ECMM427
Group Development Project

CA2 Demonstration

LTN Project

Demonstration Requirement

- 20-minute demonstration of the prototype system covering:
  - the current feature set of the prototype
  - a brief introduction into the implementation (e.g., structure of your software repository)
  - a discussion of the operational setup (e.g., how is the current prototype hosted)
  - presentation of the planned work for the rest of term 2.
- Overall, the presentation should allow the client to understand the current state of the client as well as get a first impression of the skills and resources required to maintain, extend, and operate your prototype.

Presentation Structure

1. Current Features
2. Implementation
3. Demonstration
4. Future features and components

Current Feature Set of the Prototype

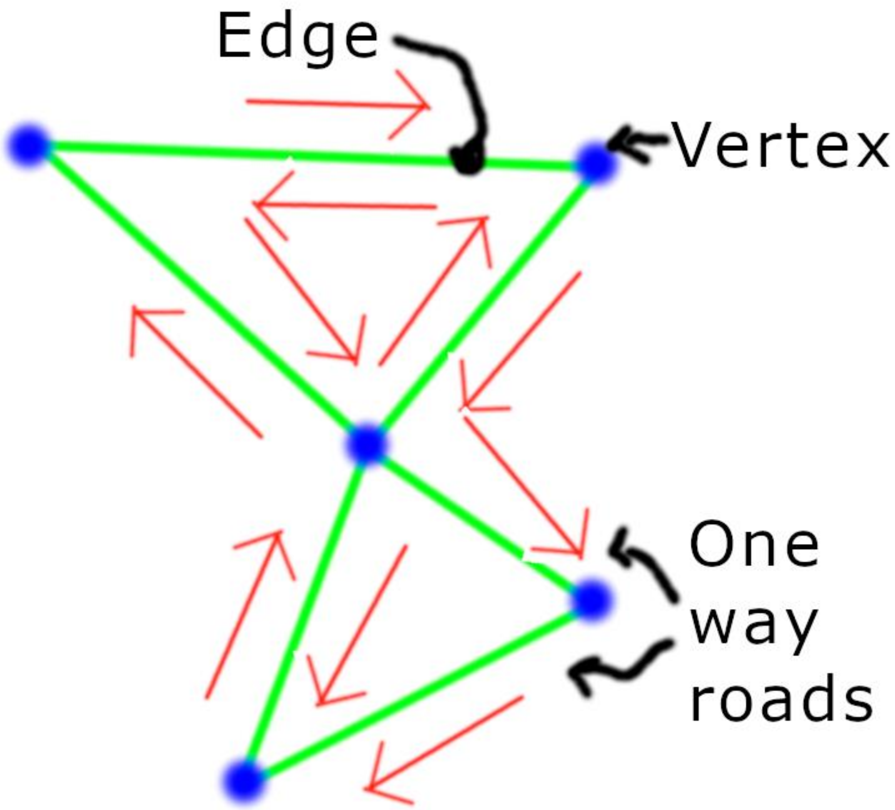
1. Traffic Simulation
2. Visual Representation of the ongoing simulation
3. Measuring key variables of the simulation
4. Result presentation

Traffic Simulation components

1. Graph based map
   - Dynamic pavement generation
   - Barriers
2. Agents
   - Dynamic agent velocity
   - Agent collision
   - Braking distance-based separation
3. Agent Manager
   - Stochastic Spawning of Agents
   - Stochastic Destination determination
4. Agent Pathfinding

