Name ____ Period______

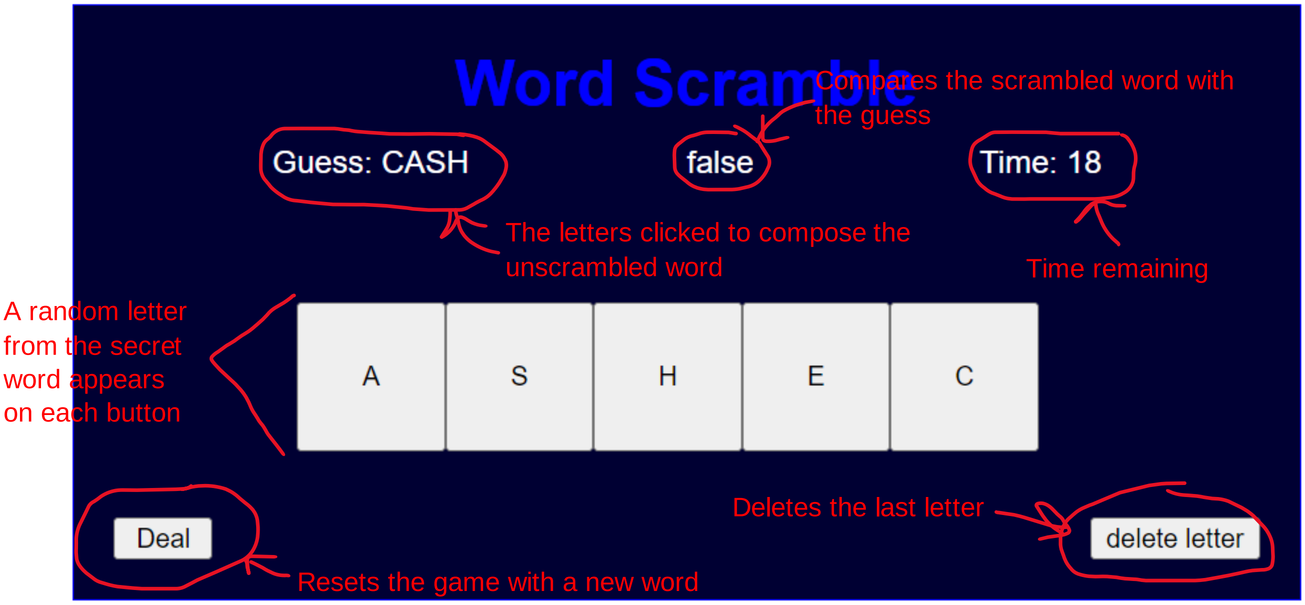
Word Scramble

Your Tasks (Mark these off as you go)

□ Review the program requirements
□ Write a function to create the buttons
□ Write the `scrambleWord` function
□ Write the `checkWord` function
□ Write the `deal` function
□ Write the `deleteLetter` function
□ Write the `updateTime` function
□ Write the `gameOver` function
□ Receive credit for this lab guide

□ Review the program requirements

For this lab you will create a game that challenges the user to guess a scrambled word. How the game works is described below,



Compares the scrambled word with the guess

The letters clicked to compose the unscrambled word

Time remaining

A random letter from the secret word appears on each button

Deletes the last letter

Resets the game with a new word

When the game loads, a random secret word is selected from a word bank. The word is scrambled by placing a different letter (chosen at random) from the word on each button.

As the user clicks on the buttons, the letters appear in the Guess field.

The user has the option to delete the last letter clicked by clicking the delete letter button.

When the guess is the same as the secret word, the word false changes to true.

The time left counts down to 0 and is displayed in the Time field.

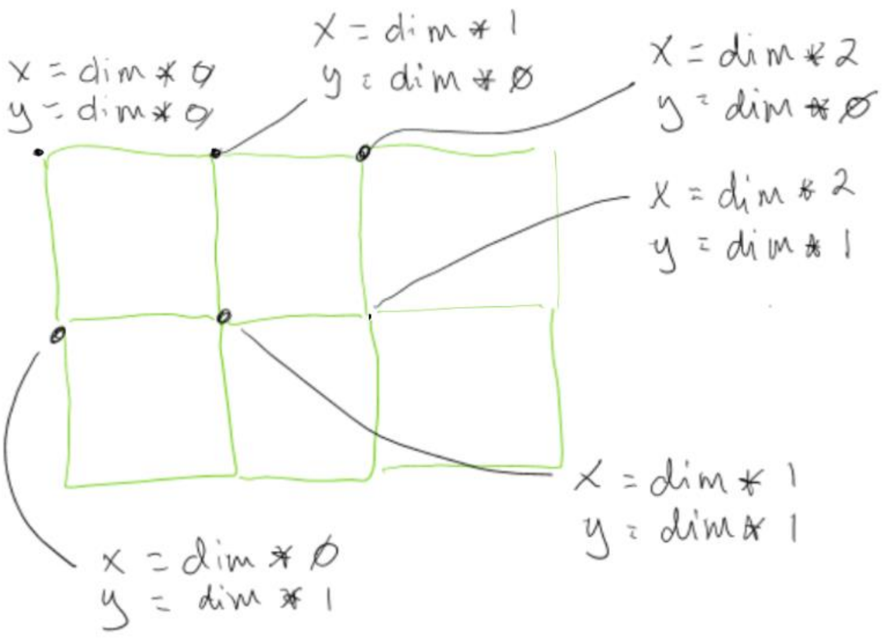
When the Deal button is clicked,

- A new word is selected and the buttons are populated with random letters from the word
- The boolean field is set to false
- The guess field is set to “”
- The time is reset to 30 and the count down resumes

□ Write a function to create the buttons

In the previous labs, you wrote code to create grids of buttons. Below we will review how we did this.

