



**UGSEAD**  
**Uluslararası Güzel Sanatlar Eğitimi Araştırmaları Dergisi**  
**International Journal of Research in Fine Arts Education**

<http://ugsead.penpublishing.net>



## **Görsel İletişimde Kodlama Destekli Grafik Tasarımı** **Code Aided Graphic Design in Visual Communication**

**Erol ÇİTCİ**

*Mehmet Akif Ersoy Üniversitesi Sanat ev Tasarım Fakültesi, Görsel İletişim Tasarımı Bölümü*

Geliş Tarihi: 21 Mart 2020  
Kabul Tarihi: 01 Haziran 2020

**Anahtar kelimeler:**

Görsel iletişim tasarımı, Kodlama destekli tasarım, Dijital tasarım, Kodlama dilleri

DOI: 10.29329/ugsead.2020.255.2

Received: 21 March 2020  
Accepted: 01 June 2020

**Keywords:**

Visual communication design, Code aided design, Digital design, Coding languages

**ÖZET**

Son yirmi yılda yaşanan dijital dönüşüm, görüntü üretim ve görselleştirme tekniklerini hızla değiştirmiştir. Buna bağlı olarak görsel iletişim tasarımının gelişimi yeni bir ivme kazanmıştır. Uygulama ve Arayüz tasarımı, hareketli grafikler ve kinetik tipografi, logo ve fontlar, sanal gerçeklik, artırılmış gerçeklik, bilgilendirme grafikleri veya veri görselleştirmeleri gibi örnekler görsel iletişimin programlama disipliniyle olan yeni bağlamına işaret etmektedir. Her ne kadar masaüstü yayıncılık önemini koruyor olsa da görsel iletişim tasarımı bilgisayar destekli tasarımdan kodlama destekli tasarıma doğru evrilmektedir.

Bu bağlamda, kodlama becerilerinin geliştirilmesi görsel iletişim tasarımcıları açısından önemli bir etken haline gelmiştir. Fakat, kodlama alanının genişliği, öğrenilmesi gereken kodlama dili veya dillerinin türlerini ve kodlamanın görsel iletişim tasarımı alanına sağladığı olanakları belirsizleştirmektedir. Bu çalışma, kodlama becerilerinin görsel iletişim tasarımcılara sağladığı faydaları ve görsel iletişim tasarımında hangi kodlama dillerinin kullanılabileceğini araştırmıştır. Ayrıca çalışmada kodlama dillerinin kullanıldığı belli başlı görsel iletişim örnekleri de derlenmiştir.

© UGSEAD All rights reserved

**EXTENDED ABSTRACT**

The digital transformation that takes place in the last vicennial has rapidly changed the image production and visualization techniques. These new developments have brought a new momentum to the development of visual communication design. Examples such as application and interface design, motion graphics and kinetic typography, variable logos and fonts, virtual reality, augmented reality, info graphics or data visualizations point to the new context of visual communication design with the field of programming. Although desktop publishing remains its importance, visual communication design evolves from computer aided design to code aided design. Besides, the digitization of current communication tools and the spread of multimedia tools, users are giving more preference to visual and dynamic content presentations. Therefore, the importance of digital interface design, visualization methods and exploring new expression possibilities has increased very much. In this context, the development of coding abilities of visual communication designers has become an important factor in the digital age. Therefore, visual communication designer has to improve their knowledge about programming. But the main problem is that the scope of programming obscures which types of programming languages must be learned, and which possibilities and advantages offers to the field of visual communication design.

This study aims to investigate the benefits of coding skills and coding language types which is useful for visual communication designers. In addition, some of code-aided visual communication designs examples have been compiled.

The results of this study showed that scripting language suites the best for visual communication designers. These programming languages are in the category of high-quality language. Therefore, it is very easy for non-programmers to write and read scripting language. Scripting languages also have the advantages of being used in CAD programs' script editors. Languages such as JavaScript, AppleScript, VisualBasic, ActionScript, MaxScript, CommandScript or MelScript are examples of scripting languages. Among the scripting languages, JavaScript is the most widely used by visual communicators. The main reason for this is that Javascript is widely used in web browsers and is recognized by many CAD programs. Apart from the scripting languages used in script editors, it is also possible to use coding languages that have their own software directly. The widely used languages of these kind are Python, Java based Processing and C ++ based OpenFrameworks. These languages are preferred by artists and designers to design static images as well as animated and interactive videos and animations. Another graphic production technology based on coding is Computational Graphic Design tools. These design tools are known to be used in design fields such as architecture, industrial and textiles. Computational design software are applications written for a specific design job and enabling designs that are either difficult or not realized to design in CAD programs.

Studies conducted in the context of this study have shown that coding aided graphics and designs can be covered under three main topics. The first one is the scripting languages that increase the features of CAD programs. The second method is to use programming languages that have their own interfaces. Third method is using computational graphic design tools which designed (programmed) to perform certain tasks. Another important point of this study is the programming skills of the visual communication designers. In an environment where almost all images are digitized, visual communication designers have to learn more or less the possibilities of programming languages. Learning to program is a challenging process because it is a very wide area. The levels of coding languages are increasing day by day. But in any way the number of visual communication designers using coding as a design tool has increased. With its expanding libraries and newly added tags, coding languages offer visual communication designers the opportunity to produce

## 1. Giriş

1980'lerin başından itibaren bilgisayar destekli tasarım teknolojilerinin güdümünde gelişen grafik üretim, yöntem ve teknikleri 21. yüzyılla birlikte yeni bir yön ve ivme kazanmıştır. Çözünürlükleri sürekli artan ve esneyebilen ekran teknolojileri, etkileşimli yüzeyler ve bunlarla gelişen yazılım teknolojileri, iletişim olanaklarını her geçen gün daha da ileriye taşımaktadır. Mültimedya yani çoklu ortam araçların yaygınlaşmasıyla birlikte mesaj sunumları artık dinamik ve etkileşimli görsellerle gerçekleşebilmektedir. Kullanıcılar ise bilginin sürekli bir akış içinde anlık, dışavurumcu ve hikâye anlatım (storytelling) tekniklerinin yaratıcı bir biçimde kullanıldığı yöntemlerle aktarılmasını tercih etmektedirler. Mesaj iletiminde yaşanan bu değişim ve dönüşüm yeni gösterim teknik ve yöntemlerinin kullanılmasını gerektirerek görsel iletişim tasarımı yeni bir aşamaya taşımıştır.

Bir zamanların yenilikçi bir tasarım yöntemi olan masaüstü yayıncılık artık geleneksel bir üretim biçimine dönüşmüştür. Masaüstü yayıncılık, hala önemini korumakla beraber, çağdaş grafik üretim yöntemlerinin gerisinde kalmıştır. Günümüzde yayıncılık baskı teknolojilerinden dijital yayıncılığa doğru yönelmiştir. İletişim dijitalleşerek web ve mobil uygulama platformlarının yaygınlaşmasına neden olmuştur. Sanal ve artırılmış gerçeklik ortamları ve holografik görüntüler *deneyimi* tasarımın bir parçası haline getirmiştir. Bütün bu gelişmeler görsel iletişim tasarımı alanının baskı teknolojilerinden ziyade dijital görüntü teknolojilerinin gerektirdiği yeni şartlarına göre konumlanmasını zorunlu hale getirmiştir. Bu bağlamda görsel iletişim tasarımcılarının kodlama (programlama) yapabiliyor olmaları veya bu becerilerin geliştirilmesi önemli bir konu olarak öne çıkmıştır. Bunun başlıca nedeni görsel iletişim tasarımı ve yazılım alanlarının birbirleriyle kesişerek disiplinlerarası bir çalışma ortaya koymasından kaynaklanmaktadır. Örneğin tasarım programları plug-

in veya script destekleriyle ek özellikler kazanarak daha etkili ve kapsamlı bir biçimde kullanılabilirler. Bazı tasarımcıların doğrudan Phyton, Processing ve Javascript gibi kodlama dillerini kullanarak hareketli ve etkileşimli grafikler tasarlayabildikleri görülmektedir. Web ve mobil uygulama (application) tasarımlarında PHP ve CSS kodlarına ihtiyaç duyulmaktadır.

