

Traike

Niclas Heun and Trevor J. Yao

November 2023

1 General Idea & Overview

Originally introduced in 1976, the game Snake has become a classic, capturing the imagination of generations. In Snake, players navigate a growing line (i.e. a snake) using arrow keys, avoiding collisions with walls or the snake itself. Each “piece of food” consumed lengthens the snake, adding to the challenge.

In Traike (Train-Snake), we reimagine this iconic game on railway tracks. Players control a train, our modern ‘snake’, using the right and left arrow keys to adjust the train’s path at track switches. The train’s speed is automatically managed by the program. Collisions with track exits, the equivalent of Snake’s walls, result in game over.

The objective is to collect ‘food’, depicted as stationary trains on the tracks. When the player’s train aligns with another, the stationary train accelerates to join the moving one, extending the length of the ‘snake’.

2 Milestones

2.1 Game Framework with one working train

Our initial focus is to enable a single train’s smooth movement across a grid at a constant speed, responsive to player inputs. The system will track the train’s precise location, ensuring switch activations are executed safely and in a timely manner to prevent derailments. Too late keystrokes will be ignored.

2.2 Adding a second train

With one train operational, our next step is to introduce a stationary ‘food’ train onto the tracks. As the player’s train approaches, the stationary train will accelerate, eventually aligning and following at a safe, variable distance, mimicking the snake’s growth in the original game.

2.3 Reducing the distance

Our third milestone aims to reduce the distance between the connected trains, enhancing the game’s challenge. While the optimal distance is yet to be determined, our goal is to achieve a gap of just a few centimetres.

2.4 Adding a third train

The final step in our project is to add a third train to the game, further increasing the complexity and replicating the growing challenge of the original Snake game.

3 Technical challenges and solutions

3.1 Maintaining Consistent Distance Between Trains (Cohorting)

The primary challenge lies in ensuring that multiple trains follow each other at a consistent distance. This task is challenging due to the need for high precision in both distance measurement and speed control of the trains.

Proposed Solution

- **Distance Measurement:** We will try to shorten the sensor loop as much as possible and focus on an effective sensor attribution to know exactly which sensor activation is caused by which train.
- **Speed Control:** Currently, our model only support three different velocities per train. We will need to calibrate way more velocities to allow for a greater range of speed adjustments, ensuring smooth and consistent following. This may also involve estimating velocities on the fly.

3.2 Positioning - Accurate Switching

Accurately predicting train positions on the track is crucial for timely and safe switch operations, particularly given the game's dynamic nature. We need to prevent the user from switching the track too late (and causing a derailment) but need to allow "late-enough" switches for the game mechanism.

Proposed Solution

Develop a more refined speed model for the trains, enabling precise predictions of train locations (outside of sensors). This will allow us to determine the most appropriate timeout for switch operations in response to player inputs.

3.3 Integrating New Trains Without Collisions

Stationary trains have to be accelerated in advance, to join the moving formation without causing collisions. This introduces the issue of precisely predicting when we need to speed up the train. However, the user might decide to reroute the train last minute, before 'catching' the stationary train.

Proposed Solution

Based on precise predictions and collision prediction, we will speed up the stationary train ahead of time. Once we detect that the user rerouted the snake, we will decelerate and stop the unjoined train again. However, during this speed up and slowing down, the soon to be caught train, should not collide with the third still stationary train, which adds further complexity.

One milestone if this becomes too complex is to stop the snake at the next food train before restarting the snake.

4 Other Features

The game will self initialise by learning locations of trains on the track at start-up before allowing user-input, and then pick a random train as the "snake".