code Hunt". The main heading is "SELECT SECTOR". Below the heading is a grid of 15 challenge sectors, each represented by a colored square with a number and a title. The sectors are arranged in three rows of six, with the last row containing three sectors. The first six sectors (00-05) are highlighted in yellow and orange, while the remaining nine (06-14) are in a darker blue. Each sector has a circular arrow icon on its right side, indicating a sequence.

Number	Title
00	TRAINING
01	ARITHMETIC
02	LOOPS
03	LOOPS 2
04	CONDITIONALS
05	CONDITIONALS 2
06	STRINGS
07	STRINGS 2
08	NESTED LOOPS
09	1D ARRAYS
10	JAGGED ARRAYS
11	ARRAYS 2
12	SEARCH SORT
13	CYPHERS
14	PUZZLES

More difficult level

The screenshot shows the Code Hunt website interface. At the top, the browser address bar displays `https://www.codehunt.com`. The page header includes the text "LEVEL: 03.03" and "ATTEMPTS: 13". A "CAPTURE CODE" button is overlaid on the code editor. The code editor contains the following Java code:

```
1 public class Program {  
2     public static int Puzzle(int lowerBound, int upperBound) {  
3         return lowerBound * upperBound;  
4     }  
5 }  
6 }
```

Below the code editor is a table with the following columns: LOWERBOUND, UPPERBOUND, EXPECTED RESULT, YOUR RESULT, and DESCRIPTION. The table contains three rows of test cases:

	LOWERBOUND	UPPERBOUND	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
✗	1	8	40320	8	Mismatch
✗	15	24	244963328	360	Mismatch
✓	16	17	272	272	

Buttons for "RESET LEVEL" and "SWITCH TO C#" are visible in the top right corner of the interface. The language "Java" is selected.

More difficult level

The screenshot shows the Code Hunt website interface. At the top, the browser address bar displays `https://www.codehunt.com`. The page header includes the level information: `LEVEL: 03.03` and `ATTEMPTS: 27`. A navigation bar contains the Code Hunt logo, a menu icon, a help icon, a language selector (set to Java), and a `[SETTINGS]` button.

The main content area features a code editor with the following Java code:

```
1 public class Program {  
2     public static int Puzzle(int lowerBound, int upperBound) {  
3         return 40320;  
4     }  
5 }
```

Below the code editor is a table with the following columns: LOWERBOUND, UPPERBOUND, EXPECTED RESULT, YOUR RESULT, and DESCRIPTION. The table contains two rows of data:

LOWERBOUND	UPPERBOUND	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
1	8	40320	40320	
9	16	518918400	40320	Mismatch

A green message bar at the bottom of the table reads: `@ The expression 40320 is rarely used to solve this level.`

A circular "CAPTURE CODE" button is overlaid on the code editor, pointing to the `return 40320;` line.

More difficult level

The screenshot shows the Code Hunt website interface. At the top, the browser address bar displays "https://www.codehunt.com". The page header includes "LEVEL: 03.03" and "ATTEMPTS: 14". A "CAPTURE CODE" button is overlaid on the code editor. The code editor contains the following Java code:

```
1 public class Program {
2     public static int Puzzle(int lowerBound, int upperBound) {
3         return lowerBound * upperBound;
4     }
5 }
6 }
```

Below the code editor is a table with the following data:

	LOWERBOUND	UPPERBOUND	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
	1	8	40320	8	Mismatch
	15	24	244963328	360	Mismatch
	16	17	272	272	

A green message box below the table says: "You may find a loop useful on this level."

At the bottom right corner, the page number "14 / 65" is visible.

More difficult level

The screenshot shows the Code Hunt website interface. At the top, it displays "LEVEL: 03.03" and "ATTEMPTS: 22". The puzzle instruction is "Try to capture the code fragment!". The code to be captured is a Java method:

```
1 public class Program {  
2     public static int Puzzle(int lowerBound, int upperBound) {  
3         int r = 1;  
4         for(int i = lowerBound; i < upperBound; i++)  
5             r *= i;  
6         return r;  
7     }  
8 }
```

A "CAPTURE CODE" button is overlaid on the code. To the right, there are buttons for "RESET LEVEL" and "SWITCH TO C#". Below the code is a table with the following data:

	LOWERBOUND	UPPERBOUND	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
	1	8	40320	5040	Mismatch
	16	22	859541760	39070080	Mismatch
@	Looking good. Look at line 4 to capture the code.				

The page number "15 / 65" is visible in the bottom right corner.

More difficult level

The screenshot shows the Code Hunt website interface. At the top, it displays the level "03.03" and "ATTEMPTS: 17". The main area contains a Java code snippet for a puzzle. A circular "CAPTURE CODE" button is overlaid on the code. To the right, there is a table with test results and a green message box.

Try to capture the code fragment!

