

## i Informasjon om eksamen



# Høgskolen i Østfold

## EKSAMEN

**Emnekode og -navn:** ITF19019 Innføring i programmering

**Dato og tid:** 29.11.2019, 4 timer

**Fagansvarlig:**

Tore Marius Akerbæk

Joakim Karlsen

**Hjelpemidler:**

Fire egenproduserte A4-sider.

**Om oppgaven:**

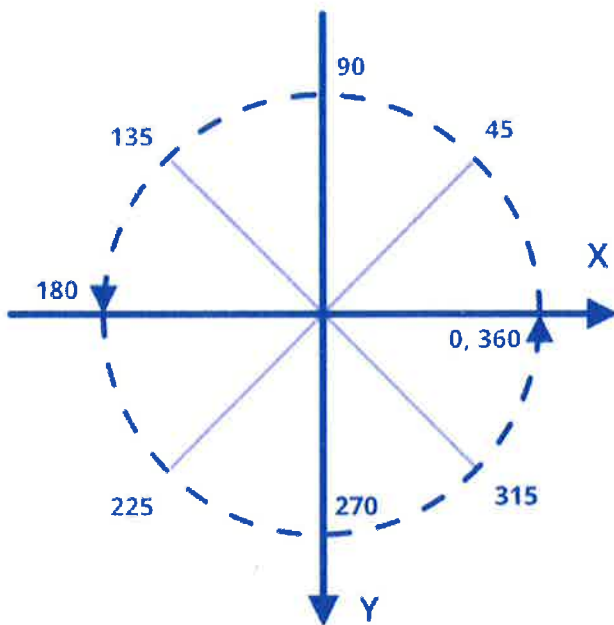
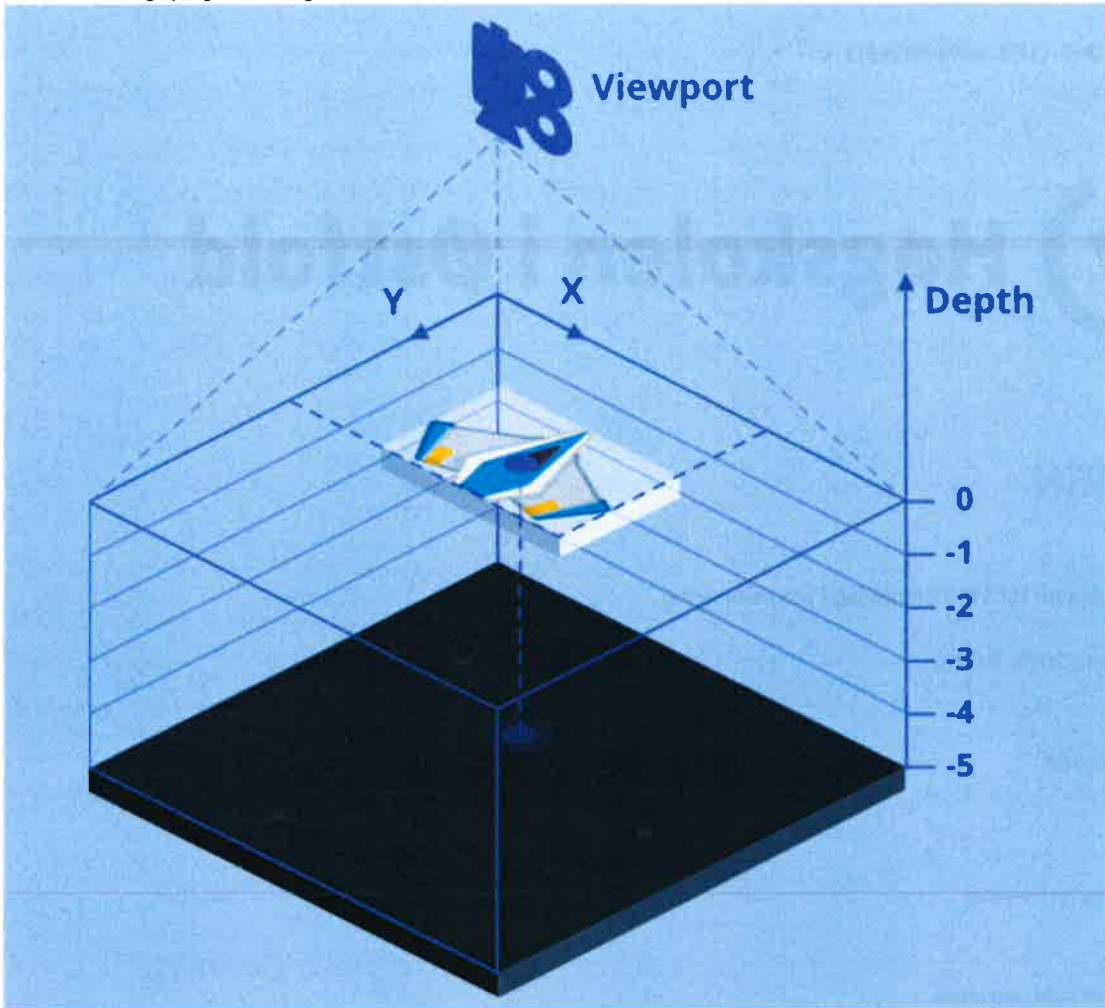
Alle spørsmålene skal besvares.

