



AutoCAD Shape/Font Tanımlama

M. Şahin GÜVERCİN

www.cizimokulu.com

Ekim 2012

Shape/Font Tanım Dosyaları (Shape/Font Description Files)

Kendi sembollerinizi ve fontlarınızı oluşturup derledikten (Compile ettikten) sonra çizimlerinizde kullanabilirsiniz. Shape'ler kullanımı Block'lara çok benzeyen objelerdir. Derlenmiş Shape tanımları dosyasından (*.shx) LOAD komutuyla yükleyeceğiniz objeleri SHAPE komutuyla çiziminizin istenen yerlerinde belirleyeceğiniz ölçek ve açılarda yerleştirebilir, mevcut Shape objelerinizin konum, ölçek ve dönüş açılarını değiştirebilirsiniz.

AutoCAD SHP Fontları özel tipte Shape Tanım dosyalarıdır. Karakterler (harfler), Shape'lerle aynı yöntemle tanımlanırlar.

Block'lar Shape'lere göre kullanımı ve uygulaması çok daha kolay olan, esnek objelerdir. Bununla birlikte Shape'ler AutoCAD tarafından çizim içinde kullanılması ve saklanması çok daha verimli objelerdir. Basit (sade) çizim parçaları çok sayıda kullanılacak ve hız daha önemliyse kullanıcı tanımlı Shape'ler çok daha verimlidir. Block tanımları drawing dosyası içinde yer alırken, shape tanımları, bağımsız bir SHX dosyasında yer aldıklarından, drawing dosyaları herhangi bir yolla transfer edilirken Derlenmiş Shape Tanım dosyaları da (*.shx) drawing dosyası ile birlikte transfer edilmek zorundadır.

AutoCAD Shape ve Font dosyaları (*.shx), Shape Tanım Dosyalarının (*.shp) AutoCAD içinden derlenmesi (compile edilmesi) ile elde edilirler. Shape Tanım (Shape Description) dosyaları herhangi bir Text Editor (Notpad gibi) ile ASCII formatında oluşturulup değiştirilebilir.

Tanım dosyalarında her Shape veya (Font içindeki herhangi bir) Karakter tamamen aynı şekilde tanımlanır ve yazılırlar. Bu bakımdan Font ve Shape dosyalarının birbirinden hiç bir farkı yoktur. Dosya (*.shp) font olarak kullanılmak üzere yazılıyorsa, en başta Font ile ilgili (Font Adı, Türü, Karakter Boyutu, Karakter Kod Tablosu,... gibi) tanımlarla başlar. Başlangıçta girilen bilgi doğrudan doğruya Shape Tanımı ise bu dosyamız artık Font Dosyası değil Shape Dosyası olur.

Öncelikle bir konuyu netleştirmek zorundayız. Shape ve Font dosyaları, gerek oluşturulmaları (İlk tanımın farklı olması dışında) ve tanımlamaların şekli bakımından, gerekse Compile işleminden sonra elde edilen dosya türü (*.shx) bakımından biri birinin aynıdır. İlk bakışta aynı görünen bu dosyaların AutoCAD ortamında kullanımları (Çağırılmaları) tamamen farklıdır. Font dosyaları bir Text Style ile ilişkilendirip, Text, DText, MText,... veya Attribute tanımlama komutları ile doğrudan kullanılabilir. Shape tanım dosyaları ise önce yüklenip (Load edilip) sonra da SHAPE komutu ile kullanılabilir. Hem Font dosyası hemde Shape dosyası, bir Çizgi Tipi (Line Type) tanımı içinde doğrudan referans verilerek kullanılabilir. Bu konu başlığı altında Font dosyalarının ayrıntılarına girmeden Shape Tanım dosyaları ele alınacaktır.

Unutmayınız ki, kendi Shape tanımlarını oluşturabilmek çok güçlü ve değerli bir beceridir. Böyle bir beceriyi edinmek son derece karmaşık, dikkat ve sabır gerektiren bir iştir. (Bunu ben söylemiyorum, AutoDESK'in ifadesi aynen böyle)

AuoCAD Express Tools menüsünden MKSHAPE fonksiyonu kullanılarak, ekrandan seçilen objelerden Shape dosyaları (*.shx) oluşturulabilir. Bu yöntemle oluşturulan dosyalar doğrudan SHX formatında olduklarından, kullanıcı için fonksiyonun sınırları içinde pek çok detayın karanlık kalacağı açıktır. Burada amaç, ilgi duyan ve ihtiyacı olan arkadaşlarımız için, kod yazarak Shape tanımlamanın tüm detaylarını açıklamaktır.

