

Manual

for

PicturesInYourHead

Version 0.4

Status

Dato	Hvem	Hvad
25-01-11	Ssn	Oprettet, kommandooversigt, indledning, ord er rimelige, andre afsnit er skitser
26-01-11	Ssn	Indholdsfortegnelse
27-01-11	Ssn	Kulisser og væsener med
29-01-11	Ssn	Filtre, verden startes verdenskort og tilstandskort er påbegyndt
30-01-11	Ssn	Mindre rettelser
13-02-11	Ssn	Verden startes lavet
24-02-11	Ssn	\$bgSound/\$nyBoks + _default boks + filtre
18-03-11	Ssn	Tekstflytninger mellem manual og byggevejledninger
22-03-11	Ssn	Illustrationer
20-04-11	Ssn	Dokumentations synkronisering
28-04-11	Ssn	Mindre korrektioner (ikke udgivet)
12-06-11	Ssn	Ny illu kommando og sætstykker
13-07-11	Ssn	Koordineret med wizzard
10-08-11	Ssn	Dokumentations synkronisering
21-08-11	Ssn	Goal som stak, Har og Ting kommando, SE, Flyt-mangler
03-09-11	Ssn	Opdateret til tool version 0.4
20-09-11	Ssn	Stavecheck
01-11-11	Ssn	Opdateret til toolversion 0.4.01, kun mindre rettelser

Indholdsfortegnelse

Indledning.....	4	Kommando: illu	33
Oversigt over alle dokumenter.....	4	Kommando: vent.....	36
Hændelsesbeskrivelser.....	5	Kommando: anbring.....	37
Sammensatte Processbeskrivelser.....	6	Kommando: fjern.....	38
Filtre.....	7	Kommando: sæt	39
Hvilke filtre prøves og i hvilken		Kommando: målEr.....	40
rækkefølge.....	8	Planlagte kommandoer	41
Kulisser	10	Kommando: kan.....	42
Specielle effekter.....	11	Kommando Har og ting.....	43
\$ Tegnet.....	11	Ord.....	44
# Tegnet.....	11	Avataren.....	44
Tegnet.....	12	Opdageren.....	44
Kommandoer.....	14	Hændelse.....	44
Kommandofelter.....	14	Autohændelse	44
Kommando: flyt.....	15	Hændelsesbeskrivelse.....	44
Kommando: vis.....	16	Processbeskrivelsen.....	44
Kommando: se.....	17	Kommando.....	44
Kommando: læg.....	19	Filter.....	45
Kommando: ting.....	20	Aktive boks.....	45
Kommando: tag.....	21	Tingbeholder.....	45
Kommando: evl (avanceret).....	22	Bokse.....	45
Kommando: tast.....	23	Kulisser.....	45
Kommando: hvis.....	25	Sætstykker.....	45
Kommando: ellers.....	26	Arkitekten.....	46
Kommando: har.....	27	Ting.....	46
Kommando: hjælp.....	28	Væsener.....	46
.....	29	Hørevidde.....	46
Kommando: lyd	30	Kontinent.....	46

Indledning

Denne manual er en gennemgang af funktionerne i **PicturesInYourHead**. Og hvad de kan.

Manualen indeholder en beskrivelse af principperne bag opbygningen af verdener i **PicturesInYourHead**. Manualen anvendes, til at få en bedre forståelse for:

- Hvorledes funktionerne i **PicturesInYourHead** virker.
- Hvordan den principielle indretning af en **PicturesInYourHead** verden er .
- De kommandoer der anvendes til at reagerer på, hvad opdageren i denne verden finder på, hvorved verdenen -eller historien- bliver præsenteret for ham.

Manualen er ikke en introduktion til, hvordan man bygger verdener, der er Byggevejledningen meget mere velegnet.

Oversigt over alle dokumenter

- **PicturesInYourHead** Manual
Som beskriver, hvorledes byggestenene i **PicturesInYourHead** virker, altså hvilke 'legoklodser' der er tilrådighed, det er her man finder lige den detalje, man har brug for.
- **PicturesInYourHead** Byggevejledning
Som beskriver principperne for, hvordan byggestenene sammensættes til verdner. Her finder man vejledning til, hvordan man bærer sig ad med at få en verden op at stå. Demonstreret med mange eksempler.
- **PicturesInYourHead** Vejledning i brug af byggesystemet
Som er det værksted, man arbejder i, når man bygger verdener. Derfor er denne vejledning opbygget som en beskrivelse af værkstedet og de værktøjer man finder i det.
- **PicturesInYourHead** PresentationsDel
Som beskriver installation og brug af **PicturesInYourHead**'s presentationsdel.
- **PicturesInYourHead** videoer
Som indeholder links og beskrivelser af videoer der demonstrerer **PicturesInYourHead**'s værktøjer og principper .

Hændelsesbeskrivelser

Introduktion

Alle hændelsesbeskrivelser er opbygget på samme måde.
<enFilterdel/en processbeskrivelse>

F.eks.

<se&mand/vis [Manden er høj og går med stok]>

se&mand

er filterdelen af hændelsesbeskrivelsen.

vis [Manden er høj og går med stok]

er processbeskrivelsen af hændelsesbeskrivelsen.

< starter en hændelsesbeskrivelse.

> afslutter en hændelsesbeskrivelse.

/ adskiller filterdel og processbeskrivelse.

Når Opdageren indtaster:

Hvordan ser manden ud -

så vil filterdelen *se&mand* forstå at det indtastede indeholder:

hvordan **ser manden** ud

og da begge tekstdele er tilstede i det tastede, (& kræver dette) udføres processbeskrivelsen, denne vil vise sætningen:

Manden er høj og går med stok.

Sammensatte Processbeskrivelser

Kommandoer kan sættes sammen, og derved kan, alle mulige også meget omfangsrige, processer aktiveres, når et filter forstår noget der er tastet.

En hændelsesbeskrivelse består som nævnt af:

<filterdel/processbeskrivelse>

filterdelen vil lytte efter hvad der bliver tastet og lytte efter alle autohændelser, (se kommandoen **tast**). Når filterdelen forstår det tastede, så udføres processbeskrivelsen. Processbeskrivelsen kan bestå af en enkelt kommando eller flere kommandoer adskilt med . (punktum).

F.eks.:

se&mand/vis [Manden er høj og går med stok. Han råber truende efter dig]. lyd(råb).flyt >

kommandoerne vis, lyd, flyt danner tilsammen en sammensat processbeskrivelse. Alle kommandoer bliver udført i den rækkefølge de er anført. Men nogle kan overspringes se **hvis, har, hjælp** og **ellers**. 'råb' er navnet på en lydfil i **PicturesInYourHead**'s lydfil mappe på harddisken.

Filtre

Dette afsnit beskriver, hvorledes filtre opbygges.

Et filter starter hændelsesbeskrivelsen. Filteret bestemmer om det, der er tastet af opdateren, kan forstås dvs. indeholder de tekstdele som filteret forventer. Til filteret skal der altid være knyttet en processbeskrivelse, som angiver, hvad der skal ske, når filteret forstår noget, der er tastet.

Et filter består af almindelige tegn og følgende specialtegn

& Og
| Eller
- ikke
* alt muligt

{ } anvendes til at bestemme rækkevidden af | - &. { } kaldes også en blok

Eksempler:

*

Det simpleste filter, dette filter forstår alting.

_*

Et filter der ingenting vil forstå. ! Læses: ikke alt muligt (Mere for princippet skyld)

Brug af blokke

Specialtegnene bliver behandlet fra venstre mod højre. I mere komplicerede filtre er det praktisk at bruge blokke { }.

{angrib|slå|spark|}&mand

vil forstå enhver sætning der indeholder en af tekststumperne: angrib, slå, spark samt tekststumpen mand.

f.eks.:

angrib manden

slå manden

spark manden i maven.

Men også:

mon den herremand kan slå en frø?

For at vise at man kan få filtre tilpasset til alle situationer, laver vi nu filteret lidt om, så det

ikke forstår den sidste tekst.

{angrib|slå|spark|}&mand&-frø

som læses:

{angrib *eller* slå *eller* spark} og mand og ikke frø

så selv om både slå og mand findes som tekststumper i teksten, så vil filteret ikke reagerer fordi det nu også kræver at tekststumpen frø ikke må findes i den.

Hvilke filtre prøves og i hvilken rækkefølge

Kun filtre der er indenfor hørevidde vil blive prøvede.

Den korte forklaring på hørevidde er, at det der er indenfor hørevidde, er den boks, Avataren er i, de ting han har og de ting der er, i den boks han er i og Avataren selv alle er indenfor hørevidde .

