Computer Controlled Railroad Simulator

CSCI 460-Senior Capstone Project
Spring 2012
Kayla Pope

Project Description
Definition and Requirements

- Design a track definition protocol
  - Users define their own track layouts.
- Develop a graphical interface
  - Current status of CCR
- The simulator supports multiple trains.
- Simulate error conditions

Definition and Requirements

- Modify the 2010 CCR Train Operating System
  - Real mode or Simulator mode
- Support the protocol for decoders (stationary and mobile) and Auxiliary Input Units (AIU) as specified by the National Model Railroad Association (NMRA).
- Develop API functions that match the protocol for the existing CCR functions.
Solution

- Two major parts
  - Track designer
  - Simulator
- Diagram of data structures
  - trackStructure(trackPt)
  - Turnouts/sensors
  - Train

Demonstration

- Create new track layout
- Delete Segments
- Include turnouts
Demonstration

- Add two trains
- Run the simulator
- Increase/decrease speed
- Change turnouts

Simulator

- Trains follow color of the track
  - “Compass”
  - Next direction restriction
  - “Blind”—don’t know what’s coming next

- Not a perfect algorithm
  - User must define plausible track

Curve is too sharp for train to follow.
myTrains[0]

Point loc
\begin{tabular}{|c|c|}
\hline
int X & int Y \\
\hline
\end{tabular}

int direction

char state

int firstcar

string color

myTrains[4]

1. myTrack[val1] is drawn

2. Box is drawn to “cut” track

3. myTrack[val2] is drawn
Exceptions

- Changing the state of a turnout—in certain circumstances

- Depending on the order in which segments are drawn.

- Only rarely messes up the drawing, but still unreliable.

Other Exceptions

- Not compatible with actual train
- Moving algorithm—train gets “stuck”
- Train with multiple cars (circular queue)
- Cancelling when in edit mode
Methodology

1. Several meetings
2. Design one small piece
   i. Keep in mind the overall goal
3. Implement small piece to prove conceptual understanding
4. Modify as necessary
   i. Make sure any modifications allow for success of overall goal
5. Repeat steps 2-4

- Curves/Segments
  - Bezier
  - Regular (4 points)
  - Regular (3 points)
    - All sensors when initialized
Learning and Development Process

Strategies

- Understand the goal
- Create data structures that can accomplish the goal
- Draw pictures
- Implement (small parts at a time)
- Modify as necessary

- Discuss with ANYONE who will listen
  - Helps to get another perspective
  - EVEN non-CS people
- Advice from classmates
Knowledge

- Event Programming
  - C# language
  - Event Handlers
  - Inter-Form communication
  - Graphics
- Programming Languages
  - Parameter Passing
- Machine Language
  - Racing conditions
- Data Structures
  - Classes
  - Public/private methods
  - Accessors/Modifiers
  - Queues
- Operating Systems
  - Threading
  - Scheduling
- Others
  - Read/Write Files

Extensions

- Plug in to actual train
- Improve algorithm to move train
- User interface
  - Get rid of so many Message Boxes!
- Restrict aspects—do not rely on user to create plausible track
- More efficient code
  - Custom events rather than methods between forms
Advice

- FIND MOTIVATION AND RUN WITH IT… EARLY
  - Progress helped me find motivation
- Take walk-throughs seriously
  - Be proactive
  - Know what you want to get out of it
- Know the goal
- Know your data structures
- Leave time to test and modify

Question & Answer
Thank you!