li atio e elo me t it E je

Title:	Applicati evel pme t with E e
Su title:	Fr m the very basics
Auth r:	A res la c
C nt t:	a resbla c mail. m
Versi n	

Content

of each one to serve as an introduction for the practical examples to come in the next chapters.

- Building a framework. Hopefully the reader's mind won't be filled already with preconceptions about this subject. In any case, only the foundations of what could be used to form a complex framework are going to be reviewed in this chapter.

    * Simpler library initialization. We have seen why and how to ini-

this chapters we will a aly e the surface c e f a Smart b ect base wi et, the mi imap.

*

pter

ut Gr p i l U er nterf e

S … y u wa t t create a GU applicati ? assume s si ce y u ch se this b k as i structive, r at least bathr m, material. u c ul le “GU ” a “library” t feel verwhelme by the lar e umber f evel pme t libraries available. As y u l k thr u h the s urce f y ur pe s urce) fav rites y u will reali e that all f them, a the applicati s that use them, share a c mm structure. this chapter we will review that structure.

At this p i t it is c ve ie t t te that the c cepts see i this b k a the applicati s

[redacted]

The fu cti s that f rm a eve t rive applicati with a GU) ca be split am tw r ups. The first r up is k w as the backe , these fu cti s eal with the actual purp se f the applicati , cru chi umbers, ec i me ia files a s . The sec r up is k w as the fr te , the prup se f these fu cti s is t prese t the results fr m the backe t their huma verl r s a t receive r ers fr m them.

etwee the backe a the fr te is where the Eve t l p lives, its missi is t c ect b th e s f the applicati . t ly betwee them but als with their e vir me t. The Eve t l p ma tai s a list f si als t l k ut f r a fu cti s relate t them. Whe a si al is receive , the Eve t l p l k it up i a list a executes the c rresp i fu cti r fu cti s.

F r the applicati t w rk, the Eve t l p ee s t be aware f eve ts i the i terface, thus it is usually pr vi e by the same library that pr vi es the GU eleme ts. t als ee s t be aware f eve ts i the system where the applicati is ru i . Eve whe the c cept is simple, i6 ,the 7e w tabl[the)-2 7 a7e 333 p33 y[the)-bstr- 1e 7 predefined interface elements, the latter known as "toolkit" or widget library. As opposed to the first, in a toolkit the canvas is just another widget.

Regardless of the method of choice, the resulting GUI has to provide the same resources to the rest of the application. A mechanism to present information to the user, a mechanism to know when the user interacts with the interface and a mechanism to retreive information that resulted from said interaction.

In the case of the plain canvas the application developer must assemble the interface elements, known as widgets, using primitive objects. A very simple text entry widget an action that may affect other parts. In this way a part collection can be "programmed" via its file as to highlight buttons when the mouse passes over them or show hidden parts when a button is clicked somewhere etc. The actions performed in changing from one state to another are also allowed to transition over a period of time, allowing animation.

[..] This separation and simplistic event driven style of programming can produce almost any look and feel one could want for basic visual elements. Anything more complex is likely the domain of an application or widget set that may use Edje as a convenient way of being able to configure parts of the display.

Except r t e usa e este bl cks, t e si tax a ED le is similar t SS. W at really sets t em appart is t at wit ED t e esi er it's ree t 2reate a lay ut esi eleme ts as e sees t. Wit SS t e esi er is limite t applyi style a a excelle t i tr ucti t Evas as alrea y bee writte i t e AP e ere ce.

> Evas is a clea isplay ca vas AP r several tar et isplay systems t at ca raw a ti-aliase text, sm t super a sub-sample scale ima es, alp a-ble b ects muc a m re.
>
> t abstracts a y ee t k w muc ab ut w at t e c aracteristics y ur isplay system are r w at rap ics calls are use t raw t em a w. t eals a b ect level w ere all y u is create a ma ipulate b ects i a ca vas, set t eir pr perties, a t e rest is e r y u.
>
> Evas ptimises t e re eri pipeli e t mi imise e rt i re rawi c a es ma e t t e ca vas a s takes t is w rk ut t e pr rammers a , savi a l t time a e er y.
>
> t's small a lea , esi e t w rk embe e systems all t e way t lar e a p wer ul multi-cpu w rkstati s. t ca be c mpile t ly ave t e eatures y u ee r y ur tar et plat rm i y u s wis , t us keepi it small

