Draft 1 Final Report - Programming: The Next Step 2024

Bruny Krijgsman – 14484757

SCENARIO

Purpose: Scenario that describes a primary school student's use of an online educational math game app.

Individual: Student A, a fourth-grade primary school student who wants to practice math in a fun way.

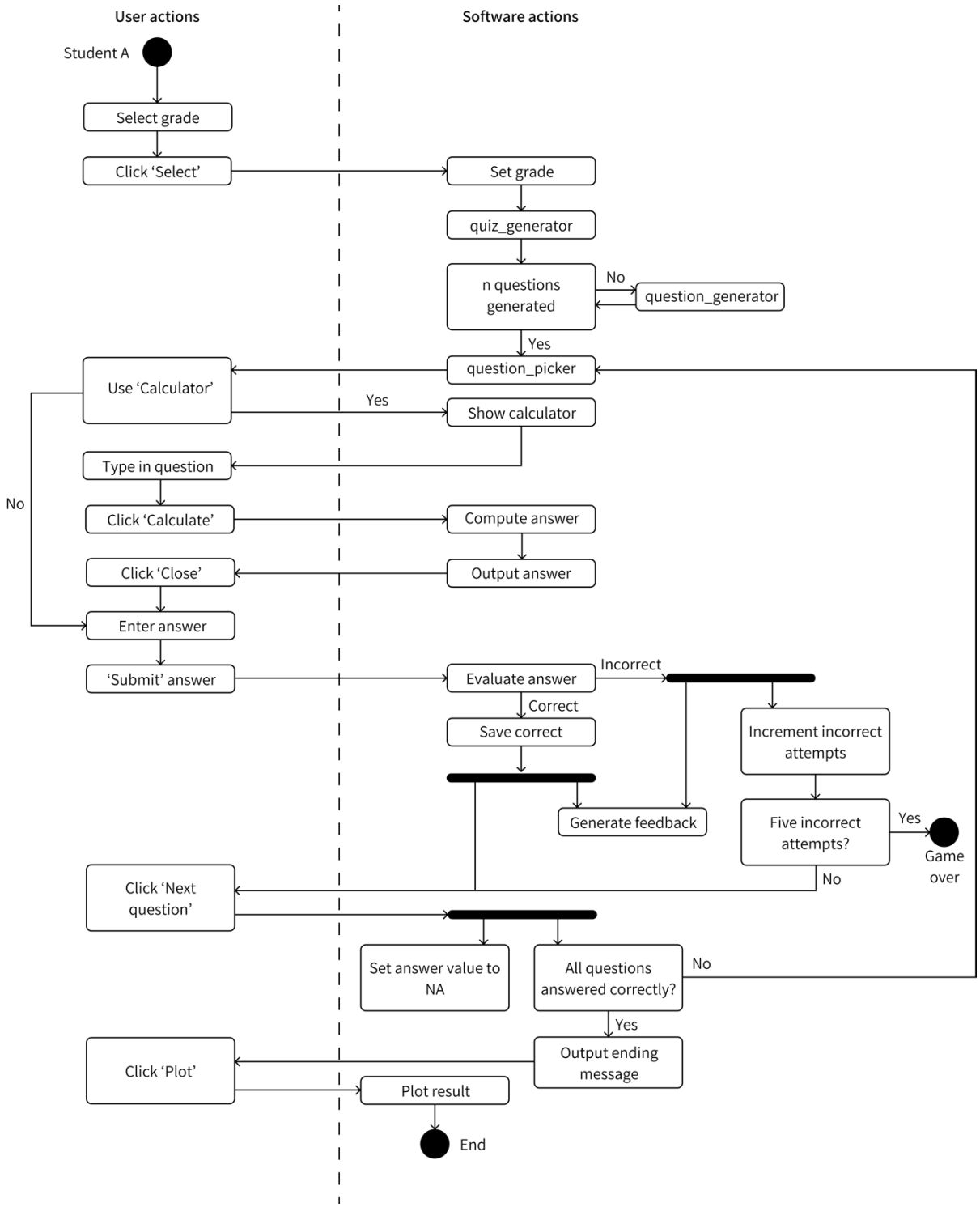
Equipment/software: Any computer with an internet browser. Also R/R Studio and the following packages need to be installed: “shinyjs”, “shiny”, “shinyWidgets”, “ggplot2”, and “mathQuest”.

Scenario:

1. Student A opens R studio and installs and loads the package mathQuest.
2. Student A fills in the ‘name’ and ‘question_count’ argument for the runMathQuest function and runs it.
3. Student A arrives on the landing page and sees the instructions, a selection input to select the corresponding grade student A is in together with a ‘Select’ and ‘Information’ action button, a calculator action button, a progress bar above a monster cartoon, and some other cartoons in the header.
4. Student A reads the instructions about the app, containing its purpose and how to use it/play the game.
5. Student A is asked to select the primary school grade he/she is in. Student A clicks on the select input and selects “Fourth grade”.
6. Student A confirms their choice by clicking the action button “Select”.
7. The app presents the first math question (number, operator, number), randomly generated, and adapted to the math level of fourth graders, in the middle of the screen with a text input to type down the numeric answer to the question and a “Submit” action button.

