**Fishing Game Development Specification**

**Overview**

The goal of this project is to develop a fishing game that involves a learner attempting to catch fish and then correctly answering questions. The game will have the following features:

- There will be a total of 16 fish swimming in the background.

- There will be 16 different questions available in the game.

- The learner can attempt a maximum of 10 questions and therefore, catch 10 fish.(8 correct answers and you win)

- The learner must answer 8 questions correctly to win the game with an 80% score.

- If the learner guesses 3 questions incorrectly, they will have to retry from the beginning. (how many times can they do this?)

**Game Components**

**Fish Container**

- The fish container will hold the swimming fish in the background.

- On click, the container will drop the fishing line.

- On click, the fishing line will go up with a fish.

- Then the question will appear after you catch a fish.

**Fishing Line**

- When the fishing container is clicked, a fishing line will drop down. To hook the fish you the learner will just click again.

**Fish**

- There will be a total of 16 fish swimming in the background.

- Randomize the questions on catch click.

**Questions**

- There will be 16 questions available for the learner to answer.

- The learner can attempt a maximum of 10 questions.

- they are random and trigger multiple events:

- if correct you get a correct toaster.

- if incorrect you get a try again toaster and one fish chance is marked red now.

**Win Condition**

- The learner must answer 8 questions correctly to win the game with an 80% score.

**Retry Condition**

- If the learner guesses 3 questions incorrectly, they will have to retry from the beginning.

**Chances**

- There are 3 fish chances that are displayed at the top.

- fish turn red as you get wrong answers.

- a toaster is also triggered with a wrong answer.

**Game Flow**

1. Learner clicks the fish container.

2. Fishing line drops down.

3. Learner clicks a fish.

4. The selected fish is caught(swap out hook for a fish) and the fishing line goes up.

5. This triggers a question screen to appear.

6. The learner answers the question.

7. If the answer is correct, the learner can attempt to catch another fish.

8. If the answer is incorrect, a counter is incremented.

9. If the counter reaches 3 incorrect answers, the learner must retry from the beginning.

10. If the learner catches 10 fish and answers 8 questions correctly, they win the game.

**Technical Requirements**

- The game should be built using HTML(lk-web-components), CSS, and JavaScript.

- Utilize event handling to trigger actions such as dropping the line, catching fish, and showing questions.

- Implement a random selection of questions for each playthrough.

- Use local storage or cookies to keep track of the learner's progress. Tally component, toaster, spa template, and modal dialog.

**User Interface**

- Create an intuitive and user-friendly interface for the game.

- Ensure that the learner receives feedback on correct/incorrect answers and progress towards winning or retrying.

**Testing**

- Thoroughly test the game to ensure all features work as expected.

- Test for edge cases, such as handling of incorrect answers and retries.

- A toaster appears after the 4th question is answered correctly.

- If you get 3 wrong a toaster appears saying you need to restart.

- At 8 correct answers you win the game and a congratulations toast appears.

**Deployment**

- Host the game on a web server for accessibility. https://w3.ibm.com/services/gbslearn/content/course/fish\_game/test

**Additional Notes**

- Adding visual and audio effects to enhance the gaming experience.

- Implement a scoring system with the tally and breadcrumbs to track the learner's progress and provide feedback.

- Adding notifications(toaster) at benchmarks in gameplay (half-way point).