Graph Based Map

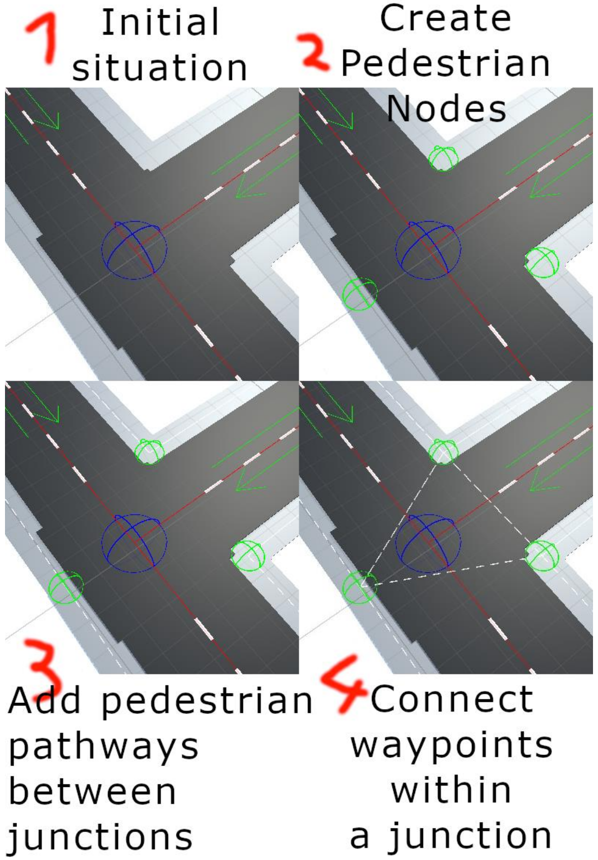
- Map consists of Waypoints (representing junctions) and directed Edges (representing roads)
- Movement along edge is only possible along the direction of the edge
- Nodes are placed at a location of natural junctions and/or dead ends.



- Requirements for prototype road arrangement:
  - one T junction (two roads meet at a right angle)
  - one cross junction (two roads intersect each other)

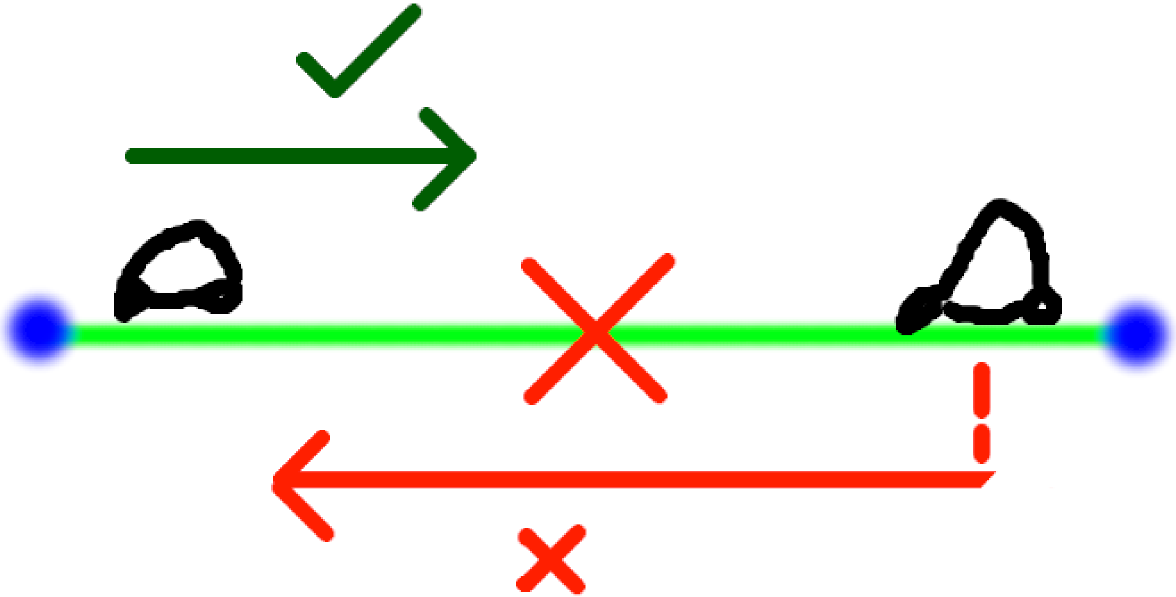
Dynamic Pavement Generation

- Pavement for pedestrians is generated at runtime
- Nodes placed in equal distance around junctions and connected between them in accordance with the roads


1
Initial situation
2
Create Pedestrian Nodes
3
Add pedestrian pathways between junctions
4
Connect waypoints within a junction

Barriers

- Edges may be inaccessible to specific form of traveler.
- Barriers may be placed on edge: to prevent travelers passing through the block.