8. Student A computes the answer to the question, enters the correct answer in the text input, and clicks “Submit”.
9. Student A receives feedback on the given answer: “Well done! Your answer is correct”.
10. Student A clicks on the action button “Next question”.
11. The app randomly selects the next math question on a fourth-grade level and presents it in the middle of the screen with a text input to type down the numeric answer.
12. Student A enters an incorrect answer in the text input and clicks “Submit”.
13. Student A receives feedback on the given answer: “Incorrect. The correct answer is *correct answer*”.
14. Student A clicks on the action button “Next question”.
15. The app randomly selects the next math question on a fourth-grade level and presents it in the middle of the screen with a text input to type down the numeric answer.
16. Student A clicks on the calculator button, after which a calculator appears in a pop-up.
17. Student A types in the question, clicks “Calculate”, and the calculator gives the correct answer.
18. Student A clicks “Close” to close the calculator, types down the correct answer in the text input, and clicks on the action button “Submit”.
19. Student A receives feedback on the given answer: “Well done! Your answer is correct”.
20. Student A clicks on the action button “Next question”.
21. The app randomly selects the following questions until student A correctly answers all questions (including repeating the questions answered incorrectly).
22. Student A receives a message after answering all questions: “You completed the quiz! Click plot to see your results”.
23. Student A clicks on the action button “Plot”.
24. Student A sees two plots showing the percentage of all questions answered correctly on the first try and the percentage of questions answered correctly on the first try per category (addition, subtraction, multiplication, and division).
25. Student A closes the app by closing the internet browser.

FLOWCHART


User actions
Software actions
Student A
Select grade
Click 'Select'
Set grade
quiz_generator
n questions generated
No
question_generator
Yes
question_picker
Use 'Calculator'
Yes
Show calculator
Type in question
No
Click 'Calculate'
Compute answer
Click 'Close'
Output answer
Enter answer
'Submit' answer
Evaluate answer
Incorrect
Correct
Save correct
Increment incorrect attempts
Generate feedback
Five incorrect attempts?
Yes
Game over
No
Click 'Next question'
Set answer value to NA
All questions answered correctly?
No
Yes
Click 'Plot'
Output ending message
Plot result
End

HOW TO USE THE APP & EXAMPLE APPLICATION

Description

Math Quest is a fun and playful quiz app that allows elementary school children (fourth grade through eighth grade) to practice different parts of math, such as addition, subtraction, multiplication, and division. The app randomly generates math questions adapted to the selected school grade level. The application incorporates gamification and learning strategies to promote and make learning more fun for students and optimize their enjoyment to achieve the best results. For example, students are encouraged to answer all the questions correctly to 'defeat' the monster and all incorrectly answered questions will be repeated until correct. The app can be personalized by selecting the grade (math level), setting a name, and choosing the number of questions that they want to practice.

Running the app

Installation

To install the mathQuest package, run the following command in the console:

```
remotes::install_github("asarafoglou-ptns/Krijgsman-MathQuest")
```

How to run the app

The mathQuest app can be started by running the function runMathQuest().

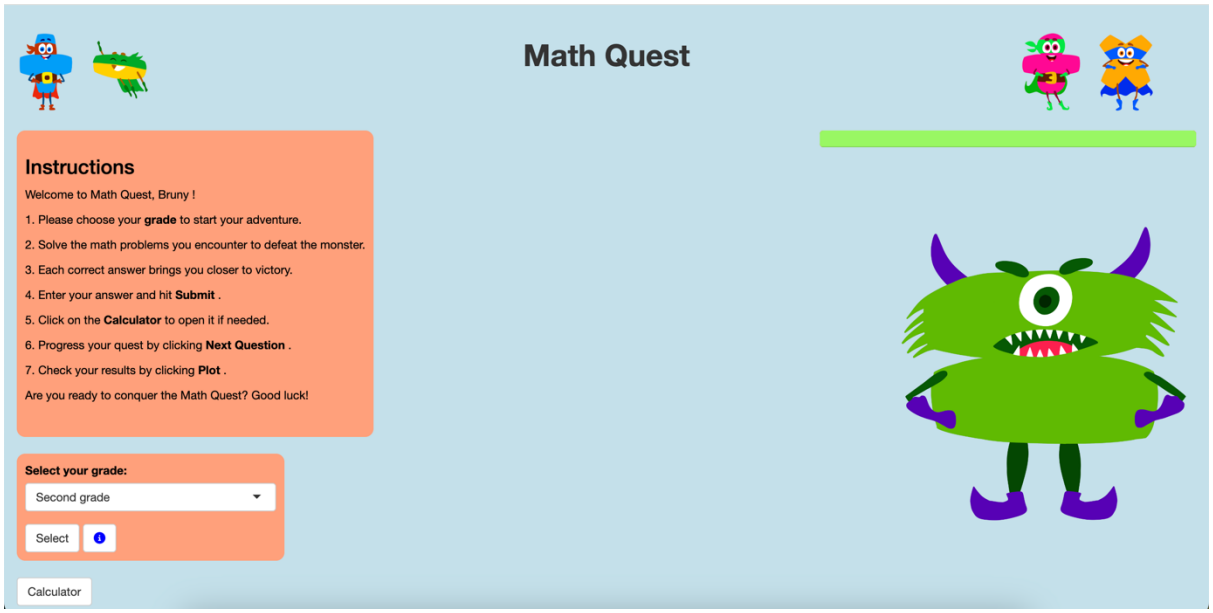
```
library(mathQuest)

mathQuest::runMathQuest(name = "Bruny", question_count = 15)
```