Shape tanım dosyasında her satır istenilen uzunlukta yazılabilir. Uzun satırlarda 128 karakterden sonrakiler Compile edilmezler. Compile işlemi sırasında boş satırlar ve noktalı virgülün sağındaki karakterler işleme alınmaz. Demek oluyor ki, noktalı virgül kullanarak Shape Tanım dosyasına açıklamalar yazılabilir.

Her bir Shape Tanımı bir başlık satırı ile bunu takip eden (en az) bir veya fazla satırdan oluşan tanımlamalardan oluşur. Tanımlamalar (Specification Bytes) birbirinden virgül ile ayrılmış nümerik (tamsayı) değerlerden oluşur ve 0 (sıfır) ile sonlandırılır.

***shapenumber,defbytes,shapenumber
specbyte1,specbyte2,specbyte3,...,0**

shapenumber : 1 ile 258 arasında değişen (Unicode Font'larda 32768'e kadar olabilen), başında yıldız (asteriks) işareti olan Tamsayı. Unicode olmayan font dosyalarında 256, 257 ve 258 Derece işareti, artı/eksi işareti ve Çap işareti gibi sembolik tanımlamalar için kullanılır. Unicode fontlarda ise bu bölüm U+00B0, U+00B1 şeklinde Onaltı tabanlı (Hexadecimal) sayılar şeklinde kullanılır. Font dosyalarında shapenumber ilgili karakterin ASCII kod numarasıdır. Shape dosyalarında ilgili Shape e herhangi bir numara verilebilir.

Defbytes : Shape'in tanımı için kullanılan Tanım byte'larının (specbytes) sayısı. Tanımı sonlandırmak için kullanılacak 0 (sıfır) da sayılacaktır. Her bir Shape Tanımı için en çok 2000 byte kullanılabilir.

Shapenumber : Tanımlanan Shape'e verilen ad. Shape Adı büyük harflerle yazılmalıdır. Küçük harf içeren Shape Adları gözardı edilirler ve genelde Font dosyalarında karakterin etiketi anlamında kullanılırlar.

specbyte : Shape Tanım byte'ı. Herbir tanım byte'ı, vektör uzunluğu, yönü ve doğrultusunu veya özel anlamı ifade eden bir sayısal (tamsayı) kod olarak yazılır. Tanım byte'ları Ondalık sayı veya Onaltı Tabanlı sayı olarak yazılabilir. Söz konusu kod sayısının ilk karakteri 0 (sıfır) ise, bundan sonra gelen 2 basamak Onaltı Tabanlı (Hexadecimal) sayı olarak okunacaktır.

Vektör uzunlukları ve Yön kodları

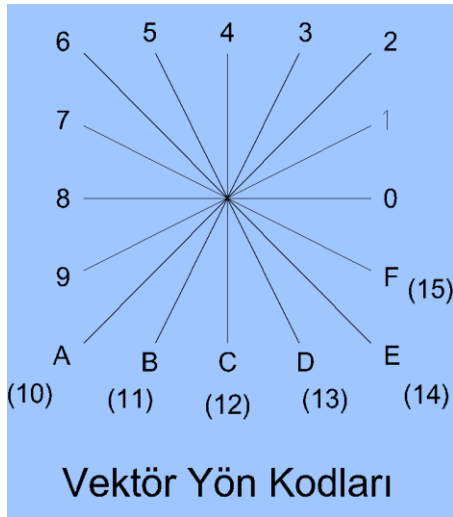
Shape Tanım Byte'larının en basiti olan vektör tanımı, Vektör Uzunluğunu birim cinsinden tanımlayan kod ve, vektörün Yön kodundan oluşur. Her bir Vektör Uzunluğu ve Yön kodu gurubu 3 karakterli bir diziden oluşur.

İlk karakter : 0 (sıfır) olmalıdır ki, AutoCAD takip eden 2 karakteri 16 Tabanlı (Hexadecimal) değer olarak algılasın.

