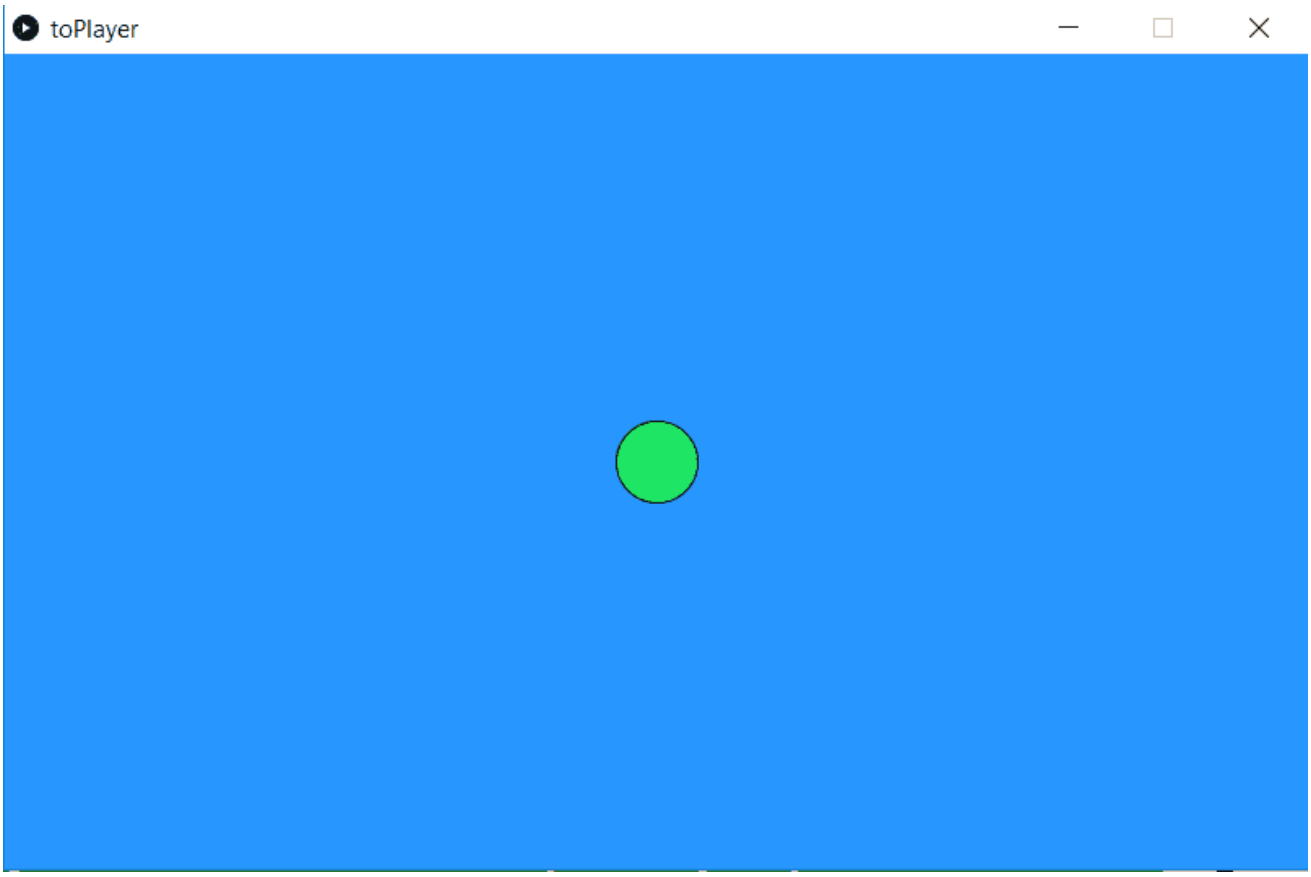
ToPlayer

PÅ BOKMÅL | LAST NED PDF

Introduksjon

No skal me lage eit spel som to personar kan spele mot kvarandre. Me har kalt det ToPlayer, men du kan kalle det det du vil. Målet er å dytte ein figur, eller ei spelebrikke, ut av vindauget på motstandaren si side. Spelet ser ganske enkelt ut, men du kan vidareutvikle det slik du vil. Når du har kome gjennom heile oppgåva skal spelet likne på det du ser i biletet under. Du får bestemme farger og litt anna sjølv, så det vil avhenge av kva du likar.



