

Creating fancy user interfaces



with Edje



Bruno Dilly

I'm a developer of the **EFL**
and have developed projects for three years with it.

With many contributions to free software projects,
I'm now Senior Software Developer at

ProFUSION
embedded systems

This presentation

What is Edje ?

Where is Edje ?

How Edje works ?

What can I do with Edje ?

- graphical design & layout library.
- powerful.
 - scripts
 - externals
- easy to use.
- used by Elementary and WebKit EFL port.

EDJE ON TIZEN ARCHITECTURE

APPLICATION

Web Applications

CORE

API

Web API

COMPONENT

Application Framework

Graphics & UI

Multimedia

Web

Messaging

Location

Security

System

Base

Connectivity

Telephony

PIM

KERNEL

Linux Kernel

EDJE ON EFL STACK

ELEMENTARY

WEBKIT EFL

ECORE

EDJE

EVAS

EINA



THE NAME “EDJE”

- library.
- source.
- binary.

HOW EDJE WORKS

EDJE
source

EDJE
binary

EDJE
library

.edc

<code>
<code>
<code>

→
edje_cc

.edj

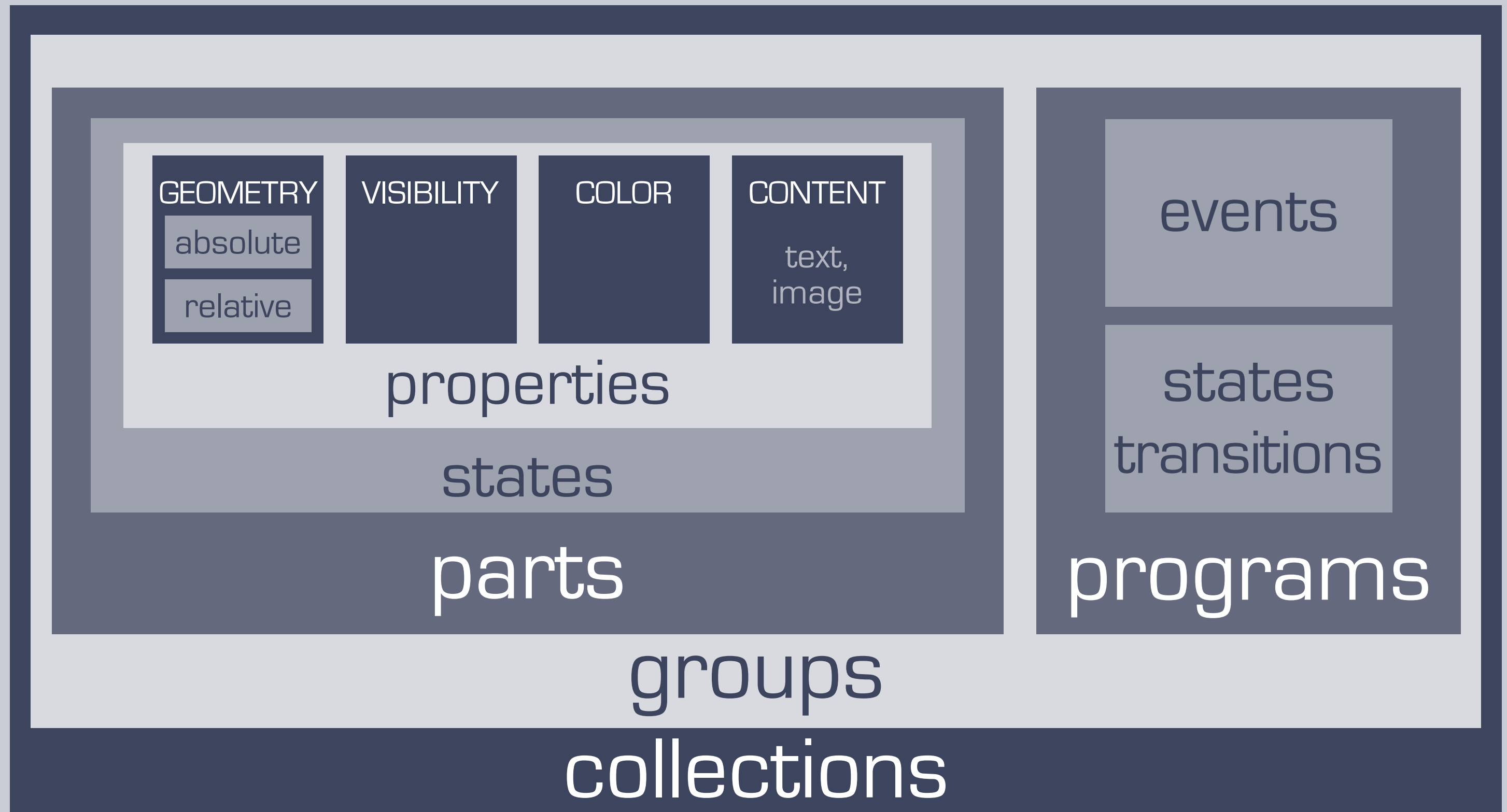
0110101
1010110
0001011

→