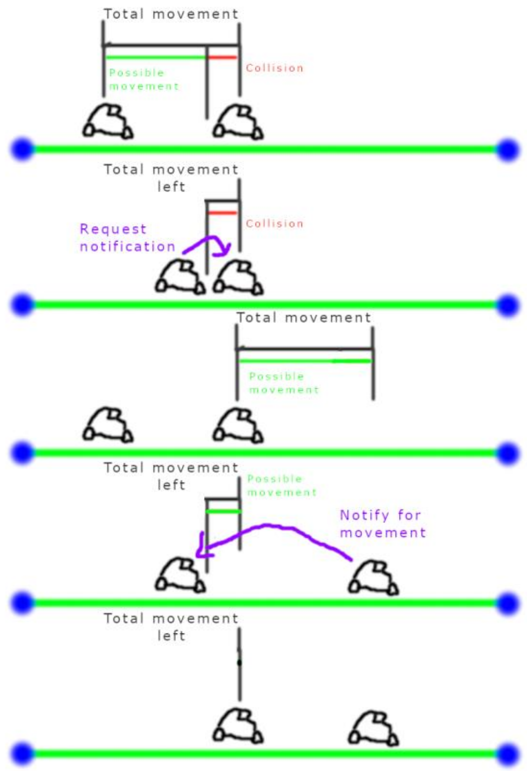


Agents

- Representing individual travellers
- Position based on edge endpoints and distance along it
- Variable velocity component
    - Maximum velocity
    - Acceleration rate
    - Deacceleration rate

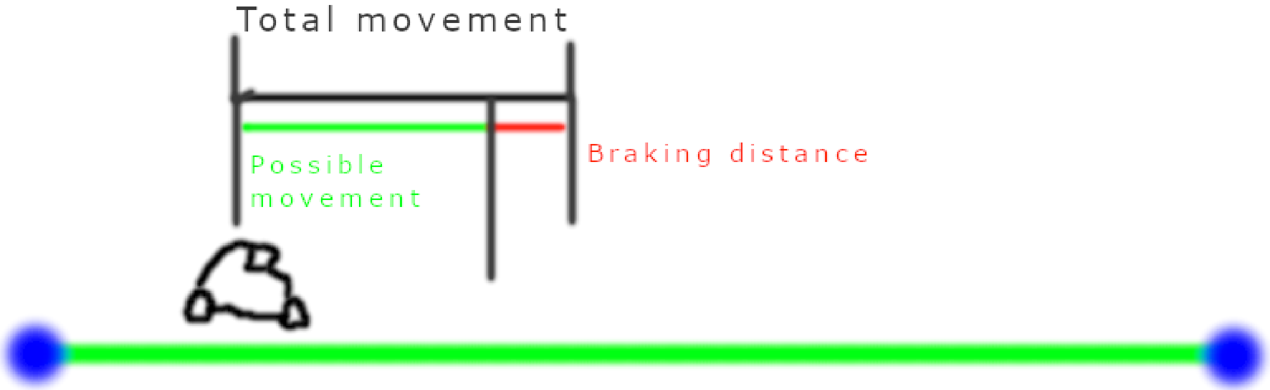
Collision component

- Travelers may not occupied position occupied by another
- A traveller is not allowed to move if it the movement would result in moving inside another
- Travelers moved sequentially
  - Traveller is guaranteed to move its maximum possible distance



Braking distance-based separation

- Travelers maintain separation sufficient to come to a complete halt should the car in front of them stop suddenly
- The car begins to de-accelerate if another car occupies any point within its braking distance.



Agent Behaviour

- Buildings control spawning of new agents
- Initial spawn-position determined by the building spawning the agent and the nearest edge
- Destination determined with a weighted stochastic process

Agent Behaviour – Destination Selection

- Several building types exist in simulation
  - Residence, office, shops, etc.
  - Each building type given specific weighting
- Roulette selection used to choose building type using arbitrary weight
- Destination chosen to be random building of chosen type
- If buildings don't exist or not possible to travel to destination, the agent despawns

Agent Pathfinding

- Once a source and destination point have been generated by the simulation
- Path-finding algorithm calculates, best path from source to destination points.
- If an agent is to spawn in an area of the map where the path is not possible, the agent despawns.

