Towards Generalizing Expert Programmers' Suggestions for Novice Programmers

Michelle Ichinco, Aaron Zemach and Caitlin Kelleher

Need for Computer Programming Knowledge

- Huge number of end-user programmers
  - 90 million in the US[1]
- Shortage of computing degree holders[2]

[1] C. Scaffidi, M. Shaw, and B. Myers, "Estimating the numbers of end users and end user programmers," in *Visual Languages and Human-Centric Computing*, *2005 IEEE Symposium on*, 2005, pp. 207–214.

[2] National Center for Women & Information Technology, "Computing Education and Future Jobs: A Look at National, State & Congressional District Data," Dec. 2011.

Opportunities for K-12 Classroom Instruction are not Universal

- Not enough teachers
- Lack of resources
- CS unrepresented in required curriculum

- For example, in the US:
  - 10% of high schools teach computer science[3]

[3] Microsoft. "A National Talent Strategy" http://www.microsoft.com/en-us/news/download/presskits/citizenship/MSNTS.pdf

Classroom vs. Independently

	Classroom	Independently
Motivation	Grades, Teachers	Novice programming environments (like Scratch, Alice, Looking Glass)

Classroom vs. Independently

	Classroom	Independently
Motivation	Grades, Teachers	Novice programming environments (like Scratch, Alice, Looking Glass)
Materials	Books, teacher created instructional materials	Websites, tutorials, massive open online courses

Classroom vs. Independently

	Classroom	Independently
Motivation	Grades, Teachers	Novice programming environments (like Scratch, Alice, Looking Glass)
Materials	Books, teacher created instructional materials	Websites, tutorials, massive open online courses
Questions	Teachers, classmates	Forums like Stack Overflow

Classroom vs. Independently

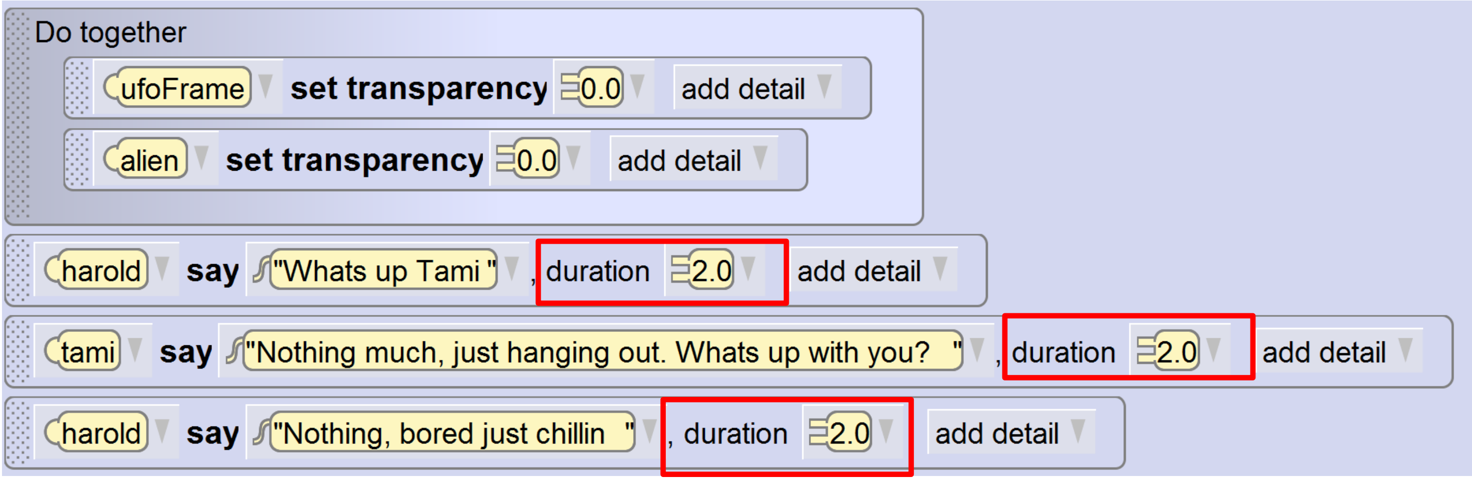
	Classroom	Independently
Motivation	Grades, Teachers	Novice programming environments (like Scratch, Alice, Looking Glass)
Materials	Books, teacher created instructional materials	Websites, tutorials, massive open online courses
Questions	Teachers, classmates	Forums like Stack Overflow
Collaboration	Classmates	Online communities of novice programmers

Classroom vs. Independently

	Classroom	Independently
Motivation	Grades, Teachers	Novice programming environments (like Scratch, Alice, Looking Glass)
Materials	Books, teacher created instructional materials	Websites, tutorials, massive open online courses
Questions	Teachers, classmates	Forums like Stack Overflow
Collaboration	Classmates	Online communities of novice programmers
Code Review	Teachers	??

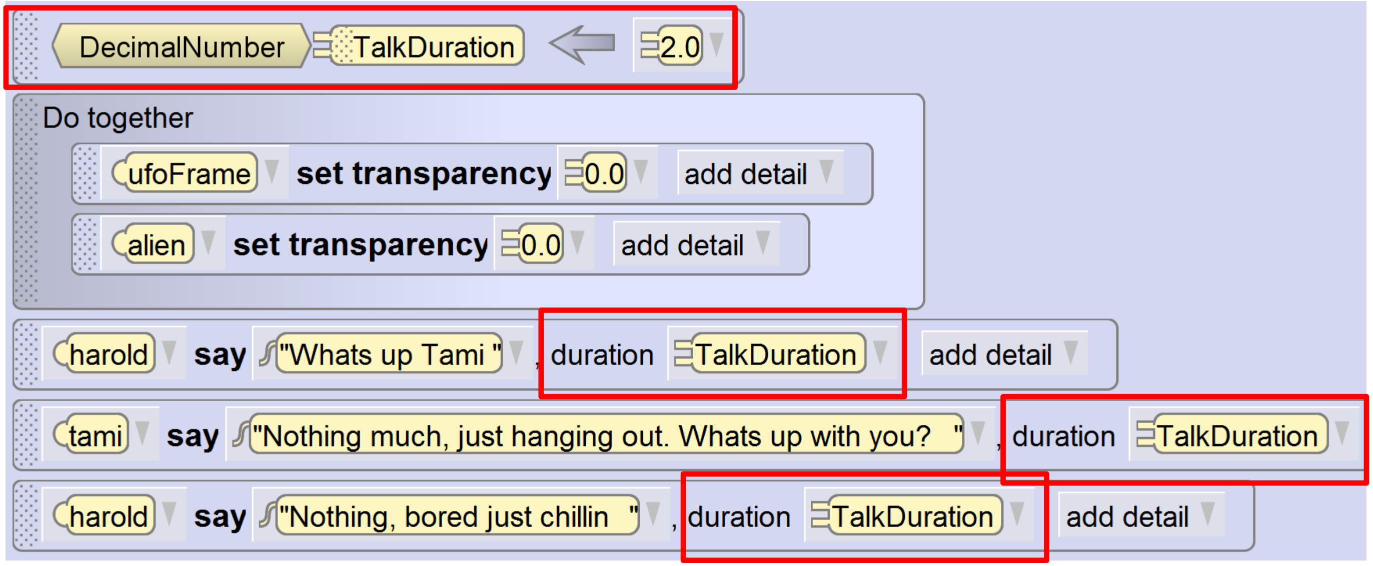
Code Review

Original Program



```
Do together
    ufoFrame set transparency 0.0 add detail
    alien set transparency 0.0 add detail
harold say "Whats up Tami " , duration 2.0 add detail
tami say "Nothing much, just hanging out. Whats up with you? " , duration 2.0 add detail
harold say "Nothing, bored just chillin " , duration 2.0 add detail
```

Suggestion: introduce variable



```
DecimalNumber TalkDuration ← 2.0
Do together
    ufoFrame set transparency 0.0 add detail
    alien set transparency 0.0 add detail
harold say "Whats up Tami " , duration TalkDuration add detail
tami say "Nothing much, just hanging out. Whats up with you? " , duration TalkDuration
harold say "Nothing, bored just chillin " , duration TalkDuration add detail
```

Experienced Programmers as Mentors

3 million professional programmers

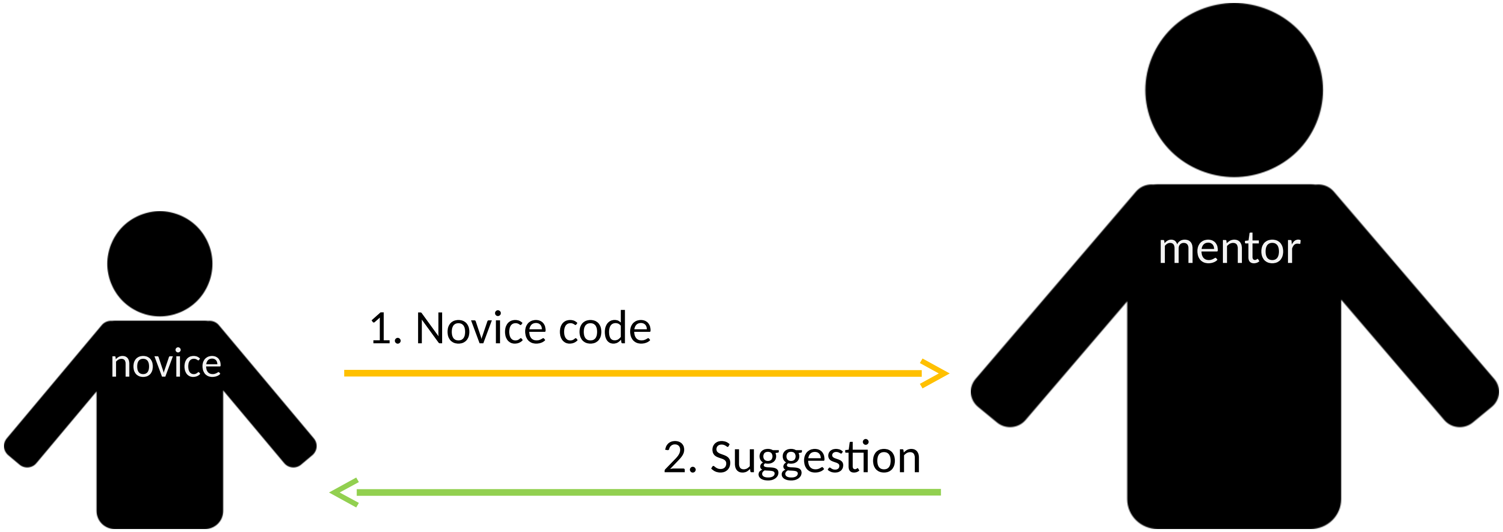
90 million end user programmers

+ 55 million students in k-12 education _

145 million potential novice programmers

~ 50 novices/experienced programmer

Experienced Programmer Mentoring

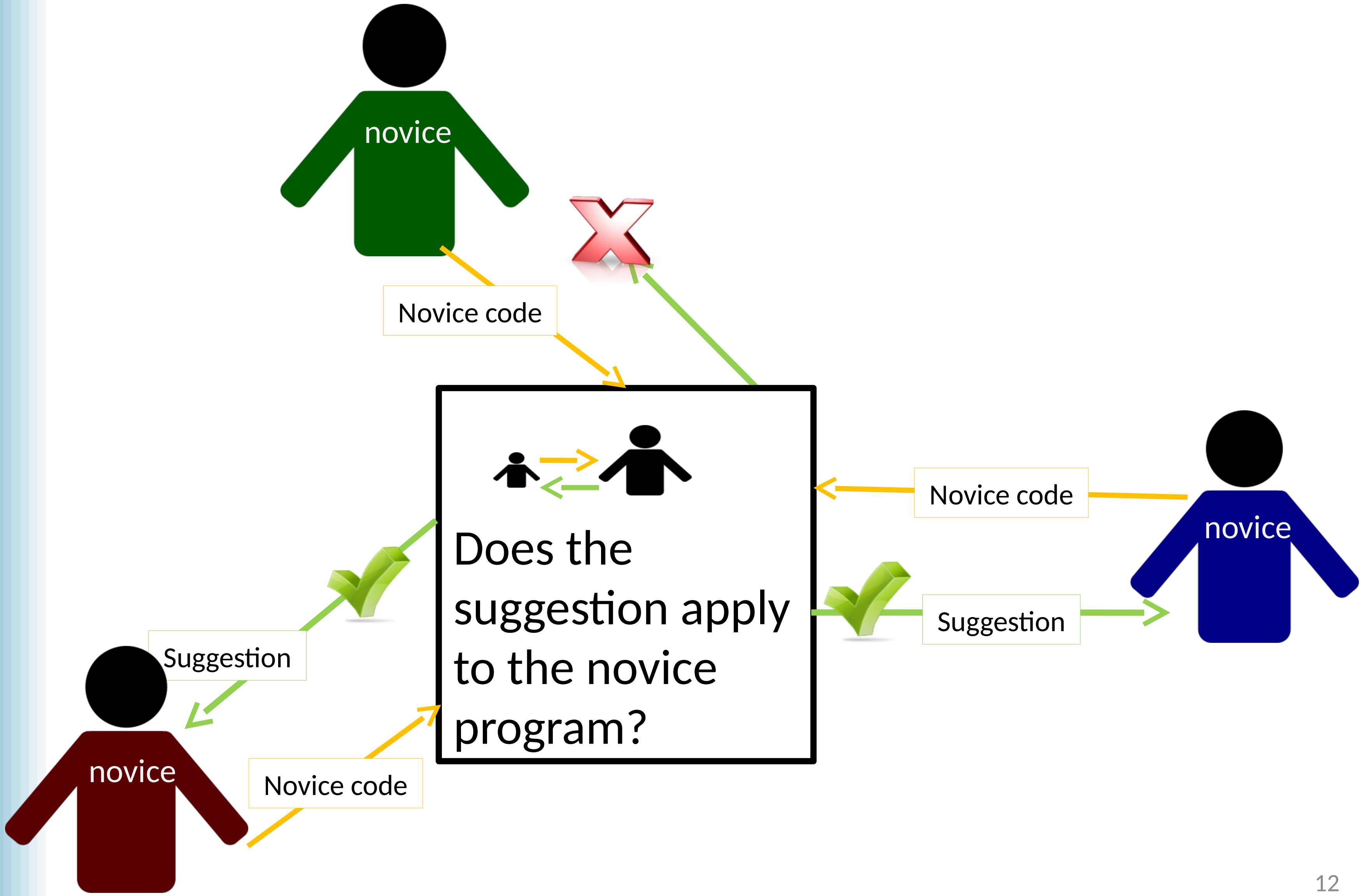

novice

1. Novice code

2. Suggestion

mentor

Large-Scale Mentoring



novice

Novice code

Does the suggestion apply to the novice program?

Novice code

novice

Suggestion

Suggestion

novice

Novice code

Exploratory Study Questions

- **Do mentor suggestions have the potential to teach novices to improve their code?**

Exploratory Study Questions

- Do mentor suggestions have the potential to teach novices to improve their code?
- **Can mentors write programs to generalize whether novice programs should receive certain suggestions?**

Exploratory Study Questions

- Do mentor suggestions have the potential to teach novices to improve their code?
- Can mentors write programs to generalize whether novice programs should receive certain suggestions?
- **What do mentors need in a tool to code these programs?**

Related Work

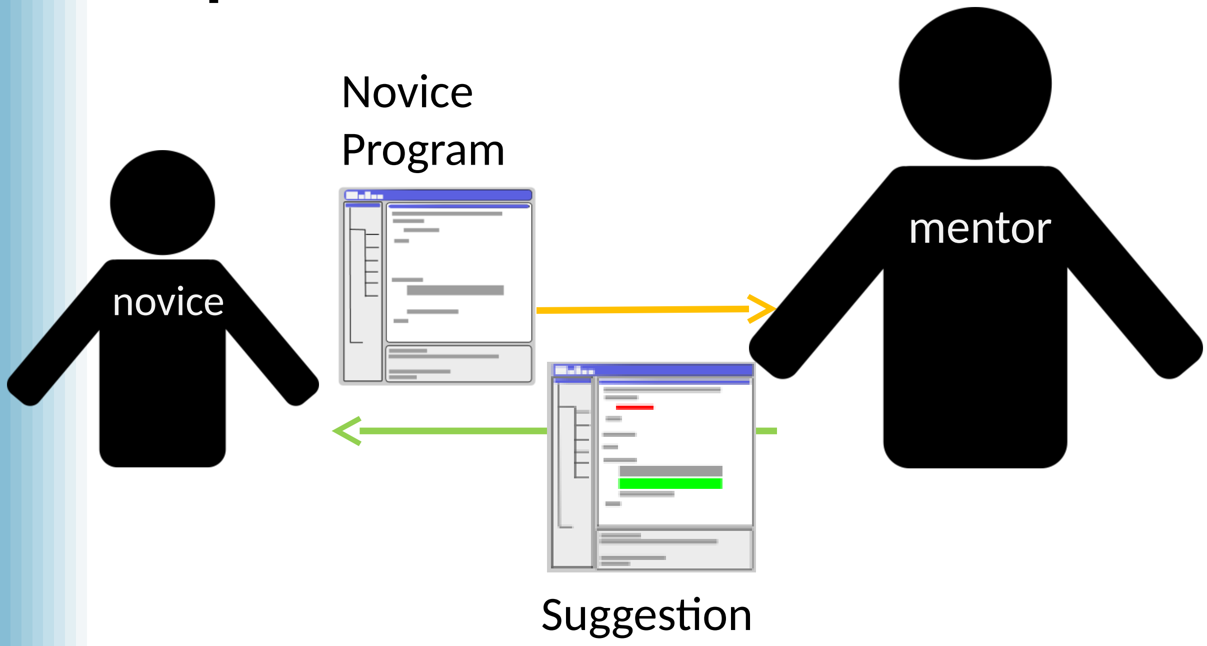
Finding Code to Improve

- Crowdsourced bug fixes
- Static Code analysis
- Code smell detection
- Automated grading systems

□ Find and fix bugs and errors

□ Look at issues beyond the scope of many novice programs

Proposed Crowdsourced Code Review



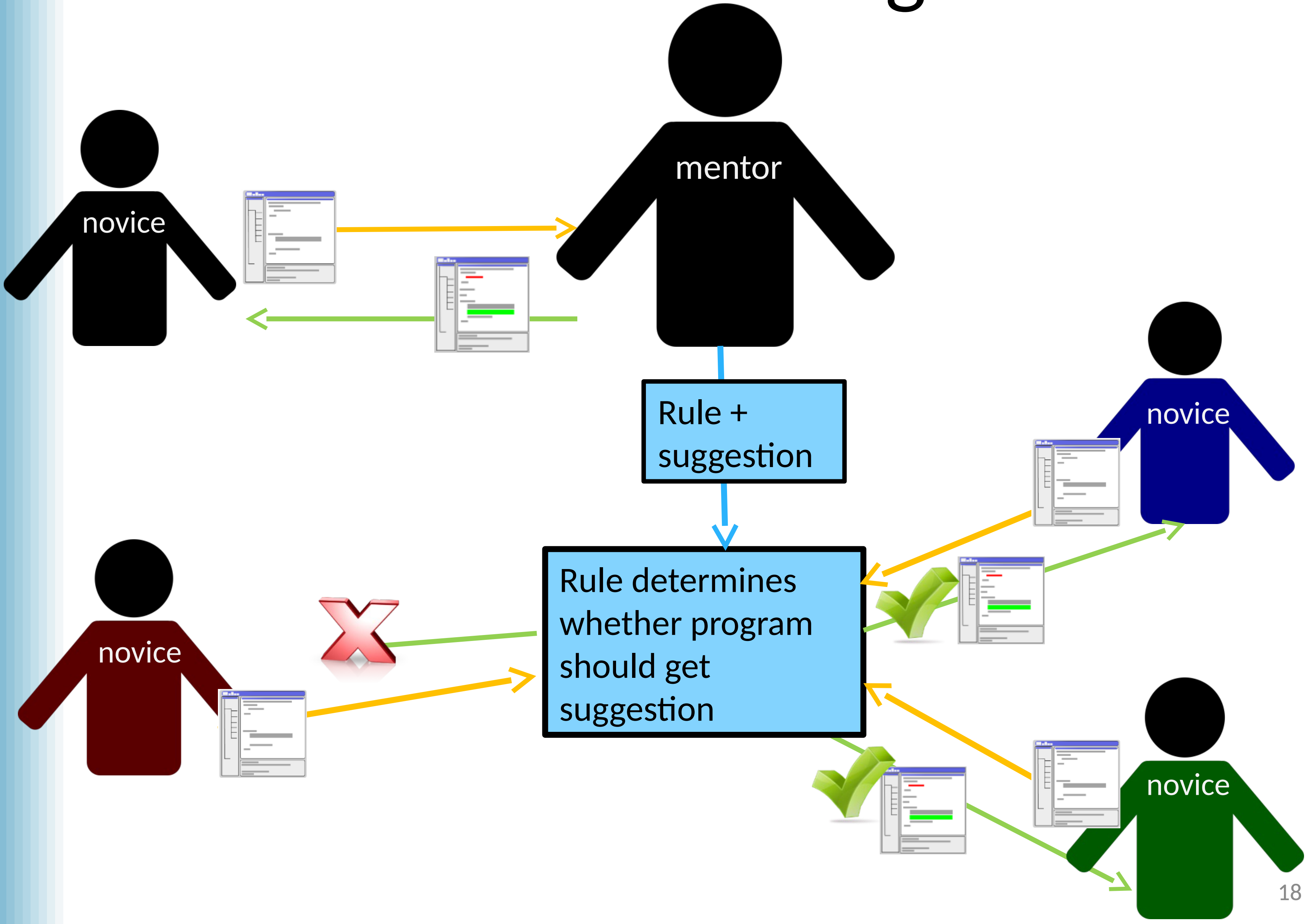
Novice Program

novice

mentor

Suggestion

Code Review at a Large Scale



Rule + suggestion

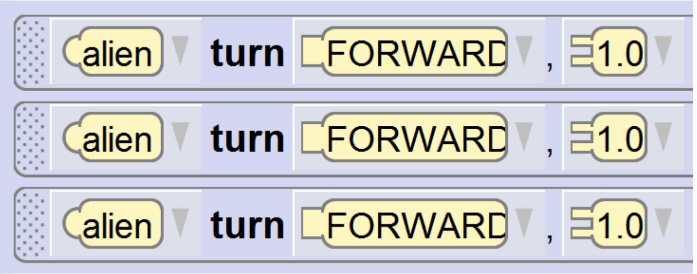
Rule determines whether program should get suggestion

DEFINITIONS!

