tell me to survive

David Li, Michael Mauer, and Andy Jiang

May 17, 2016

Cornell University

MOTIVATION

- Lots of programming games already

MOTIVATION

- Lots of programming games already
- Object-oriented programming?

MOTIVATION

- Lots of programming games already
- Object-oriented programming?
- Object-oriented thinking!

MOTIVATION

- Lots of programming games already
- Object-oriented programming?
- Object-oriented thinking!
- Concepts from real world examples

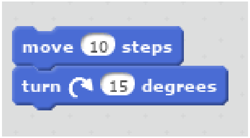
MOTIVATION

- Lots of programming games already
- Object-oriented programming?
- Object-oriented thinking!
- Concepts from real world examples
- Concepts to code?

MOTIVATION

- Lots of programming games already
- Object-oriented programming?
- Object-oriented thinking!
- Concepts from real world examples
- **Game mechanics** to code?

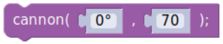
RELATED WORK

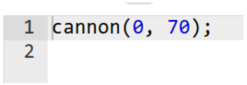


Scratch's block interface

- Visual Programming
  - Scratch, Looking Glass, CodeSpells, Blockly Games, LightBot, Human Resource Machine, . . .

RELATED WORK



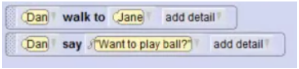


```
cannon(0, 70);
```

Blockly Games concreteness fading

- Visual Programming
  - Scratch, Looking Glass, CodeSpells, Blockly Games, LightBot, Human Resource Machine, . . .
- Concreteness Fading
  - Blockly Games

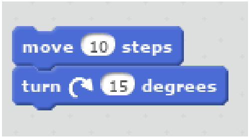
RELATED WORK

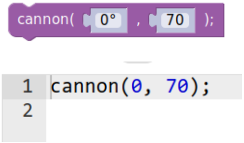

Dan walk to Jane add detail
Dan say "Want to play ball?" add detail

Looking Glass OOP (credit: Looking Glass tutorial)

- Visual Programming
  - Scratch, Looking Glass, CodeSpells, Blockly Games, LightBot, Human Resource Machine, . . .
- Concreteness Fading
  - Blockly Games
- Object-Oriented Programming
  - Looking Glass, CodeSpells

RELATED WORK





```
cannon(0, 70);
```

Key inspirations:

- Visual Programming
  - **Scratch**, Looking Glass, CodeSpells, **Blockly Games**, **LightBot**, Human Resource Machine, . . .
- Concreteness Fading
  - **Blockly Games**
- Object-Oriented Programming
  - Looking Glass, CodeSpells

RELATED WORK

- Combination is key

FINAL PROJECT

- Idea: solve puzzles with object-oriented code

FINAL PROJECT

- Idea: solve puzzles with object-oriented code
- Scope: OOP fundamentals

FINAL PROJECT

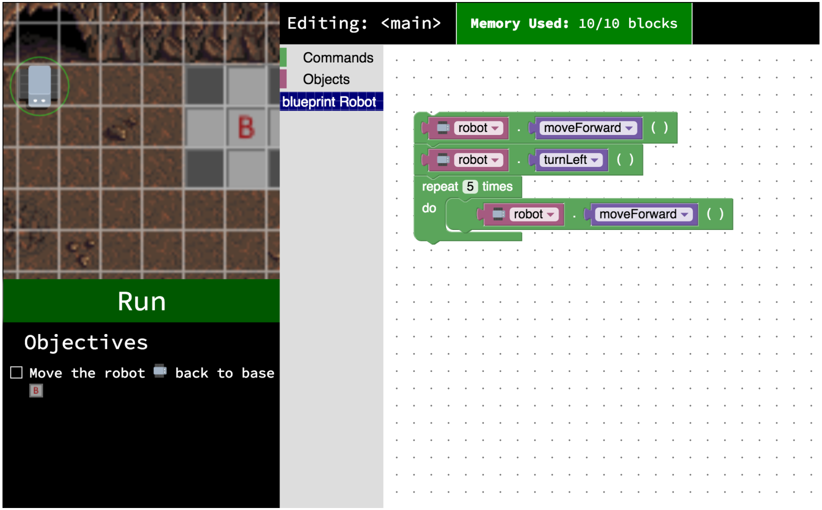
- Idea: solve puzzles with object-oriented code
- Scope: OOP fundamentals
- Goal: ease transition

FINAL PROJECT

Force use of OOP via:

- Limited block complexity
- Block limit

FINAL PROJECT



Editing: <main>

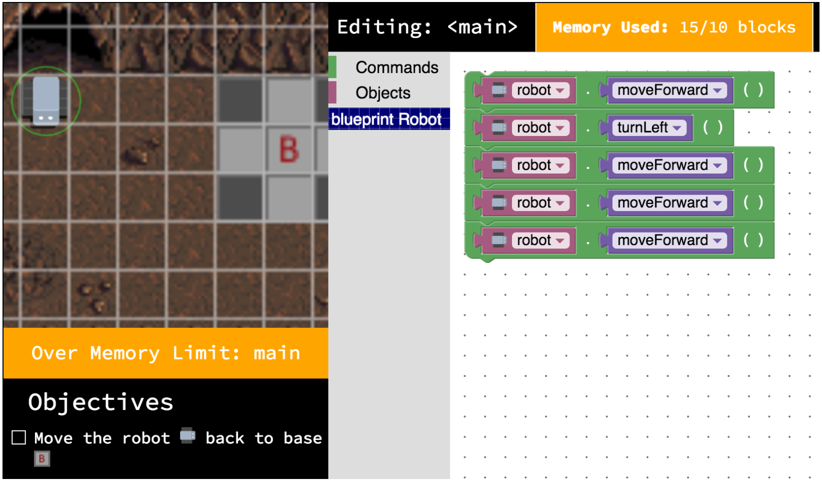
Memory Used: 10/10 blocks

```
robot . moveForward ( )
robot . turnLeft ( )
repeat 5 times
do
    robot . moveForward ( )
```

Objectives

☐ Move the robot back to base B

FINAL PROJECT



Editing: <main>

Memory Used: 15/10 blocks

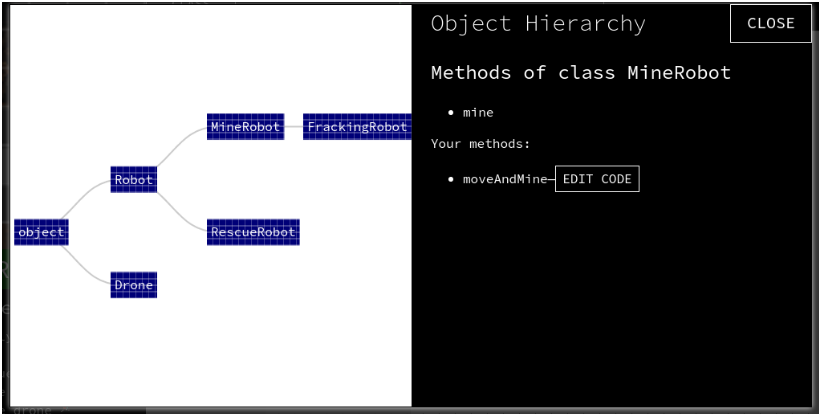
```
robot . moveForward ( )
robot . turnLeft ( )
robot . moveForward ( )
robot . moveForward ( )
robot . moveForward ( )
```

Over Memory Limit: main

Objectives

☐ Move the robot back to base B

FINAL PROJECT



Object Hierarchy

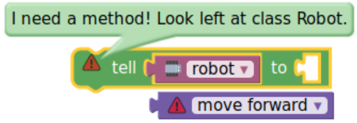
Methods of class MineRobot

- mine

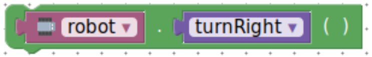
Your methods:

- moveAndMine

CONCRETENESS FADING



⇓



⇓

```
robot.moveForward()
robot.turnRight()
```

Three stages:

1. Textual description
2. Python syntax on blocks
3. Code editor

CONCRETENESS FADING

Considerations:

- Fading + new concept = confusion
- Scaffolding

CONCRETENESS FADING

Considerations:

- **Fading + new concept = confusion**
  - Introduce either a new concept or a faded block per level
- Scaffolding

CONCRETENESS FADING

Considerations:

- Fading + new concept = confusion
    - Introduce either a new concept or a faded block per level
- **Scaffolding**
    - Keep object hierarchy, warnings, tooltips after fading
    - Unintended: players used faded blocks to help them write code

USER STUDY

- “Robot Commander Aptitude Test”
- 5 questions
- Mix of syntax and concepts

RESULTS

- 12 people completed the game and post-test
- Pre-test mean: 3.167 ($s = 1.267$)
- Post-test mean: 4.25 ($s = 0.754$)