Den længere forklaring er:

Hørevidde er en betegnelse for den gruppe filtre, som bliver taget medtaget i undersøgelsen af hvilket filter, der forstår en hændelse. Når opdageren taster noget, så er det ikke alle filtre i alle bokse der bliver afprøvet. Reglen er den, at de filtre som Avatarens sætstykker har og de filtre der findes i den aktive boks' sætstykker, samt de filtre der er i sætstykkerne for tingbeholderen for Avataren og tingbeholderen for den aktive boks vil være indenfor hørevidde. Kun filtre for de sætstykker der er sat op på kulissen medtages. Alle andre filtre er 'udenfor hørevidde'. De filtre, der anføres i `_default` boksen, vil altid blive prøvet efter alle andre filtre, der er tilrådighed. Se afsnittet `_default` boksen i **PicturesInYourHead** byggevejledning.

Filtre er sorteret, så de prøves i rækkefølge. Denne rækkefølge bestemmes ud fra ,hvor lang en tekststreng et filter skal bruge, før der er en chance for at det forstår noget. Her er nogle eksempler:

`<se&mand /se>` kræver se (2 tegn) + mand (4 tegn) i alt 6 tegn.

`<se&hamster/se>` kræver se (2 tegn) + hamster (7 tegn) i alt 9 tegn.

Da `se&hamster` skal bruge 9 tegn ,vil dette filter blive afprøvet før `se&mand` , som kun skal bruge 6 tegn.

I dette eksempel er det jo lige meget hvilken rækkefølge filtrene afprøves i. Men det er ikke altid tilfældet.

f.eks.

<spring&over&ulv/vis [Det lykkedes at undgå ulven] (13 tegn)

<spring|hop/vis [Ja hoppe og springe det kan enhver] (6tegn)

taster opdageren

spring over ulv

vil vi jo helst have følgende svar

Det lykkedes at undgå ulven

Men hvis filteret *spring/hop* blev prøvet først, ville det også forstå ordet spring og være tilfreds hermed og vi ville få følgende svar:

Ja hoppe og springe det kan enhver

og det var jo ikke meningen

så tommelfingerreglen er: jo mere tekst et filter kræver, jo før vil det blive afprøvet.

Her er nogle mere avancerede eksempler

<manden&manden/vis> i alt 12 tegn.

Her er to filterdele, der skal være opfyldt hver på 6 tegn, hvad filterdelens tekst er, er ligegyldigt for prioriteringen

<manden|manden/vis> i alt 8 tegn

Kun en af filterdelene behøver at blive opfyldt, altså kun 8 tegn i alt.

<mand/vis> i alt 4 tegn

Og så en helt speciel en:

<mand&nd/vis> i alt 6 tegn

Dette filter vil forstå alt, der indeholder mand, men blive udført før filteret *mand* fordi filteret kræver *nd* også, men en tekst der indeholder mand vil også indeholde *nd*. Det er dog meget sjældent man får brug for sådant et 'dobbelt konfekt' trick.

Kulisser

Boksens sider, top og bund og selve boksens rum kaldes under et for kulisser. En kulisse har 3 opgaver.

1. På den skriver arkitekten, hvad kulisserne er for noget, en væg, et skovbryn, etc.
Det gør han ved at skrive på sætstykket, som hænger på kulisserne.
2. Til kulisserne knyttes hændelsesbeskrivelser i form af et filter, der kan forstå noget opdageren taster og en processbeskrivelse, som angiver, hvad der skal ske, når filterdelen, har forstået det tastede. Dette skrives også på sætstykket, der hænger på kulisserne.
3. Kulissen er også der, hvor man angiver, hvilken anden boks kulissen forbinder til, igen ved at anføre det på et sætstykke der hænger på kulisserne.

Kulisserne betegnes

- T Boksens Loft
- Ø Boksens Øst væg
- V Boksens Vest væg
- S Boksens syd væg
- N Boksens Nord væg
- B Boksens bund
- R Selve rummet i Boksen
- X Kan bruges til 'Special effects'
- Y Kan bruges til 'Special effects'
- Z Kan bruges til 'Special effects'.

Tekster på en kulisse

Teksten på en kulisse vil blive vist, når et filter, der har kommandoen **se**, i sin processbeskrivelse forstår, hvad opdageren taster.

f.eks.

Teksten på et sætstykke der hænger på kulisse Ø er:

Grøn eng med mange smukke blomster i alle farver.

En hændelsesbeskrivelse for kulisserne ser sådan ud:

<se&øst/se>

Opdageren taster:

se mod øst

og resultatet bliver:

se mod øst -

Grøn eng med mange smukke blomster i alle farver.

Herudover vil normal tekst, der står i R kulissen, blive vist af **flyt** kommandoen når avataren kommer ind i boksen. Tingbeholderens indhold, for den boks avataren kommer ind i, bliver også vist af **flyt** kommandoen.

Specielle effekter

\$ Tegnet

Var vor **Ø** kulisse forbundet til en anden boks, der er en eng, så kan man slippe nemt om ved det, ved at skrive følgende i R Kulissen for engBoksen:

Du står på en \$Grøn eng med mange smukke blomster i alle farver\$. Lige på midten af engen er der et træ, der hænger ligesom en dukke i træet.

Alt hvad der står omgivet af \$ tegn i R kulissen, vises nemlig af en **se** kommando på en kulisse i en anden boks, som er forbundet til engboksen. På denne måde kan man slippe for, at skrive om engen på de omgivne boksers kullisser og nøjes med at gøre det et sted, nemlig i R kulissen for selve eng boksen.

Samme mulighed bruges til at give en kort beskrivelse af, hvad der er i ting beholderen.

Vi har f.eks. En boks der skal være en kniv. Vi skriver i R kulissen for kniv boksen:

\$En \$skarp blinkende \$kniv\$.

Når **se** eller **flyt** kommandoerne viser, hvad der er i tingbeholderen, vil de skrive:

En kniv

Anvendte vi ikke \$ tegnene i dette tilfælde, så vil **se** og **flyt** kommandoen ikke vise noget.

Tekst omgivet af \$ bruges altså til at give en kort beskrivelse af det væsentlige ved en boks.

Denne specielle effekt har kun virkning for tekst skrevet på R kulissen's sætstykker.

Tegnet

Når **flyt** kommandoen flytter avataren fra en boks til en anden, sker det, fordi **flyt** kommandoen er angivet i en processbeskrivelse, hvor filteret forstår at avataren skal flytte sig, f.eks står avataren måske i et hus og kigger ud på engboksen. I den kulisse i husboksen der vender ud til engen, kan vi, udover at angive at kulissen er forbundet til engen, angive følgende hændelsesbeskrivelse:

<gå&nord/flyt>

taster opdageren:

Gå mod nord

Du står på en Grøn eng med mange smukke blomster i alle farver. Lige på midten af engen er der et træ, der hænger ligesom en dukke i træet.

Som nævnt ovenfor sørger **flyt** kommandoen for, at den normale tekst på r kulissen i engboksen bliver vist, når avataren går ind i engboksen.

Det kan vi gøre mere ud af. Stadigvæk på den nord kulisser der vender ud mod engen, skriver vi følgende:

En gammel trædør står på klem. #Du går gennem døren#

Taster opdageren nu:

se mod nord

Vises:

En gammel trædør står på klem.

Grøn eng med mange smukke blomster i alle farver

Og taster opdageren nu:

gå nord

så vises:

Du går gennem døren

Du står på en Grøn eng med mange smukke blomster i alle farver. Lige på midten af engen er der et træ, der hænger ligesom en dukke i træet.

Tekst omgivet af **#** tegnet anvendes altså til at angive, hvad der sker, når man passerer gennem en kulisser til en anden boks. Feltet: **{[Tekst der vises]}** i **flyt** kommandoen kan også bruges, men når man senere skal forbedre sin verden, kan man let overse, at denne tekst også skal rettes. Når man tager og lægger en ting ved **tag** og **læg** kommandoerne, sker det samme, står der noget tekst i **##** på tekstdelen af sætstykket, vil det blive vist.

| Tegnet

Sætstykket på en kulisser kan skiftes ud med et andet sætstykke, det kan f.eks. være, at der er et vindue i en væg, som avataren smadrer. For at dette kan ses, skal vi have udskiftet sætstykket med et andet, der viser at vinduet er smadret. Alle kommandoer kan udskifte et sætstykke når kommandoen udføres.

Teksten på sætstykket med vinduet kunne være:

Her er et vindue, der ikke kan lukkes op. (sætstykke v-1)

Og

Teksten på sætstykket, når vinduet er smadret, kunne være:

Her er et smadret vindue, der ligger lidt glasskår i vindueskarmen. (sætstykke v-2)

Den første tekst omskriver vi nu til:

Her er et vindue, der ikke kan lukkes op.

|vinduet går i stykker, de fleste af glasskårene falder udenfor|

(sætstykke v-1)

Når vi så smadrer vinduet skal sætstykket skiftes ud. Det kunne gøres ved følgende hændelsesbeskrivelse

<{sma/knal}&vindue/vis [du knalder vinduet med din sko] sætst(v-2)>

Beder opdageren avataren om at se på vinduet, dernæst smadre det, og så se på vinduet igen sker der følgende:

se vest

Her er et vindue, der ikke kan lukkes op.