- Suggestion
- Rule

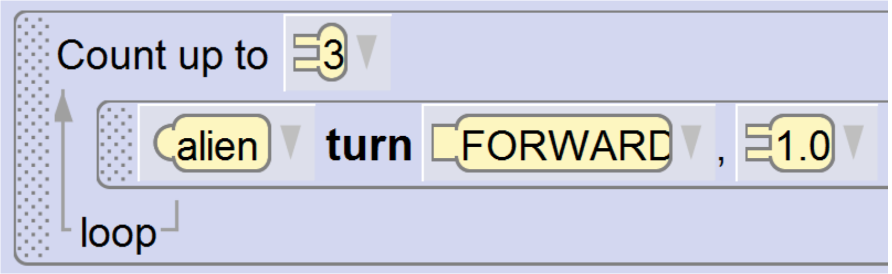
What is a Suggestion?

Original Novice Code



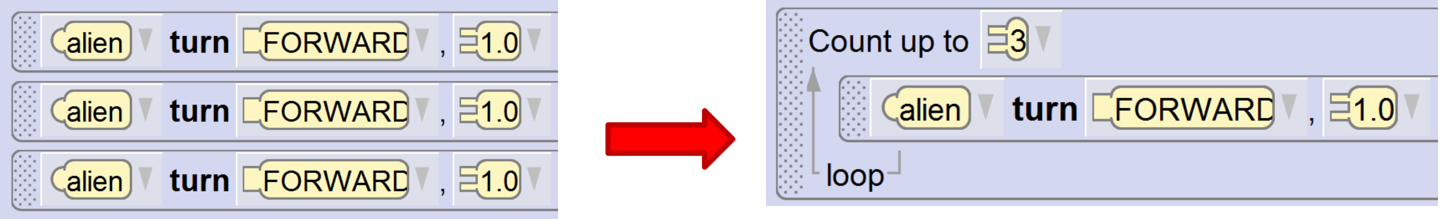
```
alien turn FORWARD , 1.0
alien turn FORWARD , 1.0
alien turn FORWARD , 1.0
```

Improved Code



```
Count up to 3
    alien turn FORWARD , 1.0
loop
```

What is a Rule?

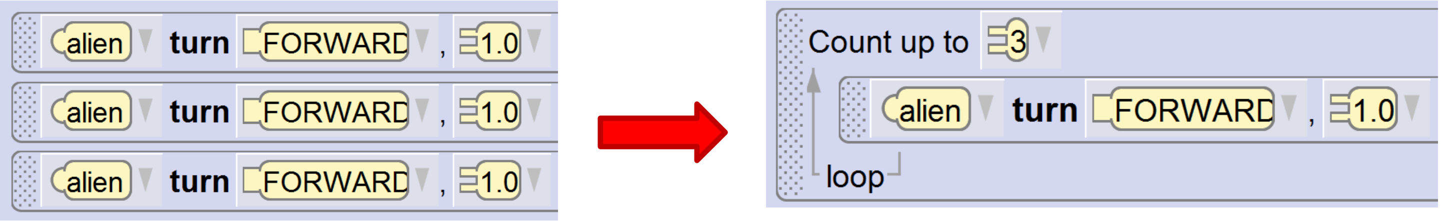


```
alien turn FORWARD , 1.0
alien turn FORWARD , 1.0
alien turn FORWARD , 1.0
```

```
Count up to 3
    alien turn FORWARD , 1.0
loop
```

```
 foreach(Statement s: statements){
 if(prevStatement.isIdenticalTo(s)){
       return true;
    }
 prevStatement = s;
}
return false;
```

What is a Rule?

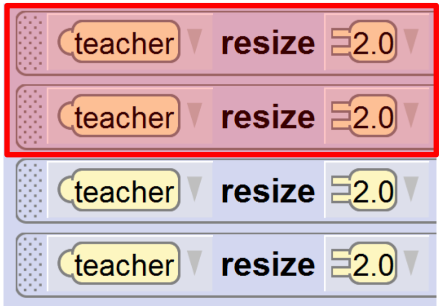


```
alien turn FORWARD , 1.0
alien turn FORWARD , 1.0
alien turn FORWARD , 1.0
```

```
Count up to 3
    alien turn FORWARD , 1.0
loop
```

```
foreach(Statement s: statements){
if(prevStatement.isIdenticalTo(s)){
          return true;
     }
prevStatement = s;
}
return false;
```

Novice code submission



```
teacher resize 2.0
teacher resize 2.0
teacher resize 2.0
teacher resize 2.0
```

Count loop suggestion

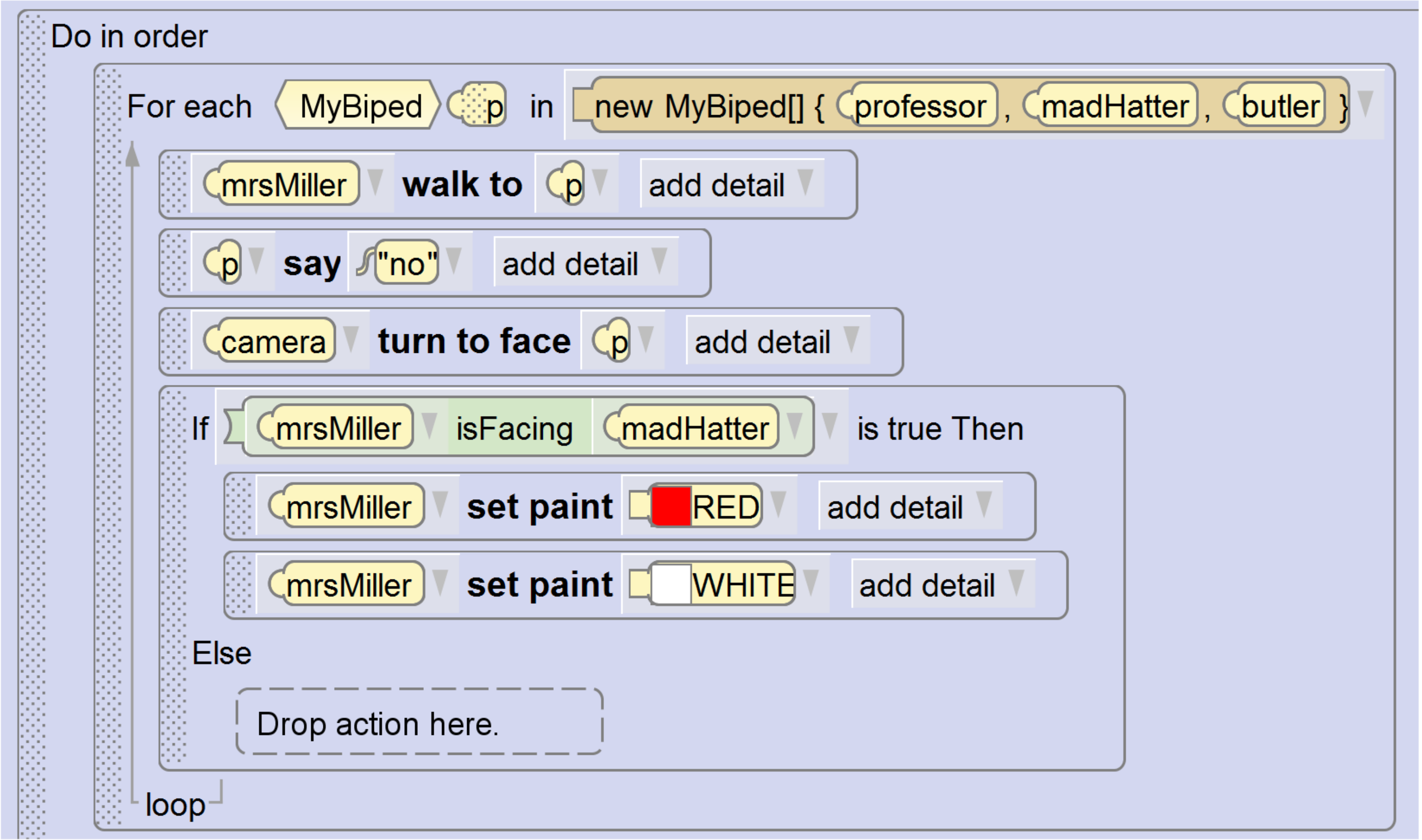
Exploratory Study Questions

- Do mentor suggestions have the potential to teach novices to improve their code?
- Can mentors write rules to generalize whether novice programs should receive certain suggestions?
- What do mentors need in a tool to code these rules?

Exploratory Study: Participants

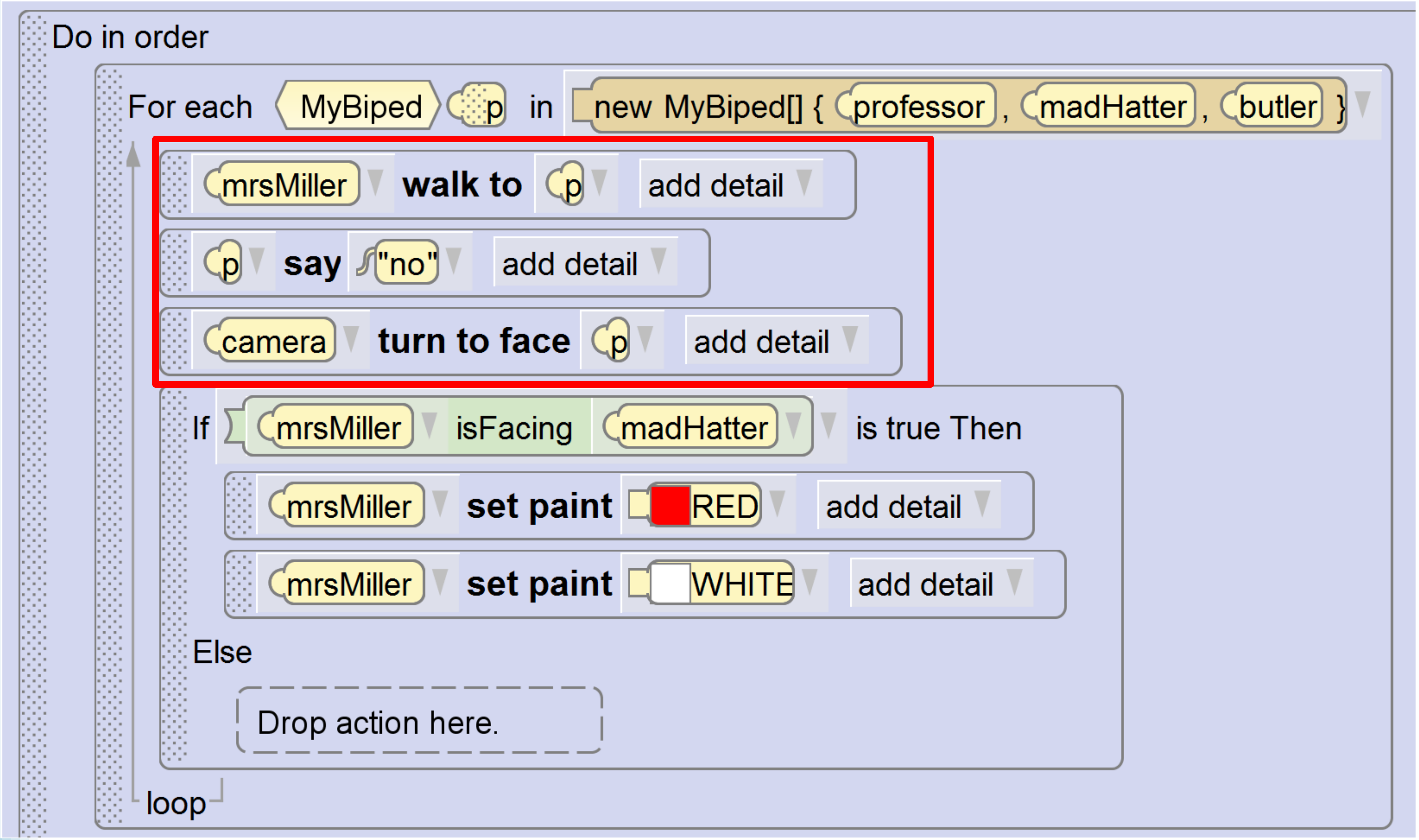
- 21 participants
  - Avg. 15 years of programming experience (std. 10)
  - Software developers, software engineers and programmers
  - age 19 to 68
  - 5 female, 16 male

Looking Glass



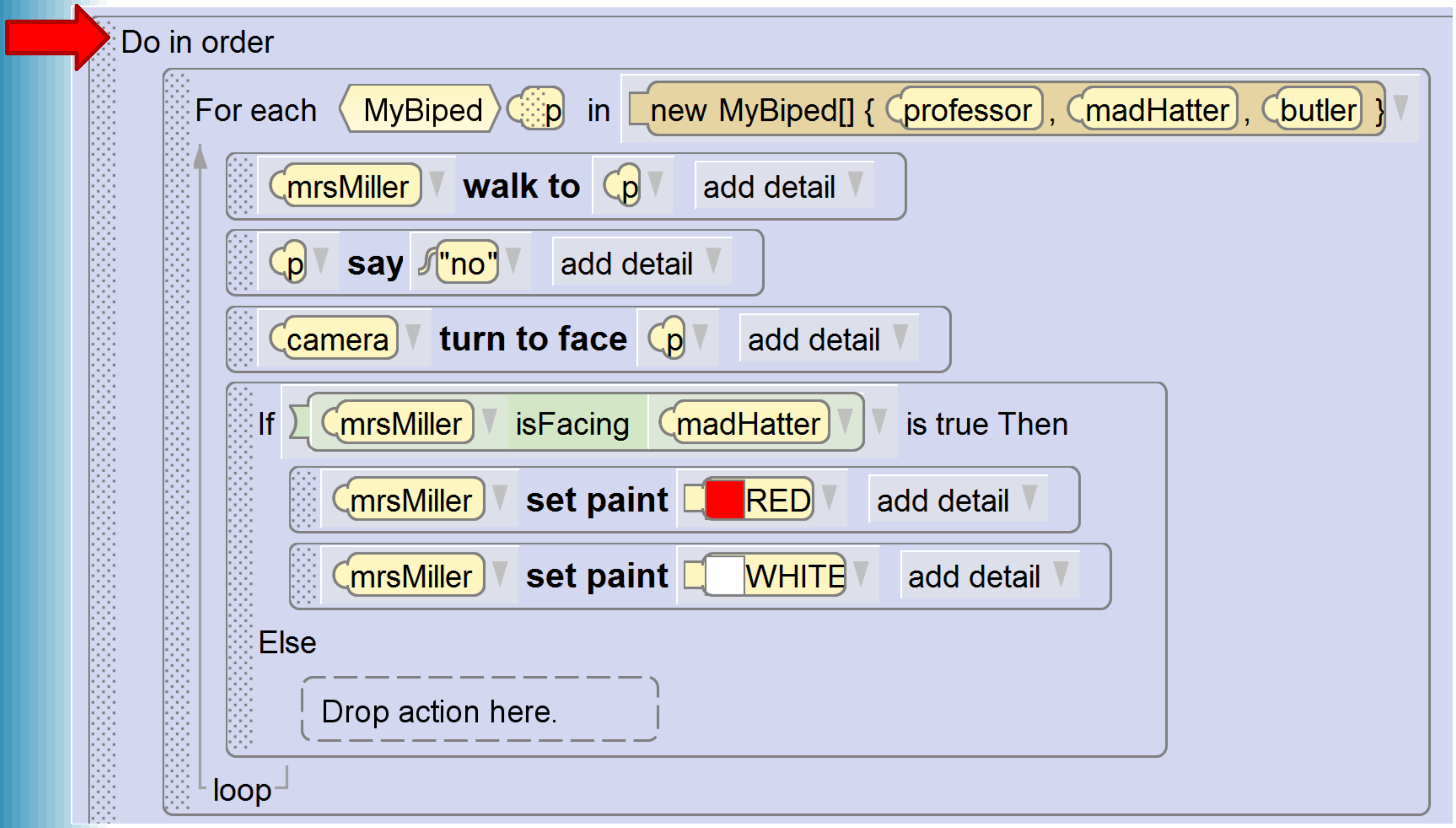
```
Do in order
    For each MyBiped p in new MyBiped[] { professor , madHatter , butler }
        mrsMiller walk to p add detail
        p say "no" add detail
        camera turn to face p add detail
        If mrsMiller isFacing madHatter is true Then
            mrsMiller set paint 🟥RED add detail
            mrsMiller set paint ⬜WHITE add detail
        Else
            Drop action here.
    loop
```

Looking Glass



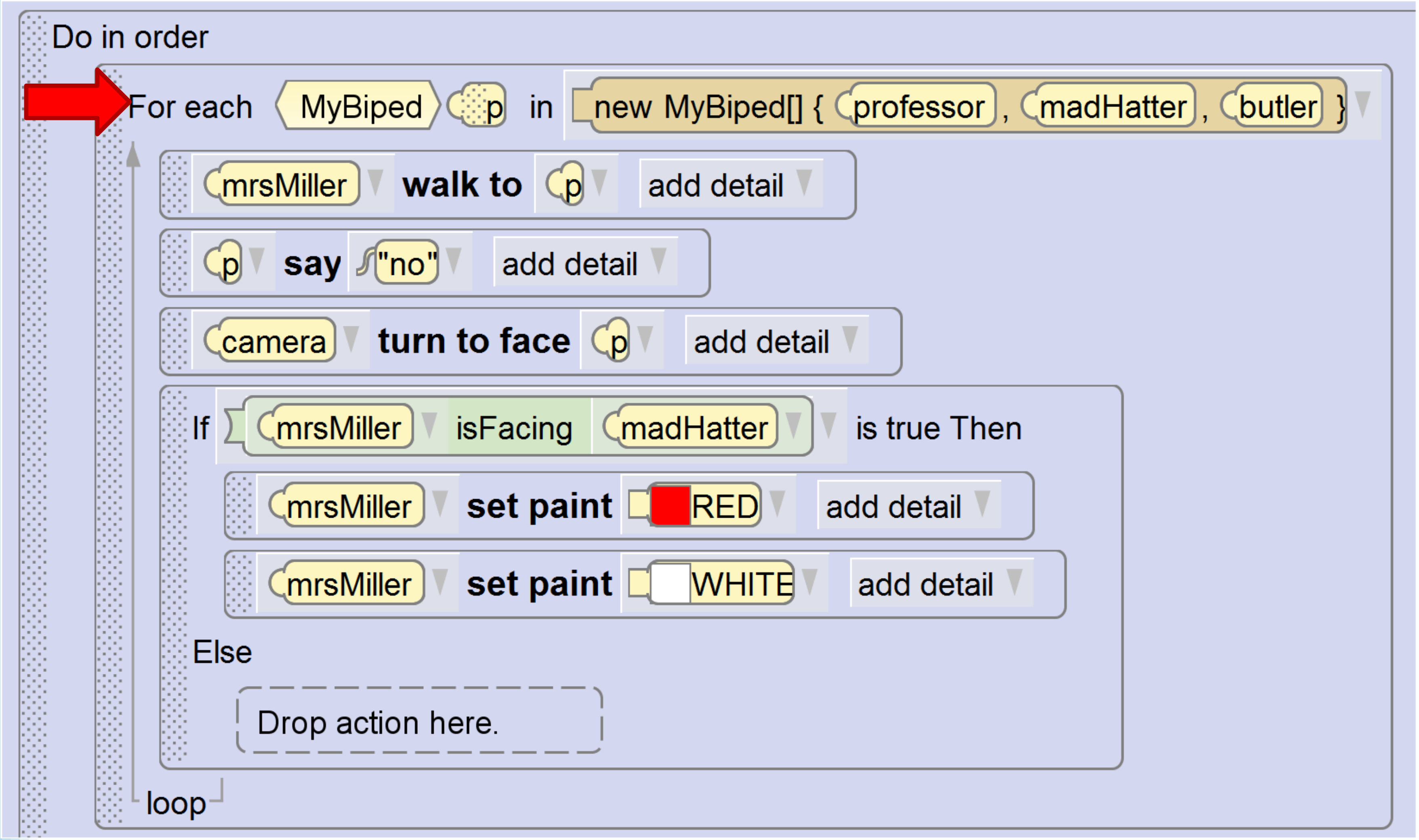
```
Do in order
    For each MyBiped p in new MyBiped[] { professor , madHatter , butler }
        mrsMiller walk to p add detail
        p say "no" add detail
        camera turn to face p add detail
        If mrsMiller isFacing madHatter is true Then
            mrsMiller set paint RED add detail
            mrsMiller set paint WHITE add detail
        Else
            Drop action here.
    loop
```

