Readme File

2024-04-01

Math Quest: Rogue Minds

Objectives

Enhance Mental Math Abilities: The primary objective of MathQuest - Rogue Minds is to provide elementary schoolers with an engaging gaming experience that encourages quick and accurate mental calculations. By incorporating math problems into a thrilling gameplay environment, the project aims to stimulate and strengthen mental arithmetic skills.

Teach and Reinforce Arithmetic Concepts: Through the gameplay mechanics of MathQuest - Rogue Minds, fundamental arithmetic concepts such as addition, subtraction, multiplication, and division are taught and reinforced. Each game event presents players with math problems of varying difficulty, ensuring a solid foundation in basic math skills.

Shift Reliance from Digital Devices: In an age dominated by digital tools, MathQuest - Rogue Minds seeks to redirect students' reliance on calculators and computational software towards their own mental capabilities. By providing a captivating alternative that emphasizes mental math, the project encourages a deeper engagement with arithmetic concepts.

Introduce Elements of Strategy and Unpredictability: Drawing inspiration from the roguelike genre, MathQuest - Rogue Minds introduces elements of strategy, unpredictability, and risk. Players navigate procedurally generated dungeons, where each decision and calculation impacts their progress. By infusing excitement and challenge into the learning process, the project fosters a positive attitude towards mental math.

Improve Application in Academic and Real-Life Scenarios: Beyond the gaming environment, MathQuest - Rogue Minds aims to facilitate the practical application of arithmetic skills. Boss encounters within the game mirror real-life scenarios, emphasizing the relevance and importance of mental math in everyday situations. Through repeated exposure and practice, players develop the confidence to apply their skills across various academic and real-life contexts.

Key Features

Math-Based Combat: Engage in math-based combat with enemies, where each encounter presents a mathematical challenge that must be solved to progress.

Educational Integration: MathQuest - Rogue Minds aligns with educational curriculum goals and collaborates with educators to ensure effective integration of math concepts into gameplay.

Engaging Visuals and Audio: Immersive visuals and audio enhance the gaming experience, making MathQuest - Rogue Minds both educational and entertaining.

Usage

Prerequisites

- Java Development Kit (JDK) installed (version 8 or higher)

- Apache Maven installed (optional but recommended for building and managing dependencies)

Running the Application

Command Line Interface To run the application from the command line, navigate to the project directory and execute the following command:

```
java -jar MathQuest.jar [options]
```

Replace [options] with any necessary command-line options or arguments.

IDE (Integrated Development Environment) If you prefer to run the application from an IDE such as IntelliJ IDEA or Eclipse, you can import the project into your IDE and run the Main class directly.

Building from Source Using Apache Maven

If you have Apache Maven installed, you can build the application from source using the following command:

```
mvn clean package
```

This command will compile the source code, run tests, and package the application into a JAR file named MathQuest.jar located in the target directory.

Without Apache Maven

If you do not have Apache Maven installed, you can compile the source code manually using the following commands:

```
javac -d bin src/main/java/com/example/Main.java
```

This command will compile the Main.java file and place the compiled class files in the bin directory.

To package the application into a JAR file, navigate to the bin directory and use the following command:

```
jar cvf MathQuest.jar com/example/*.class
```

Running Tests

Unit Tests

To run unit tests, execute the following command:

```
mvn test
```

This command will run all unit tests located in the src/test/java directory.

Integration Tests

Integration tests are located in the src/test/java directory and can be executed along with unit tests using the mvn test command.

Contributing

If you'd like to contribute to the project, please follow the guidelines outlined in CONTRIBUTING.md.

License

This project is licensed under the MIT License - see the LICENSE file for details.

Authors

- Gurshaan Gill
- Justin Edwin Lee
- Justin Lai
- Darren Chen
- Ryan Gary Tse