**Iteration 3#**

## Requirements Statement

1. Create a Single Page Application (SPA) using Vuejs and JavaScript ES2018.
2. 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.

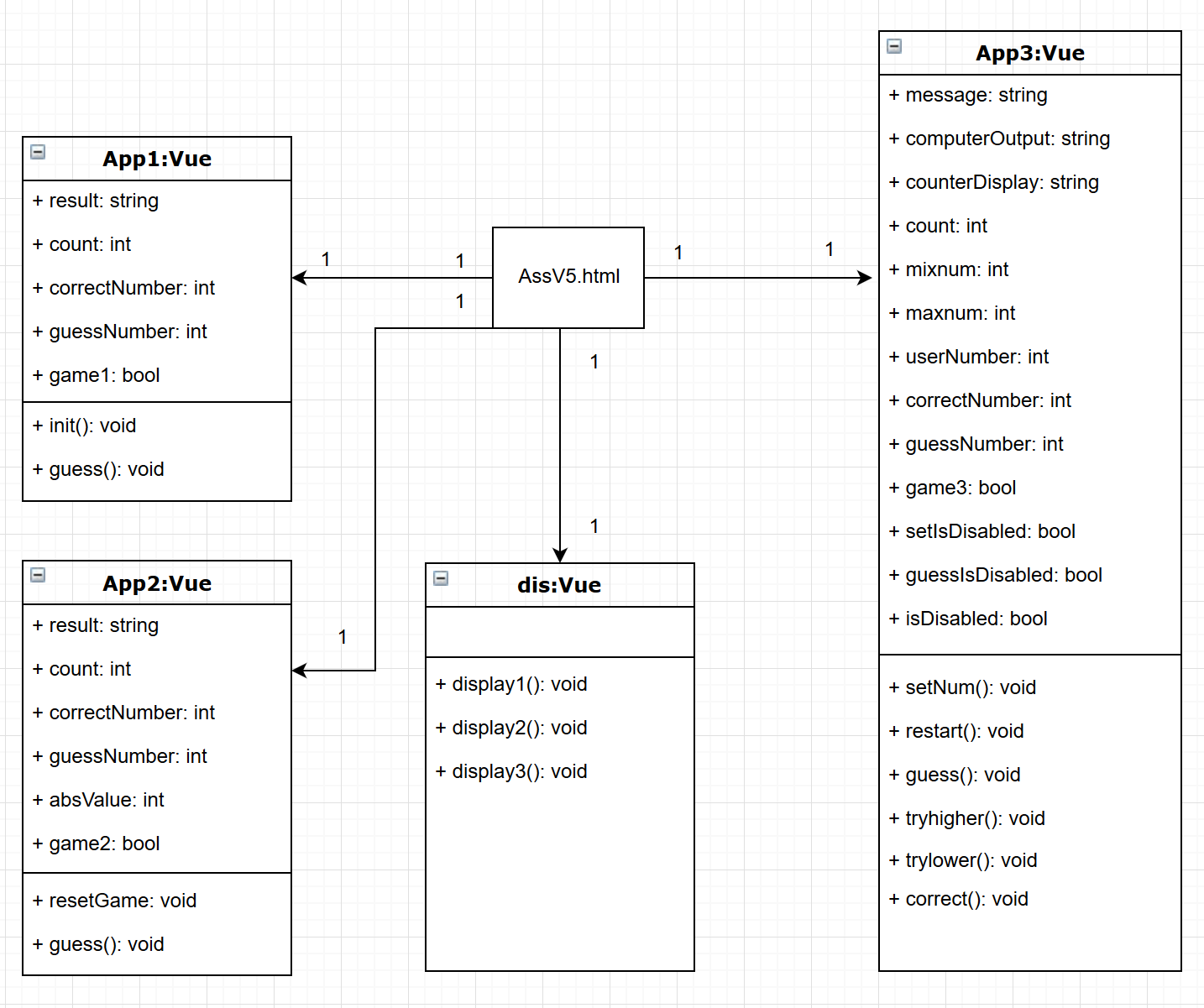
## Iteration Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Milestone | Planned Start Date | Actual Start Date | Planned End Date | Actual End Date | Task |
| Iteration |  |  |  |  |  |
| Planning | 26/8/2019 | 26/8/2019 | 26/8/2019 | 26/8/2019 | 1.Write a programming, which can play Number Guess Game with computer. The USER 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.  3.Layout four games in one page. Use four buttons to choose the game. |
| Analysis | 26/8/2019 | 26/8/2019 | 26/8/2019 | 26/8/2019 | Analyze project requirements. List the required functionality for the project. |
| Design | 27/8/2019 | 27/8/2019 | 27/8/2019 | 27/8/2019 | 1.Design the Single Page Application’s website Layout.  2.Use Case for the programming. |
| Coding | 29/8/2019 | 29/8/2019 | 29/8/2019 | 29/8/2019 | Coding for the second literation. |
| Testing | 31/8/2019 | 31/8/2019 | 31/8/2019 | 31/8/2019 | Make a testing plan and do the test.  Record the result of the test. |