Knald vinduet i stykker

du knalder vinduet med din sko

vinduet går i stykker, de fleste af glasskårene falder udenfor

se vest

Her er et smadret vindue, der ligger lidt glasskår i vindueskarmen

Teksten omgivet af tegnet | på sætstykket anvendes altså til at angive, hvad der sker, når et sætstykke i en kulisse skydes væk og erstattes med et andet sætstykke.

Man kan også se, at vi på en fiks måde, har fået adskilt hvad avataren gør, fra hvad det er der sker med vinduet. Det gør teksten nemmere at rette i senere når verden skal forbedres.

Kommandoer

Indledning til kommandobeskrivelse

Hver kommando beskrivelse er opbygget således:

Kommando: **kommandonavn**

Virkning:

Beskrivelse af hvad kommandoen gør

Muligheder:

Hvilke felter kommandoen kan indeholde. { } angiver kommandofelter som kan udelades

Eksempler:

Eksempler på kommandoer

Anvendelse:

Hvad kommandoen typisk anvendes til.

Kommandofelter

Næsten alle kommandoer indeholder feltet [**Tekst der vises**] og feltet **sætst(liste af sætstykker der ændres)**

Feltet:[**Tekst der vises**]

Når kommandoen udføres vises '**Tekst der vises**'. Denne tekst kan bestå af bogstaver og tal og specialtegn. Dog kan man ikke bruge følgende tegn: ' og " og [].

I kommandobeskrivelserne betyder {} omkring feltet, at man kun behøver at tage det med, hvis man har brug for det.

Feltet: **sætst(liste af sætstykker der ændres)**

Her angives hvilke kulisser, der skal have ændret deres sætstykker, se afsnittet

Sætstykker i **PicturesInYourHead** Byggevejledning F.eks. betyder **r-1 n-2**, at kulisser **r** skal have sat sit sætstykke **1** op og kulisser **n** skal have sat sit sætstykke **2** op. Skal alle kulisser have sat sætstykker med samme nummer op f.eks. **6** kan man skrive: **a-6**. Hvor '**a**' står for alle kulisser. I kommandoerne er beskrevet, hvilken boks der får ændret sine sætstykker.

Kommando: **flyt**

Virkning:

Flytter avataren fra den boks han opholder sig i, den aktive boks, til den boks som det sætstykke, **flyt** kommandoen er skrevet på, er forbundet med.

Flyt kommandoen kan vise en tekst når den udføres, og også ændre sætstykkerne for de kulisser som den boks har, som avataren nu forlader.

Findes der noget tekst på sætstykket som kommandoen er skrevet på, så vil den tekstdel der er omgivet af # # også blive vist, når **flyt** udføres. Denne ## tekst vises, efter den tekst som er angivet i **flyt** kommandoen. Og til allersidst vises sætstykke teksten for r kulissen i den boks avataren er flyttet til, samt tekst på r kulissen der er omgivet af \$ tegn for de bokse, der er i tingbeholderen, for den boks der flyttes til.

Er der en lyd i gang, da vil lyden blive stoppet af **flyt** kommandoen, hvis den er startet med en **bag** variant. En illustration startet med en **bag** variant vil også blive fjernet.

Muligheder:

flyt {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}

En **flyt** kommando skal stå tilsidst i en processbeskrivelse, eller lige foran **ellers** kommandoen. Sætstykkerne ændres for den boks, der forstår hændelsen og som avataren derfor flytter væk fra. Altså den aktive boks som avataren forlader.

Eksempler:

<gå&vest/flyt [Du går ud gennem døren] sætst (r-2)>

<kravl&ned/flyt [Du kravler ned ad stigen og ender i den mørke kælder]>

<gå&vest/[Bunk - av- du kan ikke gå gennem en murstensvæg]

Anvendelse:

Kommandoen bruges til at flytte Avataren fra et rum til et andet. Teksten omgivet af ## på kulissens sætstykke vil altid blive vist, også hvis der ikke er nogen boks forbundet med kulissens sætstykke. På denne måde, kan man vise grunden til, at avataren ikke kan flyttes i sætstykkets retning.

Kommando: vis

Kommandoen vil vise den tekst der er angivet i **[Tekst der vises]**.

Virkning:

Viser **[Tekst der vises]** og kan evt. Også skifte sætstykkerne på boksens kulisser.

Muligheder:

vis {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}

Sætstykkerne ændres, for den boks der forstår hændelsen normalt den Aktive boks.

Eksempler:

I sætstykke x-1 kunne stå

<hvad&er&her>vis [Hov - der flyver en guldsmed forbi] sætst (x-2)>

I sætstykke x-2 kunne stå

<{se|fang}&guldsmed/vis [Guldsmede er hurtige- den er væk nu]>

hvad&er&her>vis [For lidt siden var her en guldsmed, men ellers er her ikke noget]>

Den første kommando fortæller om guldsmeden, men ændrer samtidigt sætstykket for kulissen, så guldsmeden ikke ses mere.

</vis>*

Denne hændelsesbeskrivelse laver slet ingenting, der er ingen tekst at vise og ingen sætstykker at ændre.

Anvendelse:

Kommandoen bruges typisk til at angive småting der sker, når opdageren har gjort noget.

Til at fortælle hvad en kulisse er for noget, er det mest praktisk at bruge teksten på kulissens sætstykke ved hjælp af kommandoen **se**. Som eksemplet ovenfor viser er **x y z** kulisserne gode til at beskrive de ting, der ikke har noget med retningerne i boksen at gøre.

Kommando: **se**

Virkning:

Kommandoen **se** viser, hvad der står på kulissens sætstykke. Denne sætstykketekst beskriver, hvad kulissen er. F.eks. En kampestensvæg, eller en bjergside. Bokse kan være forbundet til hinanden, uden at der er synlige skillevægge. F.eks. Kan en eng grænse op til en skov. Derfor vil **se** kommandoen også vise den tekst, der er omgivet af \$\$ i r kulissen i den boks kulissens sætstykke forbinder til. **Se** viser også den tekst, der står omgivet af \$\$ på r kulissen for de ting, der er i tingbeholderen for den boks, der forstod hændelsesbeskrivelsen.

Muligheder:

se [{ Tekst der vises}] {sætst(liste af sætstykker der ændres)}

Sætstykkerne ændres for den boks, der forstår hændelsen, normalt den Aktive boks.

Eksempler:

`<se&styrbord/se>`

Vil vise den tekst, der står på det sætstykke, som kommandoen er skrevet på. I dette tilfælde kunne sætstykke teksten være: 'Styrbord ræling'.

`<se&øst/se> [Det er ved at blive tåget] sætst(ø-2)>`

Her kunne sætstykketeksten for kulisse ø være:

'Langt borte på sletten kan du skimte en hytte med lys i de små vinduer.'

og kulisse teksten for sætstykke 2 på ø kulissen være:

'Det er for tåget til at man kan se noget'.

Når opdageren taster se mod øst, vil resultatet blive:

se mod øst

Det er ved at blive tåget

Langt borte på sletten kan du skimte en hytte med lys i de små vinduer.

se mod øst

Det er for tåget til at man kan se noget

Teksten, som kulissen viser, når man ser på den, ændres altså ved at ændre sætstykket.

F.eks. kan en kulisse, der optræder som en låst dør, efter at Avataren har prøvet at brække døren op, få ændret sit sætstykke til et der viser 'Nogen har prøvet at brække døren op'.

Anvendelse:

Se kommandoen bruges til at orientere sig i den boks avataren er kommet ind i. **Se** kommandoen skal skrives på hvert sætstykke, som skal kunne vise sit sætstykkets tekst. Det er dog ikke strengt nødvendigt for **r** kulissen, da dennes tekst vises, når **flyt** kommandoen flytter avataren til boksen. Men det er godt at gøre det. Er opdageren blevet forvirret over hvor han befinder sig, vil han nemlig så få vist samme tekst, som da han kom ind i boksen. Som vist ovenfor kan **sætst (liste af sætstykker der ændres)** bruges til at ændre på kulissen, efter at man har set på den, og **[Tekst der vises]** bruges til at fortælle, hvad det er der sker.

Kommando: læg

Virkning:

Denne kommando bruges til at angive, at avataren lægger en ting fra sig, som han bærer på. Tingen bliver flyttet fra avatarens tingbeholder til tingbeholderen for den aktive boks. Herfra kan avataren så tage den igen. Se kommandoen **tag**. Findes der noget tekst på sætstykket, så vil den tekstdel der er omgivet af **# #** også blive vist når **læg** udføres. Denne **##** tekst vises efter den tekst, som er angivet i **læg** kommandoen.

