

## Nödvändiga beroenden:

- Arvsheirarkin från MotorVehicle till Car, Volvo240, Saab95, PlatformVehicle, Scania och SemiTruck. Dessa är nödvändiga för enormt mycket återanvändning av kod.
- Att SemiTruck har en Composition relation på Car i form av en ArrayList<Car>. Detta är nödvändigt eftersom Semitruck ska kunna lasta bilar. På samma sätt är WorkShops Composition relation på MotorVehicle nödvändig.
- MotorVehicles composition av en Timer är nödvändigt för att kunna uppdatera programmet med reglebundna intervaller. Timers beroenden av ActionListener och ActionEvent är nödvändiga av samma anledning. Dock borde denna relationen vara mellan Model och Timer enligt MVC.
- Att CarController skapar en CarView och att CarView kallar på metoder i CarControllern som skapade den. Detta behövs för att knapparna ska kunna göra saker. Knapparna har i sin tur ett nödvändigt beroende på ActionListener och ActionEvent.

## Onödiga beronden:

- Pilen mellan DrawPanel och MotorVehicle är eftersom DrawPanel använder getModelName() för att veta vilken bild den ska välja. Detta kan tas bort genom att vid skapandet av DrawPanel ge den en lista med de aktuella Modellnamnen. Samma gäller för moveIt
- Att kollisionshanteringskoden använder sig av Directions för att undvika att bilen inte kan kollidera med en vägg den inte kör mot. Dvs undvika att bilen räknas som "förbi" väggen den nyss kolliderade med.

## SoC och SRP:

- Flera metoder kan delas upp för bättre läsbarhet. T.ex Constructorn i DrawPanel
- Bättre uppdelning enligt MVC
  - Carview borde döpas om till något med widgets och bilda controllern tillsammans med Carcontroller
  - Mycket från CarControllern bör vara en del av modellen, t.ex timern och kollisionlogiken. Delen som startar programmet bör ligga i en application klass.
  - Drawpanel bör ha färre beroenden och istället få nödvändig information om modellen via en observer
- (Widgets i CarView hade kunnats lägga i olika klasser, så är det CarView som skapar dem. Vill man ändra eller lägga till en widget blir det lättare.)
- initComponents metoden i CarView är alldeles för lång och bör delas upp i flera metoder

## Refaktoriseringsplan:

Bättre användning av MVC:

- Skapa en Application klass dit man flyttar koden för att köra programmet.
- Skapa en klass som hanterar modellens tillstånd där Timern finns.
- Skapa en observer interface som skickar informationen om uppdateringar till DrawPanel

Få bort kollisionhanterings beroende av Directions:

- Ändra så HandlewallCollision körs efter move()
- Kolla efter kollision utan direction (bara utefter koordinater)
- Om kollision ska bilen få väggens position
- Resten av koden är som vanligt

Få bort drawPanels beroende av MotorVehicle.

- Skicka en lista med de aktuella modellnamnen när drawPanel skapas i Applicationen
- Ändra så att bilderna väljs utifrån denna lista
- Ta bort att man skickar in en lista med MotorVehicles

Förbättra konstruktörer

- Kolla på konstruktörer och se om koden i dem kan extraheras till metoder som kallas när konstruktören körs

Förbättra initComponents

- Extrahera kodbitar till engla metoder som initComponents kallar på