

Programmeringsoppgave – Pong

I denne oppgaven skal dere lage deres egen versjon av spillklassikeren Pong (<http://en.wikipedia.org/wiki/Pong>). Pong er et veldig enkelt "tennisspill" for to spillere. Spillbrettet består av en ball og to rektangler som hver spiller kan styre opp og ned i hver sin side av banen. Poenget med spillet er enkelt og greit å treffe ballen og å få motstanderen til å bomme på ballen.

For å lage dette spillet er det ganske mye vi må holde styr på. Vi må ha variabler som holder posisjonen og hastigheten til ballen, og variabler som holder posisjonen og hastigheten til spillerbrikkene. I tillegg trenger vi variable for å holde styr på poengene til hver spiller, og en del til. For at dette ikke skal bli alt for vanskelig skal vi gå igjennom programmet steg for steg. (Etter hvert punkt kan det være lurt å kjøre programmet for å se om det er noen alvorlige feil i koden).

1. Opprett en ny mappe som du kaller f.eks "Pongspill", her lagrer du filene "bat.png" og "ball.jpg" som du finner på www.olemarius.net. Der finner du også et programskjelett hvor du skal lage ditt program. Lagre fila "Pong.bus" i den SAMME filmappa som du lagret de andre 2 filene.
2. Som sagt så trengs det mange variable for å få programmet til å fungere. For å få en bedre oversikt over koden er det lurt å ha alle disse variablene øverst i programmet. I tillegg til variablene som nevnes i 2 a, b og c vil du hvertfall trenge disse variablene: *ballX*, *ballY*, *player1X*, *player1Y*, *player2X*, *player2Y*, *player1Score*, *player2Score*, *batSpeed*, *ballSpeedX*, *ballSpeedY*. Til å begynne med kan du sette alle disse variablene til 0.
 - a) Du vil trenge 4 taster på tastaturet for å styre spillet, 2 for hver spiller, opp og ned. Bestem deg for hvilke 4 taster du vil styre spillet med og bruk eksempelprogrammet "tastaturkodeprinter" til å finne koden til hver av disse tastene. Lag 4 variable som lagrer disse kodene.
 - b) Lag 2 variable "ballImg" og "batImg" som holde bildet av ballen (ball.jpg) og av spilleren (bat.png). Du må bruke funksjonen loadImage() for å få til dette.
 - c) For å kunne på bevege på ballen og spillerene så må du konvertere variablene "ballImg" og "batImg" til "Sprite", dette gjøres med funksjonen toSprite(). Du trenger en sprite for ballen og en for hver av spillerene (finn på enkle og fornuftige variabelnavn til de nye spritene, som f.eks *ball*, *player1* og *player2*).
3. Lag funksjonen "startRound()". Denne funksjonen er den som skal kjøres hver gang en ny omgang startes. Her må vi sette startposisjonen til ballen (*ballX* og *ballY*), startposisjonen til spillerbrikkene (*player1X*, *player1Y* og *player2X* og *player2Y*). Du må å sørge for at spiller og ball spritene er synlige. For å få til dette så trenger du funksjonene *spriteMove()*, *height()*, *width()* og *spriteMove()*.
4. Lag funksjonene "player1Move()" og "player2Move()". Disse funksjonene skal vi bruke til å flytte spillerbrikkene. Her må vi sjekke om vi holder nede knappene for å flytte brikkene, og forandre på posisjonen til disse om vi gjør det. Til dette må vi bruke if-tester og funksjonen *keyPressed()* i tillegg til *spriteMove()*. Du kan nå sette verdien i variabelen *batSpeed* til den hastigheten du vil at spilleren skal kunne bevege seg. Hastigheten måles i antall piksler pr. tastetrykk.
5. Lag funksjonen "goal()". Denne funksjonen skal kjøres hver gang en av spillerene scorer mål. Den tar inn et tall som argument (det som står inne i parentes), dette tallet representerer hvilken spiller

det er som har scoret. Om spiller 1 scorer pluss du på et poeng på spiller 1 sin poengscore og det samme for spiller 2. Deretter bruker du `println()` til å skrive ut stillingen. (Her kan det hende at du får bruk for funksjonen `cls()` ...)

6. Lag funksjonen `moveBall`

Dette er den mest omfattende funksjonen i spillet og det er denne som skal flytte ballen. Ballens x-posisjon og ballens y-posisjon må oppdateres for at det skal bli noe bevegelse i spillet. Du bestemmer ballens hastighet ved å forandre verdien til variablene *ballSpeedX* og *ballSpeedY*. Her er det en ekstra nøtt og finne ut hvordan du kan bruke *ballSpeedX* og *ballSpeedY* til å huske hvilken retning ballen beveger seg i. Hvis ikke kan det hende at du trenger en ekstra variabel eller to for å huske på ballens retning.

I tillegg må vi sjekke om ballen treffer noen av ytterkantene. Om ballen treffer bunnen eller toppen av skjermen må ballen sprette av disse. Og treffer ballen høyre eller venstre side av skjermen betyr det at en av spillerne har scoretmål. Da må goal funksjonenn kjøres. Til slutt må vi også sjekke om ballen treffer en av spillerene så den skal sendes tilbake.

For å få til dette trenger du forskjellige funksjoner som vi har brukt tidligere og den nye funksjonen `spriteCollision()`.

Spillet ditt burde nå være klart til å spilles. Men det er kanskje noen småting til du burde tenke på, hva skjer f.eks hvis en spiller holder opp-knappen inne veldig lenge?...

Ekstra utfordringer:

- Gjør om programmet slik at ballens hastighet og retning forandrer seg ut i fra hvordan spilleren treffer den.
- Legg til en start/stopp/pause knapp
- Legg til 2 ekstra spillere slik at det er spillere på alle fire kanter av banen