Muligheder:

læg {[**Tekst der vises**]} {sætst(liste af sætstykker der ændres)}

Sætstykkerne ændres for den boks, der forstår hændelsen og derfor bliver lagt.

Eksempler:

<læg&sten/læg [Stenen falder på gulvet med et bump]

<sæt&glas/læg[Glasset smutter ud af hånden på dig og går i stykker] sætst(r-2)>

Anvendelse:

Kommandoen anvendes mest praktisk i **r** kulissen, for den ting der skal lægges. Som eksemplerne viser kan [**Tekst der vises**] bruges til at angive, hvad der sker. Har man flere forskellige **læg** hændelser, er det mest praktisk at bruge **##** teksten på selve tekststykket. **sætst (liste af sætstykker der ændres)** bruges til at ændre sætstykket for den ting, der lægges. - hvis det for eksempel skal vises, at den går i stykker. Sætstykke **r-2** for glasset i eksemplet skal så vise, at det er gået i stykker og ikke mere kan bruges, men bare er glasstumper.

Kommando: **ting**

Virkning:

ting undersøger om tingbeholderen i den boks, som avataren er i, -den aktive boks-, indeholder de tingbokse, som er angivet i liste af bokse. Anføres tingboksnavnet med - så betyder det, at boksen med dette boksnavn ikke må være i tingbeholderen i den aktive boks. Anføres tingboksnavnet med + så betyder det, at boksen med dette boksnavn skal være i tingbeholderen for den aktive boks. Er situationen den, at tingbeholderens indhold er som kommandoen kræver, så er kommandoen en succes og **succestekst** vises. Ellers vises **fejlttekst** og resten af kommandoerne i den sammensatte processbeskrivelse vil ikke blive udført, se dog **ellers**.

Muligheder:

ting (liste af bokse) [{succestekst}/{fejlttekst}] {sætst(liste af sætstykker der ændres)}.

Læg mærke til at (**liste af bokse**) skal med og at der skal stå + eller - foran boksnavnene, samt at [/] skal med, medens {**succestekst**} og {**fejlttekst**} kan udelades.

Boksene der er nævnt i (**liste af bokse**) skal også være oprettede.

Sætstykkerne ændres for den aktive boks, men kun når boks betingelsen ikke blev opfyldt.

Eksempler:

<drik&vand&glas/ting (+Glasmedvand)[Ah-det var rart/Her er ikke noget glas med vand]>

Anvendelse:

Kommandoen bruges til at afgøre, hvad der skal ske i en boks afhængigt af, hvad boksen indeholder.

Kommando: **tag**

Virkning:

Denne kommando bruges til at få avataren til at tage en ting. Tingen bliver så flyttet fra den boks avataren er i -den aktive boks'- tingbeholder til avatarens tingbeholder . Herfra kan avataren så lægge den fra sig igen, i en anden boks. Se kommandoen **læg**. Findes der noget tekst på det sætstykke, som **tag** kommandoen er skrevet på, så vil den tekstdel, der er omgivet af # # også blive vist når **tag** udføres. Denne ## tekst vises efter den tekst, som er angivet i **tag** kommandoen.

Muligheder:

tag {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}

Sætstykkerne ændres for den boks, som forstår hændelsen og som derfor bliver taget.

Eksempler:

<tag&sten/læg [Du samler stenen op]

<fang&solsort/læg[Solsorten forvandles til et slimet kryb]sætst(r-2)>

Anvendelse:

Kommandoen angives mest praktisk i r kulissen for den ting der skal tages. Som eksemplerne viser kan [**Tekst der vises**] bruges til at angive, hvad der sker, når avataren tager en ting. Har man flere forskellige **tag** hændelser, er det mest praktisk at bruge ## teksten på selve tekststykket. **Sætst(liste af sætstykker der ændres)** bruges til at ændre sætstykker, på den ting der tages.

Sætstykke *r-2* for solsort, skal så vise, at den ikke mere er en solsort, men er et 'slimet kryb'.

Kommando: **evl (avanceret)**

Virkning:

Denne kommando udfører en smalltalk sætning i kontekst myWorld. Smalltalk sætningen skal være indesluttet i " ". Udtrykket vil blive udført.

Note: Kræver kendskab til Smalltalk og til arkitekturen for **PicturesInYourHead**

Muligheder:

evl ("Smalltalk eksp."){[Tekst der vises]} {sætst(liste af sætstykker der ændres)}

Sætstykkerne ændres for den boks der forstår hændelsen, normalt den Aktive boks.

Eksempler:

udeladt

Anvendelse:

Med denne kommando kan alle mulige 'special effects' laves. Men hele **PicturesInYourHead** kan også ødelægges.

Kommando: **tast**

Virkning:

Denne kommando bruges til at simulere, at opdageren har tastet noget fra tastaturet. Hver af disse tastninger skal være adskilt af en eller flere blanke. Den simulerede indtastning kaldes også 'en autohændelse'. Sådanne 'autohændelser' er den måde, man får boksene til at give hinanden besked på. For at forhindre at 'de gale bokse' har et filter, der tilfældigvis forstår en autohændelse, så er det en god ide at putte et \$tegn ind i autohændelsen. Dette bevirker, at kun de filtre der passer helt eksakt med autohændelsen vil forstå den. Autohændelser kan kun høres af bokse der er indenfor hørevidde.

Muligheder:

tast (liste af Auto hændelser) {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}

En autohændelse kan kun indeholde normale bogstaver, tal og \$. Et mellemrum vil adskille to autohændelser i listen af autohændelser.

Sætstykker ændres for den boks, der forstår hændelsen.

Eksempler:

```
<tag&kat/tast ($sussiFangKat)
```

Anvendelse:

I eksemplet ovenfor forestiller vi os, at vi går en tur med hunden Sussi. Hunden Sussi er altså i tingbeholderen for avataren. I denne 'Sussi boks' er der angivet et filter, der ser således ud:

```
<$sussiFangKat/læg [Sussi drøner af sted efter katten] sætst(r-2)>
```

Resultatet bliver følgende:

```
tag og fang den kat
Sussi drøner af sted efter katten
```

På grund af **læg** kommandoen er Sussi nu også hoppet ud af avatarens tingbeholder og befinder sig i den aktive boks' tingbeholder. Går avataren nu videre til en anden boks, har han ingen hund med længere. Det er grunden til **{sætst(liste af sætstykker der ændres)}** **sætst (r-2)** . 'Sussi boksen' skal nu indeholde et filter, der aktiverer en kommando, som gør det muligt at kalde Sussi tilbage igen. Det kunne f.eks. være:

<kom&sussi/tag[Sussi kommer glad tilbage, katten sidder i et træ] sætst(r-1)>

Alt i alt bliver det til følgende:

tag og fang den kat

Sussi drøner af sted efter katten

kom tilbage sussi

Sussi kommer glad tilbage, katten sidder i et træ.

Og avataren kan gå videre med Sussi igen siddende i avatarens tingbeholder klar til nye æventyr.

Kommando: hvis

Virkning:

Denne kommando bruges til at indføre tilfældigheder. Kommandoen udfører et terningkast med en terning der indeholder 0 til 10 øjne. Et tal **ciffer** anføres i kommandoen og er det større en det antal øjne terningen viser, så har **ciffer** vundet over terningen og kommandoen gennemføres med succes. I dette tilfælde vises succестekst og der fortsættes med næste kommando i processbeskrivelsen.

Vinder terningen over **ciffer** vises **fejlttekst** og de sætstykker der er angivet i **{sætst(liste af sætstykker der ændres)}** bliver sat op og resten af kommandoerne i processbeskrivelsen springes over. Se dog **ellers** kommandoen.

Muligheder:

hvis ciffer [{succestekst}/{fejlttekst}] {sætst(liste af sætstykker der ændres)}.

Læg mærke til at **ciffer** skal anføres tillige med specialtegnene [/], men **{succestekst}** og **{fejlttekst}** kan udelades.

Sætstykkerne ændres for den boks, der forstår hændelsen, men kun når **ciffer** taber

Eksempler:

Det er ret svært at fange en fisk med hånden, så opdageren skal prøve i snit 5 gange før det lykkes:

```
<fang&fisk/hvis 2 [Yep - du har en sprællende fisk i hånden/Øv den smuttede væk].
```

Den artige hund Baquii plejer at komme, når der kaldes, men ikke altid, så i 9 ud af 10 tilfælde vil det lykkes.

```
kald&Baquii/hvis 9 [Baquii kommer glad løbende/Baquii lader som om den ikke kan høre].
```

Anvendelse:

I en sammensat kommando er **hvis** kombineret med **ellers** meget nyttig. Se her:

```
<angrib&dværg/hvis 6[Dværgen snubler og falder, han ser op på dig og beder om nåde/Dværgen undviger og slår dit sværd ud af din hånd]. ellers. tast ($tabSværd)>
```

Vinder terningen over **6** vises teksten: Dværgen undviger og slår dit sværd ud af din hånd. Og dernæst udføres *Tast(\$tabsværd)*, som så skal forstås af sværd boksen f.eks. ved at den har følgende filter:

```
<$tabSværd/læg> . Se også ellers kommandoen
```

Kommando: **ellers**

Virkning:

ellers kan anvendes i sammensatte processbeskrivelser, som indeholder **hvis, har, hjælp og ting**

Disse kommandoer har alle det tilfældes, at de kan udføres med succes eller ikke.