First, we created a function that accepted 3 parameters which represented the dimension, x position, and y position of each button. We then declared a variable (`dim`) to represent the dimension of each button. This new variable allowed us to create many buttons in terms of this dimension. The placement of each button in terms of the dim is illustrated below,



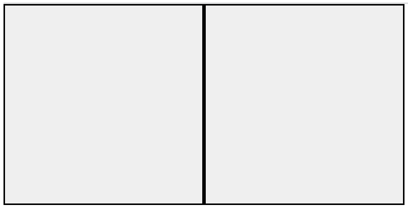
To make our grid interactive required that we add action listeners to each of the buttons. To tell us which button is clicked, we also added an id to help us identify which button was clicked.

The example below illustrates how to create two interactive buttons using our function. `checkWord` is the function called each time a button is clicked. You will implement `checkWord` later.

Code | **Output**

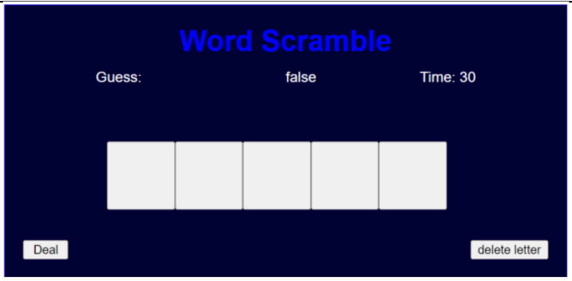
```
var dim = 50;
var b0 = makeButton(dim, dim*0, dim*0);
var b1 = makeButton(dim, dim*1, dim*0);

function makeButton(d, xPos, yPos){
  var b = document.createElement("button");
  b.style.width = dim + "px";
  b.style.height = dim + "px";
  b.style.position = "absolute";
  b.style.top = yPos + "px";
  b.style.left = xPos + "px";
  b.addEventListener("click", checkWord)
  gameDiv.append(b);
  return b;
}
```



Refer to the makeButton function above. Write code that calls the function to create the grid shown below. The position of each button should be defined in terms of dim.

Code | **Output**



□ Write the scrambleWord function

When the game loads, a random word is selected from a word bank and is assigned to a variable called secretWord. Each letter from the secretWord must appear on a different button. Consider the algorithm below which could be used to accomplish this task. To better understand how the algorithm works, let's give the secretWord a value,

secretWord = "EARTH"

Step 1: Assign the secret word to a temp variable called letters (letters = "EARTH")
Step 2: Select 1 random letter from the letters and place this on the first button (for example, "T")
Step 3: Remove all the letters before and after the letter selected in the previous step and reassign the result to letters (letters = "EARH")
Step 4: Select 1 random letter from the letters and place this on the next button (for example, "E")

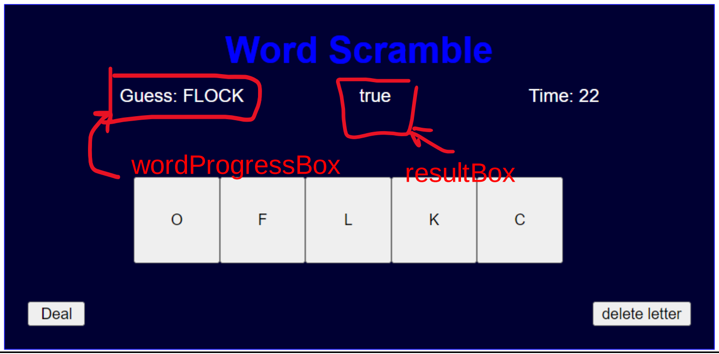
Step 5: Remove all the letters before and after the letter selected in the previous step and reassign the result to letters (letters = “ARH”)

Repeat the above until the last letter has been placed.

Refer to the algorithm above that could be used to populate the buttons with random letters from the secret word. Write code to implement this algorithm.

☐ Write the checkWord function

Each time a button is clicked, checkWord is called. As the buttons are clicked the letters need to populate the wordProgressBox. If the guess is the same as the secret word, the resultBox needs to update to true. This is illustrated below. Write the checkWord function below.



☐ Write the deal function

Each time the game loads or the deal button is clicked we need to do the following,

- Call getWord() (written for you)
- Call scrambleWord() (Written previously)
- Clear the timeout timer (endGameTimer)
- Clear the interval timer (gameTimer)
- Reset the time to 30
- Reset the wordProgress variable
- Result the resultBox
- Reset the wordProgressBox
- Reassign the endGameTimer with a new timeout timer, that calls gameOver after time seconds
- Reassign the gameTimer with a new interval timer that calls updateTime every second

Write the deal function the meets these requirements.

□ Write the deleteLetter function

Each time the delete button is clicked, the `deleteLetter` function is called. The delete letter function removes the last letter from the `wordProgress` variable and updates the `wordProgressBox`. Write the `deleteLetter` function below.

□ Write the updateTime function

The `updateTime` function keeps track of the time left and displays it in the `timeHolder` div. Write the `updateTime` function below.

□ Write the gameOver function

The `gameOver` function is called when the time runs out. The `gameOver` function alerts the user that the game is over along with the secret word. It also clears the interval time, `gameTimer`. Write the `gameOver` function below.

□ Receive Credit for this lab guide

Submit this portion of the lab to Pluska to receive credit for the lab guide. Once received, your completed code challenges will also be graded and will count towards your final lab grade.