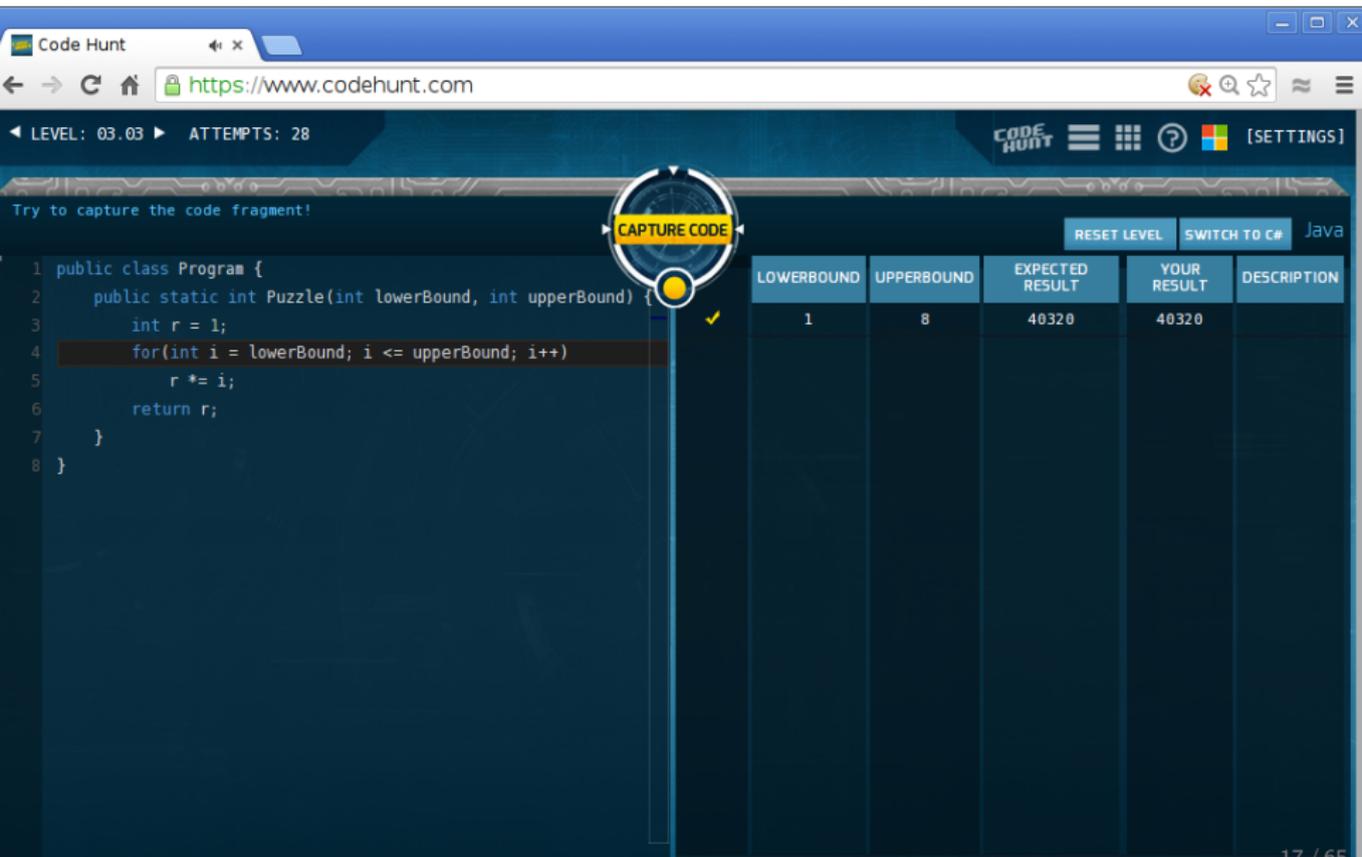
```
1 public class Program {
2     public static int Puzzle(int lowerBound, int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

LOWERBOUND	UPPERBOUND	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
1	8	40320	5040	Mismatch
16	22	859541760	39070080	Mismatch

e You may find the expression `<int> <= <int>` useful on this level.

16 / 65

More difficult level



The screenshot shows the Code Hunt website interface. At the top, the browser address bar displays "https://www.codehunt.com". The page header includes "LEVEL: 03.03" and "ATTEMPTS: 28". A "CAPTURE CODE" button is overlaid on the code editor. The code editor contains the following Java code:

```
1 public class Program {  
2     public static int Puzzle(int lowerBound, int upperBound) {  
3         int r = 1;  
4         for(int i = lowerBound; i <= upperBound; i++)  
5             r *= i;  
6         return r;  
7     }  
8 }
```

Below the code editor is a table with the following columns: LOWERBOUND, UPPERBOUND, EXPECTED RESULT, YOUR RESULT, and DESCRIPTION. The table contains one row of data:

LOWERBOUND	UPPERBOUND	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
1	8	40320	40320	

The page also features a "RESET LEVEL" button, a "SWITCH TO C#" button, and a "Java" language selector. The footer of the page shows "17 / 65".

Kinds of hints

- ▶ Line hints: “Look at **line 4** to capture the code.”
- ▶ Positive recommendation hints: “You may find **a loop** useful on this level.”
- ▶ Negative recommendation hints: “**The expression 40320** is rarely used to solve this level.”

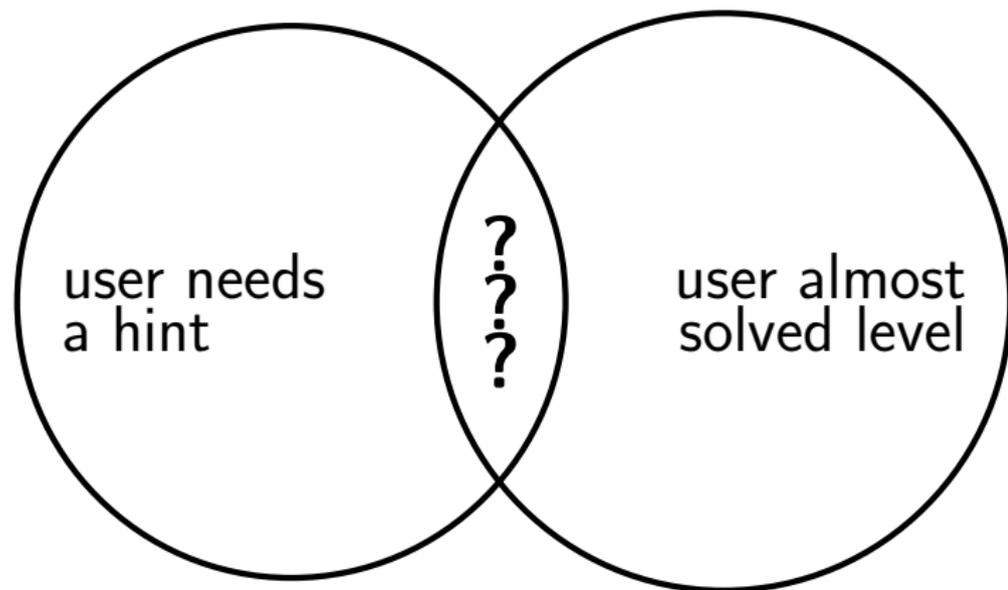
Line hints

- ▶ Generated using program synthesis
- ▶ Solve level based on user's attempt, return lines that differ

Line hints

- ▶ Generated using program synthesis
- ▶ Solve level based on user's attempt, return lines that differ