Agent Pathfinding Implementation

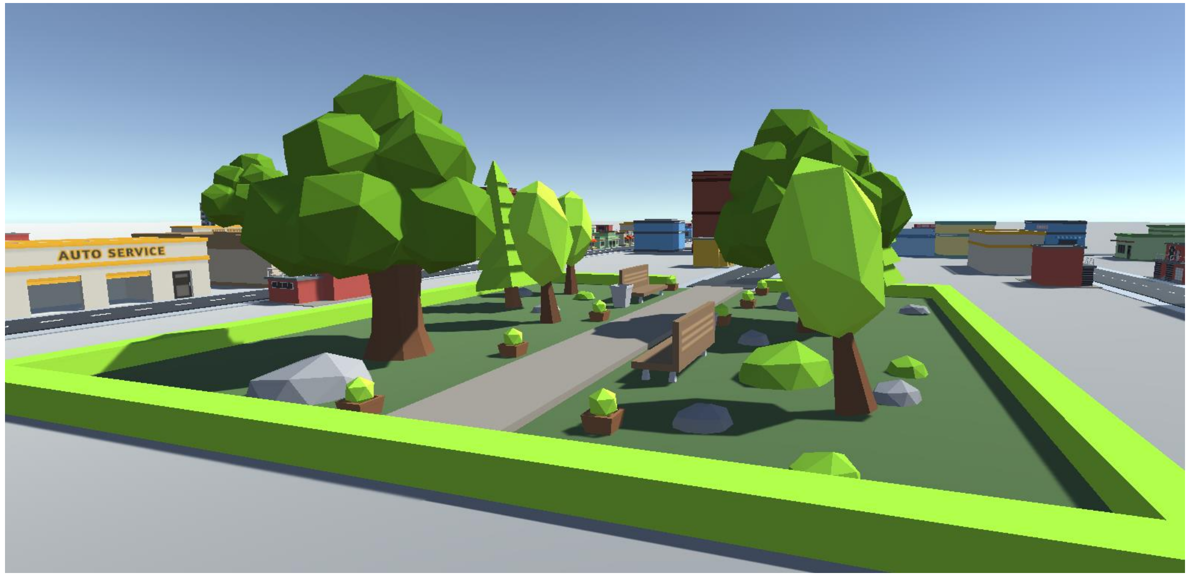
- Constructor taking the initial and destination locations, and uses the graph to create a path through the nodes
- Djikstra algorithm pathfinding
- path is defined as a list of waypoints.
- ConstructPath algorithm: builds the path using the shortest path in djikstra and also addresses some issues where the agents would pass the destination
- Created other helper methods:
  - IsDestinationBetween: checks if a point is between two waypoints
  - ClosestWaypointOnEdge: determines which waypoint is closer to a position on an edge.

Visual Representation Components

- Map
- Agents
- Menu

Visual Representation: Map

- Roads
  - Automatically generated from waypoints and edges.
- Buildings for Spawning
  - Manually placed and impact the stochastic spawning of agents.
- Other
  - Manually placed buildings and parks to make the scene more real.

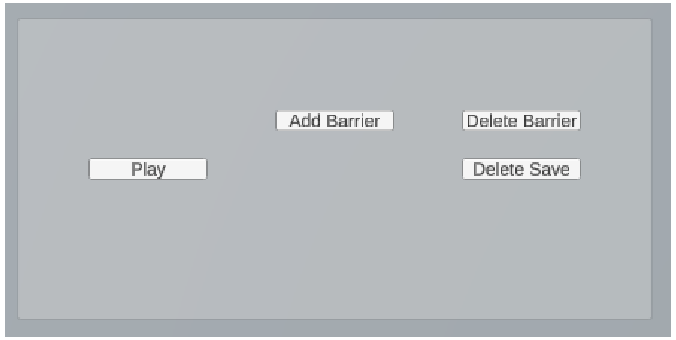


Visual Representation: Agents

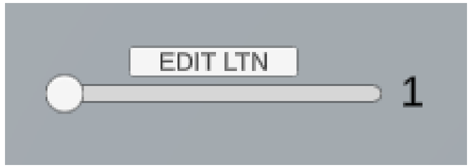
- Cars
  - Cars are displayed as a basic car of a random pre-defined colour and model.
- Pedestrians
  - Pedestrians are displayed as small blue pawns.
- Movement
  - Agents move along their specified paths (road or pavement) and turn instantly at junctions.

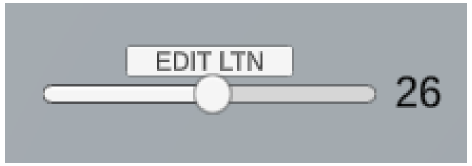
Visual Representation: Menu

- Edit menu:
  - This menu gives some options for custom LTN creation and saving configurations.



