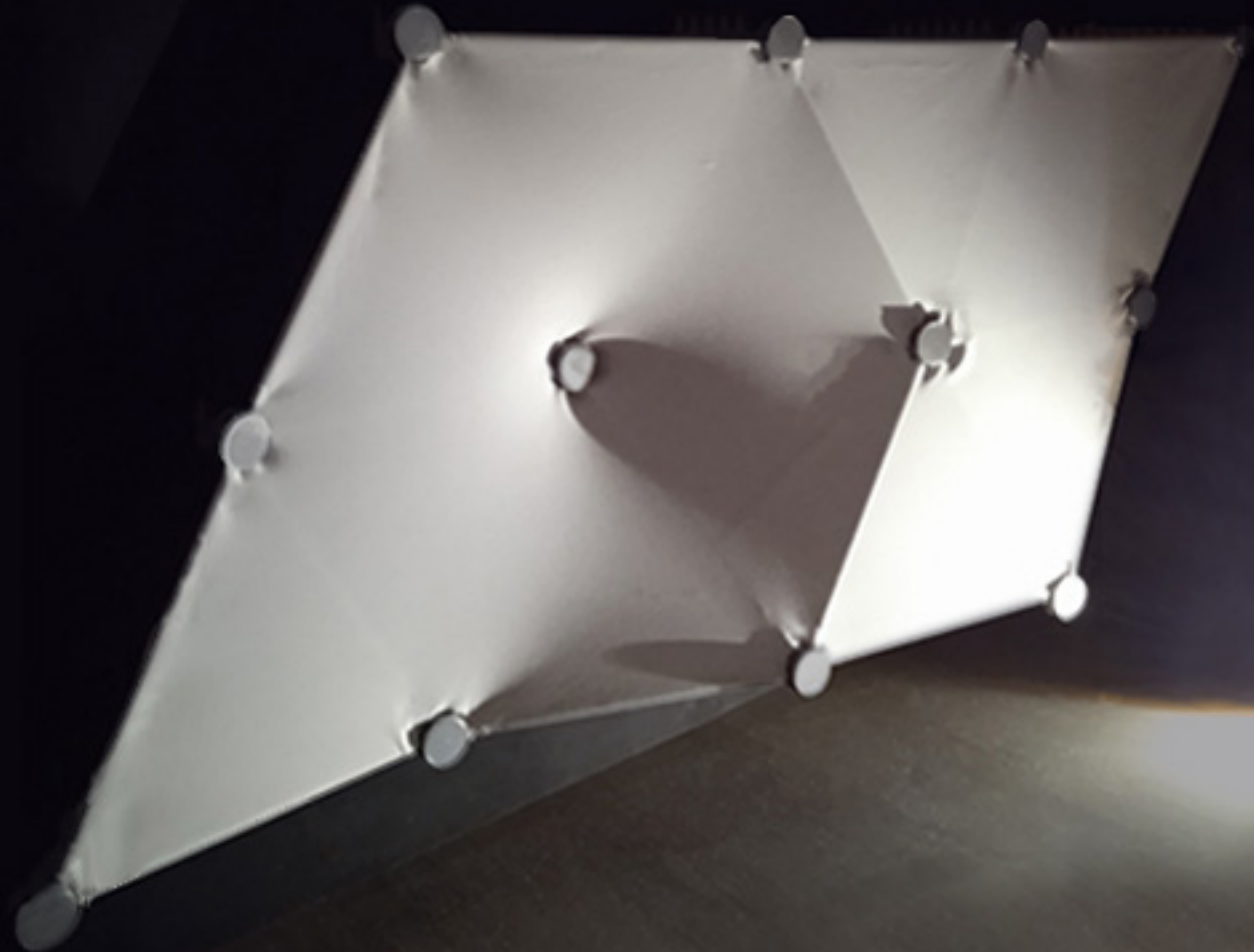


ΔΙΑΔΡΑΣΤΙΚΟΣ ΤΟΙΧΟΣ: ΑΝΤΑΝΑΚΛΩΝΤΑΣ ΠΤΥΧΕΣ ΔΗΜΟΣΙΑΣ ΖΩΗΣ

Επιβλέπων:
Ματθαίος Παπαβασιλείου

Φοιτήτρια:
Ειρήνη - Μαρία Γεωργαντά



ΔΙΑΔΡΑΣΤΙΚΟΣ ΤΟΙΧΟΣ: ΑΝΤΑΝΑΚΛΩΝΤΑΣ ΠΤΥΧΕΣ ΔΗΜΟΣΙΑΣ ΖΩΗΣ

Σπουδάστρια: Ειρήνη - Μαρία Γεωργαντά
Αριθμός μητρώου: ar16004

Επιβλέπων καθηγητής: Ματθαίος Παπαβασιλείου
Τομέας Ι - Αρχιτεκτονικός Σχεδιασμός

Διπλωματική εργασία
Σεπτέμβριος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αρχιτεκτόνων Μηχανικών



Ευχαριστώ τον κ. Ματθαίο Παπαβασιλείου για τη συνεργασία του και την εν γένει καθοδήγησή του στην τελική αυτή ευθεία των σπουδών μου.
Ευχαριστώ την ομάδα από το εργαστήριο ΠΟΙΩ του δήμου Αθηναίων για την προθυμία τους και την πολύτιμη βοήθειά τους στην υλοποίηση της ιδέας της παρούσας διπλωματικής.
Ευχαριστώ τον διδάκτορα της Σχολής Αρχιτεκτόνων Μηχανικών Ε.Μ.Π. κ. Παναγιώτη Νικολαΐδη, για την συμβολή του στην συγγραφή του κώδικα στη VIsip.