- ▶ Can only give such hints when the user is close to a solution
 - ▶ Otherwise, the hint would have to say “change everything” .

Are line hints useful?



Are line hints useful? Yes, a little.

- ▶ Compute hints, only show to randomly selected users.

Are line hints useful? Yes, a little.

- ▶ Compute hints, only show to randomly selected users.

	show hint	hide hint
solved within 1 attempt	83 %	57%
solved eventually	98.4%	98%

- ▶ Summary: majority of users don't need help, but a statistically significant minority do go faster.

Are line hints useful? Yes, a little.

- ▶ Compute hints, only show to randomly selected users.

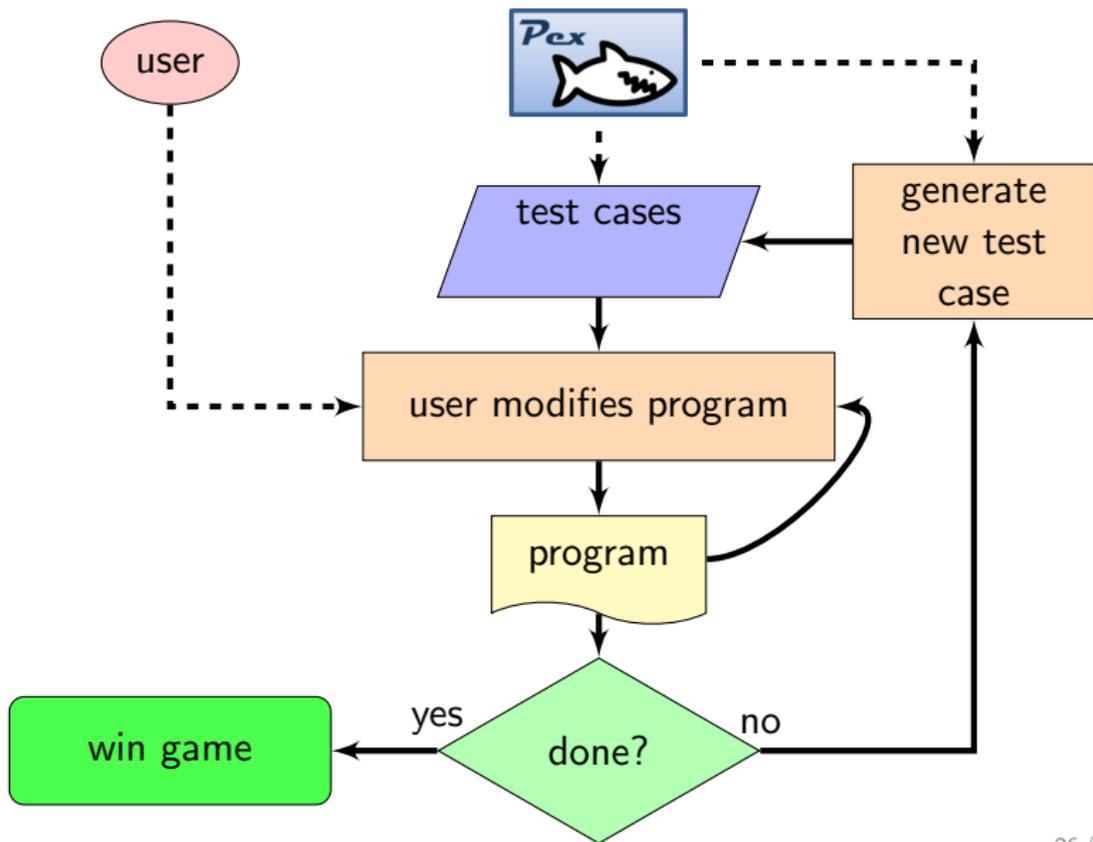
	show hint	hide hint
solved within 1 attempt	83 %	57%
solved eventually	98.4%	98%

- ▶ Summary: majority of users don't need help, but a statistically significant minority do go faster.
- ▶ But the help only helped users complete the level faster; almost all of them completed it either way.

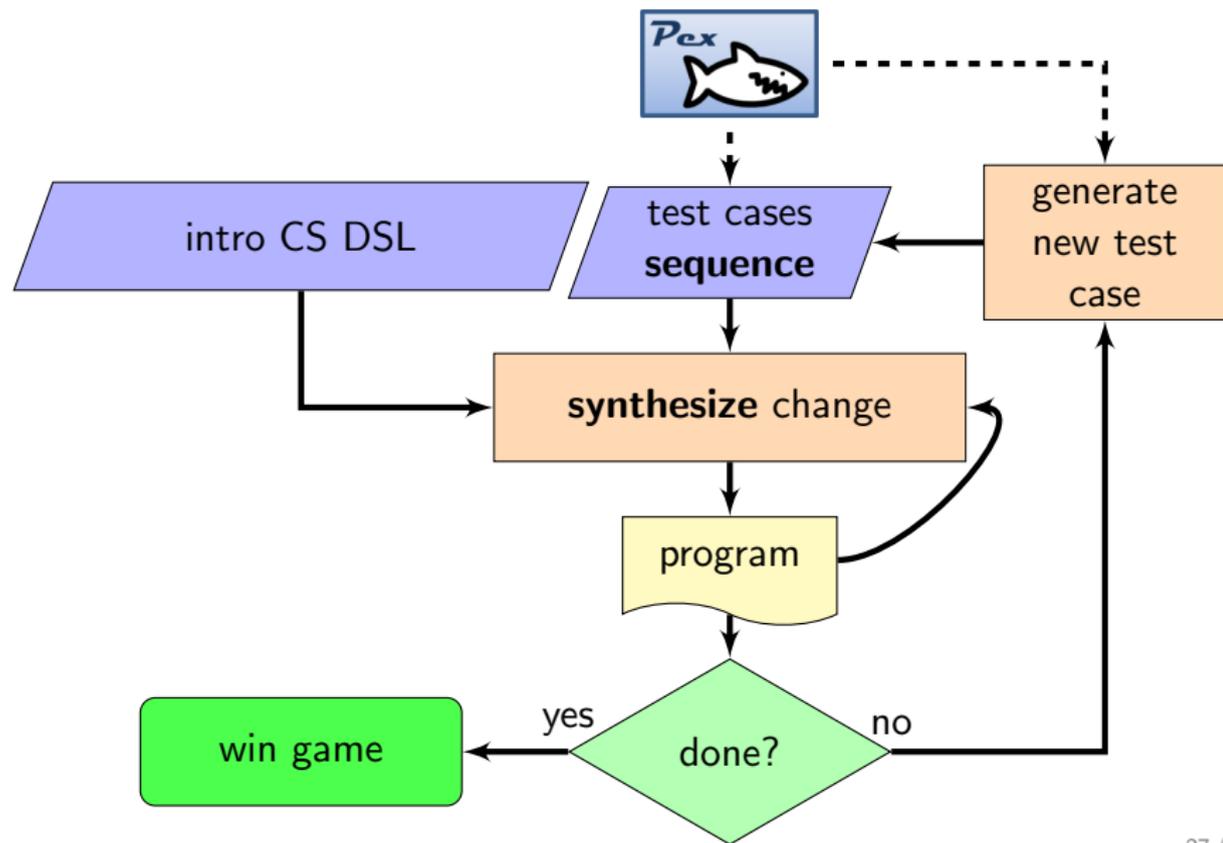
How are line hints generated?

- ▶ Requires an aside on TDS synthesizer.

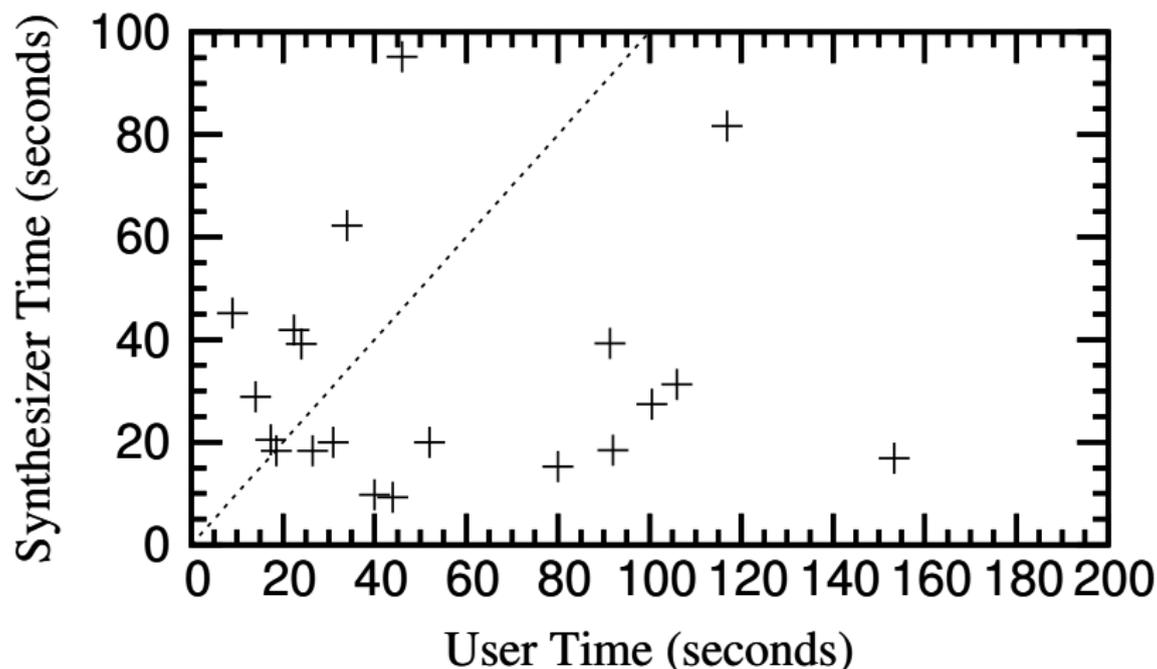
Code Hunt workflow



Synthesizer plays Code Hunt