Looking Glass



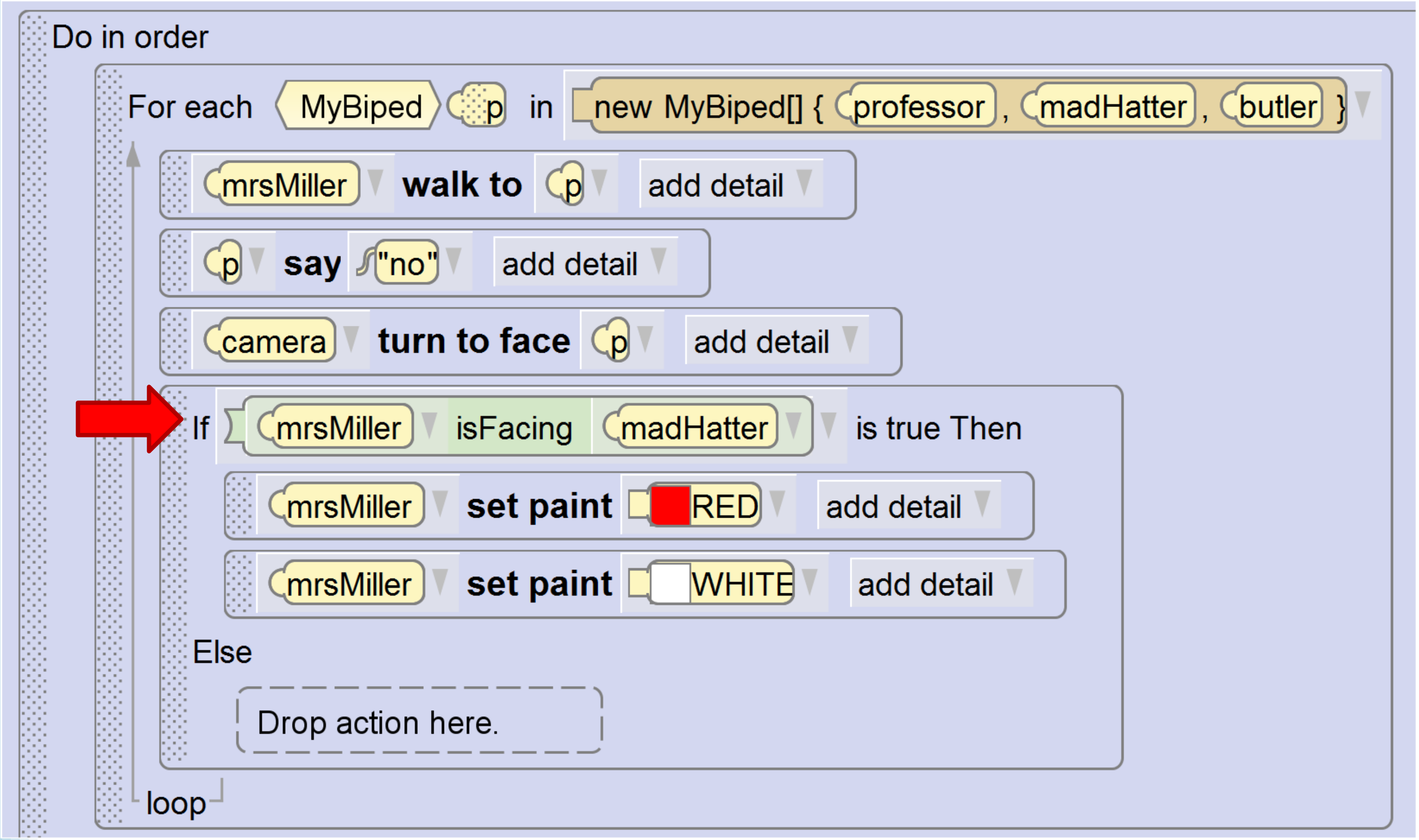
```
Do in order
    For each MyBiped p in new MyBiped[] { professor , madHatter , butler }
        mrsMiller walk to p add detail
        p say "no" add detail
        camera turn to face p add detail
        If mrsMiller isFacing madHatter is true Then
            mrsMiller set paint RED add detail
            mrsMiller set paint WHITE add detail
        Else
            Drop action here.
    loop
```

Looking Glass



```
Do in order
    For each MyBiped p in new MyBiped[] { professor , madHatter , butler }
        mrsMiller walk to p add detail
        p say "no" add detail
        camera turn to face p add detail
        If mrsMiller isFacing madHatter is true Then
            mrsMiller set paint 🟥RED add detail
            mrsMiller set paint ⬜WHITE add detail
        Else
            Drop action here.
    loop
```

Looking Glass



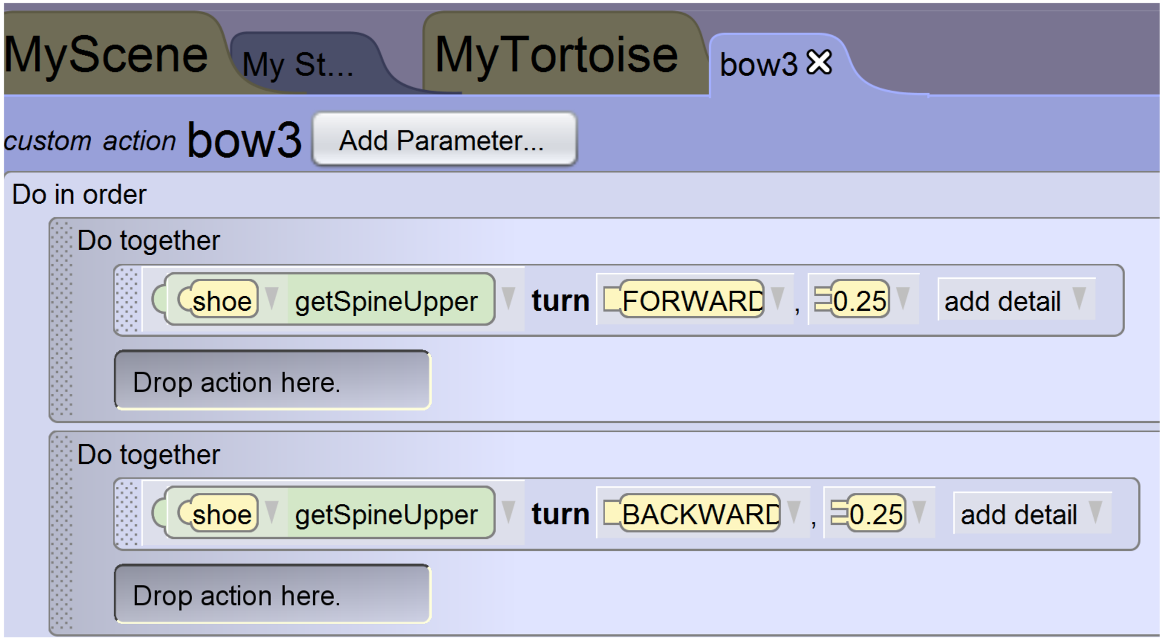
```
Do in order
    For each MyBiped p in new MyBiped[] { professor , madHatter , butler }
        mrsMiller walk to p add detail
        p say "no" add detail
        camera turn to face p add detail
        If mrsMiller isFacing madHatter is true Then
            mrsMiller set paint 🟥RED add detail
            mrsMiller set paint ⬜WHITE add detail
        Else
            Drop action here.
    loop
```

Exploratory Study: Suggestions and Rules

- Two types of instructions:
  - Make a suggestion
  - Write a rule

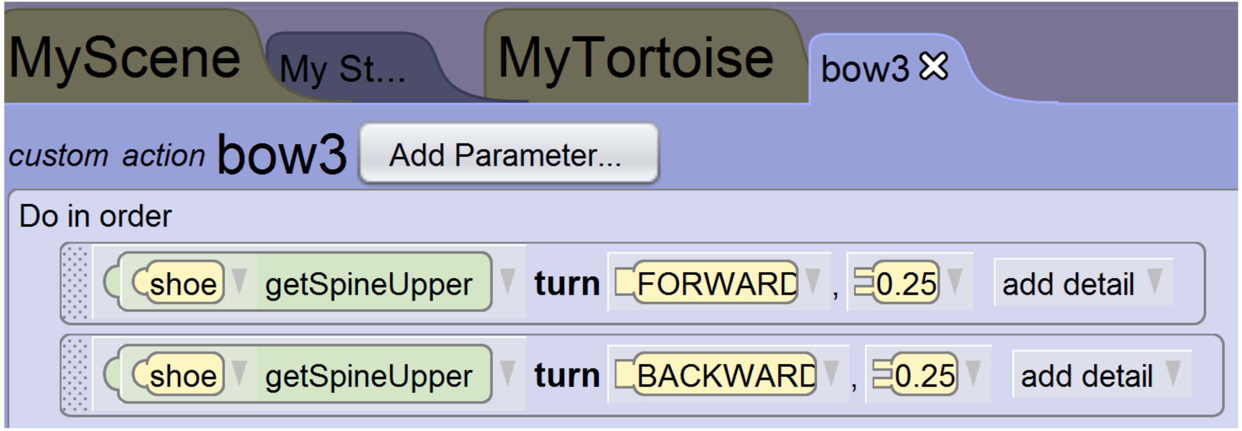
Make a Suggestion

Material given to participant



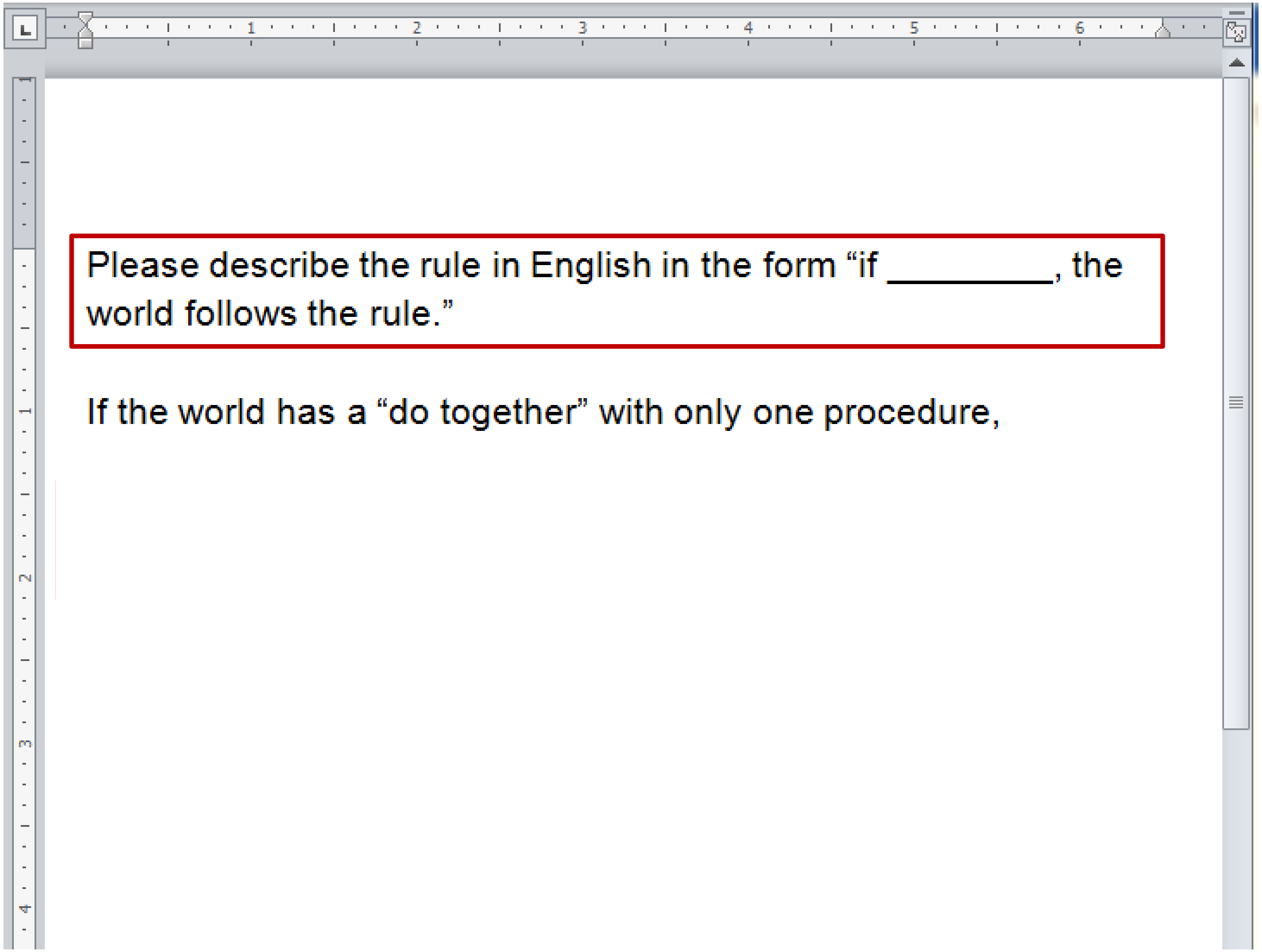
```
custom action bow3
Do in order
    Do together
        shoe getSpineUpper turn FORWARD , 0.25 add detail
        Drop action here.
    Do together
        shoe getSpineUpper turn BACKWARD , 0.25 add detail
        Drop action here.
```

Participant’s suggestion



```
custom action bow3
Do in order
    shoe getSpineUpper turn FORWARD , 0.25 add detail
    shoe getSpineUpper turn BACKWARD , 0.25 add detail
```

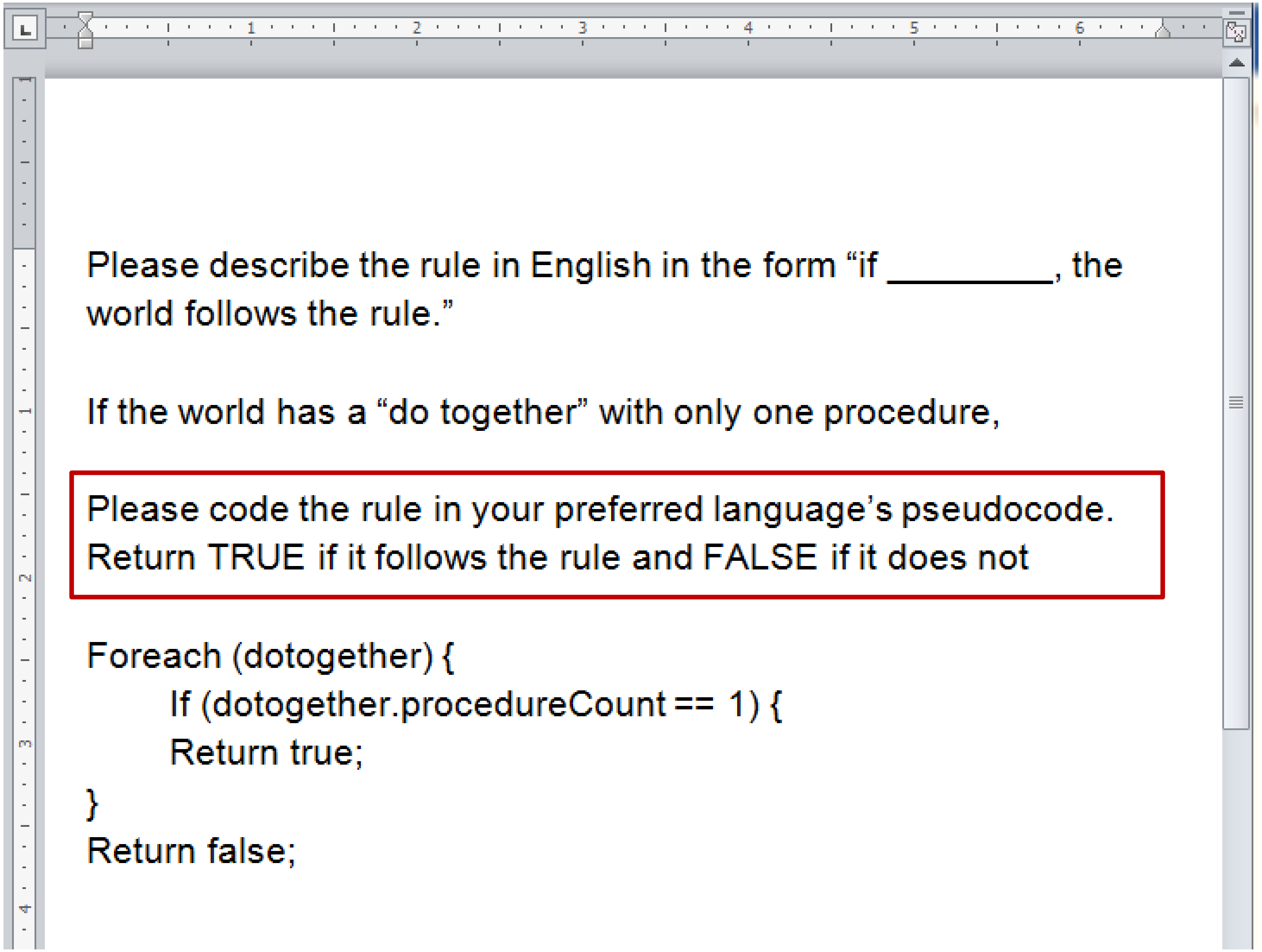
Write a Rule



Please describe the rule in English in the form “if __________, the world follows the rule.”

If the world has a “do together” with only one procedure,

Write a Rule



Please describe the rule in English in the form “if ________, the world follows the rule.”

If the world has a “do together” with only one procedure,

Please code the rule in your preferred language’s pseudocode. Return TRUE if it follows the rule and FALSE if it does not

```
Foreach (dotogether) {
        If (dotogether.procedureCount == 1) {
        Return true;
}
Return false;
```

Study format

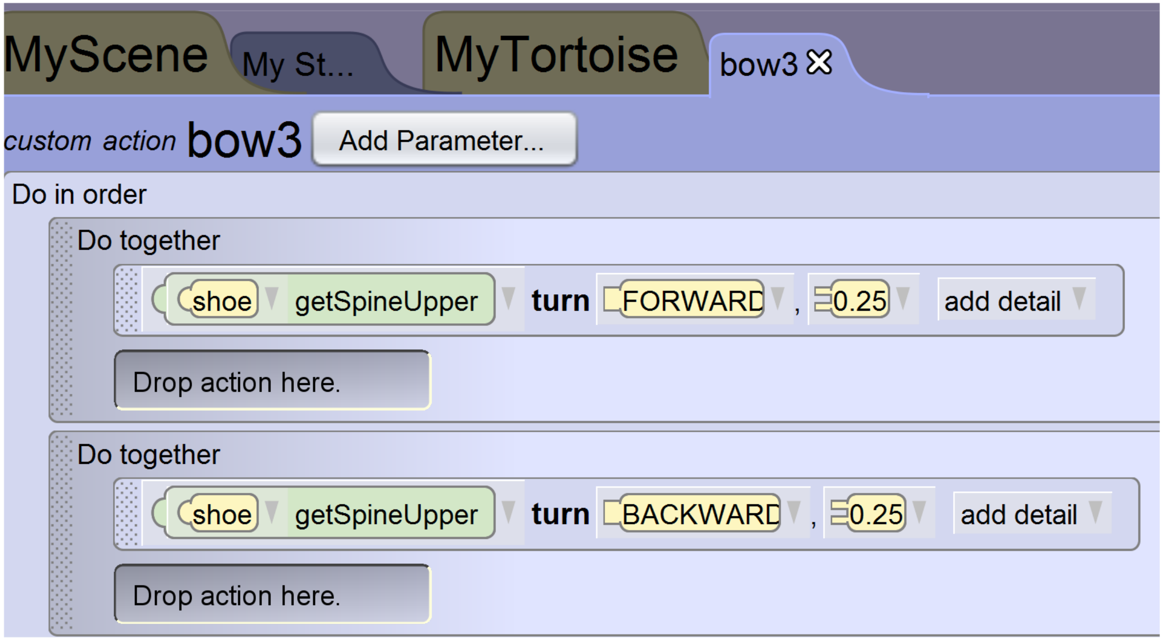
- Task 1:
  - Make a suggestion
  - Write a rule
- Tasks 2-4:
  - Write a rule (for a provided suggestion)

Exploratory Study Results

- **Do mentor suggestions have the potential to teach novices to improve their code?**
- Can mentors write rules to generalize whether novice programs should receive certain suggestions?
- What do mentors need in a tool to code these rules?