## Task Summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Time in Phase (hr.)** | **Plan** | **Actual** | **Start** | **Stop** | **To Date** | **To Date %** |
| Planning | 0.5hr | 0.5hr | 14:00 | 14:30 | 26/8/2019 | 10% |
| Analysis | 0.5hr | 0.5hr | 15:00 | 15:30 | 26/8/2019 | 10% |
| Design | 0.5hr | 0.5hr | 11:00 | 11:30 | 27/8/2019 | 10% |
| Coding | 1.5hr | 1.5hr | 9:00 | 10:30 | 29/8/2019 | 30% |
| Test | 1.5hr | 1.5hr | 12:00 | 13:30 | 31/8/2019 | 30% |
| Postmortem | 0.5hr | 0.5hr | 18:00 | 18:30 | 1/9/2019 | 10% |
| Total | 5hr | 5hr |  |  |  | 100% |

## Class Diagram



## Plan for how feature works: Use Case

1. Description:

User 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.

1. Pre-Conditions:

User open the main page of the game, click the button “Game Three”, and the third game appear to the website. In the initial state of the game, only two buttons are available, “Set Number” and “Restart Game”. The message shows “Please enter a number (between 0-99) for computer guess.”

1. Main Flow:
2. The user enters a number between 0 and 99 (e.g.: 45), click the button “Set Number”. The message shows “You selected number is: 45). Button “Set Number” unavailable, button “Begin to Guess” available.
3. User click button “Begin to Guess”, computer begin to guess the number. Counter shows “Time:1”, Computer output “Computer: I guess your number is 49”. Buttons “Try Higher”, “Try Lower”, “Correct” are available, button “Begin to Guess” unavailable.
4. User can use the buttons “Try Higher”, “Try Lower”, “Correct” to tell the computer whether its guess correct. User click “Try Higher”, the computer will output a number between last guessed number and the max value, and vice versa. The counter keep count and keep change to its output.
5. When the computer gets the correct number and user click button “Correct”, game over. The message shows “Congratulation!! You got it in n trials.”. Computer output “YOHO!!”. Only button “Restart Game” available, which ask the user to restart a new game.
6. Alternative Flow:
7. When the user mistakenly selects a key and the computer has no Numbers to guess, the computer outputs "you lied to me."
8. When the user mistakenly selects “Correct”, but the guess number is not equal to correct number, the computer outputs "you lied to me."
9. Exception Flow:
10. When the textbox of Set number is empty, user cannot start the game.
11. When user enters non-digital information, or the number is out of range, the message shows “Invalid enter! Try again please.”
12. Post- Conditions:

User can use button “Restart Game” to restart the game, or choose one of the four buttons, at the top of the website, to choose another new game to play.

## Test Plans:

|  |  |  |
| --- | --- | --- |
|  | Test Plans | Expected Results |
| 1 | Test the function of button “Set Number”. | When the user enters a significant number, the information displays the correct number entered by the user. When the user enters invalid information, the message prompts the user entered an invalid enter and needs to enter again. |
| 2 | Test the function of button “Restart Game”. | When the user clicks the button “Restart Game”, the game returns to its original state. |
| 3 | Test the game, including “Begin to Guess”, “Try Higher”, “Try Lower”, “Correct”. | User use the button “Begin to Guess” to start the game.  User can use the buttons “Try Higher”, “Try Lower”, “Correct” to tell the computer whether its guess correct. User click “Try Higher”, the computer will output a number between last guessed number and the max value, and vice versa. User click “Correct”, game over. |
| 4 | Test the counter. | Play the game and record the number of guesses. Compare with the final output. |

