

## KINETIC TYPOGRAPHY - FIGURATION AND TECHNOLOGY

Nikolina Stanić Loknar , Diana Bratić , Ana Agić   
University of Zagreb, Faculty of Graphic Arts, Zagreb, Croatia

**Abstract:** *Kinetic typography - text in motion is an animation method of characters that has a video form instead of some "static" form such as picture, poster or book. The most important element for figuration of kinetic typography is the choice of font. Furthermore, one should think about the letter cut, the size and color of the characters, and the background color on which the animation takes place. It can be created in various ways, most often using software that applies a multitude of effects to the text or letter character, creating dynamic solutions. The effects vary from the simplest such as "fade-in" and "fade-out" (entering and exiting text in and out of the frame). Static characters can expand, narrow, move slowly or rapidly, grow and change in a variety of ways to very complex ones in which the author builds an entire story or promotional video by carefully combining software capabilities. However, each software has its limitations and for this reason the kinetic typography presented in this paper is programmed using codes. In a wide range of available programming languages due to the simple interface that does not require advanced programming concepts and gives exceptional results in the field of kinetic typography, Processing was chosen. The Processing programming language is intended for generating and modifying graphics and is based on the Java programming language. The most important difference between Processing and Java is that Processing offers a simple programming interface that does not require advanced levels of programming such as classes, objects, or animations. It also allows advanced users to use them. Processing uses a variety of typography rendering approaches such as raster and vector solutions and allows typography to be programmed and displayed on the Web independently of the user's Web browser and font database. Processing enables the use of visual elements in animation, including typographic ones, by introducing interaction to the user. The user is no longer a passive observer but actively participates in the performance of the application whose final appearance is not predefined but arises from the actions of each individual user.*

*For the purposes of this paper, individual letters were created in a font-making program. The letters made are of various written classifications and cuts, which with their variety contribute to the attractiveness of the animation. In the creating of motion typography in this paper, the programming language Processing was used. Written program codes that manipulate words, letters, or parts of characters to create interesting visual effects for the viewer that aim to hold the viewer's attention and convey the desired message or emotion. There are no strict rules and patterns when making kinetic typography. In kinetic typography, each author determines his own rules, method of production, and there are no same solutions.*

**Key words:** kinetic typography, Processing, coding, character

### 1. INTRODUCTION

One of the greatest inventions in human history is the letter or what it enables and that is the transmission of written words. Typography is what communication looks like. But it is almost impossible to look and read at the same time because they are different perceptions. There is beauty in the language and beauty in the way it is presented. It all started about two millennia ago. The rules of typography are centuries old, and although the technologies have changed, the goal has always remained the same: a beautiful setting in the service of a pleasant and fruitful reading experience (Felici, 2012).

With the advent of film and graphic animation, the possibility of matching text and motion emerged. Researchers at the Human Computer Interaction Institute and School of Design at Carnegie Mellon University have traced the first use of kinetic typography to the 1959 Alfred Hitchcock's 1959 film "North by Northwest." In the opening credits, type is used in a movable format. A year later, the effect was used again in "Psycho." "This work stemmed in part from a desire to have the opening credits set the stage for the film by establishing a mood, rather than simply conveying the information of the credits," researchers wrote (Lee, 2015).

Today, kinetic typography is considered as a trend in web design development. Kinetic typography implies letters in motion. It is an animation technique used to make letters shrink, expand, fly, move slowly, and change in a number of ways for the user. The effect can be simple and short with small modifications or quite complex and long lasting. The use of kinetic typography has exploded recently due to the greater use

of the technique in web design. Kinetic typography is gaining in popularity as a background effect on websites and in web-based videos. All this was made possible by a larger broadband network and increased speed of surfing the Internet and the web. Kinetic typography is used for several reasons because it can add emphasis to a particular content. It can help convey tone and feelings. This can be an affordable option for those on a limited budget. Given the popularity of kinetic typography, today there are various tools and software that allow the creation of moving text. Professional programs with a multitude of tools are available, but also free programs that offer a wide range of tools and templates for creating interesting solutions in the field of kinetic typography. Some software is easier to use, so it is easy for beginners to master (Windowsreport, 2020).

A group of authors from the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan introduced a web-based integrated design environment named "TextAlive Online" that supports creating Kinetic Typography videos synchronized with songs available online. Experienced users create Java codes themselves and create kinetic typography while those less skilled use the templates offered (Kato et al, 2015).

In this paper we have chosen Processing because it is an open source programming language, and its environment has been built for the electronic arts and visual design community. Processing files have a .pde extension at the end of every file. They can be easily exported as Java applets and uploaded to the Internet. Also the files can be opened in Adobe Illustrator and saved as an vector .eps file which can be printed in high resolution (Ahn and Cordova, 2008). Processing is real programming. It is not some pretend language to help you get started; it has all the fundamentals and core concepts that all languages have (Shiffman, 2008).