How to use the app

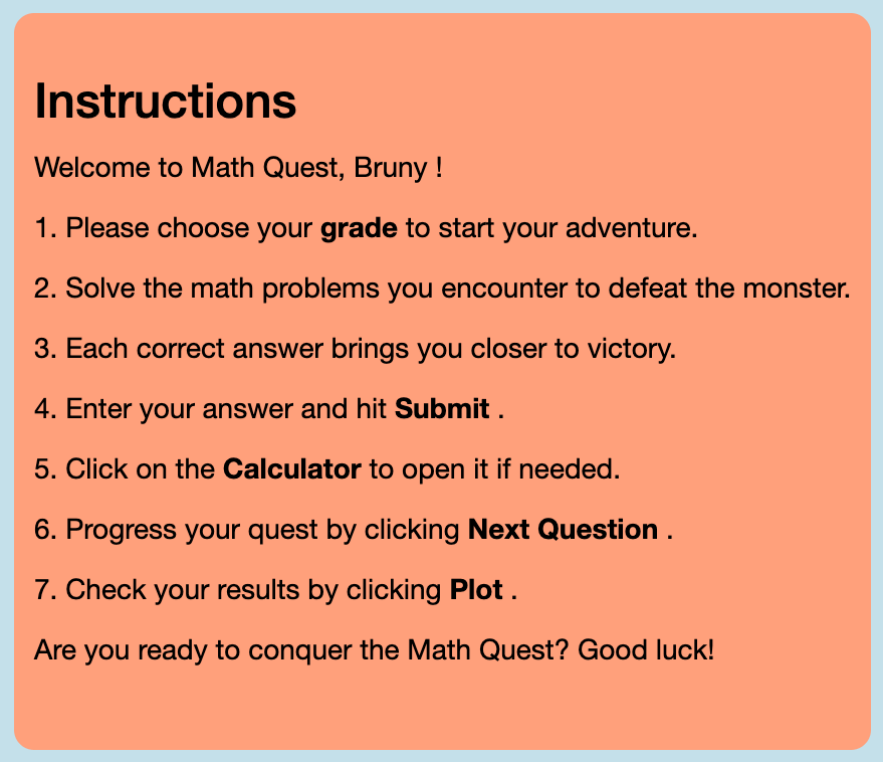
Example application

Landing page. After starting the app, the user arrives at the landing page and sees the instruction box, the box to select the grade they are in, a calculator button and a monster with a progress bar above their head. In the header, superhero cartoons are displayed which help the user defeat the monster during the quest.