## 2. Çalışmanın Amacı

Dijital çağda gerçekleşen teknolojik gelişmeler görsel iletişim tasarımcılarının kodlama becerilerinin geliştirmelerine ilişkin yeni bir durumun gelişmesine neden olmuştur. Günümüzdeki örnekler görsel iletişim tasarımcısının herhangi bir tasarım yazılımını kullanabiliyor olmasının, çağdaş grafik tasarım açısından ayrıcalıklı bir yeti olmadığını göstermektedir. Ayrıca görsel iletişimcilerinin tasarım becerilerini programlama bilgisi ile zenginleştirmeleri hem çağın getirdiği tasarım şartlarını takip etmek için hem de bilgisayar teknolojilerinin görsel iletişim tasarımına sunduğu fırsatları değerlendirerek yeni, özgün ve daha etkin grafiksel ifadeler elde etmek adına önemlidir. Bu çalışmada görsel iletişim tasarımının güncel durumu ele alınmış olup, kodlama teknolojileri bağlamında dijital teknolojilerin getirmiş olduğu yeni tasarım koşulları ele alınmıştır. Çeşitli örnekler üzerinden de kullanılan bazı tasarım örnekleri incelenerek, görsel iletişimde kullanılabilecek kodlama dilleri türleri ve kodlama dillerinin genel olarak alana sağlamış olduğu katkı ve yeni olanaklarının araştırılması amaçlanmıştır.

Bu çerçevede, öncelikli olarak programlamaya ilişkin bazı temel kavramlar açıklanmaya çalışılmıştır ve görsel iletişim tasarımı özelinde tasarım dünyasında gerçekleşen bilgisayar destekli tasarımdan *kodlama destekli tasarıma* geçiş süreci örnekler üzerinden ortaya konmaya çalışılmıştır.

## 3. Çalışmanın Önemi

İletişim araçlarının dijitalleşmesi, çoklu ortam araçlarının, web ve mobil uygulama platformlarının yaygınlaşması ve kullanıcıların daha çok etkileşimli görselleri ve dinamik içerik sunumlarını daha çok tercih etmeleri nedeniyle dijital arayüz tasarımlarının, görselleştirme yöntemlerinin ve bilgisayar teknolojilerinin sağladığı yeni ifade olanaklarının kullanılmasının önemini arttırmıştır. Fakat çoklu ortam araçlarının geliştirilmesi ve bu araçlarda kullanılacak olan uygulamaların (application) tasarımı ve görselleştirme tekniklerinin geliştirilmesi disiplinlerarası bir çalışma gerektirmektedir. Bu alanda yer almak isteyen görsel iletişim tasarımcılarının da programların işleyişine, algoritmalara ve programlama alanının terminolojisine hâkim olmaları gerekir. Ayrıca bireysel olarak çalışan görsel iletişim tasarımcıları da başta web tasarımlarında olmak üzere etkileşimli ve hareketli grafikler, uygulama tasarımları, animasyonlar ve bilgilendirme grafikleri gibi tasarımlarda da zaman zaman kodlama desteğine ihtiyaç duymaktadırlar. Bu nedenle görsel iletişim tasarımcılarının kodlama bilmeleri belirli avantajlar sağlamaktır. Öncelikli olarak yazılımcılara olan bağımlılık azalmakta ve kodlama dillerinin sağladığı olanaklardan daha fazla yararlanmasını sağlamaktadır. Ancak kodlama yani programlama konusu çok geniş bir uzmanlık alanıdır. Yüzlerle ifade edilen programlama dilinin varlığından bahsedilmektedir. Dolayısıyla görsel iletişim tasarımcısının kullanabileceği ve ihtiyaç duyduğu programlama dillerinin belirlenmesine ihtiyaç duyulmaktadır.

Bu çalışma, görsel iletişim tasarımcıları açısından kodlama becerilerine sahip olmanın önemini açıklamasının yanı sıra tasarım çalışmalarında ne tür ve hangi programlama dillerinin kullanılabileceğini ortaya koyması bakımından önemlidir. Çalışmanın bir başka önemli noktası ise görsel iletişim tasarımında kodlamanın kullanıldığı belli başlı örnekleri derlemesidir. Toplanan örnekler ve tespit edilen kodlama türleri bu alanda çalışmak isteyen tasarımcılara rehber niteliği taşımaktadır.

## 4. Temel Kavramlar

### 4.1. Programlama

Programlama ya da diğer adıyla bilinen kodlama genel anlamda yazma süreci, test etme ve hataları ayıklama gibi birçok aşamayı içinde barındıran ve bireyin yoğun bir bilişsel beceri sergilemesini gerektiren karmaşık bir problem çözme sürecini ifade eder (Cevahir ve Özdemir, 2017:

322) Bilişim teknolojileri bağlamında düşünüldüğünde ise bilgisayarlara, tabletlere, akıllı telefonlara veya programlanabilir herhangi bir cihaza komut verebilmek için, belirli amaçları gerçekleştirebilmelerini sağlamak için elektronik aygıtların programlanması olarak tanımlanabilir.

#### 4.2. Sistem Programlama Dilleri ve Kodlama Dilleri

Makine ve bilgisayarların belirli görevleri yerine getirebilmeleri için programlanmaları gerekmektedir. Bu nedenle yazılımcılar tarafından farklı görev ve amaçlar çerçevesinde farklı kodlama dilleri geliştirilmiştir. Bu programlardan bazıları çok büyük sistemsel görevler için yazılırken, bazıları daha küçük görevleri yerine getiren, daha özelleşmiş programlama dilleridir. Bütün diller insanlar tarafından kolay okunabilme derecesine göre yüksek veya düşük seviyeli programlama dilleri olarak sınıflandırılırlar (Karabak ve Güneş, 2013:176). Görsel iletişim tasarımında kullanılan diller yüksek seviyeli dil grubunda yer alan dillerdir. Ancak görsel iletişim tasarımcılarının ihtiyaç duydukları yüksek seviyeli dillerin de sınıflandırılmasına ihtiyaç vardır. Bunun nedeni insan tarafından kolay okunduğu varsayılan Pascal, C veya C++ gibi yüksek seviyeli diller görsel iletişim tasarımcılarının ihtiyacının çok ötesindeki dillerdir. Bu tür dillerin öğrenilmesi görsel iletişim tasarımı açısından zorunlu değildir. Ayrıca yüksek seviyeli dillerin sınıflandırılmaması önemli bir soruna neden olmaktadır. Hangi kodlama dilinin öğrenilmesi gerektiğinin bilinmemesi bu alanda çalışmak isteyen görsel iletişim tasarımcılarının bir kısmında bir direnç oluşturmaktadır. Bu nedenle yüksek seviyeli dillerin derlenmesi ve sınıflandırılması bazı tasarımcılar için de bir kapı aralayacaktır.

Yüksek seviyeli programlama dillerinin sınıflandırılması konusunda Ousterhout'un (1998) çalışması yol gösterici niteliğindedir. Ousterhout yüksek seviyeli diller *sistem programlama dilleri* (system programming languages) ve *kodlama dilleri* (scripting languages) olarak iki alt başlıkta ele almıştır. (*Kodlama dilleri* bazı kaynaklarda betik diller, script dosyası veya komut dosyaları olarak da adlandırılırlar). Bunun temel nedenini *kodlama dillerinin*, *sistem programlama dillerinden* farklı görevler için tasarlanmış olmalarına bağlar ki bu da *sistem programlama dilleri* ve *kodlama dilleri* arasındaki temel farklılıkları oluşturmaktadır.

Sistem programlama dilleri, bellek sözcükleri gibi en ilkel bilgisayar öğelerinden başlayarak sıfırdan veri yapıları ve algoritmalar oluşturmak için tasarlanmıştır. Buna karşın, 'kodlama dilleri' yapıştırma (glue) için tasarlanmıştır. Yani kodlama dilleri bir dizi güçlü bileşenin varlığını varsayarlar ve öncelikle bileşenleri birbirine bağlamak için tasarlanmıştır. Sistem programlama dilleri karmaşıklığı yönetmeye yardımcı olmak için güçlü bir şekilde yazılırken, kodlama dilleri bileşenler arasındaki bağlantıları basitleştirmek ve hızlı uygulama geliştirmeyi sağlamak için belirli bir kategoriye ait değildiler. (Ousterhout, 1998: 23)

