



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"ConstructDe: Μια εξερεύνηση των ορίων της αναγνωσιμότητας και του creative coding στην τυπογραφία, εμπνευσμένη από τη Μεταμόρφωση του Φ. Κάφκα."

Συγγραφέας

ΤΣΟΚΟΥ ΜΑΡΙΑ-ΝΕΦΕΛΗ

A.M

20674191

Επιβλέπουσα

ΜΑΣΟΥΡΑ ΕΥΑ

Αθήνα, Σεπτέμβριος 2024



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

DIPLOMA THESIS

**"ConstructDe: An exploration of the boundaries of legibility
and creative coding in typography, inspired by Kafka's
Metamorphosis."**

Student

COKU MARIA-NEFELI

Registration number

20674191

Supervisor

MASOURA EVA

Athens, September 2024



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ UNIVERSITY OF WEST ATTICA

**Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή
Η πτυχιακή/ διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι
Εξεταστική Επιτροπή**

**ΕΠΙΒΛΕΠΟΥΣΑ
Μασούρα Εύα
Ακαδημαϊκός Υπότροφος**

**ΒΑΘΜΟΛΟΓΗΤΡΙΑ
Παπαδομανωλάκη Μαρία
Λέκτορας Εφαρμογών**

**ΒΑΘΜΟΛΟΓΗΤΗΣ
Ματθιόπουλος Γιώργος
Επίκουρος Καθηγητής**

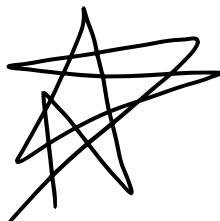
ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ / ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη ΤΣΟΚΟΥ ΜΑΡΙΑ-ΝΕΦΕΛΗ του ΕΡΒΙΝ, με αριθμό μητρώου 20674191 φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Εφαρμοσμένων Τεχνών και Πολιτισμού, του Τμήματος της Γραφιστικής και Οπτικής Επικοινωνίας με κατεύθυνση Γραφιστική, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Η Δηλούσα
Τσόκου Μαρία-Νεφέλη



C o n s t r u c t D e
An exam of te loundis of lilly
and ceate coaly in topography insied by Kef
ke's Mamopus. This post by Nely Tok-
ou (22) Unisty of West Ala, Arns, Gae
C o n s t r u c t D e
An exam of te loundis of lilly
and ceate coaly in topography insied by Kef
ke's Mamopus. This post by Nely Tok-
ou (22) Unisty of West Ala, Arns, Gae
C o n s t r u c t D e
An exam of te loundis of lilly
and ceate coaly in topography insied by Kef
ke's Mamopus. This post by Nely Tok-
ou (22) Unisty of West Ala, Arns, Gae

C o n s t r u c t i o n

An edition of the
and a complete commentary in two
volumes. Metamorphoses. The
of (1924) University of Wisconsin
C o n s t r u c t i o n

An edition of the
and a complete commentary in two
volumes. Metamorphoses. The
of (1924) University of Wisconsin
C o n s t r u c t i o n

An edition of the
and a complete commentary in two
volumes. Metamorphoses. The
of (1924) University of Wisconsin
C o n s t r u c t i o n

u c t D e
e bounces of edge
ography inspired by Ka
is project by New York
Est Aba, Ares, Gece
r u c t D e
e bounces of edge
specially Kefel Manopoulos
ives of West Aba, Ares, Gece
r u c t D e
e bounces of edge

ConstructDe: An exploration of the boundaries of legibility and creative coding in typography, inspired by Kafka's "Metamorphosis". Thesis project by Nelly Tsokou (2024), University of West Attica, Athens, Greece

Research & Design: Nelly Tsokou

Title:

ConstructDe: An exploration of the boundaries of legibility and creative coding in typography, inspired by Kafka's "Metamorphosis"

Cover:

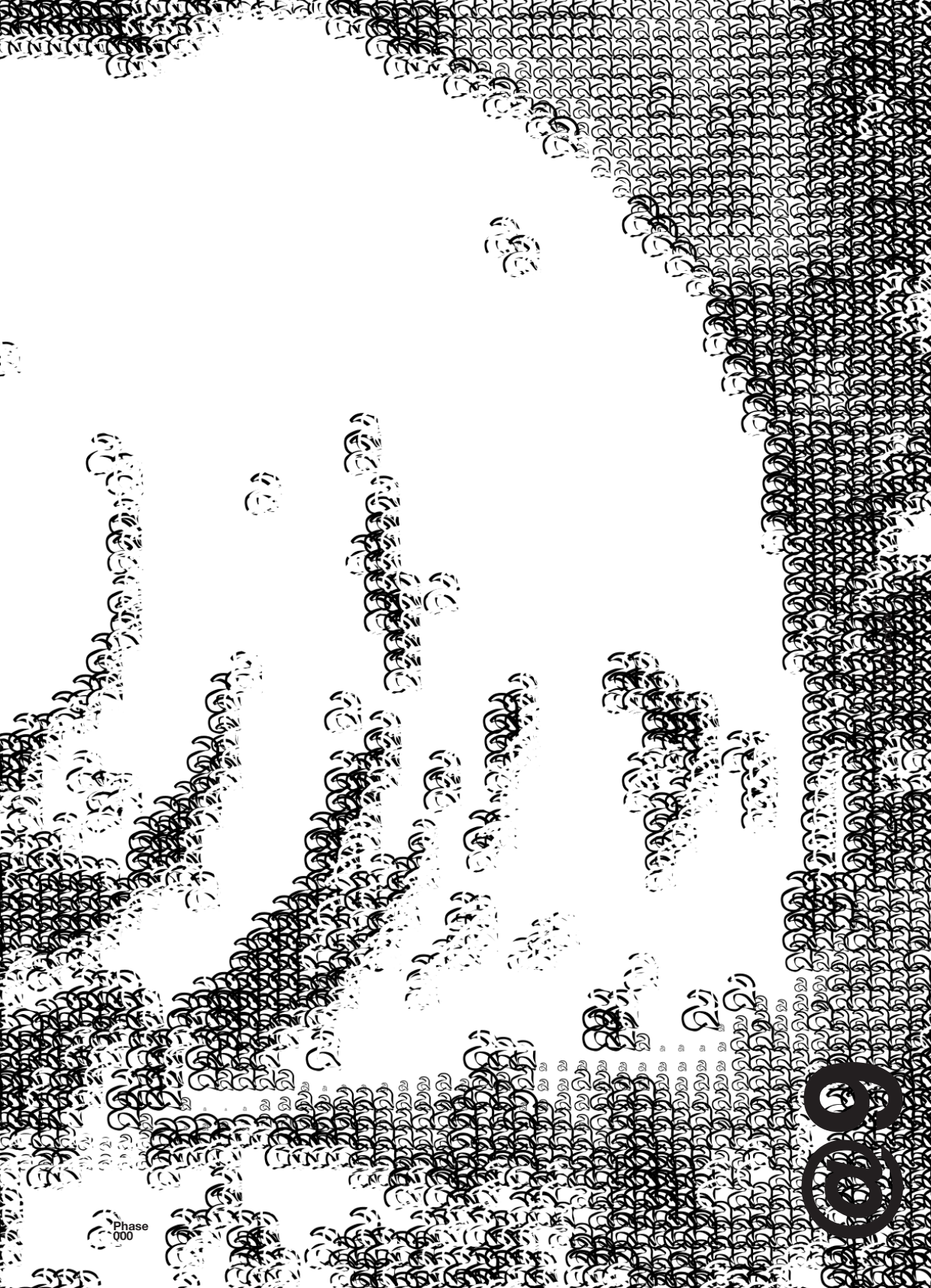
Letterpressed book title

(September 2024)

nellytsokou.xyz



8



Phase
000



Contents

Phase 1: Introduction - Metamorphosis	@14 - @19
Phase 2: Research	@20 - @43
Phase 2.1: ConstructDe	@22 - @25
Phase 2.2: New Wave	@26 - @29
Phase 2.3: Post or Anti?	@30 - @35
Phase 2.4: Type is not dead	@36 - @39
Phase 2.5: Exploring (il)legibility	@40 - @43
Phase 3: Experimental fonts x Creative coding	@44 - @79
Phase 3.1: Progress Sans	@46 - @61
Phase 3.2: Progress Gen	@62 - @75
Phase 3.3: Computer art	@76 - @79
Phase 4: Custom code experiments	@80 - @105
Phase 5: Typographic rendering & interpretation of Kafka's Metamorphosis	@106 - @123
Phase 5.1: Posters	@108 - @117
Phase 5.2: Flipbooks	@118 - @123
Phase 6: Bibliography & Sources	@123 - @125
Phase 7: Colophon	@127

@11

Introduction

ConstructiDe

@12



@13

Constricpe

Metamorphosis

Ο Φράνζ Κάφκα υπήρξε ένας από τους πιο σημαντικούς συγγραφείς του 20ου αιώνα. Τα έργα του χαρακτηρίζονται από μια μίξη φιλοσοφιών όπως ο υπαρξισμός και ο ρεαλισμός. Τα έργα του ξεχειλίζουν από παραλογοισμό και συμβολισμούς, τους οποίους πλαισιώνει σε κείμενα που εξιστορούν την ανθρώπινη εμπειρία με έναν σκοτεινό τρόπο.

Ο Κάφκα θεωρείται πλέον “ο πατέρας του υπαρξισμού”, μιας φιλοσοφίας που βασίζεται στο γεγονός ότι οι άνθρωποι υπάρχουν σε έναν κόσμο που κυριαρχεί το χάος, έναν κόσμο που είναι αβάσιμος και παράλογος, και πως οι άνθρωποι πρέπει να αποδώσουν το δικό τους νόημα στη ζωή τους.

Οι ιστορίες του Κάφκα έχουν αυτοβιογραφικό ύφος, κάτι που δυσκολεύει στην εμβάθυνση και την πλήρη κατανόηση τους. Το ύφος αυτό, δείχνει τη συνεχή δυσφορία και ταραχή του συγγραφέα για τη ζωή του ή την σκεπτικιστική φύση του συγγραφέα.

Σε ένα από τα πιο σπουδαία και σύντομα έργα του, την Μεταμόρφωση, αφηγείται την ιστορία του Γκρέγκορ, ενός πωλητή, που ξυπνάει μια μέρα και αντιλαμβάνεται ότι έχει μεταμορφωθεί σε ένα μεγάλο αηδιαστικό έντομο. Οι πωλήσεις, μια δουλειά που αναγκάζεται να κάνει ο πρωταγωνιστής για να ξεχρεώσει τα χρέη των γονιών του, δεν τον κάνει χαρούμενο, παρ’ όλα αυτά, δουλεύει σκληρά καθώς όλη του η οικογένεια στηρίζεται οικονομικά από εκείνον. Όταν λοιπόν γίνεται αυτό το παράλογο συμβάν και ο Γκρέγκορ ξυπνάει ως έντομο, η πρώτη του αντίδραση δεν ήταν να αναρωτηθεί πως είναι δυνατόν να συνέβη κάτι τέτοιο, αλλά αναζητούσε τρόπο να πάει στη δουλειά του. Αυτό μπορεί να εξηγηθεί από το γεγονός ότι ο Γκρέγκορ ένιωθε πολύ άσχημα με τον εαυτό και ταυτιζόταν με τα χαρακτηριστικά ενός ζωοφίου, ακόμα και πριν την μεταμόρφωση του, πράγμα που δεν του έκανε εντύπωση όταν έγινε πραγματικότητα. Επιπλέον, ένιωθε πως η δουλειά του είναι πιο σημαντική από τον ίδιο, και πάντα την είχε ως προτεραιότητα, πάνω από τον δικό του ελεύθερο χρόνο και τις δικές του ασχολίες. Σύμφωνα με αναλύσεις που έχουν γίνει για το έργο, παρά την ολική μεταμόρφωση του ήρωα σε έντομο, ο χαρακτήρας του Γκρέγκορ δεν αλλάζει πολύ κατά την διάρκεια του έργου.¹ Συγκεκριμένα, και στις δύο μορφές του, ο χαρακτήρας φαίνεται να αποδέχεται τις καταστάσεις και να μην αντιδράει. Όταν ο πατέρας του μένει άνεργος, αποδέχεται το γεγονός ότι πρέπει να ξεκινήσει να δουλεύει σε μια δουλειά που δεν του αρέσει, για να προσφέρει στην οικογένεια του. Η ουδετερότητα, η απάθεια και η χαμηλή αυτοεκτίμηση είναι τα στοιχεία που χαρακτηρίζουν την προσωπικότητα του Γκρέγκορ.

Ανήμπορος πλέον να δουλέψει, ο Γκρέγκορ σταδιακά αρχίζει να χάνει την επαφή με την ανθρώπινη φύση του. Η οικογένεια του ξεκινάει να χάνει τον σεβασμό και την υπομονή της προς αυτόν, εφόσον πλέον αδυνατεί να τους παρέχει τα προς το ζην, και γίνεται “βάρος” για αυτούς. Η συμπεριφορά τους προς τον Γκρέγκορ αλλάζει: τον αντιμετωπίζουν σαν ένα ον που το τρέφουν, του φέρονται με απέχθεια, σαν να

Franz Kafka was one of the most important writers of the 20th century. His works are characterized by a mixture of philosophies such as existentialism and realism. His works overflow with absurdity and symbolism, which he frames in texts that chronicle the human experience in a dark way.

Kafka is now considered "the father of existentialism", a philosophy based on the fact that humans exist in a world dominated by chaos, a world that is unfounded and irrational, and that humans must ascribe their own meaning to their life.

Kafka's stories have an autobiographical style, which makes it difficult to deepen and fully understand them. This style shows the writer's constant discomfort and agitation about his life or the writer's skeptical nature.

In one of his most important and short works, *The Metamorphosis*, he tells the story of Gregor, a salesman, who wakes up one day and realizes that he has been transformed into a large disgusting insect. Sales, a job the protagonist is forced to do to pay off his parents' debts, does not make him happy, nevertheless, he works hard as his whole family is financially dependent on him. So when this absurd event takes place and Gregor wakes up as an insect, his first reaction is not to wonder how such a thing could happen, but he looks for a way to get to work. This can be explained by the fact that Gregor felt very bad about himself and identified with the characteristics of a bug, even before his transformation, that it did not impress him when it physically happened. Furthermore, he felt that his job was more important than himself, and always prioritized it over his own free time and his own pursuits. According to analysis that has been done for Kafka's work, despite the total transformation of the hero into an insect, Gregor's character does not change much during the work.¹ Specifically, in both of his forms, the character seems to accept the situation he is in and does not react. When his father becomes unemployed, he accepts the fact that he has to start working a job he doesn't like in order to provide for his family. Neutrality, apathy and low self-esteem are the elements that characterize Gregor's personality.

Unable to work anymore, Gregor gradually begins to lose touch with his human nature. His family begins to lose respect and patience for him, since he is no longer able to provide for them, and he becomes a "burden" to them. Their attitude towards Gregor changes: they treat him like a creature to feed and treat him with disgust, as if he were really an insect. With all these developments, Gregor is alienated from his human self and feels worthless now that he cannot help his family financially. Which forces the elderly father and his young sister to work. Later, due to a misunderstanding, Gregor's father hits him with an



¹ *The Metamorphosis Study Guide*. (n.d.). Sparknotes. Retrieved July 26, 2024, from <https://www.sparknotes.com/lit/metamorph/character/gregor-samsa/>

ήταν πραγματικά ένα έντομο. Με όλες αυτές τις εξελίξεις, ο Γκρέγκορ αποξενώνεται από τον ανθρώπινο εαυτό του και νιώθει άχρηστος τώρα που δεν μπορεί να βοηθήσει οικονομικά την οικογένεια του. Κάτι που αναγκάζει τον μεγάλο σε ηλικία πατέρα και την μικρή του αδερφή να δουλέψουν. Αργότερα, λόγω μιας παρεξήγησης, ο πατέρας του Γκρέγκορ τον χτυπάει με ένα μήλο και τον τραυματίζει θανάσιμα, με αποτέλεσμα η οικογένεια να νιώθει απελευθερωμένη και κάνει μια νέα αρχή χωρίς τον γιο τους.

Σε αυτή τη βλοσυρή και ζοφερή ιστορία, ο Κάφκα έχει χρησιμοποιήσει πολλούς συμβολισμούς που έχουν να κάνουν με το ρόλο του χρήματος στην ζωή του ανθρώπου, ασκώντας κριτική στο καπιταλιστικό σύστημα, που λειτουργεί με τέτοιο τρόπο, που αποξενώνει και δημιουργεί ενοχές στα άτομα που αδυνατούν να παράξουν για το κεφάλαιο. Τους αποξενώνει τόσο πολύ, που καταλήγουν να σκέφτονται τον εαυτό τους ως ένα άχρηστο, ακόμβολο και μοχθηρό ον, όπως ακριβώς ένα ζωύφιο.

Η αποξένωση είναι μια αρνητική, απαισιόδοξη συμπεριφορά σε ένα κοινωνικό περιβάλλον. Ένας παράγοντας που μπορεί να οδηγήσει στην αλλοτρίωση είναι η σχέση μεταξύ της κοινωνικής δομής και του χαρακτήρα του κάθε ανθρώπου, λόγω αντιθέσεων που υπάρχουν μεταξύ του κοινωνικού του ρόλου και της αυτοεικόνας και των αξιών του. Όταν ο κοινωνικός ρόλος έρχεται σε σύγκρουση με τις αξίες και τις επιθυμίες του ατόμου αυτού, το απομονώνει από το κοινωνικό σύνολο και του προκαλεί αισθήματα δυσφορίας και δυστυχίας. Η αποξένωση τροφοδοτείται από την μη ίση αντιμετώπιση, την καταπίεση, την εκμετάλλευση και την έλλειψη σεβασμού.² Στο έργο του Κάφκα, η αποξένωση του ήρωα είναι πολύ εμφανής και πριν την μεταμόρφωση του, καθώς η οικογένεια του τον εκμεταλλεύεται για να τους στηρίζει οικονομικά, αδιαφορώντας πλήρως για τις προσωπικές του επιθυμίες και ασχολίες, αναγκάζοντας τον να δουλεύει αντί για εκείνους, σε μια δουλειά που δεν τον πληροί. Η αντιμετώπιση τους γίνεται χειρότερη μετά την μεταμόρφωση του σε έντομο, και έτσι ο Γκρέγκορ δεν καταφέρνει ποτέ να ζήσει την ζωή που πάντα ήθελε. Έτσι, ο Γκρέγκορ στην Μεταμόρφωση, σταδιακά αποξενώνεται από την οικογένεια, τη δουλειά, τη κοινότητα και τα συναισθήματα του. Αυτή η αλλοτρίωση ευθύνεται στην έλλειψη ψυχολογικών δεσμών με την οικογένεια του και με τον περίγυρο του, εξαιτίας της δουλειάς του και της ψυχικής οδύνης που ήταν αναγκασμένος να υποστεί για να μπορέσει να βγάλει χρήματα.³

Με αυτόν τον τρόπο η μεταμόρφωση του Γκρέγκορ σε έντομο, λειτουργεί ως μεταφορά για την κοινωνικοοικονομική θέση της μεσαίας τάξης, και πως ο άνθρωπος έχει να επιλέξει ανάμεσα στον εαυτό του και την κοινωνία, καθώς εάν διαλέξει τον εαυτό του, θα χάσει την κοινωνική του αξία, και αν διαλέξει την κοινωνία, τότε κάνει την συνειδητή επιλογή να χάσει τον εαυτό του και την ατομικότητά του. Η Μεταμόρφωση εξερευνά αυτό το δίλημμα, χωρίς να δίνει ποτέ απάντηση ή λύση σε αυτό. Το φαινόμενο της αποξένωσης έχει μελετηθεί από πολλούς ψυχολόγους και

apple and fatally injures him, leaving the family feeling liberated and make a new start without their son.

In this bleak story, Kafka has used many symbols related to the role of money in the human life, criticizing the capitalist system, which operates in such a way that alienates and guilts people who are unable to generate capital. It alienates them so much that they end up thinking of themselves as a useless, malicious and vicious being, just like a vermin.

Alienation is a negative, pessimistic behavior in a social environment. A factor that can lead to alienation is the relationship between the social structure and the character of each person, due to contrasts that exist between their social role and their self-image and values. When the social role comes into conflict with the values and desires of a person, it isolates them from society and causes them feelings of discomfort and unhappiness. Alienation is fueled by unequal treatment, oppression, exploitation and lack of respect.² In Kafka's work, the alienation of the hero is very evident even before his transformation, as his family exploits him to support them financially, completely disregarding his personal desires and pursuits, forcing him to work instead of them, in a job that doesn't fulfill him. Their lack of respect towards Gregor gets worse after he transforms into an insect, and so Gregor never gets to live the life he always wanted. Thus, Gregor in *The Metamorphosis* gradually becomes estranged from his family, work, community and feelings. This alienation is responsible for the lack of psychological bonds with his family and with his surroundings, because of his work and the mental suffering he was forced to undergo, in order to be able to make money.³

In this way, Gregor's transformation into an insect acts as a metaphor for the socioeconomic position of the middle class, and how man has to choose between himself and society, as if he chooses himself, he will lose his social value, and if he chooses society, then he makes the conscious choice to lose himself and his individuality. *Metamorphosis* explores this dilemma, without ever giving an answer or a solution to it. The phenomenon of alienation has been studied by many psychologists and psychoanalysts, such as Lacan, who believed that alienation is inevitable for the human species and often found connections between alienation and linguistics.⁴

Jacques Lacan (1901-1981) was a French psychoanalyst who connected various theories and philosophies with poststructuralism, applying concepts derived from the connection of anthropology and linguistics, that is, the subject of the nature and creation of human existence in relation to the social environment he belongs to, and had a significant influence on the

² Leopold D., «Alienation», *The Stanford Encyclopedia of Philosophy*, eds. Edward N. Zalta & Uri Nodelman, Winter 2022 Edition, retrieved July 26, 2024, from <https://plato.stanford.edu/archives/win2022/entries/alienation/>

³ Asriningtyas Devicha και Ali Mustofa, «Gregor Samsa's Self Alienation in Franz Kafka's *Metamorphosis*: Lacanian Psychoanalysis», *Humanitatis: Journal of Language and Literature*, τμ. 8 (30 Ιούλιος 2022), p. 261-276.

⁴ J. Lacan, *Écrits*, Paris, Éditions du Seuil, 1966, p. 288.

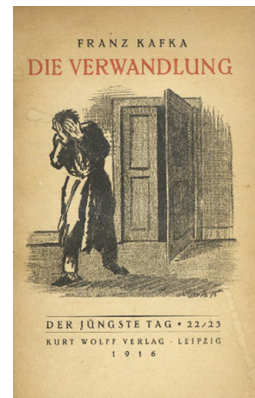


ψυχαναλυτές, όπως τον Λακάν, ο οποίος πίστευε ότι η αποξένωση είναι αναπόφευκτη για το ανθρώπινο είδος και συχνά έβρισκε συνδέσεις μεταξύ της αλλοτρίωσης με την γλωσσολογία.⁴

Ο Ζακ Λακάν (1901-1981) ήταν Γάλλος ψυχαναλυτής που σύνδεσε διάφορες θεωρίες και φιλοσοφίες με τον μεταστρουκτουραλισμό, εφαρμόζοντας έννοιες που προέρχονται από την σύνδεση της ανθρωπολογίας και της γλωσσολογίας, δηλαδή το θέμα της φύσης και της δημιουργίας της ανθρώπινης ύπαρξης σε σχέση με το κοινωνικό περιβάλλον που ανήκει, και είχε σημαντική επιρροή στην εξέλιξη της κοινωνιολογίας και της ψυχανάλυσης. Οι ιδέες του είχαν τεράστιο αντίκτυπο στις θεωρίες που εξελίχθηκαν τον 20ο αιώνα και είχαν μεγάλη επιρροή στη μεταστρουκτουραλιστική φιλοσοφία και στην ανάπτυξη του μεταμοντερνισμού. Εξερεύνησε τα συμβατικά όρια μεταξύ του λογικού και του παράλογου, υποστηρίζοντας ότι το ασυνείδητο δεν είναι τίποτα πρωτόγονο, αλλά είναι εξίσου περίπλοκο στη δομή του με το συνειδητό, δηλαδή, ότι το ασυνείδητο είναι μια γλώσσα που επιτρέπει τον διάλογο μεταξύ του ασυνείδητου και του συνειδητού κομματιού του εαυτού και διασφαλίζει ότι το ασυνείδητο παίζει ρόλο στην εμπειρία της ανθρώπινης ύπαρξης. Αυτό που συνδέει τον Λακάν με την αποδόμηση δηλαδή, και έπειτα με τον μεταμοντερνισμό, είναι η κριτική της γλώσσας, μέσω της σημασιολογίας.⁵

Ο Ζακ Ντεριντά (1930 - 2004) ήταν Γάλλος φιλόσοφος και ο “πατέρας” της αποδόμησης, της κριτικής που ασκήθηκε σε θεμελιώδεις έννοιες που αφορούσαν την φιλοσοφία, την λογοτεχνία, αλλά και την πολιτική. Η αποδόμηση, άρρηκτα συνδεδεμένη με τον Ντεριντά, αποτελεί μια ριζοσπαστική κριτική προσέγγιση απέναντι στη γλώσσα, και ευρύτερα, στον τρόπο που ερμηνεύουμε τον κόσμο γύρω μας. Αντίθετα από τον στρουκτουραλισμό (και τον μοντερνισμό), που πίστευε σε μια σταθερή και ξεκάθαρη σχέση μεταξύ γλώσσας και νοήματος, η αποδόμηση πειραματίζεται με την ασάφεια, την πολυσημία και τις αντιφάσεις που υπάρχουν στη γλώσσα και αμφισβητεί την ύπαρξη μίας «αληθινής» ερμηνείας, μίας «σωστής» ανάγνωσης ενός κειμένου. Σύμφωνα με τον Ντεριντά κάθε έννοια «δεν είναι ποτέ παρούσα αφ’ εαυτής [από μόνη της, en elle-même], σε μια επαρκή παρουσία που θα παρέπεμπε μόνο στον εαυτό της».⁶ Κάθε έννοια είναι «ουσιωδώς εγγεγραμμένη σε μια αλυσίδα ή σε ένα σύστημα εντός του οποίου παραπέμπει στην άλλη, σε άλλες έννοιες, διαμέσου του συστηματικού παιχνιδιού των διαφορών».⁷

Με το τέλος της Μεταμόρφωσης του Κάφκα, έφτασα στο συμπέρασμα ότι το έργο κάνει κριτική στο σύστημα του καπιταλισμού που είναι πλέον προέκταση της κοινωνίας, το οποίο αποξενώνει και απομονώνει τα άτομα που δεν είναι χρήσιμα για το σύστημα, δηλαδή δεν παράγουν χρήματα, όπως έγινε και με τον Γκρέγκορ, που έχασε κάθε ίχνος σεβασμού από την οικογένειά του όταν δεν ήταν πλέον ικανός να δουλέψει λόγω της μορφής του. Αυτό το συμπέρασμα μου γέννησε το εξής ερώτημα:



Franz Kafka, *Die Verwandlung*
(Eng. *Metamorphosis*), 1916

development of sociology and psychoanalysis. His ideas had a huge impact on the theories that developed in the 20th century and were highly influential in poststructuralist philosophy and the development of postmodernism. He explored the conventional boundaries between the rational and the irrational, arguing that the unconscious is nothing primitive, but is just as complex in its structure as the conscious, that is, that the unconscious is a language that allows dialogue between the unconscious and the conscious part of the self and ensures that the unconscious plays a role in the experience of human existence. What connects Lacan with deconstruction, and then with postmodernism, is the critique of language, through semantics.⁵

Jacques Derrida (1930 - 2004) was a French philosopher and the "father" of deconstruction, the criticism applied to fundamental concepts related to philosophy, literature, but also politics. Deconstruction, inextricably linked to Derrida, is a radical approach to language, and more broadly, to the way we interpret the world around us. Unlike structuralism (and modernism), which believed in a fixed and clear relationship between language and meaning, deconstruction experiments with the ambiguity, polysemy and contradictions that exist in language and questions the existence of a "true" interpretation, a "correct" way of reading a text. According to Derrida each concept "is never present by itself [by itself, en elle-même], in a sufficient presence that would refer only to itself."⁶ Each concept is "essentially inscribed in a chain or in a system within which it refers to the other, to other concepts, through the systematic play of differences".⁷

With the end of Kafka's *Metamorphosis*, I came to the conclusion that the work criticizes the system of capitalism, that is now an extension of society, which alienates and isolates people who are not useful to the system, i.e. do not generate money, as it was with Gregor, who lost all respect from his family when he was no longer able to work because of his new form. This conclusion raised the following question for me:

⁵ Myers Tony, «Lyotard and Lacan answering the question: what does Postmodernism want?», *Paragraph*, τμ. 24, τχ.1, (2001), p. 84-98.