İkinci karakter : Birim cinsinden Vektör uzunluğunun ifade eder. Vektör uzunluğu 1 ile F (15) aralığında geçerli 16 tabanlı sayı olmalıdır. Demek ki temel olarak en fazla 15 birim uzunluğunda vektör oluşturabiliyoruz. Bu sınırlama kafanızı çok meşgul etmesin, ileride özel kodlar bölümünde farklı uzunluklarda vektörlerin nasıl oluşturulacağını açıklayacağız.

Üçüncü karakter : vektörün yönünü belirler.

Aşağıdaki şekil Vektör Yön kodlarını göstermektedir.



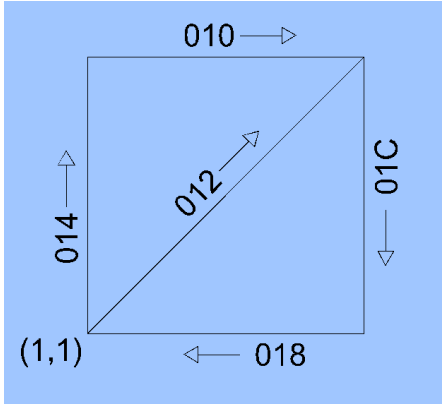
Şekilde görülen tüm yönlerdeki vektörler aynı uzunluk tanımıyla çizilmişlerdir. Diyagonal vektörler (X ve Y eksenindeki izdüşümleri bakımından) en yakındaki Ortogonal vektörün ilgili X ve Y değerlerine çakışacak şekilde uzatılırlar. Bu uzatma olayı, AutoCAD'deki "snap to grid" olgusuna benzer bir mantıkla işler. Buradaki 16 yönde vektör oluşturulabilmesi kafanıza takılmasın, ileride özel kodlar bölümünde istenen her doğrultuda vektörler oluşturmayı hatta Yay ve Eğriler çizmeyi açıklayacağız.

Aşağıda örnek olarak (rastgele seçilmiş) 230 numaralı, DBOX isimli örnek shape tanımı verilmiştir.

***230,6,DBOX**
014,010,01C,018,012,0

Örnekteki tanım baytları verilen sıra ile uygulandığında 1 birim eninde, 1 birim yüksekliğinde, bir kutu ve bunun diyagonalinden oluşan şekil ortaya çıkar. Kodun işlemlerini adım adım açıklarsak;

***230,6,DBOX** -> Shape numarası 230, tanımda 6 byte kullanıyoruz, shape adı DBOX
014 -> 4 yönünde 1 birim
010 -> 0 yönünde 1 birim
01C -> C yönünde 1 birim
018 -> 8 yönünde 1 birim
012 -> 2 yönünde 1 birim
0 -> shape tanımı sonu



Burada özellikle uyarmak istiyorum, tabanı ve yüksekliği 1 birim olan üçgenin hipotenüsü neden kök 2 değil de 1 oluyor, diye zihninizde bir direnç varsa konunun devamına geçmeyin, yukarıda açıklananları tekrar okuyun. Programlama dillerine yeni bulaştığım yıllarda, $A=A+B$ ifadesini bir türlü kabul edememiş, Nasıl olur? böyle eşitlik olmaz ki! diye zihnimdeki direnci kıramamıştım. Sonunda bu ifadede yazılan eşit işaretinin (=), eşitlik anlamında değil, atama (assignment) anlamında kullanıldığını

keşfedene kadar huzur bulamamıştım. Burada da, kareyi çizdikten sonra, diyagonalinin nasıl 1 birim olduğuna kafa yormayın. Bunun yerine karenin bir köşesinden 1 birim uzunluğunda diyagonal bir vektör tanımladığımda bu vektör çaprazdaki köşeye kadar gidiyor şeklinde düşünmekte fayda var.

Tanım byte'larında yer alan özel kodlar ve anlamları

000 : Shape tanımının sonunu ifade eder.

001 : Çizim durumunu etkinleştirme (Kalemi indirme). Her Shape tanımının başında kod 1 girilerek çizim durumu etkinleştirilip belirlenen vektör(ler) çizilmiş olur.

