Lab 1. Math Game Version 1

You will use the four lab sessions to practice the practical skills you have learned during the module, including HTML, CSS, JavaScript, as well as production management practice.

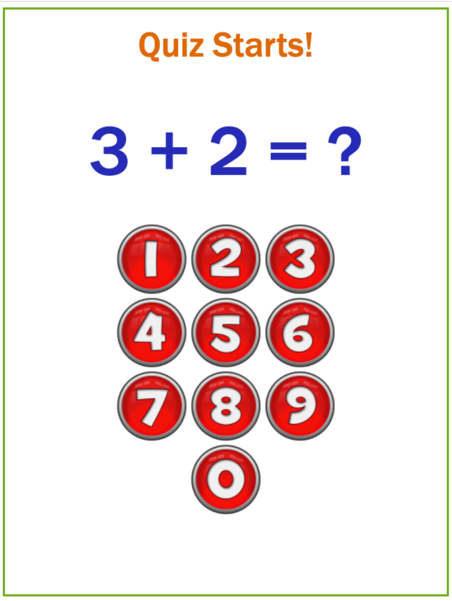
Assume you have been asked by a client to produce a web-based math game for primary school children. You will spend the remaining semester to complete this job, in the meantime following the agile and version control practice. At the end of each lab session, you are supposed to deliver an updated version of the product.

This is version 1.

Initially, the client just wishes to see whether you have grasp the picture of the game design, so you will produce a relatively static user interface with basic elements that you can present to the client. The version 1 product does not allow for any user interaction yet.

1.1. Demo Video

At the end of this lab, you need to achieve an outcome as shown by the screenshot in Figure 1. Please also watch the demo video (available at QMPlus) for the showcase. You should aim to produce an interface that is identical to the given showcase.



Quiz Starts!

3 + 2 = ?

Figure 1. UI Design

1.2. Design Thinking

If you analyse the showcase design, the area can be divided into several boxes, as shown in Figure 2.

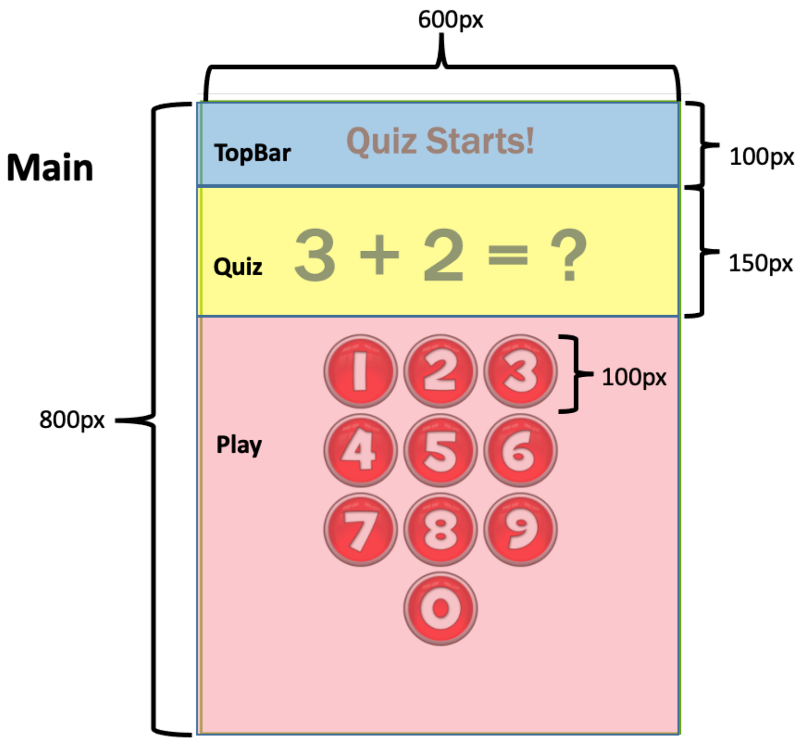


Figure 2. UI Design

There are mainly four boxes in given design:

- The outer green frame as shown in Figure 1 is the **Main** area, which contains everything included in this game.
- The **Top Bar** area can be used to display messages.
- The **Quiz** area will present the math questions.
- The remaining area is the **Play** area. The definition of this area is optional. Once the above three areas are defined, the buttons can be simply placed underneath Top Bar and Quiz boxes.