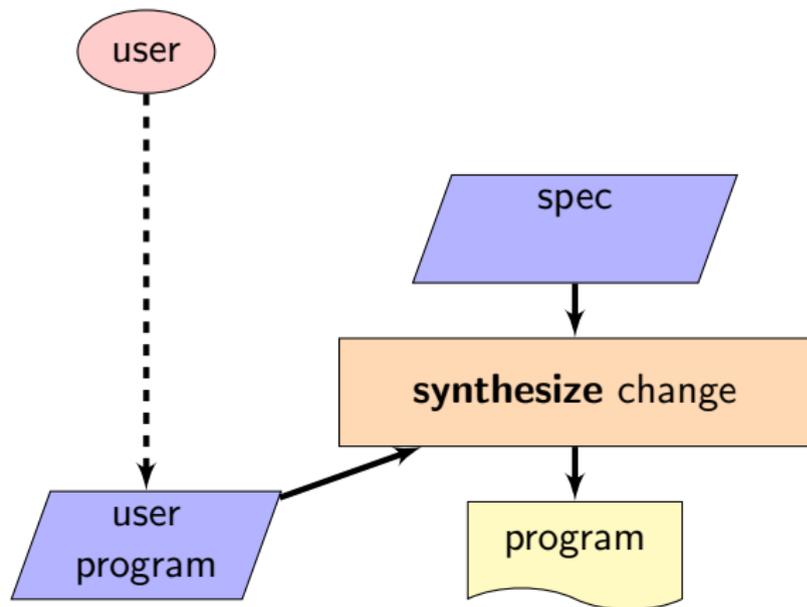


Comparison to human players for entire levels

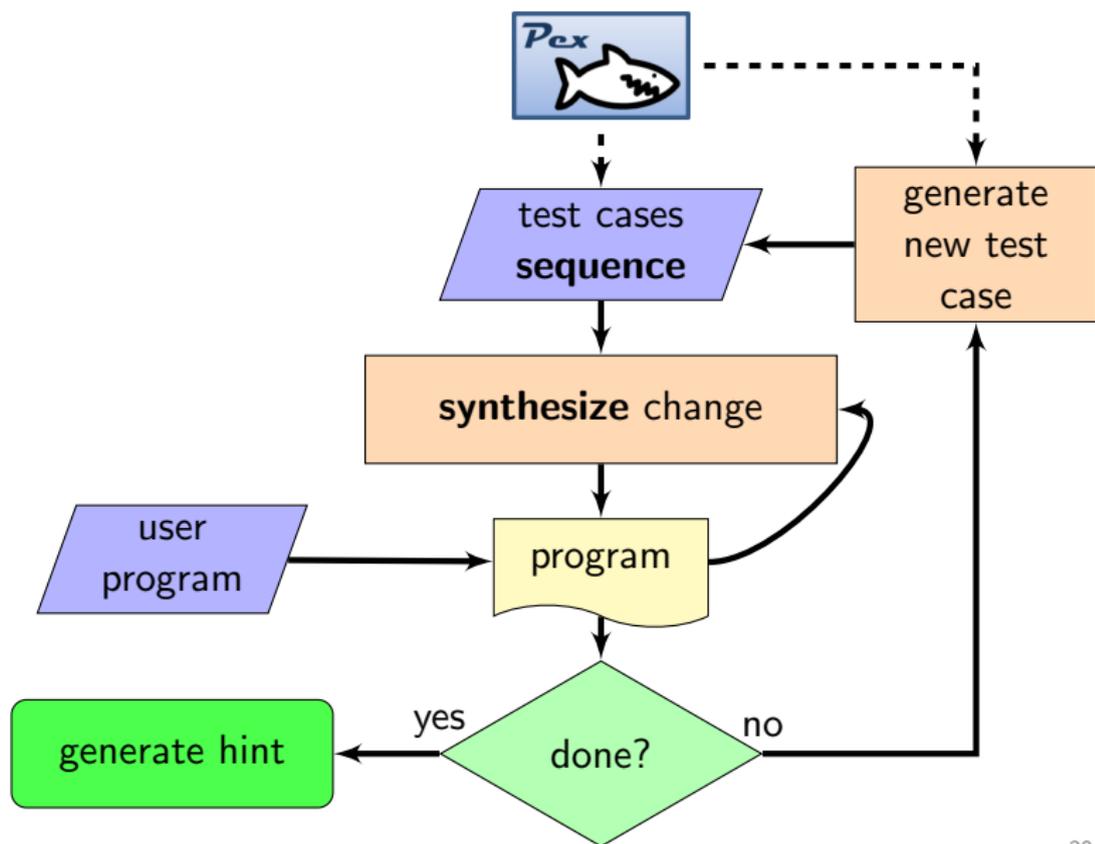


Completion times comparable to human players... use for hints?

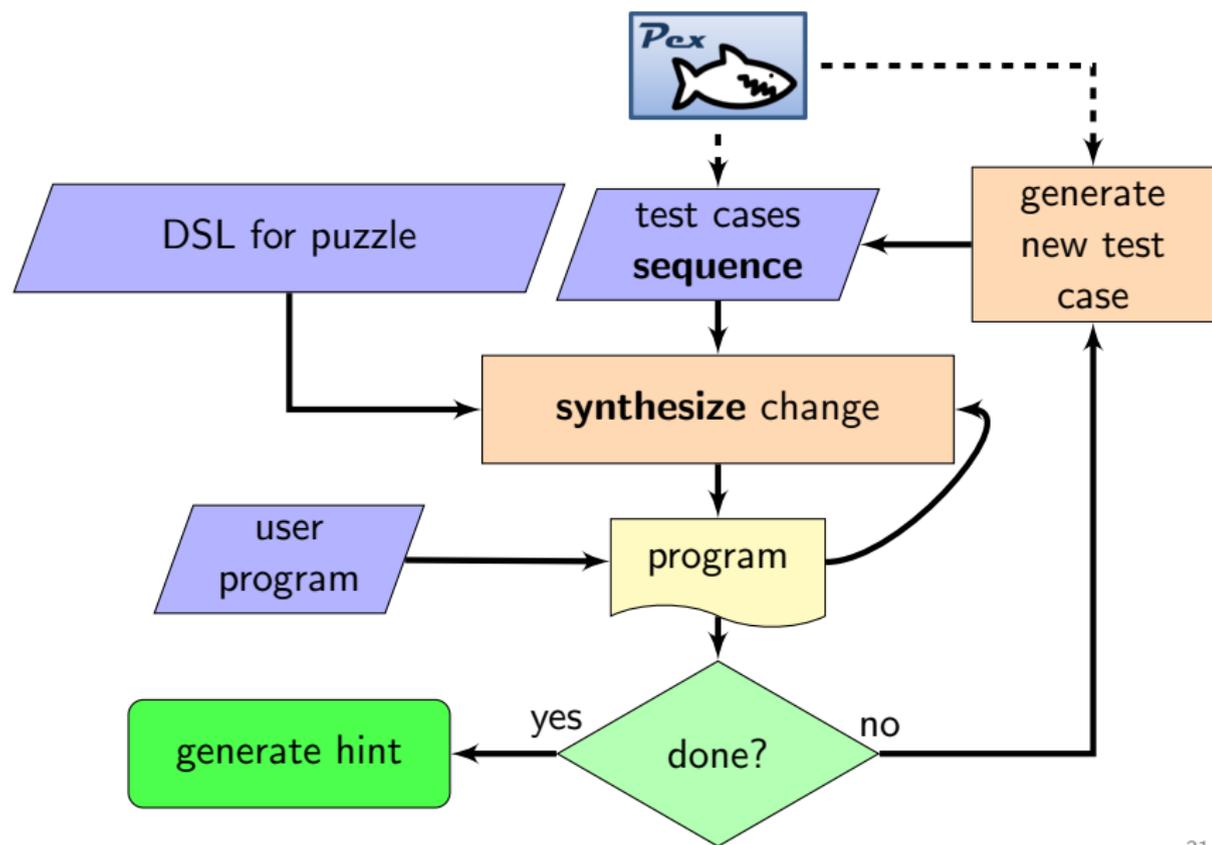
Code Hunt line hint workflow



Code Hunt line hint workflow



Code Hunt line hint workflow



Modifying program

- ▶ Replace single subexpression covered by failing test cases
- ▶ Where to modify program
- ▶ What to replace with

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Where to modify: all subexpressions

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

What to put there

- ▶ DSL defines space of expressions
 - ▶ Obtained from other users' solutions

- ▶ Prefer expressions found in user's attempt

Complication: multiple possible hints

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Complication: multiple possible hints

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = upperBound;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Complication: multiple possible hints

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i < upperBound; i++)
5             r *= i;
6         return r * upperBound;
7     }
8 }
```