## 2. EXPERIMENTAL

In this paper, several examples of kinetic typography made in Processing are presented. The characters are designed to simulate handwriting and follow the typographic rules in Fontlab. Fonts are generated and installed in a computer so that Processing can use them in program codes that create kinetic typography. Processing displays fonts using the .vlw font format, which uses images for each letter, rather than defining them through vector data. The loadFont() function constructs a new font and textFont() makes a font active (Reas and Fry, 2007). The Processing development environment is a simplified environment for writing computer code. All examples use the abbreviation of the name of GRID symposium and are shown below.

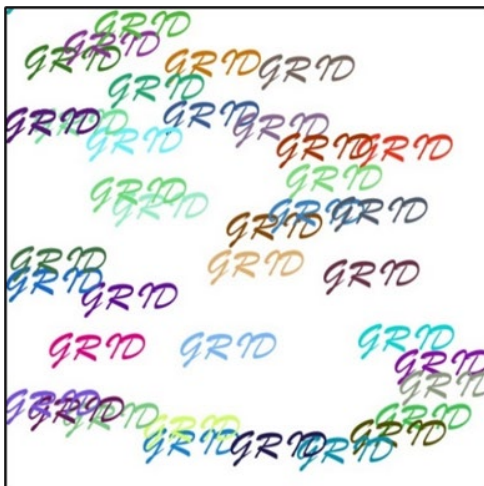


Figure 1: Sketch FrameRate 5



Figure 2: Sketch FrameRate 30

Example 1 shows the printing of "GRID" text in interaction with the user. Printing is repeated regardless of the movement of the mouse position on the screen. The *frameRate* command specifies the number of frames displayed per second. The default rate is 60 frames per second. In a figure 1, frameRate is set to 5, so that the text is clearly visible and legible. Figure 2 shows a *frameRate* 30, which is too high for this example because it increased the number of views in a few seconds, which significantly complicates readability and destroys the beauty of the display. The background color is white, while the text changes color with each imprint. The color range is whole RGB spectrum, i.e. a random range of valuable R, G, and

B components ranging from 0 to 255 which is maximum value range in Processing. A very short program code shows an interesting solution that attracts the user's attention and keeps just in order to interactions achieved by the *mouseX* and *mouseY* commands. X and y are the values of the mouse position in the coordinate system on the screen. By moving the mouse, the user changes the values of x and y, and in these places a imprint is made in a different color each time.