**Sensurfrist: 20.12.2019**

Resultatene blir publisert i Studentweb.

## i Introduksjon

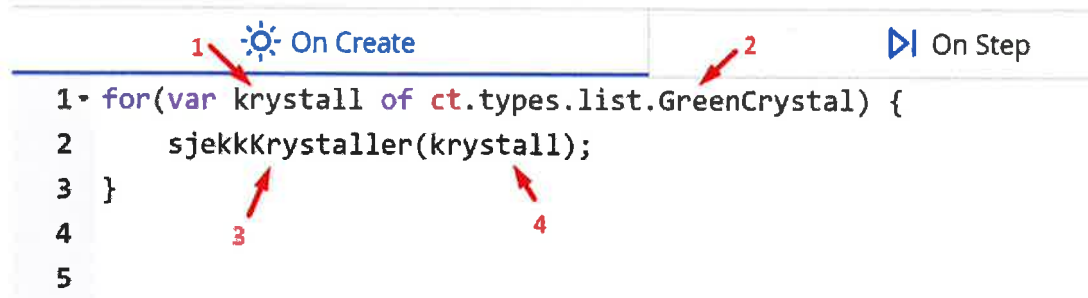
Til hjelp: i CT.js og oppgavene i eksamen opererer vi med disse forutsetningene for størrelse/aksediagram og retning.



## i Seksjon 1

Denne oppgaven går på kodelesing og -forståelse. Du blir presentert for noen kodesnutter (og dersom nødvendig, kontekst) som du skal redegjøre for hva gjør eller resulterer i.

### 1 1.1



```
1 for(var krystall of ct.types.list.GreenCrystal) {
2     sjekkKrystaller(krystall);
3 }
4
5
```

På bildet over vises en liten kodesnutt som inneholder fire grunnleggende prinsipper innenfor programmering. Navngi disse fire prinsippene.

Skriv ditt svar her...

## 2 1.2

```
1 • if(ct.place.occupied(this, this.x, this.y + 1, 'Solid')) {
2 •     if(ct.actions.Jump.down) {
3 •         this.vspd = this.jumpSpeed;
4 •     } else {
5 •         this.vspd = 0;
6 •     }
7 • } else {
8 •     this.vspd += this.gravity * ct.delta;
9 • }
```

Forklar, linje for linje, hva koden på bildet over gjør. Forutsetningene for koden er at det er laget en Action i CT.js kalt Jump som er knyttet til tastaturknappen Space, og at this.jumpSpeed er deklartert på typen ved opprettelse (on Create).

