CSE3026: Web Application Development

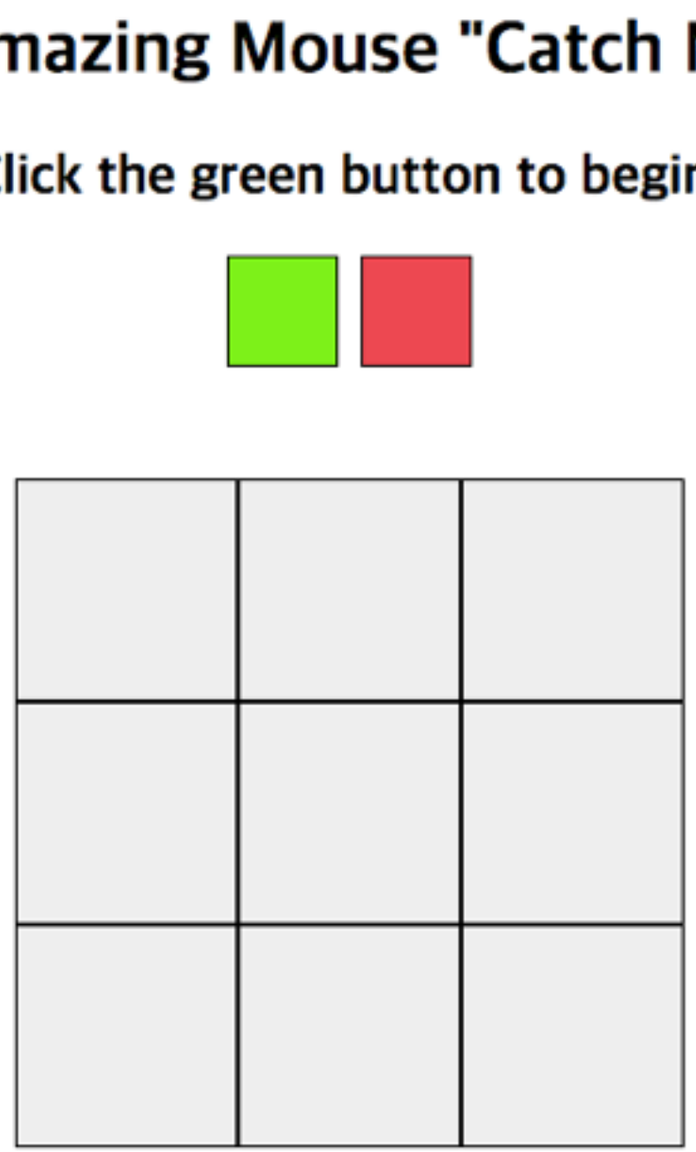
Lab 10:Prototype

Software Engineering Lab

lab(se);

Lab 10 exercises

Create a simple game that enhances your memory.



lab(se);

About the Game

Download game.zip to get started!

The rule of the game is to memorize randomly picked blocks and select them. The blocks are 9 div elements. Our provided CSS puts the divs into their proper places.

Problem contraints

In order to get full marks, you must satisfy the following constraints.

- Do not modify the given game.html, game.css
- Use only the Prototype library (do not use or import any other JS library).
- Write all your JavaScript code unobtrusively!
- must use “use strict” in your JavaScript.
- Format your JavaScript as readable as possible. Use proper indentation.
- Allow to add any functions in game.js but you should write code clearly or add comments.
- Do not add additional global variables.
- When you get elements using id or class, use $, $$ function.

Ex 1. Dom loaded & Starting game

Download game.js and write code so that when the user clicks either green or red button, proper function is called.

- Write a dom:loaded handler that sets up event handler related with both red and green buttons.
- Handle the event by calling proper function.
  - When red button is clicked, call stopToStart function.
  - When green button is clicked, call stopGame function.
- Implement stopGame function
  - Change State to “Stop”.
  - Change Answer to “0/0”.
  - Reset all components for example timer, classes, array.

lab(se);

Ex 2. Pick random blocks and show that

Write the code in the game.js so that random blocks is set and show that by adding a class.

- Implement startToSetTarget function in order.
  - Change state to “Ready!”.
  - Reset all components like previous.
  - Assign randomly picked blocks to variable named targetBlocks.
  - Call setTargetToShow function after 3sec using “interval” that is defined at top.
- Implement setTargetToShow function in order.
  - Change state to “Memorize!”.
  - Add “target” class to randomly picked blocks.
  - Call showToSelect function after 3sec.

lab(se);

Ex 3. Select answers

Write the code in the game.js so that the user select answers.

- Implement showToSelect function in order.
  - Change state to “Select!”.
  - Remove “target” class added to blocks.
  - Attach an event handler to each block.
    - When the user click a block, add “selected” class to that.
    - All selected blocks assign to variable named selectedBlocks.
    - the number of selected blocks doesn’t exceed numberOfTarget that is defined at top.
    - Selecting same block is not allowed.
- Call selectToResult function after 3sec.

Ex 4. Check answers and show result

Write the code in the game.js so that check the user's selected block whether correct or incorrect.

- Implement selectToResult function in order.
  - Change state to "Checking"
  - Remove "selected" class added to blocks.
  - Detach event handler to prohibit selecting more answers by user.
  - Check correct answers and change answer to "# of correct answer/# of total problem blocks"
    - the answer label is retained before the user click the red button.
    - Any global variables for answer label is not allowed.
- Call startToSetTarget function after 3sec to restart the game.

If you finish them all…

If you finish all the exercises:

1. you can add any other content you like to your page
2. check in with a TA to get credit for your work
3. you may be able to be dismissed

If you don't finish all the exercises on time:

1. **upload your files to gitHub** at least 16:55
2. you can get some marks from your uploaded files
3. you **can't get any marks** if you submit result after 17:00 Wednesday

Great Work!

lab(se);