Software Engineering PR280

Nick Leslie

Iteration 4

Code can be found: <https://codesandbox.io/s/nkl25x5l7p>

## Plan/Goal

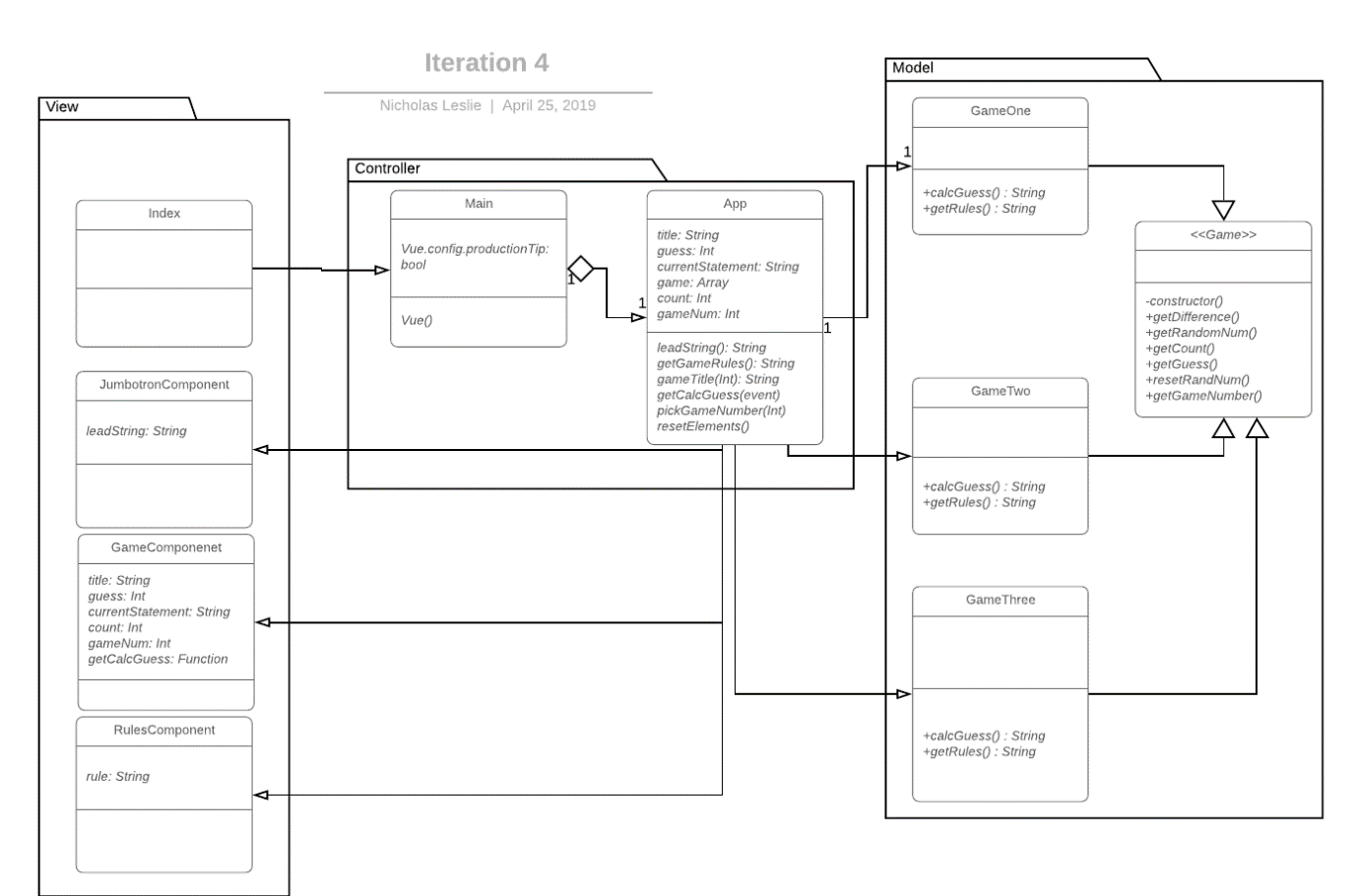
Complete Appendix Two, Task 3.

