

GtkD

Mengü Kağan ve Can Alpay Çiftçi

GtkD

GtkD ile D programlama dilinde
görsel programlama
Mengü Kağan ve Can Alpay Çiftçi

-2.1 Copyright © 2009-2015 Mengü Kağan ve Can Alpay Çiftçi



Ddili.org icerigi by [Ali Cehreli](#)² is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 United States License](#)³.

Based on a work at [digitalmars.com](#)⁴.

Permissions beyond the scope of this license may be available at <http://ddili.org/copyright.html>⁵.

-
1. <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>
 2. <http://ddili.org/>
 3. <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>
 4. <http://digitalmars.com/>
 5. <http://ddili.org/copyright.html>

İçindekiler

-1. Tanıtım	6
0. Merhaba GtkD	7
Peki Bu GtkD Kodlarını Nasıl Çalıştıracamız?	8
1. Pencere Kurma İşlemleri	9
Pencere Boyutunu Ayarlamak <code>resize(int en, int boy)</code>	9
Pencereyi Tam Ekran Yapmak <code>maximize()</code>	9
Pencereyi Tam Ekrandan Çıkartmak <code>unmaximize()</code>	9
Pencereyi Ekranda Konumlandırmak <code>move(int sağa, int aşağıya)</code> ..	10
Pencereyi Piksel Değeri Girmeden Konumlandırmak <code>setPosition(GtkWindowPosition nereye)</code>	10
Pencereye Simge Ekleme <code>setIconFromFile</code>	11
Pencereyi Saydamlaştırmak <code>setOpacity(double değer)</code>	11
Pencere Boyutlandırma <code>setSizeRequest(int en, int boy)</code>	11
Boyutlandırılmayan Pencere <code>setResizable(bool değer)</code>	11
X Düğmesine Kapama Görevi Verme	12
Dizin	13

-1 Tanıtım

Sıra GtkD'ye geldiğine göre D hakkında ciddi bir bilgiye sahipsiniz demektir. Eğer eksikleriniz varsa mutlaka [D derslerini](#) okuyun ve ardından bu derslere başlayın.

İsterseniz önce GTK'nın ne olduğundan bahsedelim. GTK, C ile yazılmış, GNU LGPL 2.1 ile lisanslanmış platformdan bağımsız bir grafik kullanıcı arayüzüdür. Örneğin GNOME masaüstü GTK ile yazılmıştır. GTK C ile yazılmış olmasına rağmen, C++ başta olmak üzere, C#, Java, Python ve hatta PHP bağlayıcıları (binding) bulunmaktadır. GtkD de GTK'nın D bağlayıcısıdır ve GTK gibi GNU LGPL 2.1 lisansına sahiptir.

GtkD ile çalışabilmeniz için aşağıdakilere ihtiyacımız var:

- DMD 2
- gtkD 1.3.2
- Bir programlama veya metin editörü (örneğin: geany, gedit)

GtkD'yi kurabilmek için ayrıntılı bilgiyi D Dili Forum'da bulabilirsiniz.

Tanıtım için zannediyorum bu kadarı yeter, bir sonraki derste gtkD ile merhaba dünya diyeceğiz. Hazırlanmayı ihmâl etmeyin.

0 Merhaba GtKD

Bu dersimizde gtkD ile pencere oluşturabilmek için gerekli modülleri öğreneceğiz.

En basit pencere, içi boş penceredir. İlk önce içi boş bir pencere nasıl oluşturulur onu öğreneceğiz. Sizce bir pencere oluşturmak için neler yapmak gerekir? Hızlıca düşünersek:

1. Gtk işlevlerini kullanmadan önce bir işlev çağırmanız gerekiyorsa o işlevi çağırmalıyız. (Sizce gerek var mıdır :o)
2. Pencere oluştur demeliyiz.
3. Pencereyi göster demeliyiz.
4. Bu yaptıklarımızı çalıştır demeliyiz.

