

Beanoves demonstration manual

Introduction

Beamer named **overlay** specifications

- ▶ This rich presentation is made with Beamer
- ▶ Visual effects will appear only on supporting viewers (like Acrobat Reader)
- ▶ 3 parts:
 1. an example with great photos of animals.
 2. How to declare named overlay specifications with \Beanoves,
 3. How to use ?(....) queries in overlay specifications.
- ▶ Some basic knowledge of standard beamer overlay specifications is required.

Beanoves demonstration manual

Beamer facts

Beamer uses overlay specification aware commands to associate material to slides. The command

```
\only<2-4>{TEXT}
```

will display “TEXT” on slides 2, 3, and 4 only.

Using explicit slide numbers is sometimes difficult, incremental specifications like

```
\item<+>{TEXT}  
\item<+(-1)-->{TEXT}
```

may help in linear situations, but this does not fit to next simple example, to which suit **named overlay specifications**.

Beanoves example about animals: Simple items

Slide 1

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example about animals: Simple items

Slide 2

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items

Slide 1

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items

Slide 2

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items

Slide 3

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items

Slide 4

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items

Slide 5

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items

Slide 6

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<-+>{\transparent{0.3}}
Chameleo
\item
\only<-+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<-+>{\transparent{0.3}}
Water
...
\end{itemize}
```

Beanoves example: Uncovered items

Slide 7

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<--+>{\transparent{0.3}}
Chameleo
\item
\only<--+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<--+>{\transparent{0.3}}
Water
...
\end{itemize}
```

Beanoves example: Uncovered items

Slide 8

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<--+>{\transparent{0.3}}
Chameleo
\item
\only<--+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<--+>{\transparent{0.3}}
Water
...
\end{itemize}
```



Beanoves example: Uncovered items

Slide 9

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<--+>{\transparent{0.3}}
Chameleo
\item
\only<--+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<--+>{\transparent{0.3}}
Water
...
\end{itemize}
```



Beanoves example: Uncovered items

Slide 10

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<-+>{\transparent{0.3}}
Chameleo
\item
\only<-+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<-+>{\transparent{0.3}}
Water
...
\end{itemize}
```

Beanoves example: Uncovered items

Slide 11

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<-+>{\transparent{0.3}}
Chameleo
\item
\only<-+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<-+>{\transparent{0.3}}
Water
...
\end{itemize}
```

Beanoves example: Uncovered items

Slide 12

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<-+>{\transparent{0.3}}
Chameleo
\item
\only<-+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<-+>{\transparent{0.3}}
Water
...
\end{itemize}
```

Beanoves example: Uncovered items

Slide 13

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\begin{itemize}
\item Air
\begin{itemize}
\item
\only<-+>{\transparent{0.3}}
Chameleo
\item
\only<-+>{\transparent{0.3}}
Gannet
\end{itemize}
\item
\only<-+>{\transparent{0.3}}
Water
...
\end{itemize}
```

