Project Plan
RailBuilder: The Great Race to Promontory
The Capstone Experience

Team Union Pacific
Trever Daniels
Declan McClintock
Jacob Young
Kyle Bush
Zach Brenz

Department of Computer Science and Engineering
Michigan State University
Fall 2017
Functional Specifications

• Simulate real-world environments
  ▪ Realistic elevation as well as tree, building, and road placement
• Simulate building railroads between known locations
• Create game that showcases this technology
Design Specifications

• Terrain is generated based on USGS map data between two user-defined locations in the US
• Designed as a game (like Rollercoaster Tycoon)
• User attempts to build railroad between points given constraints
  ▪ budget, time, land class, etc.
• “Levels” in the game vary by difficulty of building railroad
  ▪ Easy: Lansing -> Detroit
  ▪ Difficult: Sacramento -> Omaha
Screen Mockup: Main Menu
Screen Mockup: Map Creator
Screen Mockup: Map Editor

Map Editor

New Map

3 Maps Created
Screen Mockup: Terrain Generation
Technical Specifications

• Unity 3D
• C#
• USGS land class and elevation data
System Architecture

- **Unity 3D**
  - Terrain Generating Application
  - Save and Load File Generator
  - Asset Placement Algorithm
  - 3D Terrain Height Generator

- **Rail Builder Game**
  - Win State: Connect Two Rail Stations
  - Place Track On The Terrain
  - Show Off All The Technology In The Base Application

- **Manage Budget**

- **Save Load**
  - Local Storage File

- **ZIP File Interpreter**

- **Terrain Data**

---

The Capstone Experience

Team Union Pacific Project Plan
System Components

• Hardware Platforms
  ▪ Primary: Windows PC
  ▪ Secondary: Android and iOS platforms

• Software Platforms / Technologies
  ▪ Unity
  ▪ Visual Studio
Testing

• Visual confirmation of our system within Unity and within the file system
• Playtesting our user interface and game with our client
Risks

• Retrieving/Processing USGS Terrain Data
  ▪ Terrain data difficult to understand
  ▪ Mitigation: Researching USGS, seeking advice from client

• Creating Terrain Based Off Elevation Data
  ▪ Depending on how data is formatted, could be tricky
  ▪ Mitigation: Generating terrain with random height values

• Texturing Terrain Based Off Land Class Data
  ▪ Need to map the different land classes to the correct 3D models
  ▪ Mitigation: Using basic colors to classify parts of terrain

• Algorithmically Place Environment Assets
  ▪ Accurately place trees, water, roads, etc.
  ▪ Mitigation: Placing assets randomly based on terrain and asset height

• Create An Enjoyable User Experience
  ▪ Major problem in game development
  ▪ Mitigation: Playtesting with client
Questions?