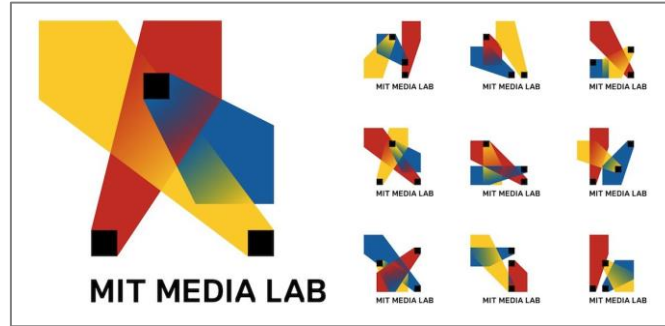
Ousterhout'ın (1998: 26) önemle üzerinde durduğu bir başka nokta da *kodlama dillerinin* genellikle yorumlandıklarını ve *sistem programlama dillerinin* ise genellikle derlendikleridir. Buna göre Pascal, C, C++ ve Java gibi diller *sistem programlama dilleri* olarak sınıflandırılırken; Perl, Python, Rexx, Javascript ve Visual Basic gibi programlama dilleri *kodlama dilleri* olarak kabul edilmektedir. Ousterhout'ın sınıflandırmasından yola çıkarak görsel iletişim tasarımcılarının kullandıkları PHP, CSS, ActionScript, MaxScript, AppleScript ve MelScript gibi diller de üstlendikleri görevler itibarıyla *kodlama dilleri* olarak sınıflandırılırlar. Genel itibarıyla *kodlama dillerinin* özellikleri şunlardır:

- Kodlama dilinin belirttiği eylem basit olabileceği gibi karmaşık da olabilir.
- Geçerli bir belgedeki tek bir nesneyi etkileyebilir veya birden fazla nesneyi karmaşık hale getirebilir ve etkileyebilir.
- Bir uygulama yazılımında yinelenen görevleri otomatik hale getirebilir.
- Manuel olarak yapmak çok zaman alan işleri kolaylaştırmak için yaratıcı bir araç olarak programlanabilir. (Adobe, 2007)

## 5. Kodlama Destekli Grafik Tasarımı

Bilgisayar kodlarının tasarım dünyasında kullanımı 1965–1975 yılları arasında gerçekleşen *erken kompüstasyonel çağı* olarak da adlandırılan dönemde mimarlık alanında yapılan araştırmalarla başlamıştır. Son yıllarda mimarlıkla birlikte mühendislik, endüstriyel tasarım ve tekstil tasarımında yaygınlık kazanmaya başlayan hesaplamalı tasarım yaklaşımlarının temeli bu dönemde atılmıştır. Hesaplamalı tasarım teknolojileri bilgisayar işleyişinin çift değişkenli birimi olan 0 ya da 1'in birbiriyle oluşturdukları kombinasyon değerleri ile şekillenmektedir. Dolayısıyla Parametrik Tasarım, Üretken (Generative) Tasarım, Algoritmik Tasarım, Evrimsel Tasarım ve Hesaplamalı (Computational) tasarım gibi kombinasyon temelli yaklaşımlar bilgisayarların işleyiş mantığını içeren hesaplamalı tasarım anlayışlarıdır. Bu tür tasarım uygulamalarında bilgisayar desteğinden çok bilgisayarların işleyiş mantığını içeren kodlama dillerinin desteği ön plana çıkmaktadır. Bu nedenle günümüzdeki tasarım anlayışının İngilizcesi Computer Aided Design (CAD) olan Bilgisayarlı Destekli Tasarımın *Kodlama Destekli Tasarım* kavramına doğru evrildiği görülmektedir.

Benzer biçimde Görsel İletişim Tasarımında da Bilgisayar Destekli Tasarımdan *Kodlama Destekli Tasarım*'a doğru bir yönelim yaşanmaktadır. Generative yani üretken tasarım; belirli tasarım değerlerini değiştirerek yapılan parametrik tasarımlar ya da bir tasarımın belirli bir 'evrimsel' süreç sonucunda şekillenmesi, görsel iletişim tasarımında bilgisayar teknolojilerine bağlı olmaksızın uzun süreden beri var olan tasarım yöntemlerdir. Tasarımın bir süreç olduğu gerçeğinden hareketle günümüzde kullanılan pek çok logonun, fontun ve hatta bir afiş tasarımının belirli parametreler çerçevesinde *üretken* biçimde *evrimleştiği* görülür. Ancak yazılım teknolojilerinin tasarımcılara sağladığı olanaklar hesaplamalı tasarım alanını, belirli bir tasarım tarzına dönüştürerek, onu görünür hale getirmiştir. MITLab logosu (Resim 1) veya *Casa da Musica Logo Generator* (Resim 7) yazılımı, görsel iletişim tasarımı alanında kullanılan hesaplamaya dayalı tasarım ürünleridir. Değişken yazıtları ve logolar, bazı hareketli afişler ve duyarlı (responsive) arayüzler de kodlama destekli grafik tasarım örneklerindedir.



**Resim 1.** MIT Media Lab dinamik logo tasarımı, <https://tr.pinterest.com/pin/566538828111468142/>

Tıpkı diğer tasarım disiplinlerinde olduğu gibi görsel iletişime sağlanan kodlama desteği tasarım bağlamında yeni bir anlayış ve yaklaşım getirmektedir. Henüz yeni sayılabilecek olan bu durumun ileriki yıllarda sunacağı imkanlar henüz netleşmemiş olsa da kodlama teknolojilerindeki gelişmeler görsel iletişim tasarımının programlama alanı ile daha çok kesişeceğini göstermektedir. Ancak bugüne kadar geline süreçte yapılan incelemeler kodlama destekli grafik ürün ve çalışmaların üç temel başlık altında ele alınabileceğini göstermiştir. Bunlardan ilki CAD programlarının özelliklerini arttıran script destekleridir. İkinci yöntem ise doğrudan programlama dillerini kullanmaktır. Üçüncüsü ise belirli görevleri yerine getirmek üzere programlatılan hesaplamaya dayalı grafik yazılım araçlarını kullanmaktır.

### 5.1. CAD Yazılımlarında Kodlama Kullanımı

Çok yakın bir zamana kadar grafik tasarımcılardan CorelDRAW, Flash, Illustrator, Photoshop veya In-Design gibi CAD yazılımlarından birini ya da birkaçını kullanabilmeleri istenirken, bilişim teknolojilerinin gelişimine bağlı olarak asgari düzeyde de olsa kodlama becerilerine de sahip olmaları beklenmektedir. Güncel yazılımlar çeşitli kod (script) destekleriyle, daha etkileyici, daha heyecan

verici ve bazı durumlara göre iletişim görevini daha iyi yerine getirebilecek grafiklerin tasarlanmasına olanak vermektedirler.

CAD yazılımlarında kodlama ile tasarım temelde iki şekilde gerçekleştirilmektedir. İlki eklenti ve uzantılar sayesinde, ikincisi ise programların script editörlerini kullanmaktır.

### 5.1.1. Program Eklentileri (Plug-in) ve Uzantıları (Ekstensiyonlar)

Yazılım ya da uygulamaları daha gelişmiş versiyonlarına yükseltmek veya özelleştirebilmek için program eklentilerinin (plug-in, add-ons) veya uzantılarının (ekstention) yüklenmesi gerekir. Örneğin çevrimiçi video izlenebilmesi için Adobe Flash Player veya Quiktime eklentisine ihtiyaç vardır. Yine ağ tarayıcısı üzerinden PDF metinlerini okumak için bilgisayarda PDF okuyucu yüklü olması gerekmektedir. CAD yazılımlarının da buna benzer kendilerine özgü pek çok eklenti ve uzantıları mevcuttur. Shadow Studio, Adobe After Effect programında nesnelere kolayca gölgeleme etkileri verebilen bir plugin iken, Vray de AutoDesk Maya, 3DsMax ve benzeri yazılımlarda hesaplamalı görüntü oluşturan (rendering) bir eklentidir. Scriptographer<sup>7</sup> isimli eklenti Illustrator programında tasarımlara ‘rastgele’ (random) özellikler eklemeye yarar.

Eklenti ve uzantılar genelde çevrimiçi edinilebilirler ve herhangi bir kodlama bilgisi gerektirmeden doğrudan tek bir tıklama ile kurulumları yapılabilir. Fakat programlama bilgisine sahip kişiler kendi eklenti ve uzantılarını yazabilmektedirler. Birçok yazılım şirketi, geliştiricilerin önünü açmak ve kendi düşünemedikleri noktalarda geliştiricilerin yardımında bulunması amacıyla CAD yazılımlarına açık kaynak (Opensource) özellikleri ekleyerek, eklenti ve uzantı yapımını desteklemektedir. Hatta bazı yazılımlar script editörlerinde yazılan scriptleri plug-in olarak kullanılabilmesine olanak sağlamaktadır. Autodesk’e ait 3dsMax ve Maya bu özelliğe sahip programlara örnek oluşturmaktadır. Yani “Maxscript ile yapılan uygulamalar plug-in olarak da kullanılabilirler (Özsağlam, 2009: 291).