Når de udføres med succes, så vil resten af kommandoerne i den sammensatte processbeskrivelsen blive udført. Udføres de ikke med succes vil resten af kommandoerne i processbeskrivelsen blive oversprunget. Men indføres **ellers** blandt de kommandoer der efterfølger **hvis, har, ting** eller **hjælp**. Så vil kommandoerne mellem **hvis, har, hjælp, ting** og **ellers** kommandoen blive udført hvis succes. Og de der følger efter **ellers**, vil ikke blive udført. Ifald **hvis, har, hjælp, ting** ikke blev udført med succes, vil kommandoerne efter **ellers** blive udført. Hvis **ellers** optræder i en sammensat kommando uden **hvis, har, hjælp, ting** gives en fejlmeddelelse. Det samme sker hvis **ellers** optræder flere gange i en sammensat kommando. Eller ifald der ikke følger nogen kommandoer efter **ellers**.

Muligheder:

ellers

Bemærk at der kun kan være een **ellers** kommando mellem < og >, samt at der skal være en **hvis, hjælp, ting** eller **har** kommando til venstre for **ellers**. **ellers** kan heller ikke stå tilsidst i en processbeskrivelse.

Eksempler:

```
<fang&hund/hvis 6[Du fanger hunden/hunden smutter fra dig].tast($givHundSnorPå). ellers.  
flyt[du opgiver og går din vej]>
```

Anvendelse:

ellers er rigtig god til at vælge mellem to sæt kommandoer i processbeskrivelsen afhængigt af hvad **hvis, har, ting** eller **hjælp** fik som resultat. Eksemplet viser en situation hvor avataren prøver at fange en hund. Lykkes det, giver avataren hunden snor på, lykkes det ikke, så opgiver han og går videre til en anden boks.

Kommando: **har**

Virkning:

har undersøger om avatarens tingbeholder, indeholder de tingbokse, der er angivet i liste af bokse. Anføres boksnavnet med - , så betyder det, at tingboksen med dette boksnavn ikke må være i tingbeholderen. Anføres boksnavnet med + , så betyder det, at tingboksen med dette boksnavn skal være i tingbeholderen for avataren. Er situationen den, at tingbeholderens indhold er som kommandoen kræver, så er kommandoen en succes og **succestekst** vises. Ellers vises **fejlttekst** og resten af kommandoerne i den sammensatte processbeskrivelse vil ikke blive udført, se dog **ellers**.

Muligheder:

har (liste af bokse) **[{succestekst}/{fejlttekst}]** **{sætst(liste af sætstykker der ændres)}**.

Læg mærke til at **(liste af bokse)** skal med og at der skal stå + eller - foran boksnavnene, samt at **[/]** skal med medens **{succestekst}** og **{fejlttekst}** kan udelades.

Boksene, der er nævnt i **(liste af bokse)**, skal også være oprettede.

Sætstykkerne ændres for den aktive boks, men kun når boks betingelsen ikke blev opfyldt.

Eksempler:

```
<tag&sværd/har(+spyd)[Du lægger dit spyd og tager sværdet/Du tager sværdet].tast
($lægspyd).tast($tagsværd). ellers. tast($tagsværd)>
```

```
<kravl&igennem/har (-kasse) [/Kassen er for stor, så du stiller den ].
flyt[Du kravler igennem det snævre hul].
ellers.
tast ($sætkasse).flyt [Du kravler igennem det snævre hul]>
```

Anvendelse:

Kommandoen kan bruges til at afgøre, hvad der skal ske afhængigt af, hvad avatarens tingboks indeholder. Det første eksempel viser, at kommandoen kan bruges til at sørge for, at avataren ikke har mere end et våben til rådighed. Det andet eksempel viser, at avataren nogen steder bliver nødt til at lægge nogen ting, før han kan gennemføre sin opgave.

Autohændelserne \$lægspyd, \$sætkasse,\$tagsværd skal kunne forstås af tingboksene spyd, kasse og sværd.

Kommando: hjælp

Virkning:

Hjælp anvendes til at give opdageren en lille besked, som hjælper ham videre, når han ikke kan finde ud af, hvad han skal gøre. Når verden startes op, skal man sørge for at en målboks er angivet: se **Verden startes**. Hjælp kommandoen vil finde, den næste hændelse der skal ske, for at bringe avataren videre til målet. Og viser denne i forkortet form. Dette kan selvfølgelig kun virke, når verden hænger så godt sammen, at der er sådan en hændelse i den boks hvor **hjælp** udføres. Afhængigt af opdagerens niveau, som han angiver når han starter sin opdagelsesrejse, får han mere eller mindre hjælp. Se afsnittet **Verden startes** i byggevejledningen til **PicturesInYourHead**.

Finder **hjælp** kommandoen, at det er på tide at give hjælp, så vises **succestekst**, ellers vises **fejlttekst** og optræder **hjælp** i en sammensat processbeskrivelse vil resten af kommandoerne blive oversprunget hvis kommandoen ikke giver hjælp. Se dog **ellers**.

Muligheder:

hjælp [{succestekst}/{fejlttekst}] {sætst(liste af sætstykker der ændres)}.

Læg mærke til at [/] er nødvendige, medens {succestekst} og {fejlttekst} godt kan udelades.

Sætstykkerne ændres for den boks, der forstår hændelsen, men kun når kommandoen ikke giver hjælp.

Eksempler:

```
<*/hjælp [How- du har en seddel i lommen, her står.../ak - ja, det går ikke for godt :-(>
```

```
<åben&køleskab/flyt [Da du åbner køleskabet lyder der et svup og du bliver suget ind i det og døren smækker igen]sætst(r-3)>
```

Anvendelse:

Læg mærke til det filter der er anvendt i eksemplet '*', det er det simplest mulige filter, som forstår alle hændelser. Det er også det filter, der vil blive prøvet sidst af alle de lyttende filtre, som er indenfor hørevidde. Det vil derfor virke således, at når opdageren har tastet noget som er 'helt i skoven' og ingen andre filtre, derfor kan forstå det, så vil '*' filteret forstå det tastede og aktivere hjælpefunktionen. Afgør hjælpekommandoen, at det er på tide at give noget hjælp, så vil den give hjælpen, som er et forslag til, hvad der kan tages for at komme videre.

I eksemplet ovenfor forestiller vi os, at avataren skal åbne køleskabet som sin næste handling for at komme videre, det forestiller vi os, at opdageren ikke har kunnet finde på, Er hans niveau 1 (jeg er på besøg) skal han 3 gange skrive noget ingen filtre forstår før hjælpen

gives.

Det kan se således ud:

gå ud

ak – ja, det går ikke for godt :-(

hop op på køkkenbordet -

ak – ja, det går ikke for godt :-(

åben en skuffe -

ak – ja, det går ikke for godt :-(

tag en kniv -

How- du har en seddel i lommen, her står..åben... køleskab...

Som man kan se, består hjælpen i, at **PicturesInYourHead** foreslår en hændelse, som det filter, der bringer avataren videre, kan forstå .

Se **PicturesInYourHead** byggevejledning afsnittet hjælpefunktionen om tips til at bruge hjælp kommandoen.

Kommando: lyd

Virkning:

lyd bruges som navnet siger til at styre lyde i **PicturesInYourHead**. Selve lyden angives som navnet på en .wav fil. **PicturesInYourHead** kan holde styr på afspilningen af en mængde lyde, men kan kun spille to lyde samtidigt, på hver sin lydkanal. Der er derfor en række varianter, man kan angive for at styre, hvordan lyde skal afspilles. Bruger man ikke nogen af dem, så vil en lyd blive spillet en gang på standard kanalen.

Muligheder:

lyd (liste af lyde) {[Tekst der vises]} sætst(liste af sætstykker der ændres)}.

Bemærk at lydfilen skal findes, samt at **stp** varianten ikke kan bruges sammen med andre varianter. Og for hver lydfil kan samme variant ikke bruges mere end en gang. Sætstykkerne ændres for den boks der forstår hændelsen.

Eksempler:

Varianter:

frs tal	lyden afspilles tal gange i træk.
gen tal	lyden gentages tal gange, 0 betyder uendeligt.
pri	lyden vil blive spillet på pri kanalen
stp	lyden vil blive afbrudt
bag	Lyden vil blive afbrudt når en flyt kommando udføres.

frs <tal> betyder at lyden skal spilles <tal> antal gange i træk.