002 : Etkin çizim durumundan çıkma (Kalemi kaldırma) Kod 2 girilerek çizim durumu etkin durumdan çıkarılınca, belirlenen vektör çizilmeden yeni konuma gidilmiş olur.

003 : Vektör uzunluklarını bir sonraki byte'a bölme.

004 : Vektör uzunluklarını bir sonraki byte ile çarpma.

Kod 003 ve Kod 004'ü 1 byte uzunluğunda tamsayı ölçek faktörü (1'den 255'e kadar) takip eder. Ölçek faktörü (çarpım ya da bölüm) işlemi kümülatif (birikimli) dir. Yani 003,2 ve peşinden 003,6 girilirse, takip eden vektörler $2 \times 6 = 12$ 'de 1'i uzunlukta olacaklardır. Shape tanımlarında ve Text Fontlarında, özellikle alt shape kullanıldığında ölçek faktörü shape tanımının sonunda eski değerine döndürülmelidir. AutoCAD sizin adınıza ölçek faktörünü geri döndürmez.

005 : O anda geçerli pozisyonu saklama.

006 : Saklanan pozisyona geri dönme.

Kod 005 ile yalnızca 4 pozisyon saklanabilir. Yeni bir pozisyon saklanmadan önce saklanmış pozisyonlardan ihtiyaç duyulanlar Kod 006 ile alınmış olmalıdır. Aksi halde pozisyon yığını doluyken kod 005 kullanılarak yeni pozisyon kaydedildiğinde, compile işlemi sırasında bir hata ile karşılaşılmayacak fakat, tanımlanan Shape kullanılmaya çalışıldığında, nnn numaralı Shape tanımında Pozisyon yığın aşımı (Position stack overflow in shape nnn) hata mesajı alınacaktır.

Kod 005 ile yığına alınan pozisyon(lar) kod 006 kullanılarak alındıklarında, pozisyon yığını boşaltılmış olacaktır. Benzer şekilde, Pozisyon yığınındaki geçerli kayıt sayısından fazla sayıda kod 006 uygulandığında, compile işlemi sırasında herhangi bir hata mesajı alınmayacak, tanımlanan Shape kullanılmaya çalışıldığında, aynı hata mesajı alınacaktır.

007 : Bir sonraki byte ile verilen numaralı Shape'i çizer.

Unicode olmayan font dosyalarında kod 007'yi takip eden byte 1 ile 255 arasında bir karakterin ASCII kodunu ifade eden tamsayıdır. Unicode font dosyalarında bu karakter numarası 1 ile 65535 arasında 2 byte ile ifade edilir. Shape Tanım dosyalarında ise, aynı dosya içinde tanımlanan ve numarası verilen shape numarasıdır, Numarası yazılan shape, alt-shape olarak çizilecektir.

Shape tamamlanınca geçerli shape tanımına dönülür. Kod 001 ve kod 002 ile tanımlanan çizim durumları ile kod 003 ve kod 004 ile tanımlanan ölçek değerleri, alt-shape sonunda eski hallerine döndürülmezler.

Alt-Alt-Shape kullanırken, çağırılan Alt-Shape içinde Çizim Durumu, Ölçek faktörü değişiklikleri gerekiyorsa eski durumlarına döndürmek için gerekli byte'lar yazılmalıdır.

Normalde vektör tanımları önceden belirlenmiş (yukarıda Vektör Yön Kodları şeklinde verilen) 16 yönde ve en çok 15 birim uzunluğunda olabilirler. Bu durum shape tanımını çok sade ve verimli bir hale getirmekle birlikte, bazı durumlar için sınırlamalar getirilmiş olur. Kod 003 ve kod 004 kullanılarak 15 birim uzunluğundaki vektör uzunluğu sınırı aşılmış olur. Kod 008 ve kod 009 kullanılarak bahsedilen normal vektörler dışında, daha özgür ve sınırsız yön ve doğrultularda vektörler tanımlanabilir.

008 : takip eden 2 byte ile X-Y yer değiştirmeleri belirlenir.

8,X-yerdeğiştirme,Y-yerdeğiştirme şeklinde yazılır. Buradaki X ve Y yer değiştirmeleri -128 ile +127 arasında değişen tamsayılarla ifade edilirler. Negatif değerler için tamsayının başına - işareti konur, pozitif yer değiştirmelerde + işaretinin yazılması isteğe bağlıdır. Okunabilirliği ve takip edilebilirliği kolaylaştırmak için parantezler kullanılabilir.

