Zero Gravity Lesson App – Project Overview

Overview

This is a Java Swing application designed to teach children about zero gravity in a fun, interactive way. The app features an animated/dynamic avatar, a conversational lesson, a short quiz, dynamic feedback with badges, and progress tracking. The goal is to create a playful, educational environment simulating an astronaut training lesson.

Github Link : https://github.com/GoyalAnhad/Zero-Gravity

Language and Tools Used:

1. **Java SE & Java Swing:** For GUI, event handling, and custom drawing.
2. **Java AWT:** For graphics, custom cursors, and low-level UI customisation.
3. **Wikipedia API: For chat-based, real-world fact fetching via HTTP REST API.**
4. **Local file storage: "progress.txt" for tracking quiz attempts and scores.**

Key Components:

1. **Dynamic Animated Background:** Displays a lively, animated “outer space” scene with twinkling stars and comets behind all screens, increasing engagement.
2. **Welcome/Story Panel:** Presents an interactive introduction via dialogue bubbles, simulating a mission control briefing and explaining the learning journey. Allows the user to step through a story before starting the lesson.
3. **Lesson Panel (Educational Content):** Provides a visually appealing, story-driven explanation about zero gravity (microgravity), with images and fun facts. Uses friendly, accessible language and integrates the avatar for immersive learning.

5. **Chat with Avatar (Dialogue System):** Lets the user type questions and get answers from the avatar in a chat-like interface. Can answer both pre-programmed (“Why study zero gravity?”) and open-ended questions (using the Wikipedia API). Provides basic AI simulation–fetches summaries from Wikipedia for questions it doesn’t know, simulating a smart teaching assistant.
6. **Quiz Panel: Presents** a multiple-choice quiz about the lesson (3 questions, easily extendable). Handles user interaction, disables options after an answer, and gives instant feedback (“Correct!” or “Oops!”). Keeps score internally for results and badges.
7. **Dynamic Results & Badges:** At the end of the quiz, shows a custom message based on the user’s score. Awards a gold, silver, bronze, or fail badge image, giving visible reward for learning progress. Motivates users to replay for a better score.

8. **Progress Tracking:** Saves lesson completion and quiz scores to a local file (progress.txt) for basic progress tracking. Allows the user (or a teacher) to see if the lesson was completed and how well they did.

9. **Reusable UI Elements:** Provides utility functions (like a stylish back button) to keep the interface consistent and user-friendly.

Execution

1. Place required image files in the working directory.

2. Compile and run the Java application

   (For Compile) -   javac ZeroGravityLessonApp.java

   (For Run)   -   java ZeroGravityLessonApp

Unsolved Challenges

1. Webcam Integration

2. Speech-to-Speech (Voice) Interaction

3. An animated or dynamic avatar guiding the lesson

References and Inspiration

1. **Java Documentation**: docs.oracle.com/javase/8/docs/api/

2. **Wikipedia API Docs**: en.wikipedia.org/api/rest_v1/

3. ChatGPT: https://chatgpt.com

4. **Images**: Public domain or open-source icons, user-provided avatars and badges.

Execution

- **Run via Maven: (**mvn spring-boot:run) The application will start on port **8080**.

- **Testing the Endpoints:**
  - **Parsing a PDF:**
  - (curl -X POST -F "file=@/path/to/your.pdf" http://localhost:8080/api/parse)
  - **Generating a Password:**
  - (curl -X POST “http://localhost:8080/api/createPassword?firstname=John&dateOfBirth=1990-01-01&accountType=Savings")

- **Running Errors:**
  Despite the code being logically sound, the application currently experiences some runtime errors when calling the LLM API. These errors are under investigation, and additional error handling or retries might be necessary in future iterations.

- **LLM API Response Handling:**
  The integration with the LLM API may require further improvements to gracefully handle different error conditions.

References

- **Oracle Java Documentation:**
  https://docs.oracle.com/javase/8/docs/api/javax/net/ssl/HttpsURLConnection.html

- **Spring Boot Documentation:**
  https://docs.spring.io/spring-boot/docs/current/reference/html/

- **Apache PDFBox Documentation:**
  https://pdfbox.apache.org/

- **Jackson Documentation:**
  https://github.com/FasterXML/jackson

Acknowledgments

- **Assistance:**
  This project was developed with valuable guidance and support from both ChatGPT and Claude.

- **Note:**
  The current version of the project contains some running errors that are being worked on.