Ευχαριστώ την Αθηνά, τον Νίκο, τον Γιάννη, τη Χριστίνα και όλα τα άτομα που ήταν κοντά μου στη διάρκεια της εργασίας αυτής.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου προς την οικογένειά μου, καθώς η υποστήριξή της σε όλα τα επίπεδα ήταν καθοριστική για την επιτυχή ολοκλήρωση των σπουδών μου στο Εθνικό Μετσόβιο Πολυτεχνείο.

“Δέν υπάρχει δυσκολία πού δέν ἔχει λύση ἐν Χριστῷ.
Δώσου στόν Χριστό καί Ἐκεῖνος θά σοῦ βρεῖ λύση!
Μή φοβᾶσαι τίς δυσκολίες!”

-Άγιος Πορφύριος ο Καυσοκαλυβίτης-

*Εἰς ανάμνηση της πορείας
αυτῆς της διπλωματικῆς*

στην Μητέρα μου

Περιεχόμενα

Εισαγωγή	8
Αναφορές	10
Πορεία δουλειάς	15
Τελική πρόταση	24
Κατασκευή	34
Αυτοματοποίηση	41
Προτάσεις εφαρμογής	52
Επίλογος	57

Διατύπωση προβληματισμού

Όπως το φυσικό περιβάλλον δεν είναι προκαθορισμένο αλλά μεταβάλλεται συνεχώς από διάφορες δυνάμεις, έτσι και ο δημόσιος χώρος θα μπορούσε να αποκτήσει τέτοια χαρακτηριστικά που να αντικατοπτρίζει αυτή την πραγματικότητα.