edje_player
(or other app)



(displayed theme)

EDJE CONCEPT



- descriptive language.
- verbose.
- blocks and properties.

```
block {  
    block {  
        property: value;  
        property: value;  
        block {  
            property: value;  
        }  
    }  
}
```

- each part is an Evas Object.
- parts:
 - rect
 - image
 - text / textblock
 - swallow
 - group
 - externals

- react to events.
- change part's states.
 - animation
- emit signals.
- run scripts.
- called by. ←
 - signals (code or theme)
 - scripts
 - other programs

SIMPLE EXAMPLE

```
collections {  
  group {  
    name: "simple_example";  
    min: 400 400;  
    parts {  
      ...  
    }  
  }  
}
```


SIMPLE EXAMPLE

```
programs {
```

```
    ...
```

```
}
```

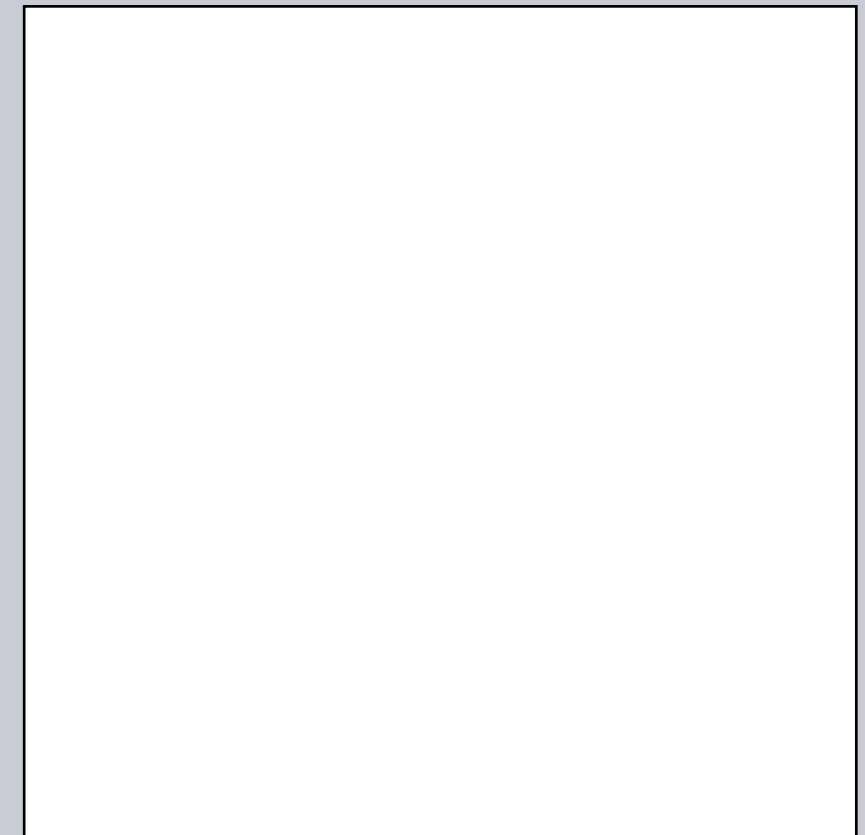
```
}
```

```
}
```

SIMPLE EXAMPLE

```
part {  
    name: "bg";  
    type: RECT;  
    mouse_events: 0;  
    description {  
        state: "default" 0.0;  
        color: 255 255 255 255;  
    }  
}
```

result:



SIMPLE EXAMPLE

```
part {  
    name: "blue_rect";  
    type: RECT;  
    mouse_events: 1;  
    description {  
        state: "default" 0.0;  
        color: 0 0 255 255;  
    }  
}
```

SIMPLE EXAMPLE

```
rel1 {  
    relative: 0.1 0.1;  
}
```

```
rel2 {  
    relative: 0.3 0.3;  
}
```

```
}
```

```
}
```

result:



SIMPLE EXAMPLE

```
part {  
    name: "red_rect";  
    type: RECT;  
    mouse_events: 0;  
    description {  
        state: "default" 0.0;  
        color: 255 0 0 255;  
    }  
}
```

SIMPLE EXAMPLE

```
rel1 {  
    to: "blue_rect";  
    relative: 0 2;  
}  
rel2 {  
    to: "blue_rect";  
    relative: 1 3;  
}  
}
```

SIMPLE EXAMPLE

```
description {  
    state: "down" 0.0;  
    inherit: "default" 0.0;  
    rel1.relative: 3 2;  
    rel2.relative: 4 3;  
    color: 255 0 0 0;  
}
```

result:



SIMPLE EXAMPLE

```
program {  
    name: "blue_rect.clicked";  
    signal: "mouse,clicked,1";  
    source: "blue_rect";  
    action: STATE_SET "down" 0.0;  
    transition: ACCELERATE 0.5;  
    target: "red_rect";  
    after: "red_rect.restore";  
}
```


SIMPLE EXAMPLE

```
program {  
    name: "red_rect.restore";  
    action: STATE_SET "default" 0.0;  
    target: "red_rect";  
}
```

- a bit of logic.
- languages:
 - embryo
 - lua
- complexes behaviors.
- messages handling.



EDJE MESSAGES

- communication with code.
- alternative to signals.
- many parameters.
- more customized themes.
- more initial work.

- Elementary widgets.
- Emotion.

Video Player

- no C code.
- Elementary externals.
- Emotion externals.
- few lines of Embryo script.

CUSTOMIZING ELEMENTARY



group: “elm/radio/base/default”

images block:

- image: “radio.png” COMP;
- + image: “radio_red.png” COMP;

“radio” part:

- image: “radio.png”;
- + image: “radio_red.png”;

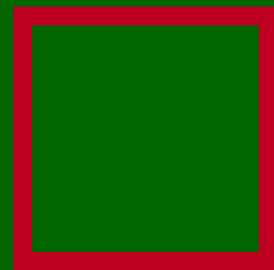
“elm.text” part:

- color: 0 0 0 255;
- + color: 255 0 0 255;

CUSTOMIZING WEBKIT EFL



Paris



London

group: “webkit/widget/checkbox”

- **add clipper part:**

```
+ part {  
+     name: "red_clipper";  
+     type: RECT;  
+     mouse_events: 0;  
+     description {  
+         state: "default" 0.0;  
+         color: 255 80 80 255;  
+     }  
+ }
```


- “check_button” part:

+ `clip_to: "red_clipper";`



TIPS

- use messages for most customization.
- well defined, high level signals.
- use macros.
- debug:
 - substitute complex parts by rects
 - emit signals