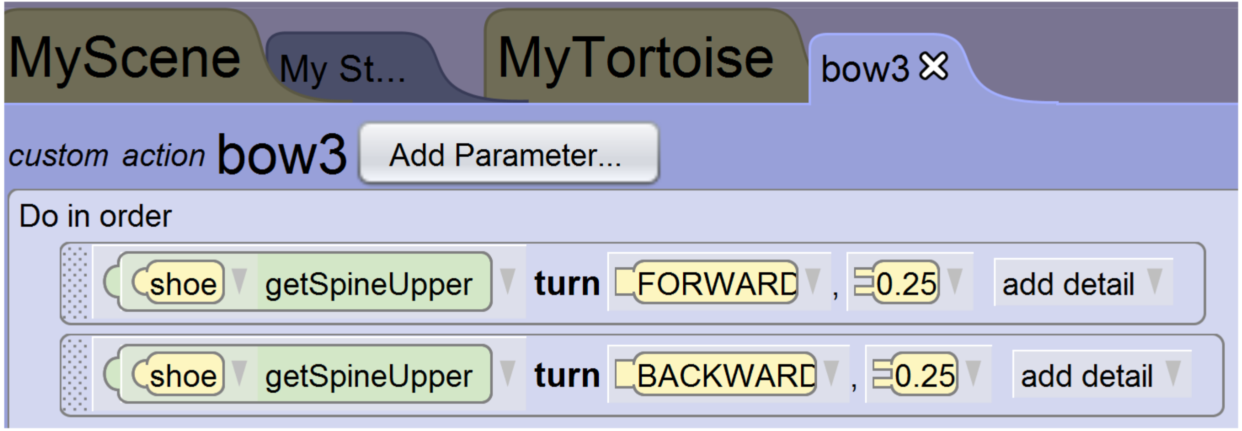
Make a Suggestion

Material given to participant

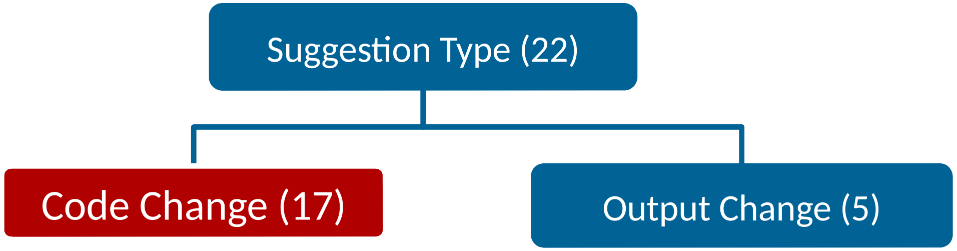


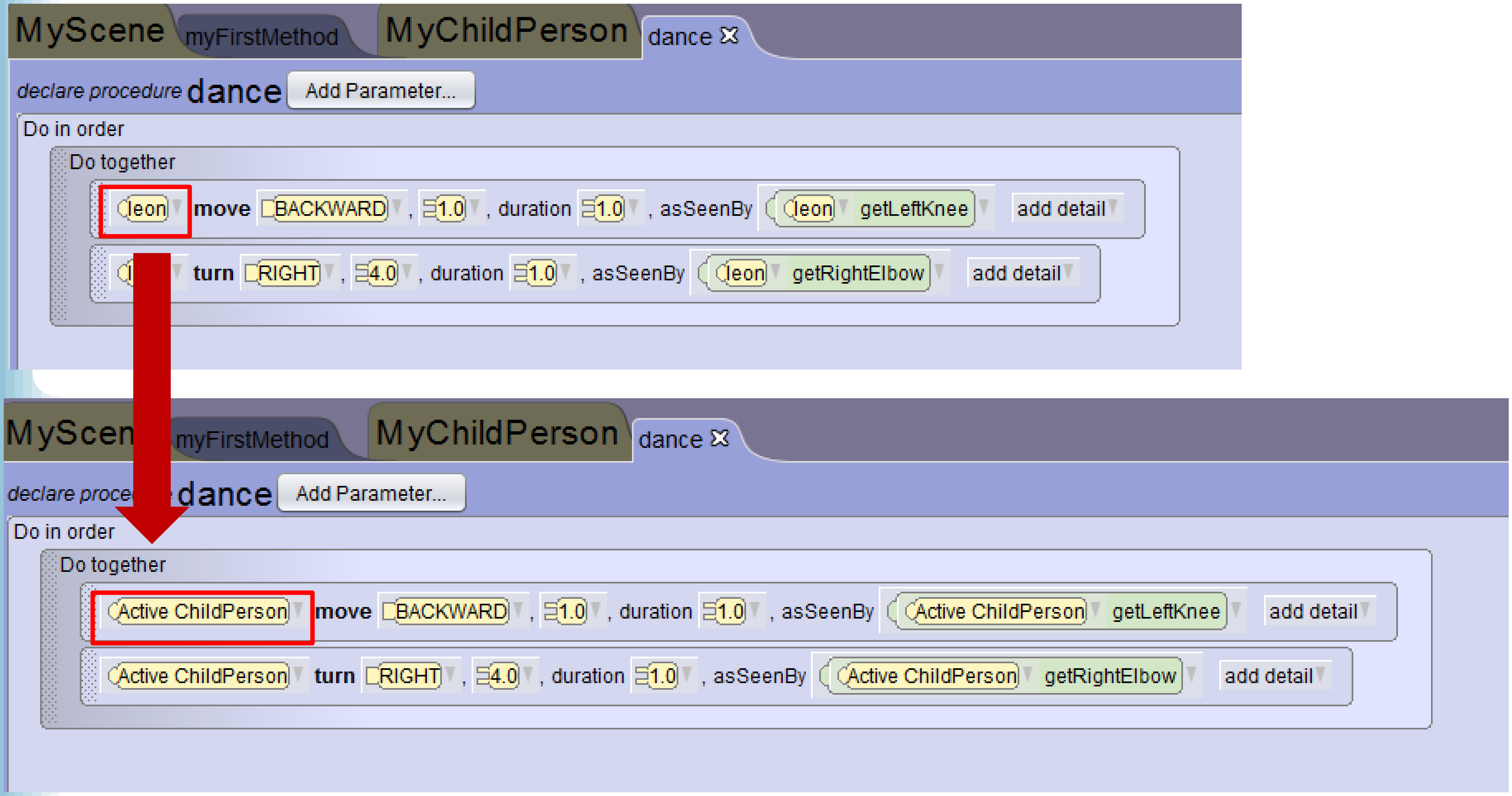
```
custom action bow3
Do in order
    Do together
        shoe getSpineUpper turn FORWARD , 0.25 add detail
        Drop action here.
    Do together
        shoe getSpineUpper turn BACKWARD , 0.25 add detail
        Drop action here.
```

Participant's suggestion



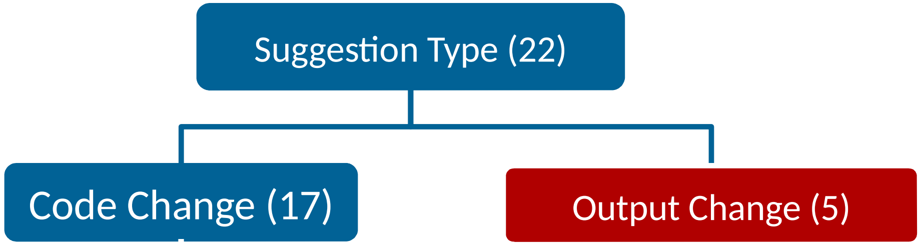
```
custom action bow3
Do in order
    shoe getSpineUpper turn FORWARD , 0.25 add detail
    shoe getSpineUpper turn BACKWARD , 0.25 add detail
```





```
declare procedure dance
Do in order
  Do together
    leon move BACKWARD , 1.0 , duration 1.0 , asSeenBy leon getLeftKnee
    l turn RIGHT , 4.0 , duration 1.0 , asSeenBy leon getRightElbow
```

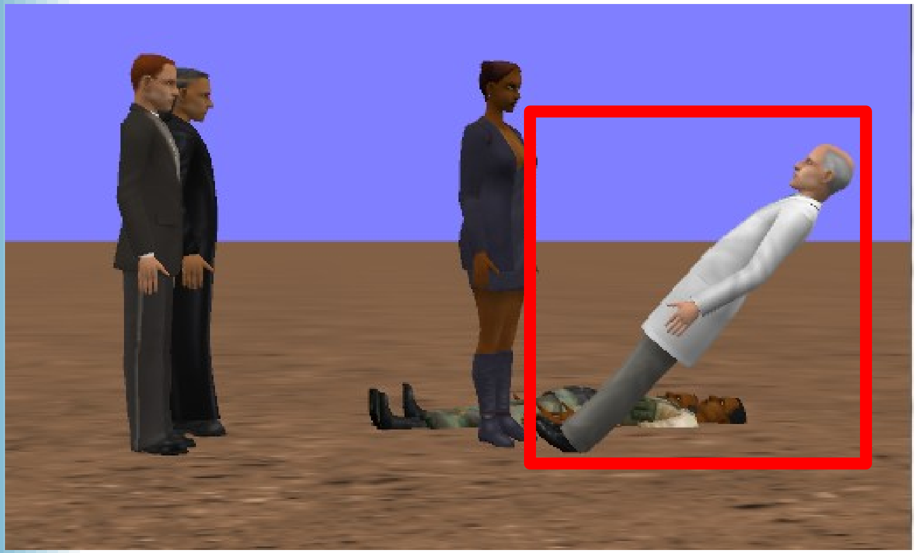
```
declare proce dance
Do in order
  Do together
    Active ChildPerson move BACKWARD , 1.0 , duration 1.0 , asSeenBy Active ChildPerson getLeftKnee
    Active ChildPerson turn RIGHT , 4.0 , duration 1.0 , asSeenBy Active ChildPerson getRightElbow
```

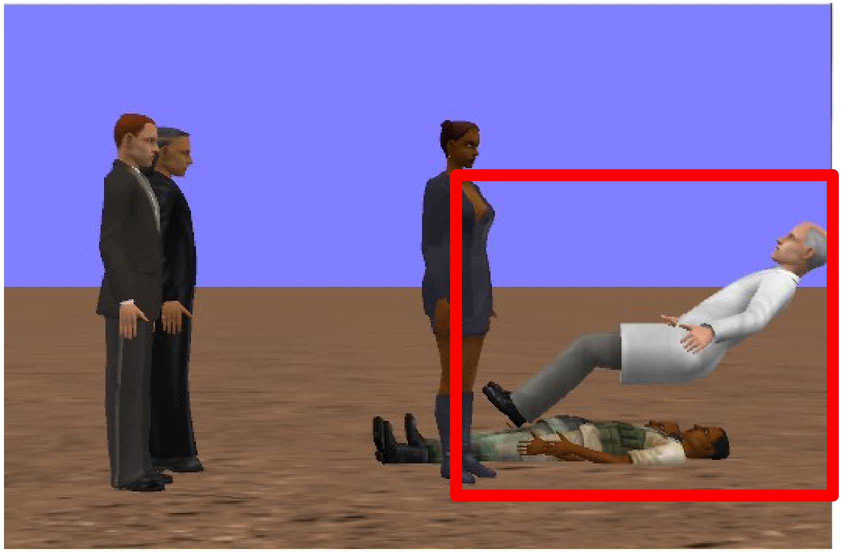


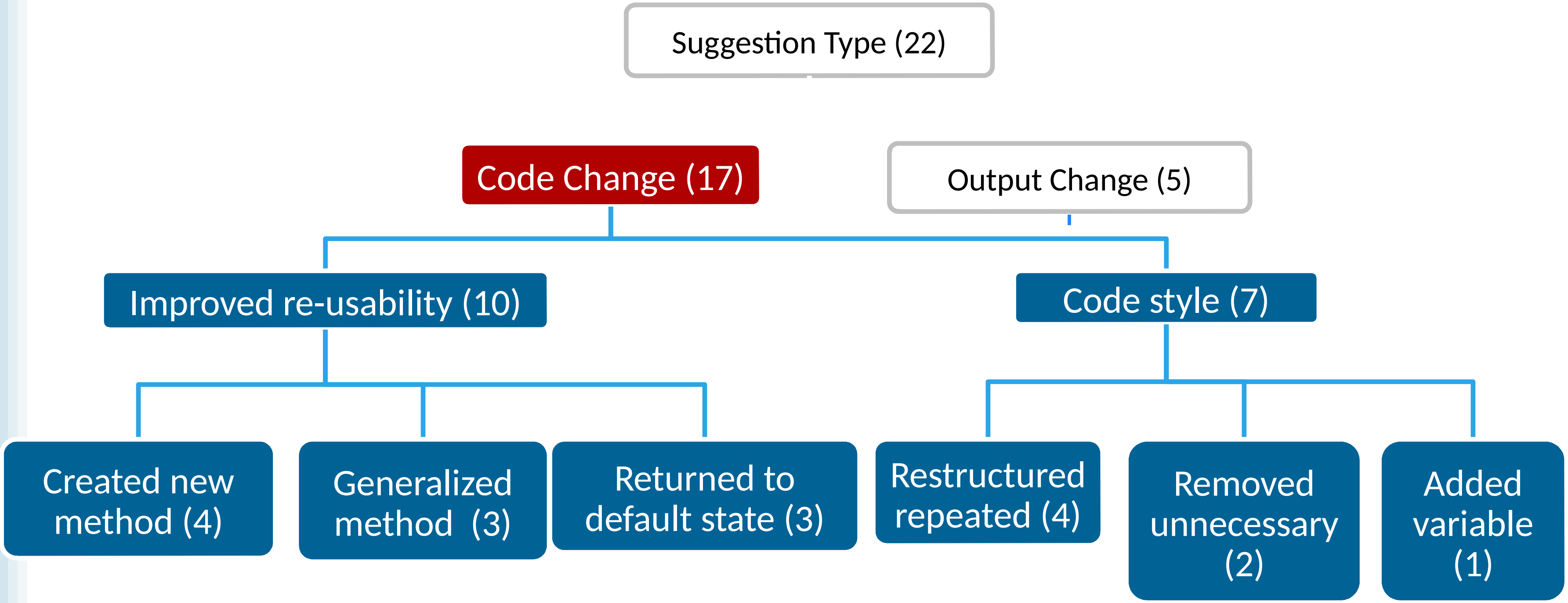
Suggestion Type (22)

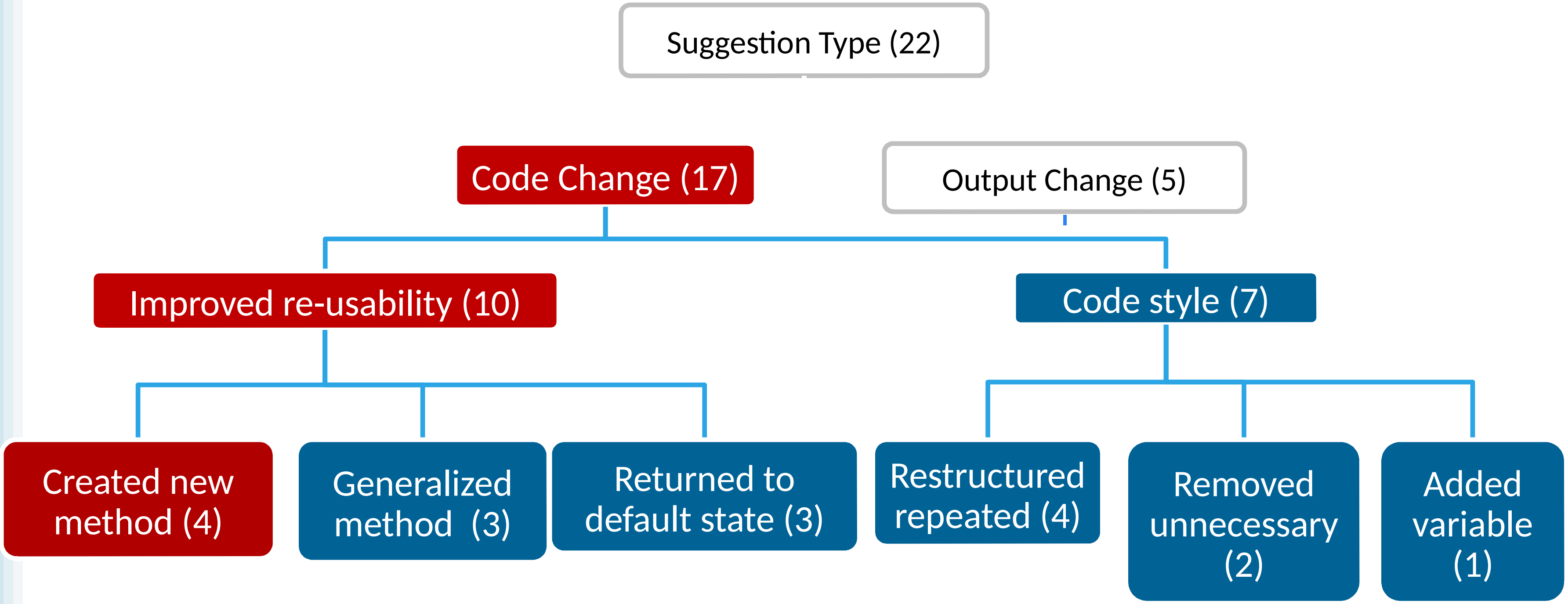
Code Change (17)

Output Change (5)






Suggestion Type (22)
Code Change (17)
Output Change (5)
Improved re-usability (10)
Code style (7)
Created new method (4)
Generalized method (3)
Returned to default state (3)
Restructured repeated (4)
Removed unnecessary (2)
Added variable (1)



Suggestion Type (22)

Code Change (17)

Output Change (5)

Improved re-usability (10)

Code style (7)

Created new method (4)

Generalized method (3)

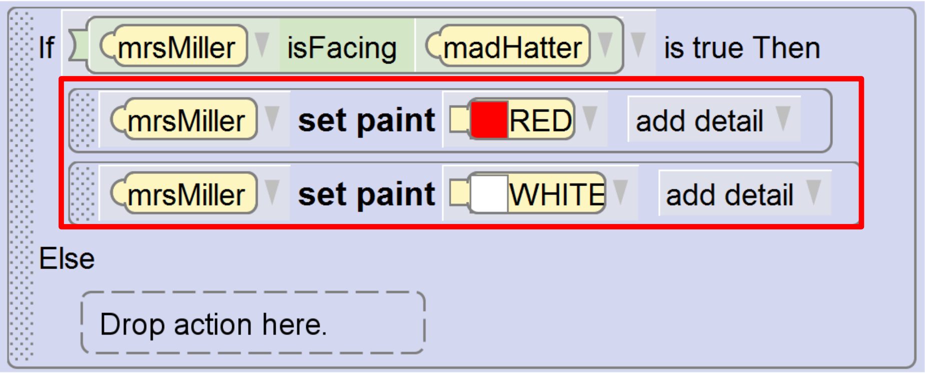
Returned to default state (3)

Restructured repeated (4)

Removed unnecessary (2)

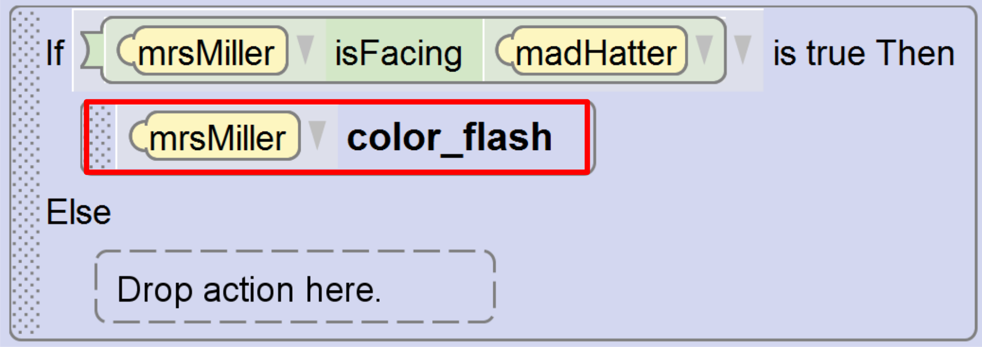
Added variable (1)