. Conv ni nt librari s

T e rmal pr cess t et a ca vas up a ru i ca be b t ers me. Evas supp rts multiple re eri e i es, li e t e s tware, xre er a pe l av rs 11 a ramebu er evices. ut be re a y re eri ca be e t e evel per as t c mplete a Evas_E i e_ structure wit t e require i rmati ab ut t e tar et a)2 . T is rmati ces t e evel per t researc t e i ere t u cti s t et t at i rmati r r eac tar et. Alter atively e ca use a s rtcut available r m st t em.

As y u mi t ave reali e by at t is p i t, i te t qu te t e fficial AP re ere ce at every c a ce et. T is e c mes strai t r m t e “T e Ec re Mai p” pa e:

> Ec re is a clea a ti y eve t l p library wit ma y m ules t l ts c ve ie t t i s r a pr rammer, t save time a e rt.
>
> t’s smal -3 l r3.)-26 t esi e-3 l as)-2smas t)33 e 3 l 2 r2smaembe e-3 l systemT [t

pter

y efault, Ec re aware ess is limite t system si als like HUP r K. A iti al

t■at a ﬁramew rk es t ecessarily mea lar e s ﬁtware libraries. A ﬁramew rk ca be see as library ﬁ ﬁu cti s etermi e by t■e similiarity i t■e pr ﬁle ﬁ t■e applicati s t■at use it. a ﬁramew rk, s■ rter evel pme t time mea s eit■er m re speciﬁc pr ﬁles r m re c mplex library c e.

t■is secti a t■e ﬁ ll wi subsecti s we will evel p a applicati ﬁramew rk ﬁ r applicati s wit■ a speciﬁc pr ﬁle: “A 11 esk t p applicati t■at es ’t require a y ex tic ma ipulati ﬁ its t■eme r c ﬁ urati ﬁles”. A list ﬁ t■e tasks t■e ﬁramew rk must perﬁ rm ﬁ ll w:

- **C ﬁ u ti n (Us n E e_C ﬁ).**
  - itiali ati a s■ut w ﬁ t■e ecesary services.
  - Savi c ﬁ urati c■a es exit.
  - ecall t■e previ usly save values i itial ati .
  - tr l t■at t■e ecessary c ﬁ urati a t■eme ﬁles exist.
- **nteﬁ e n ement**
  - reate wi ws wit■ t■eir pr perties likte ﬁr m a ive E e b ect.
  -

```
appli ation_na e_ et("Plain Ed e Viewer")
if (! i pler_init())
    return EXIT_F IL RE

if(!argu ent _par e(path, group, arg , argv))
    return EXIT_F IL RE

 ainWindow =  i pler_window_new("window/ ain",  LL)
```

```
        n ap =  n ap_add( a n anva )
        n ap_the e_ et( n ap,  pler_ob e t_add( a n anva , "w dget. n ap"))
       ed e_ob e t_part_ wallow( a nLa out," wallow. n ap",  n ap)
        n ap_v ewport_ et( n ap,v ewport)

       e ore_ a n_loop_beg n()
   }
```

u mi t tice t e c e s ippet is split i t ree bi er secti s. t ese secti s we

```
            fprintf( tderr, "Error: Eva  failed to initiali e.\n")
            return F LSE
        }
        ...
```

The rest of the i itiali ati fu cti s ca be fou i the VS rep sit ry, eci e t

```
        Eva _ b e t *o
        o = ed e_ob e t_add( anva )
        if( i pler_ob e t_file_ et(o, e ore_ onfig_the e_with_path_get("the e/
        {
            return o
        }
        el e
        {
            if( i pler_ob e t_file_ et(o, e ore_ onfig_the e_with_path_get("the
            {
                return o
            }
        }
        return   LL
    }
```

```
        if(path !=   LL)
        {
            if(!e ore_file_exi t (path))
            {
                fprintf( tderr, "Warning: Failed to find the the e file '   '.\n", path
                return F LSE
            }
            el e
            {
                if(!ed e_file_group_exi t (path, group))
                {
                    fprintf( tderr, "Warning: Failed to find group '   ' in the e file
                    return F LSE
                }
                el e
                {
                    ed e_ob e t_file_ et(o, path, group)
                    ewa _ob e t_ how(o)
                    return TR E
                }
            }
        }
        el e
            return F LSE
    }
```