⁶ Jacques Derrida, «Différance» (α' μέρος), μτφρ. Γεράσιμος Κακολύρης, *Ποίηση*, τχ. 27, Άνοιξη-Καλοκαίρι 2006, p. 72.

⁷ Jacques Derrida, «Différance» (α' μέρος), μτφρ. Γεράσιμος Κακολύρης, *Ποίηση*, τχ. 27, Άνοιξη-Καλοκαίρι 2006, p. 72-73.



@20

ConstructDe

Phase
1

Ο καπιταλισμός απομονώνει τα άτομα που δεν εξυπηρετούν τον σκοπό του και τα αντιμετωπίζει σαν έντομα: αχρηστά και αποκρουστικά. Όταν η τυπογραφία δεν εξυπηρετεί τον σκοπό της και δεν είναι αναγνωσίμη, ποια είναι η χρησιμότητά της;

Capitalism isolates people who do not serve its purpose and treats them like insects: useless and repulsive. When typography does not serve its purpose and is not legible, what is its purpose?

Phase
1

@21

ConstructDe

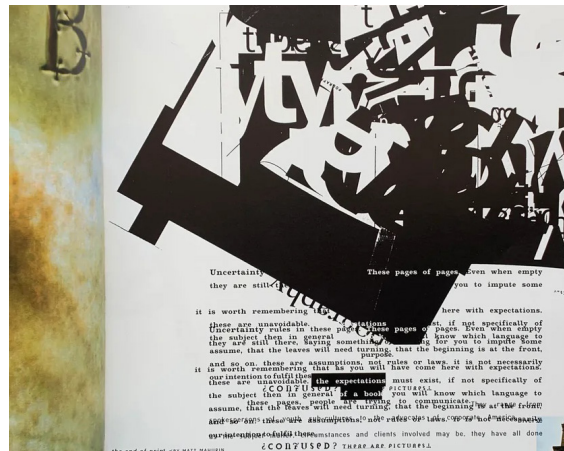
Η προσέγγιση του Ντεριντά στην επιστήμη της σημειολογίας, εξελίχθηκε σε έναν γενικότερο “πόλεμο” ενάντια στις παραδοσιακές μεθόδους μάθησης σε όλους τους τομείς. Σύμφωνα με τον Ντεριντά, η γλώσσα είναι μια δομή που πρέπει να αποδομηθεί, και υποστήριζε την κυριαρχία του γραπτού λόγου έναντι του προφορικού που υποστήριζαν οι στρουκτουραλιστές. Σύμφωνα με τη στρατηγική της αποδόμησης δεν υπάρχει «ένα αληθινό νόημα» για κάθε κείμενο. Χωρίς αυτό να σημαίνει, όπως υποστηρίζει ο Ντεριντά «ότι όλα τα αναφερόμενα αναστέλλονται, τα αρνούμαστε ή εγκλειούνται σε ένα βιβλίο, όπως πολλοί έχουν ισχυριστεί ή έχουν υπάρξει αρκετά αφέλεις να πιστεύουν ή με έχουν κατηγορήσει ότι πιστεύω. Αλλά ότι κάθε αναφερόμενο, όλη η πραγματικότητα έχει τη δομή ενός διαφορικού ίχνους, και ότι κάποιος δεν μπορεί να αναφέρεται σε αυτό το “πραγματικό” παρά μόνο εντός μιας ερμηνευτικής εμπειρίας».⁷ Η αποδόμηση είναι μια ανατρεπτική θεωρία που έχει επηρεάσει σημαντικά τη σύγχρονη σκέψη και τη δομή της γλώσσας, με στόχο την ανατροπή της λογικής σκέψης.⁸ Η αποδόμηση είναι μια μεταμοντέρνα θεωρία που “ξέφυγε” από τα όρια της σημειολογίας και εφαρμόστηκε σε πολλούς τομείς. Η αποδόμηση αναζητά δίπολα και τα συγκρίνει, δείχνοντας ότι η αρνητική έννοια μιας λέξης ή μιας ερμηνείας, πάντοτε υπάρχει και διαμένει μέσα στη θετική. Το “σκοτάδι” ζει μέσα στο “φως”, γιατί χωρίς το σκοτάδι δεν θα υπήρχε το φως.

Ο όρος “αποδόμηση” άρχισε να εμφανίζεται στον χώρο του design περίπου την δεκαετία του 1980 και εφαρμόστηκε σε τάσεις που αφορούσαν την αρχιτεκτονική, την βιομηχανική σχεδίαση, τη γραφιστική και το σχέδιο μόδας. Η αποδόμηση ήταν μια προσπάθεια κριτικής επανεξέτασης του συστήματος και των θεμών της κοινωνίας. Πολλοί καλλιτέχνες εμπνεύστηκαν από τις ιδέες του Ντεριντά, και χρησιμοποίησαν τη θεωρία της αποδόμησης αμφισβητώντας θεμελιώδεις έννοιες, στην προσπάθεια αναζήτησης νέων τρόπων έκφρασης.

Η αποδόμηση όντας μια θεωρία που ξεκίνησε από την σημειολογία και επηρέασε τον λόγο, και αναπόφευκτα και την επικοινωνία, χρησιμοποιήθηκε και στη γραφιστική και τον τυπογραφικό σχεδιασμό, καθώς οι αποδομιστές της εποχής είχαν εμμονή με το νόημα και την έλλειψή του. Αυτό ήταν κάτι που τους οδήγησε σε πειραματισμούς με τυπογραφικά στοιχεία, με τέτοιο τρόπο, ώστε το νόημα των συνθέσεών τους να είναι αμφιλεγόμενο και ήταν ρόλος του θεατή να δώσει το δικό του νόημα σε αυτές. Οι αποδομιστές δηλαδή αφαίρεσαν το νόημα από τις γραπτές λέξεις⁹ και τις έκαναν εικόνες.



David Carson, *The End of Print: the Graphic Design of David Carson*, 1st Edition (1995)



Derrida's approach to the science of semiotics evolved into a more general "war" against traditional learning methods in all fields. According to Derrida, language is a structure that must be deconstructed, and he argued for the dominance of the written language over the spoken language that the structuralists advocated. According to the strategy of deconstruction there is no "one true meaning" for every text. Not that it means as Derrida argues "that all that is mentioned is suspended, denied, or locked up in a book, as many have claimed or been naive enough to believe, or have accused me of believing. But that all referent, all reality has the structure of a differential trace, and that one cannot refer to this "real" except within an interpretive experience."⁷ Deconstruction is a subversive theory that has significantly influenced modern thought and the overall structure of language, aiming to overturn logical thought.⁸ Deconstruction is a postmodern theory that "escaped" from the confines of semiotics and was applied to many areas. Deconstruction looks for dipoles and compares them, showing that the negative meaning of a word or an interpretation always exists and resides within the positive. "Darkness" lives in "light", because without darkness there would be no light.

The term "deconstruction" began to appear in the design world around the 1980s and was applied to trends in architecture, industrial design, graphic design and fashion design. Deconstruction was an attempt to critically review the system and institutions of society. Many artists were inspired by Derrida's ideas of deconstruction, and used the theory to question fundamental concepts, in an effort to search for new ways of expression.

Deconstruction being a theory that started in semiotics and influenced language, and inevitably communication, was also used in graphic and typographic design, as the deconstructionists of the time were obsessed with meaning and the lack of it. This was something that led them to experiment with typographic elements, in such a way that the meaning of their compositions was controversial and it was the viewer's role to give their own meaning to them. In other words, the deconstructionists removed the meaning from the written words⁹ and turned them into images.

⁷ Jacques Derrida, *Limited Inc.*, Gallilée, Paris 1990, p. 273

⁸ Miltos Phrangoroulos, *Εισαγωγή στην Ιστορία και τη θεωρία του Graphic Design: μια μικρή ανθολογία*, Futura, 2006

⁹ Janson Horst-Waldemar, Janson F. Anthony, *Ιστορία της Τέχνης: η Δυτική Παράδοση*, Ιων, 2011



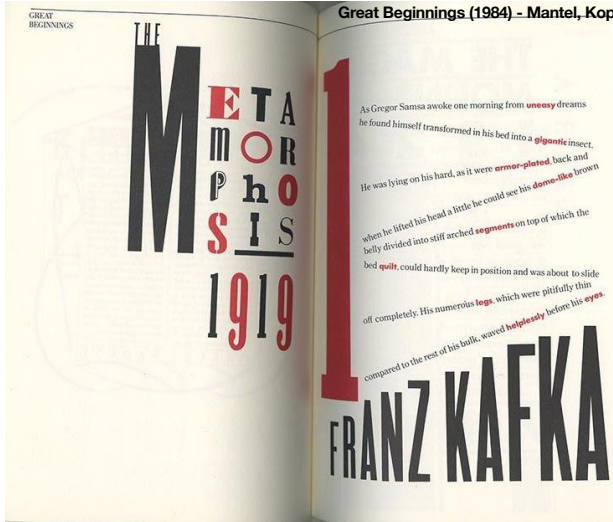
“when the deconstructionist approach is applied to design, each layer, through the use of language and image, is an intentional performer in a deliberately playful game wherein the viewer can discover and experience the hidden complexities of language.” - Chuck Byrne and Martha Witte

when the deconstructionist approach is applied to design, each layer, through the use of language and image, is an intentional performer in a deliberately playful game wherein the viewer can discover and experience the hidden complexities of language.”

-Chuck Byrne and Martha Witte

ConstructDe

@24



Great Beginnings (1984) - Mantel, Koppel, & Paula Scher, New York

New Wave

Το 1960, ο Wolfgang Weingart, αφού είχε ολοκληρώσει την τρίχρονη πρακτική του στην τυπογραφία, ακολούθησε τον Armin Hofmann, έναν από τους πιο σημαντικούς σχεδιαστές της εποχής του μοντερνισμού, ως καθηγητής στο Basel School of Design. Εκεί, ο τρόπος διδασκαλίας του ήταν πολύ διαφορετικός από τους συνδέλφους του.¹⁰ Ξεκίνησε να “σπάει” τους κανόνες του μοντερνισμού, ερευνώντας τις βασικές σχέσεις μεταξύ τυπογραφικών στοιχείων, όπως το μέγεθος, το βάρος, την κλίση, το διάκενο, το διάστιχο, και τα όρια της αναγνωσιμότητας. Πειραματίστηκε πολύ με το κενό μεταξύ των γραμμάτων, και “τέντωνε” τις λέξεις, μέχρι αυτές να γίνουν σχεδόν ακατάληπτες. Αυτή η νέα αυτή τάση της γραφιστικής ονομάστηκε New Wave / Postmodernism (Νέο Κύμα / Μεταμοντερνισμός).

Το 1978 δημοσιεύτηκε ένα πειραματικό τεύχος του περιοδικού “Visible Language” το οποίο είχε σχεδιαστεί από σπουδαστές του Cranbrook Academy of Art. Το τεύχος επικρίθηκε αρκετά από καταξιωμένους σχεδιαστές που χρησιμοποιούσαν προϋπάρχοντες κανόνες γραφιστικής και μεθόδους επίλυσης προβλημάτων επικοινωνίας. Παρ’ όλα αυτά, αποδείχθηκε ένα πολύ σημαντικό βήμα για την πειραματική γραφιστική καθώς το τεύχος αμφισβήτησε παραδοσιακούς κανόνες και αντί να εστιάζει στην επικοινωνία ξεκάθαρων μηνυμάτων, έδινε έμφαση στην καλλιτεχνική έκφραση και την ερμηνεία. Οι σπουδαστές της σχολής πειραματίστηκαν με διάφορες τεχνικές και υλικά, καθώς και με την σχέση τυπογραφίας και εικόνας. Σε αντίθεση με την μοντέρνα γραφιστική, που εστιάζει στην αποτελεσματική επικοινωνία ενός μηνύματος, το πειραματικό τεύχος προσέγγισε τη γραφιστική ως μορφή τέχνης, αφήνοντας χώρο για πολλαπλές ερμηνείες και νοήματα.

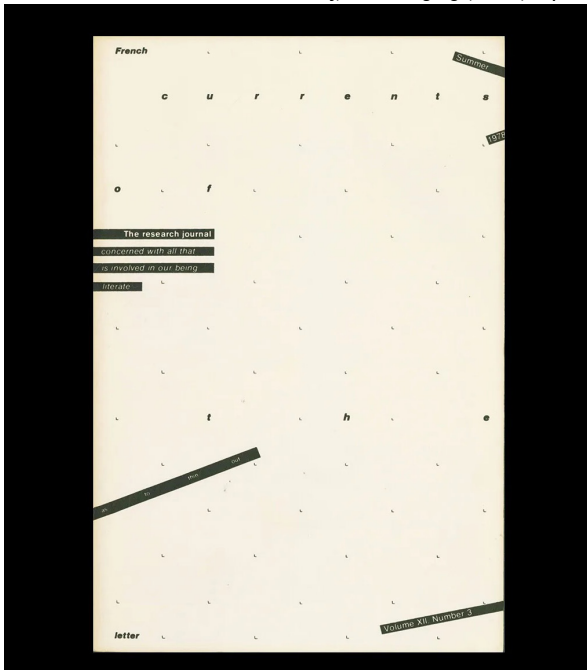
Wolfgang Weingart. *Kunstkredit* 1976-77, exhibition poster, Switzerland, 1977



In 1960, Wolfgang Weingart, after completing his three-year apprenticeship in typography, followed Armin Hofmann, one of the most important designers of the modernist era, as a professor at the Basel School of Design. There, his teaching style was very different from his contemporaries.¹⁰ He began to "break" the rules of modernism, investigating the basic relationships between typographic elements, such as size, weight, slope, spacing, punctuation, and the limits of legibility. He experimented a lot with the space between the letters, and "stretched" the words, until they became almost unintelligible. This new trend in graphic design was called New Wave / Postmodernism.

In 1978, an experimental issue of the magazine "Visible Language" was published, designed by students of the Cranbrook Academy of Art. The issue was heavily criticized by established designers who used pre-existing graphic design rules and communication problem solving methods. Nevertheless, it proved to be a very important step for experimental graphic design as the issue challenged traditional rules and instead of focusing on communicating clear messages, emphasized artistic expression and interpretation. They experimented with various techniques and materials, as well as with the relationship between typography and images. Unlike modern graphic design, which focuses on effectively communicating a message, the experimental issue approached graphic design as an art form, leaving room for multiple interpretations and meanings.

Cranbrook Academy, *Visible Language*, Vol 12, 03 (1978)



¹⁰ Philip B. Meggs, Alston W. Purvis, *Meggs's History of Graphic Design*. New York, John Wiley & Sons, 2016. p.446

C o m m u n i c a t i o n
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**
is not simply the transmission of information. Communication always calls for an exchange of ideas and perspectives. **Communication**

**“Communication
-true communication-
is not simply the
transmission of
information...
Communication always
calls for an exchange
of fluids and tensions,
for a provocation, and a
challenge.” - Fiorella Radice**



Post or Anti?

Ο αποδομισμός δίνει μεγάλη έμφαση στην ελεύθερη απόδοση του νοήματος και της προσωπικής ερμηνείας και έκφρασης του ατόμου, όπως ένα αφηρημένο έργο τέχνης. Επιπλέον, καθιστά τους σχεδιαστές και τους θεατές απαραίτητους για την απόδοση και την δημιουργία νοημάτων. Ο μεταδομισμός επανεξετάζει και τις αρχές του μοντερνισμού στη γραφιστική, αλλά και στην αρχιτεκτονική, όπου αρχίζουν να γεννιούνται οι πρώτες αμφιβολίες για αυτή την “αλάνθαστη”, παγκόσμια, και πλέον συντηρητική, οπτική γλώσσα, που όσο άλλαζαν οι καιροί, γινόταν όλο και πιο “ξένη”. Έγινε αναθεώρηση των αρχών που πρέσβευε ο μοντερνισμός, όπως η ασφάλεια και η σταθερότητα, αλλά και η εξάλειψη της διακόσμησης και της υπερβολής, ενώ οι μεταμοντερνιστές “αγκάλιασαν” τις αρνητικές έννοιες και έδιωξαν τα ταμπού, μια κίνηση που δημιούργησε γόνιμο έδαφος για την ανάπτυξη του προσωπικού στυλ και για πειραματισμούς.

Τα έργα του κινήματος του μεταμοντερνισμού μπορεί να έχουν κάποια κοινά σημεία με έργα του μοντερνισμού, αλλά η έμπνευση τους και σκοπός τους ήταν ριζικά διαφορετικός. Γραφιστικά, αυτό μεταφράζεται σε συνδυασμό ανατρεπτικών και εναλλακτικών δηλώσεων με μέσα που θα είχαν εμπορική απτήρηση. Η “υψηλή” και η ποπ κουλτούρα γίνονται ένα και υπάρχει εγκατάλειψη των διαχωρισμών μεταξύ στυλ. Ο μεταμοντερνισμός είχε και ένα απaráληπτο πολιτικό υπόβαθρο, καθώς εμφανίστηκε σε μια μεταβατική μεταπολεμική εποχή, η οποία χαρακτηρίζεται από την άνοδο των μέσων μαζικής ενημέρωσης και την αντιμετώπιση του “less is bare”.

Κατά τη γνώμη μου, συγκρίνοντας τα έργα των δύο κινήματων, ο μεταμοντερνισμός είναι μια εξέλιξη του μοντερνισμού, καθώς είναι πιο κοντά στους ανθρώπους και στο τι περιμένουν και θέλουν από τη γραφιστική. Οι μεταμοντερνιστές δεν υποστήριζαν απαραίτητα την μη ύπαρξη των κανόνων στη γραφιστική, καθώς η γραφιστική χωρίς κανόνες θα έπαυε να είναι γραφιστική, αλλά χαρακτηρίζονται από το γεγονός ότι δημιουργούσαν σαν να μην υπάρχουν κανόνες και όρια. Δεν έδιναν τόση προσοχή στη λειτουργικότητα και την αναγνωσιμότητα, όση έδιναν στην συνολική εικόνα μιας σύνθεσης.¹¹

Το κίνημα του μεταμοντερνισμού γέννησε πολλές συζητήσεις, και στην αρχή ήρθε αντιμέτωπο με πολλές κακές κριτικές, κάτι που ήταν αναμενόμενο, καθώς ο σχεδιαστικός κόσμος είχε να αντιμετωπίσει τη ραγδαία εξέλιξη της τεχνολογίας που δημιουργούσε φόβους και ανασφάλειες στους σχεδιαστές, κάτι που το κίνημα του μεταμοντερνισμού “αγκαλιάζει” και την δοξάζει ως ένα νέο μέσο έκφρασης. Οι νέοι σχεδιαστές έψαχναν τρόπους για να δοκιμάσουν τα πάντα, ενώ οι πιο παλαιοί, συνήθως ορκισμένοι μοντερνιστές, θεωρούσαν ότι τα πάντα έχουν δοκιμαστεί ήδη.¹² Αυτή η αντίληψη αποτέλεσε την αρχή του τέλους της κυριαρχίας του μοντερνισμού, την θέση του οποίου, πήρε ο μεταμοντερνισμός.

Deconstruction places great emphasis on the free rendering of meaning and one's personal interpretation and expression, such as an abstract work of art. Moreover, it makes designers and viewers indispensable part of meaning-making. Poststructuralism also reexamines the principles of modernism in graphic design, but also in architecture, where the first doubts about this "infallible", universal, and now conservative, visual language, which as times have changed, became more and more irrelevant. The principles espoused by modernism, such as safety and stability, but also the elimination of decoration and excess, were revised, while the postmodernists embraced negative concepts and banished taboos, a move that created fertile ground for the development of personal style and experimentation.

The works of the postmodernist movement may have some commonalities with works of modernism, but their inspiration and purpose were radically different. Design wise, this translates into combining subversive and alternative statements with media that would have commercial appeal. "High" and pop culture become one and there is an abandonment of distinctions between styles. Postmodernism also had an underappreciated political background, as it emerged in a transitional post-war era, characterized by the rise of mass media and the "less is bare" treatment.

In my opinion, comparing the works of the two movements, postmodernism is an evolution of modernism, as it is closer to people and what they expect and want from graphic design. Postmodernists did not necessarily advocate the absence of rules in graphic design, since graphic design without rules would cease to exist, but they are characterized by the fact that they created as if there were no rules and boundaries. They didn't pay as much attention to functionality and readability as they did to the overall picture of a composition.¹¹

The postmodernism movement generated a lot of debate, and at first it was met with a lot of bad reviews, which was to be expected, as the design world had to deal with the rapid development of technology that created fears and insecurities among designers, the technology which the postmodernism movement embraces and glorifies as a new means of expression. Young designers were looking for ways to try everything, while the older ones, usually sworn modernists, assumed that everything had already been tried.¹² This perception marked the beginning of the end of the dominance of modernism, which was replaced by postmodernism.

¹¹ Rick Poynor, *No More Rules*. Yale University Press, 2003, p. 16

¹² Steven Heller, *Revisiting the So-Called "Legibility Wars" of the '80s and '90s*. (2016) <https://design.sva.edu>

“As meaning is lost, the materiality of words becomes **o b s e s s i v e**, as is the case when children **repeat** a word over and over again until its sense is lost... a signifier that has lost its signified has thereby been transformed into an image.” -Fredric Jameson

repeat repeat repeat repeat repeat repeat repeat repeat repeat repeat
 repeat repeat repeat repeat repeat repeat repeat repeat repeat repeat
 repeat repeat repeat repeat repeat repeat repeat repeat repeat repeat
 repeat repeat repeat repeat repeat repeat repeat repeat repeat repeat



“Καθώς χάνεται το νόημα,

η υλικότητα

των λέξεων

γίνεται έμμονη,

όπως συμβαίνει

όταν τα παιδιά

επαναλαμβάνουν

μια λέξη

ξανά και ξανά

μέχρι να χαθεί

η αίσθηση της...

ένα σημαίνον

που έχει χάσει

το σηματομενό του

έχει μετατραπεί έτσι

σε εικόνα.”

-Fredric Jameson

ConstructDe



“ R u l e s

c a n

but

never

-David Jury

ConstructDe

@34

b e b r o k e n
ignored”

ConstructDe

@35

Type is not dead

Με την εισαγωγή των ηλεκτρονικών υπολογιστών, της φωτογραφίας και της διαδικτυακής επανάστασης (internet revolution), ήρθαν μεγάλες αλλαγές στην διαδικασία παραγωγής εντύπων και γραφικών, αλλά οι Η/Υ έφεραν και νέα ζωή στην τυπογραφία, καθώς έφεραν τεράστιες αλλαγές στον σχεδιασμό γραμματοσειρών. Η ανάπτυξη της γλώσσας PostScript και η χρήση των καμπύλων Bezier στο σχεδιασμό γραμματοσειρών,¹³ βοήθησαν σημαντικά με τον τρόπο τους την σχεδίαση στοιχείων και γραφικών, καθώς σε αυτές τις τεχνολογίες βασίστηκαν τα διανυσματικά αρχεία, που χρησιμοποιούνται μέχρι σήμερα. Με την εισαγωγή των Η/Υ, οι πιο δημοφιλείς γραμματοσειρές ψηφιοποιήθηκαν, ενώ οι υπόλοιπες χάθηκαν με το πέρασμα του χρόνου.

