1. Project Overview

Java | Encog 3.4 | Tunnel Game Autopilot

This project implements a neural network-based autopilot to control a plane in a tunnel game. The goal is for the neural network to navigate the plane through an endlessly scrolling tunnel without touching the edges, using only the Y-axis for movement. The network is trained using the Encog 3.4 library.

2. Training Data Extraction

- Training data is collected by manually controlling the plane and recording the game state and actions taken.
- The game state is represented as a grid (30x20), with each cell indicating the presence or absence of an obstacle.
- Only the columns ahead of the plane are used as input features to reduce dimensionality.
- Data is saved in `training_data.csv` for use in training the neural network.

3. Neural Network Topology & Configuration

- Library: **Encog 3.4**
- Input Layer: Number of nodes = number of sampled grid cells ahead of the plane
- Hidden Layers: 1 or more (configurable)
- Output Layer: 1 node (move up or down)
- Activation Function: Non-linear (e.g., sigmoid or tanh)
- Training Hyperparameters: Learning rate, epochs, etc. (see `TrainEncogNN.java`)
- Model is saved as `trained_network.eg`

4. Training Process & Hyperparameters

- Training is performed using the data in `training_data.csv`.
- Network is trained in under 1 minute.
- Hyperparameters (learning rate, epochs) are set for fast convergence and good generalization.
- Model is validated by testing in the game environment.

5. Integration with Game

- The trained neural network is loaded at runtime and used to control the plane automatically.
- Game logic and neural network code are integrated in the `ie.atu.sw` package.
- To run the autopilot, execute the main class: `ie.atu.sw.Runner`

6. Switching Between Autopilot and Manual Mode

The game can be run in either **autopilot** or **manual** mode. This is controlled by a boolean flag in the game code (e.g., `autopilot` variable).
If `autopilot` is set to `false`, the game will allow manual control by the player.
If `autopilot` is set to `true`, the neural network will control the plane automatically.

To switch modes, open the relevant game file (e.g., `GameWindow.java` or `Runner.java`) and set the `autopilot` variable as desired before running the game.

7. Testing & Results

- The autopilot consistently survives for more than 30 seconds in the tunnel game.
- Training time is less than 1 minute.
- Performance can be further improved by collecting more training data or tuning hyperparameters.

8. How to Run

1. Ensure `encog-core-3.4.jar` is in the project directory.
2. Build the project and create `ai.jar` (see instructions below).
3. Run the autopilot using:
   `java -cp ./ai.jar;encog-core-3.4.jar ie.atu.sw.Runner`

9. Extras

- Game images and additional resources are included in the `images/` directory.
- Trained model and training data are included for reproducibility.

10. Packaging as ai.jar (Eclipse Instructions)

1. Right-click your project in Eclipse → **Export** → **Java** → **JAR file**.
2. Select your source files and output location (name it `ai.jar`).
3. Set the main class to `ie.atu.sw.Runner` if prompted.
4. Finish export. Your `ai.jar` is ready to run as described above.

11. ai.jar File Details

- **ai.jar** is the main executable JAR file for this project.
- It contains all compiled code and is ready to run using the command:
  `java -cp ai.jar;encog-core-3.4.jar ie.atu.sw.Runner`
- Make sure **encog-core-3.4.jar** and the **resources/** folder are in the same directory as ai.jar.
- This JAR was created using Eclipse's Export → JAR feature, with `ie.atu.sw.Runner` as the main class.

12. Final Submission Checklist

- **ai.jar** (main executable JAR file)
- **src/** (all Java source code)
- **resources/** (trained model, training data, etc.)
- **images/** (game images, if required)
- **encog-core-3.4.jar** (Encog library JAR)
- **README.html** (or PDF, as required)
- **No IDE files** (like .project, .classpath, .settings/ etc.)
- Tested on a different computer (recommended)

Note: Only include the files/folders listed above in your final ZIP submission.