Original Program

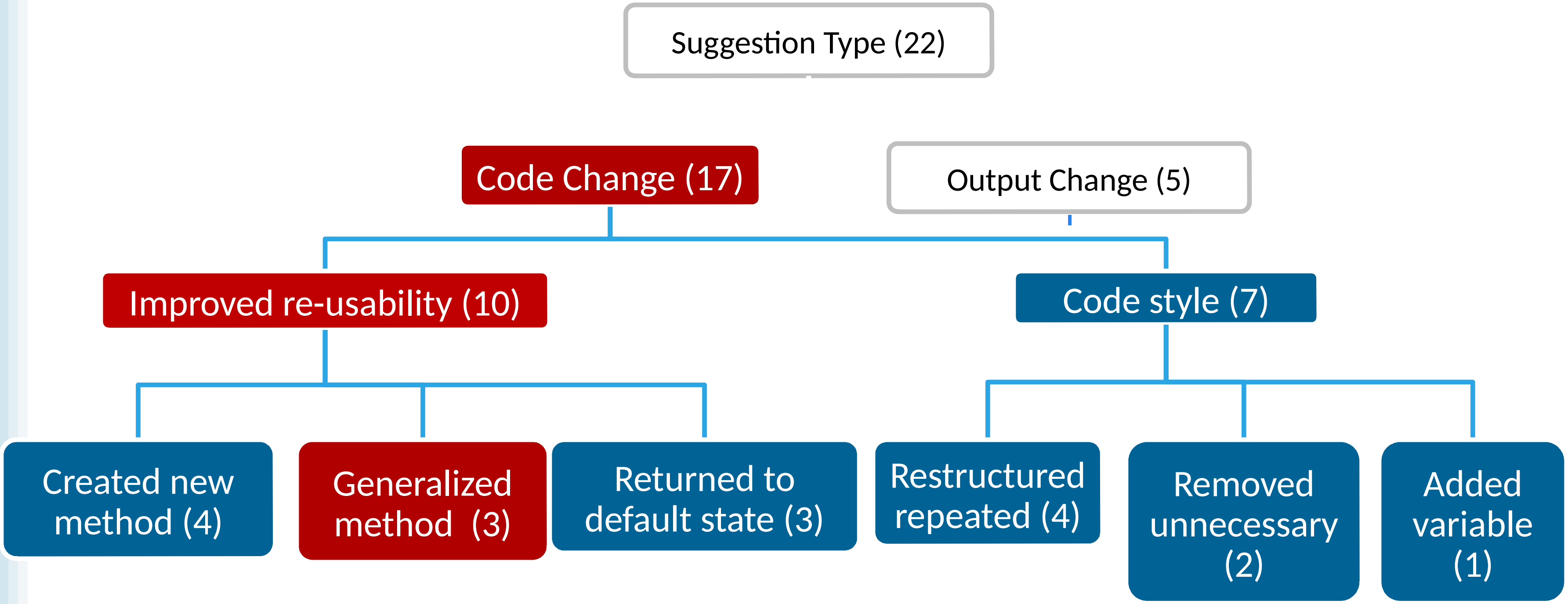


```
If  mrsMiller isFacing madHatter is true Then
    mrsMiller set paint RED add detail
    mrsMiller set paint WHITE add detail
Else
    Drop action here.
```

Suggested Change



```
If  mrsMiller isFacing madHatter is true Then
    mrsMiller color_flash
Else
    Drop action here.
```



Suggestion Type (22)

Code Change (17)

Output Change (5)

Improved re-usability (10)

Code style (7)

Created new method (4)

Generalized method (3)

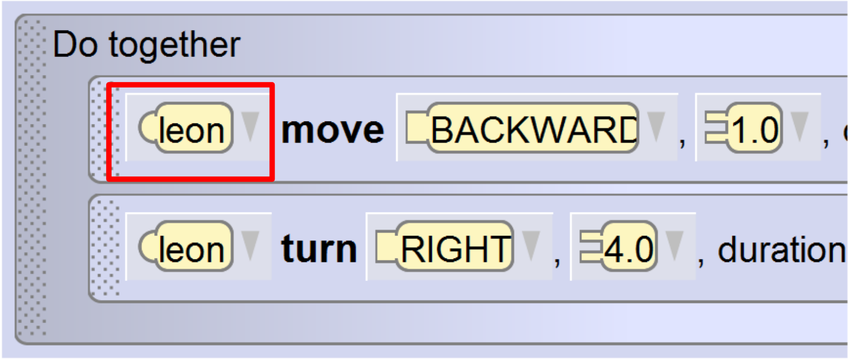
Returned to default state (3)

Restructured repeated (4)

Removed unnecessary (2)

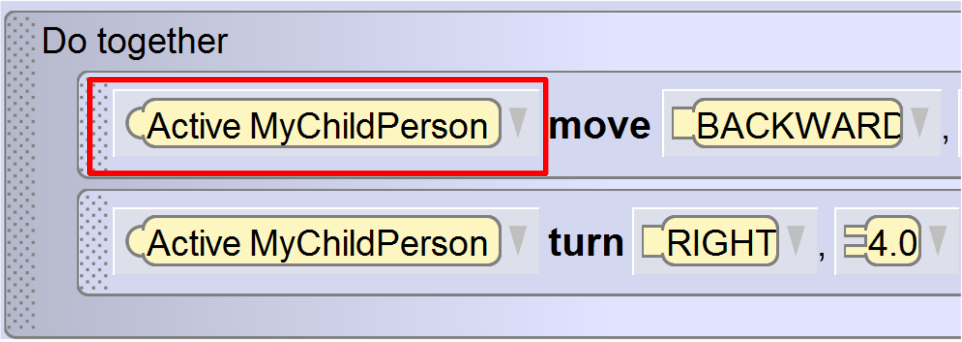
Added variable (1)

Original Program

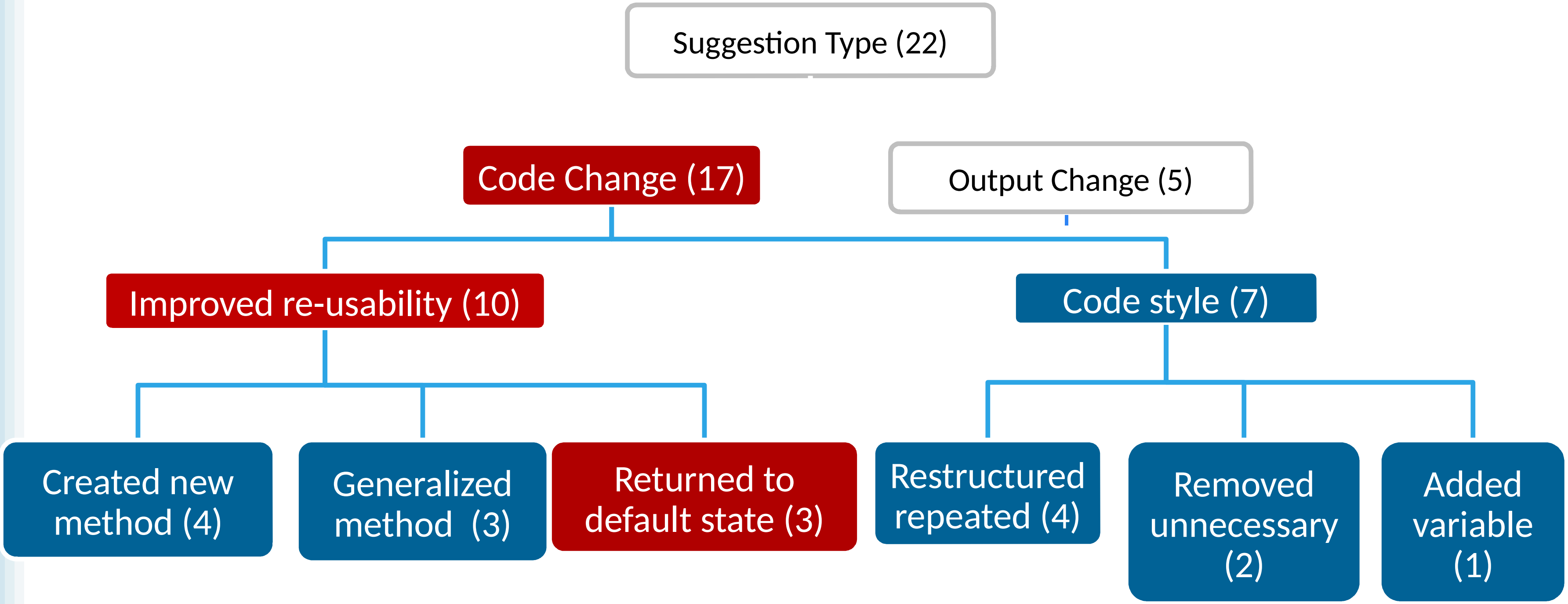


```
Do together
    leon move BACKWARD , 1.0 , (
    leon turn RIGHT , 4.0 , duration
```

Suggested Change



```
Do together
    Active MyChildPerson move BACKWARD ,
    Active MyChildPerson turn RIGHT , 4.0
```



Suggestion Type (22)

Code Change (17)

Output Change (5)

Improved re-usability (10)

Code style (7)

Created new method (4)

Generalized method (3)

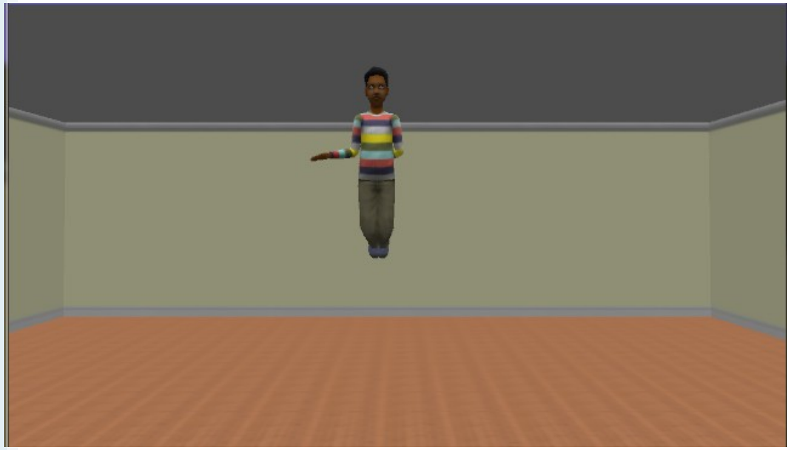
Returned to default state (3)

Restructured repeated (4)

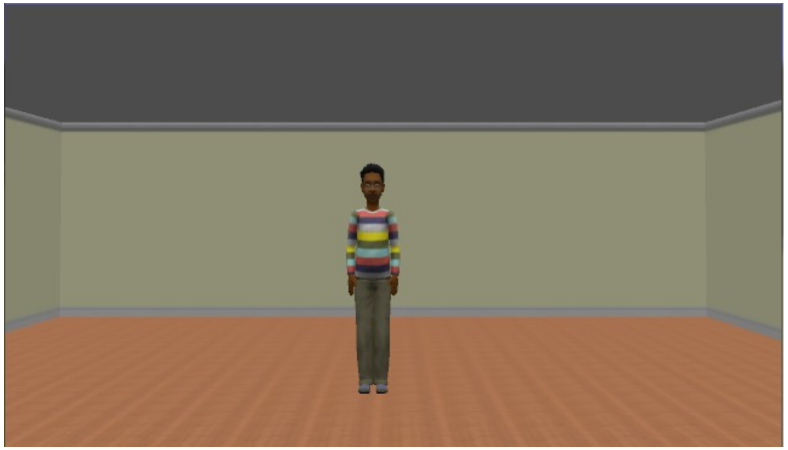
Removed unnecessary (2)

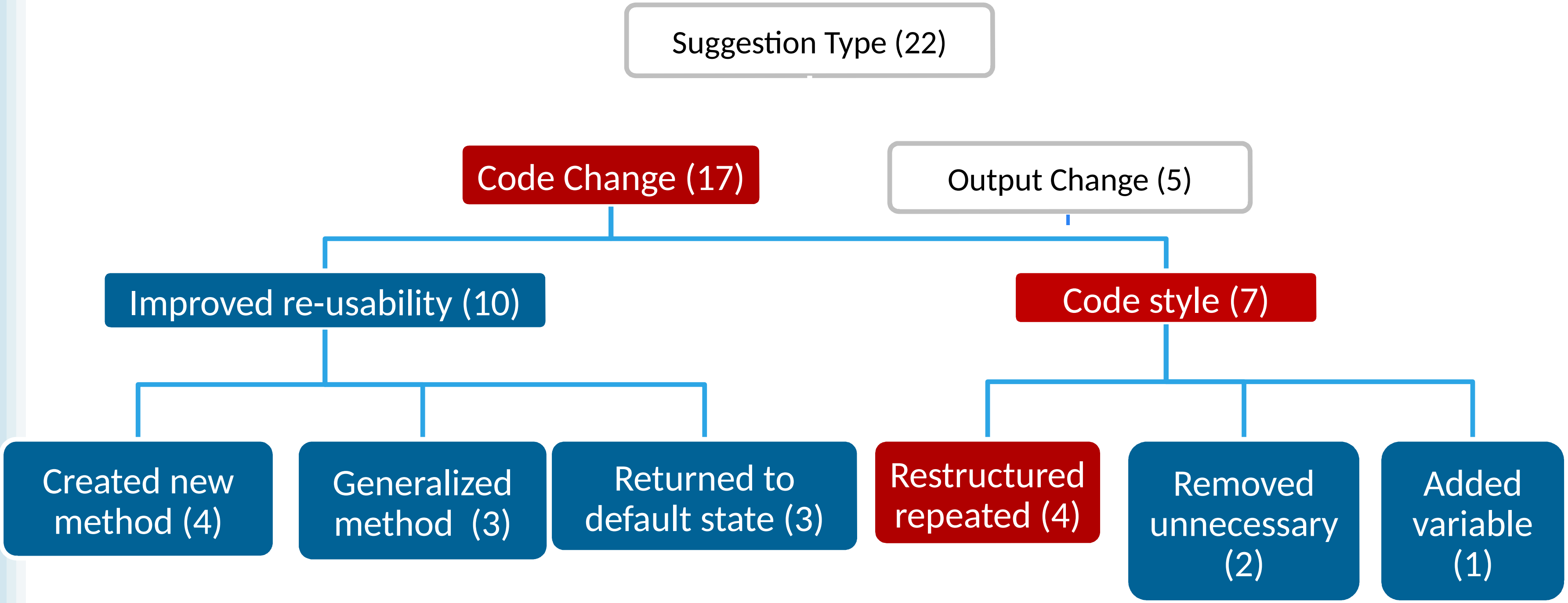
Added variable (1)

Original Program



Suggested Change





Suggestion Type (22)

Code Change (17)

Output Change (5)

Improved re-usability (10)

Code style (7)

Created new method (4)

Generalized method (3)

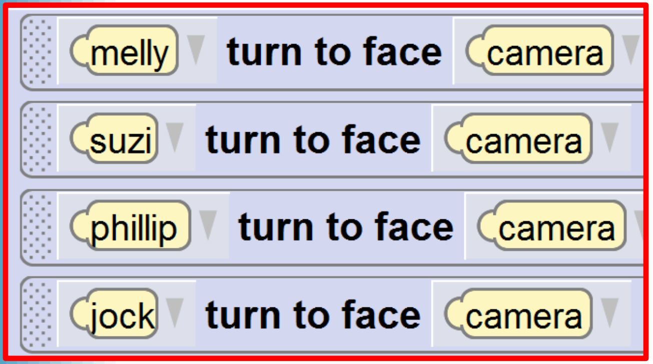
Returned to default state (3)

Restructured repeated (4)

Removed unnecessary (2)

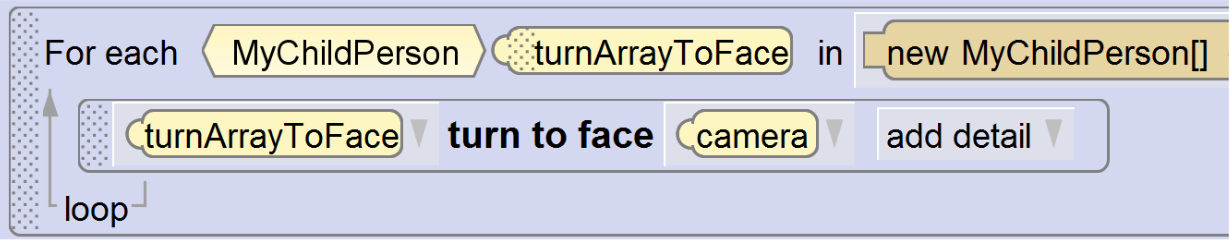
Added variable (1)

Original Program

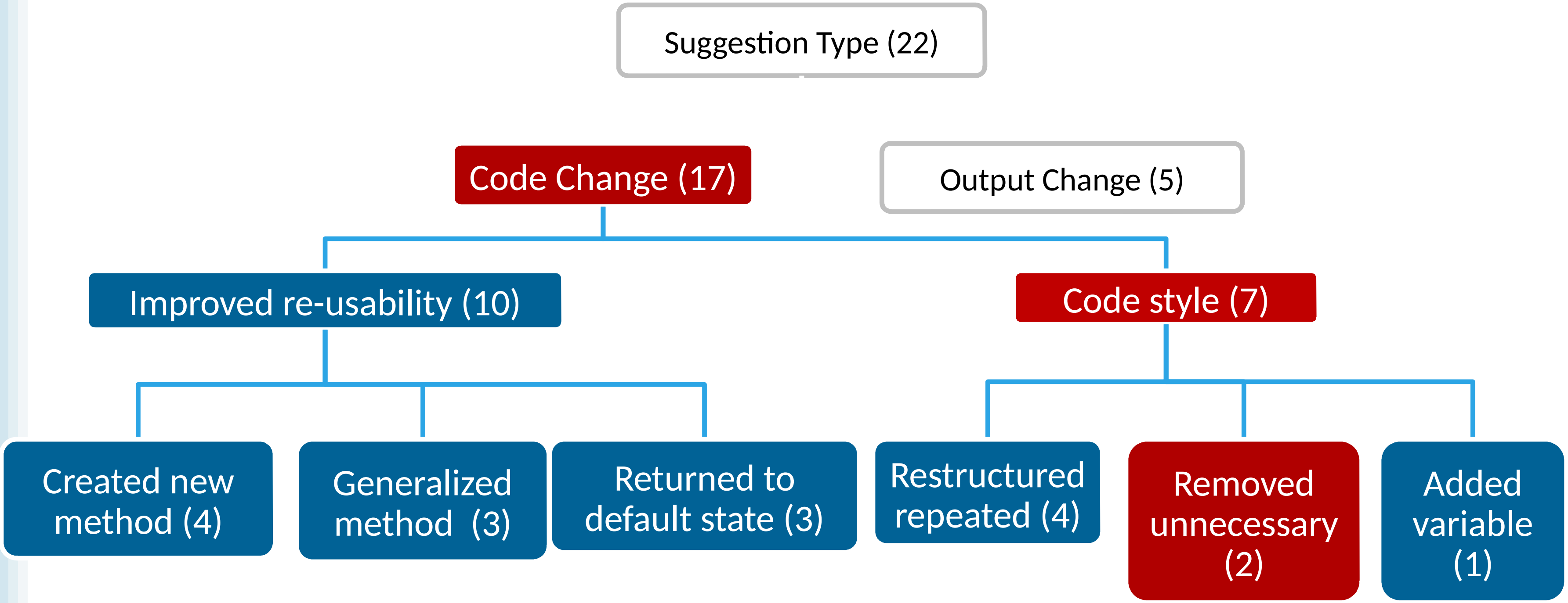


```
melly turn to face camera
suzi turn to face camera
phillip turn to face camera
jock turn to face camera
```

Suggested Change



```
For each MyChildPerson turnArrayToFace in new MyChildPerson[]
    turnArrayToFace turn to face camera add detail
loop
```



Suggestion Type (22)

Code Change (17)

Output Change (5)

Improved re-usability (10)

Code style (7)

Created new method (4)

Generalized method (3)