8,(-10,3) örneğinde, 10 birim sola ve 3 birim yukarıya yerdeğiştirme tanımlanmış olur. Kod 008'i takip eden 2 byte tamamlandığında shape tanımı normal vektör moduna döner.

009 : Bu kodu takip eden ve (0,0) ile sonlandırılan seri haldeki byte çiftleriyle belirlenen çoklu X-Y yer değiştirmeleri tanımlanır.

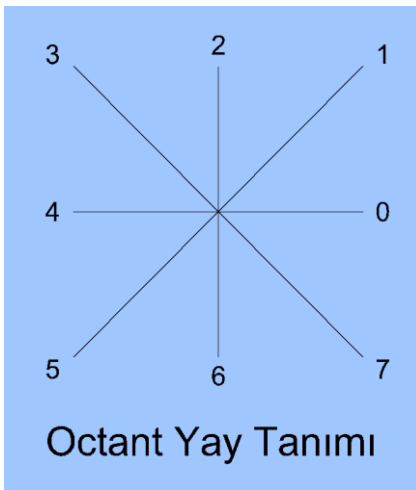
Kod 009 kullanarak standart olmayan vektörler serisi çizilebilir. Prensip olarak Kod 008'in çoklu uygulaması olarak düşünülebilir. Vektörler serisini tanımlayan X-Y yer değiştirme çiftleri herhangi bir sayıda olabilir. Kod 009 dan sonra gelen vektörler dizisi, (0,0) yer değiştirme vektörüyle sonlandırılır.

9,(3,1),(3,2),(2,-3),(0,0) örnek kodu ile üç tane standart olmayan vektör çizilerek Normal Vektör durumuna dönülür. Kod 009 dan sonra gelen X-Y yer değiştirmeleri serisi (0,0) ile sonlandırılmalıdır. AutoCAD sonlandırma byte çiftini gördüğünde, takip eden byte'ların Normal Vektör veya Özel kod olduğunu anlayacaktır.

00A : Takip eden 2 byte ile belirlen Octant Arc çizme. Octant Arc : Tam çemberin 8'de 1'i kadar Yay

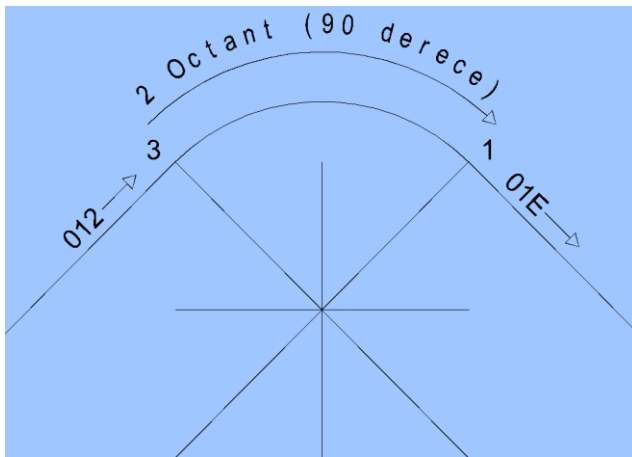
Tanımdan da anlaşılacağı gibi, 45 derece ve katlarından birinden başlayıp, gene 45 derece ve katları kadar yay çizme işlemi. Aşağıdaki şekilde görüldüğü gibi, 1/8'lik adımlar, saat 3 yönünden başlayıp saatin tersi yönde ilerleyecek şekilde belirlenmiştir.

Arc tanımı; **10,radius, (-)OSC** biçimindedir.



İlk tanım byte'ı radius 1 ile 255 arasında bir tamsayıdır. İkinci tanım byte'ı çizilecek yayın yönünü (pozitif saatin tersi yönü, negatif değer saat yönünü belirler), başlangıç octant'ını (0'dan 7'ye kadar) ve yayın uzunluğunu (octant sayısı cinsinden) ifade eder. yay uzunluğu olarak 0 girilirse 8 octant alınır, bu da tam çember anlamına gelir. Okunabilirliği ve takip edilebilirliği artırmak için parantezler kullanılabilir.

