Lab 4. Math Game Version 4

Here are the user stories for version 4:

- Allow users to choose from four different types of math operations: addition, subtraction, multiplication, and division. (The division can be the floor division without remainders.)
- Allow users to choose five different difficulty levels where the range of numbers will differ.

4.1. Demo Video

At the end of this lab, you need to achieve an outcome as shown by the screenshot in the demo video (available at QMPlus). There is a new box on the top as a setup zone, as shown in Figure 1.

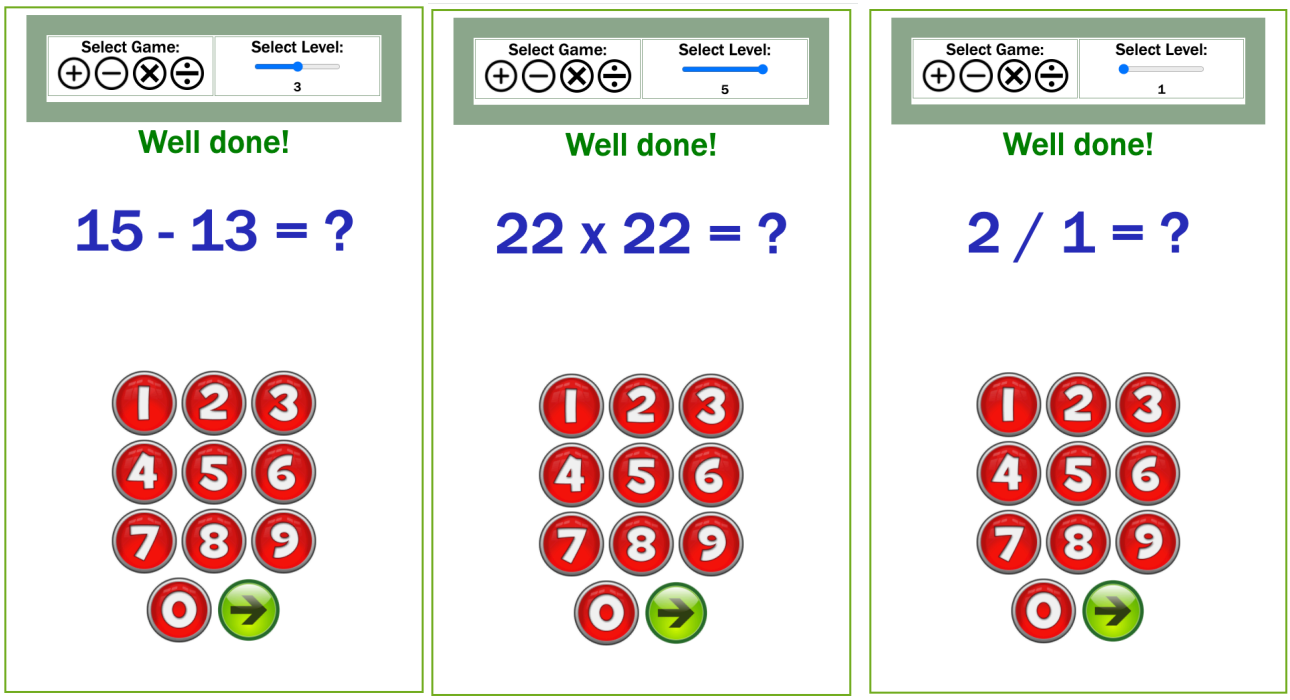

Figure 1. Three Possible Messages During Game Play

4.2. Design Thinking

CSS and HTML: A new box is needed on top of the existing boxes. Inside the box four buttons and a slider as well as some text info are required. Again, it would be easier in JS coding if these buttons are named after their respective operators.

JS: Some functions need to be updated to reflect the change of operator.

4.3. Implementation

Follow these steps to make changes from the last version:

1. Add a new box with the class name of "setup". Define the style appropriately with CSS.

2. Add four types of operation buttons within the “setup” box. Define the style appropriately with CSS. Name the button IDs as “+”, “-“, “x”, “/”.

3. Add a level slider with a text indicator using this code:
```
<input type="range" min="1" max="5" value="1" step="1" id="slider"><br>
<span id="level" style="font-size: 20px;"></span>
```

4. Now move to the JS code area.

5. In the beginning of the JS code, add some declaration for a slider object:
```
slider = document.getElementById("slider");
document.getElementById("level").innerHTML = slider.value;
```

6. Add a function for the slider:
```
slider.oninput = function(){
document.getElementById("level").innerHTML = this.value;
}
```

7. Add a function for `changeMode()` which the operation buttons can be associated with.

8. In the beginning of JS code, declare a global variable `mode="+"` . (By default it is within the addition mode.)

9. Within the function `changeMode()` retrieve the object ID and assign it to `mode`, then call the `newQuiz()` function to start a new quiz.

10. Inside `buttonPressed()` function, update the displayed message so it includes `mode`.

11. Update `newQuiz()` the function with the operands reflecting the levels:
```
var max = 5*slider.value;
var min = 1+5*(slider.value-1);
left = Math.floor(Math.random() * (max - min + 1)) + min;
right = Math.floor(Math.random() * (max - min + 1)) + min;
```

12. Update `checkAnswer()` the function with the operands reflecting the levels:
```
switch(mode) {
          case "-":
          result = left-right;
          break;
          case "x":
          result = left*right;
          break;
          case "/":
          result = Math.trunc(left/right);
          break;
          default:
          result = left+right;
          }
```

4.5. Troubleshooting

If you have done something tragic and cannot revert to the last working order, you can always count on GitHub and download a historical version.

4.6. Submission

Please submit your work on QMPlus/EBU6305/Assessment/**Lab 4**.

Please note – if you are not attending the lab in person, your work will NOT be assessed, and you will receive zero mark for your submission.

Make sure all related files (including images) are within one project folder. Zip the project folder as one file to submit on QMPlus.

The following marking criteria will be used. QMPlus marking rubrics can only use integer points system. The equivalence of points and marks are as shown below.

Attempt in the lab	0.5 mark	1 point
Difficulty Levels	1 mark	2 points
Different operations: + - * /	1 mark	2 points
Total	2.5 marks	5 points

4.7. Future Improvement (Optional)

- Replace / with ÷ symbol, considering that the target users are primary school children who may not have the knowledge of fraction.
- The setup selection button should be highlighted when selected. This would comply with the UI design principles.
- Adjust the random range for each level to make it more realistic.
- Add a backward-deletion key to allow users to delete an entry.
- Implement mobile compatibility so that the web page can be displayed on mobile phones appropriately.