

GUI («Graphical User Interface»)

- Om GUI
- JavaFX

Se også på

- Infoskrivet <https://www.uio.no/studier/emner/matnat/ifi/IN1010/v18/GUI/fx-intro-in1010.pdf>
- Programkoden i <https://www.uio.no/studier/emner/matnat/ifi/IN1010/v18/GUI/>

Hvorfor trenger vi GUI?

Tripp-trapp-tresko med og uten GUI

```
+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+
```

```
$ java TTT1
X spiller 8
Hva spiller 0? 5
O spiller 5
X spiller 1
Hva spiller 0? 3
O spiller 3
X spiller 2
Hva spiller 0? 7
O spiller 7
Vinneren er O!
```

```
$ java TTT2
+---+---+---+
| X |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
Hva spiller 0? 5
+---+---+---+
| X |   |   |
+---+---+---+
|   | O |   |
+---+---+---+
|   |   | X |
+---+---+---+
Hva spiller 0? 8
:
Hva spiller 0? 2
Vinneren er O!
+---+---+---+
| X | O |   |
+---+---+---+
|   | O |   |
+---+---+---+
| X | O | X |
+---+---+---+
```



Fordeler med GUI

- Mer intuitivt å bruke
- Færre muligheter for brukerfeil
- Visuelt mer tiltalende

Ulemper

- Mer komplisert å programmere
- Mange ulike GUI-biblioteker å velge blant
- Svært få biblioteker fungerer for både Linux, Mac og Windows.

Historien om Qt

- Opprinnelig laget av det norske firmaet *Trolltech* i 1994.
- Hovedideen var at programmer skrevet i C++ kunne linkes med et Qt-bibliotek og så kjøre på ulike systemer.
- Sannsynligvis det mest brukte generelle vindusbiblioteket i 1990- og 2000-årene.
- Solgt til Nokia i 2008 (og siden til Digia og The Qt Company).
- Finnes i både kommersiell og åpen kildekode-versjoner.
- Brukes i dag i Google Earth, Mathematica, Tesla-biler, ...



Hvorfor GUI i IN1010?

- Allmennkunnskap for programmerere
- Et godt eksempel på bruk av OO
- Viser en ny programmeringstankegang:
Hendelsesorientert programmering (neste uke)
- Et godt eksempel på bruk av parallellisering og tråder
(neste uke)

Java og GUI

Java har alltid hatt som mål at programmene skal kunne kjøres uendret på alle plattformer. Derimot skal utseendet tilpasses operativsystemets «standard».

AWT («Abstract Window Toolkit») fra 1995:
et ganske enkelt grensesnitt mot OSets vindussystem

Swing fra 2007:
et mer avansert og generelt system bygget oppå AWT

JavaFX fra 2012:
et enda mer komplett og velstrukturert system

Hvordan lager vi et GUI-program?

Teatermetaforen

De som laget JavaFX tenkte seg at det en bruker ser i et GUI-vindu, minner mye om det man ser på teater når teppet går opp.

Teater



class Stage

Scene



class Scene

Kulisser



class Pane

Teatret

Et teater settes opp på denne måten:

```
import javafx.application.Application;
import javafx.stage.Stage;

public class Xxx extends Application {
    @Override
    public void start(Stage teater) {
        // Sett opp scene og kulisser

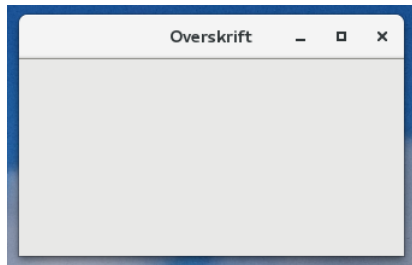
        teater.setTitle("Overskrift");
        // teater.setScene(scene);
        teater.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```


Hvordan lager vi et GUI-program?

Selv om det ikke er ferdig, kan programmet kjøres:

```
$ javac Xxx.java  
$ java Xxx
```



Hvordan lager vi et GUI-program?

- Alt med JavaFX ligger i biblioteket og vi må importere klassene vi trenger.
- Alle JavaFX-programmer må være subklasser av Application.
- main må kalle launch for å starte JavaFX.
- Hovedklassen må definere en start som setter opp det hele; parameteren angir hovedvinduet («teateret»).
- setTitle angir vinduets tittel.
- show angir at teppet skal gå opp.