### 5.1.2. Script Editörleri

Script editörü günümüzün tasarım programlarının neredeyse tamamında bulunan bir araçtır. Tasarım programlarında işlemler çoğunlukla araçlarla (tools) ve menülerde yer alan özelliklerle gerçekleştirilse de script editörlerini zaman alan, tekrar eden veya hassasiyet gerektiren işlerde kullanmak çok pratik bir yöntemdir. CAD programlarına script editörlerin eklenmesi uzun bir süreden beri var olduğu bilinmektedir (Bkz: Tablo 1). Örneğin Adobe Photoshop’daki script editörünü 2002’de piyasaya sürülen Photoshop 7 ile kullanıma girmiştir. Grafik tasarımcıların kullandığı bir başka yazılım olan CorelDRAW’a script editörünün eklenmesi 1996 yılında CorelDRAW 7 ile gelmiştir. AutoCAD yazılımında ise piyasaya ilk sürüldüğü tarihten (1982) beri script editörü bulunmaktadır. Benzer biçimde Autodesk’in diğer programı olan Maya’da ilk sürüm tarihi olan 1998’den beri Script editörü kullanılmaktadır.

**Tablo 1.**

*Yazılımların tanıdıkları kodlama dilleri listesi.*

Yazılım Adı	Kodlama (Script) Dili
Dreamweaver	HTML, XHTML, PHP, CSS, JavaScript, CFML, Visual Basic, C#
Photoshop	AppleScript, Visual Basic, JavaScript
Illustrator	AppleScript, Visual Basic, JavaScript
AfterEffects	ExtendScript, JavaScript <sup>8</sup>

<sup>7</sup> <http://blog.choppingblock.com/2009/03/generative-art-in-illustrator-scriptographer/>

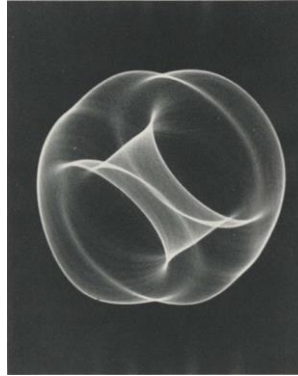
<sup>8</sup> <https://helpx.adobe.com/after-effects/using/expression-language-reference.html>

InDesign	AppleScript, Visual Basic, JavaScript <sup>9</sup>
CorelDRAW	CorelSCRIPT, JavaScript, Visual Basic for Applications (VBA), Microsoft Visual Studio Tools for Applications (VSTA) <sup>10</sup>
Flash (Animate)	ActionScript <sup>11</sup>
3DsMax	MaxScript <sup>12</sup>
Maya	MelScript, Python <sup>13</sup>
AutoCad	CommandScript <sup>14</sup>
Cineam4D	C++, Python <sup>15</sup>

Script editörlerinde komutlar kod yazım kurallarına göre işlenir. Bu nedenle, script editörünü kullanacak görsel iletişim tasarımcıları kodlamaya aşina olmaları gerekir. Ancak yazılımların okuyabildikleri kodlama dilleri farklılık göstermektedir. Bu nedenle tasarımcının kullanacak olduğu kodlama diline hâkim olması gerekir. Adobe uygulamaları olan Photoshop, Illustrator ve In-Design'ın script editörlerinde üç farklı kodlama dilinde komutlar yazılabilmektedir (Bkz.: Tablo 1) Ancak MacOS işletim sistemi kullananlar AppleScript'te komut yazabiliyorken, Windows işletim sistemi kullananlar Visual Basic kodlama dilini kullanmak durumundadırlar. Her iki işletim sisteminde kullanabilecek komutlar yazabilmek için JavaScript kodlama dilinin kullanılması gerekir (McWilliams, 2003; Adobe, 2002; Adobe, 2007).

## 5.2. Doğrudan Kodlama Dilleri Kullanarak Grafik Üretimi

Web tasarımları nedeniyle PHP ve CSS etiketleri görsel iletişim tasarımcılarının kullandıkları ilk bilgisayar kodları olmuştur. Bunun öncesinde kodlamaya dayalı görüntü üretimlerine, Laposky'nin 1950'li yıllarda gerçekleştirdiği ve Dijital Sanatın ilk örnekleri sayılabilecek çalışmalar örnek olarak verilebilir. (Avcı Tuğal, 2018: 117) Bu dönemde çeşitli algoritmalar oluşturularak bilgisayara görüntüler çizdirilmenin başlıca nedeni ekran ve fare teknolojilerinin henüz yeterince gelişmemiş olmasıdır.



**Resim 2.** Oscillon 40, Ben Laposky, 1952, <https://digitalartarchive.siggraph.org/artwork/ben-laposky-oscillon-40/>

<sup>9</sup> <http://turkiyetasarimvakfi.org/wp-content/uploads/2018/11/indesign-dosya-adi-delete.pdf> ve

[https://www.adobe.com/content/dam/acom/en/devnet/indesign/sdk/cs6/scripting/InDesign\\_ScriptingTutorial.pdf](https://www.adobe.com/content/dam/acom/en/devnet/indesign/sdk/cs6/scripting/InDesign_ScriptingTutorial.pdf)

<sup>10</sup> [http://www.softgraf.com/cursodesign/Apostila\\_Macro\\_Programming\\_Guide\\_X7.pdf](http://www.softgraf.com/cursodesign/Apostila_Macro_Programming_Guide_X7.pdf)

<sup>11</sup> <https://helpx.adobe.com/animate/using/writing-scripts.html>

<sup>12</sup> <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/3DSMax-Getting-Started/files/GUID-FE40F021-5444-4709-BF08-DF1F1F0960C3-htm.html>

<sup>13</sup> <https://www.autodesk.com/autodesk-university/class/AutodeskR-MayaR-Scripting-MELtm-and-Python-2012>

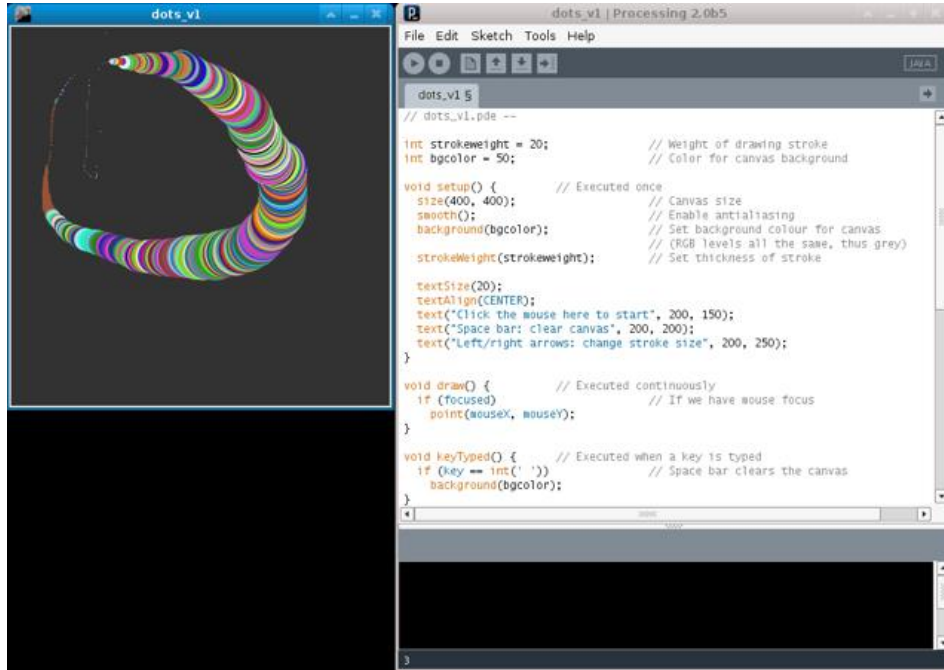
<sup>14</sup> <https://knowledge.autodesk.com/support/autocad-1t/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/AutoCAD-LT/files/GUID-95BB6824-0700-4019-9672-E6B502659E9E-htm.html>

<sup>15</sup> <https://developers.maxon.net/?p=3089>

Bilgisayarın mimaride katı ve mantıksal bir makine olarak değil de yaratıcı bir yükseltme aracı, bir tasarım ortağı olarak görülmeye başlaması 1960'ların ilk yıllarında gerçekleşmiştir. İlk CAD yazılımı ise Ivan Sutherland'ın tasarladığı SKETCHPAD uygulamasıdır (Vardouli, 2012: 25-33). Ancak grafik tasarım da dahil olmak üzere, görsel iletişim tasarımında kodlamanın bir görüntü üretim aracı olarak kullanılması 1990'ların ilk yıllarında başlar. Yaygınlık kazanması ise ancak 2000'lerden sonra gerçekleşmiştir. Processing ve DrawBot gibi basit programlama dillerinin geliştirilmesi ve bunların hem sanatçılar hem de tasarımcılar tarafından bir üretim aracı olarak kullanılmaya başlaması bu yıllara denk gelir. Aşağıda yer alan yazılımlar görsel sanatçılar ve tasarımcıların arasında en yaygın kullanılan kodlama dilleridir.

