

# Primitivt liv: Herbivore og Planter

## Tema 1



### 0 Indholdsfortegnelse

<b>1 Beskrivelse.....</b>	<b>2</b>
<b>2 Indeholdte input-filer .....</b>	<b>2</b>
<b>3 Kravoversigt .....</b>	<b>3</b>
<b>4 Anbefalet fremgangsmåde.....</b>	<b>3</b>
<b>5 Rettelser fra biologen.....</b>	<b>4</b>

Dette dokument indeholder en beskrivelse af det første tema for projektet i grundlæggende programmering. Inden du giver dig i kast med opgaverne, er det vigtigt, at du har fået basisprojektet op at køre. Dette beskrives i dokumentet **ITU Simulator – beskrivelse af bibliotek**.

Dette dokument er opbygget således at **1 Beskrivelse** giver et indblik i hvad vores "miniverden" skal kunne understøtte når temaet er gennemført. **2 Indeholdte input-filer** viser de filer, programmet skal kunne indlæse når temaet er gennemført. **3 Kravoversigt** giver et overblik over diverse krav, der skal understøttes. **4 Anbefalet fremgangsmåde** nedbryder diverse krav og giver et bud på, hvordan I kan angribe temaet fra start til slut.

## 1 Beskrivelse

Naturen er et vildt sted på mange måder. Der lever en masse forskellige slags dyr, i mange forskellige habitater – nogle er golde, andre er frugtbare og blomstrer med græs og planter. Vi tager i første tema udgangspunkt i det søde og simple; grønt græs og friske kaniner. I vores lille verden lever planter og kaniner i harmoni; kaninerne spiser lidt græs her og der, får en masse søde unger. Samtidigt trives græs symbiotisk<sup>1</sup> i denne lille begrænsede verden. Kaninerne nyder at de frit kan grave huller og tunneller, der spreder sig over vores lille firkantede plet. Et stille og roligt kredsløb, hvor der er ro om natten, for der sover kaninerne i deres små huller og tunneler; men hvem ved, hvor de titter frem næste morgen?

## 2 Indeholdte input-filer

```
t1-1a.txt  
t1-1b.txt  
t1-1c.txt  
t1-2a.txt  
t1-2b.txt  
t1-2cde.txt  
t1-2fg.txt  
t1-3a.txt  
t1-3b.txt  
tf1-1.txt
```

Alle input filer har den følgende form:

```
N  
type antal  
type min-max  
...
```

N angiver størrelsen på verdenen. Type er typen af aktør / element, der skal tilføjes til verdenen, f.eks. en kanin (anføres 'rabbit'). Derefter følger enten ét tal som beskriver **præcist** antallet der skal tilføjes til verdenen, eller to tal (med en bindestreg imellem) som giver et interval for, hvor mange der skal tilføjes (altså minimum og maksimum). Her bør der tilføjes et tilfældigt antal af typen (imellem intervallet).

Vi anbefaler man sætter de forskellige input-filer ind i mappen 'data' i det udleverede projekt.

### 3 Kravoversigt

Når følgende krav er opfyldt, er tema 1 gennemført. Alle krav og input-filer har et ID for at hjælpe på vej. Således kan man f.eks. med fordel bruge input-filerne **t1-2cde** når man løser krav **K1-2c**, **K1-2d**, og **K1-2e**, altså når tallet efter bindestreg er det samme. I første tema er kravene grupperet i emner. Dette vil ikke nødvendigvis være tilfældet senere. Uover de obligatoriske krav, følger *frivillige* krav nederst i kravoversigten.

#### Græs

- **K1-1a.** Græs kan blive plantet når input filerne beskriver dette. Græs skal blot tilfældigt placeres.
- **K1-1b.** Græs kan sprede sig.
- **k1-1c.** Dyr kan stå på græs uden der sker noget med græsset (Her kan interfacet *NonBlocking* udnyttes).

#### Kaniner

- **K1-2a.** Kaniner kan placeres på kortet når input filerne beskriver dette. Kaniner skal blot tilfældigt placeres.
- **k1-2b.** Kaniner kan dø, hvilket resulterer i at de fjernes fra verdenen.
- **K1-2c.** Kaniner lever af græs som de spiser i løbet af dagen, uden mad dør en kanin.
- **k1-2d.** Kaniners alder bestemmer hvor meget energi de har.
- **K1-2e.** Kaniner kan reproducere.
- **k1-2f.** Kaniner kan grave huller, eller dele eksisterende huller med andre kaniner. Kaniner kan kun være knyttet til et hul.
- **K1-2g.** Kaniner **søger mod** deres huller<sup>2</sup> når det bliver aften, hvor de sover. Det er ikke et krav at kaniner når hen til deres huller.