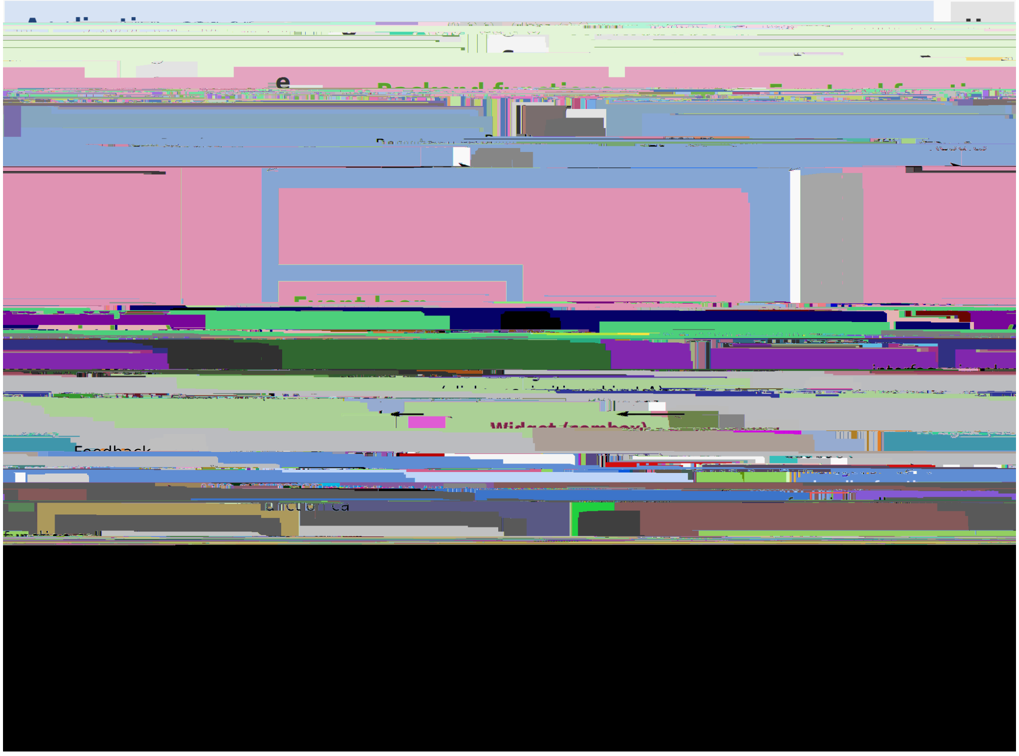
This fu cti is als quite simple. t basically calls *e je_ bject_ le_set* after ru i s me checks t see if the parameters are vali . Afterr_sete,2 te)-   _sesi the t weizasi_bbeantsethet)2 6 efaulte)2 7 vaue)2 6 i ese)2 6 the)2 6 _ b eci)2 6 ai)]TJ177.67 -1 .

pter 4

ntrodu tion to idget

Graphical User terfaces t ly t isplay i f rmati , they c vey i f rmati . terface eleme ts have a mea i f their w a this mea i alters the user's perception the i f rmati isplaye , f r better r f r w rse. A flexible i terface esi system mea s the esi er ca a m re mea i t the i f rmati . Features like multiple states a tra siti s exte this capacity t the p i t where the esi er's creativity is the limit.

As the applicati matures the umber f elme ts i the i terface will r w. These eleme ts will be r upe by s me c mm pr perty r prup se. Fu cti s t eal with these r ups as a u it are als i t be create . This is t a u ique pr cess a are als a y e crea ta t eal with is ta t is c mm lyywi et"t.-6 7 Wwi ets)- 76 w)2)2 ky)- 76aisare als appl



r er t be p ssible fi r tfie rest fi tfie applicati t i teract e ectively witfi a wi et it fias t pr vi e tfie Eve t l p witfi ew si al types represe ti abstracte versi fi its i terfiace si als. t basically i tercepts wfiat w ul be tfierwise rmal i terfiace c rresp with the am u t f w rk each e mi ht e tail, the ivisi s c c r with the