

Kapitel 5: Ti_system-modulen

Övning 2: Mer om att utbyta data

I den här lektionen ska vi fortsätta titta på ti_system-modulen och hur vi kan använda den för få ett program att reagera på tangenttryckningar.

Mål:

- Detektera vilken tangent som tryckts ner.

Modulen **ti_system** innehåller även funktioner som kan läsa av tangentbordet.

Vi ska använda detta till att skapa ett program som gör att du med piltangenterna kan flytta en liten cirkelskiva över skärmen.

Starta ett nytt Python-program och importera ti_system-modulen och ti_draw-modulen. (Båda finns i menyn under Fler moduler.)



```
*Tangenten.py 3/4
from ti_system import *
from ti_draw import *
```

Vi börjar med en röd cirkelskiva mitt på skärmen.

Skärmen är 318 pixlar bred och 212 pixlar hög med ursprunget i det övre vänstra hörnet, så mitten på ritytan får då koordinaterna (159,106). Kommandot **fill_circle()** hittar du under meny>Fler moduler>Ti Draw>Form



```
*Tangenten.py 8/8
from ti_system import *
from ti_draw import *

x=159
y=106
set_color(255,0,0)
fill_circle(x,y,5)
```

Funktionen **get_key()** från ti_system-menyn kontrollerar om en tangent har tryckts in och i så fall är resultatet namnet på den tangent som har tryckts, så den returnerar med andra ord en sträng. Är ingen tangent nedtryckt så returnerar funktionen en tom sträng.

Vi ska lagra resultatet av get_key() i en variabel, låt oss ge den namnet tangent.

Vi börjar med att sätta variabeln till en tom sträng, tangent = "".

Därefter skapar vi en while-loop som i varje varv anropar get_key().

När man trycker esc-tangenten så ska loopen avslutas.



```
*Tangenten.py 11/12
from ti_system import *
from ti_draw import *

x=159
y=106
set_color(255,0,0)
fill_circle(x,y,5)

tangent=""
while tangent!="esc":
    tangent=get_key()
```

Nu ska vi lägga till kod som kontrollerar om en piltangent har tryckts ner, och i så fall ska motsvarande koordinat ändras så att cirkelskivan flyttas i motsvarande riktning.

```
1.2 1.3 1.4 *Dok RAD 18/21
Tangenter.py
while tangent != "esc":
    tangent=get_key()
    if tangent=="left":
        x=x-10
    if tangent=="right":
        x=x+10
    if tangent=="up":
        y=y-10
    if tangent=="down":
        y=y+10
    fill_circle(x,y,5)
```

Om du testar programmet så ser du att det kanske inte är helt så som vi tänkt oss. De gamla cirkelskivorna finns kvar. Dessutom kan vi få cirkelskivan att vandra utanför skärmen.

För att säkerställa att cirkelskivan inte hamnar utanför skärmen bör koordinaterna inte vara för stora eller för små.

Du kan göra detta med funktionerna **min()** och **max()**:

min(x, y) returnerar det mindre av de två talen och

max(x, y) returnerar det större värdet.

Så om x-koordinaten blir mindre än 5 (cirkelskivans radie) så sätts den till 5 så att cirkelskivan hamnar vid vänsterkanten. Vi gör på liknande sätt i de andra if-satserna.

```
1.2 1.3 1.4 *Dok RAD 19/21
Tangenter.py
while tangent != "esc":
    tangent=get_key()
    if tangent=="left":
        x=max(x-10,5)
    if tangent=="right":
        x=min(x+10,313)
    if tangent=="up":
        y=max(y-10,5)
    if tangent=="down":
        y=min(y+10,207)
    fill_circle(x,y,5)
```

Om vi bara vill se den senaste cirkelskivan så kan vi till exempel tömma skärmen med kommandot `clear()` innan vi ritar en ny cirkel.

```
1.2 1.3 1.4 *Dok RAD 20/22
Tangenter.py
tangent=get_key()
if tangent=="left":
    x=max(x-10,5)
if tangent=="right":
    x=min(x+10,313)
if tangent=="up":
    y=max(y-10,5)
if tangent=="down":
    y=min(y+10,207)
clear()
fill_circle(x,y,5)
```

Provar ni programmet med denna ändring så märker ni att skärmen flimrar hela tiden. Detta eftersom skärmen rensas och sedan ritas om även när ingen knapp trycks in.

Du kan förhindra detta genom att ge funktionen `get_key()` något in-värde, till exempel 1. Detta gör att `get_key()` funktionen väntar på en tangenttryckning innan den returnerar ett värde.

Detta gör att programmet väntar på att en tangent ska tryckas och loopen kommer därmed temporärt stoppas.



```
1.2 while tangent != "esc":
1.3     tangent=get_key(1)
1.4     if tangent=="left":
        x=max(x-10,5)
        if tangent=="right":
            x=min(x+10,313)
        if tangent=="up":
            y=max(y-10,5)
        if tangent=="down":
            y=min(y+10,207)
        clear()
```

(Vi skulle kunna lösa detta på andra sätt. Vi exempelvis kunna spara de gamla koordinaterna och bara anropa `clear` och `fill_circle` då positionen för cirkelskivan faktiskt har ändrat. (Vi kan testa det med en `if`-sats) Fördelen med detta är vi kan låta programmet göra annat då programmet väntar på tangenttryckningar. Nackdelen är mer komplicerad kod.)