F.eks. har man en en fil: pip.wav som giver en enkelt pippelyd -piiiip- så vil **frs 3** varianten bevirke at den afspilles 3 gange i træk -piiiip- -piiiip- -piiiip-. Dette angives således:

<pip/lyd (pip frs 3)>

Gen <tal> betyder, at lyden skal gentages tal antal gange, er **tal 0** betyder det, at lyden skal gentages for altid. f.eks. vil følgende:

<pippip/lyd(pip frs 3 gen 2)>

bevirke at fuglen nu siger -piiiip- -piiiip- -piiiip- -piiiip- -piiiip- -piiiip-

Det er også muligt at spille flere lyde, har vi f.eks. en anden lydfil, hvor en ugle tuder en gang uhu.wav så kan vi blande uglen og den pippende fugl. Det angives på denne måde:

<fugle/lyd (pip frs 3 gen 0, uhu frs 2 gen 0)> **bemærk kommaet.**

og det vil så lyde således:

-piiip- -piiip- -piiip- -uhu- -uhu- -piiip- -piiip- -piiip- -uhu- -uhu- osv. osv. osv.

Der er ingen begrænsning i det antal lyde, der kan afspilles på denne måde.

Nu kan det ske, at vi kalder på en hund, som så gør, medens uflen tuder og fuglen pipper, så skal vi gerne have hunden til at gø 3 gange, medens fuglene fortsætter. Vi har en wav fil med et enkelt hundevov. Den hedder vov.wav Dette angives således:

<kald&hund/lyd (vov pri frs 3)>

taster opdageren:

kald på hund

medens fuglene er i gang sker der følgende:

Standardkanal

-piiip- -piiip- -piiip- -uhu- -uhu- -piiip- -piiip- -piiip- -uhu- -uhu- osv. osv. osv.

prikanal

vov - vov - vov

Hunden gør 3 gange, **pri** varianten sørger for at hundens gøen på **pri** kanalen blander sig med fuglenes lyde .

Men der er flere varianter, vi kan have en situation, hvor vi lader hunden gø hele tiden indtil opdageren beder den om at tie stille. Først et filter der forstår, at vi beder hunden om at gø hele tiden.

<gø&hund/lyd(vov gen 0)>

Dernæst skal vi have et filter, der kan forstå en hændelse, som får hunden til at tie stille

<ti&stille/lyd(vov stp)>

Når opdageren nu taster:

begynd at gø hund

-vov- -vov- -vov- -vov- -vov- -vov- -vov- -vov- -vov- osv. osv.

Ti nu stille

-tavshed-

Det er **stp** varianten i `<ti&stille/lyd(vov stp)>` som stopper afspilningen af vovene.

PicturesInYourHead. Vil afgive autohændelsen \$nyboks, hver gang avataren kommer ind i en ny boks. Laver man et filter, der forstår denne og kobler en lydkommando på, så kan man starte en baggrundslyd, der kan køre så længe avataren er i boksen. F.eks. vil

`<$nyboks/lyd (bølgeslag gen 0)`

kunne bruges når avataren går ud af en boks, der er en hule og befinder sig i en boks, der er en strand.

Men forlader avataren nu den boks, der er en strand og går tilbage igen til hulen, så ville det være godt, hvis bølge lyden forsvandt samtidigt. Det ordner man ved varianten **bag**

`<$nyBoks/lyd (bølgeslag gen 0 bag). Flyt>`

bag varianten bevirker, at **flyt** kommandoen afbryder bølgeslagslyden, når boksen forlades.

Som en sidste variant kunne man have en murstens væg mod syd, som man ikke kan gå igennem. På syd kulissen kan man da angive:

`<gå&syd/lyd(bong frs 2) [Av for den, du kan da ikke gå igennem væggen...]>`

taster opdageren:

gå mod syd

så:

-bong- -bong-

Av for den, du kan da ikke gå igennem væggen...

Anvendelse:

Man kan finde mange lyde på nettet f.eks.:

http://www.thefreesite.com/Free_Sounds/Free_WAVs/

Men det er også skægt at lave dem selv.

<http://audacity.sourceforge.net/download/>

Er et gratis program man kan downloade- på dansk-, det kan alt hvad man har brug for.

Endeligt har MS-Windows indbygget en simpel lyd optager: Lydoptager/Soundrecorder som kan bruges i en snæver vending.

Kommando: **illu**

Virkning:

illu bruges til at styre visningen af illustrationer i **PicturesInYourHead**. Selve illustrationen angives som navnet på en .jpg fil. **PicturesInYourHead** kan holde styr på visningen af tre illustrationer ad gangen. Det kan man forstå, som at der er tre projektorer, der kan vise hver sin illustration. En projektor viser illustrationen i midten og de to andre viser illustrationen til venstre og til højre. De kan være i gang alle tre samtidigt i vandringsvinduet. Illustrationerne vil blive vist til **PicturesInYourHead** fjerner dem, eller erstatter dem med en nye illustrationer. Arkitekten kan bestemme størrelsen af illustrationen, hvor den skal vises og hvornår den skal fjernes igen. Hertil bruges en række varianter, bruger man ikke nogen af disse, så vil en illustration blive vist midt i vandringsvinduet, i mellem størrelse .

Varianter:

Størrelse:

stor	illustrationen vil fylde det meste af vandringsvinduet
mel	illustrationen vil fylde halvdelen af vandringsvinduet
lille	illustrationen vil fylde en fjerde del af vandringsvinduet
mikro	illustrationen vil fylde en ottende del af vandringsvinduet.

Visning:

stabl	illustrationen vil lægge sig ovenpå en eksisterende illustration, der vises af den valgte projektor, som den vil skjule. Når illustrationen fjernes, kommer den gamle illustration frem igen.
stp	illustrationen vil blive fjernet og den illustration, der ligger øverst i stablen for projektoren, vises igen hvis der er nogen. Det er nødvendigt at angive illustrationsnavnet.
bag	illustrationen vil blive fjernet, når der flyttes til en anden boks.
sluk	Alle illustrationer, også de stablede, vil blive fjernet for den pågældende projektor. Angives ingen projektor vil alle illustrationer blive fjernede.
kort	Illustrationen vil blive fjernet ved næste se kommando.

Projektører

th	Illustrationen vises til højre
tv	Illustrationen vises til venstre
ct	Illustrationen vises i centreret

Man kan ikke bruge varianterne i flæng.

Størrelse:

Man kan kun bruge én af varianterne for størrelse ad gangen. Angiver man ingenting for størrelse og der allerede er vist en illustration, bliver det dennes størrelse, der gælder. Ellers svarer det til at skrive **lille**

Visning:

stp og **sluk** hvis **stp** og **sluk** bruges uden at angive en projektor, så vil den virke for alle projektorer.

Sluk kan ikke bruges sammen med et illustrationsnavn. Angives hverken illustrations navnet eller nogen varianter svarer det til **sluk**

Placering:

Kun én af **th**, **tv**, **ct**, kan bruges ad gangen. Angiver man ingenting for placering, svarer det til at skrive **ct**

PicturesInYourHead. Vil afgive autohændelsen *\$nyboks*, hver gang avataren kommer ind i en ny boks. Laver man et filter, der forstår denne og kobler en **illu** kommando på, så kan man starte en baggrundsillustration, der vises så længe avataren er i boksen. F.eks. vil

```
<$nyboks/illu hav (mel ct)
```

kunne bruges når avataren går ud af en boks, der er en hule og befinder sig i en boks. der er en strand. Illustrationen vil blive vist centreret i vandrings og fylde halvdelen af det.

Men forlader avataren nu den boks, der er en strand og går tilbage igen til hulen, så ville det være godt, hvis havillustrationen forsvandt samtidigt. Det ordner man ved varianten **bag**

```
<$nyBoks/illu hav (ct mel bag). flyt>
```

bag varianten bevirker at **flyt** kommandoen fjerner illustrationen, når boksen forlades.

I **lyd** kommandoen er flere anvendelser af *\$nyboks* beskrevet.

Muligheder:

illu {illustrationsnavn} ({liste af varianter}) {[Tekst der vises]} sætst(liste af sætstykker der ændres)).

Sætstykkerne ændres for den boks, der forstår hændelsen.
Bemærk at illustrations filen skal findes i under mappen: \illustrationer. Se afsnittet interessante filer i **PicturesInYourHead** byggevejledning.
Bemærk også at navnet på Illustrations filen ikke angives, ved **sluk** varianten.