```
import javafx.application.Application;
import javafx.stage.Stage;

public class Xxx extends Application {
    @Override
    public void start(Stage teater) {
        // Sett opp scene og kulisser

        teater.setTitle("Overskrift");
        // teater.setScene(scene);
        teater.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Scener og kulisser

Vi må også lage en scene og en (foreløpig tom) samling kulisser:

```
import javafx.application.Application;
import javafx.scene.layout.Pane;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Xxx2 extends Application {
    @Override
    public void start(Stage teater) {
        // Sett opp scene og kulisser

        Pane kulisser = new Pane();
        Scene scene = new Scene(kulisser);

        teater.setTitle("Overskrift");
        teater.setScene(scene);
        teater.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```



Ulike typer kulisser

Det finnes mengder av ting vi kan bruke som kulisser; vi skal bruke:

- tekst (class Text)
- firkanter (class Rectangle)
- sirkler (class Circle)
- linjer (class Line)
- polygoner (class Polygon)

Et eksempel: Hallo-programmet

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.text.Font;
import javafx.scene.text.Text;

public class Hallo extends Application {
    @Override
    public void start(Stage teater) {
        Text hilsen = new Text("Hallo, alle sammen!");
        hilsen.setY(40);
        hilsen.setFont(new Font(40));

        Pane kulisser = new Pane();
        kulisser.getChildren().add(hilsen);

        Scene scene = new Scene(kulisser);

        teater.setTitle("Velkommen til Java FX");
        teater.setScene(scene);
        teater.show();
    }

    public static void main(String[] args) {
        Application.launch(args);
    }
}
```



Vi må

- importere de klassene vi trenger
- opprette teksten:

```
Text hilsen = new Text("Hallo, alle sammen!");
```
- Vi kan tilpasse elementene ved for eksempel å endre fontstørrelsen:

```
hilsen.setFont(new Font(40));
```
- inkludere tekstelementet blant kulissene:

```
kulisser.getChildren().add(hilsen);
```

Nå kan programmet kjøres:

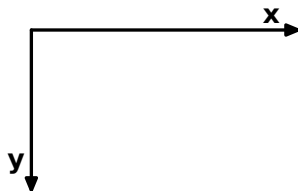
```
$ javac Hallo.java  
$ java Hallo
```



Plassering

Når vi har flere kulisser, må vi angi posisjonen så de ikke havner oppå hverandre. Koordinatsystemet er litt uvant:

- Y-aksen går *nedover*.

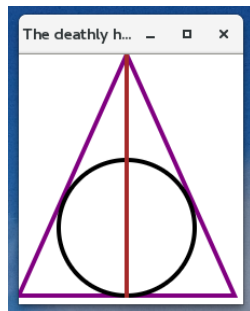


- Enheten er piksler på skjermen.

Et eksempel fra Harry Potter

Symbolet for *Deathly hallows* fra Harry Potter-bøkene kan lages slik:

- et Circle-objekt
 - sort rand med bredde 4; hvit innmat
- et Line-objekt
 - brunt, med bredde 4
- et Polygon-objekt
 - purpurfarget rand med bredde 4; hvit innmat



Et eksempel fra Harry Potter

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.shape.*;
import javafx.scene.paint.Color;

public class DeathlyHallows extends Application {
    @Override
    public void start(Stage teater) {
        final double r = 63;
        Circle stein = new Circle(r);
        stein.setCenterX(100); stein.setCenterY(224-r);
        stein.setStroke(Color.BLACK); stein.setStrokeWidth(4);
        stein.setFill(Color.WHITE);

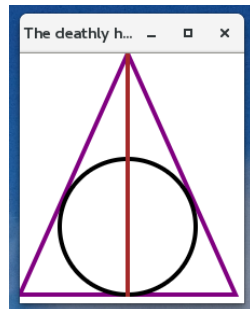
        Line stav = new Line(100,224, 100,0);
        stav.setStroke(Color.BROWN); stav.setStrokeWidth(4);

        Polygon kappe = new Polygon(0,224, 200,224, 100,0);
        kappe.setStroke(Color.PURPLE); kappe.setStrokeWidth(4);
        kappe.setFill(Color.WHITE);

        Pane kulisser = new Pane();
        kulisser.getChildren().add(kappe);
        kulisser.getChildren().add(stein);
        kulisser.getChildren().add(stav);