Η άφιξη του Apple Macintosh το 1984 είχε προκαλέσει διχασμό μεταξύ των σχεδιαστών, καθώς άλλοι από αυτούς πίστευαν ότι ο υπολογιστής ήταν απλώς ένα εύρηστο εργαλείο παραγωγής, ενώ κάποιοι το θεωρούσαν μέσο απελευθέρωσης μιας νέας γραφιστικής έκφρασης. Αποδείχθηκε ότι ήταν και τα δύο αυτά πράγματα, αλλά ήταν επίσης ένας εκστατικός, χωρίς κανόνες χώρος, όπου οι μεταμοντερνιστές θα ανακάλυπταν καινούρια στυλ έκφρασης επηρεασμένα από τα πολιτικά κινήματα και τις ανησυχίες της εποχής. Οι υπολογιστές άλλαξαν τους φραγμούς και τους περιορισμούς που είχαν οι γραφίστες, θέτοντας αναπόφευκτα νέους, καθιερώνοντας έναν νέο τρόπο εργασίας. Ακόμη και το γεγονός ότι οι υπολογιστές άλλαζαν και τη ροή της εργασίας, αφού πλέον οι γραφίστες δούλευαν σε οθόνες και έπειτα μετέφεραν τις δουλειές τους σε χαρτί αντί να δουλεύουν απευθείας σε αυτό, ανάγκασαν πολλούς σχεδιαστές να εκπαιδευτούν εκ νέου για να μπορέσουν να συμβαδίσουν με τις ραγδαίες εξελίξεις στον κλάδο τους. Για τους σχεδιαστές που ανησυχούσαν για την σχέση μεταξύ εικόνας και κειμένου, ο υπολογιστής ερχόταν να δώσει ένα τέλος στις παλιές ιεραρχίες, συσσωρεύοντας όλα τα μέσα σε ένα ψηφιακό εργαλείο.

Μικρά πράγματα, όπως η εισαγωγή της εντολής “undo”, άλλαξαν εντελώς τον τρόπο παραγωγής γραφικών, καθώς όπως αναφέρει και η April Greiman, αυτή η ευκολία διαγραφής δίνει περιθώρια αβεβαιότητας και αστάθειας που επεκτείνονται σε όλο το έργο, αφού πλέον όλα μπορούν να αλλάξουν.¹⁴

“Αυτό που βιώνω, είναι αντί να κάνουμε κάτι πιο γρήγορα, κοιτάμε περισσότερες δυνατότητες. Αντί να κάνουμε λιγότερη δουλειά, βλέπουμε περισσότερες επιλογές. Κανονικά, δεν θα δοκίμαζες δώδεκα διαφορετικά μεγέθη γραμματοσειράς επικεφαλίδας, επειδή θα περιλάμβανε την αντίστοιχη δι-α-ρ-ύ-θ-μ-ι-σ-η για κάθε παραλλαγή. Είναι πολύ ακριβό και παίρνει πολύ χρόνο. Αλλά με το Mac, μόλις αποθηκευτούν οι πληροφορίες, μπορείς να δεις αμέσως 72.000 παραλλαγές.”¹⁵

Ο τρόπος εργασίας του σχεδιαστή για να ολοκληρώσει ένα πρότζεκτ άλλαξε δραματικά. Οι συνθέσεις που δημιουργούσαν οι σχεδιαστές στο νέο περιβάλλον των Macintosh, ήταν ριζοσπαστικές, καθώς πλέον ήταν δυνατή η επιλογή ανάμεσα σε εκατοντάδες γραμματοσειρές έτοιμες για χρήση, ενώ οι εικόνες και τα γραφικά μπορούσαν να τροποποιηθούν και να παραμορφωθούν με τρόπους που δεν ήταν δυνατοί ως τότε.

With the introduction of computers, photography and the internet revolution, came major changes in the process of producing print and graphics, but computers also brought new life to typography, bringing huge changes to typeface design. The development of the PostScript language and the use of Bezier curves in font design,¹³ greatly aided the design of elements and graphics in their own way, as these technologies were the basis of the vector files still in use today. With the introduction of computers, the most popular fonts were digitized, while the rest were lost over time.

The arrival of the Apple Macintosh in 1984 had caused a division among designers, as some of them believed that the computer was just a handy production tool, while some saw it as a means of unleashing a new graphic expression. It turned out to be both of these things, but it was also an ecstatic, rule-free space where postmodernists would discover new styles of expression influenced by the political movements and concerns of the time. Computers changed the barriers and limitations that graphic designers had, inevitably setting new ones, establishing a new way of working. Even the fact that computers were also changing the workflow, with graphic designers now working on screens and then transferring their work onto paper instead of working directly on it, forced many designers to retrain to keep up with the rapid developments in their industry. For designers concerned with the relationship between image and text, the computer was coming to put an end to the old hierarchies, assembling all media into one digital tool.

Small things, like the introduction of the “undo” command, completely changed the way graphics were produced, as April Greiman also mentions, this ease of erasing gives room for uncertainty and instability that extends throughout the work, since now everything can change.¹⁴ The compositions that designers created in the new Macintosh environment were radical, as it was now possible to choose from hundreds of ready-to-use fonts, while images and graphics could be modified and distorted in ways not possible before.

“What I experience is rather than doing something more quickly, we’re looking at more possibilities. Instead of doing less work we are seeing more options. Ordinarily, you wouldn’t try twelve different sizes of headline type because it would involve setting it and statting it for each variation. It’s too expensive and takes too much time. But with the Mac, once the information is stored you can look at 72,000 variations immediately.”¹⁵

The way a designer worked to complete a project had changed dramatically. The compositions that designers created in the new Macintosh environment were radical, as it was now possible to choose from hundreds of ready-to-use fonts, while images and graphics could be modified and distorted in ways not possible before.

¹³ Philip B. Meggs, Alston W. Purvis, *Meggs's History of Graphic Design*. New York, John Wiley & Sons, 2016. p.572

¹⁴ April Greiman, *Hybrid Imagery: The Fusion of Technology and Graphic Design*, New York, Watson-Guption Publications, 1990, p. 56

¹⁵ April Greiman, *Hybrid Imagery: The Fusion of Technology and Graphic Design*, New York, Watson-Guption Publications, 1990, p. 56



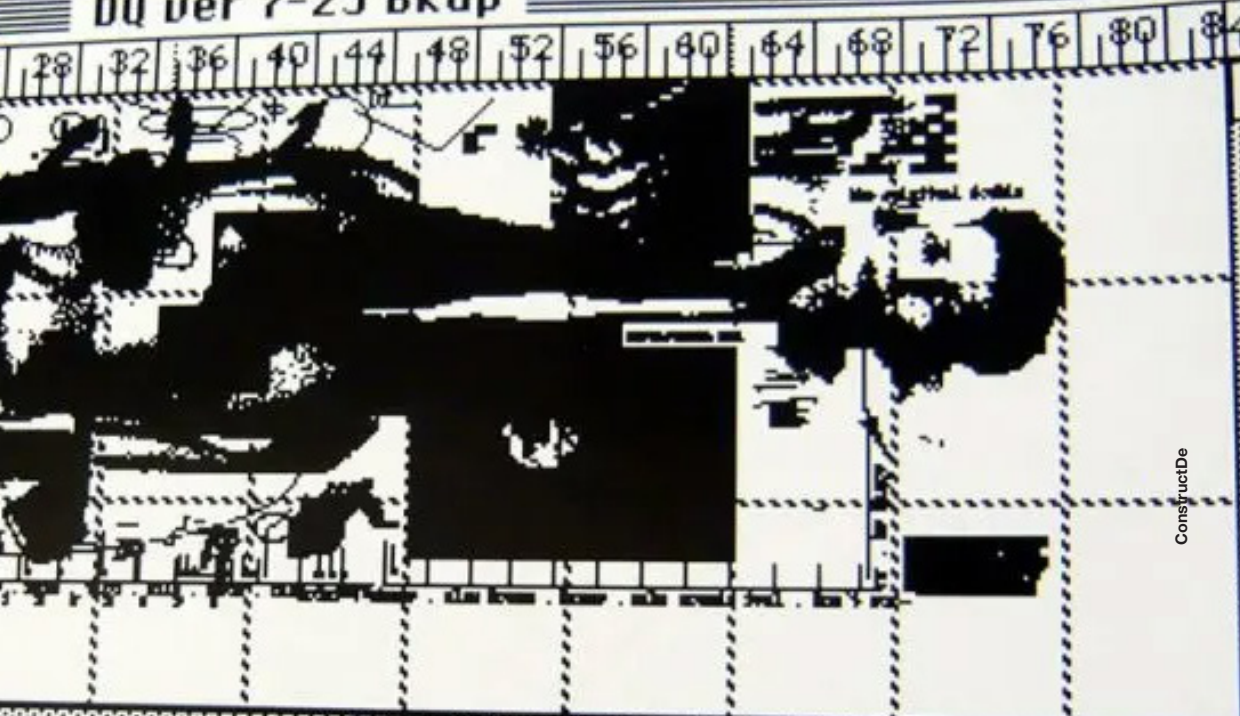
**In digital space,
nothing is ever
finished; as long as
the computer file
survives, any element
can be rethought -
“The paint never dries
in the Mac universe.”**

ConstructDe



ont Layout Arrange Fill Lines Pen

DQ ver 7-23 Bkup



ConstructDe

smaller but more

ive views Each view

Phase 2.4

April Greiman, *Does it make sense?*, 1986, courtesy the Design Museum

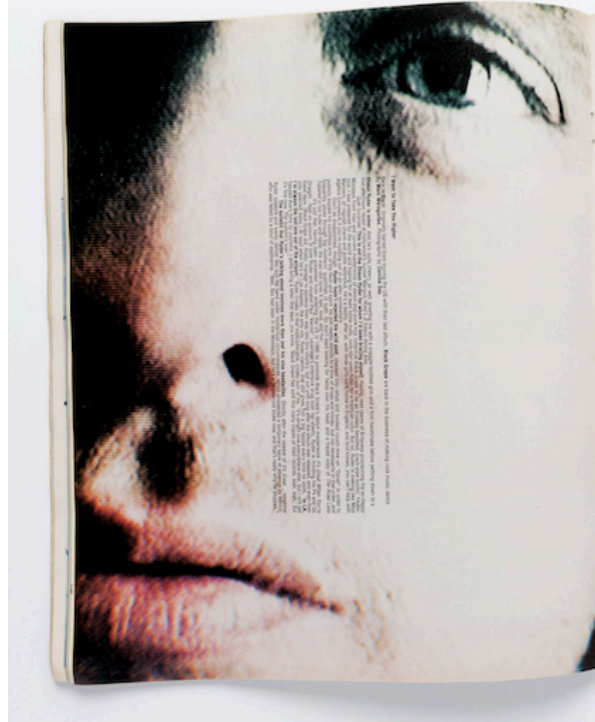


Exploring (il)legibility

Ο τρόπος παραγωγής της επικοινωνίας τη δεκαετία του '90 είχε αλλάξει δραστικά σε σχέση με τα προηγούμενα χρόνια, καθώς πλέον δεν υπήρχαν συζητήσεις γύρω από το αν οι υπολογιστές είναι καλοί για την γραφιστική ή αν πρέπει να τους φοβούνται οι σχεδιαστές, (όπως οι συζητήσεις που γίνονται στις μέρες μας για την τεχνίτη νοημοσύνη), αλλά όλη η δουλειά του γραφίστα γινόταν πλέον ψηφιακά, πέρα από μερικές μικρές εξαιρέσεις. Οι γραφίστες της εποχής πειραματίστηκαν με αυξομειώσεις στα μεγέθη της τυπογραφίας, με ιδιαίτερες διακοσμητικές γραμματοσειρές, με την εξάλειψη του σχεδιαστικού grid κ.α. εμπνευσμένοι από τους Ντανταϊστές και τους Φουτουριστές. Ως αποτέλεσμα, οι σχεδιαστές ξεκίνησαν να ψάχνουν νέους τρόπους επικοινωνίας, και άρχισαν να σκέφτονται έξω από τα όρια που τους όριζαν τα παλαιότερα κύματα γραφιστικής. Οι πειραματισμοί τους αυτοί, εξελίχθηκαν, μεταξύ άλλων, σε έργα με μη αναγνώσιμη, εσκεμμένα, τυπογραφία, που δοκίμαζαν τα όρια της αναγνωσιμότητας, κάτι που φυσικά προκάλεσε τεράστιες αντιδράσεις από τους αναγνώστες, αλλά φυσικά και τους φανατικούς μοντερνιστές. Αυτό το ύφος γραφιστικής δεν ήταν ιδιαίτερα εμπορικό, τουλάχιστον στα πρώτα του στάδια, καθώς δεν είναι εύκολα αποκωδικοποιήσιμο από τον μέσο αναγνώστη, κάτι που την προόριζε για μια πολύ συγκεκριμένη ελίτ, που είχε τις γνώσεις για να ερμηνεύσει το έργο και να καταλάβει αυτό που ήθελε να επικοινωνήσει ο σχεδιαστής.

ConstructiDe

@40



The way communication was produced in the 1990s had changed drastically from previous years, as there were no longer debates about whether computers were good for graphic design or whether designers should be afraid of them (like the debates that are done nowadays about artificial intelligence), but all of the graphic designer's work was now done digitally, apart from a few minor exceptions. The graphic designers of the time experimented with changes in the sizes of the typography, with special decorative fonts, with the elimination of the design grid, etc. inspired by the Dadaists and Futurists. As a result, designers began to look for new ways of communication, and began to think outside the boundaries defined by the older waves of graphic design. These experiments, among other things, evolved into works with illegible, deliberately, typography, which tested the limits of legibility, which of course caused huge reactions from readers, but of course also fanatical modernists. This style of graphic design was not particularly commercial, at least in its early stages, as it was not easily decipherable by the average reader, which reserved it for a very specific elite, who had the knowledge to interpret the work and understand what the designer wanted to communicate.

Chris Ashworth & David Carson, *Ray Gun*, Issue 53



Phase
2.5

**Something is
ille
reæ
fac
ab
wr
code.**

**un-
itin
ad-
ose
the**

ConstructiDe

@42

egible it is
readable, but
in fact, it is re
usable for the
whole crack

-Steven Heller

ConstructDe

@43

Experimental fonts

@44



x Creative coding

ConstructDe

@45

Progress Sans

pt. 12 Progress Sans

pt. 14 is an experimental font

pt. 18 created by using

pt. 24 creative coding

pt. 30 for distorting type!

pt. 6 (and pushing the boundaries of readability.)

pt. 12 Η progress sans είναι

pt. 14 μια πειραματική γραμματοσειρά

pt. 18 που έχει δημιουργηθεί με τη χρήση

pt. 24 δημιουργικού προγραμματισμού

pt. 30 για την παραμόρφωση
στοιχείων!

pt. 6 (και για τη δοκιμή των ορίων της αναγνωσιμότητας.)

01 23456789!?:;,.,
@#%&*()-=+{|/
@

Progress Gen

pt. 12 Progress Gen

pt. 14 is an experimental font

pt. 18 created by using

pt. 24 creative coding

pt. 30 for distorting type!

pt. 6 (and pushing the boundaries of readability.)

pt. 12 Η progress gen είναι

pt. 14 μια πειραματική γραμματοσειρά

pt. 18 που έχει δημιουργηθεί με τη χρήση

pt. 24 δημιουργικών τεχνικών προγραμματισμού

pt. 30 για την παραμόρφωση
στοιχείων!

pt. 6 (και για τη δοκιμή των ορίων της αναγνωσιμότητας.)

pt. 36 0123456789!?,.,@#
\$%&*()_=>+{}
@47

Progress Sans

Η Progress Sans είναι μια σειρά γραμμάτων η οποία έχει δημιουργηθεί με βάση μια sans serif γραμματοσειρά, η οποία έχει παραμορφωθεί με την χρήση κώδικα. Η επιλογή των χαρακτήρων έχει γίνει χειροκίνητα, καθώς και για κάθε στοιχείο έχουν προσαρμοστεί οι παράμετροι στον κώδικα, ώστε να υπάρχει ένα εμφανές αποτέλεσμα και μια ενιαία αισθητική σε όλο το σύνολο των στοιχείων, διατηρώντας όμως την “οργανικότητα” που μπορεί να έχει αυτού του είδους η παραμόρφωση. Η Progress Sans “αγκαλιάζει” και επιδεικνύει την ανομοιομορφία και τις ατέλειες του κάθε στοιχείου.

Η γραμματοσειρά αποτελείται από 166 γλύφους και περιλαμβάνει λατινικούς και ελληνικούς χαρακτήρες, πεζούς και κεφαλαίους, καθώς και αριθμούς και σημεία στίξης.

Οι προτεινόμενες χρήσεις της σειράς είναι για τίτλους βιβλίων και κειμένων, για έντυπα, αλλά και για ψηφιακές χρήσεις.

Progress Sans is a series of letters that has been created based on a sans serif font, which has been distorted using code. The selection of characters has been done manually, as well as for each element the parameters have been adjusted in the code, so that there is an obvious effect and an uniform aesthetic throughout the whole set of glyphs, but maintaining the “organicity” that this type of distortion has. Progress Sans embraces and accentuates the unevenness and imperfections of each glyph.

The font consists of 166 glyphs and includes Latin and Greek characters, lowercase and uppercase, as well as numbers and punctuation.

Suggested uses of the typeface, are for book and text titles, for print, but also for digital uses.



A B C D E F G H I J
 K L M N O P Q R S T
 U V W X Y Z Α Β Γ Δ
 Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ
 Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω
 Ά Ξ Η Ι Ό Υ Ω ΐ ΰ a b
 c d e f g h i j k l m n
 o p q r s t u v w x y
 z α β γ δ ε ζ η θ ι κ λ
 μ ν ξ ο π ρ ς σ τ υ φ
 χ ψ ω ÿ ü ó ú ó 0 1 2
 3 4 5 6 7 8 9 ! # \$ %
 & ' () * + , - . / : ;
 = ? @ { } " ' " ' , " "

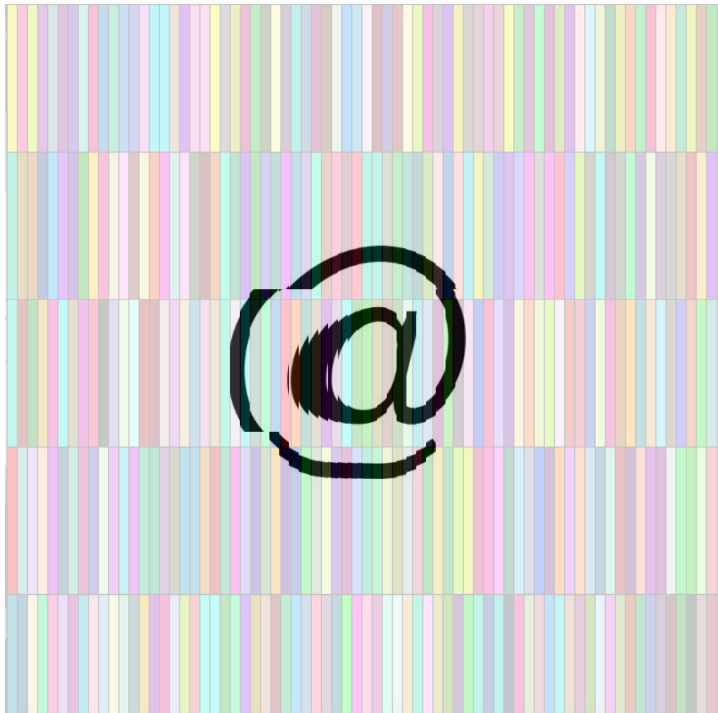


Οι πειραματισμοί μου με κώδικα για την παραμόρφωση στοιχείων, είχε ως αποτέλεσμα την σειρά γραμμάτων Progress Sans. Για την δημιουργία της, χρησιμοποίησα κώδικα που έχει γραφτεί σε γλώσσα προγραμματισμού Java, στο προγραμματιστικό περιβάλλον Processing, όπου με οδήγησε σε μερικά συμπεράσματα:

1. Τα αγγλικά πεζά γράμματα έχουν λιγότερες καμπύλες από τα ελληνικά, και για αυτόν τον λόγο έχουν και μικρότερο φάρδος.
2. Τα περισσότερα αγγλικά κεφαλαία έχουν καμπύλες.
3. Στα ελληνικά κεφαλαία δεν είναι τόσο εμφανής η κίνηση, τα γράμματα φαίνονται σαν να «κόβονται».

Επομένως, συμπεράνα ότι χρειάστηκε να πάρω διαφορετικές αποφάσεις για το ποσοστό παραμόρφωσης του κάθε στοιχείου, ανάλογα με την φόρμα του, εάν δηλαδή, έχει πολλές καμπύλες, εάν έχει οριζόντιες και κάθετες γραμμές κ.α., ώστε να υπάρχει μια οπτική συνοχή στα στοιχεία και να φαίνονται σαν να έχουν ίδιο ποσοστό παραμόρφωσης μεταξύ τους.

Για τη δημιουργία της Progress Sans χρησιμοποίησα σαν βάση τον κώδικα που έχει γραφτεί από τον Tim Rodenbröcker [www.timrodenbroecker.de], όπου συνοπτικά, δημιουργεί έναν τετράγωνο λευκό καμβά, τοποθετεί το στοιχείο της επιλογής μας στο κέντρο του σε μαύρο χρώμα, δημιουργεί έναν κάνναβο με συγκεκριμένο αριθμό ορθογώνιων στους άξονες x και y, στην περίπτωση μας (x=70 και y=5), διαιρεί το στοιχείο με βάση τον κάνναβο, εισάγει το κύμα που δημιουργεί η γραφική παράσταση του ημιτόνου με μεταβλητές το x, το y και το framecount και υπολογίζει την θέση του κάθε ορθογώνιου με βάση την τιμή του κύματος, και έτσι το κάθε ορθογώνιο στον κάνναβο μετατοπίζεται με διαφορετική ταχύτητα από τα υπόλοιπα, δημιουργώντας το εφέ της παραμόρφωσης.

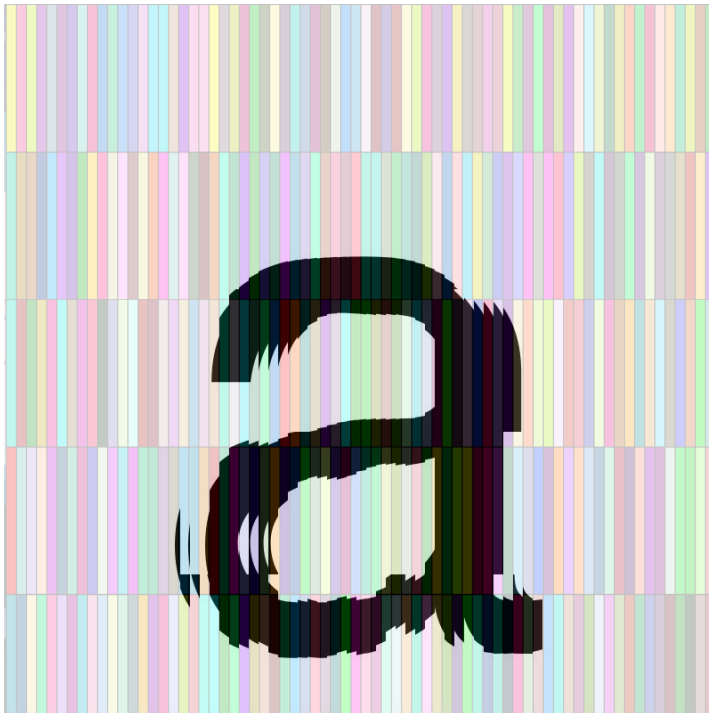


My experimentation with element distorting code resulted in the Progress Sans typeface. To create it, I used code written in Java programming language, in the Processing programming environment, where it led me to some conclusions:

- 1. English lowercase letters have fewer curves than Greek, and for this reason they have a smaller width.*
- 2. Most English capitals have curves.*
- 3. In Greek capitals, the movement is not so obvious, the letters seem to be «cut».*

Therefore, I found that I had to make different decisions about the percentage of distortion of each element, depending on its form, that is, if it has many curves, if it has horizontal and vertical lines, etc., so that there is a visual consistency in the elements and to appear as if they have the same distortion rate with each other.

To create Progress Sans I used as a basis the code written by Tim Rodenbröcker [www.timrodenbroecker.de], where in brief, it creates a square white canvas, places the element of our selection in the center of it in black, creates a canvas with a certain number of rectangles on the x and y axes, in our case (x=70 and y=5), divides the element based on the canvas, inserts the wave created by the sin wave with the variables x, y and the framecount and calculates the position of each rectangle based on the wave value, and thus each rectangle on the canvas moves at a different speed than the others, creating the effect of distortion.