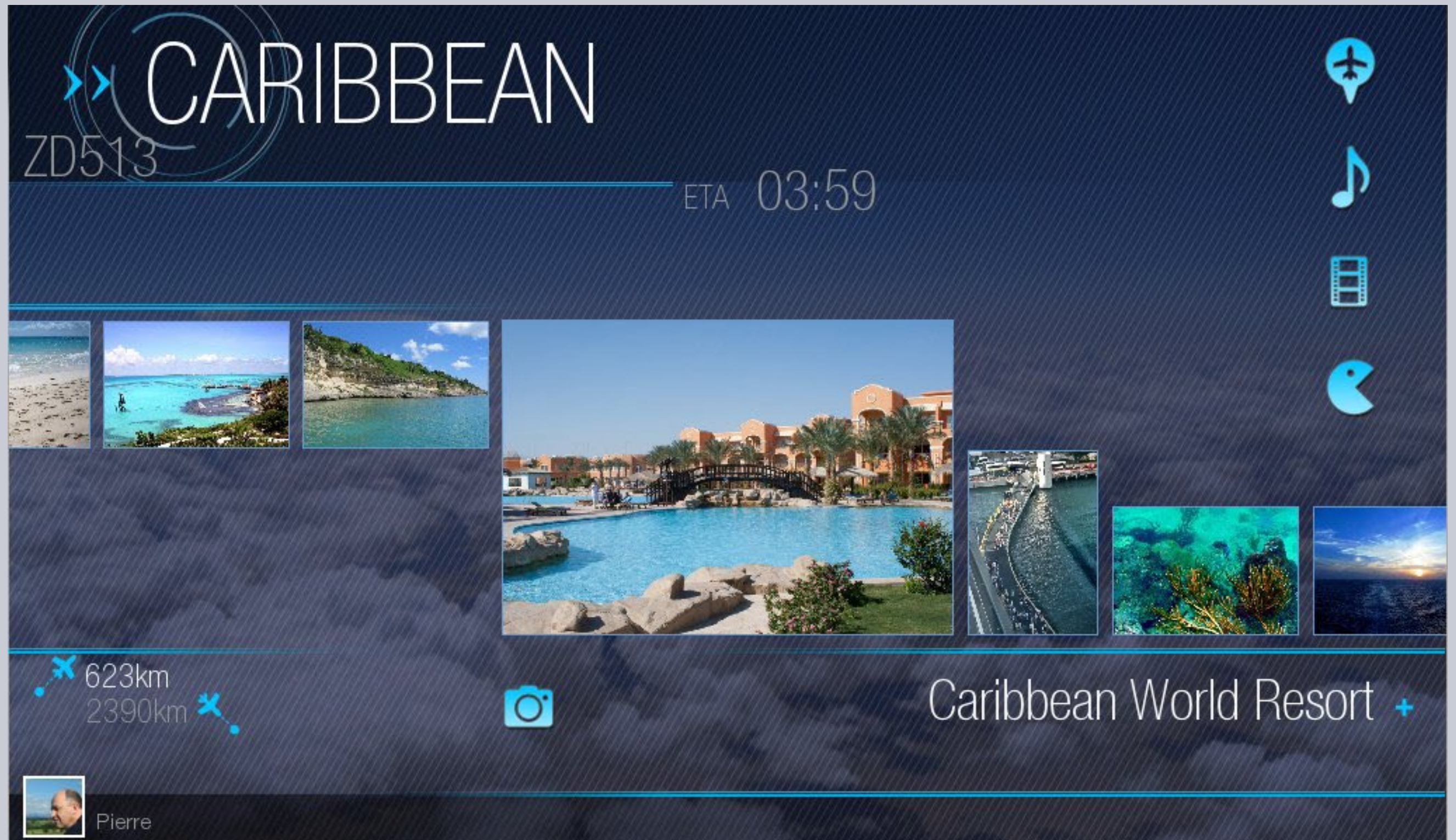


EDJE'S FUTURE

- physics.
- multisense improvements.
- code / theme interface improvements.

DEMOS ON PROFUSION BOOTH

Zodiac In Flight Entertainment System



DEMOS ON PROFUSION BOOTH

Electrolux Fridge



DEMOS ON PROFUSION BOOTH

ProFUSION Endeavour Browser



Applications using Elementary and Webkit can have layout completely customized with Edje.

- edje is easy to use yet powerful.
- edje is light thanks to Evas optimizations.

questions?



CONTACTS

bdilly at freenode

bdilly@profusion.mobi

brunodilly.org/blog

#edevelop at freenode

enlightenment-devel@lists.sourceforge.net

enlightenment.org/

Tizen sdk:
developer.tizen.org/sdk

Edje docs:
docs.enlightenment.org/auto/edje/

Examples code:
profusion.mobi/~bdilly/edje_examples/

thank you!

Bruno Dilly

profusion.mobi