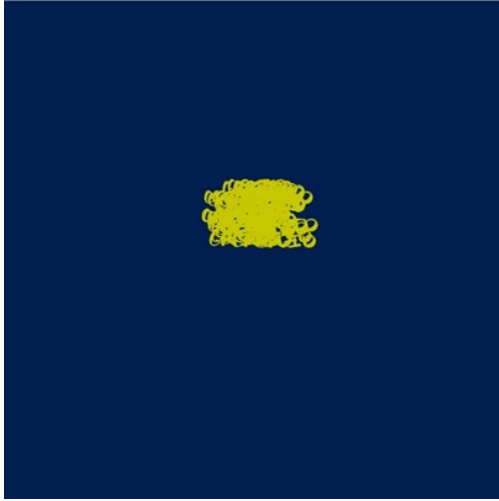


Figure 3: Sketch after click

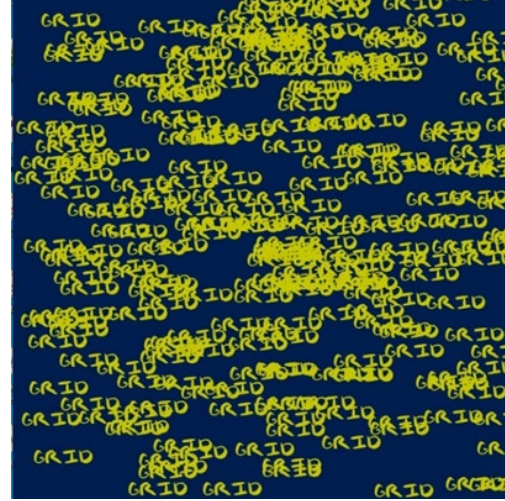


Figure 4: Sketch after four different click

Second example shows a “GRID” text printed in yellow on a blue background. A handwritten font from own font database is used. The program works in way that each mouse clicks at the position it is in, generates 50 imprints of “GRID” text which are further scattered over the entire given area on the screen. The speed and direction of each imprint is different. The difference in the directions of movement is achieved by introducing pseudo-random numbers that within the given end values determine the speed and direction of movement on the screen from the initial position of the click towards the edges. The random numbers we get from the *random()* function are not truly random; therefore, they are known as “pseudo-random.” They are the result of a mathematical function that simulates randomness. This function would yield a pattern over time, but that time period is so long that for us, it is just as good as pure randomness (Shiffman, 2012). Figure 3 shows the moment immediately after the first mouse click where the “explosion” of imprinting begins from that position. Figure 4 shows the moment when 4 mouse clicks were made in different positions so the screen shows 200 “GRID” inscriptions scattered in all directions and moving further towards the edges until they come out of the screen and in the end only the blue background remains. This program works in interaction with the user in such a way that no matter how many times the user clicks the mouse, a new explosion of text printing is created each time. In a few seconds, all the text comes out of the sketch and only the dark blue background remains visible. This code is a bit more demanding than example 1 because the action being triggered by a mouse click, but the speed and direction of movement of the letters on the screen should be programmatically determined using the conditions in the loop. In Processing *setup()* is executed once when the sketch starts and *draw()* loops forever and ever until the user quit.

Part of the animation of the third example of kinetic typography in Processing is shown in Figures 5 and 6. Opening the sketch starts a program that has a *frameRate* set to 5, which means that it displays 5 imprints of the text “GRID” in one second. The X and Y values that determine the position of each imprint in Sketch, are in the range from 0 to 300, and generates randomly, which is clearly visible in Figure 5.

Also, with each display of text, R G B color values are generated pseudo-randomly, which is especially emphasized in Figure 6. That’s way each print has its own shade. The code execution lasts until the user click the mouse. The code defines that the *mousePressed* command stops the performs of the program, i.e. only the black background color is visible on the screen. Immediately after the click, the program starts running again and very fast we have a full sketch of the text "GRID" in various colors. Regarding a parameters defined pseudorandomly, each image is actually unique. And in this example, a handwritten font from its own database of monoline strokes was used. And in this example, there is an interaction with the user where he participates in image creation.





Figure 5: Sketch after start

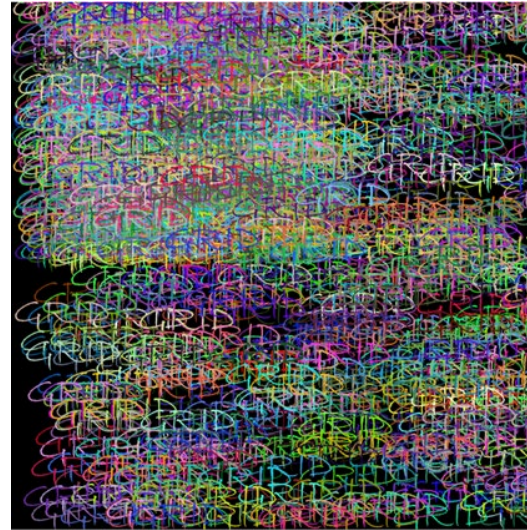


Figure 6: Sketch after few seconds

The fourth example shows a combination of computer graphics and typography in Processing. Using a *for* loop and Bezier curves, a rosette is programmed, and each "leaf" has its own color. In the same *for* loop there are also individual letter characters that form the text "GRID". In previous examples, the program understood the "GRID" text as a whole and the color or display position change, applied to all letters at once. Here each letter character has its own position, its own angle of rotation, and its own color. In this example, the dynamism is achieved by changing the color on the rosette leaves so that the impression of rotation is created which is visible in Figures 7 and 8. The letters GRID change color quickly, so that our eyes seem to flicker which further emphasizes the dynamics and movement which really no exist. The generation of letters and rosette curves occurs at the same position.