...012,10,(1,-032),01E,... shape tanımı bölümünü örnek olarak ele alırsak;



Şekilde görüldüğü gibi, sağa-yukarı bir birim uzunluğunda vektör, yarıçapı 1 birim, saat yönünde, 3 octant'ından başlayan, 2 octant devam eden yay, sağa-aşağı bir birim uzunluğunda vektör çizilmiş olur.

00B : Takip eden 5 byte ile belirlenen Yay Parçası çizme.

Kod 00B ile yay tanımlama özel kodlar arasında en karmaşık ve anlaşılması zor olanıdır.

11,start_offset,end_offset,high_radius,radius,(-)0SC şeklinde yazılır.

start_offset ve end_offset değerleri yayın başlangıç ve bitişinin octant sınırlarından ne kadar uzakta olduğunu belirler. high_radius, radius değerinin en anlamlı üst 8 bit'ini ifade eder, radius değeri 255 birimden büyük olmadıkça high_radius 0 dır.

high_radius değeri 256 ile çarpılarak radius değerine eklenerek bulunan değere göre 255'ten büyük yarıçapa sahip yay çizilmiş olur. radius ve bitiş tanımlamaları (daha önce açıklanan) kod 00A ile octant Yay tanımlama ile aynıdır. Kod 00A ile radius değeri 255'e kadar olan yaylar çizilebilirken, kod 00B ile radius değeri 255'ten büyük yaylar da çizilebilir.

start_offset başlangıç octant'ından (45 derece ve katları) yay başlangıcının derece cinsinden farkı (mesafesi) dir. Bahsettiğimiz derece cinsinden fark 256 ile çarpılıp, 45'e bölünerek elde edilir. Eğer tanımlayacağımız yay tam olarak octant noktalarından birinden başlıyorsa, start_offset değeri 0 olacaktır. end_offset değeri de aynı mantıkla belirlenecektir. Yayın bitim noktasındaki octant sınırından aynı mantıkla hesaplanan derece cinsinden değer çıkarılarak elde edilir. Eğer yay bitimi tam olarak octant noktalarından biri ise end_offset 0 olacaktır.

Örneğin, 55 dereceden başlayan, 95 derecede biten, 3 birim yarıçapında bir yay tanımlamak için;

11,(56,28,0,3,012) şeklinde kodlanmalıdır. Kodlanan değerleri teker teker açıklarsak;

start_offset = 56-> $((55 - 45) * 256 / 45) = 56$

end_offset = 28-> $((95 - 90) * 256 / 45) = 28$

high_radius = 0 -> (radius < 255)

radius = 3

starting octant = 1-> Yay başlangıcı 45 ° ile sonraki octant (90 °) arasındadır.

ending octant = 2-> Yay bitışı 90 ° octantı ile sonraki octant (135 °) arasındadır.

00C : X-Y yer değiştirmeleri ve Bombe değeri ile Yay tanımlama.

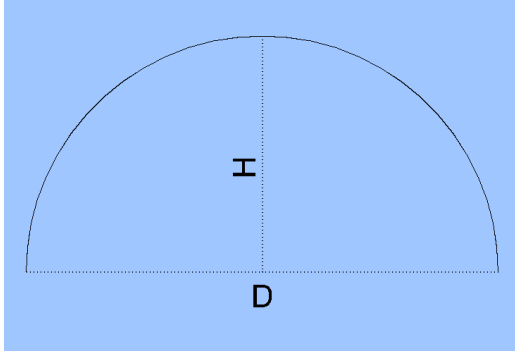
Daha önce açıklanan kod 008 ve kod 009 ile X-Y yer değiştirmelerinin tanımlanmasına çok benzer. Kod 00C ve kod 00D özel kodları kullanılarak X-Y yer değiştirmelerinin yanı sıra, yer değiştirme vektörünün bir fonksiyonu olarak bombe tanımlanır. 00C ile tek bir Yay parçası tanımlanırken, 00D ile (0,0) tanım byte'ı ile sonlandırılıncaya kadar çoklu yay tanımlanabilir.

Arc tanımlamak için 00C kodunu takip eden üç byte kullanılır:

00C,X Yerdeğiştirmesi,Y yerdeğiştirmesi, Bombe (Bulg)

Yay parçasını tanımlayan her üç parametre de -127 ile +127 aralığında birer tamsayıdır.