Steg 1: Kom i gang med metodane `setup` og `draw`

Start med å åpne Processing. Når du har gjort det må du skrive to metodar. Alle metodane har sitt eige namn, og det er viktig at to metodar ikkje kan ha same namn. Dei to metodane du skal skrive er `setup` og `draw`. Bortsett frå namnet, så skriv me dei på akkurat same måte. Difor viser me deg korleis du skal skrive `setup`-metoden, og så må du sjølv prøve å skrive `draw`-metoden.

Slik ser `setup`-metoden ut:

```
void setup(){

}
```

Sjekkliste

- [ ] Skriv `draw`-metoden under `setup`-metoden.

- [ ] Sjekk at metodane fungerer ved å køyre programmet. Klikk på pila oppe til venstre for å gjere det: 

Viss programmet køyrer og alt er i orden vil det dukke opp eit nytt bitte lite vindauge på skjermen din. Viss du fekk det opp kan du gå vidare til steg 2. Viss ikkje, så kan du sjekke at du har fått med alt i koden din ved å samanlikne med koden under.

```
void setup(){

}

void draw(){

}
```

Lagre koden din før du går vidare.

Forklaring av setup og draw

Alt innhaldet i `setup` skjer ein gong når programmet startar. Så går programmet vidare til `draw`. Alt innhaldet i `draw` skjer på nytt og på nytt heilt til programmet stoppar.

Steg 2: Bestem storleik og bakgrunnsfarge på vindauget

No er det storleiken på vindauget og bakgrunnsfarga som skal kodast. For å gjere det treng me berre to kodelinjer. Begge skrivast inn i `setup`-metoden. Å skrive kode inni ein metode vil seie at me skriv mellom krøllparentesane, altså mellom `{` og `}`.

Storleiken blir bestemt ved å skrive denne kodelinja:

```
size(200, 600);
```

Tala mellom parentesane bestemmer storleiken på vindauget.

Koden som bestemmer bakgrunnsfarga ser slik ut:

```
background(40, 150, 255);
```

Tala inni parantesane bestemmer farga.

Her er ei sjekkliste med ting du kan gjere for å sjekke at du forstår koden din. For kvart punkt må du starte programmet og sjå kva som endrar seg. Det er viktig at du forstår koden din før du går vidare til neste steg.

Sjekkliste

- [ ] Gjer det fyrste talet i parentesane til `size` dobbelt så stort.
- [ ] Prøv å få vindauget til å dekke heile skjermen din.
- [ ] Set tala inni parentesane til `background` til å vere `(255, 0, 0)`.
- [ ] Sett tala inni parentesane til `background` til å vere `(0, 255, 0)`.
- [ ] Sett tala inni parentesane til `background` til å vere `(0, 0, 255)`.
- [ ] Sett tala inni parentesane til `background` til å vere `(0, 255, 255)`.
- [ ] Set alle tala inni parentesane til `background` til å vere 0.
- [ ] Prøv å få bakgrunnen til å bli kvit.
- [ ] Prøv å få bakgrunnen til å bli lilla.
- [ ] Finn ein storleik du likar på vindauget ditt.
- [ ] Finn ei bakgrunnsfarge på vindauget ditt som du likar.

Køyr programmet ditt og sjekk at du får vindauget til å visast før du går vidare. Her er koden me har laga så langt, hugs at tala dine kanskje er litt forskjellige frå våre tal.

```
void setup(){
  size(800, 500);
  background(40, 150, 255);
}

void draw(){

}
```

Steg 3: Spelebrikke

Me skal lage spelebrikka, og så skal me gjere slik at den kan bevege seg.

