

DVA229, F# - A Simple Graphics Editor

Beskrivning

Den enkla grafiska redigeraren fungerar på så sätt att man genom visualisering kan bygga, ladda, spara och ändra en scen utifrån enkel geometri. Det finns två olika former att skapa i scenen vilket är cirklar och kvadrater. Dessa kan bli tilldelade en färg, bli

förstorade/förminskade, flyttas i scenen och tas bort. För att få detta att fungera användarvänligt, responsivt och simpelt implementerade vi ett event-drivet program som tar hjälp av grafiska knappar och liknande. Genom detta kan användaren utföra alla de olika operationerna genom knapptryck i ett grafiskt interface på ett logiskt sätt.

En stor del av vår lösning till projektet är gjord med hjälp av biblioteken System.Drawing och System.Windows.Forms som används för programmets funktionalitet och utseende.

Representationen av vår scen är i form av en lista innehållande objekt utifrån en egengjord class-hierarki. Class-hierarkin består av det som behövs för att skapa nya "Shapes", t.ex. X- och y-position, size och color, men den innehåller även metoder för att rita objektet i fråga på skärm. För att skapa vår dynamiska scen med reaktiv programmering använde vi oss utav observables och en rekursiv loop som tar en lista av shapes som in-parameter. Observables kopplas till samtliga knappar och utifrån dessa reagerar programmet på knapptryck med ett rekursivt anrop till loopen med en ny objekt-lista, vilket gör att programmet körs i en evig loop med en scen som uppdateras.

Problem och fynd

Vi hade en del problem med att rita objekt på skärmen med snabba uppdateringar om man snabbt klickade på knappar då det uppstod s.k. "flickering" för varje uppdatering. Detta löstes genom att skapa en egengjord class till vår huvud-"window" som kallas "MainWindow" i filen GUI.fs. I denna class ärver vi från klassen "Form" och kan därmed sätta egenskapen "DoubleBuffered" till true, vilket löste problemet. Ett intressant fynd vi gjorde i samband med detta och optimering av CPU-arbete var att använda funktionerna "Invalidate" och "Update", vilket gjorde det möjligt att välja specifika regioner att uppdatera och rita om i scenen. Detta implementerades så att vi enbart ritade om det område som det nuvarande valda objektet befinner sig i.

Ett annat problem var att ha flera olika typer av shapes i samma lista och rita dem utifrån deras egenskaper. Man behövde t.ex. kalla DrawEllipse om det var så att objektet var en cirkel, eller

DrawRectangle om det var en kvadrat. Detta löstes genom att införa en abstrakt "draw"-funktion i bas-klassen som sedan override:as av sub-klasserna. Ritningarna blev därmed individuella och anpassade för varje shape utan att det syns utifrån.

Ett mindre problem var hur man skulle spara och ladda filer. Vi visste inte hur filsystemet kunde anropas och vår lösning blev att "hårdkoda" den absoluta sökvägen till en textfil, men då krävdes det att man skrev in sökvägen i koden för varje specifik användare, vilket inte är så bra programmering. Det löstes genom att anropa en biblioteksfunktion som hämtar projektets nuvarande mapp och returnerar dess sökväg som en sträng.

User Manual

Programmet körs i Visual Studio och biblioteket FSharpX krävs för att kompilera koden. När programmet är byggt och körs så öppnas ett fönster som heter "Simple Graphics Editor". Här lyder en kort sammanfattning av hur programmet används följt av vad varje knapp har för funktionalitet:

Kort sammanfattning

- För att skapa, styra och redigera objekt i scenen används knapparna i den nedre "toolbaren". (OBS! för att kunna redigera eller flytta objekt krävs det att man markerar vilket objekt man vill använda först. Detta görs med knappen "Select next", eller tangenten "N".)
- För att ladda eller spara till fil finns en drop-down menu uppe i vänstra hörnet med namn "File".
- Det går att använda snabbtangenter för diverse kommandon. Dessa står inom paranteser i varje knapps textruta. (OBS! Snabbtangent står inte skrivet i de knappar som används för att flytta på objekt, men snabbtangenterna W, A, S och D kan användas för detta.)

Knapparnas funktionalitet

- Add square (T) button lägger till en rektangle med förbestämd storlek, färg och position.
- Add circle (Y) button lägger till en cirkel med förbestämd storlek, färg och position.
- Set color (C) ändrar färg på ett markerat objekt efter vilket som är valt i drop down menyn bredvid. (Vad markerat objekt innebär förklaras i Select next button nedan.)
- Drop down menu har blå som standard, menyn ändrar den förbestämda färgen som används vid skapningen av nya rektanglar och cirklar.

- Select next (N) traverserar igenom objekten i scenen och ett markerat objekt belyses med en ram. Att markera ett objekt tillåter användaren att ändra specifika objekts storlek, färg, position och ger möjligheten att ta bort ett objekt.
- Resize smaller (Z) gör ett markerat objekt mindre i både höjd och bredd.
- Resize bigger (X) gör ett markerat objekt större i både höjd och bredd.
- Remove (R) tar bort ett markerat objekt från scenen.
- Move Forward (F) flyttar ett objekt i Z-led så att det visas vid överlappning av andra objekt.
- Move Backward (G) flyttar ett objekt i Z-led så att det göms vid överlappning av andra objekt.
- ← ↑ ↓ → (A, W, S, D), flyttar ett markerat objekt runt i scenen enligt riktning som indikeras av pilarna.
- Save sparar scenen som en sträng till en fil som automatiskt döps till Shapes.txt.
- Load laddar en fil och läser dess sträng rekursivt och skapar nya objekt med de värden som hämtades från filen.
- Exit stänger ner programmet.