

Anonymous Visitor

And some Code Generation Black Magic

Mail: pierre.Misse-Chanabier@inria.fr

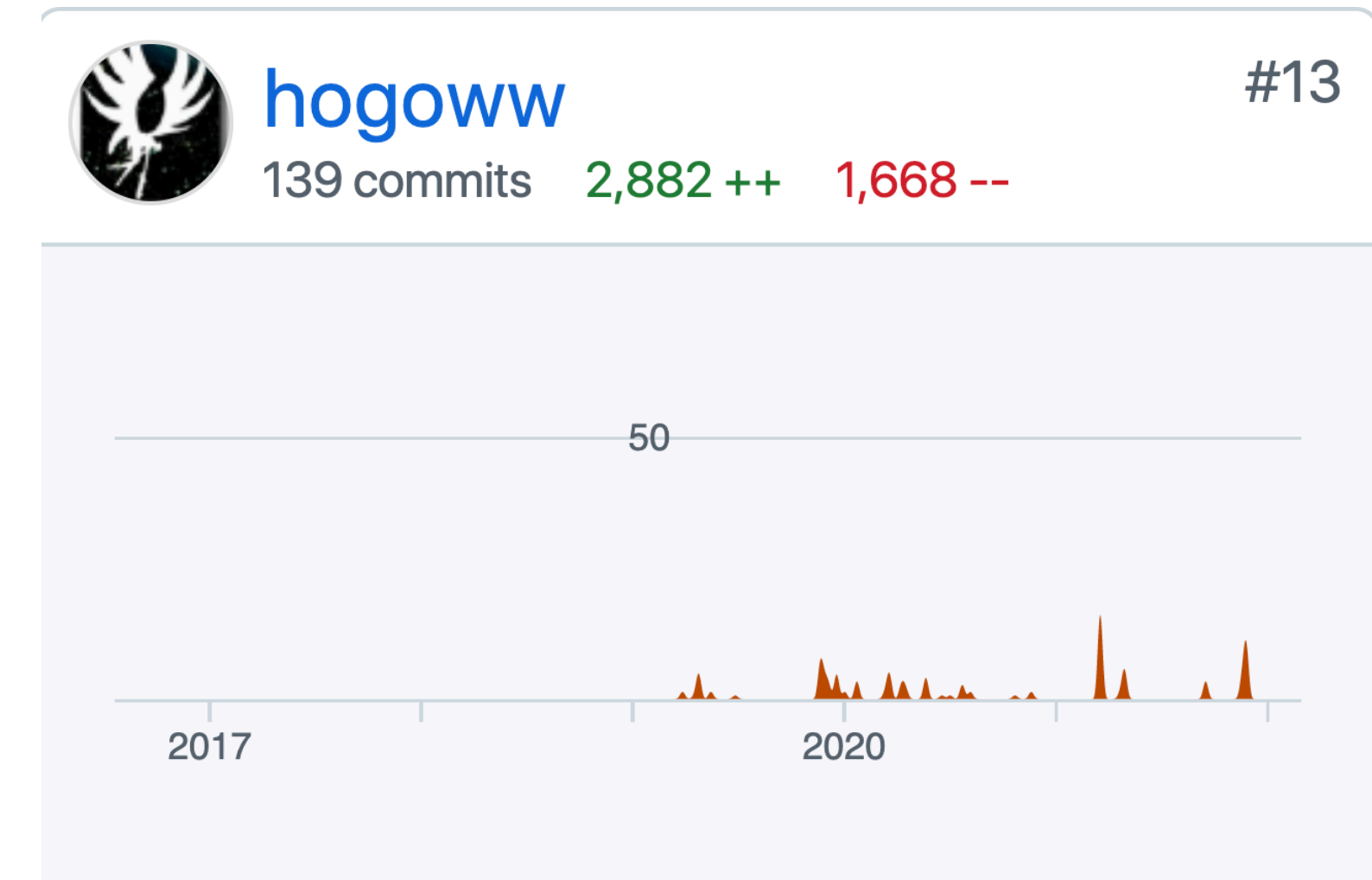
GitHub: <https://github.com/hogoww>

Discord: @hogo#8547



Bit about me

- PHD student at RMOD in VM generation and Testing
- Multi Pharo project contributor and Pharo hacker
- NewTools, Iceberg, Pharo-Launcher, PharoVM ...
Pharo: Code Completion, RBAST, CalypsoBrowser...
- Finally simplified the drag and drop for Calypso in P10 !





Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



I work on languages

And Abstract Syntax Trees

visitBlockNode: aBlockNode

```
self visitArgumentNodes: aBlockNode arguments.  [:anArg |  
self visitNode: aBlockNode body  anArg aMessage: 1 + 2 ]
```

AST visitors are everywhere

(In your Pharo image)