### 5.2.1. Processing

Processing 2001 yılında sürümü yapılan görsel odaklı bir programlama dili ve ortamıdır. Projenin çıkış amaçlarından biri, tasarım ve sanat öğrencilerine programlamanın nasıl yapıldığını öğretmek ve daha teknik bölümlerde okuyan öğrencilere grafiklerle çalışmanın daha kolay bir yolunu sunmaktır (Reas ve Fry, 2010: vii). 100'den fazla kütüphane içeren yazılımın çekirdek dili ve ek kütüphaneleri, C programlama diliyle aynı öğeleri olan Java'yı kullanır. Çoklu ortam türleriyle (video, resim, ses gibi) sorunsuz ve kolay bir şekilde çalışabilmesi, çok kolay öğrenilebilir olması, programlama prensiplerini anlamaya olanak sağlaması nedeniyle birçok tasarımcı ve sanatçı tarafından kullanılmaktadır (Reas ve Fry, 2007: 1-7).

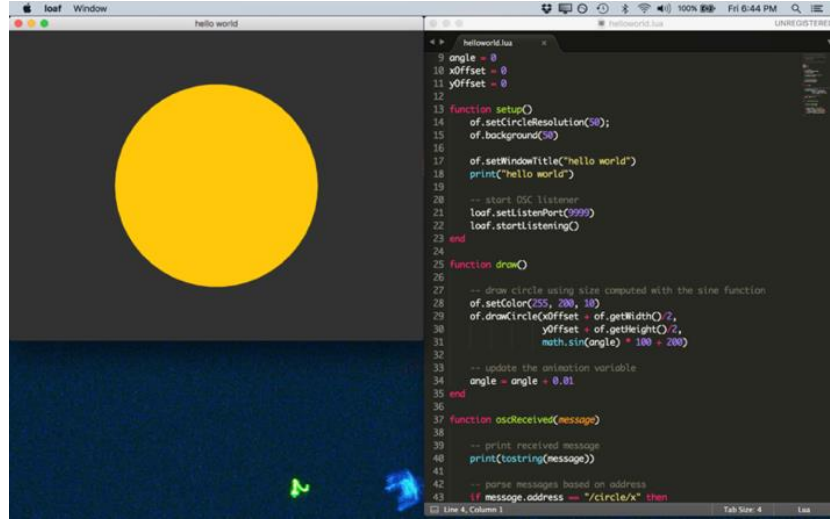


Resim 3. Processing yazılımının arayüzü.

### 5.2.2. openFrameworks (oF)

Zachary Lieberman, Theo Watson ve Arturo Castro tarafından kurulan "Yaratıcı Kodlama" için tasarlanmış açık kaynaklı bir programlama dilidir. Noble (2009) openFrameworks'ı (oF) etkileşimli tasarım ve yeni medya sanatı ile uğraşan sanatçılar ve tasarımcılar için bir çerçeve program olarak tanımlar ve oF komut satırlarının C++ programlama dili ile yazılması nedeniyle kod yazımlarının C++ diline uygun kullanılması gerektiğini aktarır. Üzerine temellendiği dilin çok daha esnek ve düşük düzeyli olması nedeniyle openFrameworks'ta Processing'te mümkün olmayan özellikler tasarlanabilmektedir (Noble, 2009: 153).

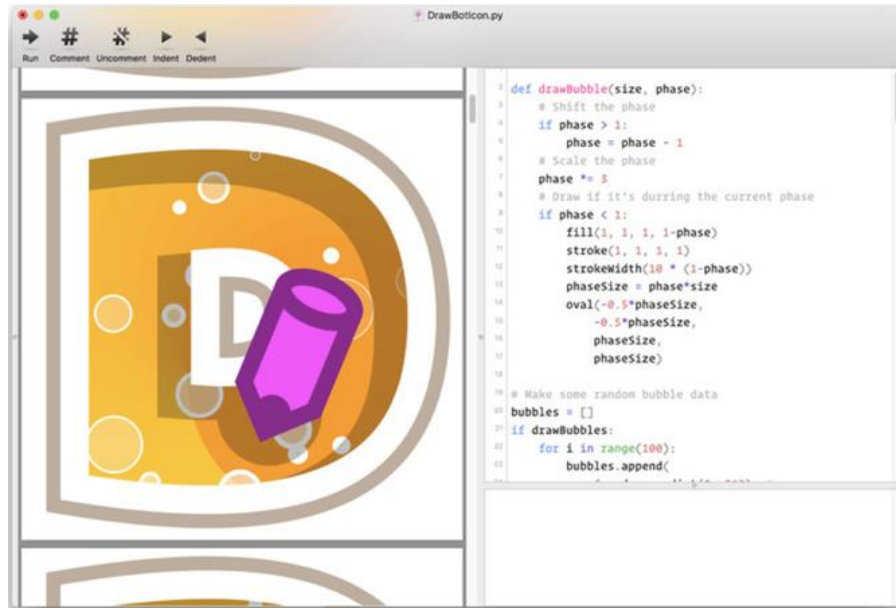




**Resim 4.** openFrameworks yazılımının arayüzü,  
[http://www.danomatika.com/user/pages/03.code/loaf/loaf\\_workflow.png](http://www.danomatika.com/user/pages/03.code/loaf/loaf_workflow.png)

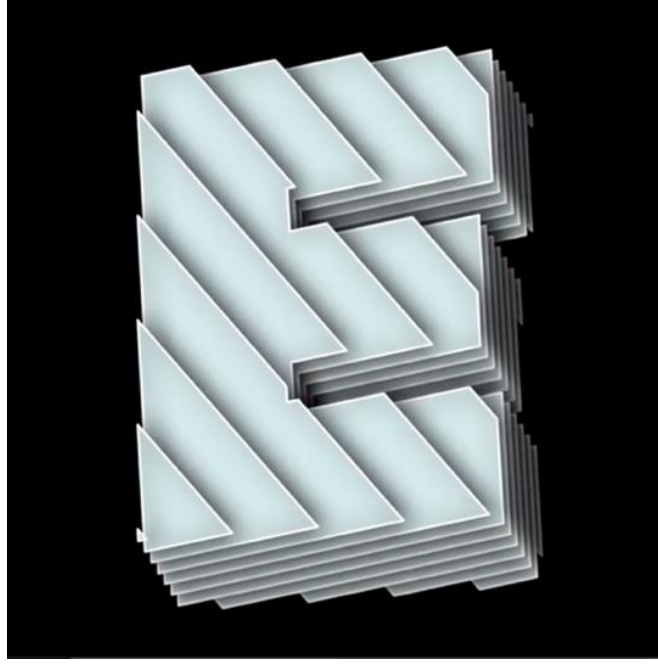
### 5.2.3. DrawBot

2003 yılında DesignRobots adıyla ilk sürümü sunulan DrawBot, iki boyutlu grafikler çizmek için Python programlama dilini kullanan ücretsiz bir MacOS uygulamasıdır. DrawBot, kodlamaya yeni başlayanlar için programlamanın temellerini öğretmek için ideal bir araçtır. Kodlama bilmeyenler DrawBot ile çeşitli grafikler üretirken değişkenlere, koşullu ifadelere, fonksiyonlara aşina olurlar. Elde edilen sonuçlar, yüksek çözünürlüklü olarak pdf, svg, film, png, jpeg, tiff gibi farklı dosya biçimlerinde kaydedilebilmektedir.<sup>16</sup>



**Resim 5.** Drawbot yazılımının arayüzü, <https://learn.adafruit.com/getting-started-with-drawbot?view=all>

<sup>16</sup> <https://www.drawbot.com>



*Resim 6.* Hareketli GIF tasarımı, Matthew Van Den Heuvel, 2016.