Eksempler:

illu Plateau (mel bag ct)

Vil vise illustrationen Plateau, som fylder halvdelen af vandrings vinduet, centreret.
Illustrationen forsvinder, når boksen forlades ved en **flyt** kommando

*<se&rede>vis [Et stykke oppe kan man, mellem fyrregrenene ,se havørnens sjuskede rede].
illu rede (stabl kort)>*

På grund af varianten **stabl** vises illustrationen **rede** ovenpå plateau illustrationen og på grund af **kort** varianten vil redeillustrationen forsvinde ved næste **se** kommando og plateau illustrationen komme igen. Man kan godt have mange **stabl** illustrationer ovenpå hinanden i en stabel, de underliggende billeder vil så komme frem efterhånden, som de overliggende billeder i stablen fjernes.

<\$sluk/illu(sluk)>

Vil slukke for alle projektorer og glemme alt om, hvilke illustrationer der ligger i dem.

Kommando: **vent**

Virkning:

Kommandoen vil bevirke at **PicturesInYourHead** venter den angivne tid. Tiden angives i sekunder eller millisekunder mSek. (Der går 1000 millisekunder på et sekund). Angives ingen enhed, så forstår kommandoen det som sekunder.

Medens der ventes, vil opdateren ikke kunne taste noget, men lydende vil stadig blive afspillet og illustrationer og tekst vist.

Muligheder:

vent ventetid {enhed} {[Tekst der vises]} sætst(liste af sætstykker der ændres)}.
Sætstykkerne ændres for den boks, der forstår hændelsen.

Eksempler:

bræk&lås&op/vent 20 [Øjeblik- jeg skal lige finde ud af det]>

Anvendelse:

Denne kommando bruges til at lave ventetider, hvor opdateren ikke kan taste noget.

Den kan også være god at bruge i en sammensat processbeskrivelse imellem kommandoerne, hvis der sker en hel del, så opdateren får det hele med. f.eks. lige før en afsluttende flyt kommando.

Kommando: **anbring**

Virkning:

Anbring anbringer en boks i tingbeholderen for en anden boks, der f.eks. er udenfor hørevidde. Typisk bruges kommandoen ved opstarten af verdenen til at placere de ting verdenen skal indeholde, i de bokse hvor de sidenhen skal kunne findes. Se **Verden startes**. I byggevejledningen for **PicturesInYourHead**.

Kommandoerne **læg** og **tag** anvendes normalt i selve historien.

Muligheder:

anbring boksnavn i enAndenBoks {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}.

Bemærk at **boksnavn** og **enAndenBoks** skal være oprettede. Sætstykkerne ændres for **boksnavn**. Altså for den boks der anbringes.

Eksempler:

<\$init/anbring sølvetui i hule2>

*<drage&angrib&spyd/vis[dragen slynger dig rundt så du taber alle dine våben].
anbring kniv i steppe1
anbring spyd i lillesø*

Anvendelse:

Ved opstart af verdenen. Typisk anbringes denne og lignende 'konfigurerings' kommandoer i boksen **_init**. I det **PicturesInYourHead**, når verden startes, placerer avataren i boksen **_init** og sender autohændelsen **\$init** ind i verdenen i det håb, der er en boks som forstår denne autohændelse.

I det andet eksempel mislykkes et angreb på en drage og avataren skal så finde sine våben igen i de omliggende bokse.

Opholder en boks sig allerede i en tingbeholder for en anden boks, vil den blive fjernet fra tingbeholderen her, før **anbring** sætter den i en ny boks. Kommandoen **fjern** vil i modsætning hertil fjerne en boks fra en tingbeholder uden at sætte den i en anden tingbeholder. Og så kan den fjernede boks ikke findes igen nogen steder, før **anbring** kommandoen anvendes til at sætte den fjernede boks på plads igen et nyt sted.

Udover at **anbring** bruges når verdenen startes op, kan den anvendes i selve historien, når en ting skal anbringes i bokse, der er udenfor hørevidde.

Kommando: fjern

Virkning:

fjern bruges til at fjerne en boks fra tingbeholderen i en anden boks, der f.eks. er udenfor hørevidde.

Muligheder:

fjern boksnavn fra enAndenBoks {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}.

Bemærk at **boksnavn** og **enAndenBoks** skal være oprettede. Sætstykkerne kan ændres for **boksnavn**. Altså for den boks der fjernes.

Eksempler:

*<drage&angrib&spyd/vis[dragen slynger dig rundt så du taber alle dine våben].
fjern kniv fra vandreren.
fjern spyd fra vandreren.*

>

Anvendelse:

fjern bruges i situationer hvor en boks skal fjernes helt og ikke bare skifte mellem at ligge i tingbeholderen for den aktive boks og at ligge i tingbeholderen for avataren. (Hertil bruges kommandoerne **tag** og **læg**). Der kan f.eks. ske det, at avataren bliver slynget rundt af en drage og alle hendes våben forsvinder. Skal hun kunne finde våbnene igen, bruges **anbring** kommandoen i stedet for **fjern**. Skal en fjernet boks senere komme til syne igen et eller andet sted i verden, så bruges **anbring** kommandoen også. Som for **anbring** er **fjern** god at anvende, til at fjerne ting der ligger i tingbeholderen for bokse, der er udenfor hørevidde.

Kommando: sæt

Virkning:

sæt sætter de angivne sætstykker for kulisserne i boksen boksnavn, som f.eks er udenfor hørevidde. Sæt anvendes typisk ved opstart af verden. Ellers bruges **sætst(liste af sætstykker der ændres)** i de øvrige kommandoer til at ændre sætstykkerne i en boks.

Muligheder:

sæt sætst(liste af sætstykker der ændres) i boksnavn {[Tekst der vises]} {sætst(liste af sætstykker der ændres)}.

Bemærk at **boksnavn** skal være oprettet.

Sætstykkerne ændres for **boksnavn**. Men de sætstykkeændringer, der angives i den sidste **liste af sætstykker der ændres** vil blive ændret i den boks, der forstod hændelsen.

Eksempler:

```
<$init/
...
  sæt tilst(r-1 n-2 s-3) i dragen
...
>
```

Eksemplet viser at sætstykkerne for kulisserne **r**, **n** og **s** sættes til 1,2 og 3 i boksen **dragen**

Anvendelse:

sæt anvendes til at sætte sætstykkerne på plads når verden startes. Se **Verden startes i** byggevejledningen for **PicturesInYourHead**. På denne måde kan man sikre sig, at sætstykkerne står rigtigt, når avataren kommer ind i boksene på sin vandring og at ting, han støder på, præsenterer sig korrekt. F.eks. at en kasse der skal åbnes, har sit låg lukket, når avataren møder den for første gang. Når avataren går rundt i verden, er der også brug for at ændre sætstykker på rum og ting, men det gøres mest bekvemt ved at bruge **liste af sætstykker der ændres** i kommandoerne.

I sjældne tilfælde får man brug for at ændre sætstykkerne i bokse udenfor hørevidde.

For eksempel kunne avataren opholde sig i en boks, der er et kontrolrum for en dæmning.

Hvor han så tager i et håndtag, der åbner for dæmningens sluseporte, når han derefter går ud, opdager han at alt er oversvømmet af det strømmende vand.

Kommando: **målEr**

Virkning:

målEr anvendes når verden startes op til at angive, hvilken boks der er avatarens mål. Og hvilken af verdenens bokse der er avataren. Ifald denne kommando ikke bruges, vil **PicturesInYourHead** mene at avataren er den boks der hedder 'digSelv' og at målboksen hedder `_goal`. Målboksen anvendes af **hjælp** kommandoen. Skal den give en fornuftig hjælp, så må den jo vide hvad avatarens mål er.

Muligheder:

målEr boksnavn avatarEr enAndenBoks {[Tekst der vises]} sætst(liste af sætstykker der ændres)}.

Bemærk at **boksnavn** og **enAndenBoks** skal være oprettede.

Sætstykkerne kan ændres for **enAndenBoks**. Altså for avataren. Vi kunne f.eks. sætte hende til at sove ved historiens begyndelse. Så må opdageren først finde ud af at få hende vækket.

Obs: *ændres navnet på den boks der er avataren når historien er i gang, kan der ske mange underlige ting.*

Eksempler:

```
<$init/målEr prinsessekammer avatarEr digSelv
```

Anvendelse:

Anvendes når verden startes. Se **Verden startes** i byggevejledningen for **PicturesInYourHead**. Her er også forklaret hvordan man sætter mellemliggende mål op se afsnittet: **Point, niveauer og kommandotricks** i byggevejledningen.

Planlagte kommandoer

Dette afsnit indeholder en oversigt over planlagte kommandoer som **PicturesInYourHead** senere vil blive udvidet med. De kan derfor ikke bruges i denne udgave af **PicturesInYourHead** men vil i en eller anden udformning komme i en senere udgave

Kommando: kan

Virkning:

Undersøger om en bestemt autohændelse kan forstås af de filtre der er indenfor hørevidde. Den er ikke indlysende, og der skal også bruges en mrk kommando i boksen der forstås jeg tror vi skal samle mere erfaring først - bl. A med træthed hos avatar.