Beanoves example: Uncovered items

Chronology

Beanoves example: Uncovered items

Chronology



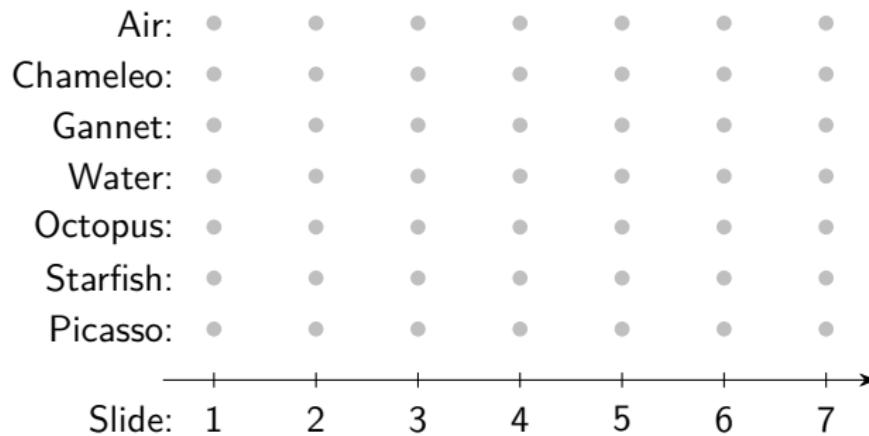
Beanoves example: Uncovered items

Chronology



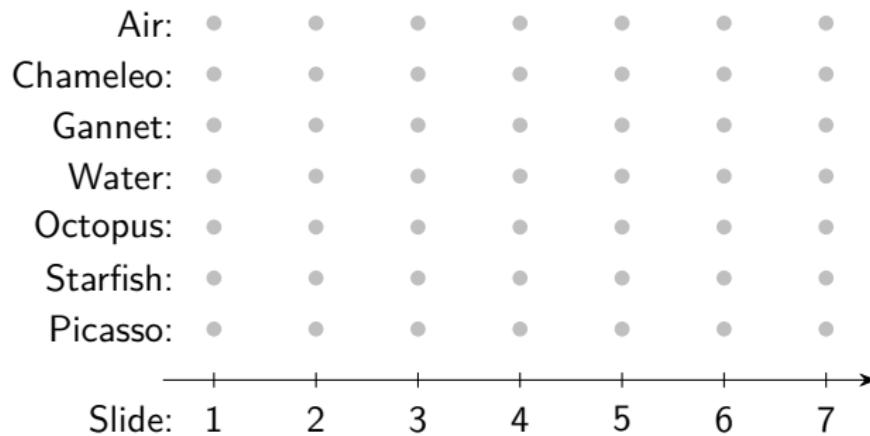
Beanoves example: Uncovered items

Chronology



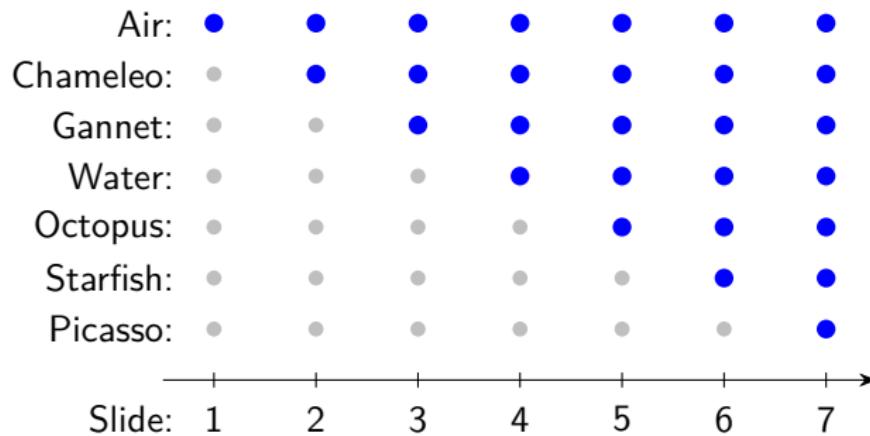
Beanoves example: Uncovered items

Chronology



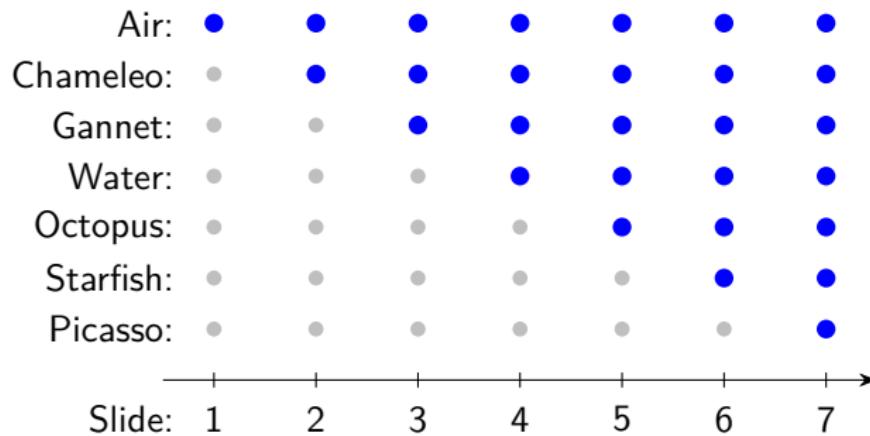
Beanoves example: Uncovered items

Chronology



Beanoves example: Uncovered items

Chronology



Beanoves example: Uncovered items + images

Slide 1

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + in

Slide 2



- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + in

Slide 3

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish



Beanoves example: Uncovered items + in

Slide 4

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish



Beanoves example: Uncovered items + images

Slide 5

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + image



Slide 6

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + images

Slide 7



- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + images

Slide 8

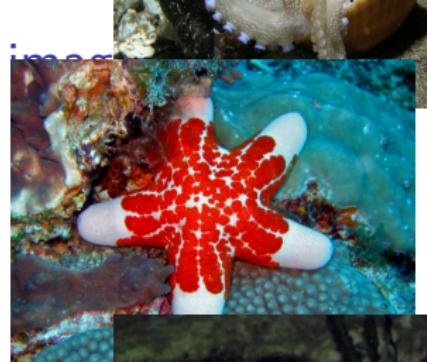


- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + images

Slide 9

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish



Beanoves example: Uncovered item

Slide 10

- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish



Beanoves example: Uncovered items + images

Chronology

Beanoves example: Uncovered items + images

Chronology

Slide: 1 2 3 4 5 6 7 8

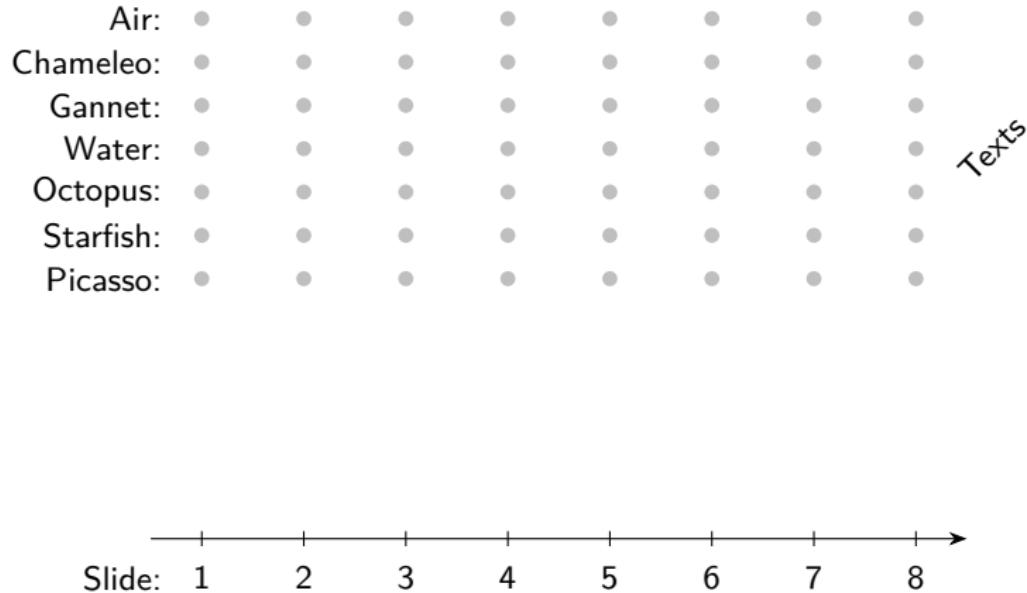
Beanoves example: Uncovered items + images

Chronology

Slide: 1 2 3 4 5 6 7 8

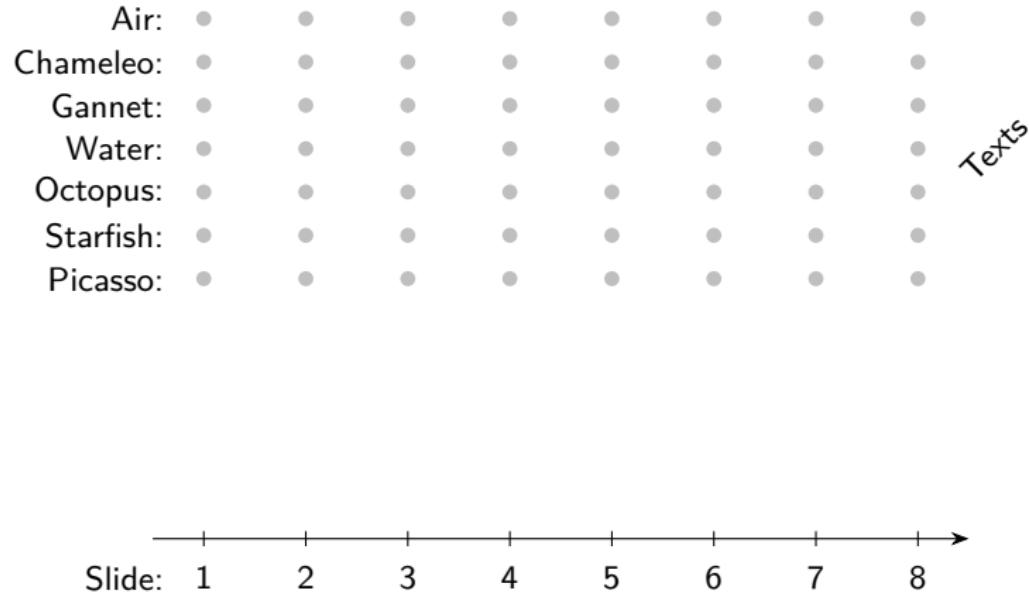
Beanoves example: Uncovered items + images

Chronology



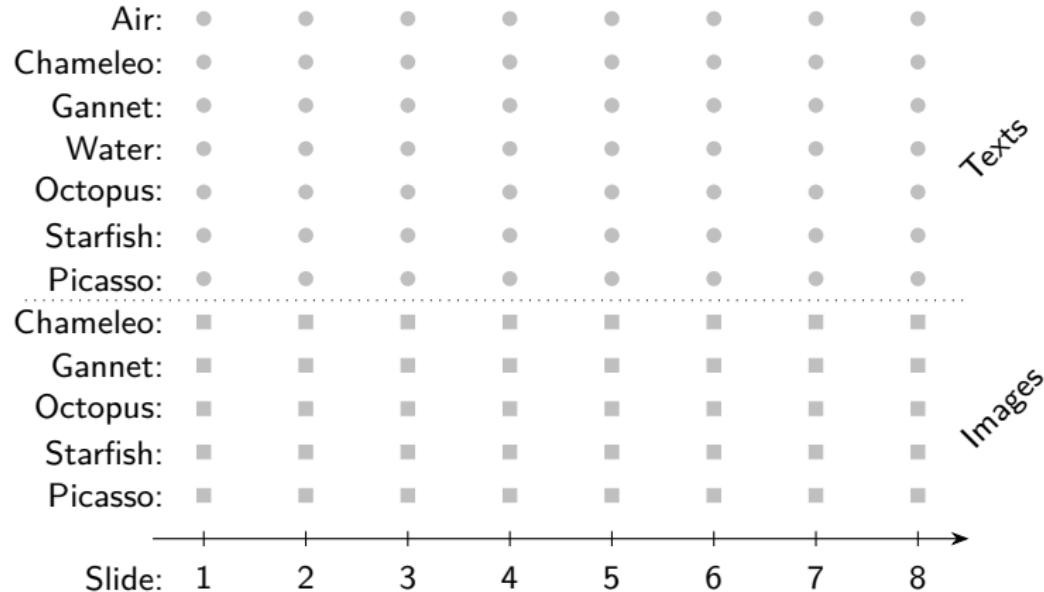
Beanoves example: Uncovered items + images

Chronology



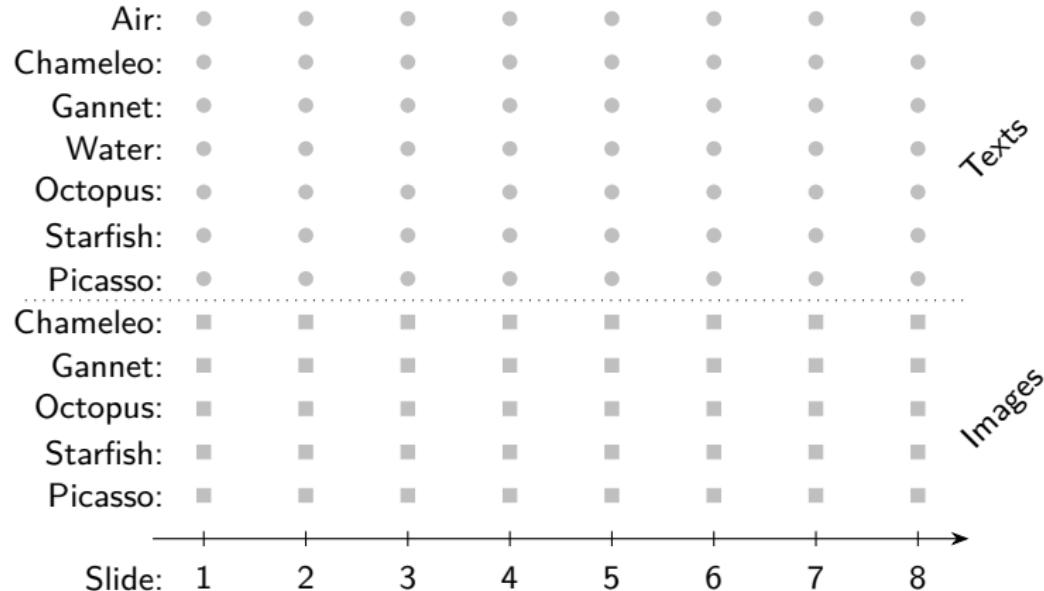
Beanoves example: Uncovered items + images