Complication: multiple possible hints

```
// Puzzle(1, 8) == 40320 (not 5040)
// Puzzle(16, 22) == 859541760 (not 39070080)
1 public class Program {
2     public static int Puzzle(int lowerBound,
int upperBound) {
3         int r = 1;
4         for(int i = lowerBound; i <= upperBound; i++)
5             r *= i;
6         return r;
7     }
8 }
```

Solution: choose smallest change

- ▶ Get multiple solutions from synthesizer.
- ▶ Compute edit distance in characters for each and take the minimum.

Solution: choose smallest change

- ▶ Get multiple solutions from synthesizer.
- ▶ Compute edit distance in characters for each and take the minimum.
- ▶ Downside: can't return a hint as soon as the first one is computed.

Complication: single line programs

- ▶ Single line attempts are common, especially on earlier levels.
- ▶ User could add newlines to get more detail.

Solution: more detail than just line number

The screenshot shows the Code Hunt game interface. At the top, the browser address bar displays `https://www.codehunt.com`. The game header includes the text "LEVEL: 00.02" and navigation icons for "CODE HUNT", a menu, a help icon, and a settings icon labeled "[SETTINGS]".

The main instruction reads: "Discover the arithmetic operation applied to 'x'." Below this is a code editor with the following code:

```
1 public class Program {
2     public static int Puzzle(int x) {
3         return x + 11;
4     }
5 }
6 }
```

A "CAPTURE CODE" button is positioned over line 4 of the code. To the right, there are two buttons: "RESET LEVEL" and "SWITCH TO C#".

Below the code editor is a table with the following data:

X	EXPECTED RESULT	YOUR RESULT	DESCRIPTION
0	1	11	Mismatch

A green tooltip is displayed over the table, containing the text: "@ Getting close. Look at numbers on line 4 to capture the code."

At the bottom of the screen, the footer contains "Microsoft Research" on the left and "ABOUT | TERMS OF USE | PRIVACY & COOKIES" on the right. The page number "55 / 65" is visible in the bottom right corner.

Solution: more detail than just line number

- ▶ If line hint covers entire program, give kind of expression like “number” or “variable” .

Recommendation hints

- ▶ Positive recommendation hints:
“You may find **a loop** useful on this level.”
- ▶ Negative recommendation hints:
“**The expression 40320** is rarely used to solve this level.”

How to generate recommendation hints

- ▶ Use data!
- ▶ Have lots of attempts for which we know both if they are correct and which concepts (e.g. “loops” or “the expression 40320”) they contain.