Her er koden som skal brukast:

```
fill(30, 230, 100);
ellipse(400, 250, 50, 50);
```

Sjekkliste

- [ ] Skriv inn dei to kodelinjene i `draw`-metoden og test at programmet køyrer.
- [ ] Byt ut eitt og eitt av tala inni parentesen til `ellipse` og finn ut kva dei gjer.
- [ ] Byt ut eitt og eitt av tala inni parentesen til `fill` og finn ut kva dei gjer.
- [ ] Prøv å få spelebrikka til å bli rosa.
- [ ] Prøv å få spelebrikka til å bli så stor at den dekker heile vindauget.
- [ ] Still tilbake storleiken på spelebrikka og vel ei farge du likar.

No skal me gjere slik at spelebrikka er klar til å bevege seg sidelengs. Som du sikkert har funne ut, så bestemmer det fyrste talet i parentesen kor spelebrikka står plassert sidelengs. Me vil gjere dette talet til ein variabel, altså til eit tal som kan variere.

For å få det til deklarerer me variabelen heilt i toppen av programmet vårt (utanfor `setup`- og `draw`-metodane). Det gjer me ved å skrive denne kodelinja:

```
int x;
```

Denne vesle koden er ganske enkel. Med `int` fortel me PC-en at me skal ha eit heiltal, og `x` er namnet me har valt å gi variabelen vår. Me kunne kalla den noko heilt anna, til dømes `pute` eller `boks`, men her gjer `x` nytta.

Me må gi `x` ein verdi som den har til å starte med, det gjer me inne i `setup`-metoden. Alt me treng å skrive er dette:

```
x = 90;
```

Så må me bytte ut det fyrste talet i parentesen til `ellipse` med `x`. Då ser koden for ellipse slik ut:

```
ellipse(x, 250, 50, 50);
```

Sjekkliste

- [ ] Skriv inn dei tre kodelinjene som vart forklart over. Pass på at du plasserer dei på riktig stad i koden!
- [ ] Test at programmet ditt køyrer. Det skal ikkje skje noko nytt, bortsett frå at spelebrikka kanskje har flytta litt på seg.
- [ ] Bytt ut verdien til `x` med til dømes `180` og sjå at spelebrikka flyttar seg.
- [ ] Sjå om du får til å plassere spelebrikka så langt ut til høgre at du berre kan sjå halvparten av den.
- [ ] Plasser spelebrikka midt på skjermen sidevegs.

Her er koden me har laga så langt.

```
int x;

void setup(){
  size(800, 500);
  x = 400;
}

void draw(){
  background(40, 150, 255);
  fill(30, 230, 100);
  ellipse(x, 250, 50, 50);
}
```

Steg 4: Lag kontrollarar!

No skal me lage ei `if`-setning som registrerer om du trykkar på ein knapp.

Skriv denne koden inni `draw`-metoden etter spelebrikka:

```
if(keyPressed && key == 'a'){
  x+=10;
}
```

Denne koden sjekkar om me har trykka ein knapp, og at den knappen er `a`. Viss begge desse vilkåra er sanne, så skal `x`, altså plasseringa til spelebrikka vår, endre seg med 10 pikslar mot høgre.

Forklaring av if

Ei `if`-setning sjekkar om noko er sant. Viss det er sant, så skjer det som står inni `if`-setninga. Ei `if`-setning er bygd opp av ein test og noko som skal bli utført:

```
if(test){
  dette skjer berre viss testen er sann.
}
```

`If`-testar er enkle å setje opp som munnlege setningar, og så gjere dei om til kode etterpå. Me skriv det på same måte som ei ordentleg `if`-setning, slik at det er lett å gjere om til kode. Inne i `if`-setninga har me kome med nokre døme.

```
Viss ( eitt eller anna er sant) så skal dette skje{
  me får meir liv
  me hoppar,
  noko flyttar på seg,
}
```

Det er veldig vanleg å bruke ei `if`-setning til å sjekke om ein spesiell knapp er trykka på, eller om musepeikaren er innanfor eit bestemt område. Eller ein kan sjekke kor mange liv spelaren har att, og viss det er 0 kan spelet si "GAME OVER".