Chronology



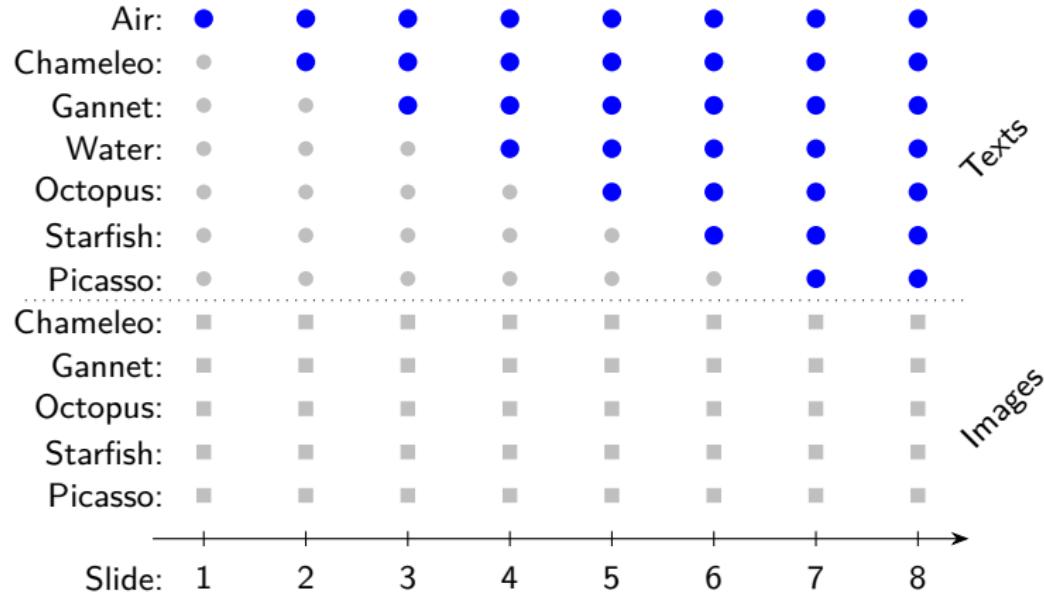
Beanoves example: Uncovered items + images

Chronology



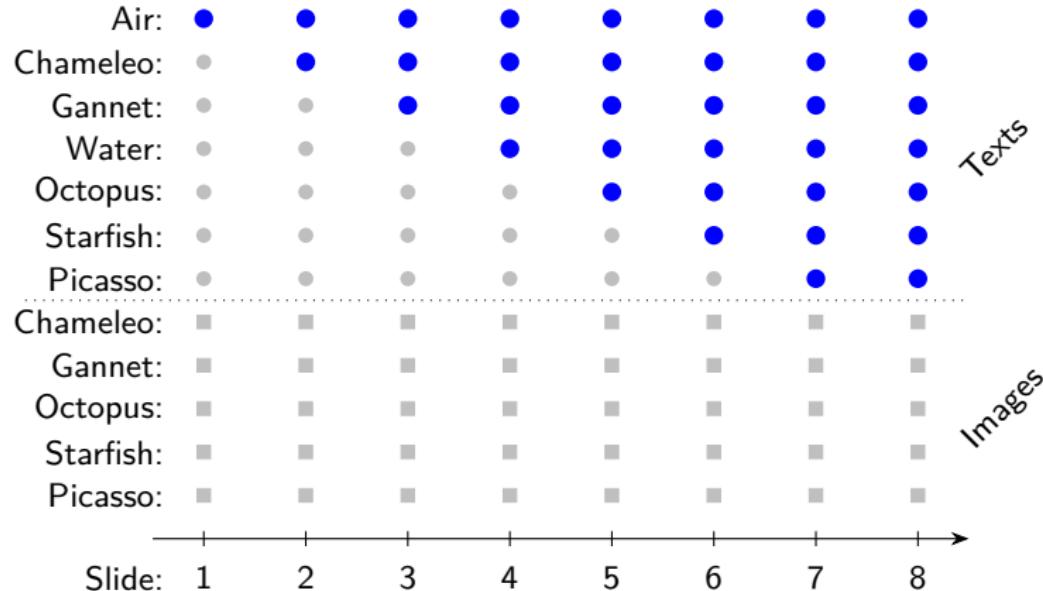
Beanoves example: Uncovered items + images

Chronology



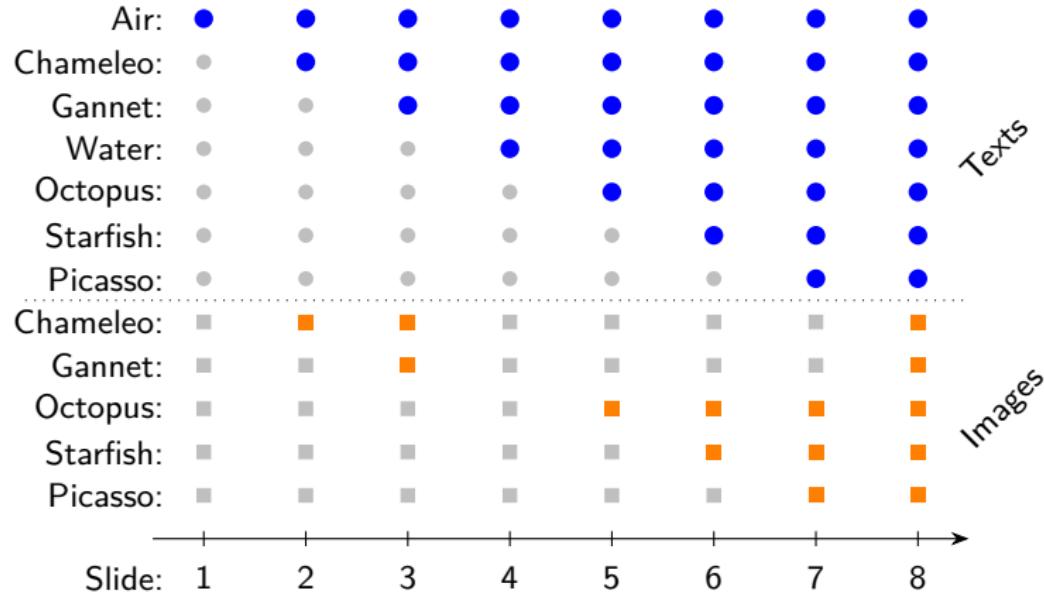
Beanoves example: Uncovered items + images

Chronology



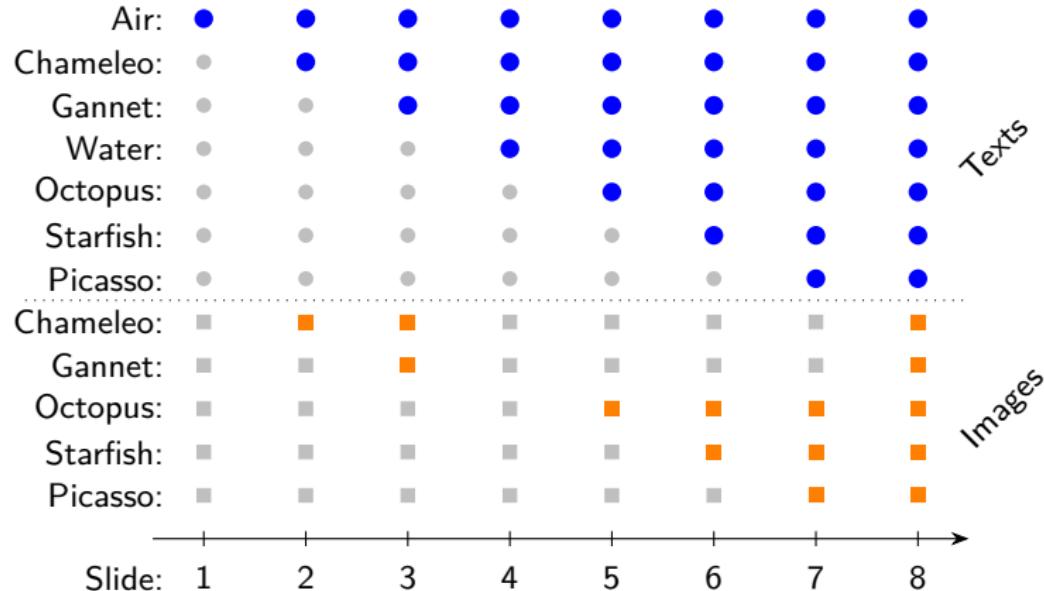
Beanoves example: Uncovered items + images