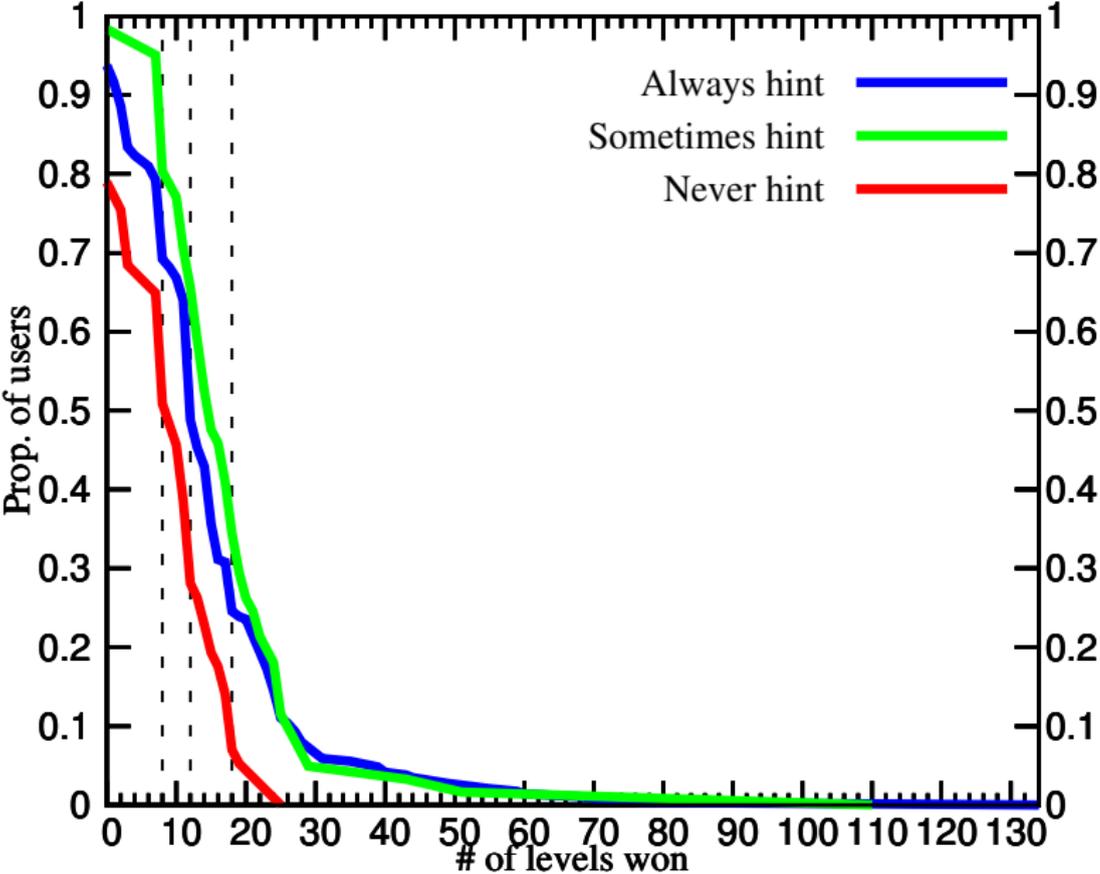
How to generate recommendation hints

- ▶ For every concept, count users that use it by whether it appears in a solution.
- ▶ If often tried but does not end up in solutions, warn users of that concept away from it.
- ▶ If often appears in solutions, suggest it to users if no other hint is available.

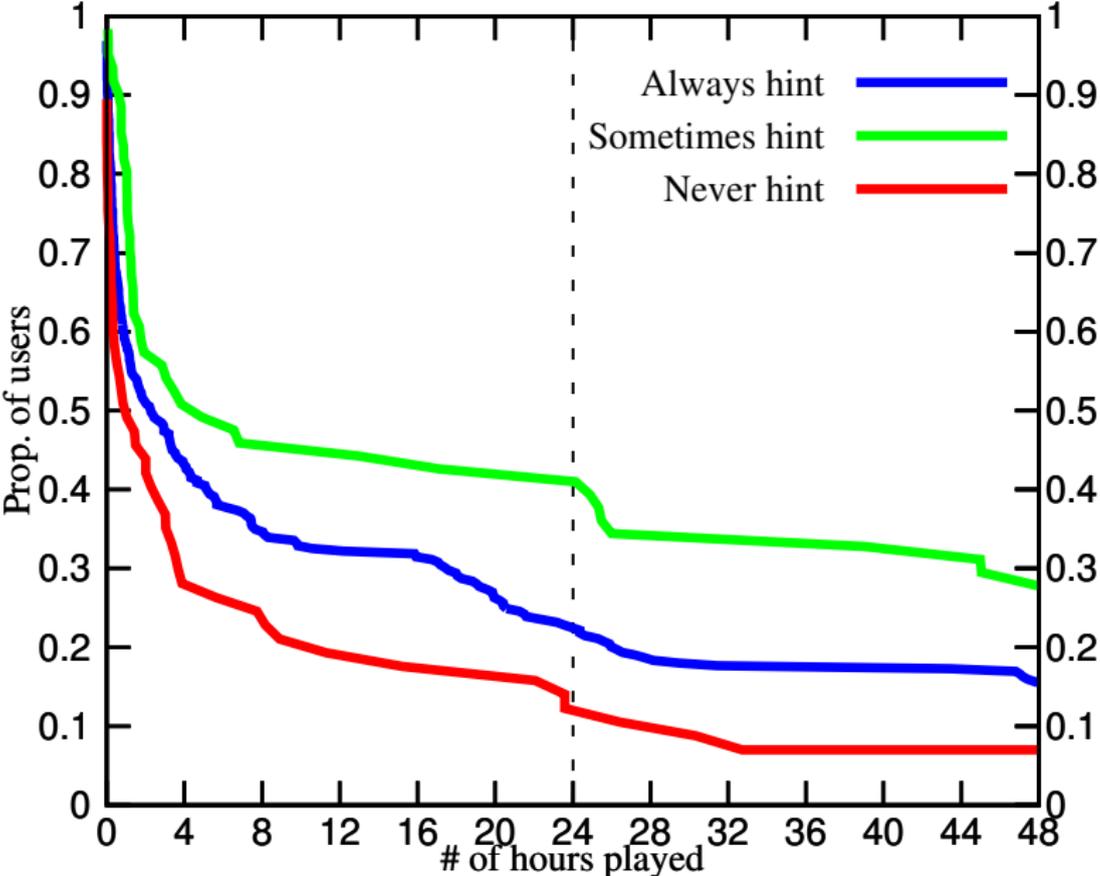
Evaluation

- ▶ Ran experiment where hints were disabled for some users.
 - ▶ “Always” : always show hints
 - ▶ “Never” : never show hints
 - ▶ “Sometimes” : give hints on some levels but not others
- ▶ Will show data on one-day period after user started playing

