# Zadání

## Část 1: pasivní vizualizace (až 10 bodů)

Základní funkční požadavky (5 bodů)​ : Program bude možné spustit z příkazové řádky (dávkou Run.cmd) s volitelným celočíselným parametrem určujícím, který ze scénářů dopravní simulace (poskytnutých instancí třídy ​ Simulator metodou ​ getScenarios​ ) má být spuštěn. Nebude-li parametr zadán, spustí se scénář 0. Program zobrazí všechny klíčové elementy dopravní sítě (tj. křižovatky, silnice a vozidla na silnicích) spuštěného scénáře způsobem, který umožňuje posoudit jejich aktuální stav. Vizualizace se bude pravidelně aktualizovat.

Další požadavky: ​ Elementy budou rozmístěny v okně tak, aby se maximálně využil jeho prostor, ale zároveň nedošlo ke zkreslení souřadnic v nějakém směru (tj. Zobrazení zachovává poměr vzdáleností a úhly). Velikost okna musí být možné měnit a vizualizace se musí nové velikosti přizpůsobit. Součástí odevzdání bude dokumentace v předepsaném formátu.

## Řešení

Prvním krokem při tvorbě aplikace bylo rozdělit aplikaci na třídy pro lepší a jednodušší programovaní. Ve třídě View.java jsem vytvořila jednoduché GUI s dvěma tlačítky pro ovládaní Timerem a JComboBoxem pro nastavení scenáře přímo z aplikace. Ovladání tlačítek zajišťuje třída Controller, která volá příslušné metody na reakci od uživatele.

Dalším nejdůležitějším krokem v tvorbě samotné simulace dopravy bylo přepočítat všechny souřadnice z třídy Simulator, které byly zadány v metrech na souřadnice okna (pixely). Taky bylo nutné určit vhodnou změnu měřítka pro posun ve směru X a Y. To všechno je naimplementováno v hlavní vykreslovací třídě DrawPanel.java v metodách computeModelDimensions(), computeModel2WindowTransformation(int, int), model2window(Point2D).

Po každém vykreslování okna se inicializuje pomocný seznam silnic, které spojují křižovatky, přepočítává šířka silnice a nastavuje se barva pozadí. Také se zde přepočítávají souřadnice okna a volá se metoda drawTrafficState(Simulator, Graphics2D), která vykreslí celou dopravní síť a auta.

Pro vykreslení křizovatky jsem prošla skrz celé pole křižovatek (getCrossroads()). Pro každou křižovatku pomoci metody drawRoadSegment(RoadSegment, Graphics2D) vykresluji všechny silníce tvořicí křižovatku. Pomocí metod getBackwardLanesCount() a getForwardLanesCount() zobrazuji všechny jízdní pruhy silnic a přídávam každou novou silnici do List<Road>. Pro propojení křižovatky použímam metodu connectLanes(Lane, Graphics2D), ve které pomoci List<Road> zjíštuji nové souřadnice pruhů a propojuji je mezi sebou.

Na konci vykresluju auta, ve tvaru Ellipsy.

## Třidy

Celá aplikace je rozdělena na několik tříd, které se rozmístěny v balíku applikation. Struktura kódu připomíná softwarový vzor MVC. Applikace obsahuje kontroller, kde se vyvolavají Listennery pro grafické objekty. Ve třídě View jsou naimplementovany prvky GUI a jejich gettery a settery. Hlavní třida Main jen spojuje Controller a View a spouští se applikace.

* Main.java

Je hlavní třída projektu, ve která vytváří instance třídy Simulator, Controller a View. Take se tam naplní Jcombobox scénaři. Obsahuje metodu main.

* Controller.java

Controller kontroluje akci po stísknuti tlačítek a obsahuje timer Simulatoru. ​Timer zajišťuje pravidelné překreslování okna v intervalu 100 ms.

* View.java

Třída která odpovída za zobraxení prvků na okně. Obsahuje gettery a settery každého grafického objektů.

* DrawPanel.java

Je hlavní třída semestrální práce, ktera vykresluje dopravní simulace. Dědí od JPanel z Java Swing.

* + drawTrafficState(Simulator, Graphics2D)
  + computeModelDimensions()
  + computeModel2WindowTransformation(int, int)
  + model2window(Point2D)
  + drawCrossRoad(Graphics2D)
  + drawRoadSegment(RoadSegment, Graphics2D)
  + drawBackwardLanes(double, double, double, double, RoadSegment, Graphics2D, int)
    - Najde nové souřadnice pro zpětné pruhy a přída je do List<Road>.
  + drawForwardLanes(double, double, double, double, RoadSegment, Graphics2D, int)
    - Najde nové souřadnice pro pruhy jdoucí druhým směrem a přída je do List<Road>.
  + connectLanes(Lane, Graphics2D)
    - Vykresli všechné silnice z List<Road> a propojí křižovatku. Pro hledání silnic v poli byl použit stream vyhledávaní v poli.
  + drawLane(Point2D, Point2D, int, Graphics2D)
  + drawCar(Point2D, double, int, int, Graphics2D)
* Road.java

Pomocná třída pro ukladání souřadníc spojení křižovatky.