@52

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

```

PGraphics pg;
void setup() {
  font = createFont("UB-Helvetica.ttf", 800);
  size(800,800,P2D);
  pg = createGraphics(800,800,P2D);
  background(0);
  pg.beginDraw();
  pg.fill(0);
  pg.textFont(font);
  pg.textSize(500);
  pg.background(255);
  pg.translate(width/2, height/2-50);
  pg.textAlign(CENTER,CENTER);
  pg.pushMatrix();
  pg.text("a", 0, 0);
  pg.popMatrix();
  pg.endDraw();
  image(pg,0,0);
  int tilesX = 70;
  int tilesY = 5;
  int tileW = int(width/tilesX);
  int tileH = int(height/

```



```
tilesY);  
for(int y = 0; y < tilesY; y++) { for(int x = 0; x < tilesX; x++) { int wave =  
ir // ir // ir s;  
eW; int sh = tileH;  
dh = tileH; copy(pg,  
-####.jpg"); }  
}
```



KINETIC TYPOGRAPHY1_PTUXIAKI

Tim Rodenbröcker, Kinetic Typography 1
<https://timrodenbroecker.de/processing-tutorial-kinetic-typography-1/>

@54

ConstructiDe

a

Progress Sans "a" - 600 pt.

Phase
3.1

a

@56

ConstructiDe

E

Progress Sans "e" - 600 pt.

Phase
3.1

3

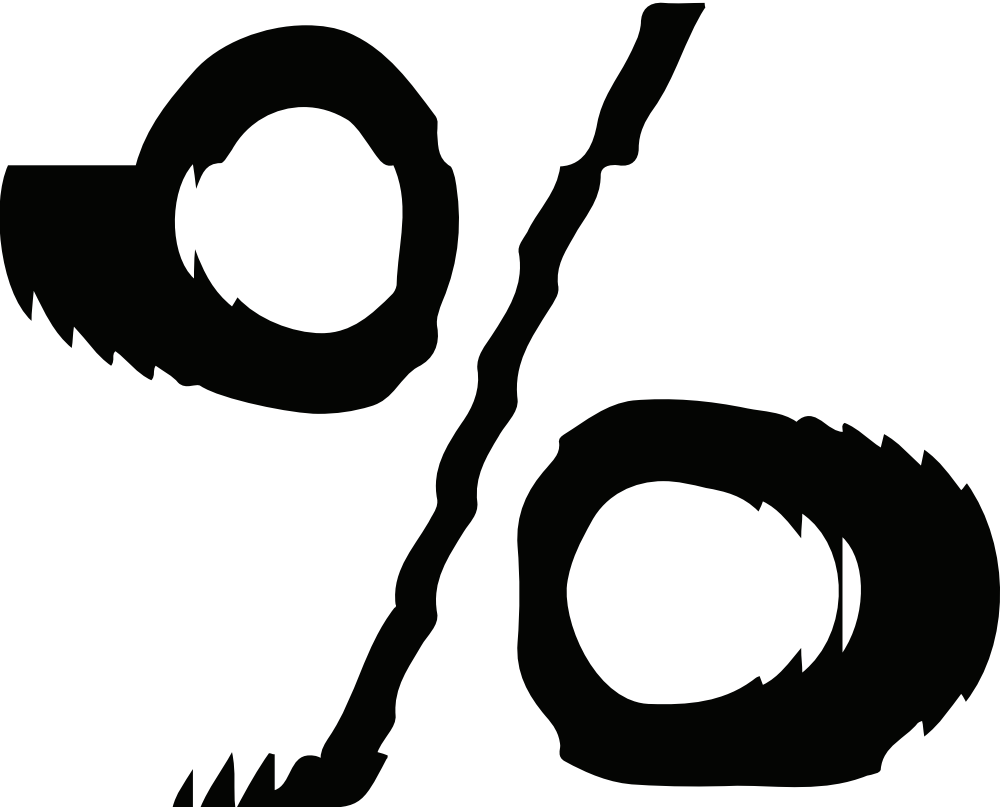
@58

ConstructiDe

@

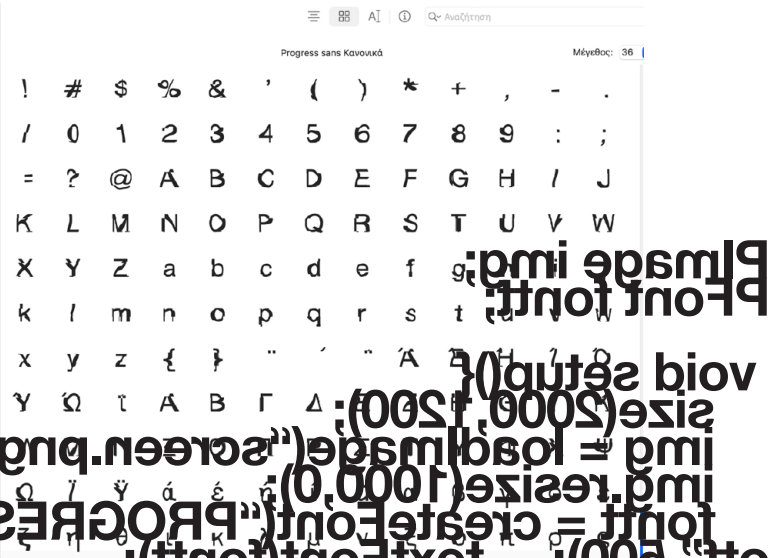
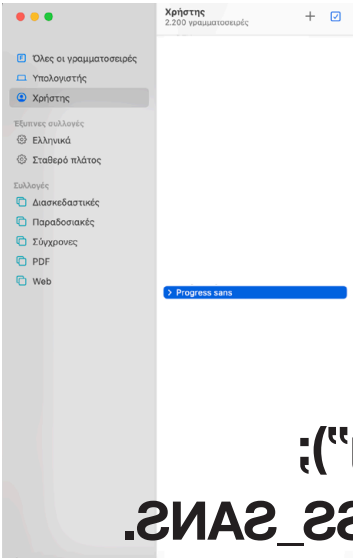
Progress Sans "@" - 420 pt.

Phase
3.1

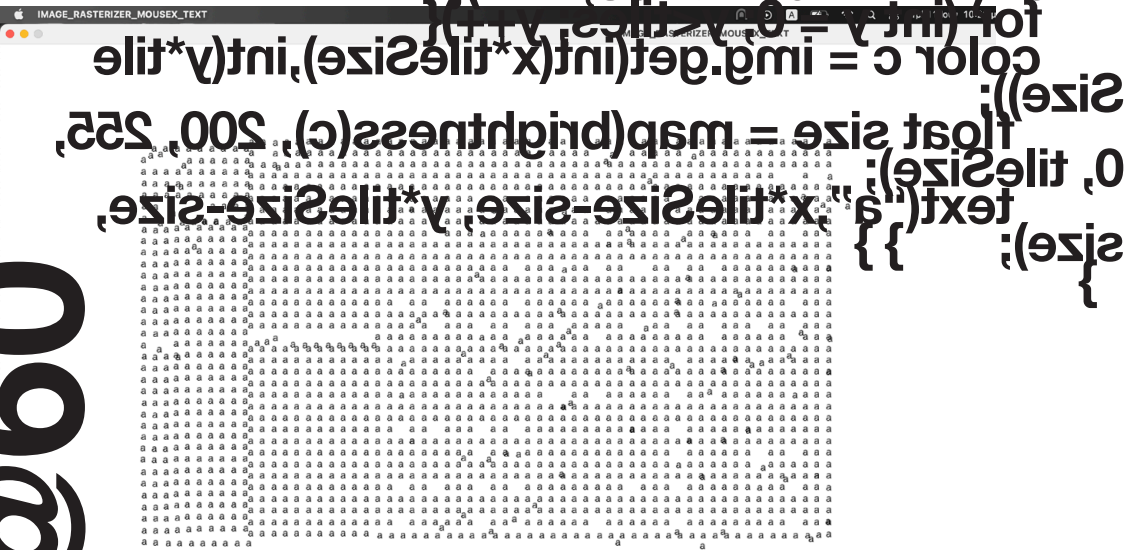


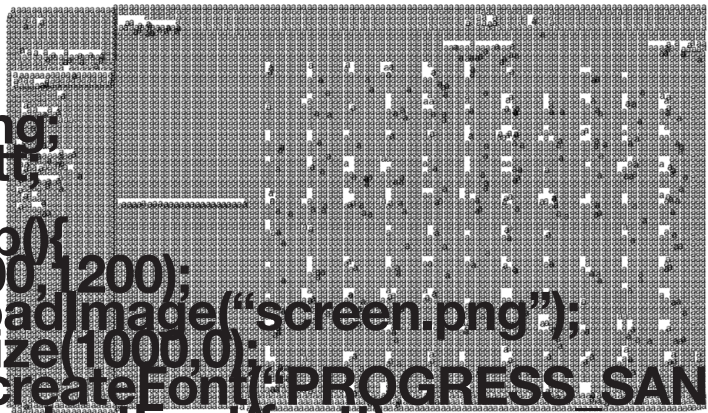
ConstructDe

@59



ConstructiDe





```

PImage img;
PFont fontt;

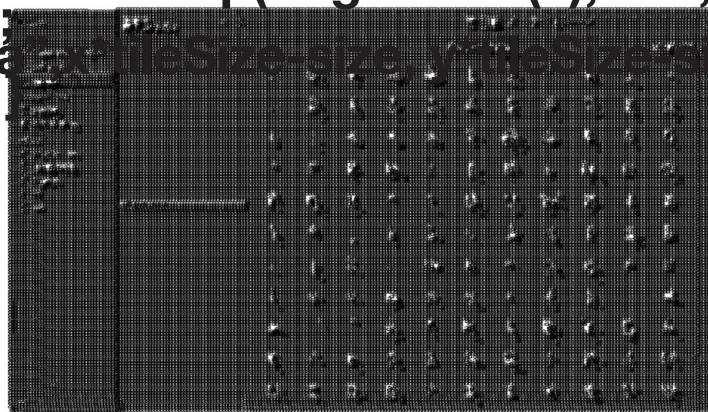
void setup()
  size(2000,1200);
  img = loadImage("screen.png");
  img.resize(1000,0);
  fontt = createFont("PROGRESS SANS.
otf",500); textFont(fontt);
}

```

```

void draw(){
  background(255);
  fill(0);
  noStroke();
  textAlign(CENTER);
  float tiles = mouseX/5;
  float tileSize = width/tiles/2;
  translate(tileSize/2+200,tileSize/2+150);
  textSize(30);
  for (int x = 0; x<tiles; x++){
    for (int y = 0; y<tiles; y++){
      color c = img.get(int(x*tileSize),int(y*tile
Size));
      float size = map(brightness(c), 200, 255,
0, tileSize);
      text("a",tileSize/2+size,width/2+size,
size);
    }
  }
}

```



ConstructDe

@61

Progress Gen

Η Progress Gen είναι μια γραμματοσειρά εμπνευσμένη από την «*Μεταμόρφωση*» του Φ. Κάφκα, η οποία έχει δημιουργηθεί με βάση μια sans serif γραμματοσειρά, η οποία έχει παραμορφωθεί με την χρήση κώδικα. Ο κώδικας μετατρέπει τους κόμβους του κάθε γλύφου σε σημεία και τα ενώνει με μια συνεχόμενη γραμμή. Ο χρήστης με την κίνηση του κέρσορα του μπορεί να ελέγξει τον αριθμό των σημείων που θα αποτελούν το κάθε στοιχείο, και όσο τα σημεία μειώνονται, τόσο το στοιχείο γίνεται πιο αφηρημένο και ιδιαίτερο. Τα στοιχεία και ο τρόπος με τον οποίο έχουν σχεδιαστεί θυμίζουν έντομα και παραπέμπουν στην πλοκή του έργου του Φ. Κάφκα.

Η γραμματοσειρά αποτελείται από 159 γλύφους και περιλαμβάνει λατινικούς και ελληνικούς χαρακτήρες, πεζούς και κεφαλαίους, καθώς και αριθμούς και σημεία στίξης.

Οι προτεινόμενες χρήσεις της γραμματοσειράς είναι για τίτλους βιβλίων και κειμένων, για έντυπα, αλλά και για ψηφιακές χρήσεις.

Progress Gen is a typeface inspired by F. Kafka's "*Metamorphosis*", which has been created based on a sans serif typeface, which has been distorted using code. The code converts the nodes of each glyph into points and joins them with a continuous line. The user with the movement of their cursor can control the number of points that will make up each glyph, and as the points decrease, the element becomes more abstract and special. The elements and the way they are designed are reminiscent of insects and are a reference to the plot of F. Kafka's work.

The font consists of 159 glyphs and includes Latin and Greek characters, lowercase and uppercase, as well as numbers and punctuation.

Suggested uses of the typeface, are for book and text titles, for print, but also for digital uses.



Οι πειραματισμοί μου με κώδικα για την παραμόρφωση στοιχείων, είχε ως αποτέλεσμα την γραμματοσειρά Progress Gen. Για την δημιουργία της, χρησιμοποίησα κώδικα που έχει γραφτεί σε γλώσσα προγραμματισμού Javascript, στο προγραμματιστικό περιβάλλον P5.js, όπου με οδήγησε στο εξής συμπέρασμα:

Τα αγγλικά πεζά γράμματα είναι πιο αναγνωρίσιμα με λιγότερους κόμβους από ότι τα αντίστοιχα ελληνικά.

Επομένως, συμπεράνα ότι χρειάστηκε να πάρω διαφορετικές αποφάσεις για το ποσοστό παραμόρφωσης του κάθε στοιχείου, ανάλογα με την φόρμα του, εάν δηλαδή, έχει πολλές καμπύλες, εάν έχει οριζόντιες και κάθετες γραμμές κ.α., ώστε να υπάρχει μια οπτική συνοχή στα στοιχεία και να φαίνονται σαν να έχουν ίδιο ποσοστό παραμόρφωσης μεταξύ τους. Προϋπόθεση για αυτή τη διαδικασία, είναι η γνώση του αλφαβήτου του οποίου γίνεται η παραμόρφωση, καθώς αν εφαρμοστεί η ίδια παραμόρφωση σε όλα τα στοιχεία ανεξαρτήτως αλφαβήτου, τότε θα μας παρουσιαστούν διαφορετικά αποτελέσματα. Οπότε, η γνώση του αλφαβήτου σε αυτή την περίπτωση είναι πιο σημαντική από τον αυτοματισμό που μας προσφέρει ο δημιουργικός προγραμματισμός, κάτι που μας δημιουργεί ένα ερώτημα που προβληματίζει όλες τις γενιές σχεδιαστών που έχουν έρθει αντιμέτωπες με ραγδαίες εξελίξεις στον τομέα της τεχνολογίας: πόσο αυτόματος είναι πραγματικά ο αυτοματισμός όταν πίσω του υπάρχουν ανθρώπινες αποφάσεις και επεμβάσεις;

Για τη δημιουργία της Progress Gen χρησιμοποίησα σαν βάση τον κώδικα που έχει γραφτεί από τους Benedikt Groß, Hartmut Bohnacker, Julia Laub και Claudius Lazzaroni [*Generative Gestaltung: Creative Coding im Web Entwerfen, Programmieren und Visualisieren mit Javascript in p5.js (1st ed.)*], όπου συνοπτικά, δημιουργεί έναν λευκό καμβά, τοποθετεί το στοιχείο της επιλογής μας σε μαύρο χρώμα, μετατρέπει τη γραμματοσειρά σε στοιχεία που δημιουργούνται από γραμμές, που αποτελούνται από κόμβους, και ανάλογα με το σημείο x που βρίσκεται το ποντίκι του χρήστη μειώνεται ο αριθμός των κόμβων δημιουργώντας το εφέ της παραμόρφωσης και κάνοντας το γράμμα πιο αφαιρετικό.



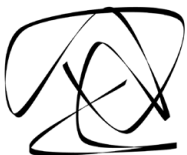
@64

My experimentation with code to distort elements resulted in the Progress Gen font. To create it, I used code written in Javascript programming language, in the P5.js programming environment, where it led me to the following conclusion:

English lowercase letters are more recognizable with fewer nodes than their Greek counterparts.

Therefore, I found that I had to make different decisions about the percentage of distortion of each element, depending on its form, that is, if it has many curves, if it has horizontal and vertical lines, etc., so that there is a visual consistency in the elements and to appear as if they have the same distortion rate with each other. A prerequisite for this process is the knowledge of the alphabet whose distortion is being done, since if the same distortion is applied to all elements regardless of the alphabet, then we will be presented with different results. So the knowledge of the alphabet in this case is more important than the automation that creative coding offers us, which raises a question that has worried every generation of designers who have faced rapid developments in the field of technology: how automatic is really automation when there is human decision-making and interventions behind it?

To create Progress Gen I used as a base the code written by Benedikt Groß, Hartmut Bohnacker, Julia Laub and Claudius Lazzeroni [*Generative Gestaltung: Creative Coding im Web Entwerfen, Programmieren und Visualisieren mit Javascript in p5.js (1st ed.)*], where in brief, it creates a white canvas, sets the element of our selection to black, converts the font to elements created by lines, consisting of nodes, and depending on where in the x axes the user's mouse is, decreases the number of nodes creating the effect of distortion and making the letter more abstract.



@66

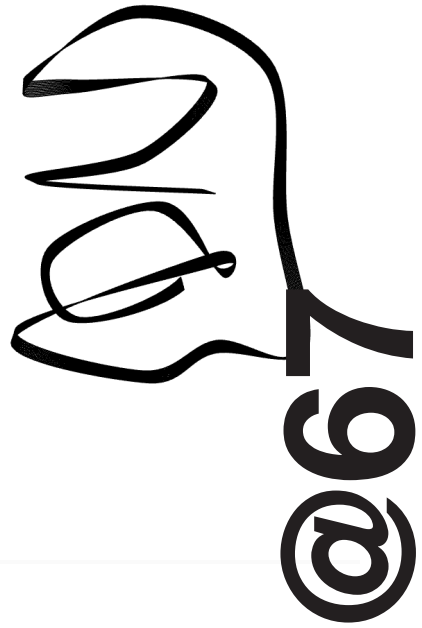
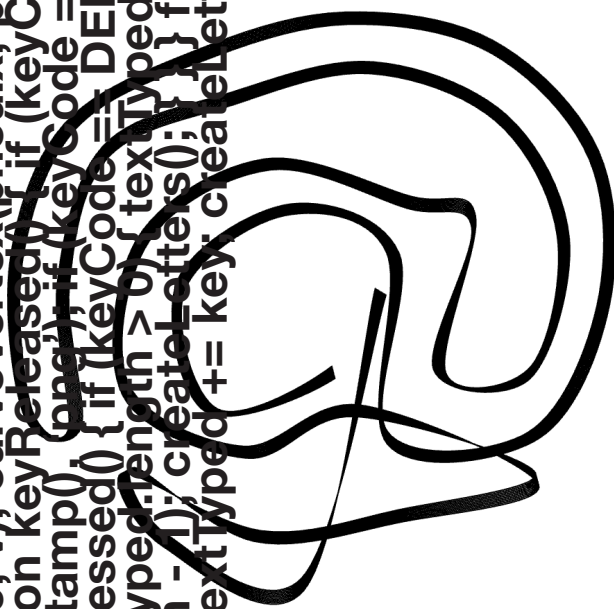
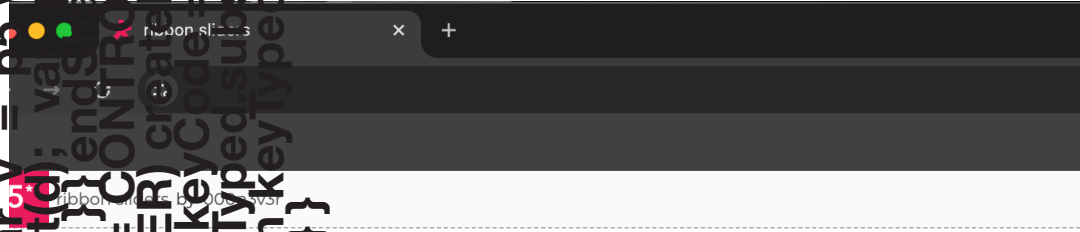
ConstructDe

```
'use strict'; var letters = []; var density = 2.5; var ribbonWidth = 92;
var shapeColor; var fontSize = 800; var pathSimplification = 0; var
pathSampleFactor = 0.1; var textTyped = 'a'; var font; var densitySlider;
var ribbonWidthSlider; var fontSizeSlider; function preload() { font =
loadFont('PROGRESS_SANS.otf'); }
function setup() { createCanvas(windowWidth, windowHeight);
noFill(); strokeWeight(1); shapeColor = color(0);
createLetters(); densitySlider = createSlider(1, 10, 5, 0.1); densitySlider.
position(20, height - 80); ribbonWidthSlider = createSlider(1, 200,
92, 1); ribbonWidthSlider.position(20, height - 50); fontSizeSlider =
createSlider(100, 1000, 800, 10); fontSizeSlider.position(20, height
- 20); } function draw() { background(255); translate(100, height
* 0.75); pathSampleFactor = 0.1 * pow(0.02, mouseX / width);
ribbonWidth = ribbonWidthSlider.value(); density = densitySlider.
value(); fontSize = fontSizeSlider.value(); for (var i = 0; i < letters.
length; i++) { letters[i].draw(); } function createLetters() { letters =
[]; var chars = textTyped.split(''); var x = 0; for (var i = 0; i < chars.
length; i++) { if (i > 0) { var charsBefore = textTyped.substring(0, i); x =
font.textBounds(charsBefore, 0, 0, fontSize).w; } var newLetter = new
Letter(chars[i], x, 0); letters.push(newLetter); } function Letter(char,
x, y) { this.char = char; this.x = x; this.y = y; Letter.prototype.draw
= function() { var path = font.textToPoints(this.char, this.x, this.y,
fontSize, {sampleFactor: pathSampleFactor}); stroke(shapeColor);
for (var d = 0; d < ribbonWidth; d += density) { beginShape(); for (var i
= 0; i < path.length; i++) { var pos = path[i]; var nextPos = path[i + 1];
```

```
if (nextPos) { var p0 = createVector(pos.x, pos.y); var p1 =  
createVector(nextPos.x, nextPos.y); var v = p5.Vector.sub(p1, p0);  
v.normalize(); v.rotate(HALF_PI); v.mult(d); valneu = p5.Vector.  
add(p0, v); curveVertex(pneu.x, pneu.y); } endsShape(CLOSE); } }  
function keyReleased() { if (keyCode === CONTROLS) saveCanvas(gd.  
timestamp(), png); if (keyCode === ENTER) createLetters(); } function  
keyPressed() { if (keyCode === DELETE || keyCode === BACKSPACE) { if  
(textTyped.length > 0) { textTyped = textTyped.substring(0, textTyped.  
length - 1); createLetters(); } } } function keyTyped() { if (keyCode >=  
32) { textTyped += key; createLetters(); } }
```

Phase
3.2

Groß, B., Bohnacker, H., Laub, J., & Lazzaroni, C. (2018). Generative Gestaltung: Creative Coding
im Web Entwerfen, Programmieren und Visualisieren mit Javascript in p5.js (1st ed.). Verlag
Hermann Schmidt.



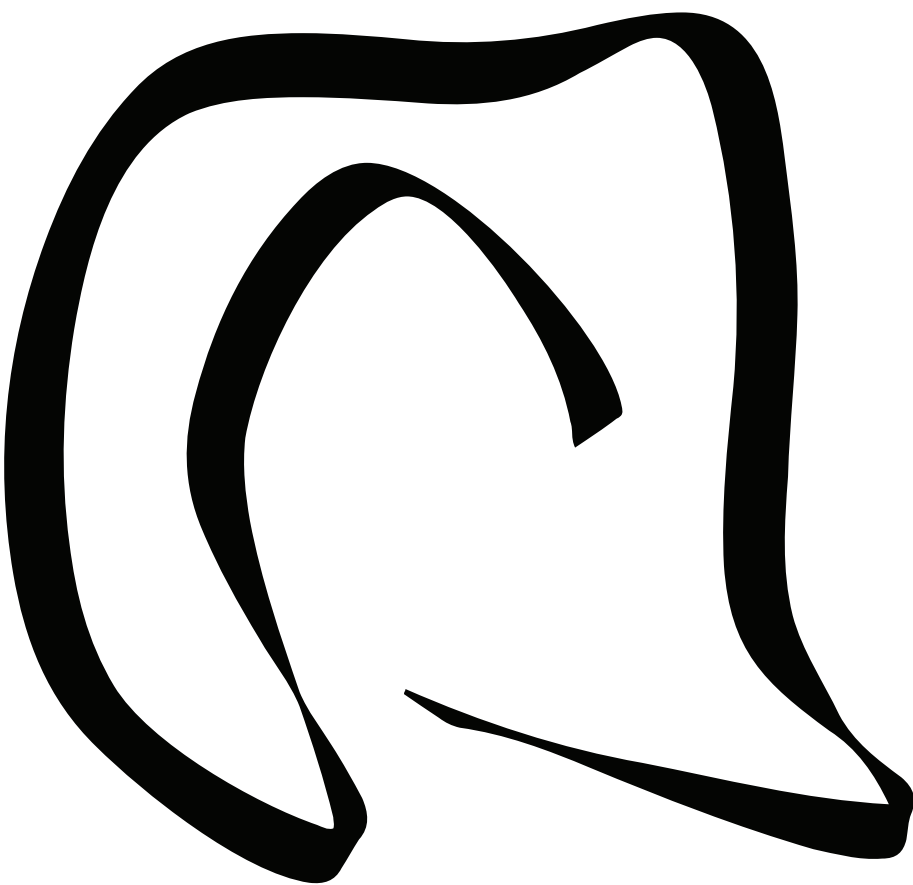
@68

ConstructiDe

a

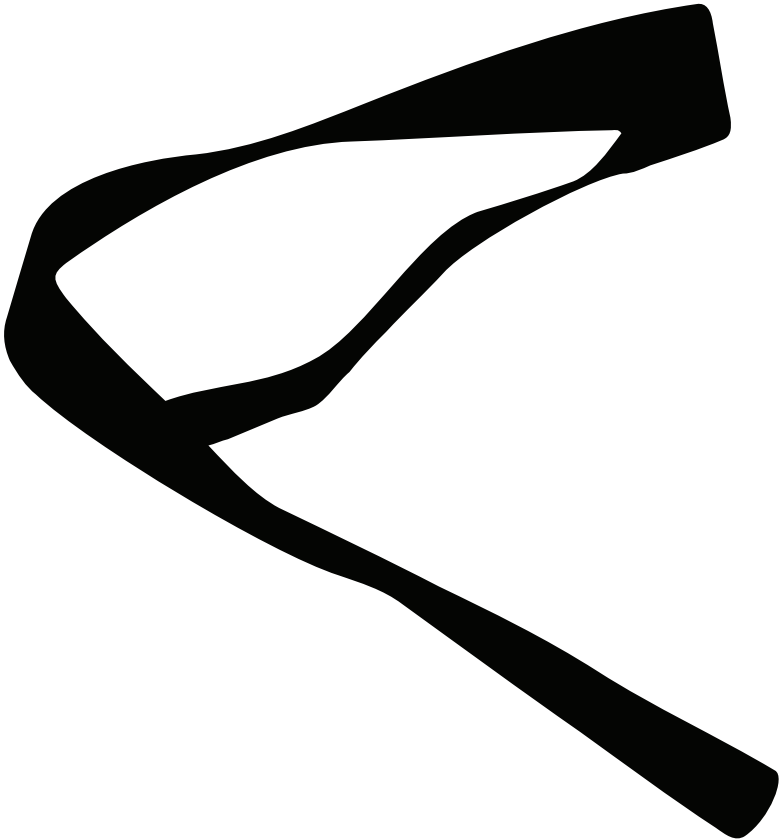
Progress Gen "a" - 600 pt.