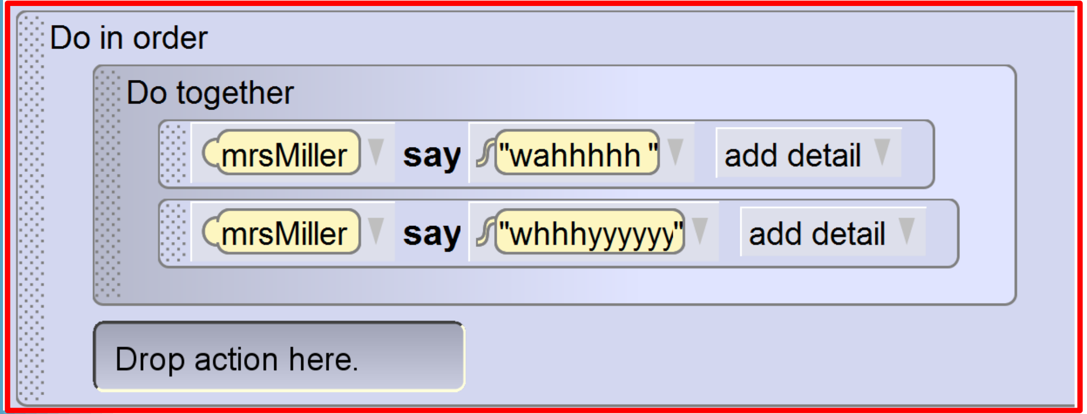
Returned to default state (3)

Restructured repeated (4)

Removed unnecessary (2)

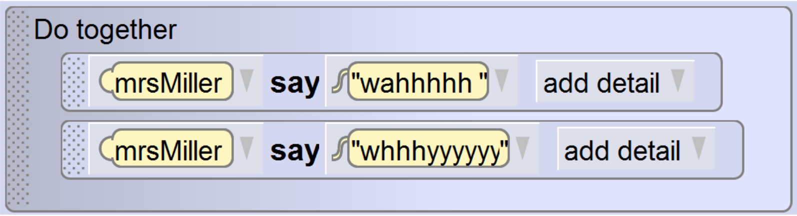
Added variable (1)

Original Program

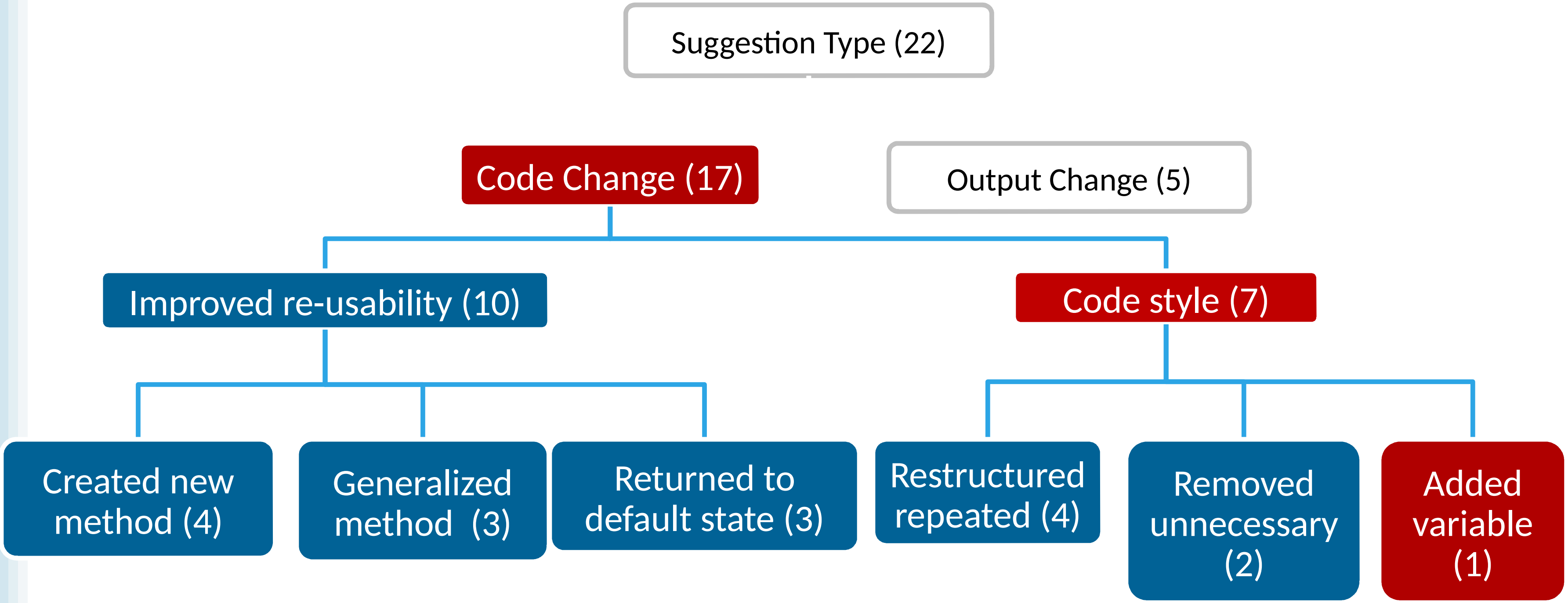


```
Do in order
    Do together
        mrsMiller say "wahhhhh " add detail
        mrsMiller say "whhhyyyyyy" add detail
    Drop action here.
```

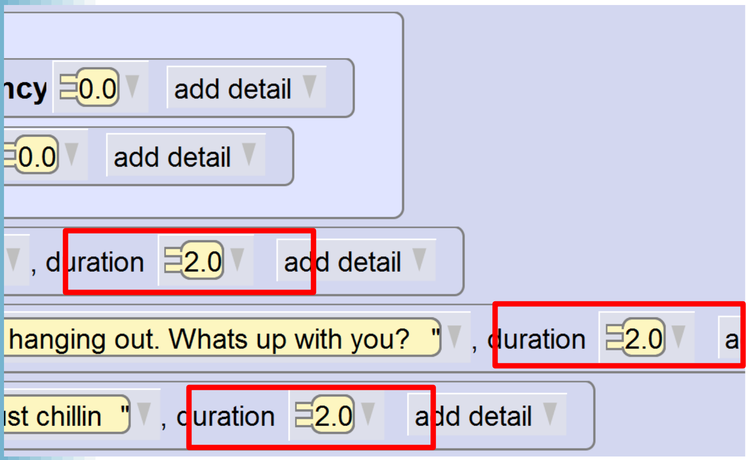
Suggested Change



```
Do together
    mrsMiller say "wahhhhh " add detail
    mrsMiller say "whhhyyyyyy" add detail
```

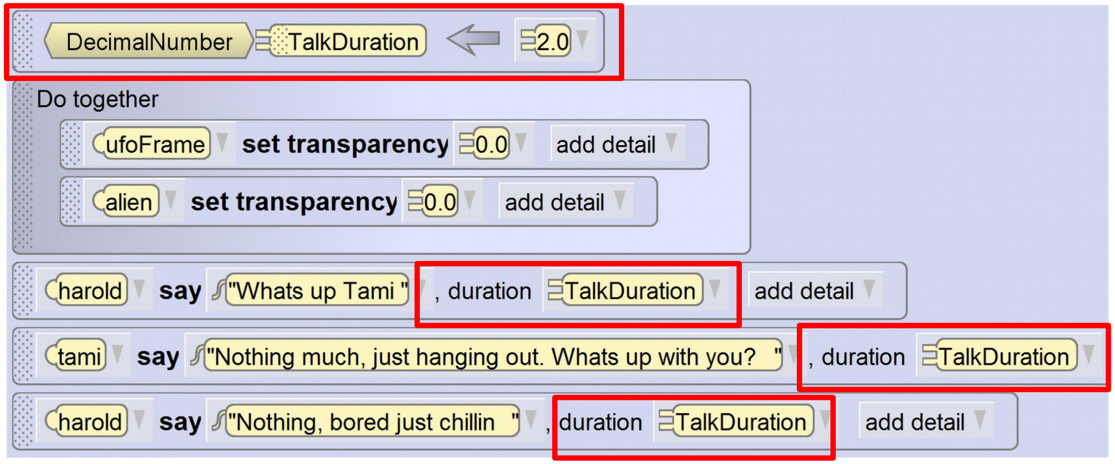


Original Program



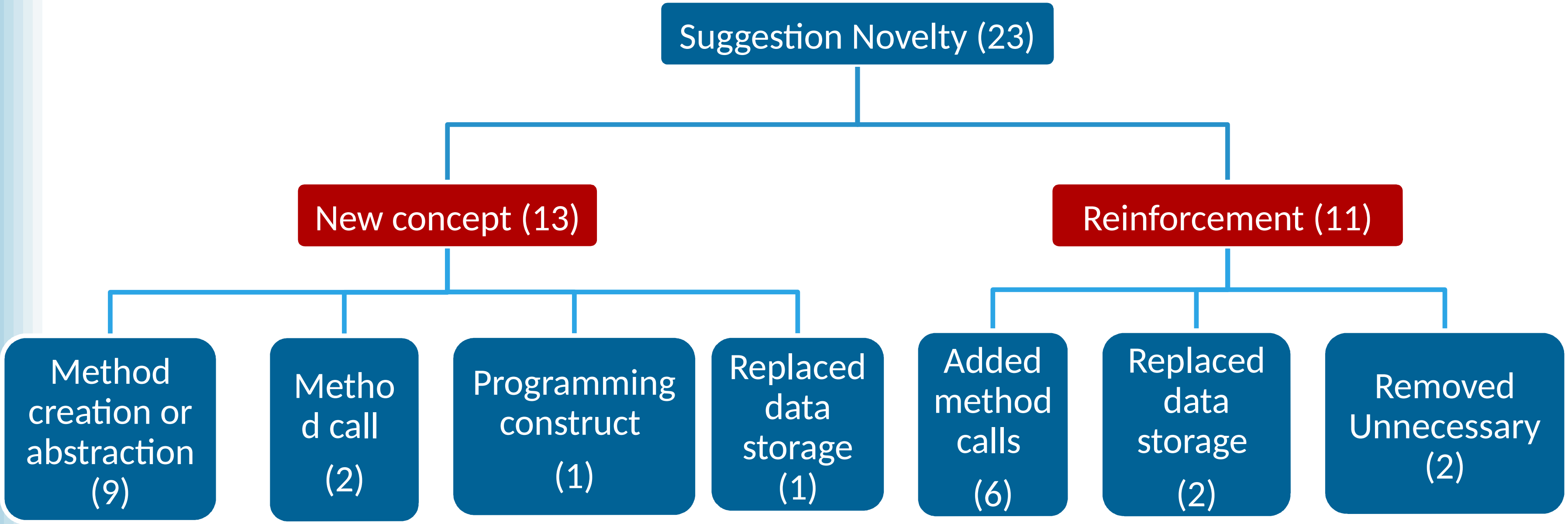
```
ncy  0.0  add detail
0.0  add detail
, duration  2.0  add detail
hanging out. Whats up with you? " , duration  2.0  a
ust chillin " , duration  2.0  add detail
```

Suggested Change



```
DecimalNumber  TalkDuration  ⇐  2.0
Do together
    ufoFrame  set transparency  0.0  add detail
    alien  set transparency  0.0  add detail
harold  say  "Whats up Tami " , duration  TalkDuration  add detail
tami  say  "Nothing much, just hanging out. Whats up with you? " , duration  TalkDuration
harold  say  "Nothing, bored just chillin " , duration  TalkDuration  add detail
```

Did the suggestions present a new programming concept?



Suggestion Novelty (23)

New concept (13)

Reinforcement (11)

Method creation or abstraction (9)

Metho d call (2)

Programming construct (1)

Replaced data storage (1)

Added method calls (6)

Replaced data storage (2)

Removed Unnecessary (2)

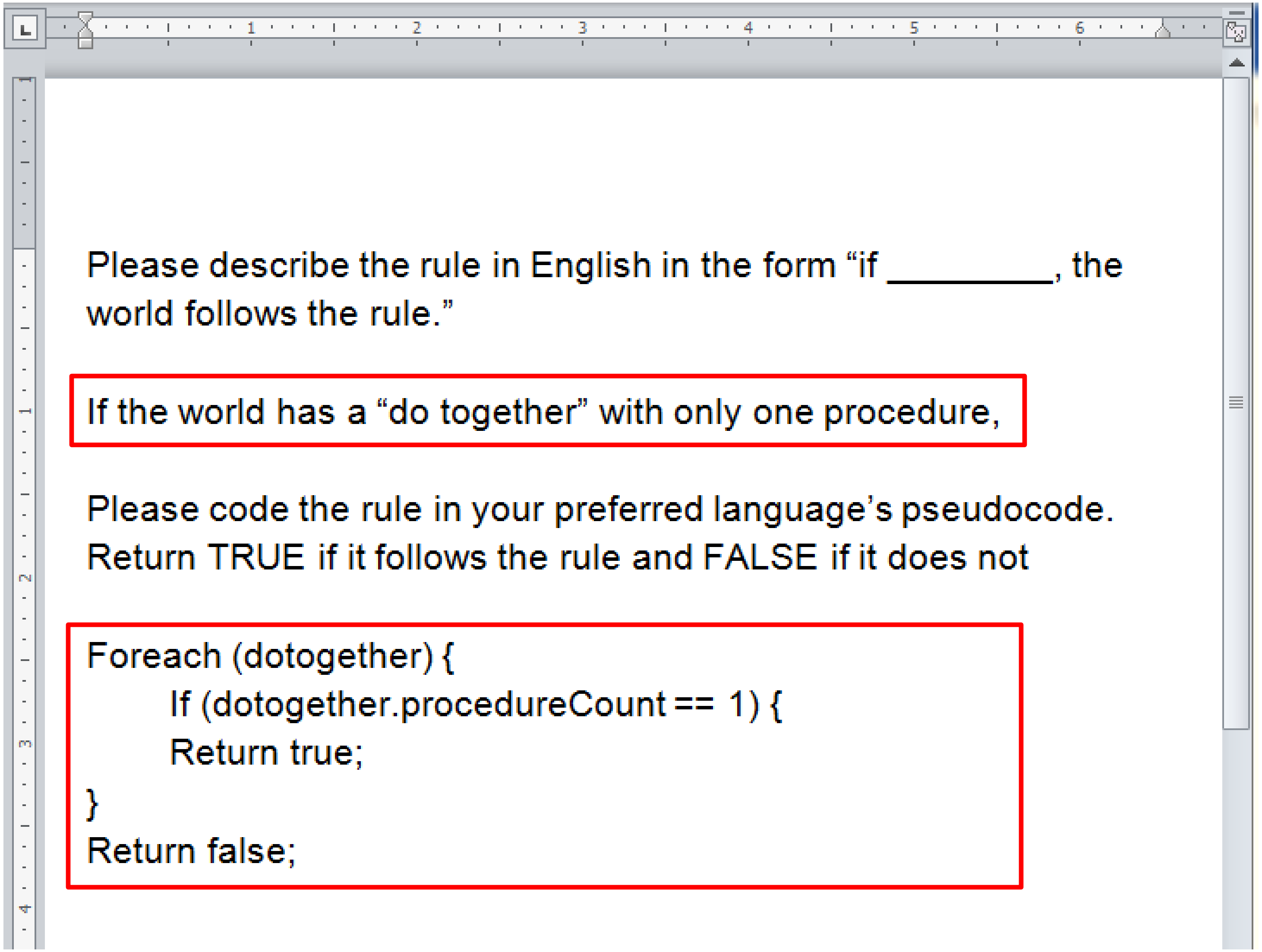
Will mentor suggestions improve novice programs?

- 17 Code Changes, 5 Output Changes
- 13 New Concepts, 11 Reinforcements

Exploratory Study Results

- Do mentor suggestions have the potential to teach novices to improve their code?
- **Can mentors write rules to generalize whether novice programs should receive certain suggestions?**
- What do mentors need in a tool to code these rules?

Rules



Please describe the rule in English in the form “if _________, the world follows the rule.”

If the world has a “do together” with only one procedure,

Please code the rule in your preferred language’s pseudocode. Return TRUE if it follows the rule and FALSE if it does not

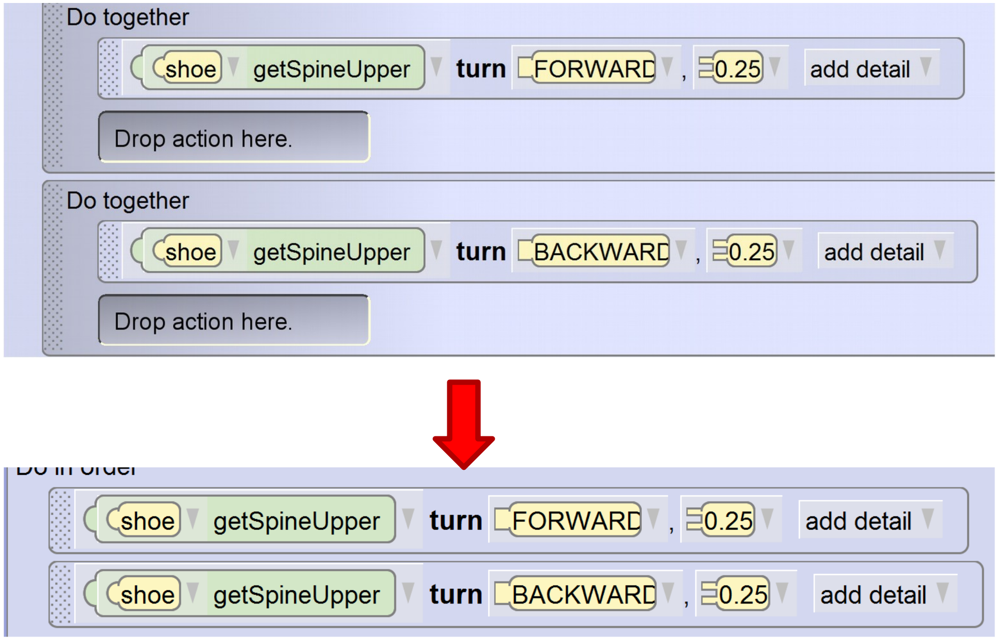
```
Foreach (dotogether) {
        If (dotogether.procedureCount == 1) {
        Return true;
}
Return false;
```

Do the rules appropriately select programs to receive the suggestions?