X-Y yer değiştirmeleri aşağıdaki şekilde görüldüğü gibi D ile ifade edilirse, yayın orta noktasının D doğrusuna dik uzaklığıda H ise, bombenin büyüklüğü: $((2 * H / D) * 127)$ dir.



Yer değiştirmenin işareti negatif ise, geçerli konumdan saat yönünde dönülerek yeni konum belirlenir.

Yarım çember 127 (veya -127) bombe değerine sahip olduğuna göre, daha büyük yaylar bu kodlar kullanılarak birbirini takip eden yay tanımlarıyla oluşturulabilir.

Bombe değeri için 0 kullanımı geçerli bir yöntemdir ve bombesi 0 olan yay, düz çizgidir. (bu kavramı paylaştığım AutoLISP fonksiyonlarda kullanmış ve açıklamıştım.)

Kod 008 ile düz çizgi tanımlanabilirse de, çoklu yay tanımlarının arasında yer alan düz çizgi tanımı için 00D kodundan çıkmadan, fakat düz çizgi olacak bölümün bombesi 0 girilerek, birbirini takip eden çok sayıda yay ve düz çizgi kombinasyonu tanımlanabilir. 00D kodunu takip eden 3'lü guruplar halindeki yay tanımları (0,0) şeklinde 0 yerdeğiştirme tanımıyla sonlandırılır. En son girilen 0 yer değiştirme değerinden sonra bombe (bulg) girilmez. Örneğin S harfi basitçe şu şekilde tanımlanabilir; **13,(0,5,127),(0,5,-127),(0,0)**

Sıfır (0) bombeli yay tanımı çoklu yay tanımları arasındaki düz çizgilerin tanımlanmasında çok kullanışlıdır; Bu yöntem çoklu yay tanımından çıkıp, düz çizgi tanımladıktan sonra yeni bir çoklu yay tanımına girilmesine göre çok daha verimlidir. Yay parçası ve çoklu yay tanımları içinde -128 sayısı kullanılamaz.

00D : Yer değiştirmeler ve bombe değerleriyle tanımlı Çoklu Yay tanımı.

00D özel kodu, 00C kodunun çoklu kullanımıdır, kullanılışı ve parametreleri, yukarıda açıklandığı gibi, 00C ile aynıdır. Özel kod 00D ile tanımlanan Yay parçaları serisi (0,0) vektörü ile sonlandırılmalıdır.

00E : Bu koddan sonra gelen kısım yalnızca Font dosyalarında ve Vertical Text için geçerlidir.

Karakter tanımlarında 00E özel kodu ile karşılaşıldığında takip eden kodlar Yazı doğrultusuna bağlı olarak işleme alınır veya işleme alınmadan geçerli. Yazı doğrultusu (Text orientation) dikey (vertical) ise bu kodlar işleme alınır, aksi halde işleme alınmaz, bir sonraki kod işleme alınarak devam edilir.

Yatay yazılarda (Horizontal Text, yazı açısı ne olursa olsun) karakterin başlangıç noktası (okunabilme yönüne göre) sol alt köşe olarak alınır. Dikey yazılarda (Vertical Text) ise başlangıç noktası karakterin üst orta noktası kabul edilir. Her karakter tanımının sonunda, kalem kaldırılarak (kod 002) bir sonraki karakterin başlangıç noktasına gidilir. Bu harf aralığı gidişi Yatay Yazılarda Sağa doğru, Dikey Yazılarda Aşağı doğrudur.

00E (14) özel kodu, karakterin başlangıç ve bitiş noktalarını yazının Yatay yada Dikey olmasına bağlı olarak, bir kerede tanımlamak için kullanılır. Böylelikle Yatay ve Dikey yazılar için ayrı ayrı karakter tanımlama gereği ortadan kaldırılmış olur.

Aşağıdaki örnek kod ve iki şekil ile büyük **A** harfi için hem Yatay hemde Dikey yazı için söz konusu başlangıç ve bitiş noktalarının tanımlanması açıklanmıştır.

```
*041,27,uca
2,14,8,(-8,-21),1,8,(8,21),8,(8,-21),
2,8,(-13,7),1,0A0,2,8,(9,-7),14,
8,(-14,-10),0
```