## Test Report Template

|  |  |
| --- | --- |
| Test Name/Number | 1 |
| Test Objective | Test the function of button “Set Number”. |
| Test Description | When the user enters a significant number, the information displays the correct number entered by the user. When the user enters invalid information, the message prompts the user entered an invalid enter and needs to enter again. |
| Test Conditions | No options are given, default options apply. |
| Expected Results | 1.User enters a valid number 45, click the button “Set Number”. The message shows “You selected number is: 45”. Button “Set Number” unavailable, button “Begin to Guess” available.  2. User enters a string, message shows “Invalid enter! Try again please.”  3. User enters a number out of range, message shows “Invalid enter! Try again please.”  4. The textbox is empty, user click button “Set Number”, nothing happen. |
| Actual Results | 1.User enters a valid number 45, click the button “Set Number”. The message shows “You selected number is: 45”. Button “Set Number” unavailable, button “Begin to Guess” available.  2. User enters a string, message shows “Invalid enter! Try again please.”  3. User enters a number out of range, message shows “Invalid enter! Try again please.”  4. The textbox is empty, user click button “Set Number”, nothing happens. |
| Test Name/Number | 2 |
| Test Objective | Test the function of button “Restart Game”. |
| Test Description | When the user clicks the button “Restart Game”, the game returns to its original state. |
| Test Conditions | No options are given, default options apply. |
| Expected Results | Clicks the button “Restart Game”, the game returns to its original state.  1. Only “Set Number” and “Restart Game” buttons available.  2. Counter value is 0.  3. Message shows "Please enter a number (between 0-99) for computer guess.”. |
| Actual Results | Clicks the button “Restart Game”, the game returns to its original state.  1. Only “Set Number” and “Restart Game” buttons available.  2. Counter value is 0.  3. Message shows "Please enter a number (between 0-99) for computer guess.”. |
| Test Name/Number | 3 |
| Test Objective | Test the game, including “Begin to Guess”, “Try Higher”, “Try Lower”, “Correct”. |
| Test Description | User use the button “Begin to Guess” to start the game.  User can use the buttons “Try Higher”, “Try Lower”, “Correct” to tell the computer whether its guess correct.  User click “Correct”, game over. |
| Test Conditions | No options are given, default options apply. |
| Expected Results | 1.User click the button “Begin to Guess”, button “Set Number” unavailable, buttons “Try Higher”, “Try Lower”, “Correct” available. Computer begin to guess the number. Counter begin to count.  2. User click “Try Higher”, the computer will output a number between last guessed number and the max value, and vice versa. Counter keep count.  3. User click “Correct”, game over. The message shows “Congratulation!! You got it in n trials.”. Computer output “YOHO!!”. Only “Restart Game” button available.  4. When the user mistakenly selects a key and the computer has no Numbers to guess, the computer outputs "you lied to me."  5. When the user mistakenly selects “Correct”, but the guess number is not equal to correct number, the computer outputs "you lied to me." |
| Actual Results | 1.User click the button “Begin to Guess”, button “Set Number” unavailable, buttons “Try Higher”, “Try Lower”, “Correct” available. Computer begin to guess the number. Counter begin to count.  2. User click “Try Higher”, the computer will output a number between last guessed number and the max value, and vice versa. Counter keep count.  3. User click “Correct”, game over. The message shows “Congratulation!! You got it in n trials.”. Computer output “YOHO!!”. Only “Restart Game” button available.  4. When the user mistakenly selects a key and the computer has no Numbers to guess, the computer outputs "you lied to me."  5. When the user mistakenly selects “Correct”, but the guess number is not equal to correct number, the computer outputs "you lied to me." |
| Test Name/Number | 4 |
| Test Objective | Test the counter. |
| Test Description | Play the game and record the number of guesses. Compare with the final output. |
| Test Conditions | No options are given, default options apply. |
| Expected Results | The record number is equal to final output. |
| Actual Results | The record number is equal to final output. |

## Error Logs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **No.** | **Type** | **Inject** | **Remove** | **Fix Time**  **(10min/block)** | **Fix Ref** | **Description** |
| **29/8/2019** | 1 | 20 | Code | Compile | 1 | trouble shooting | Vm3 is not defined,add”<script src="myScript3.js"></script> insert to html”. |
| **29/8/2019** | 2 | 20 | code | Compile | 1 | trouble shooting | Unexpected token, Add “,” |
| **31/8/2019** | 3 | 20 | code | Compile | 1 | trouble shooting | Keep clicking the “try lower” button, the last guess number is negative. Change the condition of the If judgment from” this.mixnum == 0” to “this.guessNumber == this.mixnum”. |
| **31/8/2019** | 4 | 20 | code | Compile | 1 | trouble shooting | Enter non-numeric or out-of-range Numbers, the game is still work. Make a logical judgment of what is entered. Invalid input cannot start the game. |
| **31/8/2019** | 5 | 20 | code | Compile | 1 | trouble shooting | When the game started, button” begin to guess” is still available, click it the counter added.  When the game started, disable the “begin to guess” and “set number” buttons. If user want to restart the game, the only way is clicking the “Restart Game” button. |
| **31/8/2019** | 6 | 20 | code | Compile | 1 | trouble shooting | When user click the “Correct”, the game still works. “Try higher”, “Try lower”, “Begin to guess” buttons still work, computer can continue guess, and counter add.  When user click the “Correct” button, disable guess buttons. |

## PSP Process Improvement Proposal (PIP)

|  |
| --- |
| Problem Description |
| Briefly describe the problems that you encountered. |
| The setNum function cannot tell whether user input is valid or not. |
| The "begin to guess" button is not disabled once click it. When the game started, user clicking this button does not do anything useful but increases the value of the counter. |
| The conditions of tryhigher(), trylower() functions are wrong. The error resulted in inaccurate output of the "you lied to me" message and the guess number can be negative. |
| Click “Correct” button should end the game. |
| Error Syntax. |
| Proposal Description |
| Briefly describe the process improvements that you propose. |
| Add logical judgment to avoid invalid input. |
| Disable certain buttons during the game to make the game run smoothly. |
| Trouble shooting, fix the wrong logic. |
| Fix the “Correct” button’s function. |
| Trouble shooting, fix the Error syntax. |
| Other Notes and Comments |
| Note any other comments or observations that describe your experiences or improvement ideas. |
| 1. Found some bug in literation 1 and literation 2. In next literation should fix them. |
| 1. The coding structure may be change to the popular one, like MVC. |
| 1. Use some new method of Vue. |