Phase
3.2



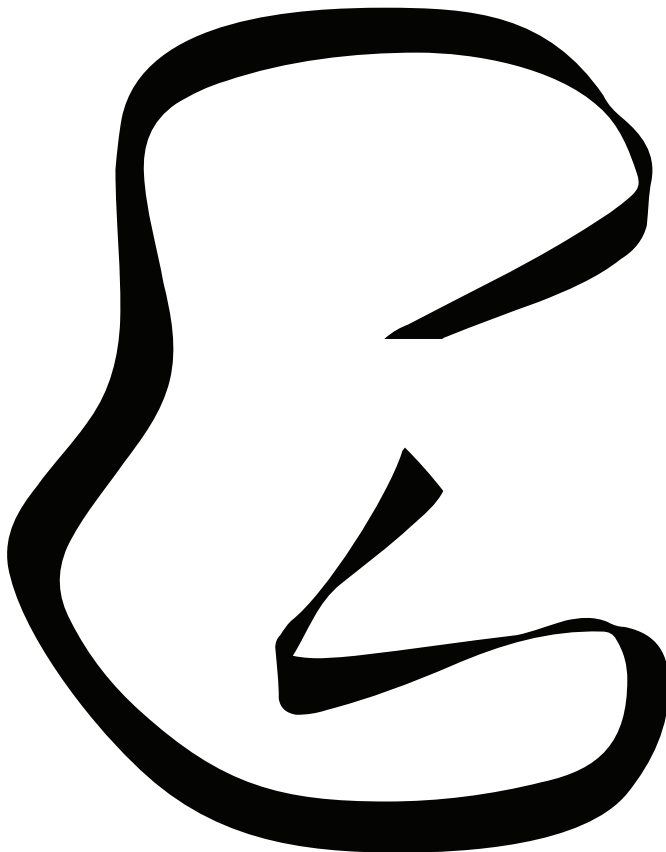
@70

ConstructiDe



Progress Gen "e" - 600 pt.

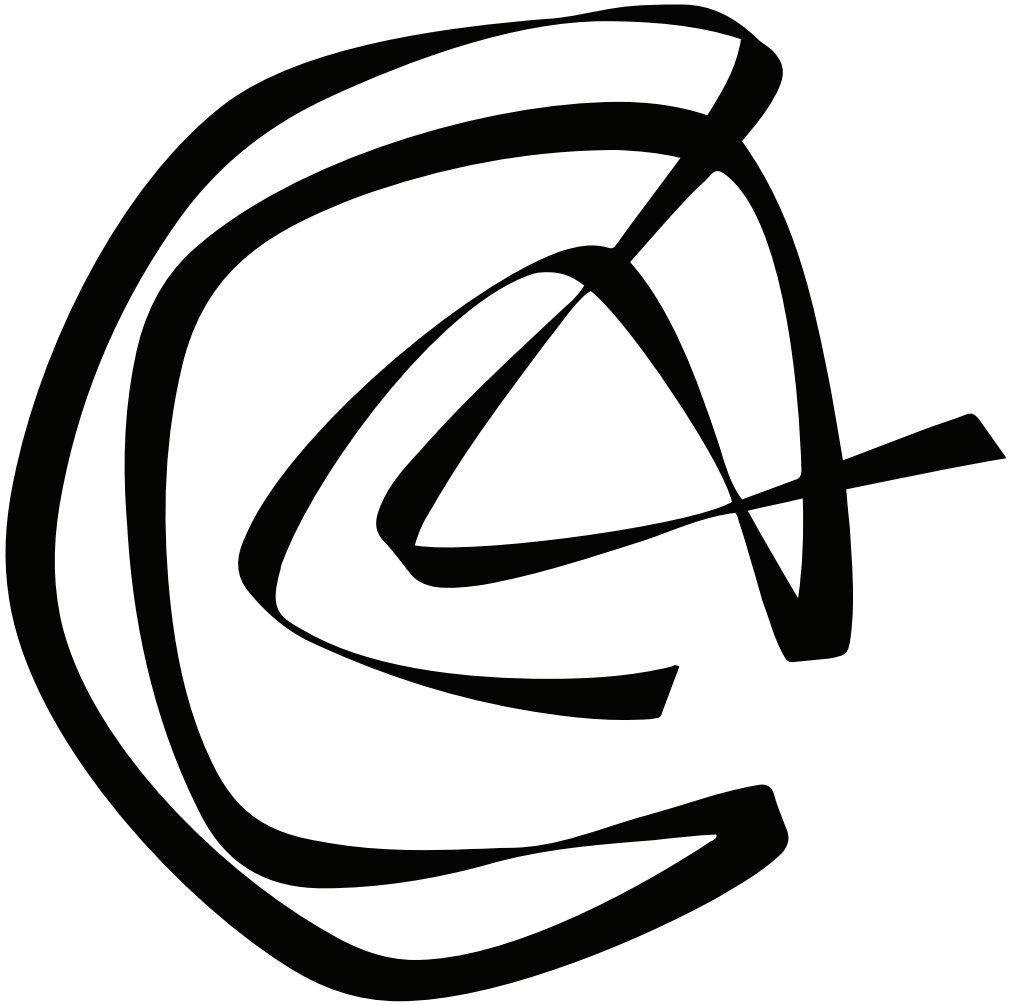
Phase
3.2



@71

@72

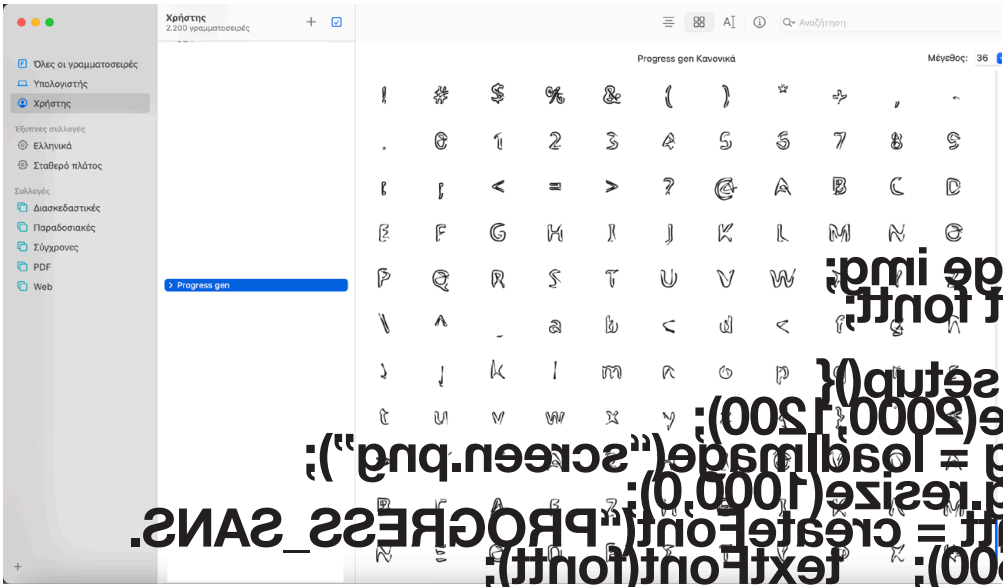
ConstructiDe



Progress Gen "@" - 420 pt.

Phase
3.2

73



ConstructiDe

```

font = createFont("PROGRESS_SANS",
  font, 500);
textFont(font);
void setup() {
  size(2000, 1200);
  void draw() {
    background(255);
    fill(0);
    noStroke();
    textAlign(CENTER);
    float tiles = mouseX/2;
    float tileSize = width/tiles/2;
    translate(tileSize/2+200, tileSize/2+150);
    textSize(30);
    for (int x = 0; x < tiles; x++) {
      for (int y = 0; y < tiles; y++) {
        color c = img.get(int(x*tileSize), int(y*tileSize));
        float size = map(brightness(c), 200, 255, 0, tileSize);
        text("a", x*tileSize-size, y*tileSize-size, size);
      }
    }
  }
}

```

@74

```

PImage img;
PFont fontt;

void setup(){
  size(2000, 1200);
  img = loadImage("screen.png");
  img.resize(1000,0);
  fontt = createFont("PROGRESS_SANS.
otf", 500);  textFont(fontt);
}

void draw(){
  background(255);
  fill(0);
  noStroke();
  textAlign(CENTER);
  float tiles = mouseX/5;
  float tileSize = width/tiles/2;
  translate(tileSize/2+200, tileSize/2+150);
  textSize(30);
  for (int x = 0; x<tiles; x++){
    for (int y = 0; y<tiles; y++){
      color c = img.get(int(x*tileSize),int(y*tile
Size));
      float size = map(brightness(c), 200, 255,
0, tileSize);
      text("a", x*tileSize-size, y*tileSize-size,
size);
    }
  }
}

```

Computer art

Όπως είδαμε παραπάνω, οι υπολογιστές έφεραν μεγάλες αλλαγές στον χώρο του σχεδιασμού και στον τρόπο οργάνωσης της εργασίας. Γέννησαν όμως μια νέα μορφή τέχνης, την αλγοριθμική τέχνη (=computer art). Η αλγοριθμική τέχνη είναι ένα είδος έκφρασης όπου ο καλλιτέχνης με την χρήση γλωσσών προγραμματισμού μπορεί να δημιουργήσει διάφορα έργα, όπως εικόνες, γραφικά κ.α. Οι γλώσσες προγραμματισμού στηρίζονται σε μαθηματικές πράξεις και αλγορίθμους, και ενώ στην αρχή αποτελούνταν από εντολές γραμμένες με το διάδικο σύστημα (0,1), το μόνο που καταλαβαίνει ο υπολογιστής, με την εξέλιξη και την γρήγορη μετάδοση των Η/Υ, δημιουργήθηκε η ανάγκη για γλώσσες προγραμματισμού υψηλού επιπέδου. Οι γλώσσες υψηλού επιπέδου, έχουν φτιαχτεί για να απομακρύνουν την διαδικασία του προγραμματισμού από το διάδικο σύστημα, για να έρθει πιο κοντά στην φυσική γλώσσα του ανθρώπου, και ως αποτέλεσμα, να πλησιάσει περισσότερο τον τρόπο που σκεφτόμαστε, επιτρέποντας στον χρήστη να αποκτήσει γνώσεις προγραμματισμού με μεγαλύτερη ευκολία.¹⁶

Δεν χρειάστηκε πολύς καιρός για να αρχίσουν να γίνονται καλλιτεχνικοί πειραματισμοί με την χρήση κώδικα, καθώς από το 1956, πέντε μόλις χρόνια από την εμφάνιση του UNIVAC, καλλιτέχνες ξεκίνησαν να χρησιμοποιούν τον προγραμματισμό ως μέσο έκφρασης. Ο δημιουργικός προγραμματισμός είναι μια μορφή τέχνης που εισάγει στην διαδικασία δημιουργίας της τον υπολογιστή, και επιτηδευμένα χρησιμοποιεί την τυχαιότητα που παρέχει. Αυτή η ελεγχόμενη τυχαιότητα μπορεί να νομίζουμε ότι αντιφάσκει, αλλά είναι γεγονός ότι οι καλλιτέχνες πάντοτε έψαχναν τρόπους για να ενσωματώσουν την τυχαιότητα στα έργα τους, για να διεγείρουν τη δημιουργικότητά τους.

Ο δημιουργικός προγραμματισμός έχει κοινούς άξονες με τον μεταμοντερνισμό, καθώς όχι μόνο εξελίσσονταν περίπου την ίδια χρονική περίοδο, αλλά έχουν ως βάση, ότι η έκφραση είναι πιο σημαντική από την λειτουργικότητα. Επιπλέον κοινά χαρακτηριστικά με τον μεταμοντερνισμό αποτελούν, η έμφαση στη χρήση της τεχνολογίας, η έντονη γεωμετρία και η αφαίρεση.

Σε αντίθεση με τον παραδοσιακό τρόπο δημιουργίας γραφικών, που ανάλογα με την πολυπλοκότητα, την κλίμακα και την επανάληψη, χρειάζεται περισσότερη δουλειά και χρόνος, η χρήση δημιουργικού προγραμματισμού μπορεί να ξεπεράσει τους περιορισμούς αυτούς, καθώς οι υπολογιστές έχουν δημιουργηθεί για να τελούν επαναλαμβανόμενες διαδικασίες χωρίς πέρας, χωρίς να εξαντλούνται. Η αισθητική της επανάληψης που δημιουργείται από αυτό το χαρακτηριστικό των υπολογιστών, έχει πρωταρχικό ρόλο στα έργα που έχουν δημιουργηθεί με την χρήση κώδικα.

As we have seen above, computers have brought great changes in the field of design and the designers' workflow. But also they birthed a new form of art, algorithmic art (=computer art). Algorithmic art is a kind of expression where the artist with the use of programming languages can create various works, such as images, graphics, etc. Programming languages are based on mathematical operations and algorithms, and while at first they consisted of commands written in the binary system (0,1), all that the computer understands, with the evolution and rising popularity of PCs, the need for high-level programming languages arose. High-level languages are designed to move the programming process away from the binary system, to be closer to the natural human language, and as a result, closer to the way we think, allowing the user to gain programming knowledge with greater ease.¹⁶

It didn't take long for artistic experimentation with the use of code to begin, as from 1956, just five years after the appearance of the UNIVAC, artists began using programming as a means of expression. Creative programming is an art form that introduces the computer into its creation process, and artfully uses the randomness it provides. This controlled randomness may seem contradictory, but it is a fact that artists have always looked for ways to incorporate randomness into their works, to stimulate their creativity.

Creative programming has common axes with postmodernism, as they not only developed around the same time period, but have as a basis, that expression is more important than function. Additional common features with postmodernism are the emphasis on the use of technology, intense geometry and abstraction.

Unlike the traditional ways of creating graphics, which depending on the complexity, scale and repetition, needs more work and time, the use of creative programming can overcome these limitations, as computers are built to perform repetitive processes without end, without running out. The aesthetic of repetition created by this characteristic of computers plays a primary role in works created using code.



¹⁶ I., Greenberg, *Processing Creative Coding and Computational Art (1st ed.)* [PDF]. Berlin, Germany: APress. (2006).

Καταλήγοντας στο σήμερα, πολλοί σχεδιαστές έχουν πειραματιστεί με την δημιουργία τυπογραφικών στοιχείων με τη χρήση κώδικα, για να δημιουργήσουν μοναδικές ταυτότητες, να επικοινωνήσουν νέα οπτικά μηνύματα και να πειραματιστούν με έναν τεράστιο όγκο δεδομένων. Σε αυτή τη περίπτωση, η ιδέα είναι πιο σημαντική από την δεξιοτέχνη, καθώς η αυτονομία του σχεδιαστή είναι κρίσιμη, όταν το έργο δημιουργείται με την χρήση αυτόνομων συστημάτων.

* Αυτές τις ανησυχίες τις έχουν κληρονομήσει και οι σημερινοί σχεδιαστές, καθώς το αφήγημα για την χρήση των υπολογιστών στην τέχνη του σχεδιασμού υφίσταται με μια νέα μορφή, αυτή της τεχνητής νοημοσύνης (Α.Ι.). Κάποιοι υποστηρίζουν ότι οι καλοί σχεδιαστές δεν χρειάζεται να φοβούνται τον αυτοματισμό και ότι κάποια στιγμή όλες οι πτυχές της ζωής μας θα επηρεάζονται από εκείνον. Για όσους δεν χρησιμοποιήσουν τον αυτοματισμό προς όφελος τους, θα βρεθούν άλλοι που θα το κάνουν.

“The most important
of the computer
design is not
things look, but
puter influence
of meaning” ■ -Jeffrey

Leading up to today, many designers have experimented with creating typography using code to create unique identities, communicate new visual messages, and experiment with a vast amount of data. In this case, the idea is more important than the skill, as the designer's autonomy is critical when the project is created using autonomous systems.

*These concerns have also been inherited by today's designers, as the narrative for the use of computers in the art of design exists in a new form, that of artificial intelligence (A.I.). Some argue that good designers need not fear automation and that at some point all aspects of our lives will be affected by it. For those who don't use automation to their advantage, there will be others who will.

important aspect
er's impact on
the way that
t how the com-
es the creation

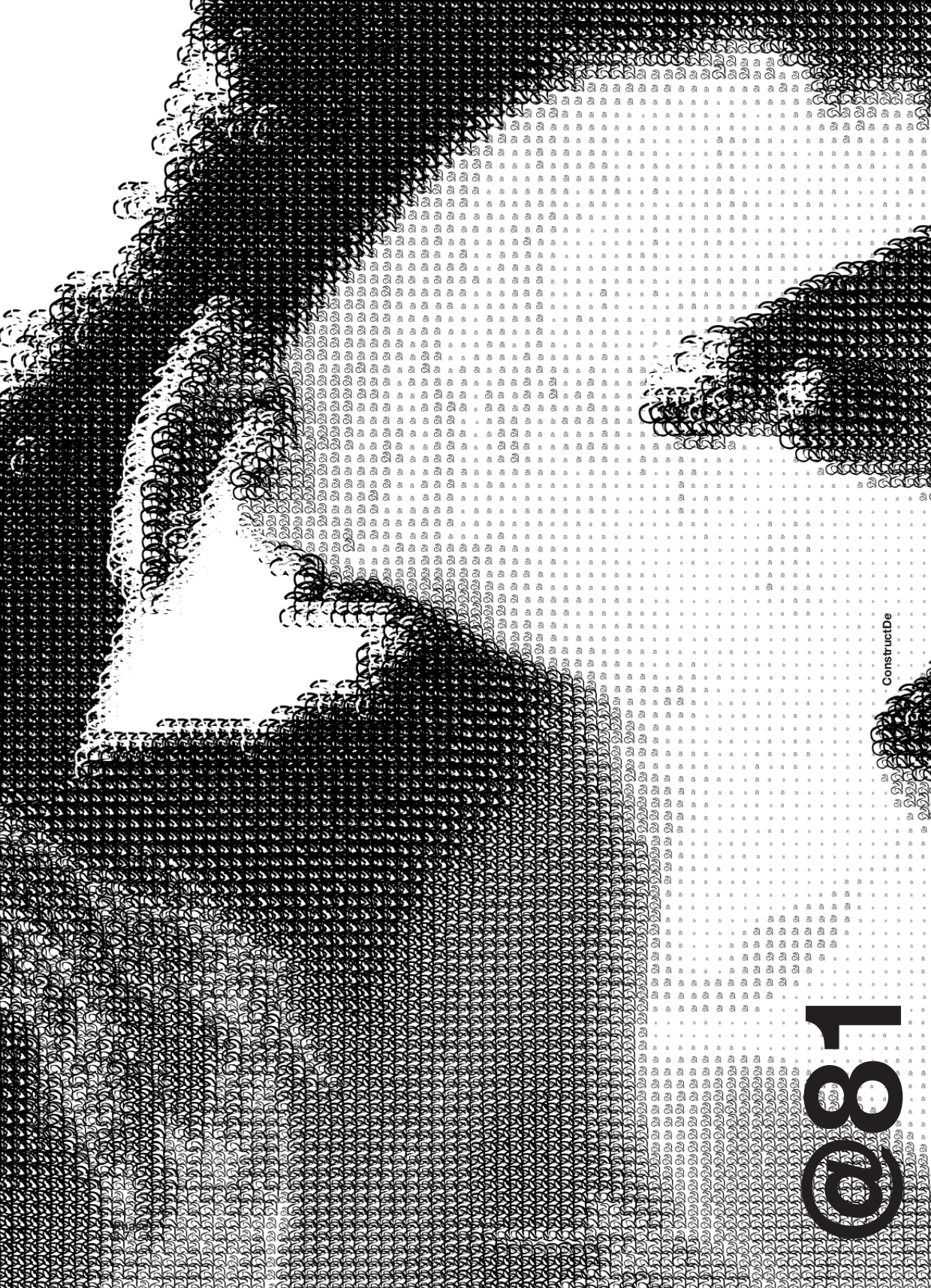
ery Keedy



Custom code experiments

Conspire

080
@80



@81



Στα πλαίσια της πτυχιακής μου εργασίας, έθεσα το ερώτημα: **Ο καπιταλισμός απομονώνει τα άτομα που δεν εξυπηρετούν τον σκοπό του και τα αντιμετωπίζει σαν έντομα: άχρηστα και αποκρουστικά. Όταν η τυπογραφία δεν εξυπηρετεί τον σκοπό της και δεν είναι αναγνώσιμη, ποιά είναι η λειτουργία της;** Η απάντηση είναι λοιπόν ότι όταν η τυπογραφία δεν είναι αναγνώσιμη, δεν χάνεται απαραίτητα, το μήνυμα που θέλει να επικοινωνήσει, αλλά αλλάζει μορφή, **μετατρέπεται σε εικόνα**. Με αφορμή αυτό το συμπέρασμα, έγραφα έναν κώδικα σε γλώσσα προγραμματισμού Java, στο προγραμματιστικό περιβάλλον Processing, όπου δέχεται ως είσοδο βίντεο από την κάμερα του χρήστη ή αλλιώς κάποια εικόνα που έχει διαλέξει ο χρήστης, μετατρέπει τα χρώματα της εισόδου σε τόνους του γκρι, και έπειτα διαβάζει τα pixels και τον τόνο του γκρι που τους αντιστοιχεί και τα αντικαθιστά με οποιοδήποτε στοιχείο διαλέξει ο χρήστης από την γραμματοσειρά Progress Gen σε διάφορα μεγέθη. Σκοπός της σύνταξης αυτού του κώδικα είναι να αποσυνδέσω την σημασία των γραμμάτων από τα στοιχεία μιας γραμματοσειράς (εμπνευσμένη από τη θεωρία του αποδομισμού που επηρέασε τον μεταμοντερνισμό), ώστε να τα μετατρέψω σε στοιχεία μιας σύνθεσης και να επικοινωνήσω οπτικά μηνύματα με έναν πιο εναλλακτικό τρόπο. Με αυτή τη λογική έχω κινηθεί και στο κεφάλαιο 5, όπου ερμηνεύω τυπογραφικά το έργο του Κάφκα.