#### 5.2.4. CSS ve Javascript

HTML, CSS ve Javascript internet sayfalarının kodlanmasında kullanılan temel teknolojilerdir. Olsson ve O'Brien (2008) bu üç öğeyi web sitesinin içerik, sunum ve davranış katmanları olarak tanımlarlar. İçerik Katmanı olarak ifade edilen HTML, Web sayfalarının yapısını oluşturan temel işaretleme dilidir. Bir web sitesinde karşılaşılan görüntü, HTML adı verilen bu etiketlerden meydana gelen bir yazım topluluğundan oluşmaktadır. Bugün Web'deki içeriğin çoğu metindir, ancak içerikler resimler, animasyonlar, ses, video veya başka şeylerle de sağlanabilir.

Web sayfalarının Sunum katmanı ise CSS'dir, bir HTML belgesindeki font, boyut, renk, veya hizalama gibi öğelerin kullanıcıya nasıl görüneceğini tanımlar (Olsson ve O'Brien, 2008:2). Başka bir tanımlamaya göre ise CSS, HTML'e ek olarak metin ve format biçimlendirme alanında fazladan olanaklar sunan bir Web teknolojisidir. Bir web sitesi içerisinde birbiriyle uyumlu renk ve yazı karakterleri gibi öğelerin her sayfa için ayrı ayrı yönetilmesi yerine tek bir yerden yapılmasını sağlayan bir çeşit kodlama türüdür (Efendioğlu, 2015).

Davranış katmanı, belgeyle eş zamanlı kullanıcı etkileşimini içerir ve bu görev genelde JavaScript tarafından gerçekleştirilir. JavaScript Netscape tarafından 1995 yılının Aralık ayının başlarında piyasaya sürülen bir kodlama dilidir (Goodman, 2007: 7, Pekgöz, 2002: 6). JavaScript ile internet sitelerinde daha kullanışlı, grafik efektleri yüksek ve çeşitli fonksiyonlara sahip web siteleri ortaya çıkmıştır (Özel vd., 2014: 252). Ancak HTML ile, dinamizmi yüksek görüntüler olan açılır kapanır menüler, sürükle-bırak öğeleri, fare ile üzerine gidildiğinde yer veya renk değiştiren nesnelere, kimlik bilgileri doğrulama ya da sipariş formunda yer alan gerekli alanların doldurulması gibi basit işlevler elde etmek oldukça zor olduğu için Javascript dilinden yararlanır.

#### 5.3. Hesaplamalı Grafik Üretim Araçları

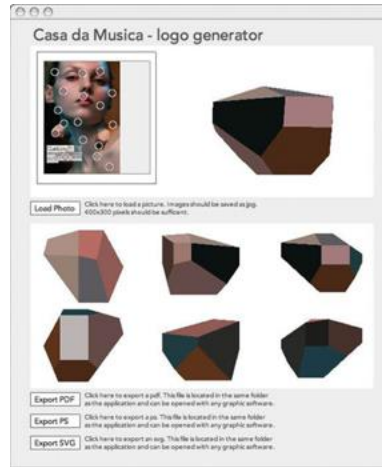
Tasarım çalışmalarında bazen bir fikrin uygulanması CAD programları ile gerçekleştirilmesi imkânsız veya çok zorlu olabilmektedir. Zaman ve parasal kısıtlar devreye girdiğinde tasarım problemini hızlı ve kolay biçimde çözecek bir programın yazılması bir çözüm yolu olabilmektedir. Hesaplamalı grafik tasarım yazılımları bu noktada devreye girerler. Çok karmaşık olmadıkları için kolay kullanımları olan hesaplamalı yazılımlar belirgin bir tasarım görevine yönelik gerçekleştirdikleri görevler dışında ekstra işlevleri yoktur. Bu tür tasarım araçları görsel iletişim tasarımcılarının desteğindeki yazılım şirketleri veya programlama bilen bilgisayar uzmanları tarafından

hazırlanmaktadır. Hesaplamalı grafik tasarım araçlarının tasarımcıya sağladığı en büyük avantaj, daha az veya hiç kodlamaya ihtiyaç duymadan tasarım yapma imkânı sunuyor olmalarıdır. Çalışma çerçevesinde yapılan araştırmalarda hesaplamalı grafik tasarım yazılımlarının logo, kurumsal imaj ve özellikle font tasarımında kullanıldıkları görülmüştür.



**Resim 7.** Casa da Musica dinamik logo tasarımı. <http://www.hexanine.com/zeroside/the-future-is-fluid-inside-dynamic-logos/>

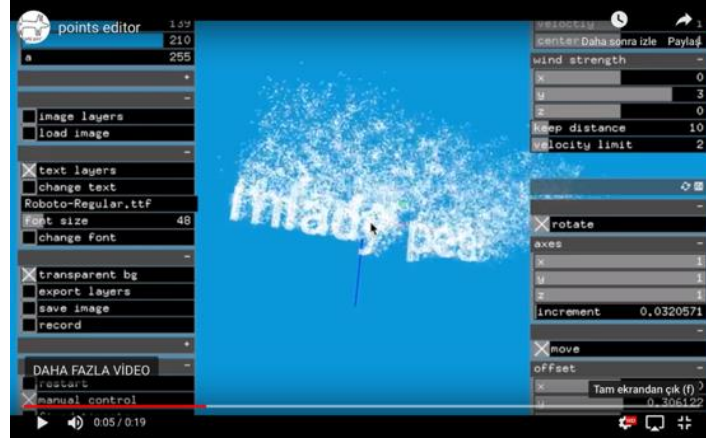
Logo tasarımlarında hesaplamalı tasarım yazılımı için en bilindik örnek Sagmeister & Walsh Tasarım Ofisi tarafından tasarlanan “Casa da Musica” (Resim 7) çalışmasıdır. Logonun değişken yapısı, temsil ettiği kimliği ortaya çıkarmada son derece başarılıdır. Ancak Casa da Musica’yı diğer dinamik logolardan ayıran özellik, renklerinin Casa da Musica Logo Generator (Resim 8) olarak adlandırılan bir yazılım tarafından belirleniyor olmasıdır. Logonun renkleri, yazılıma yüklenen herhangi bir görselde yer alan renklerin seçilerek otomatik olarak atanması ile elde edilmektedir. Başka bir ifadeyle Casa da Musica - Logo Generator, firmanın kimliği ve kullandığı görseller arasında uyumlu bir ilişki yaratan, reklam malzemeleri tasarlarken kullanılacak renk sisteminin sağlanması sürecinde kullanılmaktadır. (Delahunty, 2013: 27)



**Resim 8.** Casa da Musica Logo Generator yazılımının arayüzü,  
[https://www.underconsideration.com/brandnew/archives/the\\_17\\_sides\\_of\\_a\\_cultural\\_id.php](https://www.underconsideration.com/brandnew/archives/the_17_sides_of_a_cultural_id.php)

Points Editor, ‘Mlady Pes’ tasarım stüdyosu tarafından geliştirilen, hesaplamalı grafik tasarım yazılımıdır. 2016 yılında Slovakya’da düzenlenen "Biela Noc" isimli çağdaş sanat festivalinin görsel kimliğini tasarlama görevini üstlenmiş olan 436 tasarım stüdyosunun siparişi üzerine programlanmıştır. Yazılımın hedefi, farklı katmanlar üzerinde bulunan ve noktalardan oluşmuş yazıları, belirli bir zaman aralığında rastgele etrafa savurmaktır. Bu şekilde elde edilen görüntülerden, hareketli veya hareketsiz kesitler olarak festivalin 2016 yılının görsel kimliği oluşturulmuştur. Görsel 9’de basit ve sade tasarımıyla Points Editor’ünün arayüzü görülmektedir. Arayüzünün sağında ve solunda kullanıcı tarafından kontrol edilen değişken parametreler bulunmaktadır. Parametrelerde

yapılan değişiklikleri eşzamanlı olarak izlenebilecek ekran ise arayüzün ortasında yer almaktadır. Resim 10’da ise Points Editor’ünde tasarlanmış olan görüntünün afiş tasarımlarına uygulanmış hali yer almaktadır.