**Skriv ditt svar her...**

## 3 1.3

```
1 • if(ct.place.meet(this, this.x, this.y, 'Robot')) {
2 •     ct.room.crystals++;
3 •     this.kill = true;
4 • }
```

Forklar, linje for linje, hva koden på bildet over gjør. Beskriv også hvilke forutsetninger som ligger til grunn for at koden skal fungere (avhengigheter til andre typer, deklarte variabler o.l.).

**Skriv ditt svar her...**

## 4 1.4

```

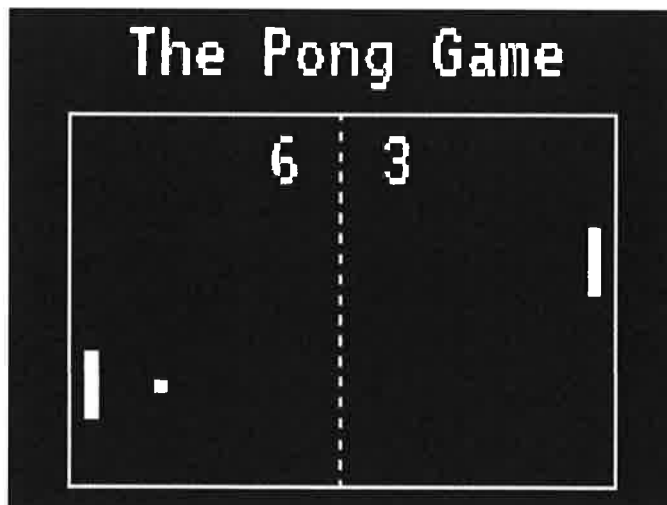
1 for(var i = 0; i < 5; i++) {
2     rcx = ct.radnom.range(0, ct.viewWidth);
3     rcy = ct.random.range(0, ct.viewHeight);
4
5     if(ct.place.free('GreenCrystal', rcx, rcy, 'Solid')) {
6         ct.types.copy('GreenCrystal', rcx, rcy)
7     }
8 }

```

I koden på bildet over er det seks feil som gjør at koden ikke vil fungere. Finn og beskriv feilene.

Skriv ditt svar her...

## i Introduksjon



I seksjon to blir du presentert for noen utfordringer som skal løses ved hjelp av programmering. I svarene trenger du ikke skrive programkode, men planlegge og forklare hvordan utfordringene kan løses ved å bryte den ned i delproblemer med egne løsninger.

I bildet over ser vi spillklassikeren Pong. Ikke ulikt bordtennis (ping pong) består spillet av to spillere mot hverandre med hver sin padel/racket, og en ball. Ballen spretter mot toppen/bunnen av spillbrettet, og det er om å gjøre å slå ballen mot den andre spilleren. Hvis man ikke treffer ballen med padelen/racketen sin, forsvinner ballen ut av spillbrettet og spilleren som slo ballen sist får et poeng.

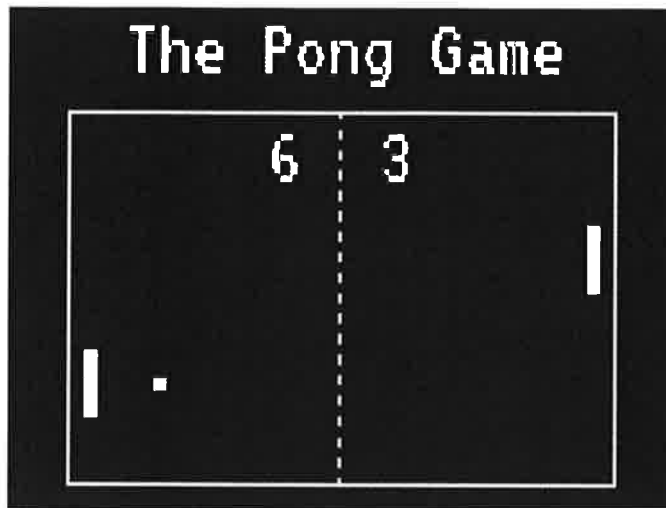
5 2.1



Ta for deg utfordringen med spillere og deres bevegelser. Beskriv utfordringer/funksjoner knyttet til disse, og bryt dem ned for å finne potensielle løsninger.

