Item 270
git_comments:

1. * * Evaluates {@link Traversal} {@link Bytecode} with the specified {@code Bindings}. These {@code Bindings} * supplied to this method will be merged with global engine bindings and override them where keys match. This * method assumes that the traversal source to execute against is named "g". * * @deprecated As of release 3.2.7, replaced by {@link #eval(Bytecode, Bindings, String)}.
2. * * Evaluates {@link Traversal} {@link Bytecode} against a traversal source in the global bindings of the * {@code ScriptEngine}. * * @param bytecode of the traversal to execute * @param traversalSource to execute the bytecode against which should be in the available bindings.
3. * * Evaluates bytecode with bindings for a specific language into a {@link Traversal}. * * @param bytecode to execute as a traversal * @param boundVars local bindings * @param language the scripting language to use to process the bytecode * @param traversalSource the specific traversal source to execute the bytecode against
4. * * @deprecated As of release 3.2.7, replaced by {@link #eval(Bytecode, Bindings, String, String)}.
5. these validations occur before merging in bytecode bindings which will override existing ones. need to extract the named traversalsource prior to that happening so that bytecode bindings can share the same namespace as global bindings (e.g. traversalsources and graphs).
6. these validations occur before merging in bytecode bindings which will override existing ones. need to extract the named traversalsource prior to that happening so that bytecode bindings can share the same namespace as global bindings (e.g. traversalsources and graphs).
7. purposefully use "x" to match the name of the traversal source binding for "x" below and thus tests the alias added for "x"

git_commits:

1. **summary:** TINKERPOP-1792 Fixed GremlinScriptEngine bug in lambda processing
   message: TINKERPOP-1792 Fixed GremlinScriptEngine bug in lambda processing GremlinScriptEngine instances should now properly select the appropriate TraversalSource from bindings rather than select a random one. The TraversalSource is locally aliased to a "hidden" variable at the time of evaluation to avoid naming clashes with local variables defined in the bytecode itself. Added a number of validations to be sure that users get appropriate errors if they try to use the eval() method the wrong way. Included new tests to ensure that GremlinScriptEngine have the appropriate behavior.
   label: code-design

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Random TraversalSource Selection in GremlinScriptEngine
   description: {{GremlinScriptEngine}} implementations make a random {{TraversalSource}} selection when processing lambdas. Obviously it should be bound more clearly to the specific {{TraversalSource}} the caller requests. Another issue that isn't completely clear is that bindings passed from the client share the same namespace as {{TraversalSource}} bindings which means that if the {{TraversalSource}} is "g" then you couldn't write a traversal like: {{withSideEffect(“g”, “hello”)}}. That could be smarter as well.
   label: code-design
2. **summary:** Random TraversalSource Selection in GremlinScriptEngine
   description: {{GremlinScriptEngine}} implementations make a random {{TraversalSource}} selection when processing lambdas. Obviously it should be bound more clearly to the specific {{TraversalSource}} the caller requests. Another issue that isn't completely clear is that bindings passed from the client share

the same namespace as {{TraversalSource}} bindings which means that if the {{TraversalSource}} is "g" then you couldn't write a traversal like: {{withSideEffect(“g”, “hello”)}}. That could be smarter as well.

jira_issues_comments:

1. GitHub user spmallette opened a pull request: https://github.com/apache/tinkerpop/pull/723 TINKERPOP-1792 Fixed GremlinScriptEngine bug in lambda processing https://issues.apache.org/jira/browse/TINKERPOP-1792 GremlinScriptEngine instances should now properly select the appropriate TraversalSource from bindings rather than select a random one. The TraversalSource is locally aliased to a "hidden" variable at the time of evaluation to avoid naming clashes with local variables defined in the bytecode itself. Added a number of validations to be sure that users get appropriate errors if they try to use the eval() method the wrong way. Included new tests to ensure that GremlinScriptEngine have the appropriate behavior. All tests pass with `docker/build.sh -t -n -i` VOTE +1 You can merge this pull request into a Git repository by running: $ git pull https://github.com/apache/tinkerpop TINKERPOP-1792 Alternatively you can review and apply these changes as the patch at: https://github.com/apache/tinkerpop/pull/723.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #723 ---- commit 30185646379b051b81484dd75ad34490fa7461b8 Author: Stephen Mallette <spmva@genoprime.com> Date: 2017-09-26T15:31:31Z TINKERPOP-1792 Fixed GremlinScriptEngine bug in lambda processing GremlinScriptEngine instances should now properly select the appropriate TraversalSource from bindings rather than select a random one. The TraversalSource is locally aliased to a "hidden" variable at the time of evaluation to avoid naming clashes with local variables defined in the bytecode itself. Added a number of validations to be sure that users get appropriate errors if they try to use the eval() method the wrong way. Included new tests to ensure that GremlinScriptEngine have the appropriate behavior. ----
2. Github user okram commented on the issue: https://github.com/apache/tinkerpop/pull/723 VOTE +1
3. Github user dkuppitz commented on the issue: https://github.com/apache/tinkerpop/pull/723 Nice set of tests. VOTE: +1
4. Github user asfgit closed the pull request at: https://github.com/apache/tinkerpop/pull/723