Chronology



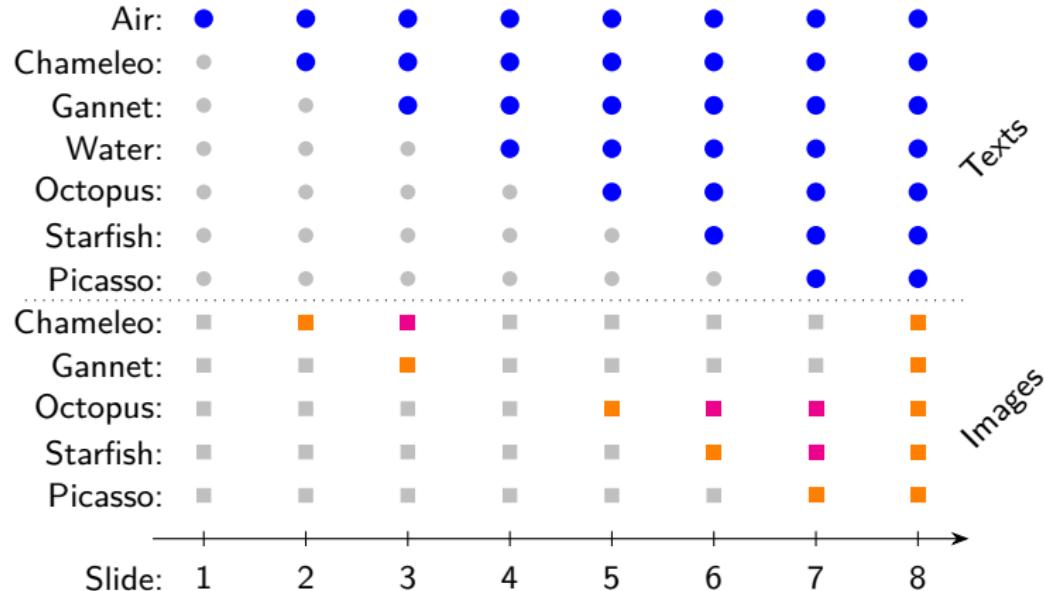
Beanoves example: Uncovered items + images

Chronology



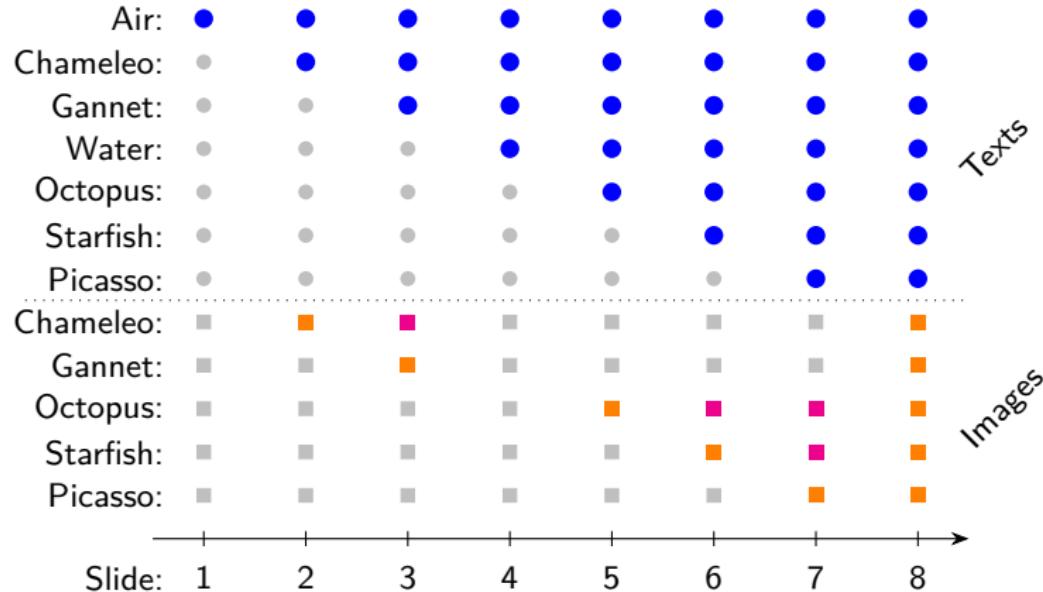
Beanoves example: Uncovered items + images

Chronology



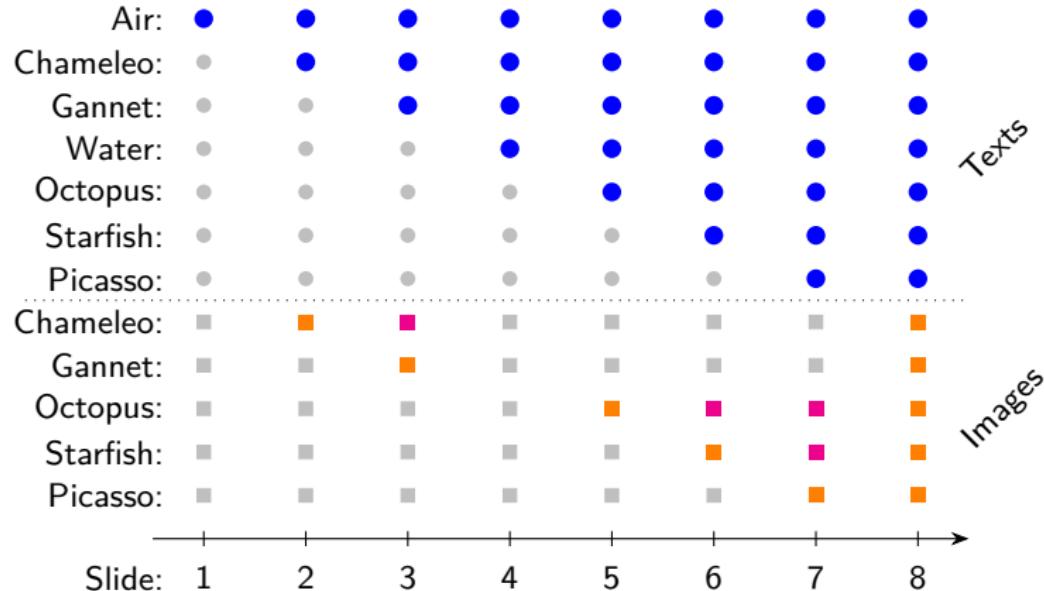
Beanoves example: Uncovered items + images

Chronology



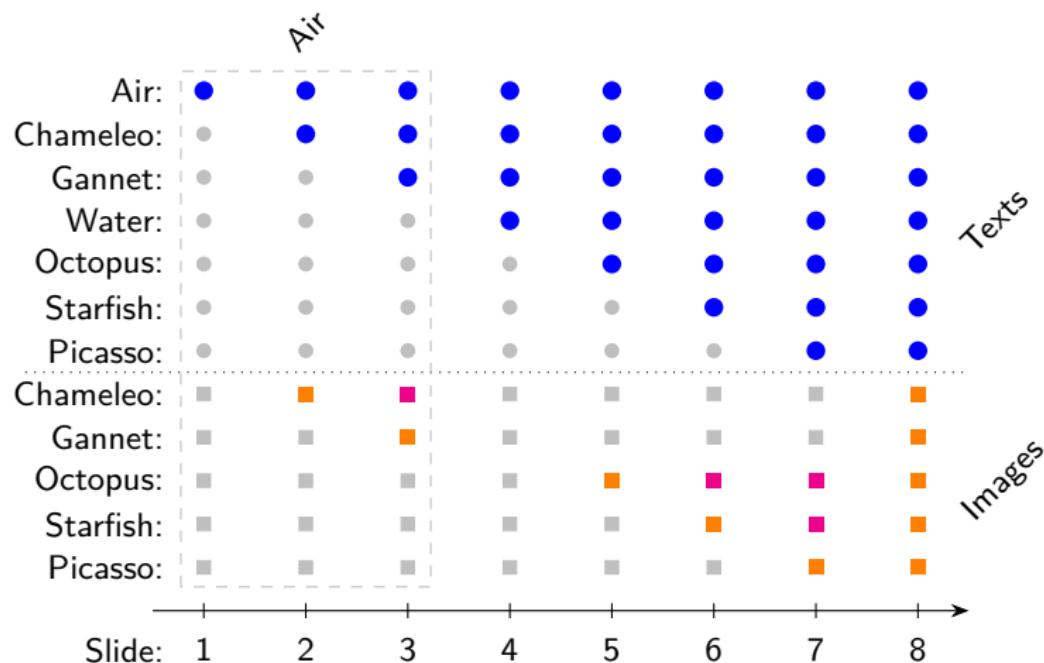
Beanoves example: Uncovered items + images