Muligheder:

kan {boksnavn} (liste af autohændelsesfiltre) [{succetekst}/{fejltekst}] {sætst(liste af sætstykker der ændres)}

liste af autohændelsesfiltre skal medtages ligeså [{succetekst}/{fejltekst}] men succetekst og fejltekst kan udelades. Autohændelsesfiltrene skal have et + tegn eller et - tegn forrest.

Sætstykkerne ændres for den boks som kan kommandoen er anført i.

Eksempler:

< klatr/prøv digSelv ([Du klatrer op ad klippevæggen/(Du er for svag til at klatre)]>
>

Kulissen forstår og gør
<\$frisk \$glad/mrk >
<\$såret/mrk >
\$udmattet/mrk

Eksemplet viser at avataren bliver bedt om at klatre, men er for træt eller for skadet til at det er muligt.

<gå&syd/kan (+\$kanPassere)[Du forlader vejen/Et eller andet blødt og usynligt forhindrer dig i at forlade vejen].flyt>

Eksemplet viser at hvis avataren f.eks. mangler en fortryllet sten, som han ikke har samlet op, så kan han ikke forlade den vej han er på

Den fortryllede sten kan f.eks. have

Kulissen forstår og gør
\$kanPassere/mrk

\$kanpassere/mrk [det glimter fra din tryllesten]

når den er samlet op.

Anvendelse:

Kan spørger enten en bestemt boks indenfor hørevidde eller alle bokse indenfor hørevidde om deres filtre forstår en autohændelse. Gør de det udføres **kan** med succes og succестekst vises og de følgende kommandoer udføres. Forstås autohændelsen ikke, så vises fejltekst. Og de efterfølgende kommandoer frem til en evt. ellers kommando udføres ikke.

Bemærk at autohændelsen selv ikke udføres. Se **tast** kommandoen. Ved f.eks. at anvende **x** kulissen i avataren kan man der oprette en række sætstykker der viser hvordan han har det ved hjælp af autohændelsesfiltre. f.eks. \$frisk, \$træt, \$udmattet,\$dødeligudmattet. **Kan** kommandoen bruges så til at undersøge avatarens tilstand.

Kommando Har og ting

Vil blive udvidet således at man kan angive det boksnavn der skal spørges om hvad dets tingbeholder indeholder. Det vil også blive muligt at anvende logiske udtryk til at kombinere de bokse man spørger om, f.eks {tang&skruetrækker}|hammer hvilket betyder at det er nødvendigt at have en tang og en skruetrækker men at en hammer også kan bruges.

Ord

Dette afsnit indeholder en oversigt over ord der har en speciel betydning i **PicturesInYourHead**.

Avataren

Er det væsen eller den person, som Opdageren styrer rundt i en verden i **PicturesInYourHead**.

Opdageren

Er den virkelige person af kød og blod, der er på opdagelse i en verden i **PicturesInYourHead**.
Og som styrer 'avataren' .

Hændelse

Er en betegnelse for, hvad der sker når noget tastes på tastaturet af Opdageren. Det som arkitekten bestemmer, skal ske når noget tastes angiver han i Process beskrivelsen for hændelsen.

Autohændelse

Er en betegnelse for en hændelse, som ikke bliver tastet på tastaturet men afsendes fra kommandoen **tast** . Autohændelser skal som hovedregel indeholde **\$** tegnet og kan ikke indeholde blanke.

Hændelsesbeskrivelse

Dette er betegnelsen for beskrivelsen af, hvad en hændelse medfører i **PicturesInYourHead**. Hændelsesbeskrivelsen består af en filterdel og en process-beskrivelses del.

Processbeskrivelsen

Dette er en betegnelse for beskrivelsen af, hvad der sker når en hændelse er forstået af et filter. Process beskrivelsen består af kommandoer.

Kommando

Er en enkelt kommando i processbeskrivelsen. Det kan f.eks. være **tag**, **læg**, **vis**, **se**,

Filter

Bestemmer om en bestemt hændelses processbeskrivelse skal udføres. Alle filtre indenfor Avatarens hørevidde lytter til, hvad der bliver tastet og når filteret forstår det tastede, så udføres, hvad der er angivet i processbeskrivelsen, der er knyttet til dette filter.

Aktive boks

Betegner den boks som Avataren befinder sig i.

Tingbeholder

Alle bokse her en tingbeholder, her kan andre bokse placeres. En boks der er placeret i en anden boks kan i denne udgave af **PicturesInYourHead** ikke selv have noget i sin tingbeholder .

Bokse

Verden er opbygget af bokse. Boksene har en øst, vest, **nord**, syd side og en top og bund. Disse sider kaldes kulisser. Desuden har de en kulisser der angiver selve rummet i boksen. Den kaldes rum kulisser og betegnes med **r**. Ydermere kan boksen udstyres med tre ekstra kulisser x,y og z til specielle effekter. Alle bokse har navne. Navnene skal være forskellige. Et boksnavn kan indeholde tegnene **a-å, A-Å** og **0-9**, samt tegnet **_** .

Kulisser

Kulisserne anvendes til at vise, hvad boksen er for noget. En eng, et værelse, eller et dyr, f.eks. en hund. På kulisser beskrives dels, hvad man kan se når man kigger i kulisserens retning, men på kulisser anbringes også en liste af hændelsesbeskrivelser som angiver hvad kulisseren skal forstå, af det opdageren taster. Processbeskrivelsen i hændelsesbeskrivelsen angiver, hvad der skal ske når filteret har forstået det opdageren taster.

Kulisserne anvendes også til at forbinde to bokse til hinanden. Kulisserne består egentligt af sætstykker, som kan skiftes ud.

Sætstykker

Kulisserne er egentligt kun et stillads hvorpå man kan hænge sætstykker. Sætstykkerne kan skiftes ud afhængigt af hvad opdageren taster. Det gamle sætstykke skydes væk og erstattes af et andet sætstykke på samme plads, der viser noget andet. F.eks. Kan sætstykket på loftkulisser skiftes ud ,hvis vejret pludselig skal blive overskyet. En anden hyppig brug er, hvis Avataren f.eks. smadrer et vindue, så kan det sætstykke der viser en væg med et vindue, skiftes ud med et sætstykke der viser at vinduet er smadret. Samtidigt kan det nye sætstykke have en forbindelse til en anden boks, så der nu er en flugtvej hvilket

det tidligere sætstykke, hvor vinduet ikke var smadret, ikke gav mulighed for. Sætstykkerne er faktisk 'arbejdshestene' i **PicturesInYourHead**

Arkitekten

Den person der bygger verdenen op.

Ting

Ting er også bokse, ofte vil man kun anvende r kulissen til at beskrive dem, men alle kulisser kan bruges, skal tingen kunne forvandle sig, eller gå i stykker kan de øvrige kulisser bruges til at beskrive hvordan tingen opfører sig når den er forvandlet til noget andet.

Væsener

Væsener er også bokse, ofte vil man kun anvende r kulissen til at beskrive dem, men alle kulisser kan bruges, er væsenet f.eks. en elefant kan det være praktisk at kunne se den fra flere sider. Avataren selv er også en Boks

Hørevidde

Den korte forklaring er, at det der er indenfor hørevidde er den boks Avataren er i , de ting han har, og de ting der er i den boks han er i og Avataren selv .

Den længere forklaring er:

Hørevidde er en betegnelse for den gruppe filtre, som bliver taget medtaget i undersøgelsen af hvilket filter, der forstår en hændelse. Når opdageren taster noget, så er det ikke alle filtre i alle bokse der bliver afprøvet. Reglen er den at de filtre som Avatarens sætstykker har og de filtre der findes i den aktive boks's sætstykker, samt de filtre der er i sætstykkerne for tingbeholderne for Avataren og tingbeholderen for den aktive boks vil være indenfor hørevidde. Kun filtre for de sætstykker der er indvalgt medtages. Alle andre filtre er 'udenfor hørevidde'. De filtre der anføres i `_default` boksen vil altid blive prøvet efter alle andre filtre, der er tilrådighed. Se afsnittet **`_default` boksen** i **PicturesInYourHead** byggevejledning.

Kontinent

Et kontinent er en samling bokse som er udtrukket af en verden. Et kontinent kan indlæses som en verden. Kontinenter bruges typisk til at gemme gode bidder af en verden, f.eks. et hus med værelser og trapper, et system af huler med fælder. Eller som en samling af enkelte gode bokse f.eks. forskellige slags væsener. Da et kontinent jo mangler nogle bokse, kan man ikke på forhånd regne med, at kunne gå en tur i det, men grupper af dets bokse kan afprøves.

Når man vil lave sin egen biblioteksverden er det en god ide at starte med at lave et kontinent som indeholder de bedste og mest genbrugelige bokse i den verden man danner kontinentet ud fra.