Class Description: version 1.0

**Form**: the main form where user will do the operations.

Attributes:

Timer: a control component of the form.

Gird: a control component of the form.

Simulator: a concrete object of class Simulator.

Methods:

Form (): constructor.

StartSimulator(): a method which will start the simulation.

PauseSimulation(): a method which will pause the simulation.

StopSimulation(): a method which will stop the simulation.

Navigate (): a method which will let user allow a group of cars come from somewhere and go to somewhere.

SaveProject(): a method which will allow the user to save the current project.

LoadProject(): a method which will allow the user to load a project.

Timer\_Enabled(): enable the timer, called in the StartSimulation method.

Timer\_Disabled(): disabled the timer, called in the PauseSimulation or StopSimulation method.

UpdateInformation(): when the simulation is on, it update the information and show them on the form every second.

**Gird:** a component of the form, contains all the crossings, lanes, moving objects and traffic lights.

Attributes:

Height: the height of the gird.

Width: the width of the gird.

Methods:

Gird(): the constructor.

AddCrossing(): add a crossing, with the type the user choose, to the gird.

RemoveCrossing(): remove a chosen crossing from the gird.

Clear: remove all the crossing in the gird.

Occupied(): check if there is already a crossing in some specific cell in the gird.

**Simulator:** the class which control the simulation.

Attributes:

Crossings: list of all the crossing.

currentStatus: current status.

Methods:

Simulator(): constructor.

Start(): start the simulation.

Pause(): pause the simulation.

Stop(): stop the simulation.

SetStartPoint(): when the user want to navigate from somewhere to another place, this is for setting the start position.

SetEndPoint():when the user want to navigate from somewhere to another place, this is for setting the end position.

Navigate(): allow the user want to navigate from somewhere to another place.

AddCrossing(): Add a new crossing.

RemoveCrossing(): Remove a crossing.

RatateCrossing(): Rotate a crossing.

SetFlow(): set the number of flow for a specific land.

Connected(): check whether two lanes are connected.

GetCurrentState(): get current state.

GetAllCorssing(): get all the crossings in the current simulation.

CalculateNextSecond(): calculate all the changes of the simulation for the next second.

**Crossing:** the class which contains the information for a crossing.

Attributes:

Lanes: the lanes surrounded around the crossing.

TrafficLights: the traffic lights on each lane.

Location: the position of the crossing.

Methods:

Crossing(): constructor.

RightRotate(): rotate the crossing 90 degree clockwise.

GetAllLanes(): get all lanes around this crossing.

GetAllTrafficLights(): get all traffic lights in the lanes those are around the crossing.

**CrossingTypeA, CrossingTypeB:** special type of crossing.

**TrafficLight:** the traffic light on the lane.

Attributes:

RedInterval: the interval for the red light.

GreenInterval: the interval for the green light.

YellowInterval: the interval for the Yellow light.

Methods:

TrafficLight(): constructor.

SetInterval(): set the interval for red, green and yellow light of this traffic light.

**Lane:** the lanes which contains all the moving objects.

Attributes:

Flow: the current number of moving objects on this lane.

MaxFlow: the capacity of this lane.

MovingObjects: all the moving objects on this lane.

Methods:

Lane(): the constructor.

SetFlow(): set the number of moving objects on this lane.

GetFlow(): get the number of moving objects on this lane.

ObjectJoin(): a new moving object comes to this lane, flow increase by 1.

ObjectLeave(): a moving object leaves this lane, flow decrease by 1.

ActiveAllObjects(): activate all the moving objects on this lane.

DeActiveAllObjects(): deactivate all the moving objects on this lane.

**MovingObject:** the class for the moving objects (cars or pedestrians).

Attribute:

TimerCounter: to decide whether this object should turn to another new lane or stay on the current lane.

Actived: the status of the car.

Methods:

MovingObject(): constructor.

Start(): activate the car.

Stop(): deactivate the car.

Turn(): turn to another lane.

**Sequence diagrams:**

Add crossing

Delete crossing

Play

Pause

Stop

Rotate

Navigate

Alter flow