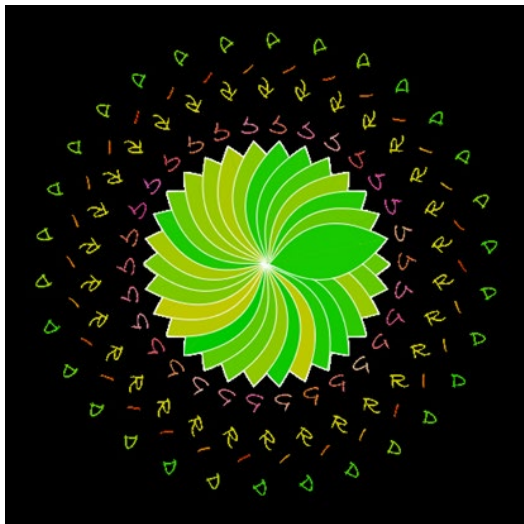


Figure 7: After start

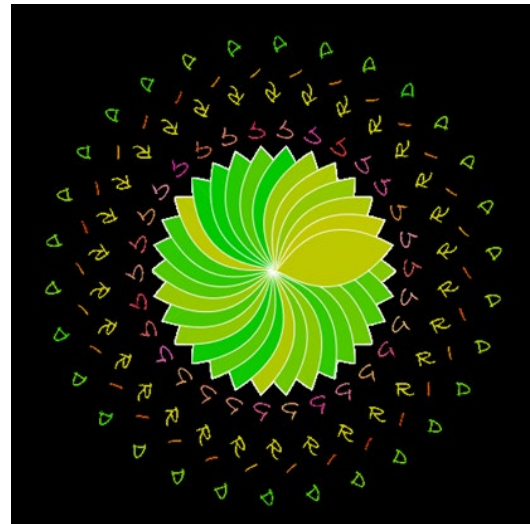


Figure 8: After few seconds

### 3. DISCUSSION

Kinetic typography programming offers unlimited possibilities of expression depending on context. In today's digital world, it is not enough to have a static display, but it is dynamism that captures the viewer's attention. In this paper, the dynamics are presented in several ways within the sketch according to the set limit values of program loops. The program affects the position of the text, color change, rotation angle, number of repetitions, speed of movement, etc. Using the Processing programming language, dynamism gets another important segment, and that is interactivity. As shown in the examples, the user becomes an

active participant who, through his actions, generates a display in a certain way, which certainly keeps him on that website longer, which is the goal of all advertisers.

#### 4. CONCLUSIONS

Processing is a programming language that offers various interactive solutions either in computer graphics or typography. Kinetic typography, i.e. text in motion, attracts the viewer's attention and keeps it much longer than a static image. This paper shows through several examples the possibilities of programming typography in motion. The use of the Processing programming language enables interactivity with the user in such a way that the user, by changing the position or clicking the mouse, on his own determines on which parts of the screen the mobile typography will be created. Likewise, the user controls the text display density which depends on the speed of the mouse movement. The introduction of random numbers increases the attractiveness of the obtained moving graphics because each mouse clicks made by the user generates different movements of letter elements on the screen. The described examples show only a small part of the possibilities provided by Processing in mobile typography programming. Such solutions contribute to the attractiveness of the content, also more impressively convey the message to consumers if they are used for marketing purposes. Mobile typography also finds its application in augmented reality with, for example, exhibits in museums, education, and entertainment.

Processing is relatively easy to learn, so even beginners with knowledge of another programming language can easily manage and with a little effort to program interesting interactive solutions either in computer graphics or typography.

#### 5. REFERENCES

- [1] Ahn, Y., Cordova, V.: "Type+Code Processing for Designers", (Center for Design Thinking - Maryland Institute College of Art, Maryland, 2008.), page 26.
- [2] Felici, J.: "The Complete Manual of Typography, A guide to setting perfect type", (Peachpit , Berkley CA, 2012.), page 11.
- [3] Kato, J., Nakano, T., Goto, M.: "TextAlive: Integrated Design Environment for Kinetic Typography" Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems 2015. (Association for Computing Machinery: New York, New York, 2015), pages 3403-3412.
- [4] Kato, J., Nakano, T., Goto, M.: "TextAlive Online: Live Programming of Kinetic Typography Videos with Online Music", Proceedings of the First International Conference on Live Coding 2015, (AIST: Tsukuba, Japan, 2015), pages 199-205.
- [5] Lee, J. C., Forlizzi, J., Hudson, S. E.: "The Kinetic Typography Engine: An Extensible System for Animating Expressive Text", URL: [https://www.cs.cmu.edu/~johnny/kt/dist/files/Kinetic\\_Typography.pdf](https://www.cs.cmu.edu/~johnny/kt/dist/files/Kinetic_Typography.pdf) (last request: 2020-10-08).
- [6] Reas, C., Fry, B.: "Processing: a Programming Handbook for Visual Designers and Artists", (The MIT Press Cambridge, Massachusetts, 2007.), pages 112, 702.
- [7] Shiffman, D.: "Learning Processing A Beginner's Guide to Programming Images, Animation, and Interaction" (Elsevier, Burlington MA, 2008.), page 17.
- [8] Shiffman, D.: "The Nature of Code: Simulating Natural Systems with Processing", (The Nature of Code, Boston MA, 2012.), page 7.
- [9] Stanojevic, M.: "Best kinetic typography software", URL: <https://windowsreport.com/kinetic-typography-software/> (last request: 2020-10-10).



© 2020 Authors. Published by the University of Novi Sad, Faculty of Technical Sciences, Department of Graphic Engineering and Design. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license 3.0 Serbia (<http://creativecommons.org/licenses/by/3.0/rs/>).