* Write a program to play a number guessing game. The USER mentally selects a number between 0 and 99 and the computer ties to guess it. The computer outputs its guess, and the User response with "Try higher", "Try lower" or “correct”. The computer should keep count of the number of guesses. The computer should complain if the USER has lied.
* PSP0.1 Project Plan Summary Iteration 4

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Nick Leslie | Date | 25/04/2019 |
| Program | Number Guessing Game | Program # | 4 |
| Instructor | Amit / Luofeng | Language | JavaScript/Vue |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Summary** | **Plan** | | |  | **Actual** | | |  | **To Date** | | |
| LOC/Hour | **100** | | |  |  | | |  |  | | |
| Planned Time | **120** | | |  |  | | |  |  | | |
| Actual Time |  | | |  |  | | |  |  | | |
| ~~% Reused~~ |  | | |  |  | | |  |  | | |
| ~~% New Reused~~ |  | | |  |  | | |  |  | | |
|  | | | | | | | | | | | |
|  |  | | |  |  | | |  |  | | |
| **Program Size (LOC)** | **Plan** | | |  | **Actual** | | |  | **To Date** | | |
| Base (B) | 276 | | |  | 221+68+ | | |  |  | | |
|  | (Measured) | | |  | (Measured) | | |  |  | | |
| Deleted (D) | 0 | | |  | 0 | | |  |  | | |
|  | (Estimated) | | |  | (Counted) | | |  |  | | |
| Modified (M) | 10 | | |  | 6 | | |  |  | | |
|  | (Estimated) | | |  | (Counted) | | |  |  | | |
| Added (A) | 50 | | |  | 66 | | |  |  | | |
|  | (A + M - M) | | |  | (T − B + D − R) | | |  |  | | |
| Reused (R) | 276 | | |  | 276 | | |  |  | | |
|  | (Estimated) | | |  | (Counted) | | |  |  | | |
| Added and Modified (A + M) | 60 | | |  | 72 | | |  |  | | |
|  | (Projected) | | |  | (A + M) | | |  |  | | |
| Total Size (T) | 336 | | |  | 364 | | |  | 276 | | |
|  | (A +M + B - M – D + R) | | |  | (Measured) | | |  |  | | |
| Total New Reusable |  | | |  |  | | |  |  | | |
|  | | | | | | | | | | | |
| **Time in Phase (min.)** | **Plan** |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning | 20 |  | 19:14 | | |  | 1:03:18 | | |  |  |
| Design | 40 |  | 22:29 | | |  | 1:39:26 | | |  |  |
| Coding | 60 |  | 2:55:30 | | |  | 6:53:54 | | |  |  |
| Compile | 0 |  | 0 | | |  | 0 | | |  |  |
| Test | 10 |  | 35:08 | | |  | 3:20:23 | | |  |  |
| Postmortem | 10 |  | 13:04 | | |  | 13:04 | | |  |  |
| Total | 160 |  | 4:25:25 | | |  | 13:20:07 | | |  |  |
|  | | | | | | | | | | | |
| **Defects Injected** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  |  |
| Design |  |  | 0 | | |  | 0 | | |  |  |
| Code |  |  | 5 | | |  | 20 | | |  |  |
| Compile |  |  | 0 | | |  | 0 | | |  |  |
| Test |  |  | 0 | | |  | 0 | | |  |  |
| Total Development |  |  | 0 | | |  | 0 | | |  |  |
|  | | | | | | | | | | | |
| **Defects Removed** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  |  |
| Design |  |  | 0 | | |  | 0 | | |  |  |
| Code |  |  | 5 | | |  | 20 | | |  |  |
| Compile |  |  | 0 | | |  | 0 | | |  |  |
| Test |  |  | 0 | | |  | 0 | | |  |  |