Bunları sağlayan kod parçacığı aşağıda :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("deneme");
    pencere.show();
    Main.run();

    return 0;
}
```

Örneğimizde neler yaptık bakalım:

İlk olarak iki tane modülü programımıza dahil ettik. Bunlar `gtk.Window` ve `gtk.Main`. Sonra D'nin ana işlevi olan `main` işlevini tanımladık. Ama burada `main` işlevini tanımlarken `string[]` türünde bir argüman tanımladık. Burada argümanın hangi türden tanımlandığı önemlidir.

Şimdi sıra tanımlanan argümanı kullanmaya geldi. Dersimizin ilk bölümlerinde pencereleri oluşturmadan önce; daha doğrusu `gtk`'nin işlevlerini kullanmadan önce başka bir işlev çağırmanız gerektiğinden söz etmiştik. Bu işlevi her `gtk` uygulamasında çağırmanız gerekir. Şimdi bu işlev hangi modülde bulunur, adı nedir, parametre alır mı, alırsa hangi türden parametre alır gibi soruların yanıtlarını alacağız.

Tanıyacağımız ilk işlev, `Main.init`. Bu işlev `gtk.Main` modülünde bulunuyor. Parametre olarak da D uygulamasının ana işlevini tanımlarken kullandığımız argümanı kullanıyor. Ve parametre türü olarak da sadece `string[]` türü kabul ediyor. Kısaca:

```
void Main.init (string[] args);
```

`Gtk` işlevlerini kullanmak için `gtk.Main` modülünde bulunan `Main.init` işlevini kullandıktan sonra sıra penceremizi oluşturmaya geldi. Pencereyi oluşturabilmek için ilk önce `gtk.Window` modülündeki `Window` sınıfını kuruyoruz. Bunun için şu satırı yazmıştık :

```
auto pencere = new Window("deneme");
```

Koddan anlaşıldığı üzere bu sınıfın kurucusu tanımlı ve parametre türü olarak **string** değer alıyor. Kurucudaki parametre, oluşturacağımız pencerenin başlığını tanımlar.

Artık uygulamamızın sonuna doğru geldik. Bundan sonraki işlev yine `gtk.Window` modülünde tanımlı. Daha önce pencere adı ile kurduğumuz sınıfın **show** işlevine erişiyoruz. Show sözcüğünün Türkçe anlamı görüntüle, göster demektir. Biz de oluşturduğumuz pencereyi göstermek için **`pencere.show;`** kodunu yazıyoruz.

Ve sonunda bu gtk uygulamasını çalıştırmak için gtkD ile ilişkili olan son kod satırımızı yazıyoruz. Bu kod satırında gtk uygulamasını çalıştır anlamına gelen **`Main.run;`** kodunu yazıyoruz. Bu kod parçası yine `gtk.Main` modülünde bulunuyor.

0.1 Peki Bu GtkD Kodlarını Nasıl Çalıştıracamız?

Bu örneği merhaba.d adında çalışma alanınıza kaydettikten sonra şu şekilde derleyebilirsiniz:

```
dmd merhaba.d -I/gtkD/yolu/src -L-ldl -L-L/gtkD/yolu/src -L-lgtkd
```

Yukarıda dikkat etmeniz gereken kısım “/gtkD/yolu” kısmıdır ki onu gtkD'yi nereye kurduysanız, o yol (path) şeklinde değiştirmeniz gerekmektedir. Örneğin ben ~/projects/gtkD klasörü altında kurduğum için şu şekilde derledim:

```
dmd merhaba.d -I~/projects/gtkD/src -L-ldl -L-L~/projects/gtkD/src -L-lgtkd
```

merhaba.d dosyasını derledikten sonra da terminalde ./merhaba yazarak pencerenizi görebilirsiniz.

1 Pencere Kurma İşlemleri

Daha önceki dersimizde başlığı olan boş bir pencere yapmasını öğrenmiştik. Bu dersimizde pencere kurma ile ilgili işlemler yapacağız.

Anlatılacak bütün işlevler `gtk.Window` modülünde tanımlı olan *Window* sınıfının üye işlevleridir.

1.1 Pencere Boyutunu Ayarlamak `resize(int en, int boy)`

`gtkD` ile oluşturulan pencerenin öntanımlı boyutu 200x200 piksel boyutundadır. Eğer öntanımlı piksel boyutunu değiştirmek isterseniz, `resize` işlevini kullanabilirsiniz. `resize` iki tane `int` türünde parametre alır. Bunlar sırasıyla `en` ve `boydur`.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere =
        new Window("450 eninde ve 570 boyunda resize.");
    pencere.resize(450,570) ; // Pencere 450 piksel eninde
                             // ve 570 boyundadır.
    pencere.show();
    Main.run();
    return 0;
}
```

1.2 Pencereyi Tam Ekran Yapmak `maximize()`

Eğer pencerenin tam ekran olmasını istiyorsanız `maximize()` işlevini kullanabilirsiniz.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("maximize");
    pencere.maximize(); // Pencereyi tam ekran yapar.
    pencere.show();
    Main.run();

    return 0;
}
```

1.3 Pencereyi Tam Ekrandan Çıkartmak `unmaximize()`

Eğer pencerenin tam ekran kipinden çıkmasını istiyorsanız `unmaximize()` işlevini kullanabilirsiniz.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("unmaximize");
    pencere.maximize(); // Pencereyi tam ekran yapar.
    pencere.unmaximize(); /*Pencereyi tam ekran kipinden çıkarır
```

```
                                ve öntanımlı boyutuna getirir. */
pencere.show();
Main.run;
return 0;
}
```

1.4 Pencereyi Ekranda Konumlandırmak `move(int sağa, int aşağıya)`

Eğer pencerenin ekranda konumunu ayarlamak isterseniz, `move` işlevini kullanabilirsiniz. `move` işlevi 2 parametre alır, parametre türü olarak `int` değeri alır ve birimi pikseldir. Ekranın nereye yerleşeceğini de sol üst köşeden başlayarak girilen değer kadar sağa(en), ve aşağıya(boy) taşır.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("move");
    pencere.move(450,570) ; // 450 piksel sağa ve 570 piksel
                          // aşağıya taşır.
    pencere.show();
    Main.run;
    return 0;
}
```

1.5 Pencereyi Piksel Değeri Girmeden Konumlandırmak `setPosition(GtkWindowPosition nereye)`

Eğer pencerenin konumunu piksel olarak değer vermeden örneğin ortala diyerek belirtmek istersek `setPosition()` işlevini kullanabiliriz.

`setPosition()` işlevi parametre olarak şu değerleri alır:

`GtkWindowPosition.POS_NONE`: Pencere konumu üzerinde herhangi bir değişiklik yapılmayacağını belirtir.

`GtkWindowPosition.POS_CENTER`: Pencereyi ekranın ortasına yerleştirir.

`GtkWindowPosition.POS_MOUSE`: Pencere, farenin o anda bulunduğu konumda açılır.

`GtkWindowPosition.POS_CENTER_ALWAYS`: Pencerenin boyutu değişse de, pencereyi hep ekranın ortasında tutar.

`GtkWindowPosition.POS_CENTER_ON_PARENT`: Eğer ikinci bir pencere varsa bu pencerenin ana pencereyi ortalamasını sağlar.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("maximize");
    pencere.setPosition(GtkWindowPosition.POS_CENTER_ALWAYS);
    // ekranın ortasında açılmasını sağlar
    pencere.show();
    Main.run;
    return 0;
}
```