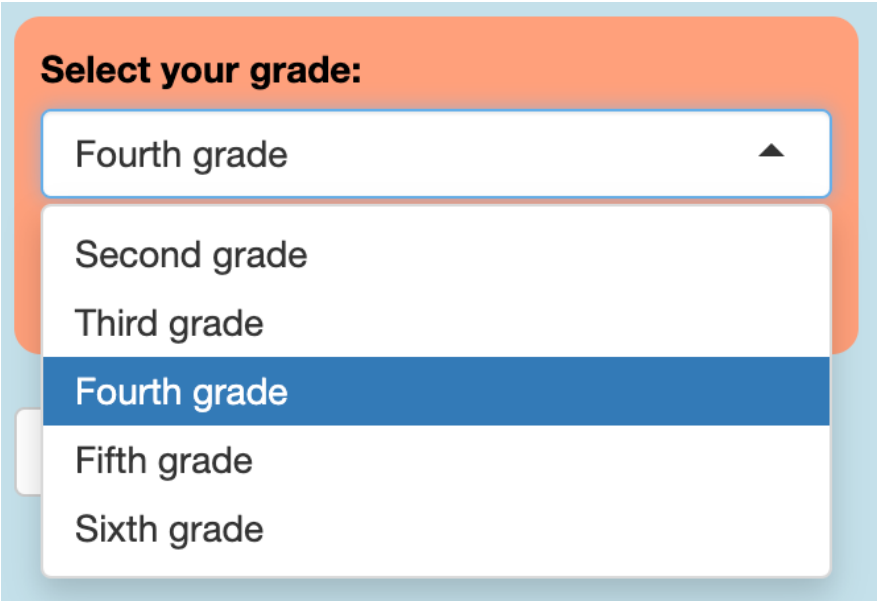
Landing page

Instructions. First the user reads the instructions of how to use the app/play the game shown in the orange text field.



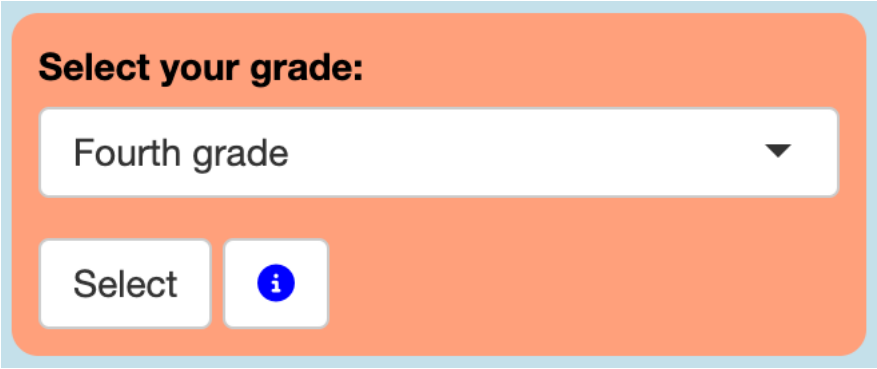
Instructions

Select grade. Then the user is asked to select the there grade. Under ‘Select your grade’, answer options are shown to the student to select the grade they are currently in. The available options go from second to sixth grade.



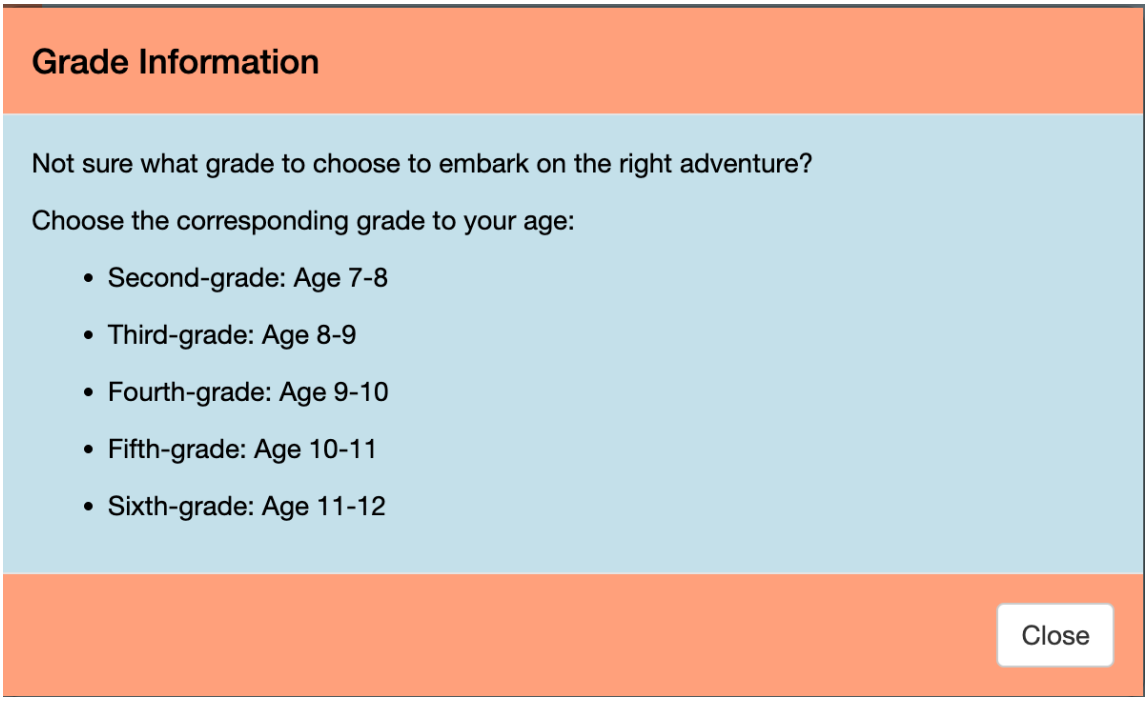
Select grade input

Select button. After choosing the grade, the user clicks the ‘Select’ button.



