

Problem:

Tät koppling mellan View och Controller:

DrawPanel har en referens till CarController och använder dess vehicles-lista direkt, vilket gör att View är beroende av Model-data via Controller.

Modellogik i Controller:

CarController innehåller logik för att hantera bilarnas rörelse och kollisioner (t.ex. i TimerListener). I ett rent MVC-mönster ska sådan logik ligga i Model, inte i Controller. Controller ska bara förmedla kommandon mellan View och Model.

View hanterar viss Modellogik:

DrawPanel laddar bilder baserat på vehicles-listan direkt från Controller, vilket innebär att View gör mer än att bara rendera data – den hanterar även viss databehandling.

Beroenden i fel riktning:

Controller (CarController) har en referens till View (CarView), vilket är en omvänd beroenderiktning. Idealt sett ska Controller inte veta om View utan bara skicka uppdateringar via ett observer-pattern eller liknande.

Design Patterns:

Drawpanel skulle ha mindre referenser till CarController ifall det fanns en observer istället för direkt koppling.

En factory (VehicleFactory) skulle göra det mer dynamiskt att skapa fler biltyper. Istället för att lägga till varje biltyp manuellt på alla ställen de behövs kan man istället lägga till bilarna med bara en funktion med hjälp av factory.