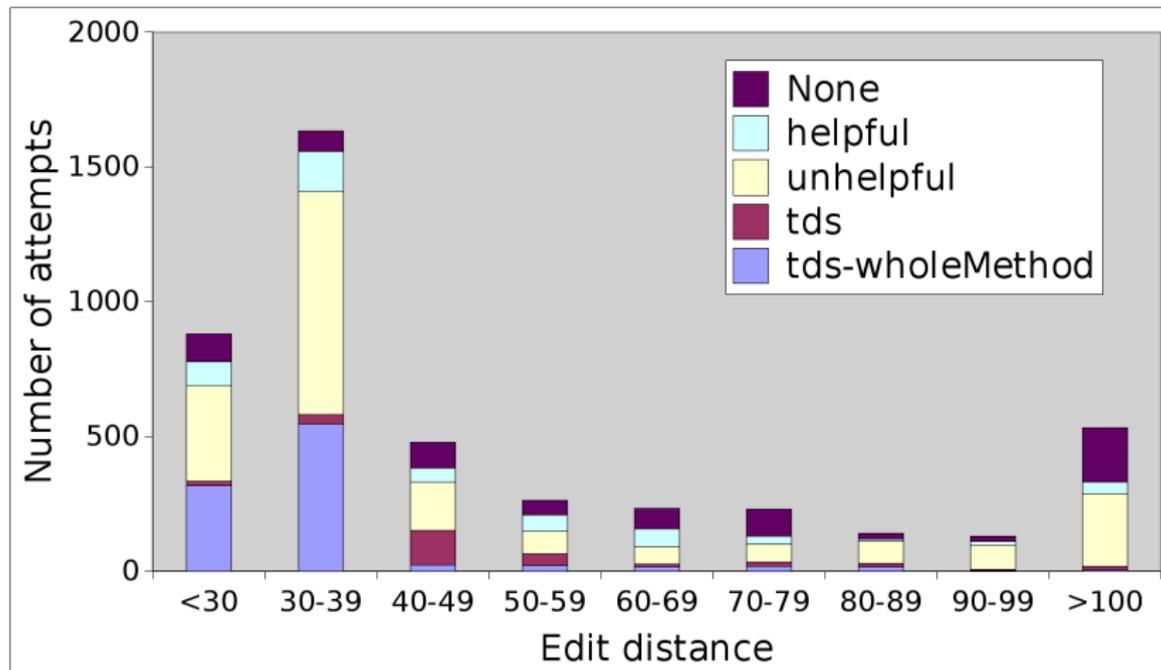
Effect on number of levels won



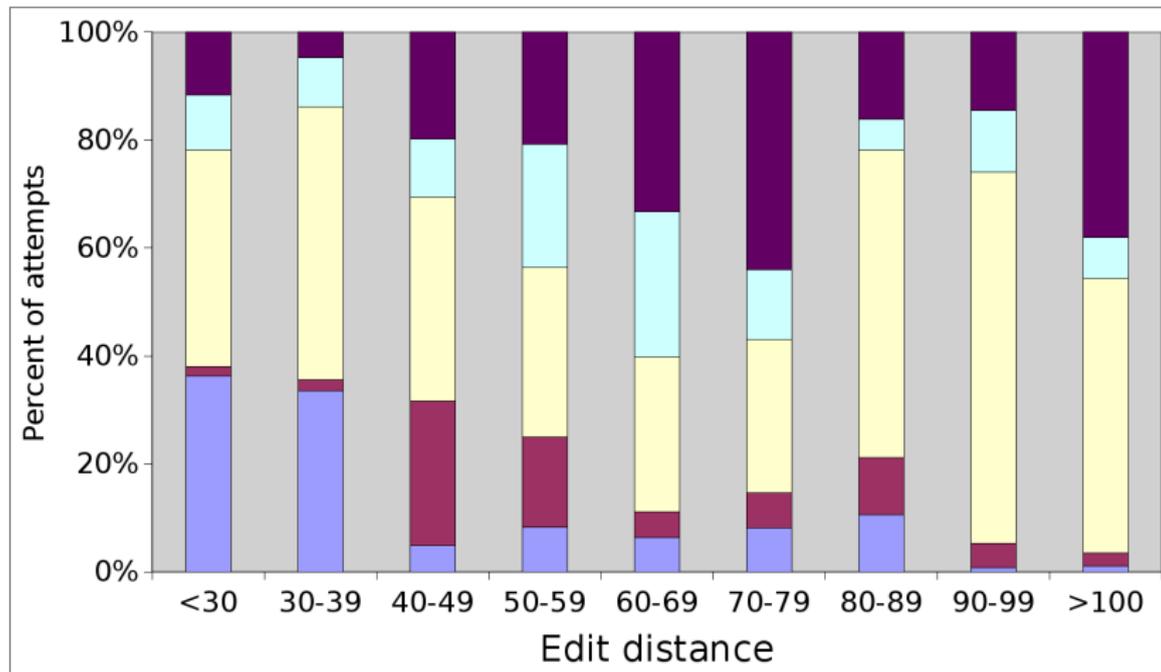
Effect on how long users play



Hint kind by edit distance



Hint kind by edit distance



Questions

Backup slides

The Greek (for recommendation hints)

Constants: $a = 10, z = 1, \tau = 0.75$

Adjusted counts: $c' = c + a, \bar{c}' = \bar{c} + a, n' = c' + \bar{c}'$

Estimated probability: $p = \frac{c'}{n'}$

Standard deviation: $\sigma = \sqrt{\frac{p(1-p)}{n'}}$

Check for significance: $p - z\sigma > \tau$