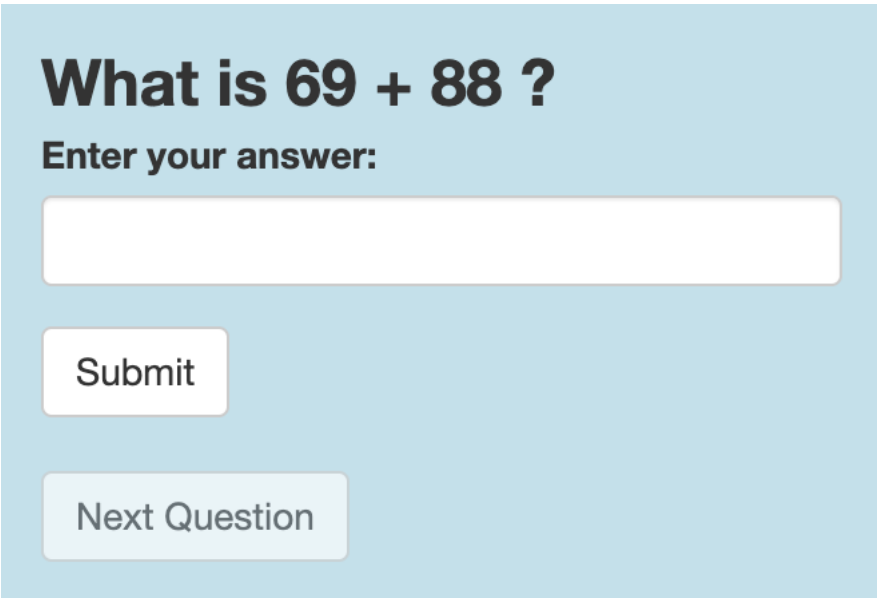
Select action button and information button

Grade information. If the user is not sure which grade to choose they can click on the 'i' icon after which the information modal will open. The pop-up concerns the grades along with the corresponding ages. After reading, the information modal can be closed by pressing 'Close'.



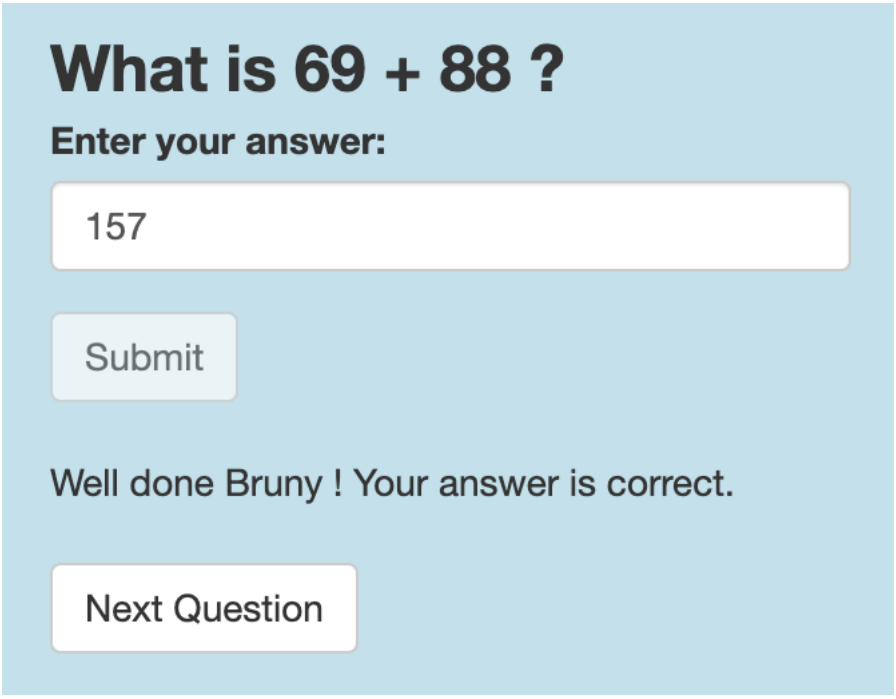
Grade information pop-up with 'Close' button

Math question. Once the user presses 'Select', the app generates the specified number of math questions at the selected math level. The questions are presented one at a time in the center of the screen. Below the question, the answer input is shown. Here the user can type down the answer and press the Submit button to check the answer. Correctly answered questions are saved for plotting their performance at the end while incorrectly answered questions are repeated randomly until the user has answered all the questions correctly.