- Good rule
- Fixable rule
- Bad rule
- Un-fixable rule

Good Rule

Suggestion:



```
Do together
    shoe getSpineUpper turn FORWARD , 0.25 add detail
    Drop action here.
Do together
    shoe getSpineUpper turn BACKWARD , 0.25 add detail
    Drop action here.
```

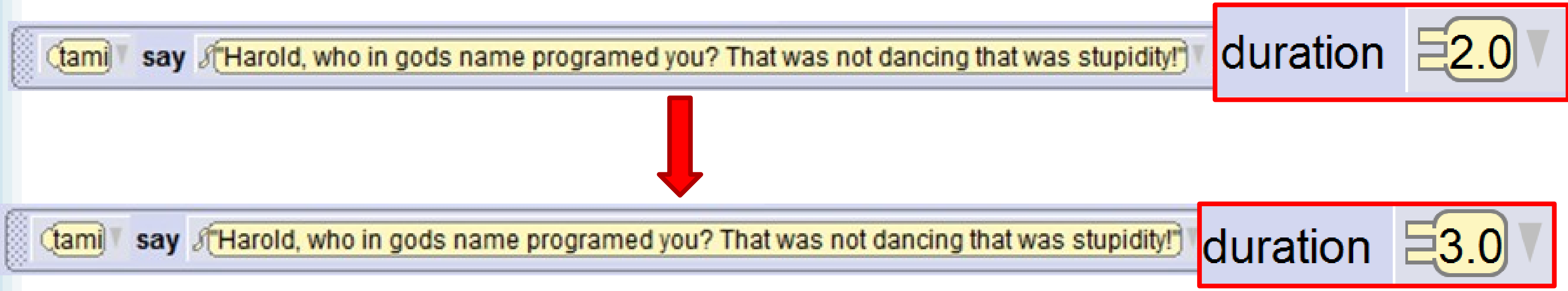
```
Do in order
    shoe getSpineUpper turn FORWARD , 0.25 add detail
    shoe getSpineUpper turn BACKWARD , 0.25 add detail
```

Rule:

```
Foreach(dotogether){
   if(dotogether.procedureCount == 1){
      return true;
}
return false;
```

Fixable Rule

Suggestion:



tami say "Harold, who in gods name programed you? That was not dancing that was stupidity!" duration 2.0

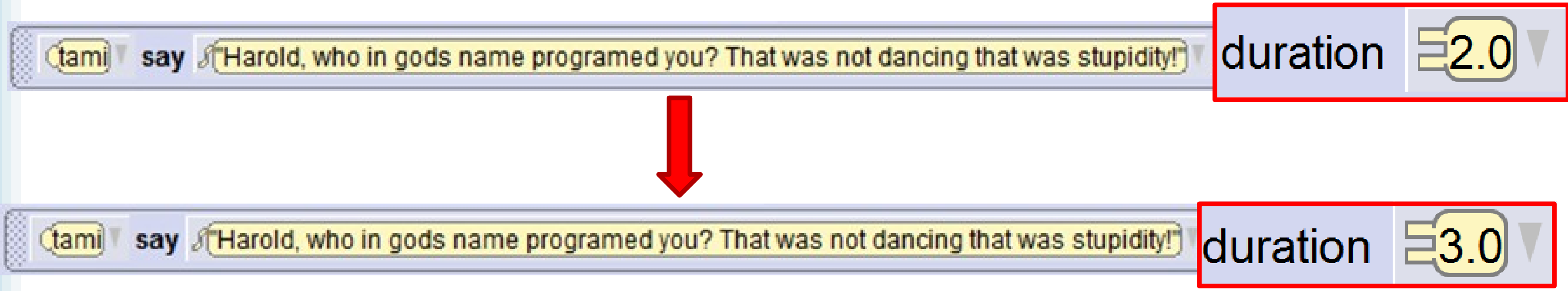
tami say "Harold, who in gods name programed you? That was not dancing that was stupidity!" duration 3.0

Rule: if the text is more than 8 words

```
While endOfCode is false
 if procedure = say
   then
      if TextString > 8
          then return true
      else continue
```

Fixable Rule

Suggestion:



tami say "Harold, who in gods name programed you? That was not dancing that was stupidity!" duration 2.0

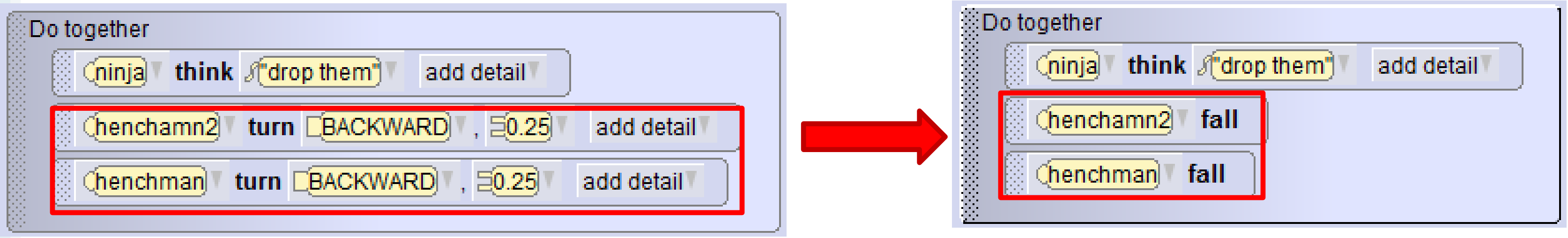
tami say "Harold, who in gods name programed you? That was not dancing that was stupidity!" duration 3.0

Rule: if the text is more than 8 words

```
While endOfCode is false
 if procedure = say
    then
       if TextString > 8 AND duration < 3
           then return true
       else continue
```

Bad Rule

Suggestion:



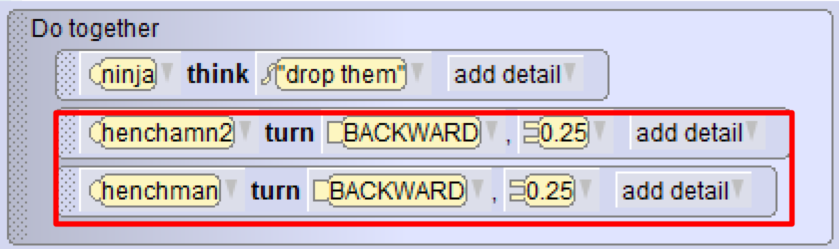
```
Do together
  ninja think "drop them" add detail
  henchamn2 turn BACKWARD , 0.25 add detail
  henchman turn BACKWARD , 0.25 add detail
```

```
Do together
  ninja think "drop them" add detail
  henchamn2 fall
  henchman fall
```

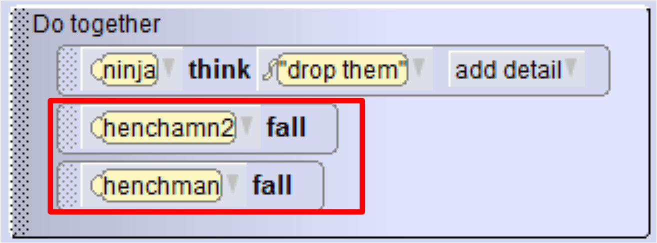
Rule idea: “if a world has the same type of object perform the same action multiple times”

Bad Rule

Suggestion:

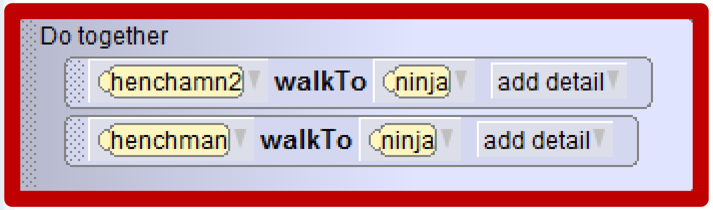


```
Do together
    ninja think "drop them" add detail
    henchamn2 turn BACKWARD , 0.25 add detail
    henchman turn BACKWARD , 0.25 add detail
```

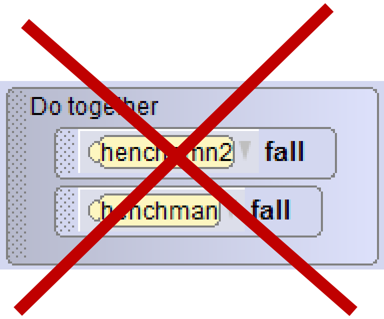


```
Do together
    ninja think "drop them" add detail
    henchamn2 fall
    henchman fall
```

Rule idea: “if a world has the same type of object perform the same action multiple times”



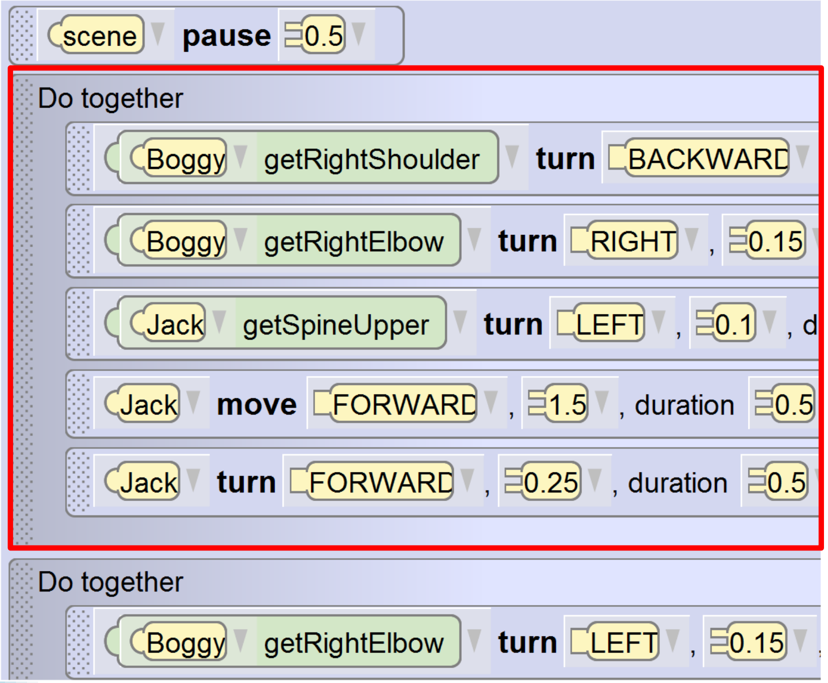
```
Do together
    henchamn2 walkTo ninja add detail
    henchman walkTo ninja add detail
```



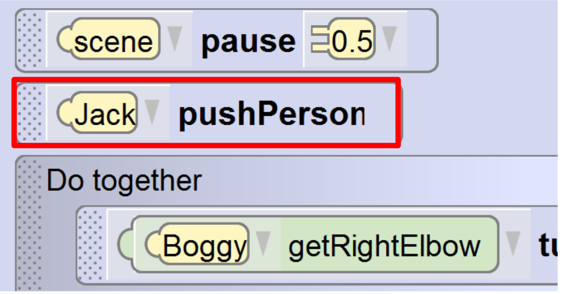
```
Do together
    henchamn2 fall
    henchman fall
```

Un-Fixable Rule

Suggestion:



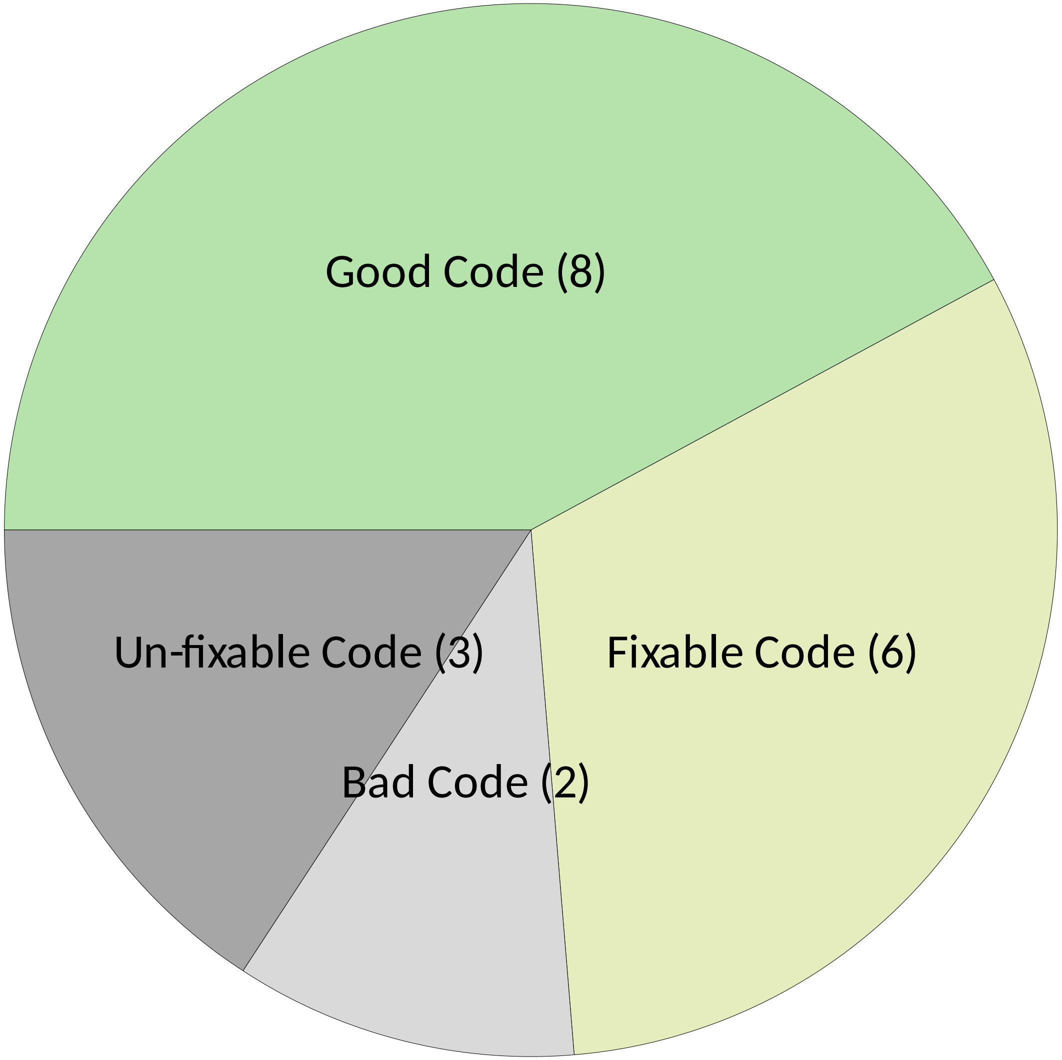
```
scene pause 0.5
Do together
    Boggy getRightShoulder turn BACKWARD
    Boggy getRightElbow turn RIGHT , 0.15
    Jack getSpineUpper turn LEFT , 0.1 , d
    Jack move FORWARD , 1.5 , duration 0.5
    Jack turn FORWARD , 0.25 , duration 0.5
Do together
    Boggy getRightElbow turn LEFT , 0.15 ,
```



```
scene pause 0.5
Jack pushPerson
Do together
    Boggy getRightElbow tu
```

Rule idea: "if do together statement contains several items"

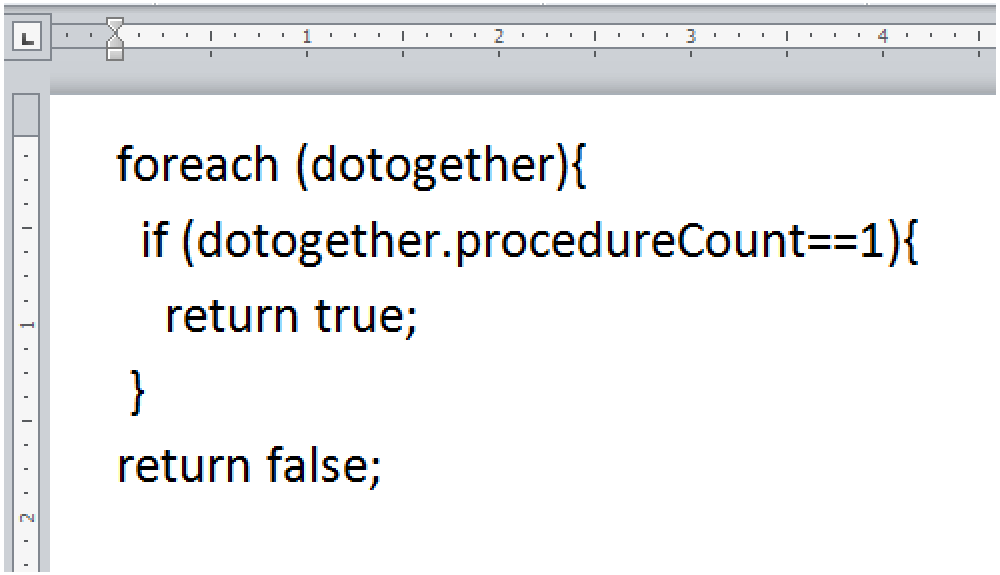
Good + Fixable Rules: 14/19



Exploratory Study Results

- Do mentor suggestions have the potential to teach novices to improve their code?
- Can mentors write rules to generalize whether novice programs should receive certain suggestions?
- **What do mentors need in a tool to code these rules?**

Rule Pseudocode



```
foreach (dotogether){
 if (dotogether.procedureCount==1){
  return true;
 }
return false;
```

J. F. Pane, B. A. Myers, and C. A. Ratanamahatana, “Studying the language and structure in non-programmers’ solutions to programming problems,” *Int. J. Hum.-Comput. Stud.*, vol. 54, no. 2, pp. 237–264, Feb. 2001.

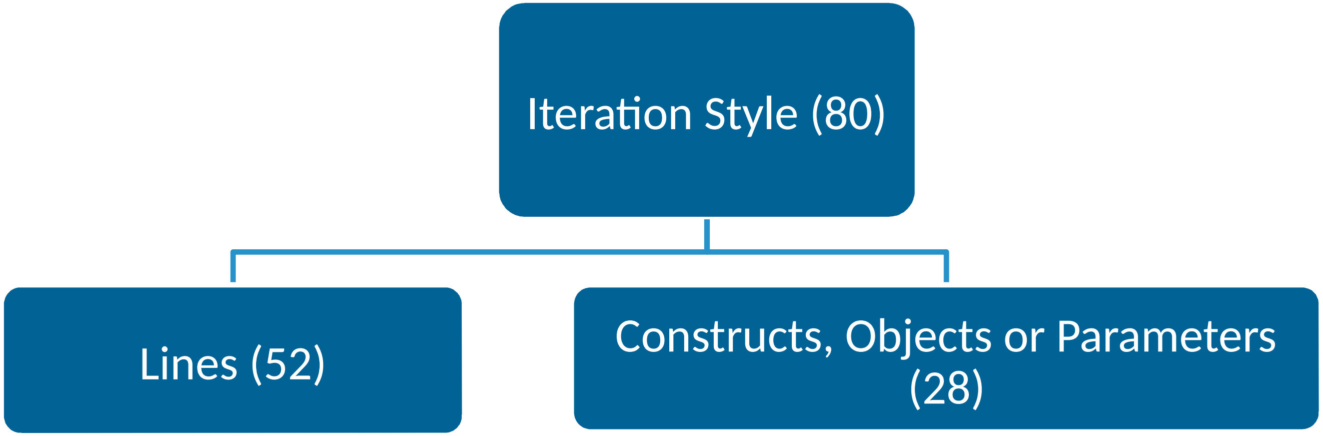
80 Iteration

```
for(Statement s: statements)
```

123 Comparison

```
if (statement.getName() == “say”)
```

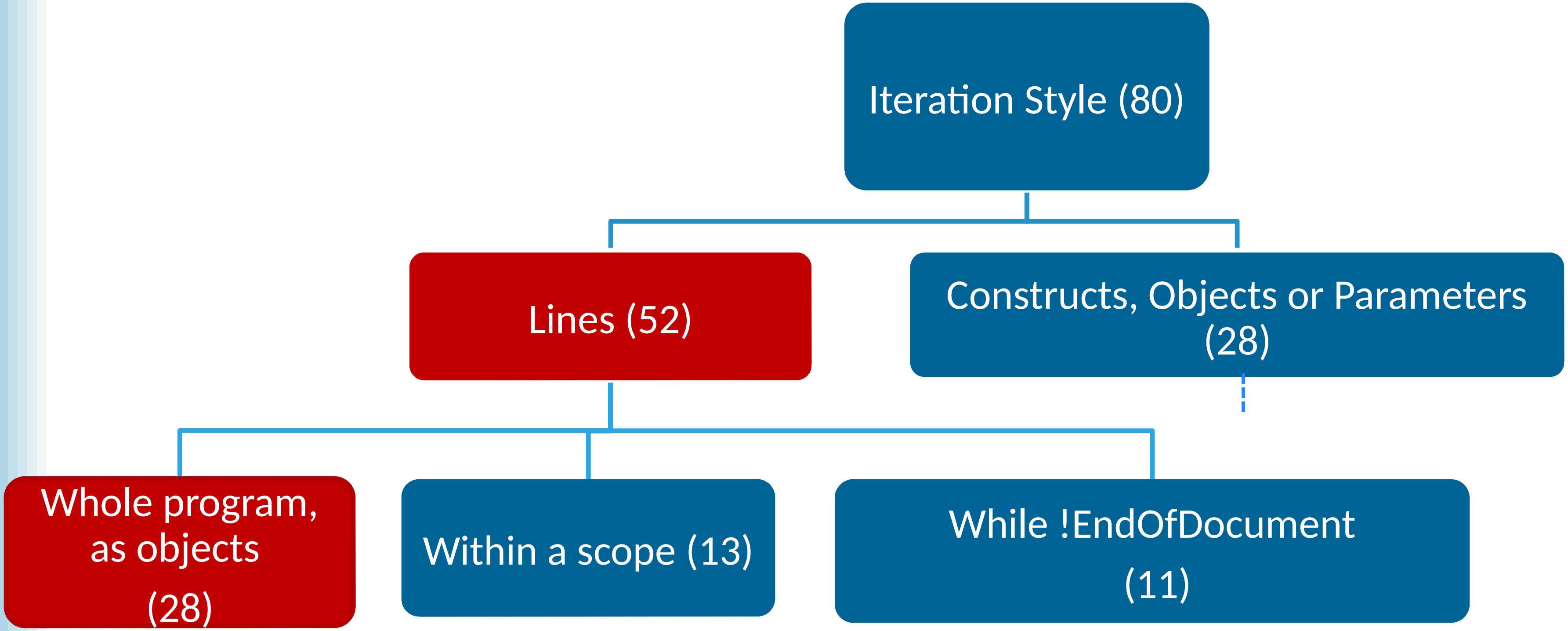
84 Miscellaneous



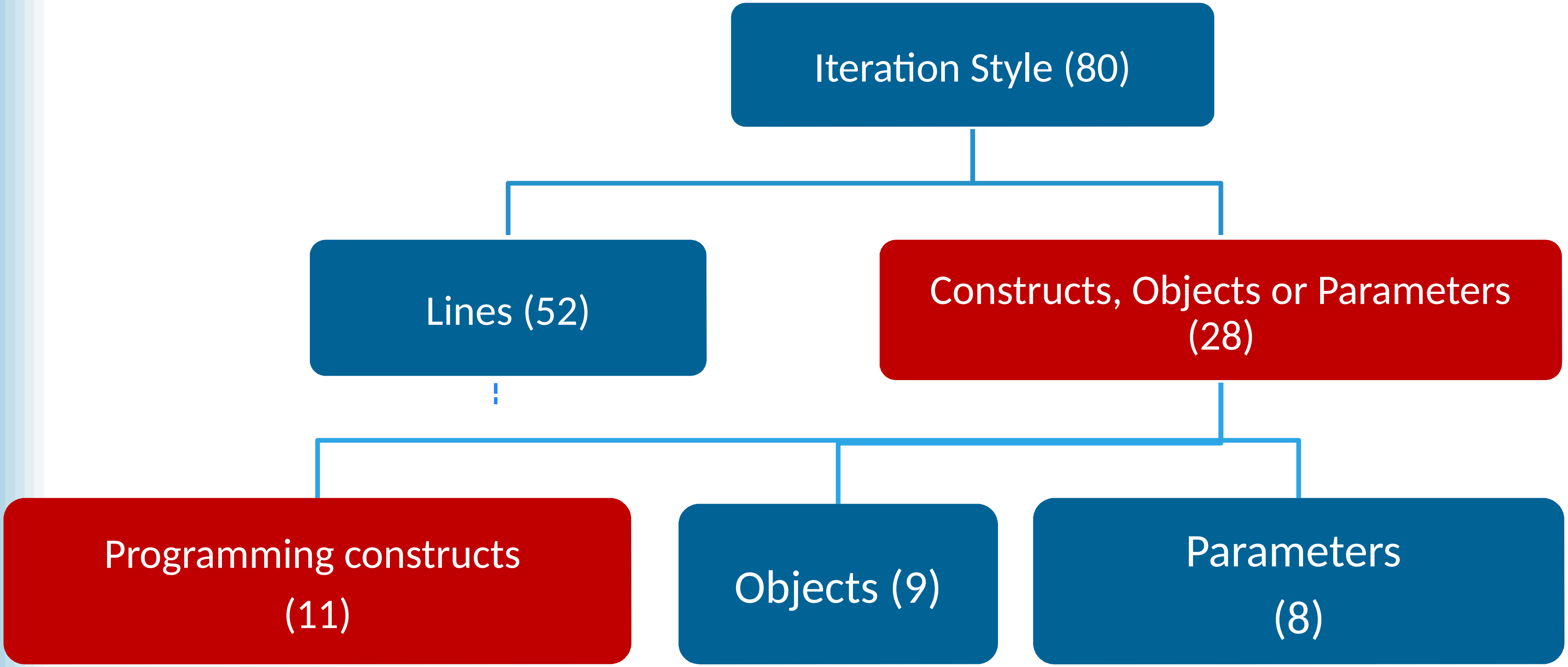
Iteration Style (80)