## Design Level Class Diagram



## Plan for how the program feature of an iteration you are working will work

**PLANNING A COMPLEX ALGORITHM**

**Iteration 4**

**DESIGN THE ROUTINE**

CHECK PREREQUISITES

Define the problem

*User guesses a number. Computer must return a random number.*

*If the number is low, user says it must be higher. Computer must then return a number higher than its randomly guessed number.*

*If the number is high, user says it must be lower. Computer must return a number lower than its randomly guessed number.*

*Continue until user says that is correct.*

Information the routine will hide

None

Inputs to the routine

*User inputs (Guess is too low, guess is too high, correct answer)*

Outputs from the routine

Randomly generated number between 0 and 99.

Pre-conditions

*Randomly generated number*

Post-conditions

Name the Routine

*calcGuess()*

Decide how to test the routine

Should return a number

If user said “Higher”. Should return a number higher than previous

If user said “Lower”, should return a number lower than previous

Research functionality available in standard libraries

Think about error handling

Think about efficiency

Research algorithms & data types

**PSEUDOCODE**

* User clicks play
* Return random number between 0 and 99
* Try higher
* Random number now between N and 99
* Repeat
* User says correct

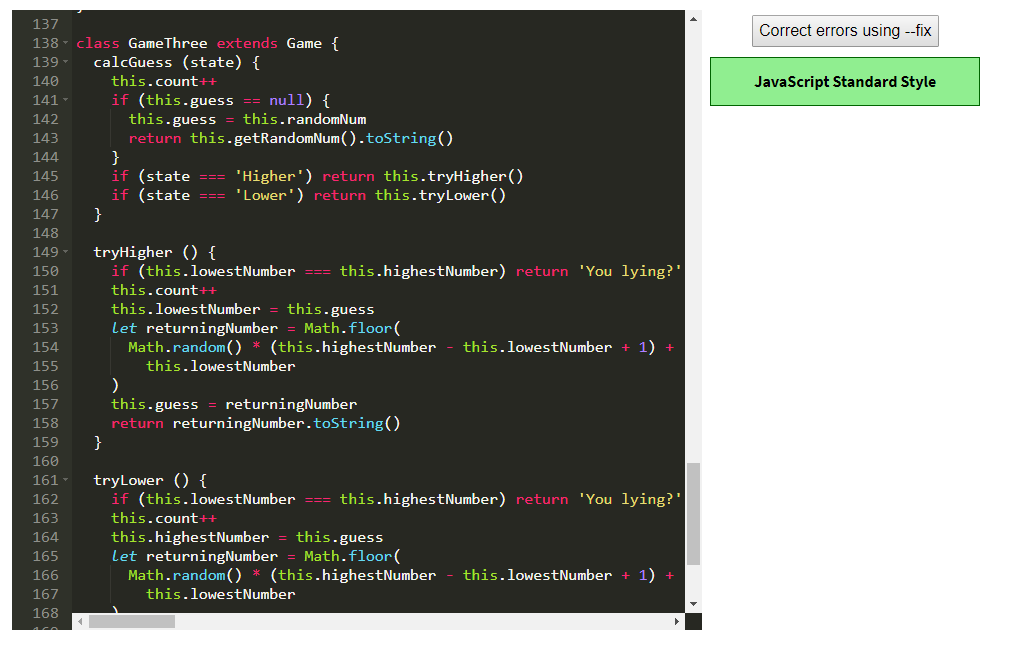
# Testing Planning

1. Test Report Template

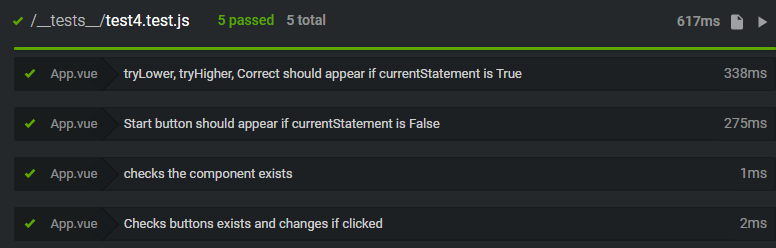
|  |  |  |  |
| --- | --- | --- | --- |
| Student | Nick Leslie | Date | 25/04/2019 |
| Program | Guessing Game | Program # | 4 |
| Instructor | Amit / Luofeng | Language | JavaScript |

|  |  |
| --- | --- |
| Test Name/Number | Test 1 |
| Test Objective | Buttons tryLower, tryHigher, correct should appear if currentStatement is a string of any length |
|  |  |
| Test Description | currentStatement = “This is a message” |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions | gameNum = 3 |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | Buttons tryLower, tryHigher, correct to appear |
|  |  |
|  |  |
|  |  |
| Actual Results | Pass |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Name/Number | Test 2 |
| Test Objective | Start button should appear without currentStatement |
|  |  |
| Test Description |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions | gameNum = 3 |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | Button.start to exist |
|  |  |
|  |  |
|  |  |
| Actual Results | Pass |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Name/Number | Test 3 |
| Test Objective | Start button to exist, and no exist once it is clicked |