Resim 9. Points Editor yazılımının arayüzü, <https://youtu.be/uVExdURa7rI>

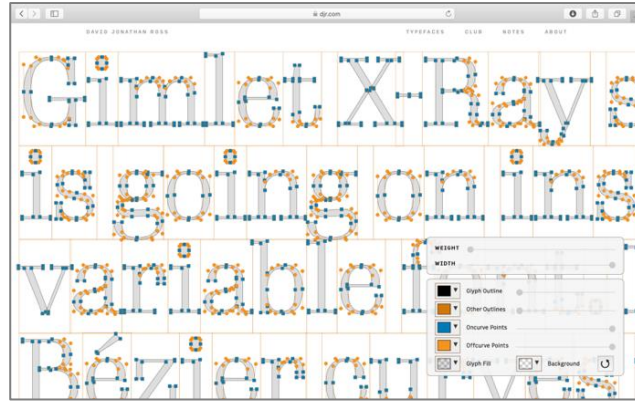


Resim 10. Biela Noc çağdaş sanat festivali 2018 yılına ait afiş tasarımları,  
<https://www.mladyes.sk/computational-design-tool-graphic-design/>

Points Editor yazılımıyla elde edilmiş olan görsel efektler bazı CAD uygulamalarında da tasarlamak mümkündür. Fakat hesaplamaya dayalı bir yazılımda rastlantısallığa dayalı böylesi bir iş çok daha kolay ve esnek bir biçimde uygulama imkânı vardır. Örneğin tasarımcı bütün tasarım sürecini başa almaya gerek kalmadan (CAD yazılımlarında olduğu gibi) uyguladığı bütün efektleri başka bir metne veya yazıyüzüne kolaylıkla uygulayabilme şansına sahiptir. Yani tasarımcı denemek istediği başka bir yazıyüzünü ya da değiştirmek istediği metni, ilgili kod satırında yer alan kısımları değiştirmesi yeterli olacaktır. Böylesi bir kolaylık, tasarımcıyı zorlu ‘üretim’ işçiliğinden kurtararak doğrudan tasarım yapmanın kendisine yönlendirebilme potansiyeline sahiptir.

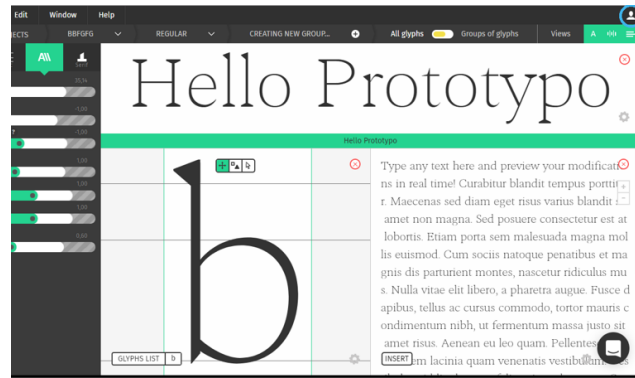
Yazı tasarımında kodlamaya dayalı tasarım örneklerinde de artış olduğu görülmektedir. Yazıyüzü tasarımı zorlu ama aynı zamanda da rasyonel bir süreçtir. Ancak araştırmacıları yazıyı kodlarla tasarlayabilme fikrine iten yine tipografinin rasyonel tarafı olmuştur. Bunun ilk örneği Donald Knuth tarafından geliştirilen Metafont uygulamasıdır. Fakat ana hedefi tipografik kuralları azami düzeyde anlayabilen ve bunları dijital ortama adapte edilmesini sağlamak (Sarıkavak, 2005) olduğu için Metafont bu çalışmanın konusu değildir. Kuşkusuz olarak Knuth’un geliştirdiği Metafont tipografi açısından devrim niteliğindeki gelişmelerden biridir ve tipografinin sayısallaştırılabilirliği açısından çok önemlidir. Fakat bu çalışma bilgisayar kodlarını yaratıcı bir biçimde kullanılarak yeni ve benzersiz formların elde edilmesini sağlayan yazılımları incelemektedir.

Font tasarım yazılımları henüz daha emekleme dönemindeyse de özgün form ve yeni yazıyüzlerini hedefleyen ve genellikle çevrimiçi çalışan uygulamalardır. Özellikle değişken (variable) ve parametrik font tasarımında kullanılan bu tür yazılımlar, belirli tipografik parametrelerde değişiklik yapma suretiyle yeni yazıyüzleri veya yazı ağırlıklarının elde edilmesine olanak sağlanmaktadır. David Jonathan Ross'un tasarladığı ve kodlaması Chris Lewis'in tarafından gerçekleştirilen Gimlet X-Ray<sup>17</sup> yazıyüzü kodlama desteği ile tasarlanan değişken bir yazıtipidir. Çevrimiçi olarak kullanılabilen küçük bir yazılım sayesinde, Gimlet X-Ray (Resim 11) yazıyüzünün ağırlığı, genişliği, gliflerin çizgi kalınlığı veya renk gibi parametrik değerlerinin değiştirilmesi ile farklı ağırlık ve stillerinin elde edilmesi mümkün olmaktadır.



**Resim 11.** Gimlet X-Ray yazıyüzü. Erişim: <https://djr.com/gimlet-xray/>

Bir başka örnek çalışma Prototipo'dur. Prototipo tasarımcılarının kendi ifadeleriyle “font ve kod severler tarafından oluşturulan kullanıcı dostu, parametre tabanlı bir yazıyüzü aracıdır.”<sup>18</sup> Prototipo, bir yazıyüzünü meydana getiren yirmiyi aşkın tipografik parametreyi, kaydırma çubukları aracılığıyla kontrol edilmesini sağlayarak yeni bir yazıyüzü tasarlanmasını sağlayan açık kaynaklı çevrimiçi uygulamadır. Resim 12’de görüldüğü üzere Prototipo yazılımının sol tarafında tasarlanan yazıyüzünün yüksekliği, genişliği, kontrastı ve tırnak yapısı gibi değişkenleri kontrol edilebilen kaydırma çubukları görülmektedir. Ekranın ortasında ve ekranın sağ tarafında ise yapılan değişikliklerin eş zamanlı olarak izlenebileceği, örnek glif ve örnek metin alanı yer almaktadır.



**Resim 12.** Prototipo yazılımın arayüzü, <https://doc.prototipo.io/#setting-yourself-up>

Prototipo yazılımı zorlu ve bazen de meşakkatli bir süreç olabilen yazı tasarımında, tasarımcılara “tekrarlayan görevleri otomatikleştirmek ve tasarımcıyı sadece şekil tasarımına odaklanmasına izin veren ve bu disiplinin tarihsel, matematiksel ve görsel kurallarını birleştiren yeni bir yazı karakteri tasarım yaklaşımı sunar.”<sup>19</sup>

<sup>17</sup> <https://djr.com/gimlet-xray/>

<sup>18</sup> <https://www.prototipo.io/about/>

<sup>19</sup> <https://www.kickstarter.com/projects/prototipoapp/prototipo-streamlining-font-creation>

## 6. Sonuç

Bu çalışma görsel iletişim tasarımında kullanılan kodlama çeşitlerini ve bunların yaratıcı biçimde kullanıldığı tasarım örneklerini araştırmıştır. Buna göre, kodlama dillerinin çeşitli grafiklere görsel efektler kazandırmada; logo, yazıyüzü, animasyon ve web tasarımı gibi pek çok alanda kullanıldıkları görülmüştür. Çoklu ortam araç teknolojilerinin yaygınlaşmasına bağlı olarak da kodlamanın görsel iletişim tasarımında temel bir tasarım aracı olacağı öngörülmektedir. Görsel iletişim tasarımının temel uygulamalarının yanı sıra, sanal ve artırılmış gerçeklik sunumlarında, bilgilendirme grafiklerinde ve veri görselleştirmelerinin grafiksel sunumlarında görsel iletişim tasarımcısının sahip olduğu kodlama bilgisi onu yazılımcılara daha az bağımlı olmasını ve kodlama alanının olanaklarını daha iyi kullanmasını sağlayacaktır.