1.3. Implementation

Follow these steps to develop the web page:

1. Open an empty HTML editor. Put the standard template in.

```
<!DOCTYPE html>
<html>
<head>
  <style>
  </style>
</head>
<body>
</body>
</html>
```

2. Within `<style>` define the boxes as three or four classes. The attributes of the classes can be defined based on the observation of the demo video.

3. Place four `<div>` boxes in the `<body>`, each following a particular class as defined. Write some initial text messages to check the layout, alignment, and font styles. Without actual buttons, your interface may look like as shown in Figure 3. (Temporarily use number text to replace the buttons.)

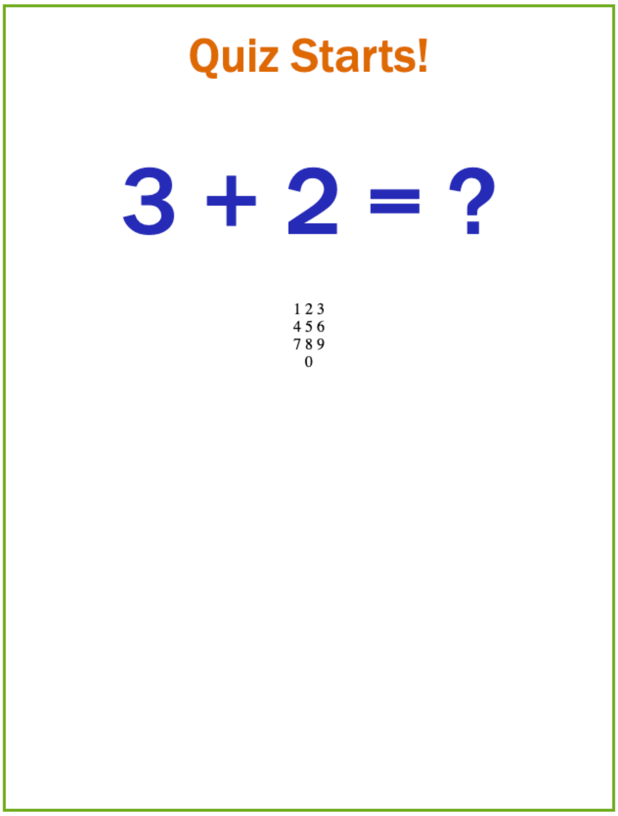


Figure 3. Initial Interface

4. Within `<style>`, give more definition to the existing <button> element (mainly give the width, height, and border).

5. Create a sub folder called ***images*** to keep all image files. All image files for the labs can be downloaded at QMPlus.

6. Add 10 buttons in the play area. Choose background images for the buttons. The code to specify a background image for a button is:

```
style="background: url(images/number1.png); background-size: contain;"
```

7. Add special effect. You may have noticed a special effect from the demo video – when a button is clicked, there is a shadow under the button. This can be implemented using the following code:

```
button:active{
  box-shadow: 0 5px #666;
  transform: translateY(4px);
}
```

8. You may also notice the frame around the button when it is clicked. Using the following code to remove it.

```
button:focus{
  outline:none;
}
```

1.4. Version Control

Now that you have completed the first version of your product, upload it to GitHub as a backup. First of all, always form a good habit of putting all project-related files within **one project folder**.

By default, the repositories you create in GitHub is public, which means everybody in the world can access them. It normally costs a subscription fee to create private repositories. The good news is it is free for students.

Please visit https://education.github.com/pack, and register for a Student Developer Pack using your university mail address. There may be an authentication process involved, which should be relatively straightforward. From now on when creating a new repository, make sure you choose the "**Private**" option instead of "Public".

Follow these steps to commit: (Watch the demo video in Lecture XX for more details.)

1. Click "New Repository"
2. Give a name to your repository, and choose "Private". Initialize the repository with a ReadMe file.
3. Upload the whole project folder (the image folders should be a subfolder) to the repository. Add some descriptions, and commit.

1.5. Submission

Please ask your TA to examine your work when you complete. Please present the following two items:

- Math Game version 1 web page
- GitHub repository commit