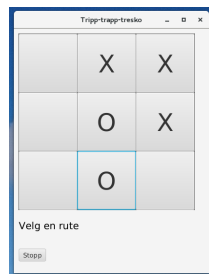
        Scene scene = new Scene(kulisser);

        teater.setTitle("The deathly hallows");
        teater.setScene(scene);
        teater.show();
    }
}
```



Plassering i rutenett

Noen ganger trenger vi å plassere kulisser i et rutenett. Til det finnes klassen `GridPane` som selv er en kulisse. (Det er altså mulig å bygge kulisser på kulisser på kulisser . . .)



Å plassere ting i et rute nett

```

import javafx.application.Application;
import javafx.application.Platform;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.layout.GridPane;
import javafx.scene.control.Button;
import javafx.scene.text.Text;
import javafx.scene.text.Font;

/* Brettet:
+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+

Spillerne:
X - maskinen
O - brukeren
*/

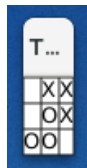
public class TTT0 extends Application {
    @Override
    public void start(Stage teater) {
        GridPane rutenett = new GridPane();
        rutenett.setGridLinesVisible(true);
        rutenett.add(new Text(" "), 0, 0);
        rutenett.add(new Text("X"), 1, 0);
        rutenett.add(new Text("X"), 2, 0);
        rutenett.add(new Text(" "), 0, 1);
        rutenett.add(new Text("O"), 1, 1);
        rutenett.add(new Text("X"), 2, 1);
        rutenett.add(new Text("O"), 0, 2);
        rutenett.add(new Text("O"), 1, 2);
        rutenett.add(new Text(" "), 2, 2);

        Pane kulisser = new Pane();
        kulisser.getChildren().add(rutenett);

        Scene scene = new Scene(kulisser);

        teater.setTitle("Tripp-trapp-tresko");
        teater.setScene(scene);
        teater.show();
    }
}

```



Hva hvis jeg trenger en fil?

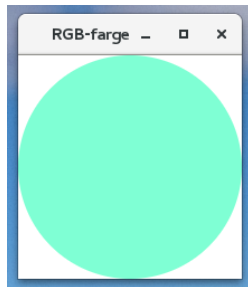
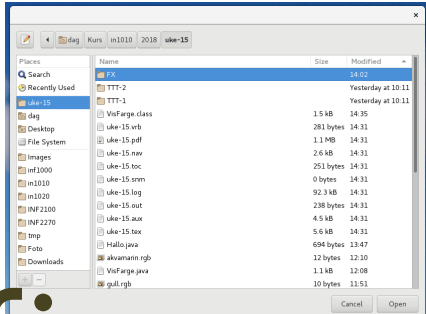
Å be om et filnavn

JavaFX kan også brukes til å be om et filnavn. La oss anta at vi har noen filer med farger angitt som RGB-verdier:

127 255 212

akvamarin.rgb

Så kan vi i programmet vårt be brukeren velge en slik fil:



Hva hvis jeg trenger en fil?

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.stage.FileChooser;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
```

```
import java.io.File;
import java.util.Scanner;
```

```
public class VisFarge extends Application {
    @Override
    public void start(Stage teater) {
        int r = 0, g = 0, b = 0;
        File file = new FileChooser().showOpenDialog(teater);
        try {
            Scanner f = new Scanner(file);
            r = f.nextInt(); g = f.nextInt(); b = f.nextInt();
        } catch (Exception e) {}

        Circle fargedemo = new Circle(100);
        fargedemo.setCenterX(100); fargedemo.setCenterY(100);
        fargedemo.setFill(Color.rgb(r,g,b));

        Pane kulisser = new Pane();
        kulisser.getChildren().add(fargedemo);

        Scene scene = new Scene(kulisser);
        teater.setTitle("RGB-farge");
        teater.setScene(scene);
        teater.show();
    }
}
```

Klassen FileChooser

Denne klassen gjør
det enkelt å be
brukeren om et
filnavn:

```
File file =
    new FileChooser().
        showOpenDialog(teater);
```



Til slutt

Husk

- JavaFX er et *enormt* bibliotek; det er lett å gå seg vill.
- Det er ofte mange måter å gjøre noe på, men ikke alle varianter passer sammen.
- Søking på internettet vil lett kunne gi kode som forvirrer mer enn den hjelper.
- Forelesningene og IN1010-introduksjonsheftet inneholder det man trenger å vite for å kunne lage enkle GUI-programmer, og løse oblig nr 6.
- Når du skal programmere GUI i IN1010, ta utgangspunkt i et program som finnes, og modifier det.