Chronology



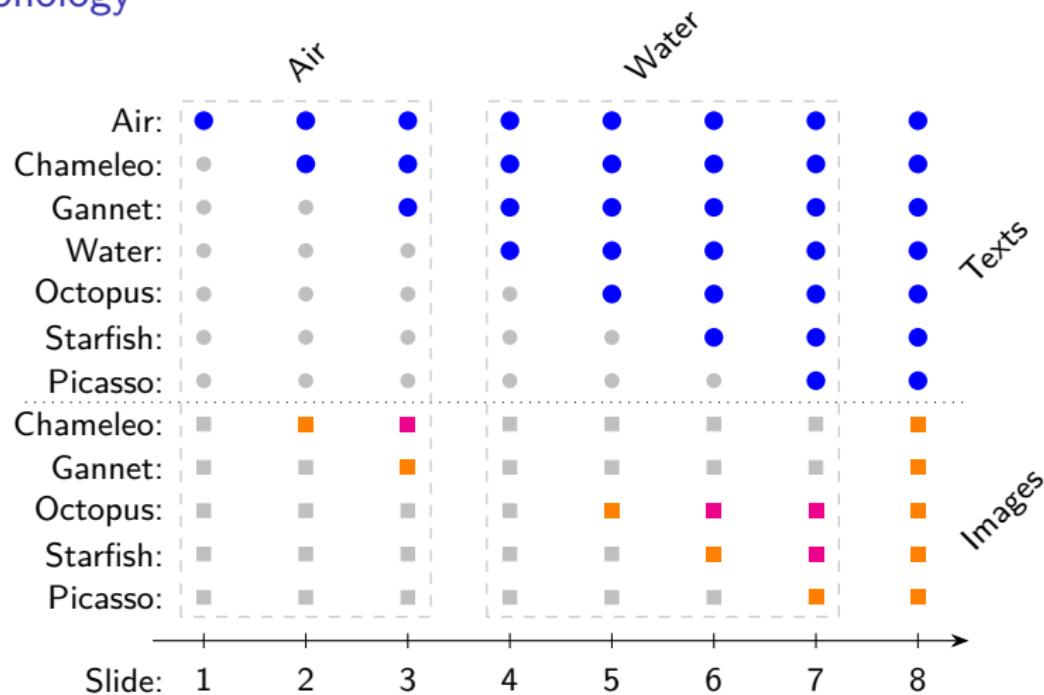
Beanoves example: Uncovered items + images

Chronology



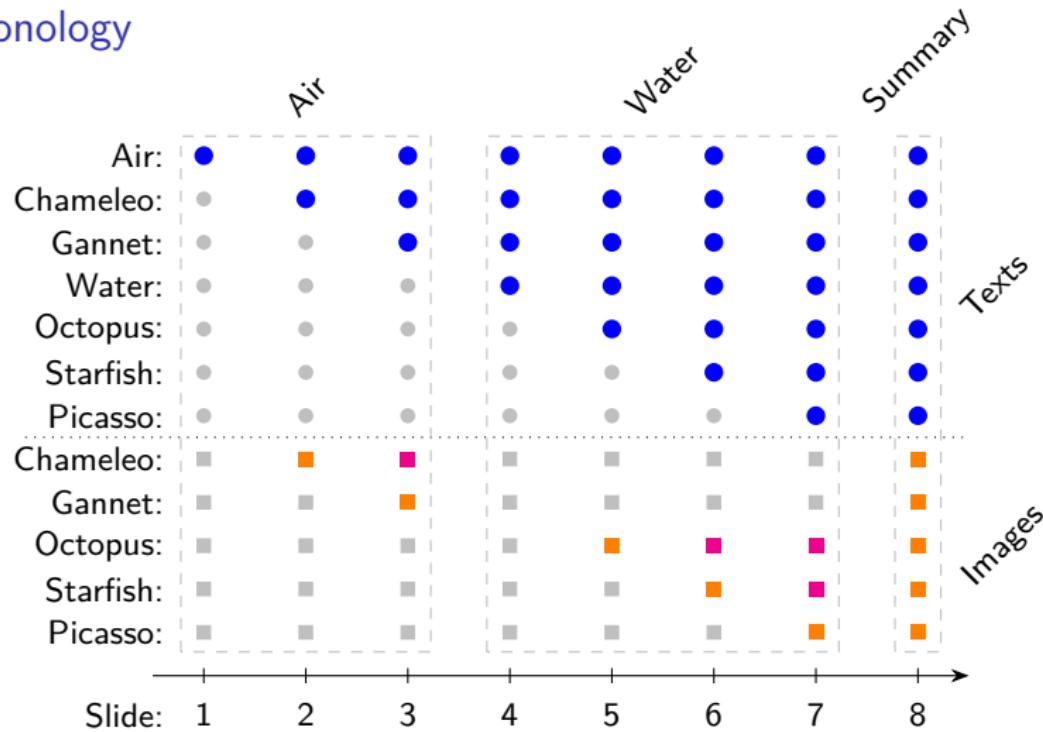
Beanoves example: Uncovered items + images

Chronology



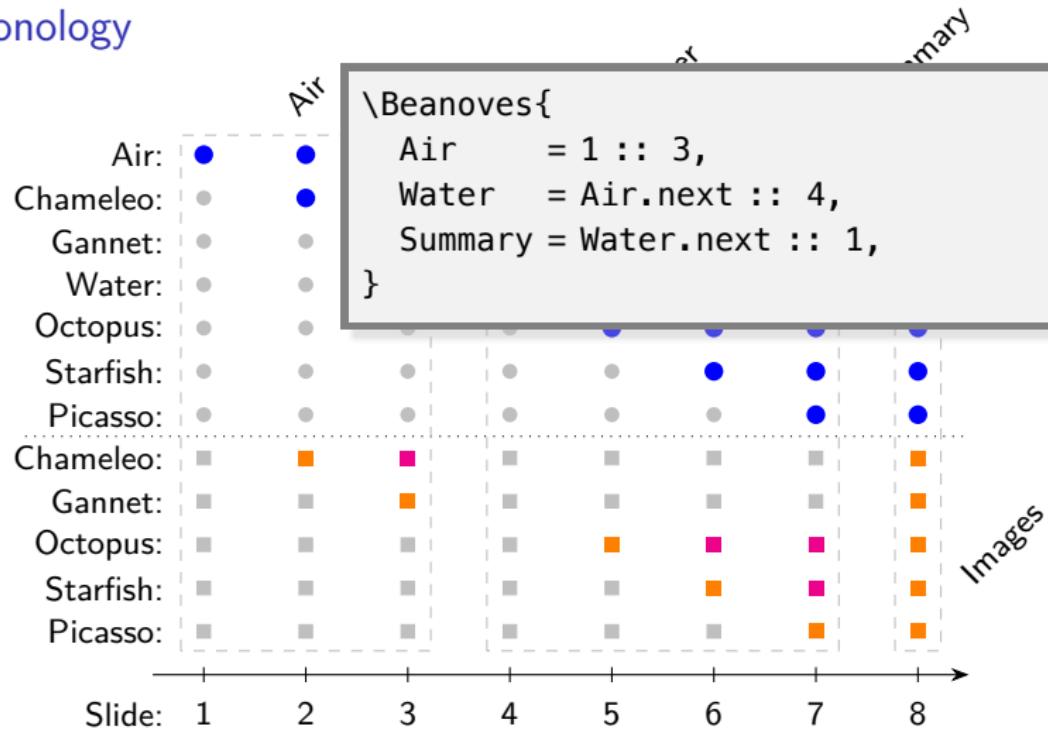
Beanoves example: Uncovered items + images