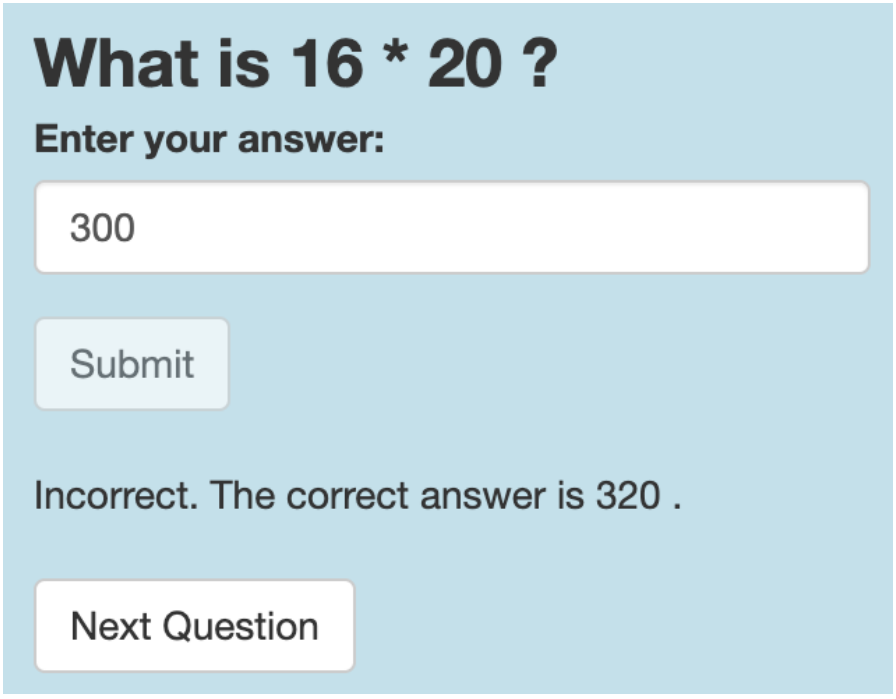


The math question, the answer input and the 'Submit' button

Feedback. After the user clicks the 'Submit' button, the application generates feedback on the given answer to the arithmetic to indicate whether the answer is correct or incorrect.



Feedback for correctly answered questions

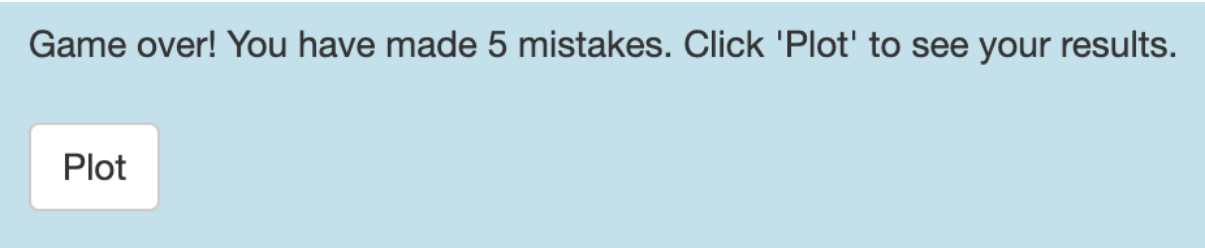


Feedback for incorrectly answered questions with the correct answer

Next question. After the feedback is generated, the 'Next Question' button becomes available on which the user needs to click to show the next question. The 'Next Question' button becomes available again only after using the 'Submit' button and is blocked after pressing it once.

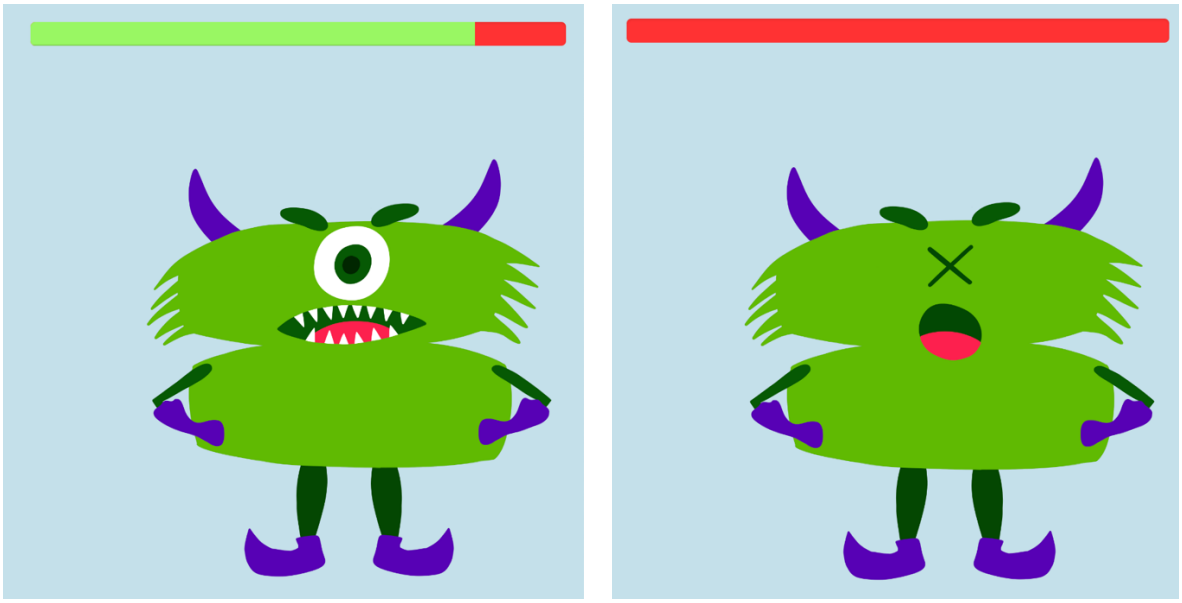
Submit. This button is only available when the answer to a question has not yet been submitted. After submitting the question, this button is disabled and is only available again after pressing the 'Next Question' button.