RESULTS

- 12 people completed the game and post-test
- Pre-test mean: 3.167 (s = 1.267)
- Post-test mean: 4.25 (s = 0.754)
- Statistically significant
    - paired t-test
    - $t(11) = 3.767$, $p \approx 0.003$

RESULTS

Individual questions:

Question	Concepts	Significant?	p	χ^2
1	Inheritance	No	0.2482	1.333
2	Instantiation	**Yes**	0.0133	6.125
3	Method invocation	No	0.1336	2.250
4	Overriding	No	0.4795	0.500
5	Subclassing	No	0.4795	0.500

RESULTS

Confounding factors:

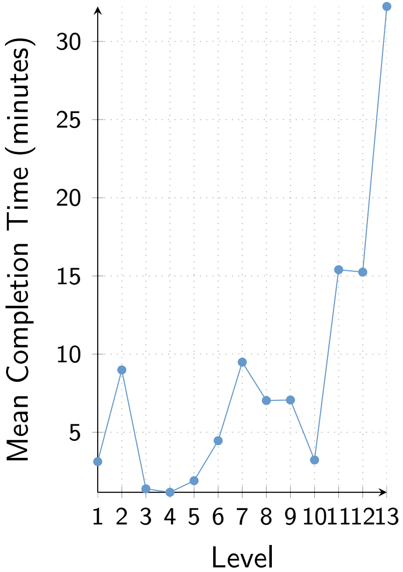
- Learning during the pre-test

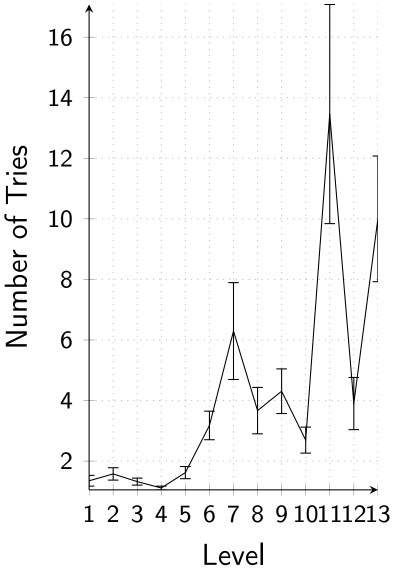
RESULTS

Confounding factors:

- Learning during the pre-test
- Not completed in one session

RESULTS


Mean Completion Time (minutes)
30
25
20
15
10
5
1 2 3 4 5 6 7 8 9 10 11 12 13
Level


Number of Tries
16
14
12
10
8
6
4
2
1 2 3 4 5 6 7 8 9 10 11 12 13
Level

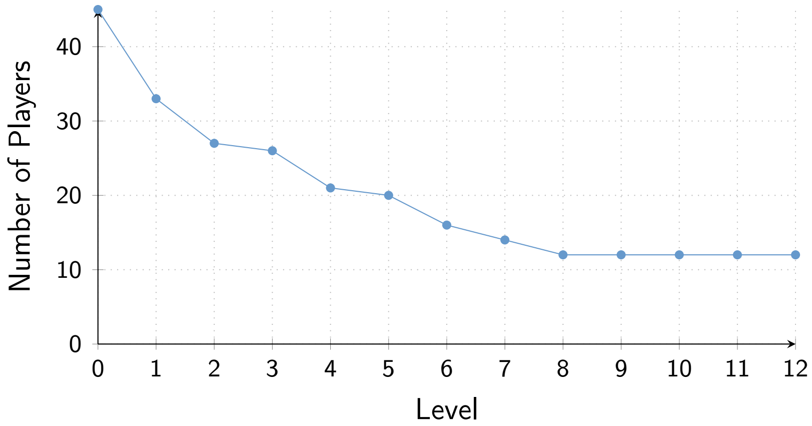
RESULTS

Subjective impressions:

Statement	Mean score
I enjoyed this game.	1.08
Before playing, I knew object-oriented programming.	0.0
After playing, I knew object-oriented programming better.	1.25

Strongly Agree → 2; Agree → 1; Neutral → 0; Disagree → -1; Strongly Disagree → -2

RESULTS



Mean completion time (estimated): ~2 hours

RESULTS

General feedback:

- Minor issues with usability and interface

CONCLUSIONS

- Statistically significant increase in
    - Overall mean
    - Performance on instantiation question
    - Test design is questionable
- Engagement is still a question
    - Needed to work closely with testers

FUTURE WORK

- More object-oriented concepts
- Improved testing methodology