

MileStone1: Final Project Plan

****Group Members****

1. Member 1: Ky Kingslien
2. Member 2: Gabe Malicoat
3. Member 3: NATHAN D WILLETT

****Project Description****

We will be simulating a traffic system where vehicles move through a grid representing a city layout. The grid will consist of road cells and intersection cells. Vehicles will move randomly through the grid, and intersections will have traffic lights that change state at set intervals. The simulation will allow for dynamic adjustment of vehicle speed and traffic light timing to observe the effects on traffic flow.

****Classes and Public Methods****

1. TrafficMain:

- Methods:
- `main(String[] args)` : Entry point of the program to initialize and start the simulation.

2. TrafficGUI:

- Methods:

- `TrafficGUI(TrafficSimulation simulation)` : Constructor to initialize the GUI with a given simulation.

- `initialize()` : Initializes the GUI components and layout.

3. TrafficSimulation:

- Methods:

- `TrafficSimulation(int width, int height)` : Constructor to initialize the simulation grid.

- `start()` : Starts the simulation.

- `pause()` : Pauses the simulation.

- `reset()` : Resets the simulation.

- `update()` : Updates the state of the simulation.

- `setVehicleSpeed(int speed)` : Sets the speed of the vehicles.

- `setTrafficLightTiming(int timing)` : Sets the timing of the traffic lights.

- `getGrid()` : Returns the simulation grid.

- `getVehicleSpeed()` : Returns the vehicle speed.

4. Grid:

- Methods:

- `Grid(int width, int height)` : Constructor to initialize the grid.

- `updateGrid()` : Updates the state of the grid.

- `getGrid()` : Returns the grid as a 2D array.

- `getVehicles()` : Returns the list of vehicles in the grid.

- `getWidth()` : Returns the width of the grid.

- `getHeight()` : Returns the height of the grid.
- `addVehicle(Vehicle vehicle, int x, int y)` : Adds a vehicle to the grid at specified coordinates.

5. Cell (abstract):

- Methods:
 - `getX()` : Returns the x-coordinate of the cell.
 - `getY()` : Returns the y-coordinate of the cell.
 - `update()` : Updates the state of the cell.
 - `getColor()` : Returns the color of the cell.

6. RoadCell (extends Cell):

- Methods:
 - `update()` : Updates the state of the road cell.
 - `getColor()` : Returns the color of the road cell.

7. IntersectionCell (extends Cell):

- Methods:
 - `update()` : Updates the state of the intersection cell.
 - `getColor()` : Returns the color of the intersection cell.
 - `setTiming(int timing)` : Sets the traffic light timing.
 - `changeLight()` : Changes the traffic light state.
 - `isLightGreen()` : Returns whether the traffic light is green.

8. Vehicle:

- Methods:

- `` Vehicle(Cell initialCell, Color color)`` : Constructor to initialize the vehicle with a starting cell and color.

- `` void move(Cell newCell)`` : Moves the vehicle to a new cell.

- `` Cell getCurrentCell()`` : Returns the current cell of the vehicle.

- `` Color getColor()`` : Returns the color of the vehicle.

- `` void update(Grid grid)`` : Updates the state of the vehicle.

9. Car (extends Vehicle):

- Methods:

- `` Car(Cell startCell, Color color)`` : Constructor to initialize the car with a starting cell and color.

10. Truck (extends Vehicle):

- Methods:

- `` Truck(Cell startCell, Color color)`` : Constructor to initialize the truck with a starting cell and color.

Inheritance Hierarchy Diagram