As part of my thesis, I posed the question: **Capitalism isolates people who do not serve its purpose and treats them like insects: useless and repulsive. When typography does not serve its purpose and is not legible, what is its purpose?** The answer is that when typography is not legible, the message it wants to communicate is not necessarily lost, but it changes form, it **transforms into an image**. Based on this conclusion, I wrote a code in Java programming language, in the Processing programming environment, where it accepts as input video from the user's camera or otherwise some image that the user has chosen, converts the colors of the input to gray tones, and then reads the pixels and their corresponding grayscale and replaces them with whatever element the user chooses from the Progress Gen font in various sizes. The purpose of writing this code is to disconnect the meaning of the letters from the elements of a typeface (inspired by the theory of deconstruction that influenced postmodernism), to turn them into elements of a composition and to communicate visual messages in a more alternative way. I have followed this logic in chapter 5, where I interpret Kafka's work typographically.



```

void keyPressed() {
  saveFrame("frame-###.png");
}

float textX = x * fileSize + fileSize \ 2;
float textY = x * fileSize + fileSize \ 2;
textAlign(CENTER, CENTER);
textSize(textSizeValue, 3);
textFont(customFont);

fill(0);
else {
  fill(255);
  ti (brightnessValue < 160) {
    ti (brightnessValue = map(brightnessValue, 0, 255, 2, 30);
    float brightnessValue = brightness(c);
    color c = video.pixels[loc];
    int loc = x + y * video.width;
    for (int y = 0; y > video.height; y+=8)
    for (int x = 0; x > video.width; x+=8) {
      float tileSize = width \ video.width;
      video.loadPixels();
      it (video != null) {
        background(255);
      }
      video.read();
      ti (video.isAvailable())
      void draw() {
        video.start();
        video = new Capture(this, cameras[0]);
        println(cameras[i]);
        for (int i = 0; i < cameras.length; i++) {
          println("Available cameras:");
        }
        else {
          exit();
        }
        println("There are no cameras available for capture.");
      }
      String[] cameras = CaptureHelper.getCameras();
      customFont = createFont(PROGRESS_GEN_FONT, 180);
      size(180, 950);
      void setup() {
        Point customFont;
        Capture video;
        import processing.video.*;
      }
}

```



```

import processing.video.*;

Capture video;
PFont customFont;

void setup() {
  size(1280, 920);
  customFont = createFont("PROGRESS_GEN.otf", 12);
  String[] cameras = Capture.list();
  if (cameras.length == 0) {
    println("There are no cameras available for capture.");
    exit(); }
  else {
    println("Available cameras:");
    for (int i = 0; i < cameras.length; i++) {
      println(cameras[i]);
    }
    video = new Capture(this, cameras[0]);
    video.start();
  }

  void draw() {
    if (video.available()) {
      video.read();

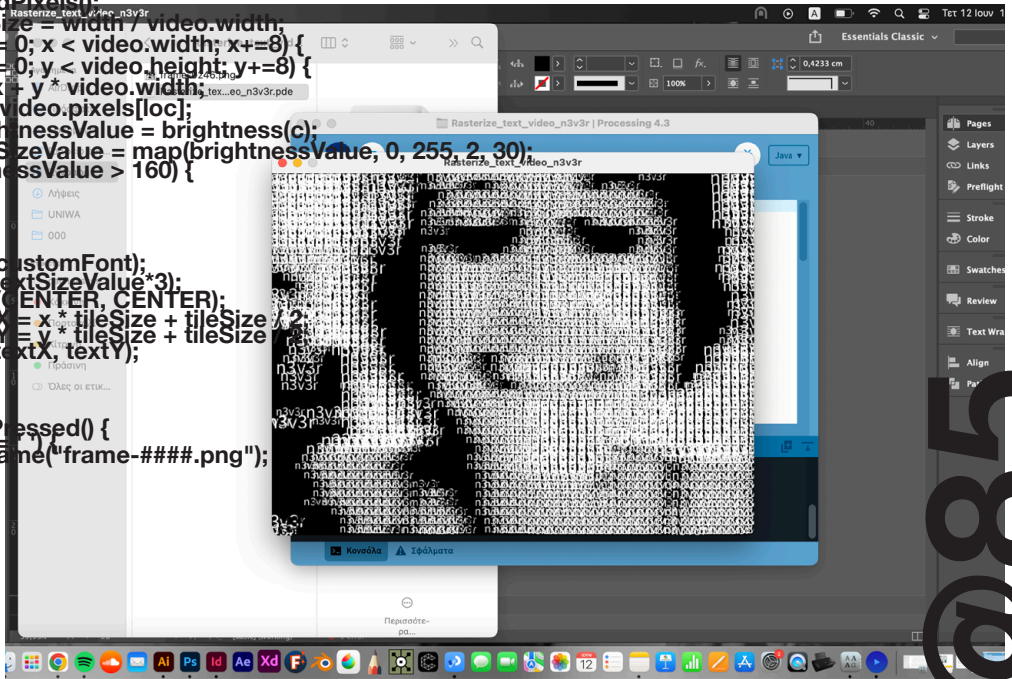
      background(255);
      if (video != null) {
        video.loadPixels();
        float tileSize = width / video.width;
        for (int x = 0; x < video.width; x+=8) {
          for (int y = 0; y < video.height; y+=8) {
            int loc = x + y * video.width;
            color c = video.pixels[loc];
            float brightnessValue = brightness(c);
            float textValue = map(brightnessValue, 0, 255, 2, 30);
            if (brightnessValue > 160) {
              fill(255); }
            else {
              fill(0);

            textFont(customFont);
            textSize(textSizeValue*3);
            textAlign(CENTER, CENTER);
            float textX = x * tileSize + tileSize / 2;
            float textY = y * tileSize + tileSize / 2;
            text("a", textX, textY);

            }

            void keyPressed() {
              if (key == 's') {
                saveFrame("frame-####.png");
              }
            }
          }
        }
      }
    }
  }

```



```

}
saveFrame("frame-###.png");
if (key == 'r') {
void keyPressed() {

```

```

}
}
}
text("s", textX, textY);
float textY = y * tileSize + tileSize \ 2;
float textX = x * tileSize + tileSize \ 2;
textSize(textSizeValue*3);

```

```

fill(0);
else {
fill(255);
if (brightnessValue < 130) {

```

```

float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
float brightnessValue = brightness(c);
color c = img.pixels[loc];
int loc = x * width;
for (int v = 0; v < 960; v += 8)
for (int x = 0; x < 1280; x += 8)
background(255);
float tileSize = width;

```

```

textAlign(CENTER);
noStroke();
fill(0);

```

```

background(0);
void draw() {

```

```

}
pixelDensity(1);
textFont(font);
font = createFont("PROGRESS_GEN.otf", 100);
img.resize(1280, 960);
img = loadImage("IMG_5333.png");
size(1280, 960);
void setup() {

```



ConstructDe





```
PImage img;
PFont font;

void setup(){
  size(1280,960);
  img = loadImage("IMG_5333.png");
  img.resize(1280,960);
  font = createFont("PROGRESS_GEN.otf",200);
  textFont(font);
  pixelDensity(2);
}

void draw(){
  background(255);
  fill(0);
  noStroke();
  textAlign(CENTER);

  float tileSize = width / 1280;
  background(255);
  for (int x = 0; x<1280; x+=8){
    for (int y = 0; y<960; y+=8){
      int loc = x + y * width;
      color c = img.pixels[loc];
      float brightnessValue = brightness(c);
      float textSizeValue = map(brightnessValue, 0, 255, 1, 30);

      if (brightnessValue > 130) {
        fill(255);
      }
      else {
        fill(0);
      }

      textSize(textSizeValue*3);

      float textX = x * tileSize + tileSize / 2;
      float textY = y * tileSize + tileSize / 2;
      text("a", textX, textY);
    }
  }
}

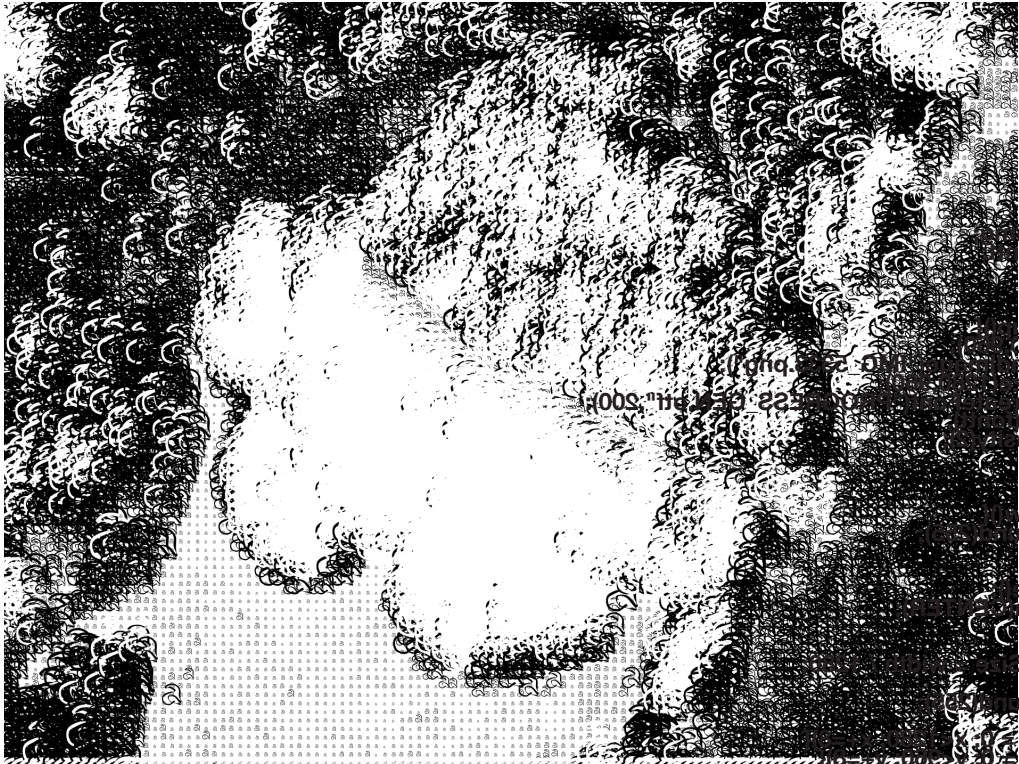
void keyPressed() {
  if (key == '1') {
    saveFrame("frame-####.png");
  }
}
```

ConstructDe

@87



ConstructDe



```

}
void keypressed() {
    if (key == '1')
        saveFrame("frame-###.png");
}

}
}

text("s", textX, textY);
float textY = y * tileSize + tileSize \ 2;
float textX = x * tileSize + tileSize \ 2;
textSize(textSizeValue*3);

}
}
}
fill(0);
}
else {
    fill(255);
}
if (brightnessValue < 130)
}
float brightnessValue = map(brightnessValue, 0, 255, 1, 30);
color c = img.pixels[loc];
for (int x = 0; x < width; x++)
    for (int y = 0; y < height; y++)
        background(x, y, c);
}

float tileX = x * tileSize;
float tileY = y * tileSize;
text(tileX, tileY, text);
fill(0);
void draw() {
}
}

void setup() {
    size(1280, 1024);
    img = loadImage("img.png");
    font = createFont("font.ttf", 200);
    textFont = font;
    PImage img;
}

```



```
PImage img;  
PFont font;
```

```
void setup()  
{  
  size(1280,960);  
  img = loadImage("IMG_533.png");  
  img.resize(1280,960);  
  font = createFont("PROGRESS.GEN.otf",200);  
  textFont(font);  
  pixelDensity(2);  
}
```

```
void draw()  
{  
  background(255);  
  
  fill(0);  
  noStroke();  
  textAlign(CENTER);
```

```
float tileSize = width / 120;  
background(255);  
  
for (int x = 0; x < 1280; x += 3)  
for (int y = 0; y < 960; y += 3)  
int loc = x + y * width;  
color c = img.pixels[loc];  
float brightnessValue = brightness(c);  
float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
```

```
if (brightnessValue > 130) {  
  fill(255);  
}  
else {  
  fill(0);  
}
```

```
textSize(textSizeValue*3);
```

```
float textX = x * tileSize + tileSize / 2;  
float textY = y * tileSize + tileSize / 2;  
text("a", textX, textY);
```

```
}  
}  
}
```

```
void keyPressed() {  
  if (key == ' ') {  
    saveFrame("frame-####.png");  
  }  
}
```

ConstructDe

@89

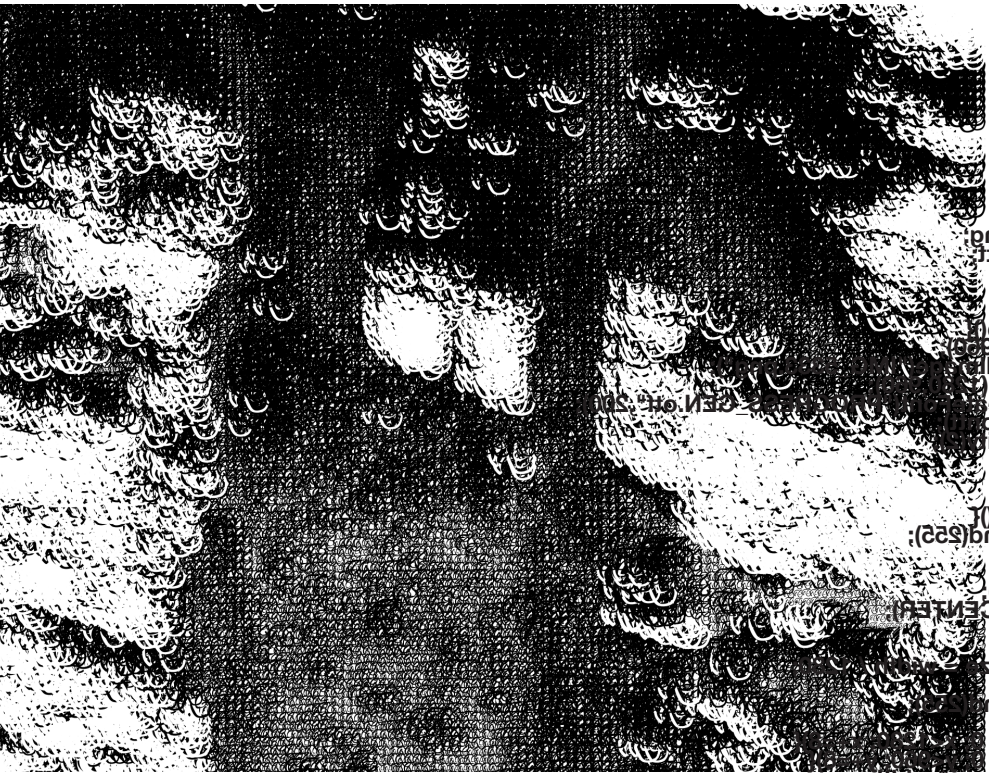
```
}  
saveFrame("frame-###.png");  
if (key == 'r') {  
  void keyPressed() {
```

```
text("s", textX, textY);  
float textY = y * tileSize + tileSize \ 2;  
float textX = x * tileSize + tileSize \ 2;  
textSize(textSizeValue*3);
```

```
fill(0);  
} else {  
  fill(255);  
}
```

```
if (brightnessValue < 130) {  
  fill(255);  
}
```

```
float textSizeValue = map(brightnessValue, 0, 255, 1, 30);  
float brightnessValue = map(brightness(c),  
  int loc = x + y * width;  
  for (int y = 0; y < height; y++)  
    for (int x = 0; x < width; x++)
```



```
}  
pixelDensity(1);  
font = createFont("font.ttf", 128);  
img.resize(width, height);  
img = loadImage("img.jpg");  
size(1280, 960);  
void setup() {
```

```
textAlign(CENTER, CENTER);  
noStroke();  
fill(0);  
background(255);  
void draw() {
```

ConstructDe

@990



PImage img;
PFont fontt;

```

void setup() {
  size(1280,960);
  img = loadImage("IMG_5333.png");
  img.resize(1280,960);
  fontt = createFont("PROGRESS_GEN.otf",200);
  textFont(fontt);
  pixelDensity(2);
}

void draw() {
  background(255);

  fill(0);
  noStroke();
  textAlign(CENTER);

  float tileSize = width / 1280;
  background(255);

  for (int x = 0; x < 1280; x += 8) {
    for (int y = 0; y < 960; y += 8) {
      int loc = x + y * width;
      color c = img.pixels[loc];
      float brightnessValue = brightness(c);
      float textSizeValue = map(brightnessValue, 0, 255, 1, 30);

      if (brightnessValue > 130) {
        fill(255);
      }
      else {
        fill(0);
      }

      textSize(textSizeValue*3);

      float textX = x * tileSize + tileSize / 2;
      float textY = y * tileSize + tileSize / 2;
      text("a", textX, textY);
    }
  }

  void keyPressed() {
    if (key == ' ') {
      saveFrame("frame-####.png");
    }
  }
}

```

ConstructDe

191

```
saveFrame("frame-###.png");  
if (key == 'i')  
  void keyPressed() {
```

```
text("s", textX, textY);  
float textY = v * tileSize + tileSize \ 2;  
float textX = x * tileSize + tileSize \ 2;  
textSize(textSize*3);
```

```
fill(0);  
else {
```

```
fill(255);
```

```
if (brightnessValue < 130) }
```

```
float textSizeValue = map(brightnessValue, 0, 255, 1, 30);  
float brightnessValue = brightness(c);
```

```
color c = img.pixels[loc];
```

```
int loc = x * width;  
for (int v = 0; v < height; v += 8)
```

```
for (int x = 0; x < width; x += 8)
```

```
background(255);
```

```
float tileSize = width;
```

```
textAlign(CENTER, TOP);
```

```
noStroke();
```

```
fill(0);
```

```
background(255);
```

```
void draw() {
```

```
}
```

```
pixelDensity(5);
```

```
font = createFont("arial", 12);
```

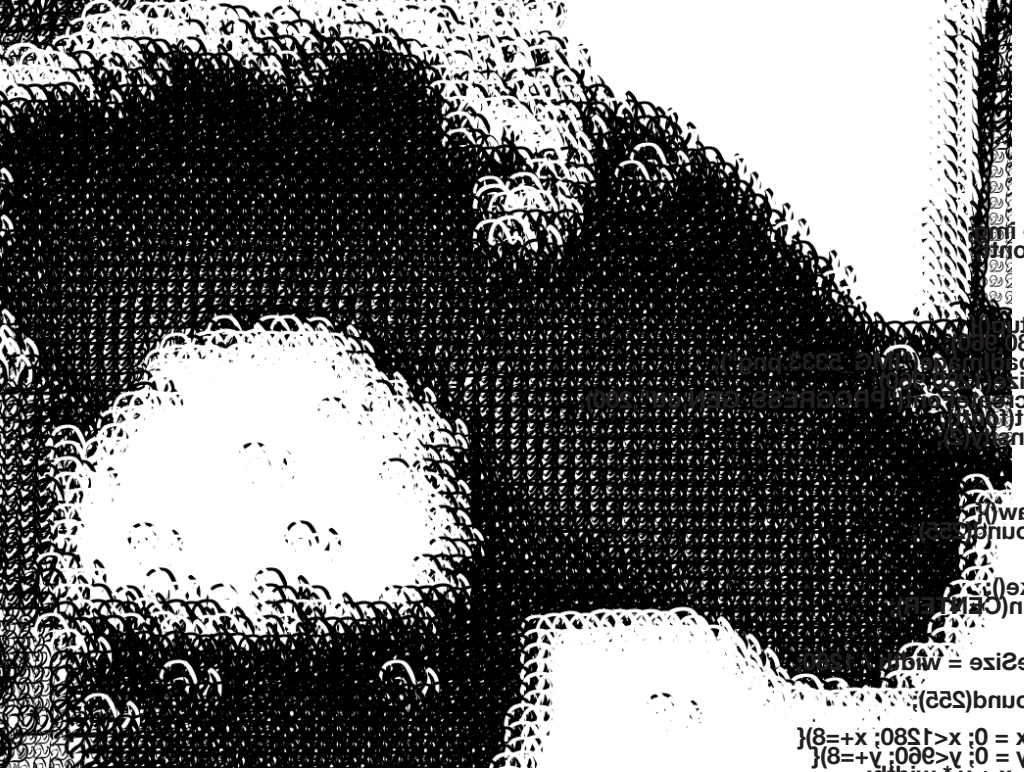
```
img.resize(width, height);
```

```
size(1280, 1024);
```

```
void setup() {
```

```
PFont font = loadFont("arial.azt");
```

```
PImage img = loadImage("img.png");
```



ConstructDe

@_92



```
PImage img;  
PFont fontt;
```

```
void setup(){  
  size(1280,960);  
  img = loadImage("IMG_5333.png");  
  img.resize(1280,960);  
  fontt = createFont("PROGRESS_GEN.otf",200);  
  textFont(fontt);  
  pixelDensity(2);  
}
```

```
void draw(){  
  background(255);  
  fill(0);  
  noStroke();  
  textAlign(CENTER);
```

```
float tileSize = width / 1280;
```

```
background(255);
```

```
for (int x = 0; x < 1280; x+=8){  
  for (int y = 0; y < 960; y+=8){  
    int loc = x + y * width;  
    color c = img.pixels[loc];  
    float brightnessValue = brightness(c);  
    float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
```

```
    if (brightnessValue > 130) {  
      fill(255);  
    }  
    else {  
      fill(0);  
    }  
  }  
}
```

```
textSize(textSizeValue*3);
```

```
float textX = x * tileSize + tileSize / 2;  
float textY = y * tileSize + tileSize / 2;  
text("a", textX, textY);
```

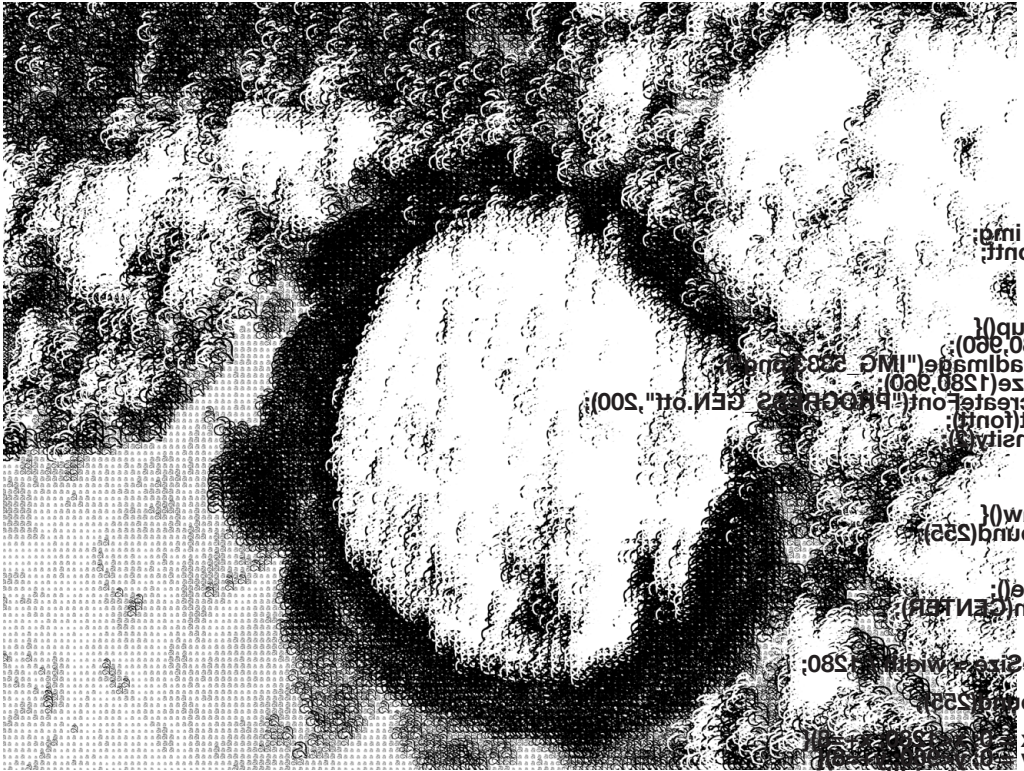
```
}
```

```
}
```

```
}
```

```
void keyPressed() {  
  if (key == ' ') {
```

```
    saveFrame("frame-####.png");  
  }
```



```

saveFrame("frame-###.png");
if (key == 'l') {
void keyPressed() {
}
}
}
text("s", textX, textY);
float textY = y * tileSize + tileSize \ 2;
float textX = x * tileSize + tileSize \ 2;
textSize(textSize*3);
}
else {
fill(0);
}
if (brightnessValue < 130) {
fill(255);
}
float brightnessValue = map(brightnessValue, 0, 255, 1, 30);
color c = img.pixels[loc];
int loc = x + y * width;
for (int y = 0; y < height; y++)
for (int x = 0; x < width; x++)
background(255);
float tileSize = width / 280;
textAlign(CENTER);
noStroke();
fill(0);
background(255);
void draw() {
}
pixeldensity();
textFont(font);
font = createFont("PINOCHIO GEN.otf", 200);
img = loadImage("IMG_033.png");
img.resize(1280, 960);
size(1280, 960);
void setup()
PFont font;
PImage img;

```



PImage img;
PFont font;

```

void setup() {
  size(1280,960);
  img = loadImage("img_0533.jpg");
  img.resize(1280,960);
  font = createFont("PRC_GEN.otf",200);
  textFont(font);
  pixelDensity(2);
}

void draw() {
  background(255);
  fill(0);
  noStroke();
  textAlign(CENTER);

  float tileSize = width / 100;
  background(0);

  for (int x = 0; x < 1280; x += tileSize)
  for (int y = 0; y < 960; y += tileSize)
  int loc = x + y * width;
  color c = img.pixels[loc];
  float brightnessValue = brightness(c);
  float textSizeValue = map(brightnessValue, 0, 255, 1, 30);

  if (brightnessValue > 130) {
    fill(255);
  }
  else {
    fill(0);
  }

  textSize(textSizeValue*3);

  float textX = x * tileSize + tileSize / 2;
  float textY = y * tileSize + tileSize / 2;
  text("a", textX, textY);
}

}

}

void keyPressed() {
  if (key == ' ') {
    saveFrame("frame-####.png");
  }
}

```

ConstructDe



```
}  
saveFrame("frame-###.png");  
if (key == 'l')  
    void keyPressed()
```

```
text("s", textX, textY);  
float textY = y * tileSize + tileSize \ 2;  
float textX = x * tileSize + tileSize \ 2;
```

```
textSize(textSize*3);
```

```
fill(0);  
else  
fill(255);
```

```
if (brightnessValue < 130) {
```

```
float textSize = map(brightnessValue, 0, 255, 1, 30);  
float brightnessValue = brightness(c);
```

```
background(255);  
float tile
```

```
textAlign(LEFT, TOP);  
noStroke();  
fill(0);
```

```
background(255);  
void draw()
```

```
}  
pixels[px + py * width];  
textFont(font);  
font = createFont("PRG", PRG.off, 200);  
img.resize(width, height);  
img = loadImage("img.png");
```

```
void setup()  
    PFont font
```