Når me skriv if-setninga vår med ord blir det slik:

```
Viss (ein trykkar ein knapp og knappen er 'a') så skal dette skje{
  spelebrikka skal bevege seg til høgre;
}
```

Me prøver å gjere setninga meir lik kode

```
viss ( einKnappErTrykka og knappen er lik 'a'){
  så skal x bli 10 større;
}
```

Så gjer me det til kode:

```
if(keyPressed && key == 'a'){
  x+=10;
}
```

Sjekkliste

- [ ] Skriv inn if-setninga i `draw`-metoden.

- [ ] Køyr programmet og trykk på `a`. Kva skjer?
- [ ] Flytt kodelinja som set bakgrunnsfarga frå `setup`-metoden til `draw`-metoden. Set den øvst i `draw`.
- [ ] Køyr programmet på nytt og sjå kva som skjer.
- [ ] Bytt ut 10-talet med eit anna tal.
- [ ] Bytt ut `a` med ein annan bokstav.
- [ ] Flytt kodelinja som set bakgrunnsfarga fram og tilbake mellom `setup` og `draw` nokre gonger og sjå om du forstår kva som skjer.
- [ ] Prøv å skrive ei `if`-setning til. Denne skal stå under den førre, og no kan du velje ein annan bokstav. I staden for at `x` blir større, så må du skrive koden slik at `x` blir mindre. Viss det er litt vanskeleg, så kan du prøve å setje opp `if`-setninga som ei vanleg setning slik me gjorde i stad.
- [ ] Bytt ut talet inni den nye `if`-setninga slik at du får spelebrikka til å gå dobbelt så raskt til venstre.

Viss koden din ikkje fungerer heilt slik den skal, så kan du sjekke her kva som er feil:

```
int x;

void setup(){
  size(800, 500);
  x = 400;
}

void draw(){
  background(40, 150, 255);
  fill(30, 230, 100);
  ellipse(x, 250, 50, 50);

  if(keyPressed && key == 'a'){
    x+=10;
  }

  if(keyPressed && key == 'l'){
    x-=10;
  }
}
```

Steg 5: Me treng ein vinnar!

Me treng fleire `if`-setningar. Desse skal skrive ein beskjed på skjermen viss ein spelar vinn. Ein vinn ved å dytte spelebrikka ut av skjermen.

Her er `if`-setninga du treng for å sjekke om venstre spelar vann. Den skal stå under dei to førre `if`-setningane.

```
if(x > 800){
  text("Venstre spelar vann!", 350, 200);
}
```

Sjekkliste

- [ ] Skriv inn den nye `if`-setninga.
- [ ] Sjekk at programmet fungerer ved å få spelebrikka ut av skjermen på høgre side. Då skal teksten visast.
- [ ] Kva trur du skjer viss du endrar `800`-talet som står inni `(x > 800)` til eit anna tal?
- [ ] Bytt ut `800` med noko mykje mindre og finn ut kva som skjer.
- [ ] Kva skjer viss du forandrar teksten til "Høgre spelar vann!"?
- [ ] Finn ut kva tala som står inni parentesane til `text` gjer ved å bytte dei ut.
- [ ] Gjer storleiken på vindauget ditt større og sjå korleis det påverkar `if`-setninga.

- [ ] Still tilbake alle tala, slik at teksten dukkar opp på skjermen berre når spelebrikka er ute av skjermen.
- [ ] Lag ei ny `if`-setning som gjer nesten akkurat det same, men som heller sjekkar om spelebrikka går ut av skjermen på venstre side.
- [ ] Test at koden din fungerer.

Her er heile koden, sjekk om din er lik.

```
int x;

void setup(){
  size(800, 500);
  x = 400;
}

void draw(){
  background(40, 150, 255);
  fill(30, 230, 100);
  ellipse(x, 250, 50, 50);

  if(keyPressed && key == 'a'){
    x+=10;
  }

  if(keyPressed && key == 'l'){
    x-=10;
  }

  if(x > 800){
    text("Venstre spelar vann!", 350, 200);
  }

  if(x < 0){
    text("Høgre spelar vann!", 350, 200);
  }
}
```