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.c m
Versi n:	1.
D te:	2 / / 3
C p r ht:	This b k is istribute u er the terms f the Attributi -ShareAlike lice se. See:

ntent

pter 1

er ie

. . uil i a framew rk. - A brief i tr ucti t the c cept f the framew rk - e sample f the mai .c file a e eral expla ati

. . . Getti the i terface up. - A me ti f the e vir me t_i it/shut w fu cti s with ut c e examples - A c e sample + expla ati f the the wi w creati fu cti s

. . . . Setti s are imp rta t t . - A brief i tr ucti t Ec re_ f refer t the c kb k). - A c e sample + expla ati f fu cti s that eal with multiple theme files.

. . . . Simpler i terface ma a eme t. - A c e sample + expla ati f the fu cti s that fetch the theme b ects

. tr ucti t wi ets. terfaces ee t res lve tw pr blems, prese ti i f rmati t users a taki r ers fr m them. The past chapters prese te e u h i f rmati t s lve the first. the f ll wi chapters we will review the sec .

. . Wi ets with E e. Si ce this is a b k ab ut E e it mi ht be a i ea t etail the way E e simplifies cust m wi et creati by reviewi the list f tasks prese te i the previ us chapter a h w E e helps with each task.

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

pre ef e i terface eleme ts, the latter k w as “t lkit” r wi et library. As pp se t the first, i a t lkit the ca vas is ust a ther wi et.

e ar less f the meth f ch ice, the resulti GU has t pr vi e the same res urces t the rest f the applicati . A mecha ism t prese t i f rmati t the user, a a acti tat may a ect ter parts. tis way a part c llecti ca be “pr ramme ” via its le as t ili t butt s we te m use passes ver tem r s w i e parts we a butt is clicke s mewere etc. Te acti s per rme i ca i r m e state t a ter are als all we t tra siti ver a peri time, all wi a imati .

[..] Tis separati a simplistic eve t rive style pr rammi ca pr uce alm st a y l k a eel e c ul wa t r basic visual eleme ts. A yti m re c mplex is likely te mai a applicati r wi et set tat may use E e as a c ve ie t way bei able t c ure parts te isplay.

Except for the usage of nested blocks, the syntax of a LESS file is similar to CSS. What a excelle t i tr ucti t Evas has alrea y bee writte i the AP efere ce.

> Evas is a clea isplay ca vas AP f r several tar et isplay systems that ca raw a ti-aliase text, sm th super a sub-sample scale ima es, alpha-ble b ects much a m re.
>
> t abstracts a y ee t k w much ab ut what the characteristics f y ur isplay system are r what raphics calls are use t raw them a h w. t eals a b ect level where all y u is create a ma ipulate b ects i a ca vas, set their pr perties, a the rest is e f r y u.
>
> Evas ptimises the re eri pipeli e t mi imise e rt i re rawi cha es ma e t the ca vas a s takes this w rk ut f the pr rammers ha , savi a l t f time a e er y.
>
> t's small a lea , esi e t w rk embe e systems all the way t lar e a p werful multi-cpu w rkstati s. t ca be c mpile t ly have the features y u ee f r y ur tar et platf rm if y u s wish, thus 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, like t■e s ftware, xre er a pe l flav rs fi 11 a framebu er evices. ut befi re a y re eri ca be e t■e evel per ■as t c mplete a Evas_E i e_ fi structure wit■ t■e require i fi rmati ab ut t■e tar et a)2 . T■is rmati ces t■e evel per t researc■ t■e i ere t fu cti s t et t■at i fi rmati r r eac■ tar et. Alter atively ■e ca use a s■ rtcut available fi r m st fi 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 refiere ce at every c■a ce et. T■is e c mes strai ■t fir 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 fi c ve ie t t■i s fi r a pr rammer, t save time a e rt.
>
> t’s smal -3 l r3.)-26 t■esi e-3 l as)-2smas fit)33 ek3 l 2 r2smaembe e-3 l systemT [t

pter

T e found tion in pr ti e

y p i ti ut t at t e E li te me t F u ati ibraries are esi e i a b ect rie te ma er wis t t raise t e wrat purists but t simpli y t e me tal ima e t e structure t e EF AP i t e rea er's ima i ati .

w t at t e pitc r s are bac i t e bar all w me t put it i m re clear terms wit a simple example:

```
Eva _ b e t button =  LL
button = ed e_ob e t_add(eva  anva )
ed e_ob e t_f le_ et(button, "the e.ed ", "button")
```

T is is a simple s ippet t at c ul be tra slate i t a m re si tactically spea i) b ect rie te la ua e li e Pyt as:

```
button = Eva _ b e t()
button.f le_ et("the e.ed ","button")
```

T e i ere ces betwee tu allys ippces c ul be ctee
itservntest me t p2T 32 p2TF)-33p2T bar s. u 33 m e ua e me t 21)- 33 bul)]TJlas

y default, Ec re aware ess is limite t system si als like HUP r K LL. A iti al libraries r m ules like Ec re_Evas re ister ew si al types f r the eve t l p t be aware f. the specific case f Evas the ew si al types eal with the i teracti betwee the user a the Evas b ects isplaye i the ca vas.

The evel per ca ma ipulate the list f ha lers as well as creati ew si al types. The latter am ther sub ects like timers a p llers excee the sc pe f this b k a are pr perly cume te by the AP refere ce a the EFL kb k.

We will be i by setti up a simple si al ha ler that will be calle a y time the applicati is cl se :

```
E ore_Event_Handler*  lo e =   LL
...
int
good_b e(void *data, int t pe, void *event)
{
    //Re oving handler for no rea on other than  PI  howoff
    if (e ore_event_handler_del( lo e))
        printf("Handler deleted\n")

    printf("Good b e! \n")
    e ore_ ain_loop_ uit()
    e ore_eva _ hutdown()
    e ore_ hutdown()
    ed e_ hutdown()
}
...
int  ain() {
...
     lo e = e ore_event_handler_add(E  RE_EVE T_SIG  L_EXIT,
                                    good_b e,"data")
    ...
    e ore_ ain_loop_begin()
    ...
```

This example m ves the library shut w pr ce ure fr m the mai fu cti t the

that a framew rk es t ecessarily mea bi c mplex, abstract s ftware libraries. A framew rk ca be see as the sta ar i ati f c mm , c mplex tasks i t a library share by r up f applicati s. With a framew rk sh rt evel pme t times mea s a

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 percepti 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

Widget (combox)
Feedback

equitative t the am u t f w rk each e mi ht e tail, the ivisi s are c c r a t with the sh rtcuts E e pr vi es. F r