

Peer-Review 3

Objektorienterad programmering med Java

Emil Björklund - embj3739
emilbjorklund@live.com

Källkod: Leo Blumenberg

April 1, 2019

Inledning

- IDE: Visual Studio Code
- Java version: JDK 8 update 191
- OS: MacOS Mojave

Peer Review

Programvarutestning

Programmet gick inte att köra direkt från terminal utan ändringar. Eftersom sökvägen till bilderna är hårdkodade och behöver vara placerade i samma mapp som .class filerna var bilderna tvugna att flyttas innan programmet kunde exekveras. När bilderna ligger i samma mapp som class filerna kunde programmet köras utan problem.

Kravvalidering

- **Det ska vara möjligt att flytta omkring och vända på fotot.**
Alla bilder går att flytta omkring och vända i fönstret utan problem.
- **Användaren vänder på ett foto genom att klicka på det, och flyttar ett foto genom att hålla nere musknappen och dra musen.**
När man klickar på foton svarar inte bilderna ibland och två klick kan behövas, dock osäker på om det beror på användarens dator eller koden.
- **Förflyttningar ska vara animerade, dvs användaren ska se hur fotot flyttar på sig när han/hon drar musen.**
Förflyttningarna upplevs hackiga då bilderna inte riktigt följer med musens rörelse i start ögonblicket utan hoppar något. Detta kan bero på att musens position inte tar hänsyn till vart i bilden musen är. I övrigt följer bilderna med bra.
- **Både när ett foto flyttas och när det vänds hamnar det "överst" så att det täcker andra foton som ligger i överlappande position.**
När ett foto byter position eller klickas med musen hamnar denna bild överst.
- **Både när ett foto flyttas och när det vänds gäller att det "översta" väljs; om flera foton täcker muspekarens position väljs det översta.**
När alla bilder ligger på hög och muspekaren klickas på högen väljs den understa bilden, detta är motsatsen av vad som är angivet.

Kodstruktur

Koden har tre stycken klasser som tillsammans utgör programmet. Main konstruktorn är i klassen CardFrame som deklarerar ett nytt objekt av typen CardFrame. I klassen CardFrame deklaras det en panel av typen CardPanel, denna panel placeras sedan i en container som visas. Objektet panel anropar sedan metoden add som läser in bilderna som ImageIcon i en lista. Denna klass är tydligt strukturerad och kompakt. Det enda som skulle kunna göras annorlunda är att läsa in bilderna i en loop som itererar på antalet bilder i mappen. En sådan approach skulle göra det enkelt för användaren att lägga till eller ta bort bilder. Klassen

CardPanel implementerar ActionListener, MouseListener och MouseMotionListener. Att enbart låta ActionListener måla upp bilderna på panelen som är beroende av en timer gör att animeringarna blir hackiga. En lösning på detta är att låsa eventen som triggas av MouseListener samt MouseMotionListener också måla upp bilderna på panelen. Att vara beroende av en timer är nog inte optimalt. För looparna som används för att iterera igenom arrayen:

```
for (int i = cardList.size() - 1; i > -1; i--)
```

skulle kunna skrivas såhär istället och bryta loopen med ett break statement:

```
for (int i = cardList.size() - 1; i >= 0; i--)
```

När bilderna målas upp blir den sista bilden i arrayen målad sist vilket resulterar i att bilden på sista positionen är överst. När bilderna ligger på hög markeras den understa bilden och det kan bero på att målningen endast sker genom ActionListener och för loopen som itererar genom arrayen endast bryts om man dubbelklickar. Detta medför att omplaceringen fortsätter tills arrayen är slut och bilden längst ner på första plats i arrayen kommer att läggas på sista position och hamna överst. I klassen Card skulle en metod för att ta hand om den relativa distansen mellan musenpekaren och bildens position vara bra. Den nuvarande implementationen tilldelar bilden nya x och y koordinater baserat på muspekarens koordinater. Detta resulterar i att oavsett vart på bilden muspekaren befinner sig kommer den nya bilden målas upp med mitten på muspekaren och animeringen uppfattas hackig.

Slutord

Helhetsintrycket var mycket bra och koden är välskriven med väl placerade kommentarer för att underlätta förståelsen. Att implementera en lista för att lagra bilderna var mycket klokt och det är något som jag tar med mig. Rapporten är utförlig och kompletterar källkoden på ett bra sätt. Det skulle kanske vara bra att dela upp rapporten i några rubriker för att enklare kunna läsa på om specifika implementeringar.