Chronology



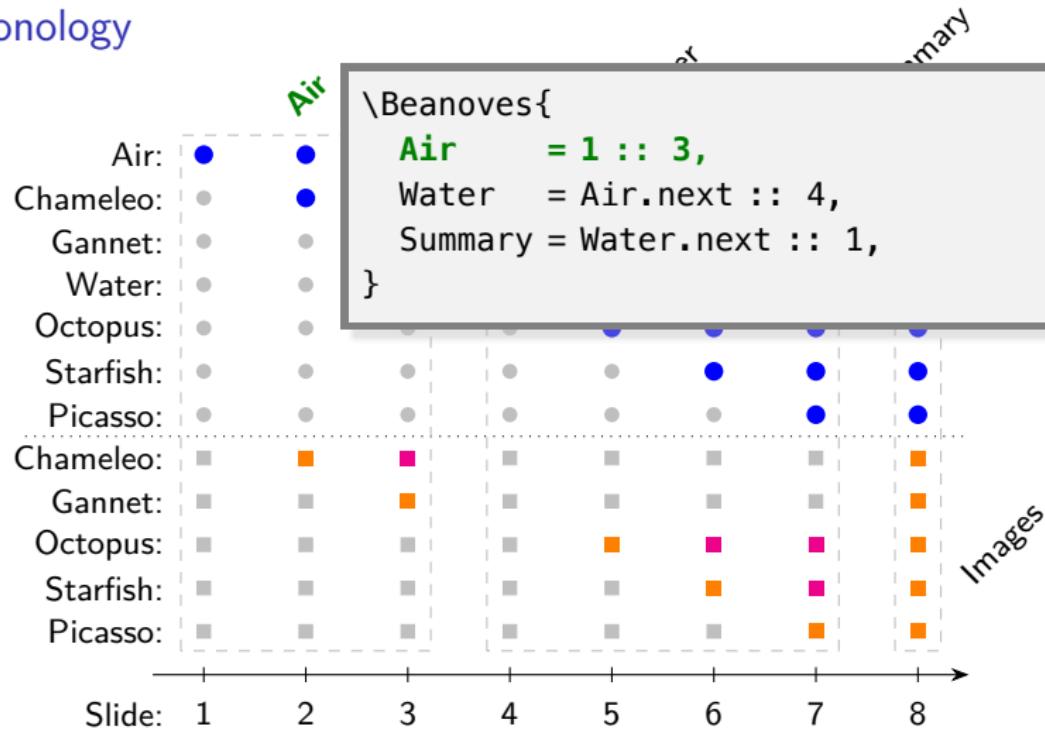
Beanoves example: Uncovered items + images

Chronology



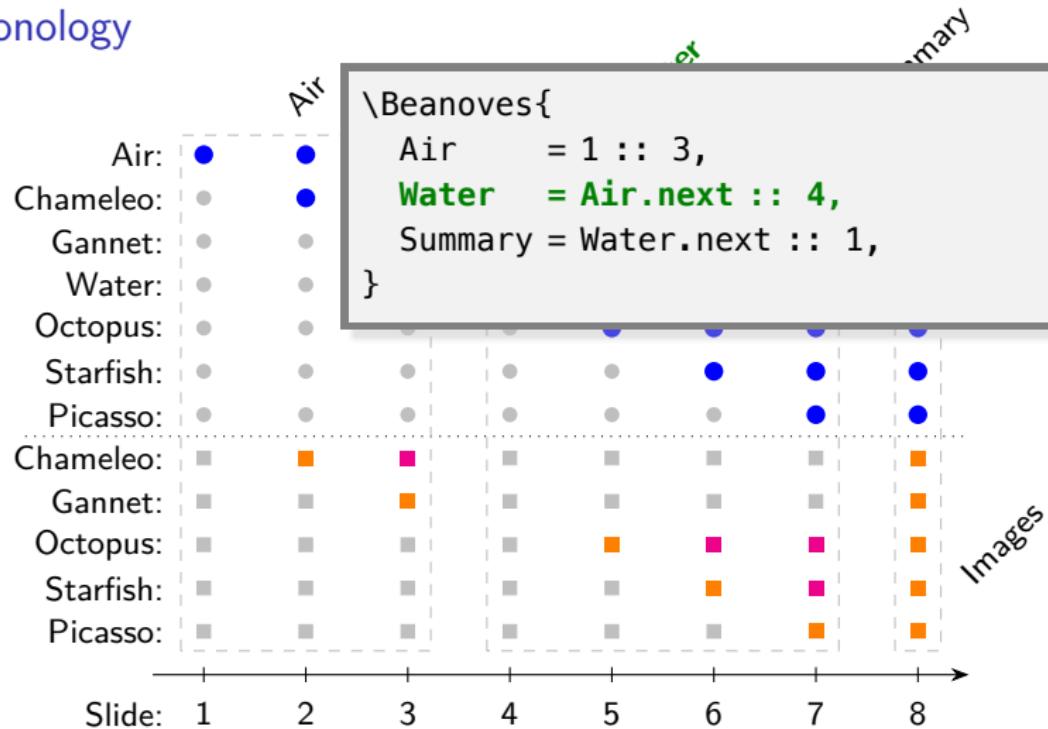
Beanoves example: Uncovered items + images

Chronology



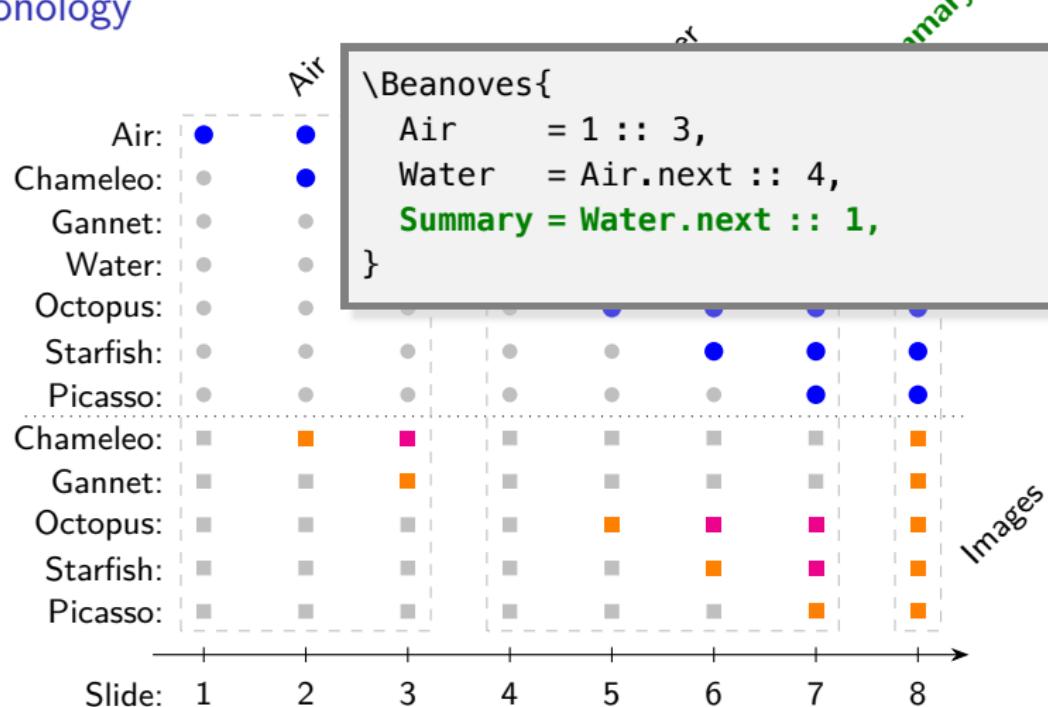
Beanoves example: Uncovered items + images