1.6 Pencereye Simge Ekleme setIconFromFile

Eğer pencereye simge(icon) eklemek isterseniz `setIconFromFile` işlevini kullanabilirsiniz. `int` türünde değer döndürür ve `string` türünde parametre alır. Parametre olarak ekleyeceğiniz resmin yolunu yazarsınız.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("deneme");
    pencere.setIconFromFile("/home/canalpay/den.png");
    // Pencereye simge koymaya yarar.
    pencere.show();
    Main.run();
    return 0;
}
```

1.7 Pencereyi Saydamlaştırmak setOpacity(double değer)

Pencereyi saydamlaştırmak için `setOpacity` işlevi kullanılır. 0 ile 1 arasındaki `double` değerler alır. Değer 1'e yaklaştıkça saydamlığı azalır.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("Saydam Pencere Yapımı");
    pencere.setOpacity(0.9); //Pencerenin saydamlığı oldukça azdır.
    pencere.show();
    Main.run();
    return 0;
}
```

1.8 Pencere Boyutlandırma setSizeRequest(int en, int boy)

Kullanımı `resize()` ile aynıdır. Tek farkı fare imleci ile pencerenin boyutu büyütülebilir ama öntanımlı boyuttan daha küçük yapılamaz.

Ayrıca bir sonraki işlev için `resize` yerine bunun kullanılması gerekecektir.

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("setSizeRequest örneği");
    pencere.setSizeRequest(450, 450);
    pencere.show();
    Main.run();
    return 0;
}
```

1.9 Boyutlandırılmayan Pencere setResizable(bool değer)

Pencere boyutlandırılabilir mi yoksa boyutlandırılmamasın mı diye ayarlamak için `setResizable()` işlevi kullanılır. "Resizable" sözcük anlamı "tekrar boyutlandırılabilir"dir. Parametre olarak `bool` değer alır. Eğer parametre olarak

false değeri girilirse boyutlandırılmaz. Eğer true değeri girilirse boyutlandırılabilir.

UYARI: setResizable işlevini doğru kullanabilmek için boyutlandırmayı **setSizeRequest** işlevi ile yapmak zorundasınız.

Örnek :

```
import gtk.Window;
import gtk.Main;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window("Boyutlandırılmayan Pencere");
    pencere.setSizeRequest(450, 450);
    pencere.setResizable(false);
    //false değeri girerek boyutlandırılmayacağını belirtmiş olduk
    pencere.show();
    Main.run();
    return 0;
}
```

1.10 X Düğmesine Kapama Görevi Verme

Eğer daha önceki örnekleri konsol üzerinde çalıştırdıysanız X düğmesine bastığınızda hala programın çalışmaya devam ettiğini görmüşsünüzdür. Programdan çıkmak için de *Ctrl + C* tuşuna basmış olmalısınız. Ama biz X düğmesine bastığımızda programın çıkmasını istiyoruz.

X düğmesine bastığımızda çıkmamız için şu kod parçacığını eklememiz gerekiyor.

```
pencere.addOnHide( delegate void(Widget aux){ exit(0); });
```

Bu kod parçacığının çalışması için de `std.c.process` ve `gtk.Widget` modüllerini eklememiz gerekiyor.

Örnek :

```
import gtk.Window;
import gtk.Main;
import gtk.Widget;
import std.c.process;

int main(string[] args)
{
    Main.init(args);
    auto pencere = new Window
    ("[X] Düğmesine Bas ve Program Sonlandırılсын!");
    pencere.setSizeRequest(450, 450);
    pencere.setResizable(false);
    pencere.addOnHide( delegate void(Widget aux){ exit(0); });

    pencere.show();
    Main.run();
    return 0;
}
```

Dizin

A

addOnHide 0

D

derleme 8

dmd 8

G

gtk.Main 7

gtk.Window 7

i

init 7

M

maximize 9

move 10

R

resize 9

run 8

S

setIconFromFile 11

setOpacity 11

setPosition 10

setResizable 11

setSizeRequest 11

show 8

U

unmaximize 9

X

X düğmesi 12