- Runtime menu:
  - This menu allows the user to speed up simulation or return to LTN editing.





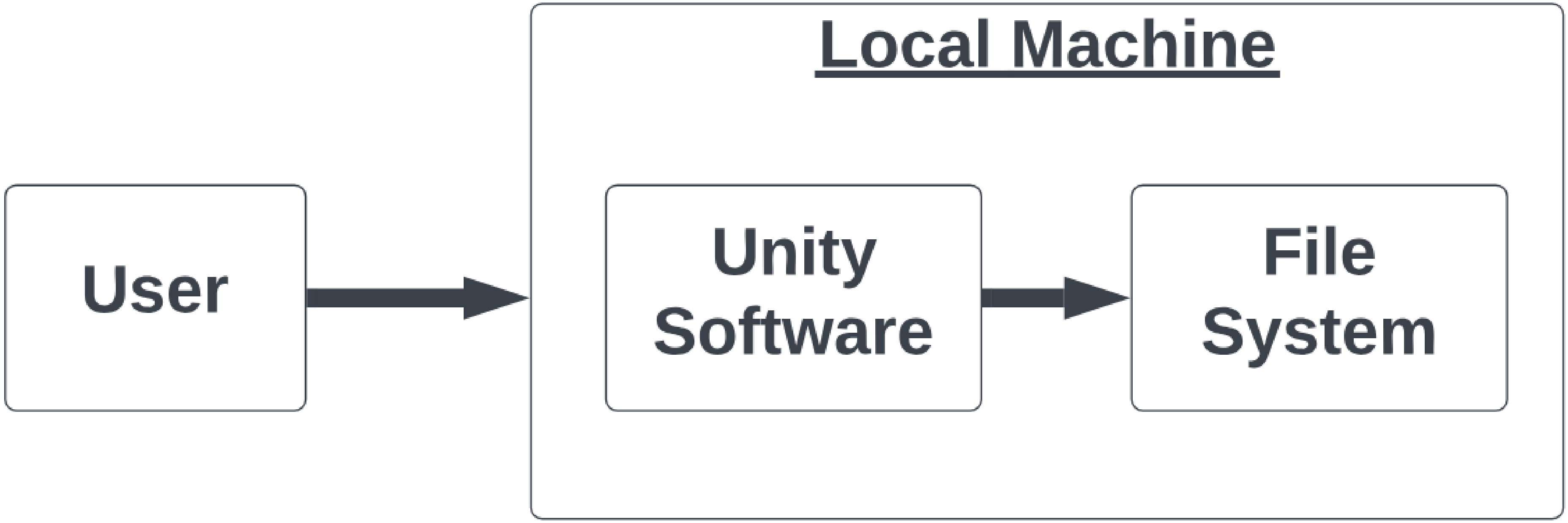
Measuring Key Variables

- Measured variables for each agent:
  - Start/end time of travel
  - Path traversed
  - Derived statistics:
    - Time taken
    - Distance travelled
    - Pollution, noise, etc.
- Statistics can be used to inform decisions and aid traffic planning