- Compiler
- Code Styler
- ParseTreeSearcher
- Code Formator
- Approximately 45 visitor subclass in the base image

```
CoTypeInferencer
EFormatter
FBDOptimizedMessagesRewriter
IconStyler
  BreakpointIconStyler
  CounterIconStyler
  CoverageRubricStyler
  DisabledBreakpointIconStyler
  DoOnlyOnceIconStyler
  DocCommentIconStyler
  ErrorNodeStyler !
  FlagIconStyler
  HaltIconStyler
    HaltOnCountIconStyler
    HaltOnceIconStyler
  MetaLinkIconStyler
  MetalinkIconStyler
  SemanticMessageIconStyler !
  SemanticVariableIconStyler !
  SemanticWarningIconStyler
  WatchIconStyler
OCASTClosureAnalyzer
  RFASTClosureAnalyzer
OCASTMethodMetadataAnalyser
OCASTSemanticAnalyzer
  RFSemanticAnalyzer
  RFASTTranslatorForEffect
  RFASTTranslatorForValue
OCScopesCollector
PlainCodeVisitor
RABstractBlockVisitor
  RBCommentNodeVisitor
  RBGenericNodeVisitor
  RBParseErrorNodeVisitor
  RBAnonymousVisitor !
  RBCombinatorVisitor
  RBDumpVisitor
  RBIDCounterVisitor
  RBMatchVisitor
  RBParseTreeSearcher
    RBParseTreeRewriter
  RBReadBeforeWrittenTester
  RBSimpleFormatter
  RGTraitCompositionVisitor
  TypingVisitor
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Reification of every visitor

Example of OCScopesCollector

- 2 Test Class as user
- 4 **very** simple methods
- Do we really need to reify it ?

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Anonymous Visitor

Reusing visitor pattern

```
visitor := RBAnonymousVisitor new.  
counter := 0.  
visitor registerBlock: [ :aNode | counter := counter + 1 ]  
  for: #messageNode.  
(Object >> #->) ast acceptVisitor: visitor
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Anonymous Visitor

AddHoc Reification

```
visitor := RBAnonymousVisitor new.  
counter := 0.  
visitor registerBlock: [ :aNode | counter := counter + 1 ]  
  for: #messageNode.  
(Object >> #->) ast acceptVisitor: visitor  
  
visitor materializeAs: #MessageNodeCounterVisitor.
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Common Code Generation

Basic

```
PharoDays compile: 'initialize  
collection := OrderedCollection new.  
'Initializing done !' crTrace'
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Common Code Generation

Format !

```
instVarName := #collection.  
PharoDays compile: ('initialize  
{1} := OrderedCollection new  
'Initializing done !' crTrace'  
format: { instVarName })
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Common Code Generation

Format , but better !

```
instVarName := #collection.  
PharoDays compile: ('initialize  
  {instVar} := OrderedCollection new  
  ''Initializing done !'' crTrace'  
  format: {#instVar->instVarName}asDictionary)
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



Common Code Generation

My personal favorite

```
instVarName := #collection.  
methodBody := String streamContents:[:stream]  
  stream << 'initialize' ; cr.  
  stream << tempName << ':='  
    << 'OrderedCollection new' ; cr.  
  stream << '''Initializing done !'' crTrace'].  
PharoDays compile: methodBody.
```

Mail: pierre.Misse-Chanabier@inria.fr

GitHub: <https://github.com/hogoww>

Discord: @hogo#8547



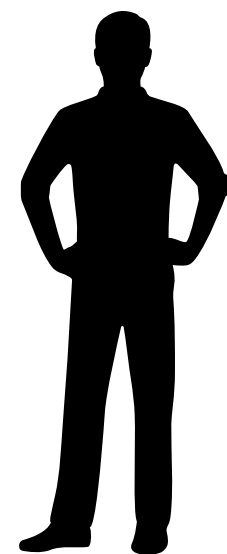
Code Generation

What I wanted

```
aCollectionInstanceVariableName := #collection.  
sourceAST := 'initialize' asMethodWithBody: [  
    aCollectionInstanceVariableName := OrderedCollection new.  
    'Initializing done !' crTrace  
].
```

```
PharoDays compile: sourceAST printString
```

It's kinda cool,
but it's black magic



Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



But I love black magic

Variable expansion in messages

```
initializerMethods := #(initializeServer initializeClient initializeUsers).  
initializationBlock := [ self initializerMethods ].  
modifiedBlock := #initializeEverything asMethodWithBody: initializationBlock.
```

initializeEverything

```
self initializeServer.  
self initializeClient.  
self initializeUsers
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547



But I love black magic

Variable expansion in static array

```
register := register asArray sort.  
registerCode := 'registerBlock:for:' asMethodWithBody:  
  [ :aBlock :aKindOfNode | (#(register) includes: aKindOfNode) ].
```

```
(#( 'argumentNode' 'argumentNodes' 'argumentVariableNode'  
  'arrayNode' 'assignmentNode' 'blockNode' 'cascadeNode'  
  'classVariableNode' 'englobingErrorNode' 'globalNode'  
  'globalVariableNode' 'instanceVariableNode'  
  'literalArrayNode' 'literalNode' 'literalValueNode'  
  'literalVariableNode' 'localVariableNode' 'messageNode'  
  'methodNode' 'node' 'parseErrorNode' 'patternBlockNode'  
  'patternWrapperBlockNode' 'pragmaNode' 'returnNode' 'selectorNode'  
  'selfNode' 'sequenceNode' 'slotInitializationNode'  
  'storeIntoTempNode' 'storePopIntoTempNode' 'superNode'  
  'temporaryDeclarationNode' 'temporaryNode' 'temporaryNodes'  
  'temporaryVariableNode' 'thisContextNode' 'unreachableStatement'  
  'variableNode' ) includes: aKindOfNode) ifFalse: [
```

Mail: pierre.Misse-Chanabier@inria.fr
GitHub: <https://github.com/hogoww>
Discord: @hogo#8547

