Introduction to Eiffel Vision 2

Christian Estler

Distributed Software Engineering Lab 2015
What it’s about...

- Eiffel Vision 2 is a UI library
- Allows for building platform independent graphical UIs
  - Windows
  - Linux
  - Mac OS
Create a new project...

EiffelStudio dialog showing the creation of a new project with options for Basic application, .NET application, and Graphics application, multi-platform, with EiffelVision 2.
A first EV2 Application (2)

Step through the wizard...
A first EV2 Application (3)

Compile and run...

Note: EiffelStudio uses Vision2 for its GUI
Structure of an EV2 UI

• Start with a main window, i.e. inherit from
  • class EV_WINDOW or
  • class EV_TITLED_WINDOW or
  • .... (some other WINDOW class you want)

• Extend window with
  • Containers
    • To define layouts, e.g. Vertical, Horizontal; or other Windows...
  • Primitives
    • E.g. Buttons, Labels, Text Fields, ...
  • Items
    • E.g. Menu bar, Tool bar, Tool bar button, ...
How to get an overview of EV?

- Your Eiffel installation comes with the **EiffelBuild** application

- EiffelBuild is a simple GUI Builder
  - Lets you pick & drop the tree structure of an UI
  - Generates code

- You can use it to get familiar with EV2
EiffelBuild: Containers, Primitives & Items

Containers

Primitives

Items
Design tree structure

Set properties of elements
EiffelBuild

Preview your UI

![TicTacToe interface](image-url)
EV2 for Games

• How to use EV2 for (board) games?

• Two main approaches
  • Use containers with background images and EV_PIXMAP
  • Use EV_MODEL to draw on a EV_DRAWING_AREA
EV2 Game based on containers (1)

- Use lots of containers with
  - Subcontainers with
  - Subsubcontainers with
  - Subsubsubcontainers with
  - ...

- Containers typically have
  - Fixed or relative positions
  - background images (*.bmp, *.png)

Tip: only have on PIXMAP per container. Add additional images “on top” via pixmap.draw_pixmap
How to build a game with containers?

- Main idea
  - Main container has background image
  - Other containers have pictures representing the game state
  - Images change on Mouse/Keyboard events

- This works fine if you don’t need a lot transparency or dragging of elements over the board game

- If you need to move elements or want to “draw” instead of using images, use MODEL
EV2 Game based on Model

- Eiffel Vision has MODEL classes

- Can be used to draw elements on a drawing area
  - Line, ellipse, pie-slice, polygon, rectangle, text, star...

- Model classes work better with transparent pictures
How to get started with EV Model

1. Start from a usual EV application

2. Add an area: EV_DRAWING_AREA to a container

3. Create objects
   - projector: EV_MODEL_DRAWING_AREA_PROJECTOR
   - world: EV_MODEL_WORLD

4. Add the world and the area to the projector

Demo: https://github.com/DOSE-ETH/eiffel_vision_demo
Your turn...

We recommend you

• Build a first EV2 application (EiffelStudio default one) and take a look at the code

• Play around with EiffelBuild quickly learn about available widgets

• Take a look at the provided example