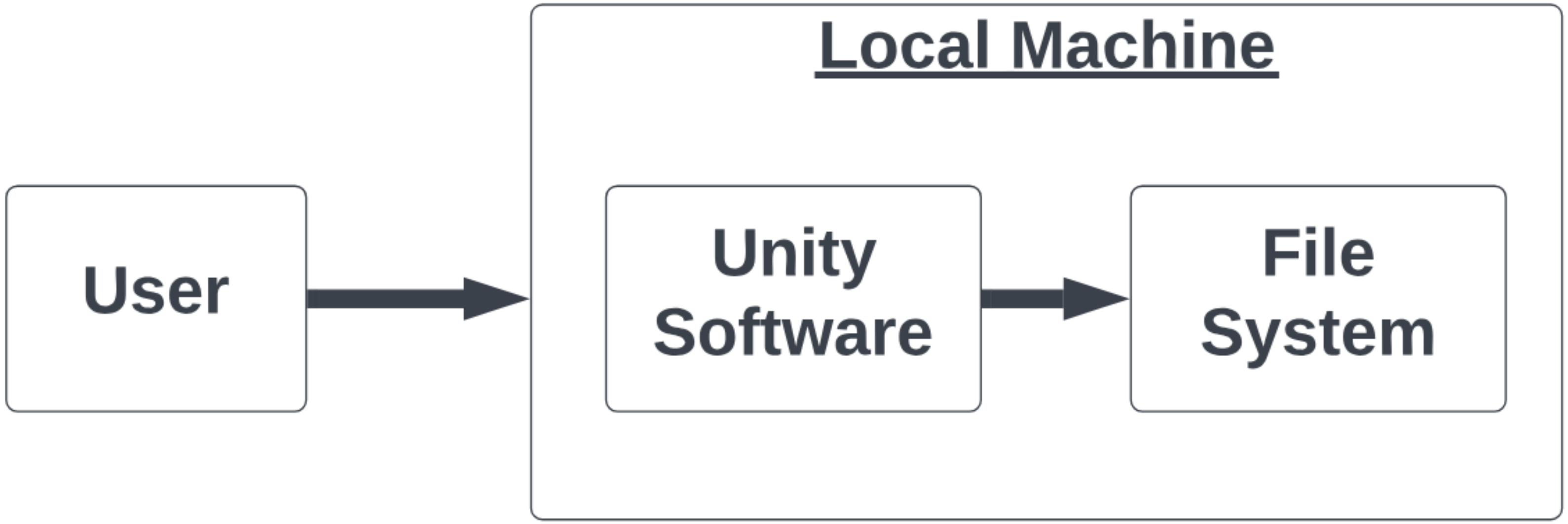
Production of Statistical and Graphical Results

- Due to the nature of the initial requirements of the MVP, the complexity of the statistics available are low.
- Travel time of the agent from source to destination can be measured with a system clock. From agent travel times, total and average air pollution can be derived using sourced statistics.
- Travel restrictions of residents can also be inferred from the number of times the simulation was forced to respawn an agent due to pathfinding failure.
- The simulation should collect key information as it runs.
- The simulation should calculate statistical data, which can be visualised in a graph.
- The simulation should visualise traffic moving on the map during simulation runtime.