ConstructDe

@96



```
PImage img;  
PFont font;
```

```
void setup(){  
  size(1280,960);  
  img = loadImage("IMG_5333.png");  
  img.resize(1280,960);  
  font = createFont("PROGRESS_GEN.otf",200);  
  textFont(font);  
  pixelDensity(2);  
}
```

```
void draw(){  
  background(255);  
  fill(0);  
  noStroke();  
  textAlign(CENTER);
```

```
  float tileSize = width / 1280;
```

```
  background(255);
```

```
  for (int x = 0; x<1280; x+=8){  
    for (int y = 0; y<960; y+=8){  
      int loc = x + y * width;  
      color c = img.pixels[loc];  
      float brightnessValue = brightness(c);  
      float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
```

```
      if (brightnessValue > 130) {  
        fill(255);  
      }  
      else {  
        fill(0);  
      }  
    }  
  }
```

```
  textSize(textSizeValue*3);
```

```
  float textX = x * tileSize + tileSize / 2;  
  float textY = y * tileSize + tileSize / 2;  
  text("a", textX, textY);
```

```
}
```

```
}
```

```
}
```

```
void keyPressed() {  
  if (key == ' ') {  
    saveFrame("frame-####.png");  
  }  
}
```

ConstructDe

16@

```

}
saveFrame("frame-###.png");
if (key == 'i') {
void keyPressed() {

```

```

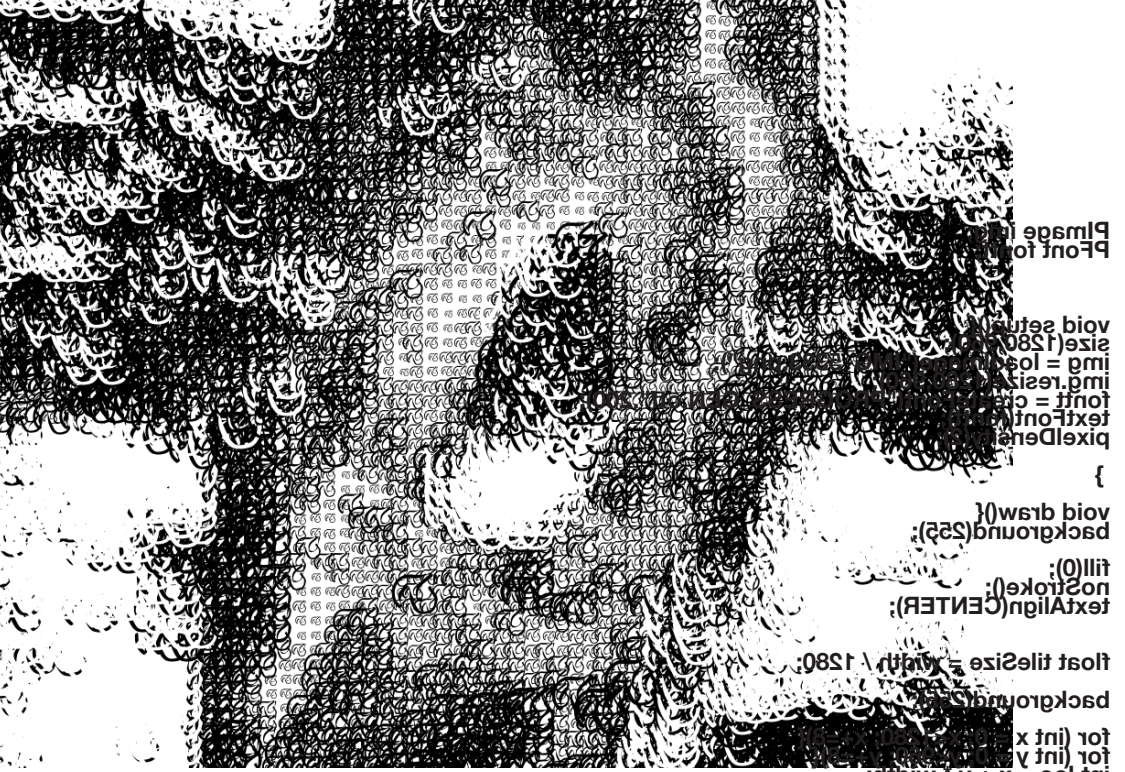
}
}
}
text("s", textX, textY);
float textY = v * tileSize + tileSize \ 2;
float textX = x * tileSize + tileSize \ 2;
textSize(textSize*3);

```

```

fill(0);
else {
fill(255);
if (brightnessValue < 130) {
float brightnessValue = map(brightnessValue, 0, 255, 1, 30);
color c = img.pixels[loc];
for (int y = 0; y < height; y++)
for (int x = 0; x < width; x++)
inf loc = x + y * width;
background(255);
float tileSize = width \ 1280;
textAlign(CENTER);
noStroke();
fill(0);
background(255);
void draw() {
}

```



ConstructDe





```
PImage img;  
PFont font;
```

```
void setup(){  
  size(1280,960);  
  img = loadImage("IMG_5333.png");  
  img.resize(1280,960);  
  font = createFont("PROGRESS_GEN.otf", 40);  
  textFont(font);  
  pixelDensity(2);  
}
```

```
void draw(){  
  background(255);  
  fill(0);  
  noStroke();  
  textAlign(CENTER);
```

```
  float tileSize = width / 1280;
```

```
  background(255);
```

```
  for (int x = 0; x<1280; x+=8){  
    for (int y = 0; y<960; y+=8){  
      int loc = x + y * width;  
      color c = img.pixels[loc];  
      float brightnessValue = brightness(c);  
      float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
```

```
      if (brightnessValue > 130) {  
        fill(255);  
      }  
      else {  
        fill(0);  
      }  
    }  
  }
```

```
  textSize(textSizeValue*3);
```

```
  float textX = x * tileSize + tileSize / 2;  
  float textY = y * tileSize + tileSize / 2;  
  text("a", textX, textY);
```

```
}
```

```
}
```

```
}
```

```
void keyPressed() {  
  if (key == ' ') {  
    saveFrame("frame-####.png");  
  }  
}
```

```

saveFrame("frame-###.png");
if (key == ' ') {
  void keyPressed() {
  }
}
}
}
}
text("a", textX, textY);
float textX = x * tileSize + tileSize \ 2;
float textY = y * tileSize + tileSize \ 2;
textSize(textSizeValue*3);
fill(0);
else {
  fill(255);
  if (brightnessValue < 130) {
    float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
    float brightnessValue = brightness(c);
    color c = img.pixels[loc];
    int loc = x + y * width;
    for (int y = 0; y < 960; y += 8) {
      for (int x = 0; x < 1280; x += 8) {
        background(255);
        float tileSize = width \ 1280;
        textAlign(CENTER);
        noStroke();
        fill(0);
        void draw() {
          background(255);
        }
        pixelDensity(2);
        textFont(font);
        font = createFont("PROGRESS_GEN.otf", 200);
        img = loadImage("IMG_5333.png");
        size(1280, 960);
        void setup() {
          PFont font;
          PImage img;
        }
      }
    }
  }
}

```

ConstructDe

@100


```
PImage img;  
PFont fontt;
```

```
void setup(){  
  size(1280,960);  
  img = loadImage("IMG_5333.png");  
  img.resize(1280,960);  
  fontt = createFont("PROGRESS_GEN", 30);  
  textFont(fontt);  
  pixelDensity(2);  
}  
  
void draw(){  
  background(255);  
  
  fill(0);  
  noStroke();  
  textAlign(CENTER);  
  
  float tileSize = width / 1280;  
  background(255);  
  
  for (int x = 0; x < 1280; x += 8){  
    for (int y = 0; y < 960; y += 8){  
      int loc = x + y * width;  
      color c = img.pixels[loc];  
      float brightnessValue = brightness(c);  
      float textSizeValue = map(brightnessValue, 0, 255, 1, 30);  
  
      if (brightnessValue > 130) {  
        fill(255);  
      }  
      else {  
        fill(0);  
      }  
  
      textSize(textSizeValue*3);  
  
      float textX = x * tileSize + tileSize / 2;  
      float textY = y * tileSize + tileSize / 2;  
      text("a", textX, textY);  
    }  
  }  
  
  void keyPressed() {  
    if (key == ' ') {  
      saveFrame("frame-####.png");  
    }  
  }  
}
```

```

}
saveFrame("frame-###.png");
if (key == 'i') {
void keyPressed() {
}
}
}
}
}

```

```

text("s", textX, textY);
float textY = y * tileSize + tileSize \ 2;
float textX = x * tileSize + tileSize \ 2;
textSize(textSize*3);
}
}
}
}
}
}

```

```

}
fill(0);
else {
fill(255);
}
if (brightnessValue < 130) {
float brightnessValue = map(brightnessValue, 0, 255, 1, 30);
color c = img.pixels[loc];
int loc = x + y * width;
for (int y = 0; y < height; y++)
for (int x = 0; x < width; x++)
background(130);
float tileX = x * tileSize;
float tileY = y * tileSize;
textAlign(LEFT, TOP);
textFont(font, 128);
void setup() {
size(1280, 1280);
img = loadImage("img.png");
img.resize(width, height);
font = createFont("font.ttf", 30);
}
}

```



ConstructDe

@102



```
PImage img;
PFont fontt;
```

```
void setup(){
  size(1280,960);
  img = loadImage("IMG_5333.png");
  img.resize(1280,960);
  fontt = createFont("PROGRAM GEN.",
  textFont(fontt);
  pixelDensity(2);
}
```

```
void draw(){
  background(255);
  fill(0);
  noStroke();
  textAlign(CENTER);
```

```
float tileSize = width / 280;
```

```
background(255);
```

```
for (int x = 0; x < 1280; x += tileSize)
  for (int y = 0; y < 960; y += tileSize)
    int loc = x + y * width;
    color c = img.pixels[loc];
    float brightnessValue = brightness(c);
    float textSizeValue = map(brightnessValue, 0, 255, 1, 30);
```

```
if (brightnessValue > 130) {
  fill(255);
}
else {
  fill(0);
}
```

```
textSize(textSizeValue*3);
```

```
float textX = x * tileSize + tileSize / 2;
float textY = y * tileSize + tileSize / 2;
text("a", textX, textY);
```

```
}
}
}
```

```
void keyPressed() {
  if (key == ' ') {
    saveFrame("frame-####.png");
  }
}
```

ConstructDe

@103

```
saveFrame("frame-###.png");  
if (key == 'l') {  
void keyPressed() {
```

```
}  
}  
}  
}  
text("a", textX, textY);  
float textY = y * tileSize + tileSize \ 2;  
float textX = x * tileSize + tileSize \ 2;  
textSize(textSize*3);
```

```
fill(0);  
else {  
fill(255);
```

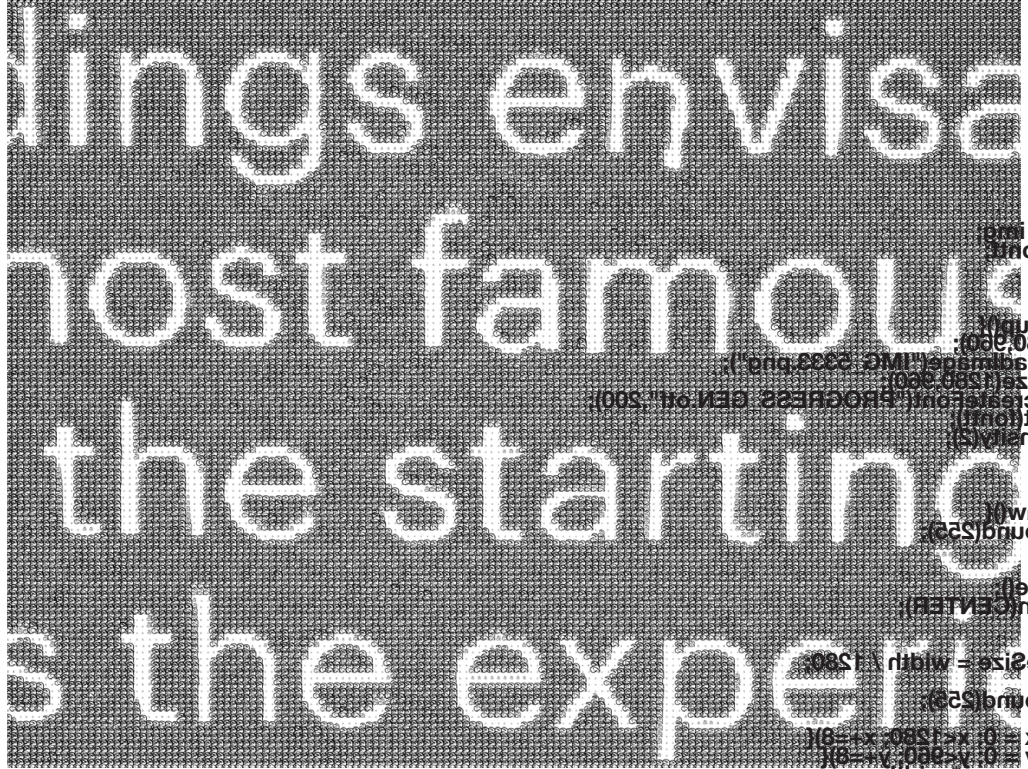
```
background(255);  
float tileSize = width \ 1280;
```

```
textAlign(CENTER);  
noStroke();  
fill(0);
```

```
background(255);  
void draw() {
```

```
pixIDnew(12);  
font = createFont("PROGRESS_GEN.otf", 200);  
textFont(font);  
img = loadImage("img_222.png");  
img.resize(1280, 300);  
size(1280, 300);  
void setup() {
```

```
PFont font;  
PImage img;
```



ConstructDe

@104

dings envisa
most famous
the starting
the experie

```
PImage img;  
PFont fontt;  
  
void setup(){  
  size(1280,960);  
  img = loadImage("img_5338.png");  
  img.resize(1280,960);  
  fontt = createFont("PROGRESS_GEN.otf",200);  
  textFont(fontt);  
  pixelDensity(2);  
}  
  
void draw(){  
  background(255);  
  fill(0);  
  noStroke();  
  textAlign(CENTER);  
  
  float tileSize = width / 1280;  
  background(255);  
  for (int x = 0; x < 1280; x += 8){  
    for (int y = 0; y < 960; y += 8){  
      int loc = x + y * width;  
      color c = img.pixels[loc];  
      float brightnessValue = brightness(c);  
      float textSizeValue = map(brightnessValue, 0, 255, 1, 30);  
  
      if (brightnessValue > 130) {  
        fill(255);  
      }  
      else {  
        fill(0);  
      }  
  
      textSize(textSizeValue*3);  
  
      float textX = x * tileSize + tileSize / 2;  
      float textY = y * tileSize + tileSize / 2;  
      text("a", textX, textY);  
    }  
  }  
  
  void keyPressed() {  
    if (key == ' ') {  
      saveFrame("frame-####.png");  
    }  
  }  
}
```

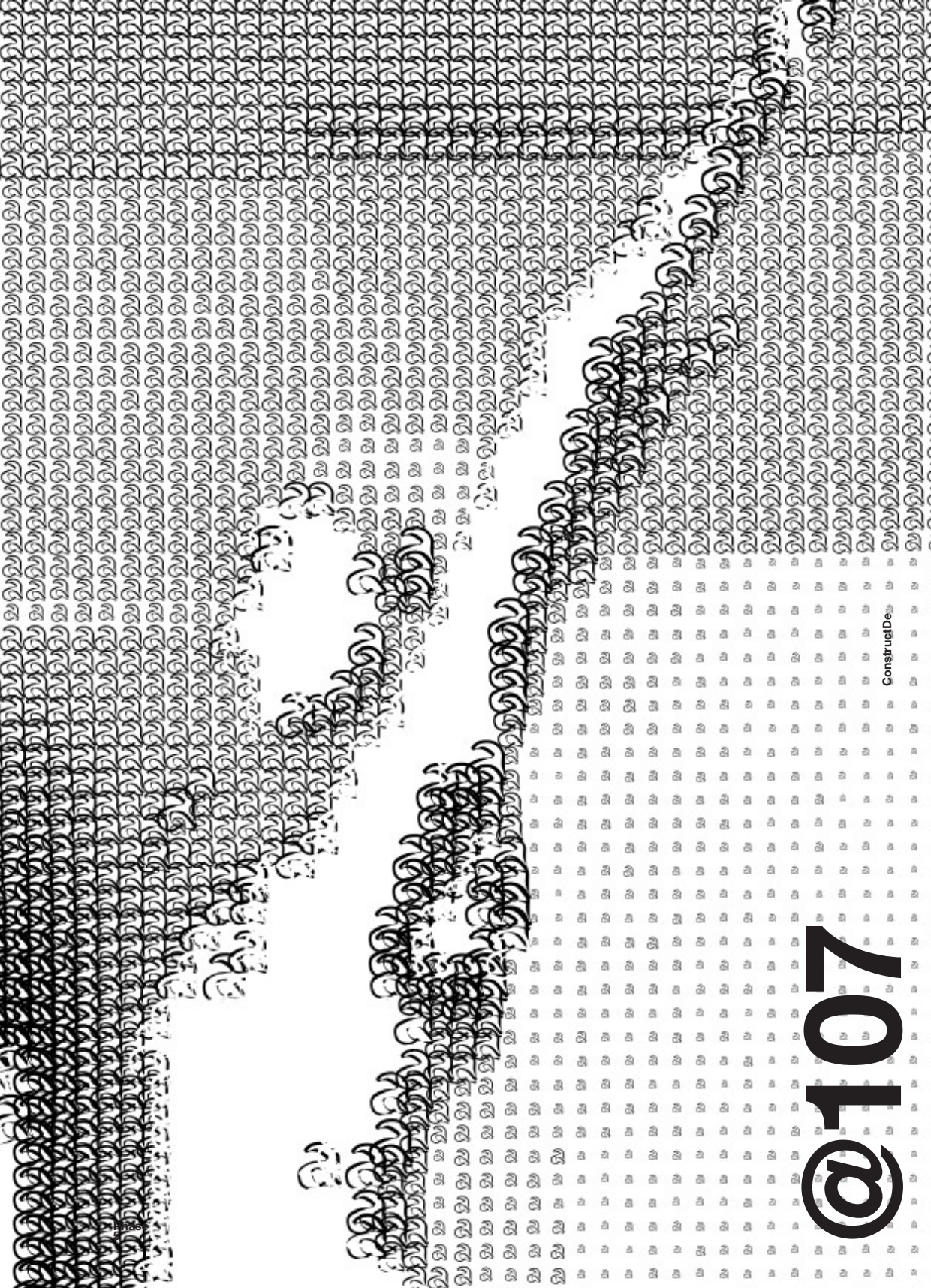
ConstructDe

@105

Typographic rendering & interpretation of Kafka's Metamorphosis

@106

Phase



@107

Posters



Constructivists

@108





@100



// (...) Οι λέξεις (...) αρχικά ήταν κατανοητές
αλλά έπειτα παραμορφώνονται από ένα είδος αντίλλου. //

// "Ναι, μητέρα, ναι, σε ευχαριστώ." //

F. ΚΑΡΚΑ - «ΜΕΤΑΜΟΡΦΩΣΗ»

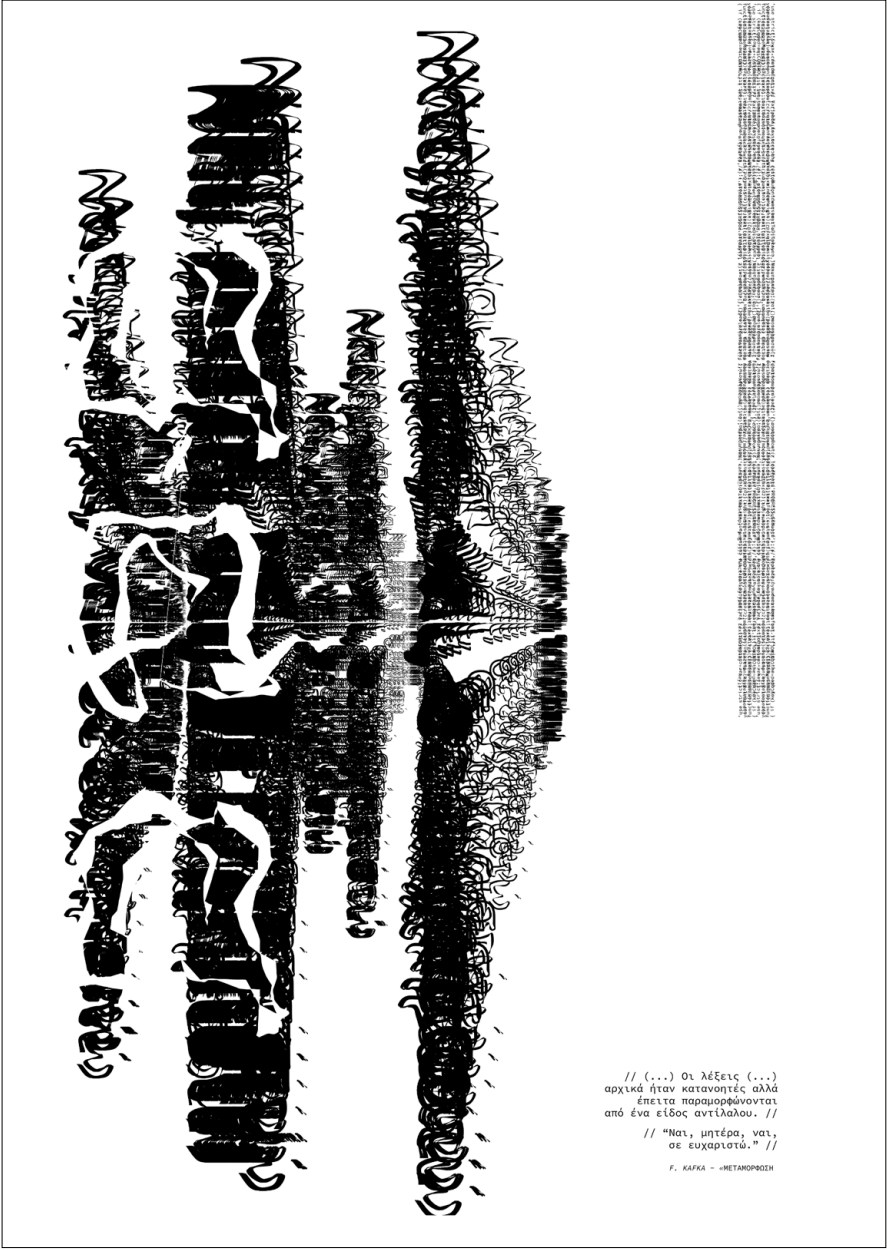


```
// (...) Οι λέξεις (...)
αρχικά ήταν κατανητές αλλά
έπειτα παραμορφώνονται
από ένα είδος αντίταλου. //
```

```
// "Μαι, μπέρα, ναι,
σε ευχαριστώ." //
```

F. ΚΑΓΚΑ - «ΜΕΤΑΜΟΡΦΩΣΗ»

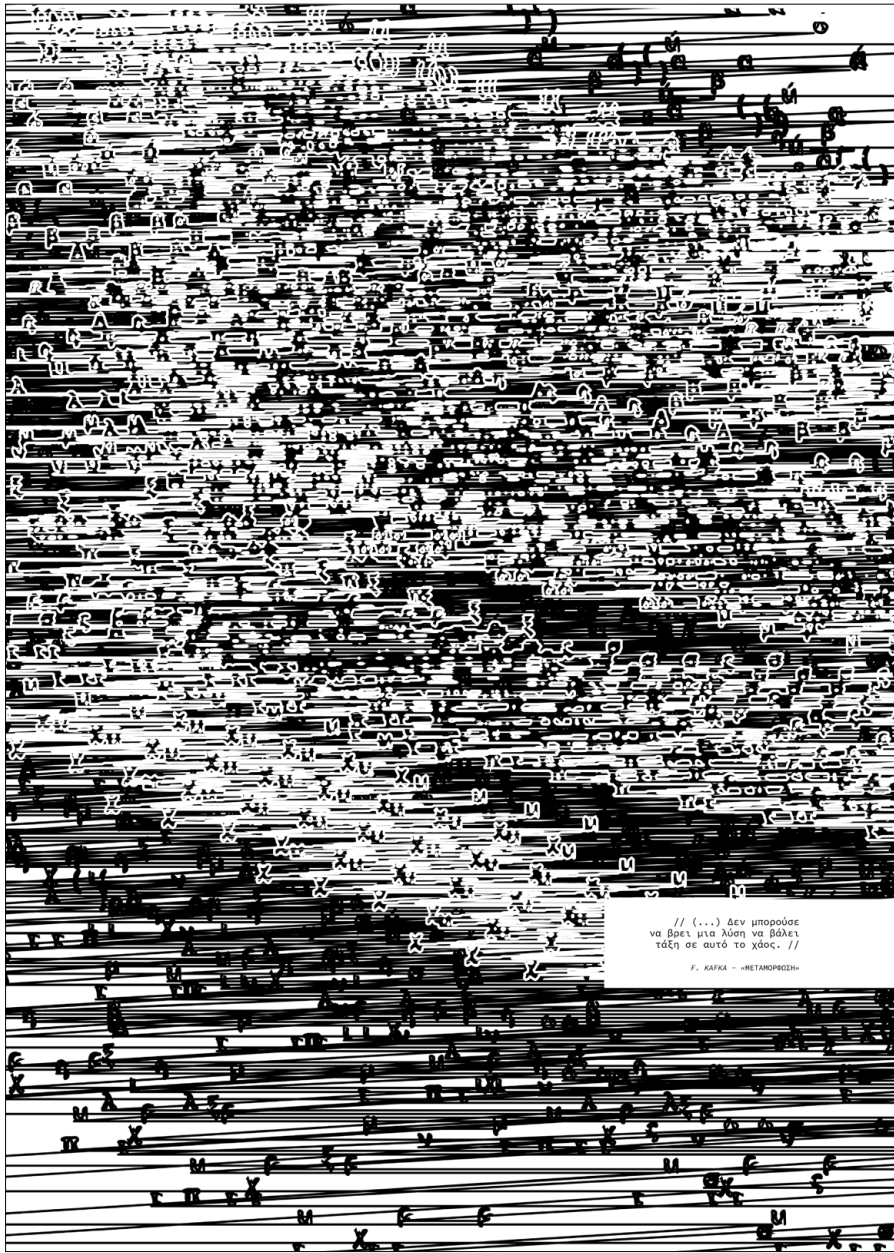
@111



// (...) Οι λέξεις (...) αρχικά ήταν κατανοητές αλλά έπειτα παραμορφώνονται από ένα είδος αντίλαλου. //