Game over. Since incorrectly answered questions are repeated randomly until all questions are answered correctly, a maximum of 5 errors is allowed. When the user makes 5 errors the game is ended and the user is shown that it is 'Game over' and they can press plot to see their achievement.



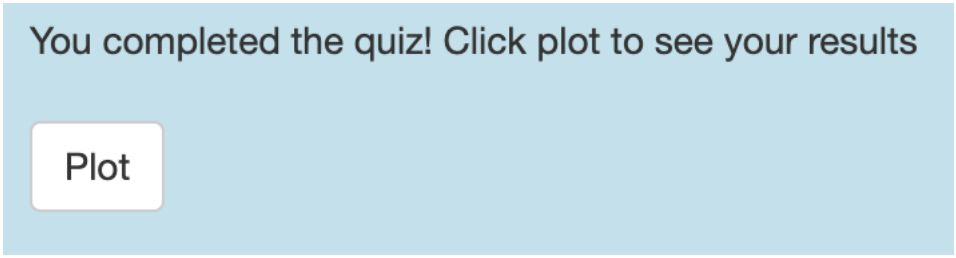
Message that shows 'Game over' after 5 mistakes

Progress bar. The progress bar is actually a health bar that indicates the life of the "monster". At first the bar is completely green and with each correctly answered question the monster's vitality decreases and the bar changes to a red color. When all questions are answered correctly the bar is completely red and the monster is defeated, i.e., the user has won.



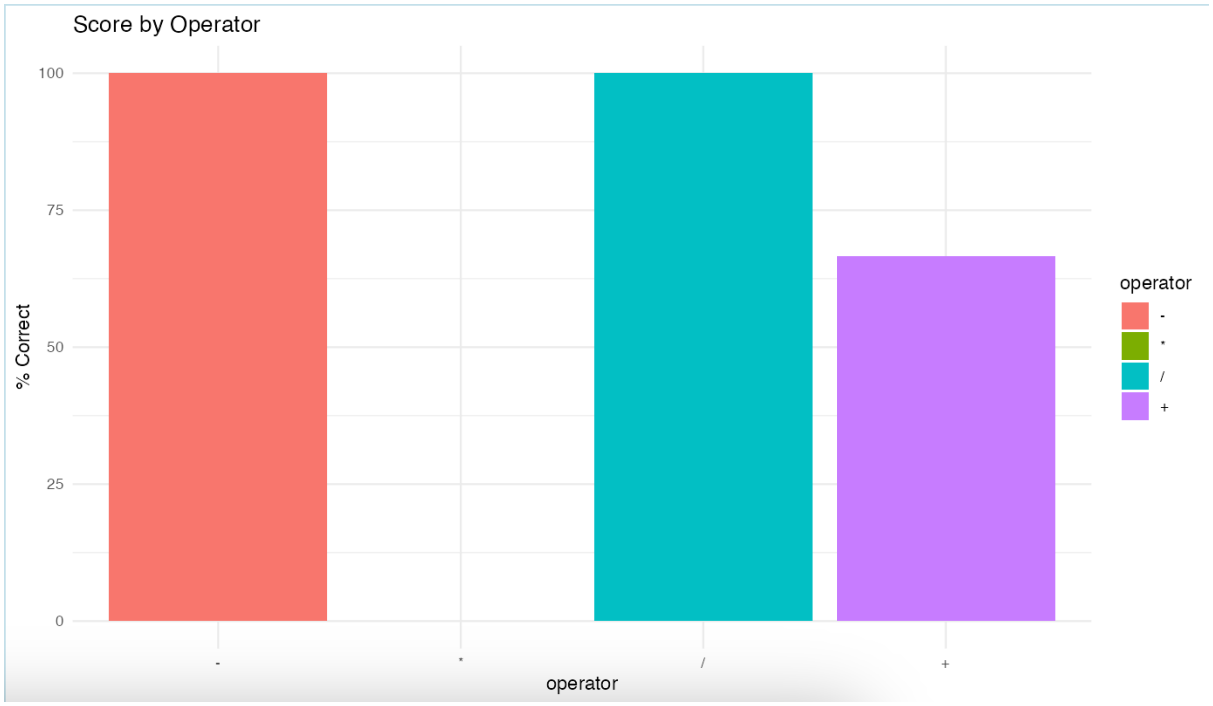
Progress bar indicating the health/life of the monster (left). Progress bar indicating the monster is defeated (right)