Chronology



Beanoves example: Uncovered items + images

Chronology

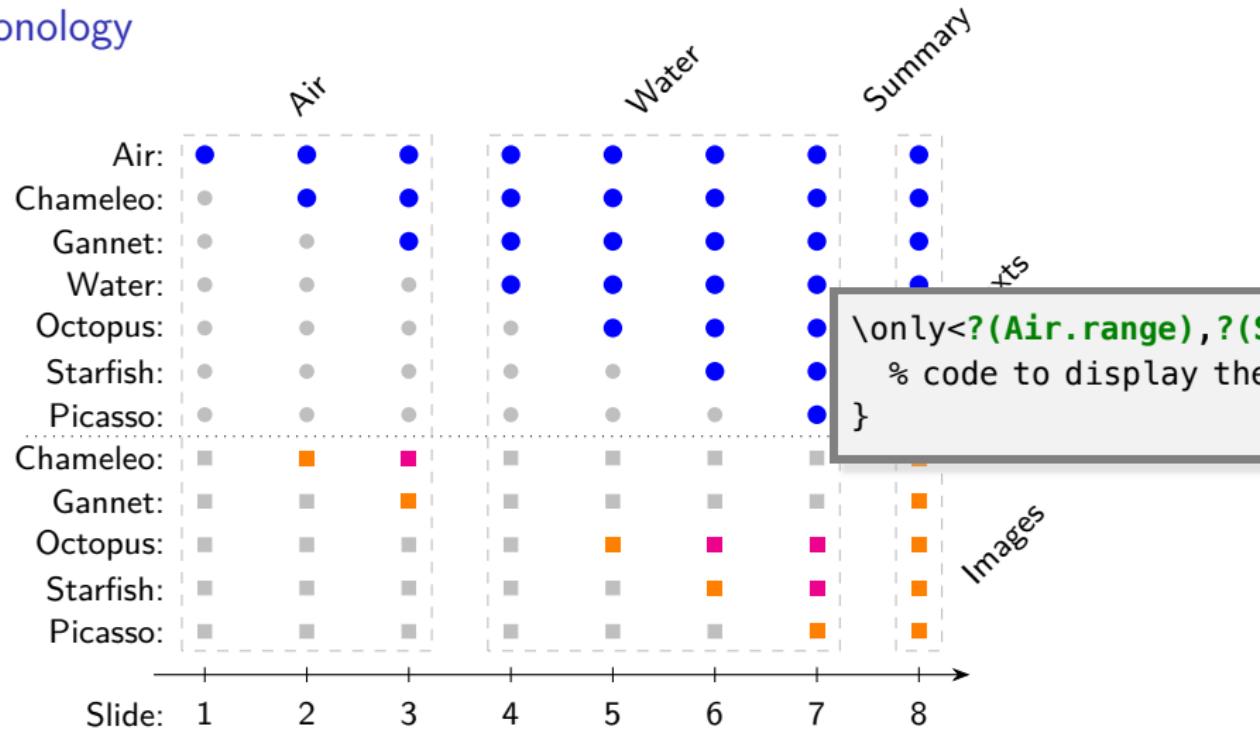


Beanves example: Uncovered items + images

Chronology

Beanoves example: Uncovered items + images

Chronology



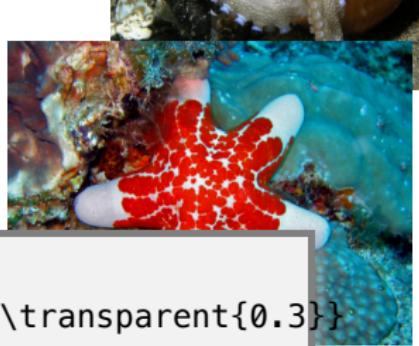
Beanoves example: Uncovered items + images

Step back



- ▶ Air
 - ▶ Chameleo
 - ▶ Gannet
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

Beanoves example: Uncovered items + Step back



- ▶ Air
 - ▶ Cham
 - ▶ Ganne
- ▶ Water
 - ▶ Octopus
 - ▶ Starfish
 - ▶ Picasso fish

```
\visible< ?(Octopus.1) -> {  
    \only < ?(Starfish.1) -> {\transparent{0.3}}  
    % code to display the Octopus...  
}
```

Beanoves manual

Defining named overlay sets

```
\Beanoves{  
    <math_k> = <start_k> :: <length_k>,  
    ...  
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	\leftrightarrow	Integer value
$\langle name_k \rangle .1$	\leftrightarrow	$\langle start_k \rangle$
$\langle name_k \rangle .2$	\leftrightarrow	$\langle start_k \rangle + 1$
$\langle name_k \rangle .\langle i \rangle$	\leftrightarrow	$\langle start_k \rangle + \langle i \rangle - 1$
$\langle name_k \rangle .length$	\leftrightarrow	$\langle length_k \rangle$
$\langle name_k \rangle .next$	\leftrightarrow	$\langle start_k \rangle + \langle length_k \rangle$
$\langle name_k \rangle .last$	\leftrightarrow	$\langle start_k \rangle + \langle length_k \rangle - 1$
$\langle name_k \rangle .previous$	\leftrightarrow	$\langle start_k \rangle - 1$

Beanoves manual

Defining named overlay sets

```
\Beanoves{  
    <math_k> = <start_k> :: <length_k>,  
    ...  
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	\leftrightarrow	Integer value
$\langle name_k \rangle .1$	\leftrightarrow	$\langle start_k \rangle$
$\langle name_k \rangle .2$	\leftrightarrow	$\langle start_k \rangle + 1$
$\langle name_k \rangle .\langle i \rangle$	\leftrightarrow	$\langle start_k \rangle + \langle i \rangle - 1$
$\langle name_k \rangle .length$	\leftrightarrow	$\langle length_k \rangle$
$\langle name_k \rangle .next$	\leftrightarrow	$\langle start_k \rangle + \langle length_k \rangle$
$\langle name_k \rangle .last$	\leftrightarrow	$\langle start_k \rangle + \langle length_k \rangle - 1$
$\langle name_k \rangle .previous$	\leftrightarrow	$\langle start_k \rangle - 1$

Beanoves manual

Defining named overlay sets

```
\Beanoves{
    Air = 1      :: 2,
    Water = Air.next :: 3,
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	↔	Integer value
Air.1	↔	1
Air.2	↔	2
Air. $\langle i \rangle$	↔	$\langle i \rangle$
Air.length	↔	2
Air.next	↔	3
Air.last	↔	2
Air.previous	↔	0

Beanoves manual

Defining named overlay sets

```
\Beanoves{
    Air = 1      :: 2,
    Water = Air.next :: 3,
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	↔	Integer value
Water.1	↔	3
Water.2	↔	4
Water. $\langle i \rangle$	↔	$\langle i \rangle + 2$
Water.length	↔	3
Water.next	↔	6
Water.last	↔	5
Water.previous	↔	2

Beanoves manual

Defining named overlay sets

```
\Beanoves{
    Air = 1      :: 2,
    Water = Air.next :: 3,
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	↔	Integer value
Water.1	↔	3
Water.2	↔	4
Water. $\langle i \rangle$	↔	$\langle i \rangle + 2$
Water.length	↔	3
Water.next	↔	6
Water.last	↔	5
Water.previous	↔	2

Beanoves manual

Defining named overlay sets

```
\Beanoves{
    Air = 1      :: 2, next
    Water = Air.next :: 3,
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	↔	Integer value
Water.1	↔	3
Water.2	↔	4
Water.$\langle i \rangle$	↔	$\langle i \rangle + 2$
Water.length	↔	3
Water.next	↔	6
Water.last	↔	5
Water.previous	↔	2

Beanoves manual

Defining named overlay sets

```
\Beanoves{
    Air = 1      :: 3,
    Water = Air.next :: 3,
}
```

Range starts and lengths are arithmetical expression involving raw integers as well as next ***named overlay references***.

Reference	↔	Integer value
Water.1	↔	4
Water.2	↔	5
Water. $\langle i \rangle$	↔	$\langle i \rangle + 3$
Water.length	↔	4
Water.next	↔	7
Water.last	↔	6
Water.previous	↔	3

Beanoves manual

Overlay specification query

- ▶ Simple specifications
- ▶ Incremental specifications
- ▶ **Specification queries**

```
\only < 4 >      {...}  
\only < 1 - 3 >  {...}
```

```
\only < + >      {...}  
\only < +( <i>) >  {...}
```

Beanoves manual

Overlay specification query

- ▶ Simple specifications
- ▶ Incremental specifications
- ▶ **Specification queries**

```
\only < 4 > {...}  
\only < 1 - 3 > {...}
```

```
\only < + > {...}  
\only < +( <i>) > {...}
```

```
\only < ?(<query>) > {...}
```

Beanoves manual

Overlay specification query

- ▶ Simple specifications
- ▶ Incremental specifications
- ▶ **Specification queries**

```
\only < 4 > {...}  
\only < 1 - 3 > {...}
```

```
\only < + > {...}  
\only < +(〈i〉) > {...}
```

```
\only < ?(〈query〉) > {...}
```

A query may be used in an overlay specification wherever an integer or a range can be.

\only may be replaced by any specification aware command.