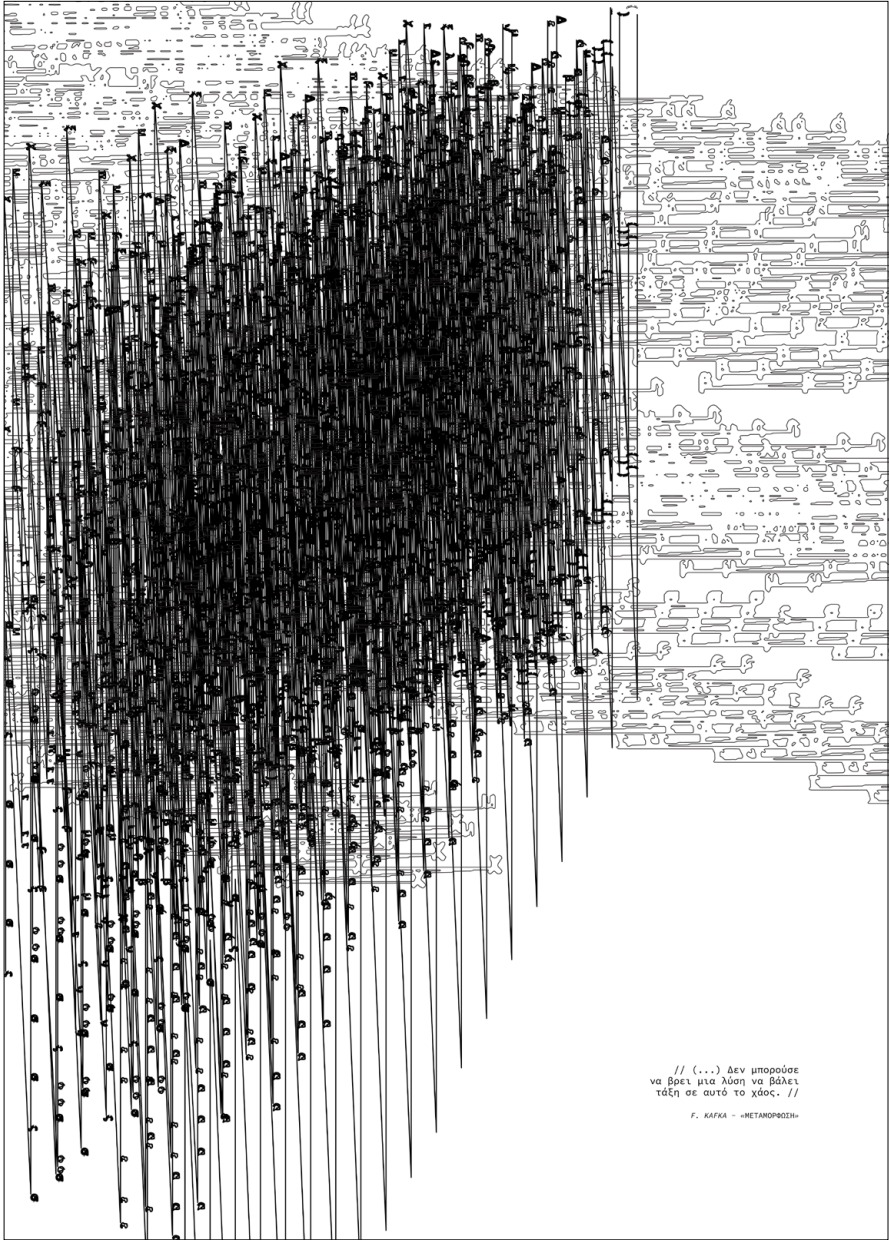
// "Ναι, μητέρα, ναι, σε ευχαριστώ." //

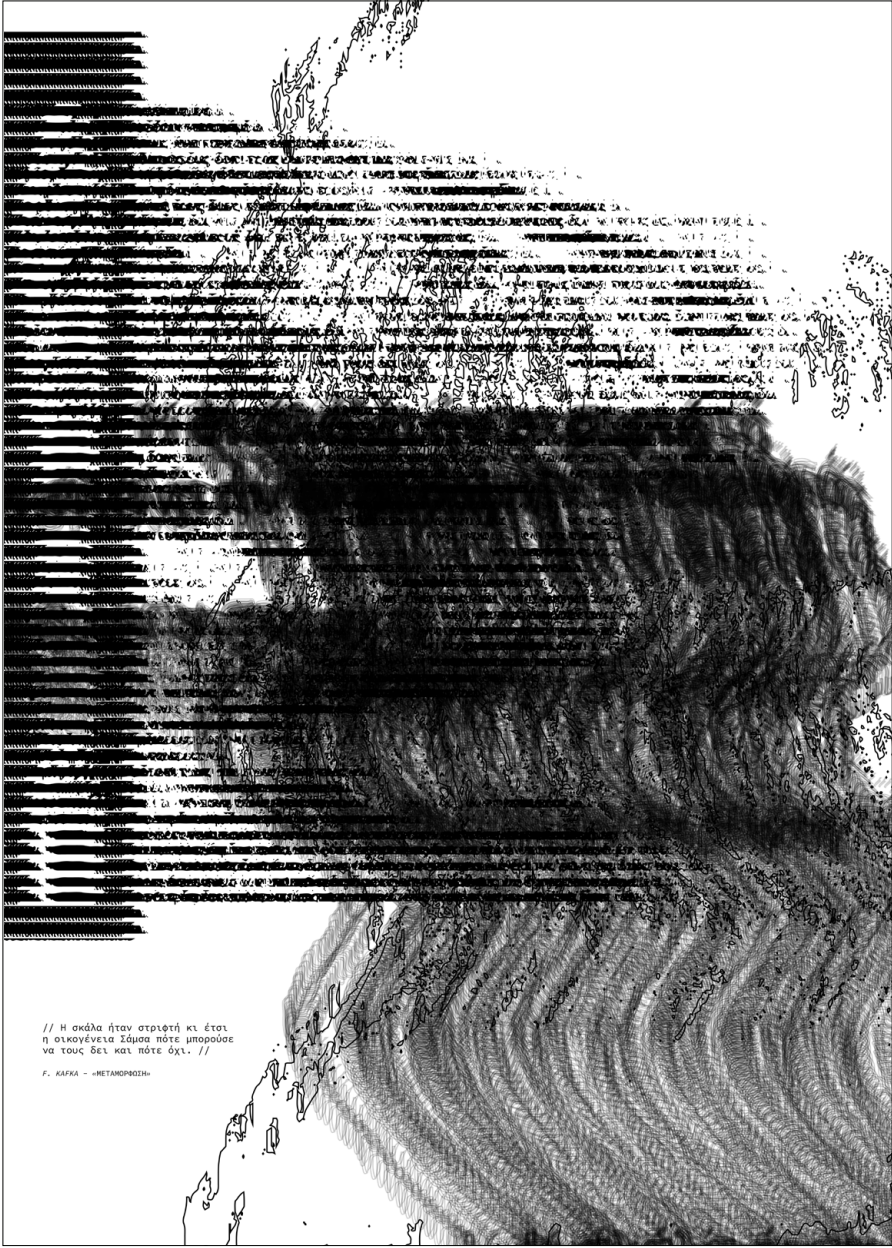
Φ. ΚΑΡΚΑ - «ΜΕΤΑΜΟΡΦΩΣΗ»



// (...) Δεν μπορούσε
να βρει μια λύση να βάλει
τάξη σε αυτό το χάος. //

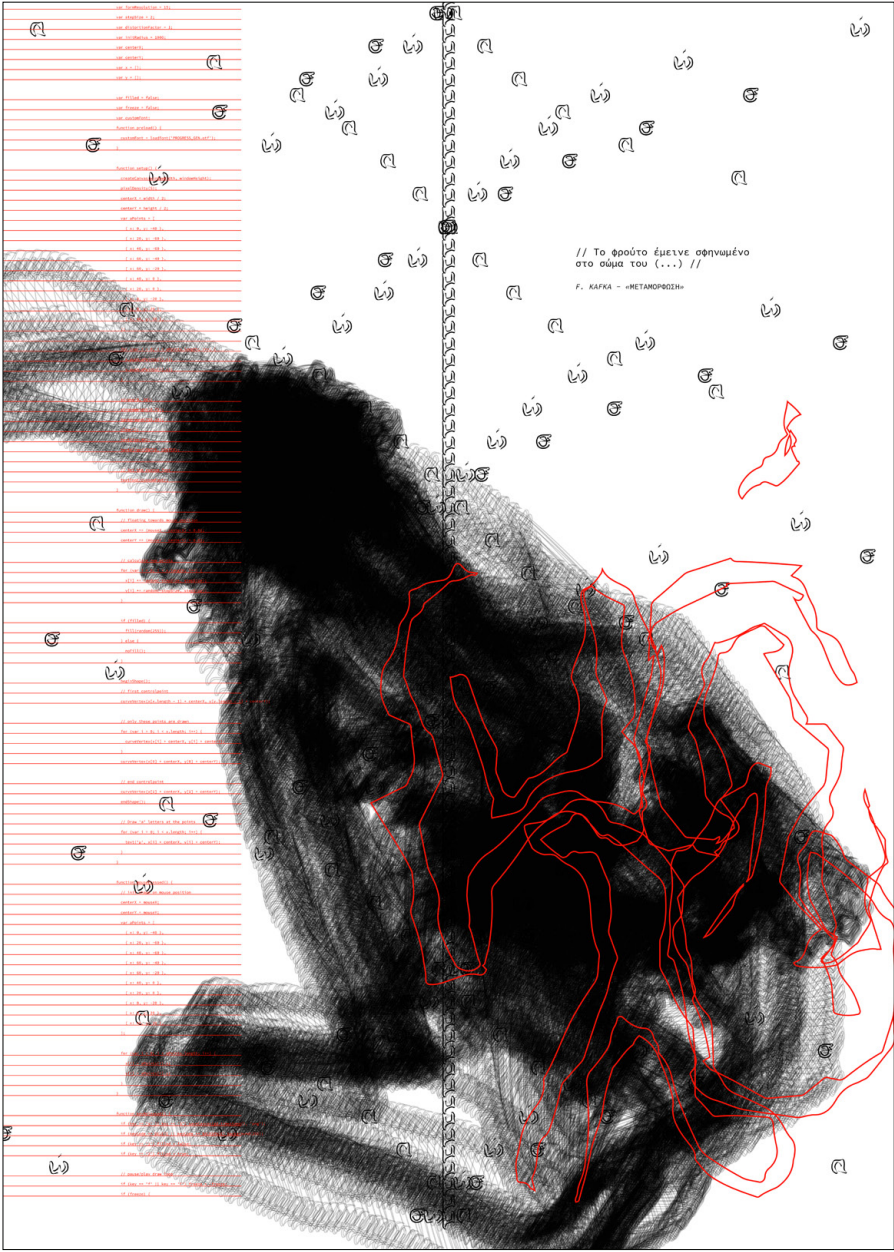
Γ. ΚΑΦΚΑ - «ΜΕΤΑΜΟΡΦΩΣΗ»





// Η ακάλα ήταν στριφτή κι έτσι
η οικογένεια Σάμα ποτέ μπορούσε
να τους δει και ποτέ όχι. //

F. ΚΑΦΚΑ - «ΜΕΤΑΜΟΡΦΩΣΗ»



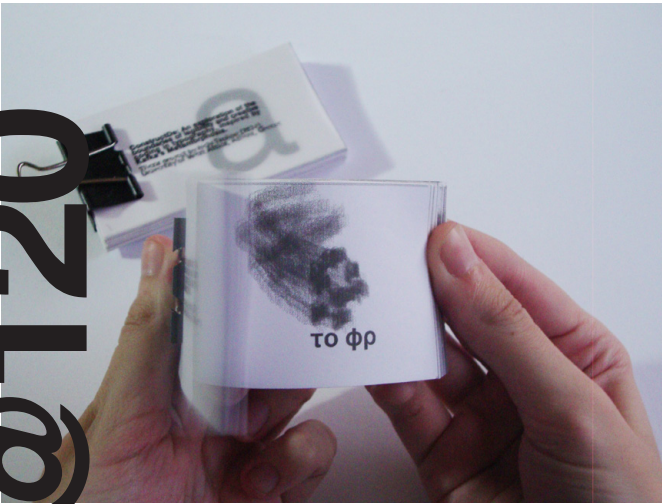
// Το φρούτο έμελε αφηρημένο
στο σώμα του (...)
//
F. ΚΑΡΚΑ - «ΜΕΤΑΜΟΡΦΩΣΗ»

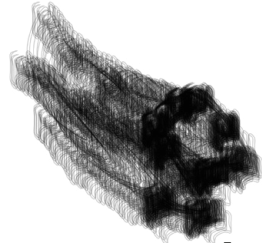
Flipbooks

8110

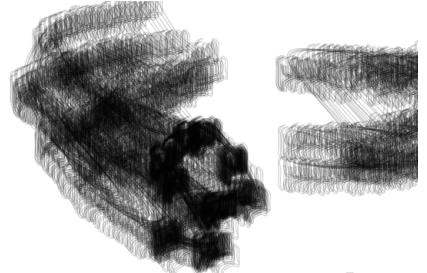


© 1999





**το φρούτο
εμείνε**

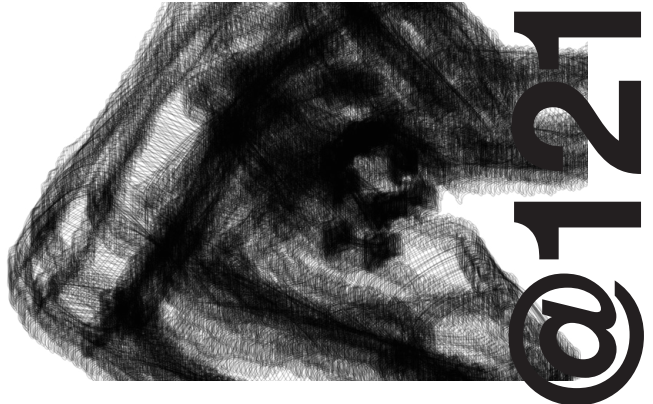


σφηνωμένο

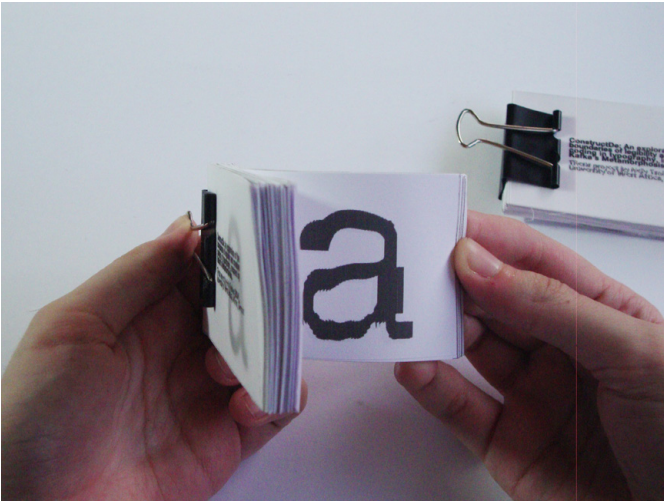


**στο σώμα
του**

ConstructDe



@121



ConstructDe

@122

2222

@123

Bibliography

1. Carson, D. (2000). *The end of print: The grafik design of David Carson*. New York: Universe Publishing.
2. McGowan, T. (2016). *Capitalism and desire: The psychic cost of free markets*. New York: Columbia University Press.
3. Doehling, R. M. (2019). *Automation in graphic design*. Georgia Southern University.
4. Matthews, B., Shannon, B., & Roxburgh, M. (2023). *Destroy all humans: The dematerialisation of the designer in an age of automation and its impact on graphic design*.
5. Poyron, R. (2003). *No more rules: Graphic design and postmodernism*. New Haven: Yale University Press.
6. McCarthy, M. (2010). *Generative art: A practical guide using Processing*. Manning Publications.
7. Ahn, Y., & Jin, G. (2016). *TYPE + CODE II: A code-driven typography*.
8. Iskandar, N. P. A. (2021). *Implicit meaning in Franz Kafka's novel "Metamorphosis"*.
9. Cong, M. (2023). *On the narrative art of Kafka's The Metamorphosis*. Journal of Education, Teaching and Social Studies.
10. Straus, N. (1989). *Transforming Franz Kafka's "Metamorphosis"*. MLN, 104 (5), 1088-1109.
11. Barfi, Z., Azizmohammadi, F., & Kohzadi, H. (2013). *A study of Kafka's The Metamorphosis in the light of Freudian psychological theory*. Journal of Applied Sciences Research, 9 (1), 551-555.
12. Kohzadi, H., Azizmohammadi, F., & Nouri, M. (2012). *A study of Franz Kafka's The Metamorphosis*. International Journal of Humanities and Social Science, 2 (16), 226-228.
13. Purwita, S. L. D., Murti, G., Cahyawati, E., Basuki, I., & Pratiwi, D. (2022). *Effect of alienation in Franz Kafka's The Metamorphosis*. Scholars International Journal of Linguistics and Literature, 5 (1), 12-17.
14. Yaron, I., & Herzog, O. (2013). *Kafka's ruins in popular culture: A story of Metamorphosis*.
15. Woods, T. (1999). *Beginning postmodernism*. Manchester: Manchester University Press.
16. Peppler, K., & Kafai, Y. (2009). *Creative coding: Programming for personal expression*. ETR&D, 57 (1), 12-24.
17. Greenberg, I. (2007). *Processing: Creative coding and computational art*. Berkeley: Apress.
18. Brown, J. D. (2019). *American graphic design in the 1990s: Deindustrialization and the death of the author*.
19. Sidra, K. (2019). *Kafka's short stories: Polysemic analysis of the world of symbols*.
20. Asriningtyas, D. L., & Mustofa, A. (2022). *Gregor Samsa's self-alienation in Franz Kafka's Metamorphosis: Lacanian psychoanalysis*.
21. Economou, I. (2003). *The cultural context of contemporary graphic design*.
22. Matthews, B., Shannon, B., & Roxburgh, M. (2023). *Destroy all humans: The dematerialisation of the designer in an age of automation and its impact on graphic design—A literature review*.
23. Uddin, M. J. (2023). *Exploring the depths of modern life as reflected in Franz Kafka's Metamorphosis*.
24. Greve, A. (2011). *The human body and the human being in "Die Verwandlung"*.
25. Bigelow, C. (2020). *The font wars, part 1 & 2*.
26. Kinross, R. (1985). *Rhetoric and neutrality*.
27. Hong, D., & Hwang, M. (2006). *The status and the prospects of deconstruction in graphic design*. In K. Friedman, T. Love, E. Corte-Real, & C. Rust (Eds.), Wonderground - DRS International Conference 2006, 1-4 November, Lisbon, Portugal.
28. Phrangopoulos, M. (2006). *Εισαγωγή στην ιστορία και τη θεωρία του graphic design: μια μικρή ανθολογία*. Futura.
29. Kafka, F. (1912). *Η Μεταμόρφωση*. (Μετάφραση: ΟΡΓΑΝΟΠΟΥΛΟΣ, Α.), ΚΑΡΑΚΩΤΣΟΓΛΟΥ.
30. Lacan, J. (1966). *Écrits*. New York: W.W. Norton & Company.
31. Meggs, P. B., & Purvis, A. W. (2016). *Meggs' history of graphic design* (6th ed.). Hoboken, NJ: Wiley.
32. Horst-Waldemar, J., Anthony, J., F. (2011). *ΙΣΤΟΡΙΑ ΤΗΣ ΤΕΧΝΗΣ Η ΔΥΤΙΚΗ ΓΙΑΠΑΔΟΣΗ*. ΚΩΝ.
33. Myers, T. (2001) *Lyotard and Lacan answering the question: what does Postmodernism want?* <http://www.jstor.org/stable/43263622>
34. Keedy, J. (1995). *Zombie modernism*.
35. Asriningtyas Devicha και Ali Mustofa, «Gregor Samsa's Self Alienation in Franz Kafka's Metamorphosis: Lacanian Psychoanalysis», *Humanitatis: Journal of Language and Literature*, τμ. 8
36. Jacques Derrida, «Différance» (α' μέρος), μτφρ. Γεράσιμος Κακολύρης, Πόηση, τχ. 27, Άνοιξη-Καλοκαίρι 2006
37. April Greiman, *Hybrid Imagery: The Fusion of Technology and Graphic Design*, New York, Watson-Guption Publications, 1990

& Sources

1. Wenzel, D. (2019, October 14). *Automated type design*. It's Nice That. Retrieved from https://www.itsnicethat.com/articles/daniel-wenzel-automated-type-design-digital-graphic-design-141019?utm_content=buffera0139&utm_medium=social&utm_source=twitter&utm_campaign=intsocial (27/07/24)
2. Victoria and Albert Museum. (n.d.). *What is postmodernism?* Retrieved from <https://www.vam.ac.uk/articles/what-is-postmodernism> (27/07/24)
3. Gorny, D. (08/2015). *Edge*. Retrieved from <https://dawidgorny.com/p/edge/> (27/07/24)
4. Variable. (2012). *Digital type wall*. Retrieved from <https://variable.io/digital-type-wall/> (27/07/24)
5. Bohnacker, H., Gross, B., Laub, J., & Lazzeroni, C. (2018). *Generative Gestaltung*. Retrieved from <http://www.generative-gestaltung.de/2/> (27/07/24)
6. Rodenbroeker, T. (May 23, 2024). *Omid Nermalhabib explores the intersection of Creative Coding and Perso-Arabic Typography*. Retrieved from <https://timrodenbroeker.de/omid-nermalhabib/> (27/07/24)
7. Flynn, J. (2018, August 8). *Why love generative art*. Artnome. Retrieved from <https://www.artnome.com/news/2018/8/8/why-love-generative-art> (27/07/24)
8. SparkNotes Editors. (n.d.). *The Metamorphosis: Analysis*. SparkNotes. Retrieved from <https://www.sparknotes.com/lit/metamorph/analysis/> (27/07/24)
9. eNotes Editorial. (n.d.). *Why is Gregor Samsa transformed into an insect in Kafka's The Metamorphosis?* eNotes. Retrieved from <https://www.enotes.com/topics/metamorphosis/questions/why-is-gregor-samsa-transformed-in-an-insect-in-381350> (27/07/24)
10. Harvard University. (2016). *Kafka's Metamorphosis: A Journey of Identity with Language as a Vehicle*. Retrieved from <https://hum11c.omeka.fas.harvard.edu/exhibits/show/readings/kafka-s-metamorphosis--a-journ> (27/07/24)
11. Baldwin, E. (n.d.). *The Metamorphosis: Analysis*. Book Analysis. Retrieved from <https://bookanalysis.com/franz-kafka/the-metamorphosis/analysis/> (27/07/24)
12. CliffsNotes Editorial. (n.d.). *The Metamorphosis: Summary and analysis*. "CliffsNotes". Retrieved from <https://www.cliffsnotes.com/literature/m/the-metamorphosis-and-other-stories/summary-and-analysis/the-metamorphosis-die-verwandlung> (27/07/24)
13. SparkNotes Editors. (n.d.). *The Metamorphosis: Symbols*. SparkNotes. Retrieved from <https://www.sparknotes.com/lit/metamorph/symbols/> (27/07/24)
14. IvyPanda Editorial. (May 21, 2024). *Study guide on The Metamorphosis: Symbols and literary analysis*. IvyPanda. Retrieved from <https://ivypanda.com/lit/study-guide-on-the-metamorphosis/symbols-literary-analysis/> (27/07/24)
15. Course Hero Editorial. (n.d.). *The Metamorphosis: Symbols*. Course Hero. Retrieved from <https://www.coursehero.com/lit/The-Metamorphosis/symbols/> (27/07/24)
16. SparkNotes Editors. (n.d.). *The Metamorphosis: Motifs*. SparkNotes. Retrieved from <https://www.sparknotes.com/lit/metamorph/motifs/> (27/07/24)
17. Eindhoven Maker Faire. (2020). *Possibilities of creative coding*. Retrieved from <https://eindhoven.makerfaire.com/wp-content/uploads/sites/174/2020/08/Possibilities-of-Creative-Coding.pdf> (27/07/24)
18. Tate. (n.d.). *Postmodernism*. Tate. Retrieved from <https://www.tate.org.uk/art/art-terms/p/postmodernism> (27/07/24)
19. TVXS Editors. (n.d.). Jacques Derrida: *O akatanoitos tis filosofias*. TVXS. Retrieved from <https://tvxs.gr/istoria/san-simera-istoria/jacque-derrida-o-akatanoitos-tis-filosofias/> (27/07/24)
22. Keedy, M. (1998). *Graphic design in the postmodern era*. Emigre. Retrieved from <https://www.emigre.com/Essays/Magazine/GraphicDesigninthePostmodernEra> (27/07/24)
23. Heller, S. (2016). *The legibility wars of the '80s and '90s*. Print. Retrieved from <https://www.printmag.com/typography/legibility-wars-translation-typography/> (27/07/24)
24. "Alienation." (2022). In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. Retrieved from <https://plato.stanford.edu/entries/alienation/> (27/07/24)

Colophon

ConstructiDe

@126

Typography **Helvetica Neue**
Linotype

Cover **Munken Kristal Rough 300gsm**

Internal pages **Munken Kristal Rough 120gsm**

Thesis supervisor **Professor Eva Masoura**

**Department of Graphic Design and Visual Communication, University of
West Attica (UNIWA), Athens, Greece**

ConstructDe

@127