Lines (52)

Constructs, Objects or Parameters (28)


Iteration Style (80)
Lines (52)
Constructs, Objects or Parameters (28)
Whole program, as objects (28)
Within a scope (13)
While !EndOfDocument (11)

```
for each procedure in myProgram
for(Statement statement: getStatements())
foreach (allMethods as k=> method)
for (statement in statements)
```



Iteration Style (80)

Lines (52)

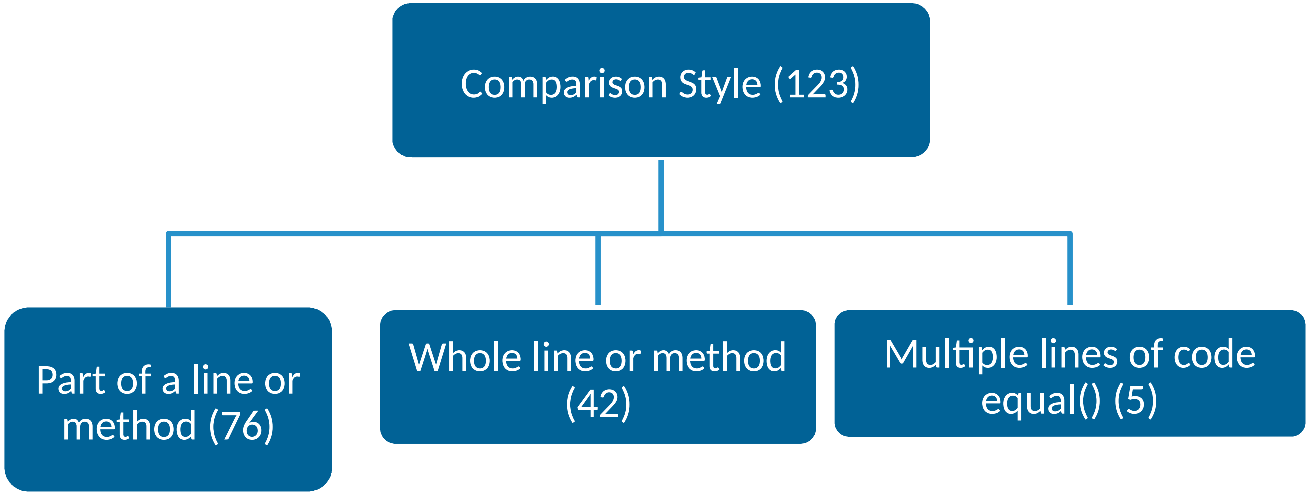
Constructs, Objects or Parameters (28)

Programming constructs (11)

Objects (9)

Parameters (8)

```
foreach (dotogether)
foreach sequence in world.Dotogethers()
Blocks blocks= findAllDoTogetherBlocks(code)
For(Block block : blocks)
```

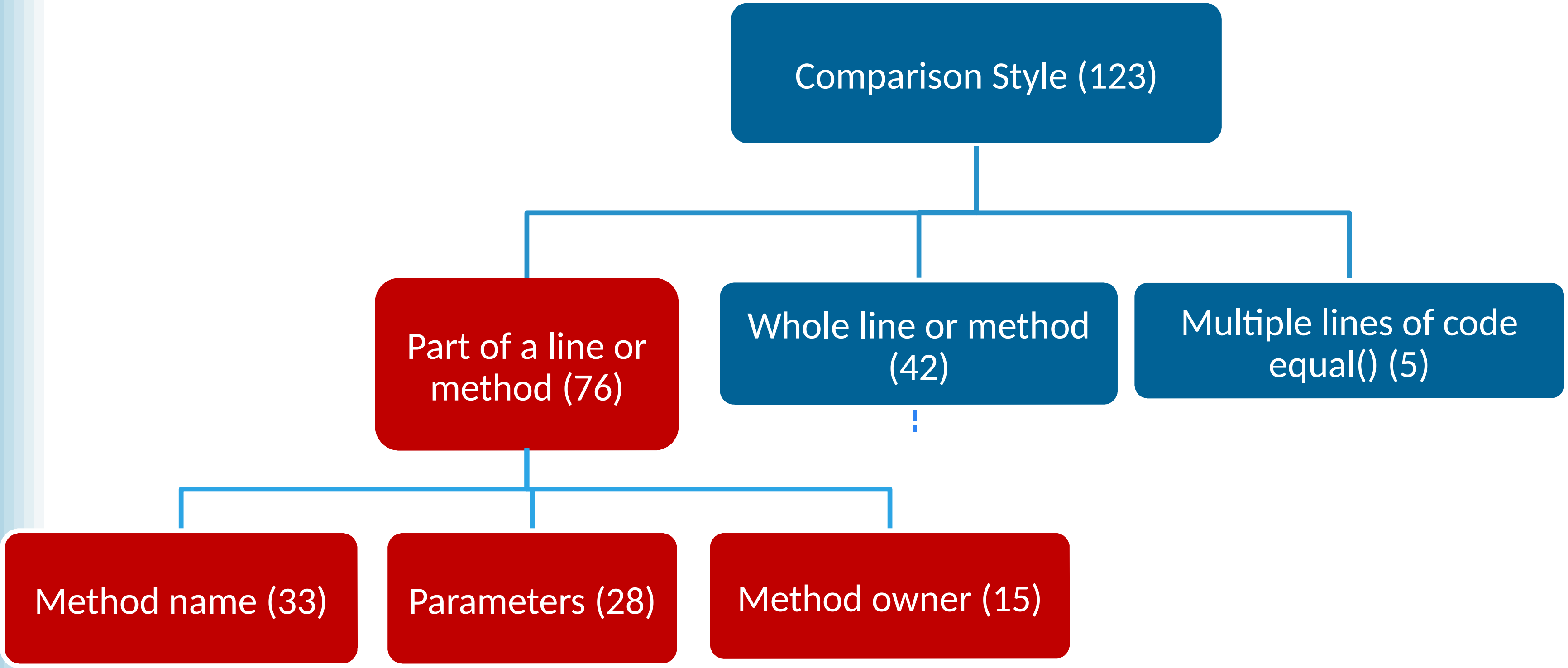


Comparison Style (123)

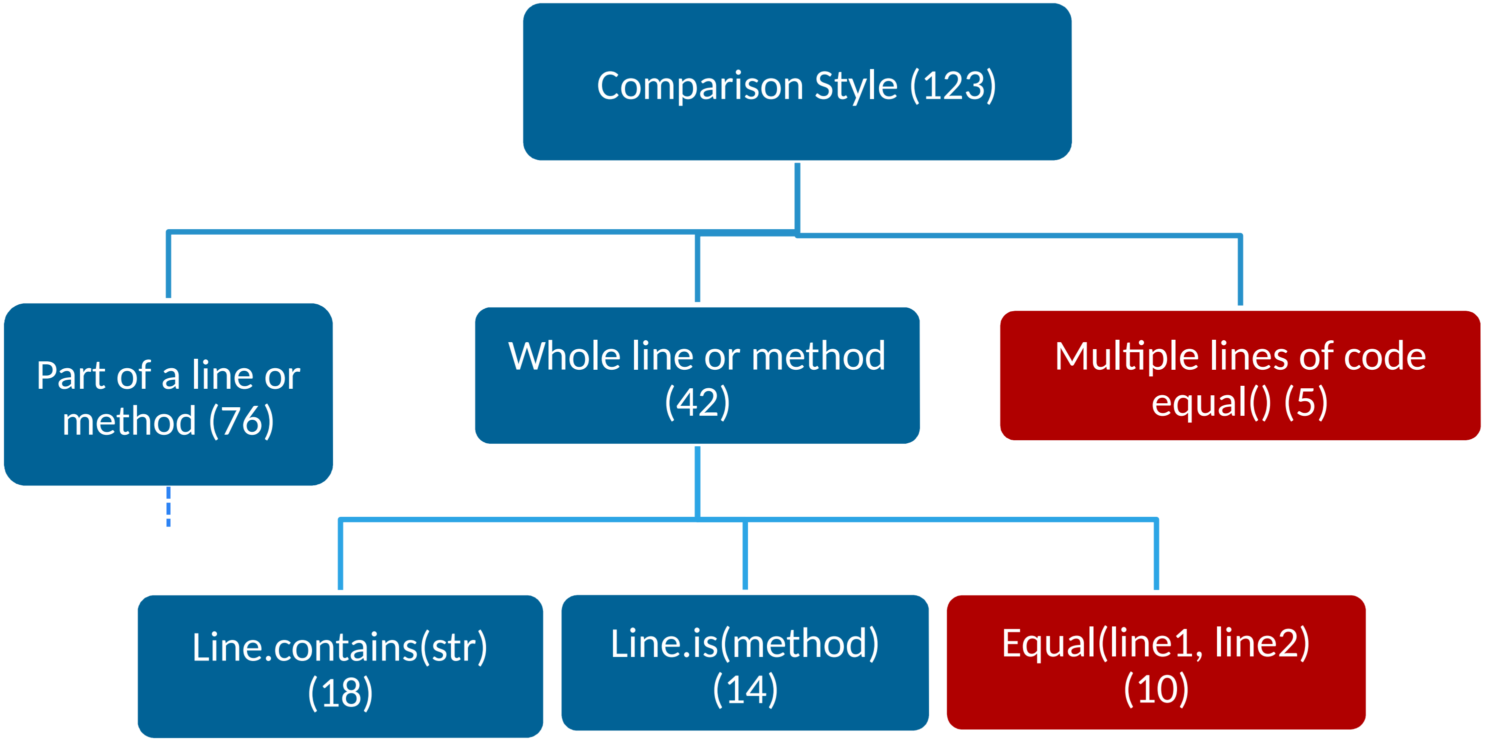
Part of a line or method (76)

Whole line or method (42)

Multiple lines of code equal() (5)


Comparison Style (123)
Part of a line or method (76)
Whole line or method (42)
Multiple lines of code equal() (5)
Method name (33)
Parameters (28)
Method owner (15)

```
if(method -> name == allMethods[k+1]->name)
if(turn direction == BACKWARDS)
if(statement.getObject() == TYPE.BIPED)
```



Comparison Style (123)

Part of a line or method (76)

Whole line or method (42)

Multiple lines of code equal() (5)

Line.contains(str) (18)

Line.is(method) (14)

Equal(line1, line2) (10)

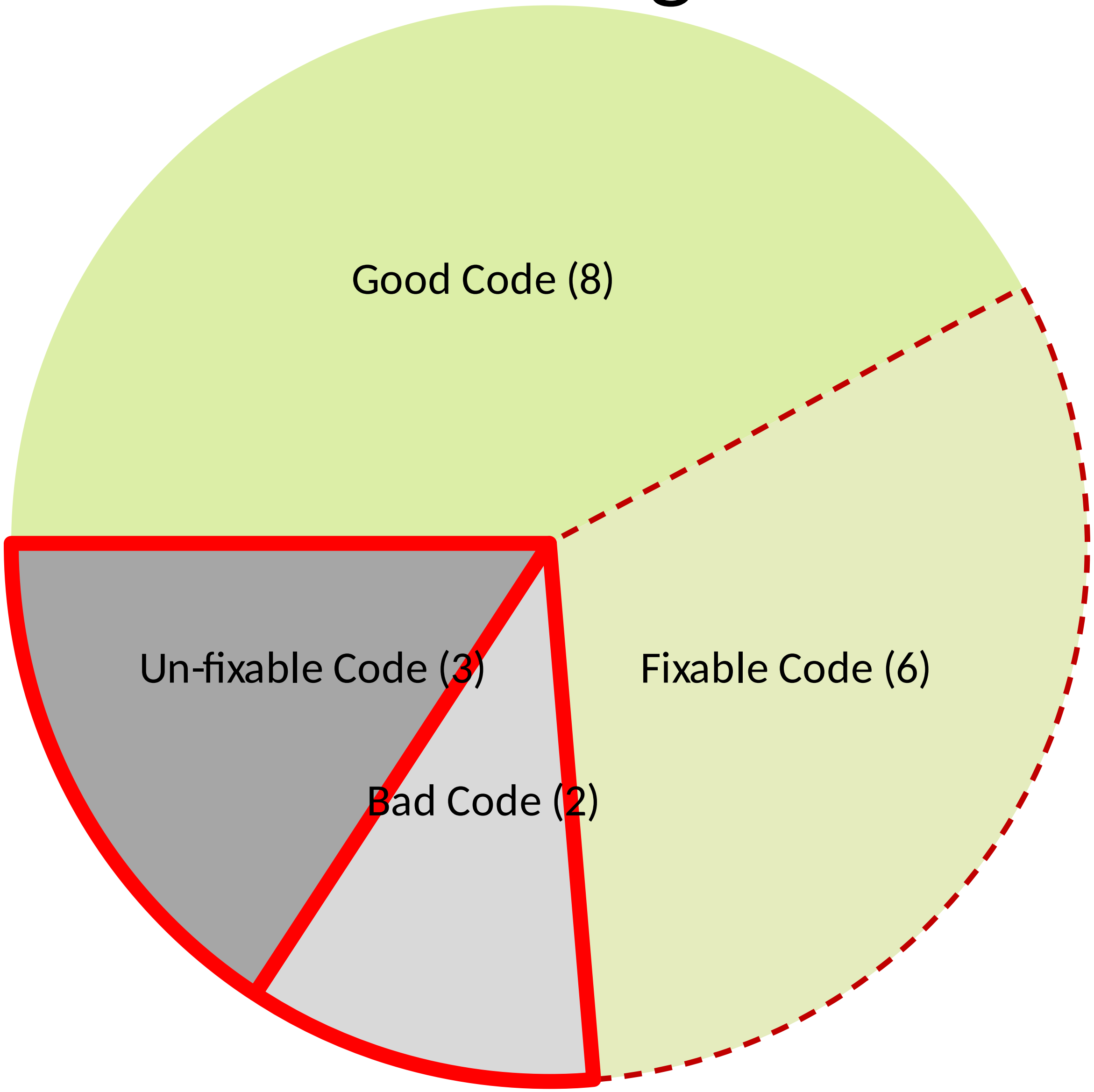
Implications for a Rule-Authoring Tool

- **Iteration**
  - Line by line
  - Through constructs
  - Through parts of methods
- **Comparison**
  - Method name
  - Method object
  - Method parameters
  - Whole lines
  - Multiple lines

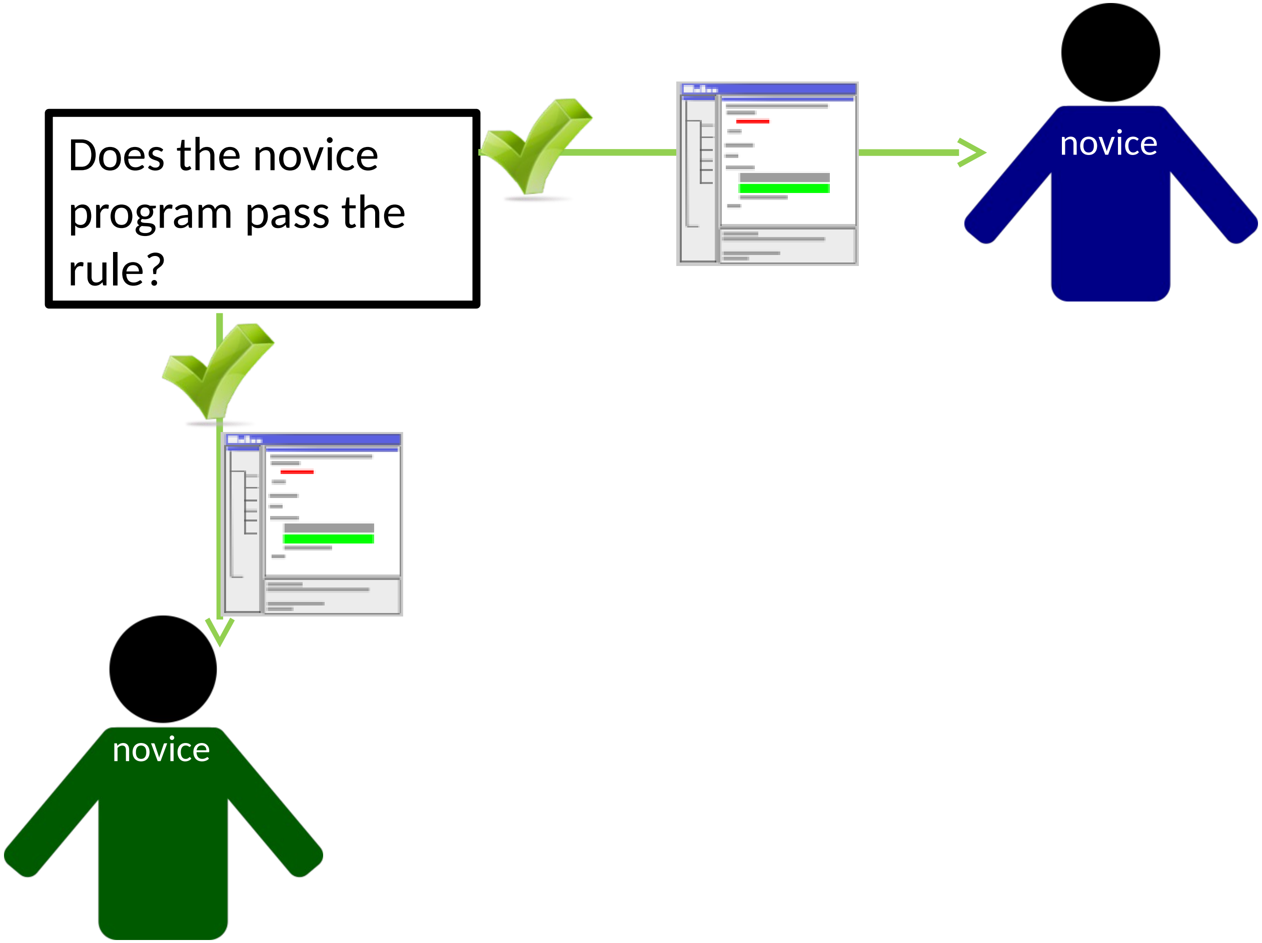
Conclusions

Experienced programmers’ suggestions and rules have the potential to mentor novice programmers at a large scale.

Future Work: Filtering and Editing Rules



Future Work: Presenting Suggestions



Does the novice program pass the rule?

MICHELLE ICHINCO
Washington University in St. Louis
michelle.ichinco@wustl.edu

NSF

Academy of Science

St. Louis | SINCE 1856