Beanoves manual

Overlay specification query syntax

- ▶ Position specifications

```
?(<integer expression with aliases>)
```

- ▶ Explicit range specifications

```
?(<start expression> :: <length expression>)
```

Both integer expressions accept aliases.

- ▶ Logical range specifications with a **range alias**:

$$\langle name_k \rangle.\text{range} \iff \langle name_k \rangle.1 - \langle name_k \rangle.\text{last}$$

where “–” stands for a dash and not a minus sign.

```
?(<name_k>.range)
```

Beanoves manual

Overlay specification query syntax

- ▶ Position specifications

```
?(<integer expression with aliases>)
```

- ▶ Explicit range specifications

```
?(<start expression> :: <length expression>)
```

Both integer expressions accept aliases.

- ▶ Logical range specifications with a **range alias**:

$\langle name_k \rangle.\text{range} \leftrightarrow \langle name_k \rangle.1 - \langle name_k \rangle.\text{last}$

where “–” stands for a dash and not a minus sign.

```
?(<name_k>.range)
```

Range queries and beamer ranges must not be combined like in
(Air.range)-10, leading to the incorrect syntax **1-2-10**.

Beanoves manual

Overlay specification query syntax

- ▶ Position specifications

```
?(<integer expression with aliases>)
```

- ▶ Explicit range specifications

```
?(<start expression> :: <length expression>)
```

Both integer expressions accept aliases.

- ▶ Logical range specifications with a *range alias*:

$\langle name_k \rangle.\text{range} \leftrightarrow \langle name_k \rangle.1 - \langle name_k \rangle.\text{last}$

where “–” stands for a dash and not a minus sign.

```
?(<name_k>.range)
```

Range queries and beamer ranges must not be combined like in
?(Air.range)-10, leading to the incorrect syntax **1-2-10**.

Beanoves manual

Incremental specifications

```
\Beanoves {
    <namek> = <startk> :: <lengthk>,
    ...
} \begin{frame}
```

Each logical overlay range has a **counter**, with dedicated alias and operations. Within a specification query:

- ▶ $\langle name_k \rangle$, with no trailing “.”, is an alias for the **counter**
- ▶ $\texttt{++}\langle name_k \rangle$ stands for the **counter** once incremented by 1
- ▶ $\langle name_k \rangle \texttt{+=}\langle i \rangle$ stands for the **counter** after being incremented by $\langle i \rangle$.
- ▶ $\langle name_k \rangle.\texttt{reset}$ stands for the **counter** once reset to its initial value.

Beanoves manual

Incremental specifications

```
\Beanoves {
    <namek> = <startk> :: <lengthk>,
    ...
} \begin{frame}
```

Each logical overlay range has a **counter**, with dedicated alias and operations. Within a specification query:

- ▶ $\langle name_k \rangle$, with no trailing “.”, is an alias for the **counter**
- ▶ $\texttt{++}\langle name_k \rangle$ stands for the **counter** once incremented by 1
- ▶ $\langle name_k \rangle \texttt{+=}\langle i \rangle$ stands for the **counter** after being incremented by $\langle i \rangle$.
- ▶ $\langle name_k \rangle.\texttt{reset}$ stands for the **counter** once reset to its initial value.

Beanoves manual

Incremental specifications

```
\Beanoves {
    <namek> = <startk> :: <lengthk>,
    ...
} \begin{frame}
```

Each logical overlay range has a **counter**, with dedicated alias and operations. Within a specification query:

- ▶ $\langle name_k \rangle$, with no trailing “.”, is an alias for the **counter**
- ▶ $\texttt{++}\langle name_k \rangle$ stands for the **counter** once incremented by 1
- ▶ $\langle name_k \rangle \texttt{+=}\langle i \rangle$ stands for the **counter** after being incremented by $\langle i \rangle$.
- ▶ $\langle name_k \rangle.\texttt{reset}$ stands for the **counter** once reset to its initial value.

Beanoves manual

Incremental specifications

```
\Beanoves {
    <namek> = <startk> :: <lengthk>,
    ...
} \begin{frame}
```

Each logical overlay range has a **counter**, with dedicated alias and operations. Within a specification query:

- ▶ $\langle name_k \rangle$, with no trailing “.”, is an alias for the **counter**
- ▶ $\text{++}\langle name_k \rangle$ stands for the **counter** once incremented by 1
- ▶ $\langle name_k \rangle \text{+=}\langle i \rangle$ stands for the **counter** after being incremented by $\langle i \rangle$.
- ▶ $\langle name_k \rangle.\text{reset}$ stands for the **counter** once reset to its initial value.

Beanoves manual

Incremental specifications

```
\Beanoves {
    <namek> = <startk> :: <lengthk>,
    ...
} \begin{frame}
```

Each logical overlay range has a **counter**, with dedicated alias and operations. Within a specification query:

- ▶ $\langle name_k \rangle$, with no trailing “.”, is an alias for the **counter**
- ▶ $\text{++}\langle name_k \rangle$ stands for the **counter** once incremented by 1
- ▶ $\langle name_k \rangle \text{+=}\langle i \rangle$ stands for the **counter** after being incremented by $\langle i \rangle$.
- ▶ $\langle name_k \rangle.\text{reset}$ stands for the **counter** once reset to its initial value.

Beanoves manual

Incremental specifications

```
\Beanoves {
    <namek> = <startk> :: <lengthk>,
    ...
} \begin{frame}
```

Each logical overlay range has a **counter**, with dedicated alias and operations. Within a specification query:

- ▶ $\langle name_k \rangle$, with no trailing “.”, is an alias for the **counter**
- ▶ $\texttt{++}\langle name_k \rangle$ stands for the **counter** once incremented by 1
- ▶ $\langle name_k \rangle \texttt{+=}\langle i \rangle$ stands for the **counter** after being incremented by $\langle i \rangle$.
- ▶ $\langle name_k \rangle.\texttt{reset}$ stands for the **counter** once reset to its initial value.

Beanoves manual

Incremental specifications in practice

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}

\begin{itemize} [ <?(Water++)-> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}

\begin{itemize} [<?(Water++)-> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}

\begin{itemize} [ <?(Water++)-> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish

```
\Beanoves{  
    Water = 1 :: 3,  
}  
\begin{frame}  
  
\begin{itemize} [ <?(Water++)-> ]  
    \item Octopus  
    \item Starfish  
    \item Picasso fish  
\end{itemize}  
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish
- ▶ Picasso fish

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}

\begin{itemize} [ <?(Water++)-> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish
- ▶ Picasso fish

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}
\setbeamercovered{ transparent = 30 }
\begin{itemize} [ <?(Water++)> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish
- ▶ Picasso fish

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}
\setbeamercovered{ transparent = 30 }
\begin{itemize} [ <?(Water++)> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish
- ▶ Picasso fish

```
\Beanoves{
    Water = 1 :: 3,
}
\begin{frame}
\setbeamercovered{ transparent = 30 }
\begin{itemize} [ <?(Water++)> ]
    \item Octopus
    \item Starfish
    \item Picasso fish
\end{itemize}
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish
- ▶ Picasso fish

```
\Beanoves{  
    Water = 1 :: 3,  
}  
\begin{frame}  
\setbeamercovered{ transparent = 30 }  
\begin{itemize} [ <?(Water++)> ]  
    \item Octopus  
    \item Starfish  
    \item Picasso fish  
\end{itemize}  
\end{frame}
```

Beanoves manual

Incremental specifications in practice

- ▶ Octopus
- ▶ Starfish
- ▶ Picasso fish

```
\Beanoves{  
    Water = 1 :: 3,  
}  
\begin{frame}  
\setbeamercovered{ transparent = 30 }  
\begin{itemize} [ <?(Water++)> ]  
    \item Octopus  
    \item Starfish  
    \item Picasso fish  
\end{itemize}  
\end{frame}
```

Beanoves manual

Why aliases are helpful

- ▶ As soon as one leaves basic frame layouts to make presentations more attractive and efficient, then beanoves aliases should come into play.
- ▶ One can organize the slides with logical names for a better understanding: aliases and integer expressions rather than raw integers make specifications more explicit
- ▶ Adding or removing a slide from one slide range does not significantly affect the other slide ranges.

Beanoves manual

Where to go next

- ▶ The official repository of this document is <https://github.com/jlaurens/beanoves/blob/demo/demo.pdf>
- ▶ The beanoves manual available from <https://github.com/jlaurens/beanoves/blob/main/source/beanoves.pdf> or the \TeX distribution contains more technical description.
- ▶ Visit the official repository at <https://github.com/jlaurens/beanoves/>. See in particular the demo directory and the examples directory inside.