**Skriv ditt svar her...**

## 6 2.2



Ta for deg utfordringen med ballen og dens bevegelser. Beskriv utfordringer/funksjoner knyttet til dette, og bryt dem ned for å finne potensielle løsninger (Ballens vinkler skal diskuteres, men det forventes ikke at den matematiske utregningen redegjøres for).

**Skriv ditt svar her...**

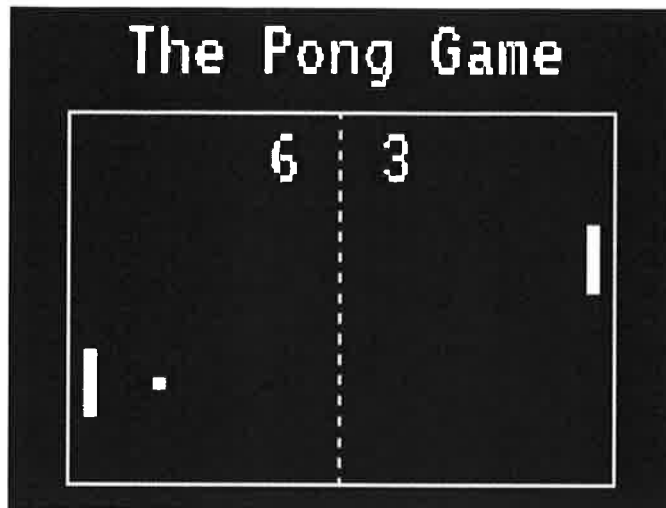
7 2.3



Ta for deg poengsystemet. Beskriv regler og oppførsel spillet må håndtere for å få et fungerende poengsystem.  
**Skriv ditt svar her...**



## 8 2.4



Hvis spillet skulle lages, hvilke typer (i CT.js) / objekter måtte vi ha tilgjengelig for å kunne lage spillet? PS: Husk at utskriften av poengene også regnes som typer.

**Skriv ditt svar her...**

**i Introduksjon**

I oppgave tre skal du selv kode noen løsninger for noen gitte utfordringer. Du kan her presentere løsningene som kode i CT.js eller ren JavaScript. Dersom du er usikker på syntax, bruk pseudokode og kommenter underveis i koden for å tydeliggjøre hva du ønsker å oppnå med koden.

**Oppgavens grunnlag**

Vi skal lage et labyrintspill. Nivå 1 er et spillbrett som er 600 x 600 pixler stort, og heter «Level1». Vi har en type kalt «Blokk» som illustrerer steiner på 30 x 30 pixler, som en type kalt «Spiller» ikke skal kunne gå gjennom (collision type «Solid»). «Spiller» starter nede i venstre hjørne av spillbrettet, og en type kalt «Utgang» (døren til neste nivå) er plassert øverst til høyre.

## 9 3.1

Skriv et forslag til kode som sørger for at typen «Spiller» ikke beveger seg på utsiden av spillbrettet.

**Skriv ditt svar her...**

10 **3.2**

Skriv et forslag til kode som plasserer 20 kopier av typen «Blokk» vilkårlig rundt på spillbrettet.

**Skriv ditt svar her...**

11 **3.3**

Skriv et forslag til kode som åpner Nivå 2 når «Spiller» kommer til «Utgang». PS: CT-funksjonen for å bytte rom er `ct.room.switch('Nytt rom')`.

**Skriv ditt svar her...**

12 **3.4**

Lag to arrayer. Begge arrayene skal inneholde fem verdier innenfor spillbrettets størrelse.

Lag en kode som løper gjennom begge arrayene og plasserer en kopi av «Blokk» for hvert koordinatpar. Et koordinatpar er satt sammen av en verdi fra hver av arrayene.

**Skriv ditt svar her...**