Plot. After answering all the questions, the user can press 'Plot' to see his/her results.



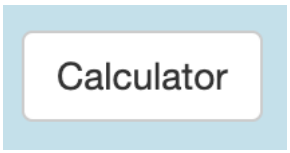
Message and plot button after answering all questions

Showing the plots. After pressing the button “Plot,” the application generates a plot of the users performance. For each operator, the percentage of correctly answered questions on the first attempt is displayed. The user can thus get a good overview of which parts could use additional practice.



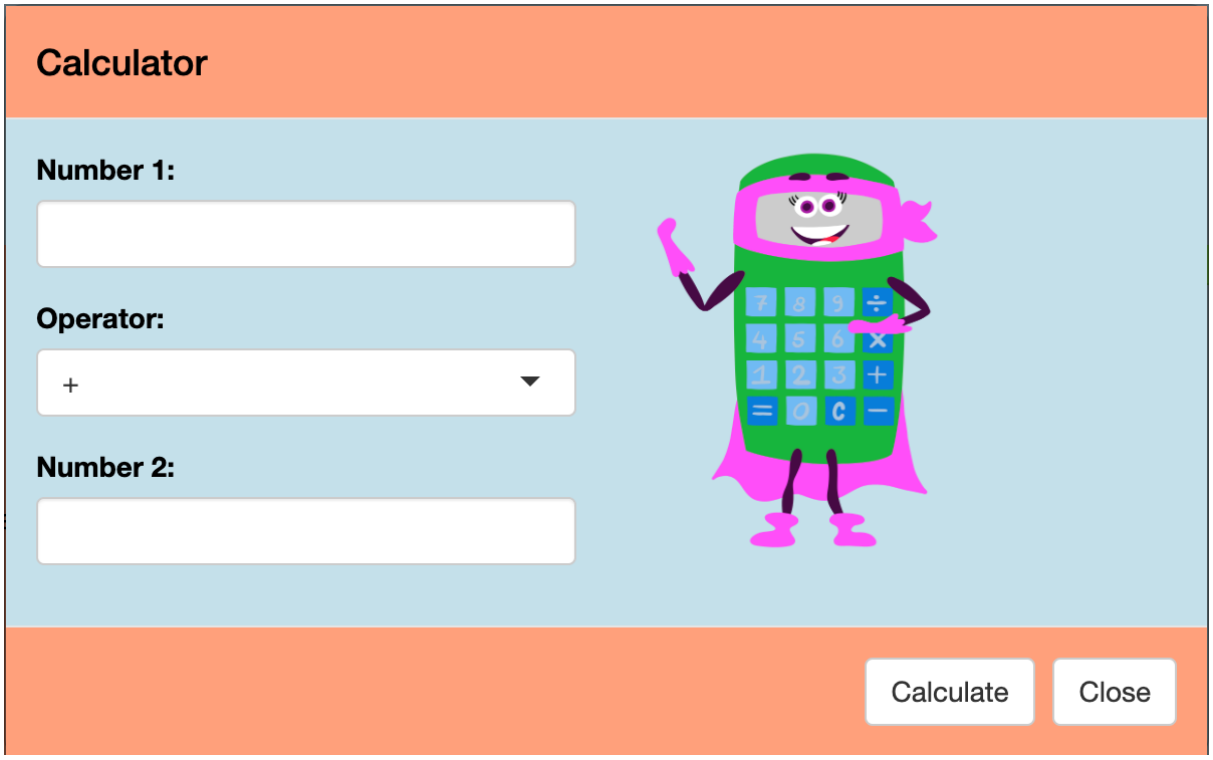
Showing the performance per operator in percentages

Calculator. Although the math questions have been developed to promote and improve mental arithmetic, the user has the option of using the built-in calculator. The calculator appears by clicking on ‘Calculator’.



‘Calculator’ button

After pressing this button, a calculator modal appears. In this modal, the user specifies both numbers and the corresponding operator. Then the user clicks 'Calculate', and the application shows the correct answer after which the user can press 'Close' to close the calculator modal again.



Calculator modal with two number inputs, an operator selection, and a 'Calculate' and 'Close' button