Implementation

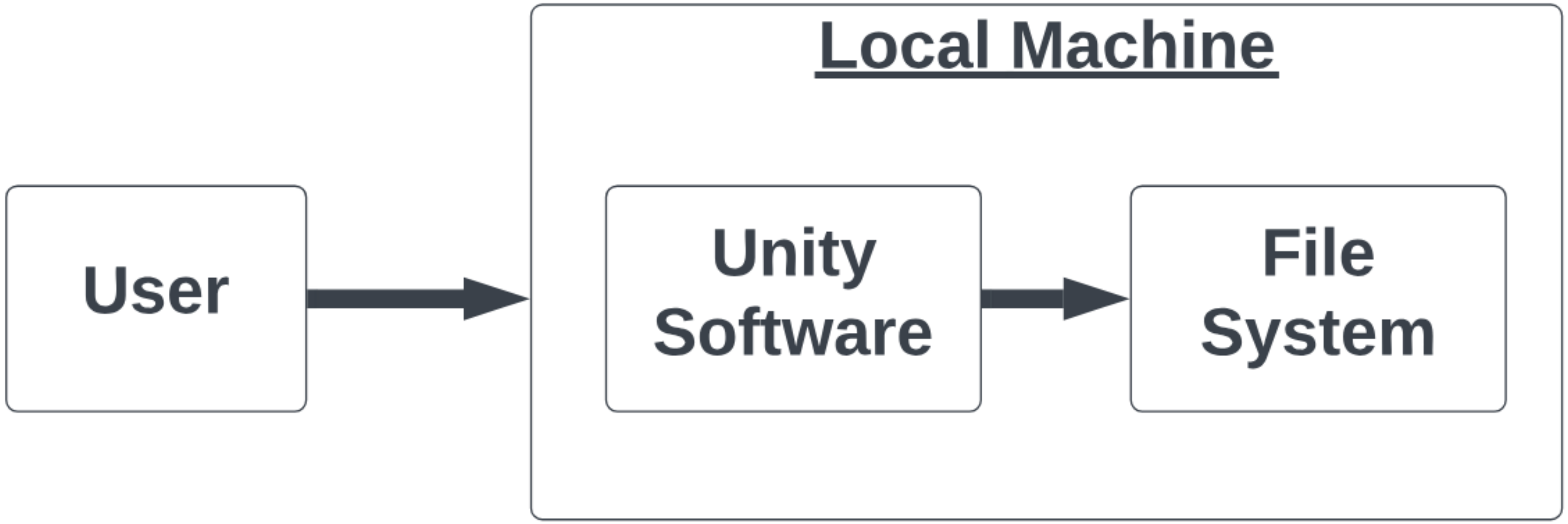


Implementation



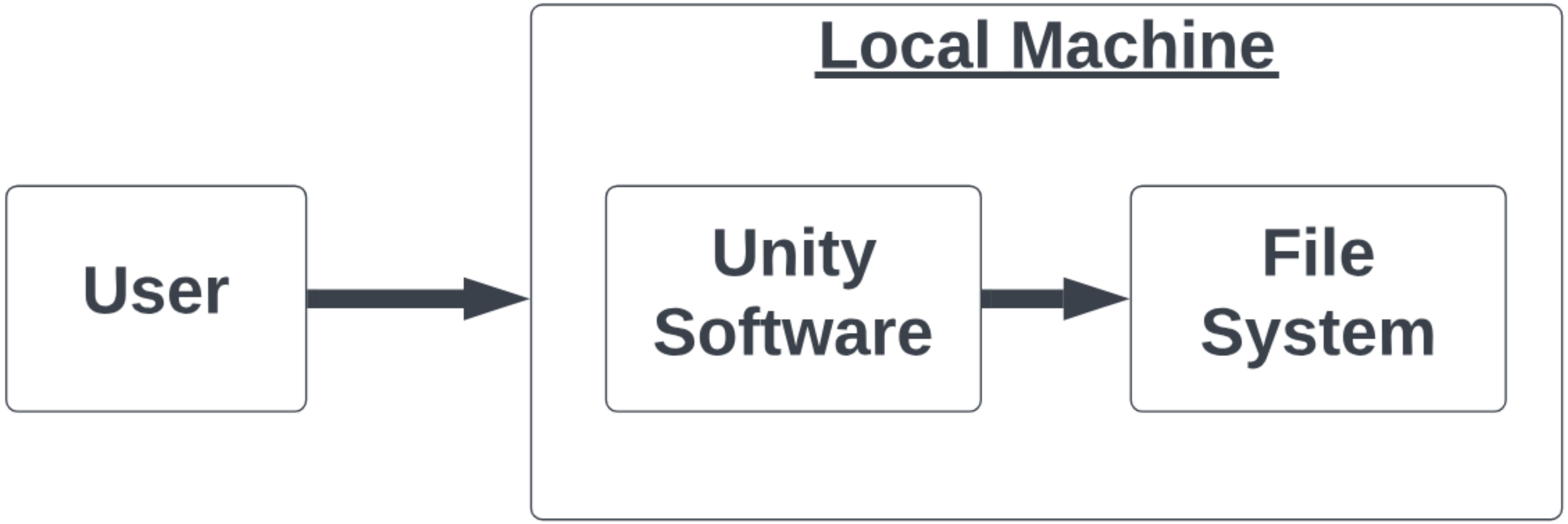
- Users:
  - Developer.
  - Client.
  - Member of the public at Royal Albert Memorial Museum.

Implementation



- Unity Software:
  - A game engine that runs the simulation.
  - Requires the Unity Editor to make changes.
  - All files are stored as a Unity project.

Implementation



- File System:
  - The files stored on the local machine where Unity Software can store information.
  - Entirely self-contained, so no Wi-Fi or Bluetooth to communicate.

The Future

- What we will do:
  - Event list.
  - Diagnostic tools / more ways to assess LTN effectiveness.
  - Other types of vehicles – including cyclists.
  - Map of Heavitree.
  - Traffic lights / pedestrian crossings.
  - Added probabalistic events.
  - User Interface improvement.
  - Improved stochastic spawning/destination determination.
    - Specific types of traffic to/from certain locations
  - Improved graphics.
- What we hope to do:
  - One-way roads.
  - Dynamic routing – e.g. high traffic by schools.
  - Floyd-Warshall pathfinding algorithm – static routing tables.
  - Buses and lanes for buses and bikes.

Demonstration