|  |  |
| Test Description | Game: [new GameThree(null, 1)], currentStatement: ‘’ |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions | gameNum = 1 |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | Button.start to exist, and no longer exist once clicked |
|  |  |
|  |  |
|  |  |
| Actual Results | Pass |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Name/Number | Test 4 |
| Test Objective | To see if Game three is added |
|  |  |
| Test Description | game: [  new GameOne(null, 1),  new GameTwo(null, 2),  new GameThree(null, 3)  ] |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | Three GameComponents to be loaded |
|  |  |
|  |  |
|  |  |
| Actual Results | Pass |
|  |  |
|  |  |
|  |  |
|  |  |

# StandardJS report



# Run the tests



# Error Log (With at least 5 errors)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 28/04/2019 | 16 | 80 | Code | Compile | 1 | 16 | [Vue warn]: Property or method "tryLower" is not defined on the instance but referenced during render. Make sure that this property is reactive, either in the data option, or for class-based components, by initializing the property. See: https://vuejs.org/v2/guide/reactivity.html#Declaring-Reactive-Properties. - Failed to declare function in props |
| 28/04/2019 | 17 | 20 | Code | Compile | 1 | 17 | SyntaxError: /src/App.vue: Unterminated string constant (223:11) |
| 28/04/2019 | 18 | 70 | Code | Compile | 1 | 18 | Random number would remain the same throughout testing, as well as highestNumber/lowestNumber. Added reset function to clear everything |
| 28/04/2019 | 19 | 70 | Code | Compile | 1 | 19 | [Vue warn]: Invalid prop: type check failed for prop "currentStatement". Expected String with value "78", got Number with value 78. - .toString() added |
| 28/04/2019 | 20 | 70 | Code | Compile | 1 | 20 | [Vue warn]: Invalid prop: type check failed for prop "currentStatement". Expected String with value "82", got Number with value 82. - .toString() added |

# Process Improvement Proposal (With at least 5 PIPs)

|  |
| --- |
| Problem Description |
| Briefly describe the problems that you encountered. |
| Lack of organization and prioritization |
| Code duplications |
| Code needs refactoring |
|  |
|  |
| Proposal Description |
| Briefly describe the process improvements that you propose. |
| 1. Proper prioritizing (Trello, Kandan) |
| 1. Check off list of completed tasks |
| 1. Split files up, so Game related logic is in its own file |
| 1. Refactor final iterations code to remove repetition and redundancy |
| 1. Use a final, unused, design technique for a program feature (Question 3) |