Çalışmada elde edilen bulgular, görsel iletişim tasarımcılarının kullandıkları programlama dillerinin yüksek seviyeli diller kategorisinde yer alan kodlama dilleri olduğunu göstermiştir. Bu dillerin yazımı ve okunması sistem programlama dillerine göre daha kolaydır ve CAD programlarının Script editörlerinde kullanılabilirler. AppleScript, VisualBasic, ActionScript, CommandScript veya MelScript gibi diller kodlama dillerine örnektir. Kodlama dilleri arasında JavaScript görsel iletişimciler tarafından en yaygın kullanılandır. Bunun ana nedeni Javascript'in yaygın olarak web tarayıcılarında kullanılıyor olması ve pek çok CAD programı tarafından tanınmasıdır. Script editörlerinde kullanılan kodlama dilleri dışında, doğrudan kendi yazılımına sahip olan kodlama dillerini de kullanmak mümkündür. Bunlar Python, Java dilindeki Processing ve C++ temelli openFrameworks gibi kodlama dilleridir. Sanatçı ve tasarımcılar tarafından tercih edilen bu dillerle statik görüntüler elde edilebildiği gibi hareketli ve etkileşimli video ve animasyonlar da tasarlanabilmektedir. Kodlamaya dayalı bir başka grafik üretim teknolojisi ise Hesaplamalı Grafik Tasarım araçlarıdır. Hesaplamalı tasarım araçları mimarlık, endüstriyel ve tekstil gibi tasarım alanlarında kullanıldığı bilinmektedir ve genellikle belirli bir görevi veya görevleri yerine getirmek üzere yazılırlar. Hesaplamalı grafik tasarım araçlarının yazımında disiplinlerarası bir ekip çalışması gerekir. Belirli bir disipline yönelik yazıldıkları için sadece yazılımcıların çalışması yeterli değildir. Bu nedenle mutlaka yazılımcılarla birlikte alandan bir uzmana ihtiyaç duyulur. Dolayısıyla bu tür disiplinlerarası çalışmalarda yer alacak olan görsel iletişim tasarımcılarının asgari düzeyde kodlama bilgisine sahip olması gerekir. Hemen hemen bütün görsellerin dijitalleştiği bir ortamda görsel iletişim tasarımcıları az ya da çok bilgisayar dilinin olanaklarını, yani sınırlar ve birlerle 'boyamayı' ve 'çizmeyi' öğrenmeleri kaçınılmaz olmuştur.

Bu çalışma görsel iletişim tasarımında kodlama eğitimi verilmesinin önemi ve bu eğitimin sağlayacağı avantajların araştırılması noktasında bir temel oluşturmaktadır. Yazılım alanının genişliği nedeniyle programlama öğrenimi yazılımcı olmayanlar için zorlu bir sürece dönüşmektedir. Bireysel çabalarla programlama öğrenen ve kodlama dillerini bir tasarım aracı olarak kullanan görsel iletişim tasarımcılarının sayısının arttığı gözlemlense de yurtdışındaki örneklerle kıyaslandığında yeterli değildir. Bunun temel nedenini ülkemizde verilen görsel iletişim tasarımı eğitimine bağlamak mümkündür. İlgili fakültelerdeki ders müfredatları incelendiğinde kodlama derslerinin yalnızca Burdur Mehmet Akif Ersoy Üniversitesi, Gedik Üniversitesi, Fırat Üniversitesi, İstanbul Medeniyet Üniversitesi, İzmir Ekonomi Üniversitesi, Sabancı Üniversitesi ve Yıldız Teknik Üniversitesi'nde verildiği görülmektedir. Dolayısıyla diğer üniversitelerin grafik tasarım ve görsel iletişim tasarımı eğitimi verilen akademik birimlerde yetişen öğrencilerin kodlama konusunda yeterli bir donanım ile yetişemedikleri sonucu ortaya çıkmaktadır. Oysa üniversiteler bilgi ve teknolojiye yeni yenilikleri takip ederek topluma yön veren ve toplumu geleceğe taşıyan kurumlar olmalıdır. Bu nedenlerden dolayı üniversiteler çağın gerekliliklerini karşılayabilmeleri için eğitim öğretim programlarını çağın gerekliliklerine göre yeniden güncellemek zorundadırlar. Yine de kodlama konusunda kendisini yetiştirmek isteyen tasarımcılar için hiçbir şey geç değildir. Kodlama dillerinin seviyeleri kullanıcı odaklı olarak her geçen gün artmaktadır. Genişleyen kütüphaneleri ve yeni eklenen etiketleri ile kodlama dilleri görsel iletişim tasarımcılarına yeni ve denenmemiş grafikler üretme imkânı sunmaktadır.

## Kaynakça

- Avcı Tuğal, S. (2018). *Dijital sanat*, Hayalperest Yayınevi, İstanbul.
- Cevahir, H., ve Özdemir, M. (2017). Programlama Öğretiminde Karşılaşılan Zorluklara Yönelik Öğretmen Görüşleri ve Çözüm Önerileri. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu* içinde, 24-26.
- Delahunty, D. (2013). *The evolution of visual identities from static identities to dynamic identities*, Doctoral dissertation, Master's Dissertation in Science in Interactive Digital Media, Trinity College of Dublin University, Dublin.
- Gelişken, U. (2011). *Herkes için Grafik ve Animasyon*, İstanbul: Kodlab Yayın Dağıtım.
- Gould, D. (2003). *Complete Maya Programming: An extensive guide to MEL and C++ API*. Elsevier.
- Goodman, D. (2007). *JavaScript bible*. John Wiley & Sons.
- Karabak, D., ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 21(2-3), 163-169.
- McWilliams, C. (2003). *Adobe Scripting: Your Visual Blueprint for Scripting in Photoshop and Illustrator*, Maran Graphics.
- Noble, J. (2009), *Programming interactivity: A designer's guide to processing*, Kaliforniya: Arduino, and OpenFrameworks, O'Reilly Media Inc.
- Olssen, T., ve O'Brien, P. (2008). *The ultimate css reference*. Avustralya: Sitepoint, Victoria.
- Ousterhout, J. K. (1998). Scripting: Higher level programming for the 21st century. *IEEE Computer Magazine*, 31(3), 23-30.
- Özel, A., Kaya, Ç., ve Eken, S. (2014). JavaScript güvenlik açıkları ve güncel çözüm önerileri. 7. Uluslararası Bilgi Güvenliği ve Kriptoloji Konferansı.
- Özsağlam, M. Y. (2009). *3ds MAX Maxscript*, Seçkin Yayıncılık, Ankara.
- Pekgöz, N. (2002), *Webmaster için JavaScript*, İstanbul: Pusula Yayıncılık.
- Reas, C., ve Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Mit Press, Massachusetts.
- Reas, C., ve Fry, B. (2010). *Getting started with processing*. Kaliforniya: O'Reilly Media Inc.
- Sanderson, H., ve Rosenthal, H. (2010). *Introducing AppleScript*. In *Learn AppleScript* (pp. 3-13). Apress.
- Sarıkavak, N. (2005). *Sayısal tipografi 1*, Ankara: Başkent Üniversitesi Yayınları.
- Vardouli, T. (2012). *Bilgisayarın bin yüzü: Bilgisayarın tasarımda insanlaştırılması (1965-1975)*. Dosya (2012), 24.

## İnternet Kaynakçası

- Adobe, (2002). *Photoshop 7.0 Scripting Guide*, Adobe Systems Incorporated. Kaliforniya, ABD, Erişim: <https://www.scribd.com/document/89813688/Adobe-Photoshop-7-Scripting-Guide>, 30.12.2019.
- Adobe, (2007). *Introduction to Scripting*, Adobe Systems Incorporated, Kaliforniya, ABD, Erişim: [https://www.adobe.com/content/dam/acom/en/devnet/illustrator/pdf/adobe\\_intro\\_to\\_scripting.pdf](https://www.adobe.com/content/dam/acom/en/devnet/illustrator/pdf/adobe_intro_to_scripting.pdf). Erişim: 30.12.2019.
- Efendioğlu, M. A. (2015). *HTML ve CSS Nedir?* Erişim: <http://www.kodyazan.com/MakaleDetay/1060/HTML--CSS-NEDIR>, 19.05.2020

- <http://blog.choppingblock.com/2009/03/generative-art-in-illustrator-scriptographer/>, Erişim: 19.05.2020
- <https://www.drawbot.com>, Erişim: 19.05.2020.
- <https://djr.com/gimlet-xray/>, Erişim: 18.05.2020.
- <https://www.prototype.io/about/>, Erişim: 18.05.2020.
- <https://www.kickstarter.com/projects/prototypeapp/prototype-streamlining-font-creation>, Erişim: 19.05.2020.
- [http://www.softgraf.com/cursodesign/Apostila\\_Macro\\_Programming\\_Guide\\_X7.pdf](http://www.softgraf.com/cursodesign/Apostila_Macro_Programming_Guide_X7.pdf), Erişim: 19.05.2020.
- <https://www.adobe.com/devnet/illustrator/scripting.html>, Erişim: 19.05.2020.
- <https://www.adobe.com/devnet/photoshop/scripting.html>, Erişim: 19.05.2020.
- <https://helpx.adobe.com/after-effects/using/expression-language-reference.html>, Erişim: 19.05.2020.