#### Kaninhuller

- **K1-3a.** Huller kan enten blive indsat når input filerne beskriver dette, eller graves af kaniner. Huller bliver tilfældigt placeret når de indgår i en input fil.
- **K1-3b.** Dyr kan stå på et kaninhul uden der sker noget.

#### *Frivillige, ikke obligatoriske, krav.*

- **KF1-1.** Huller består altid minimum af en indgang, der kan dog være flere indgange som sammen former én kanin tunnel. Kaniner kan kun grave nye udgange mens de er i deres huller.

### 4 Anbefalet fremgangsmåde

Inden du giver dig i kast med opgaverne er det vigtigt at du har fået basisprojektet op at køre. Dette beskrives i dokumentet **ITU Simulator – beskrivelse af bibliotek**. Herefter er det **meget** vigtigt at du danner dig et overblik over metoderne der hører til [\*\*World klassen\*\*](#). Specifikt er det vigtigt at kende og forstå: *delete*, *remove*, *getSurroundingTiles*, *getSurroundingEmptyTiles*, *isNight*, *isDay*, *move*, *containsNonBlocking*, *isTileEmpty* og *setTile*. Dertil bør man forstå hvordan *NonBlocking* interfacet fungerer.

Herefter anbefaler vi at du læser de **gode råd i projektbeskrivelsen**. Disse råd er altid gode at følge uanset hvor i processen man er. Dernæst anbefales det at man udfører en objektorienteret (OO) analyse som set i undervisningsgang 19. Her kan man med fordel tage udgangspunkt i **0 beskrivelse, 1 Indeholdte input filer** (selve fil-indholdet), samt **2 krav oversigten**. Ved at kigge på teksten, input-filernes indhold, samt de forskellige krav, kan I danne et overblik over hvilke komponenter der skal bygges, og hvilken funktionalitet de skal have. Her kan I også begynde at overveje potentielle sammenhænge mellem klasser (f.eks. behovet for abstrakte klasser, interfaces, hjælpe klasser med statiske funktioner, mv.). **Dette ville være gode elementer i dagbogen del omkring temaet.**

Som det fremgår af projektbeskrivelsen, er det ikke alle krav der beskriver *præcist* hvordan noget skal implementeres. Det er her op til jer som gruppe at bestemme dette. F.eks., skal en kanin kunne spise græs når den står ved siden af det, eller oven på? Husk at dokumentere jeres valg, eventuelt med kommentarer, og hvis I føler valgene er meget centrale, kunne dette også være relevant i første afsnit af dagbogen eller under designvalg. Oftest kan de forskellige krav løses på flere måder, og nogle løsninger vil være lettere at gøre an end andre. Snak sammen, og vælg ikke nødvendigvis det som er sværest for jer selv.

Nu har I således en rå skitse for hvad der skal bygges, men før I kaster jer ud i at programmere, bør I først overveje – er der noget af programmet I ønsker at refaktorisere, altså lave om? Dette er ikke nødvendigvis tilfældet i første tema, men vil blive utroligt vigtigt for de kommende temae, da vi bygger videre på systemet. **Dette kan også være gode elementer i rapportens del omkring temaet.**

Dertil kan I overveje; er der nogle krav som ikke giver specielt god mening? Tag en beslutning om hvorfor og hvordan I fortolker dem, og reflekter over dette. **Dette er også godt indhold til dagbogen og rapporten.**

Herefter kan I kaste jer ud i at implementere og teste de forskellige dele. **Med tests henvises der til den form for tests kurset underviser i (unit tests). Test mens I implementerer.** Kravene er sorteret (for hver gruppe af krav) således at de første krav er dem man bør starte med at implementere og teste. Der vil dog være krav som kræver at man har implementeret dele af andre krav "grupper". Her kan man igen kigge mod de **gode råd i projektbeskrivelsen**, hvis de er for uoverskelige til at starte med.

## 5 Rettelser fra biologen

<sup>1</sup> I praksis er der ikke tale om symbiose. Ved tale om plantespisere vil det ikke være symbiose eller mutualisme, medmindre at kaninerne f.eks. prutter kernerne fra græsset ud et andet sted (hvilket de ikke som sådan gør). Symbiose beskriver en eller flere arter der ikke er i stand til at leve uden den anden.

<sup>2</sup> I praksis vil man nok ikke kalde det huller, men i højere grad en 'rede'.