Πώς όμως αυτή η μεταβολή θα μπορούσε να συνδεθεί με την δημόσια ζωή και την ανθρώπινη δραστηριότητα μέσα στον δημόσιο χώρο;

Ποια θα έπρεπε να ήταν η «δύναμη» που θα όριζε τη μεταβλητότητα του δημόσιου χώρου σε αυτή την περίπτωση;

Εισαγωγή

Η εργασία αυτή είχε σκοπό να μελετήσει τον σχεδιασμό ενός συστήματος, το οποίο μέσα από τη μεταβλητότητα και τη διαμόρφωση χωρικών εντάσεων, θα μπορεί να αντανakλά και να επαναπροσδιορίζει την ανθρώπινη ζωή στον δημόσιο χώρο. Την ίδια στιγμή, οι περιορισμοί που τέθηκαν ήταν η στόχευση προς μια κατασκευή που θα χρησιμοποιεί απλά υλικά και στοιχεία, τα οποία θα μπορούν να συναρμολογούνται εύκολα και από άτομα χωρίς εξειδίκευση. Επίσης, επιδίωξη αποτελούσε και η χρήση ενός ανοιχτού λογισμικού, το οποίο να δίνει τη δυνατότητα στον καθένα να επέμβει και να επεκτείνει την κατασκευή, δημιουργώντας παραλλαγές.

Προς αυτή τη κατεύθυνση, η μελέτη επικεντρώθηκε γύρω από δύο παραμέτρους- στόχους που αφορούν αρχικά τον προσδιορισμό της μεταβλητότητας στην σχεδιαζόμενη κατασκευή και ταυτόχρονα την σύνδεση αυτής με την δημόσια ζωή και δραστηριότητα σε έναν δημόσιο χώρο.

Ο προσδιορισμός της μεταβλητότητας έγινε μέσα από τον ορισμό της μορφής του συστήματός μας. Η διατύπωση της τελικής μορφής προήλθε ύστερα από μία έρευνα διατύπωσης και μελέτης πολλών και διαφορετικών ιδεών. Κάθε κεντρική ιδέα που σχεδιάστηκε και μελετήθηκε, αποτελεί εξέλιξη της προηγούμενης της, περιορίζοντας τις αδυναμίες που εντοπίστηκαν και προσεγγίζοντας καλύτερα τον αρχικό στόχο.

Την ίδια στιγμή, προσπαθώντας να απαντήσουμε στο αρχικό ερώτημα της σύνδεσης των μεταβαλλόμενων χωρικών ποιοτήτων με τη δημόσια ζωή, επιλέγεται το ανθρώπινο σώμα και η κίνηση αυτού ως η «δύναμη» που προκαλεί την οποιαδήποτε μεταβολή. Έτσι, σχεδιάστηκε ένας τοίχος, ο οποίος καθίσταται κινούμενος όταν αλληλεπιδρά με τον άνθρωπο.

Πιο συγκεκριμένα, προσελκύει τον κόσμο και καταγράφει την κίνηση του ανθρώπου μέσα από την αντίστοιχη παραμόρφωση που συμβαίνει στην επιφάνειά του. Με αυτό τον τρόπο, συμμετέχει σε αυτό που διαδραματίζεται μπροστά του και αναπαριστά με τρόπο αφαιρετικό τη δράση που εκτυλίσσεται στα πλαίσια της δημόσιας ζωής. Αποτελεί ουσιαστικά ένα έπιπλο δημόσιου χώρου, το οποίο συναρμολογείται και τοποθετείται οπουδήποτε, εισάγωντας την κινητική τέχνη στον αστικό χώρο.

Για την κατασκευαστική επίλυση του συστήματος αυτού και τον σχεδιασμό του μηχανισμού κίνησής του, κατασκευάστηκαν κάποιες αρχικές μακέτες που αναπαριστούν τη μονάδα του συστήματος, δηλαδή το τρίγωνο. Στο επίπεδο αυτό δόθηκε έμφαση σε κατασκευαστικά και προγραμματιστικά ζητήματα, επιλύοντας την αυτοματοποίηση της κατασκευής και του τρόπου διάδρασής της. Αφού λοιπόν στο επίπεδο της μονάδας επιλύθηκαν όλα τα παραπάνω, κατασκευάστηκε η τελική μακέτα που αναπαριστά μέρος του προτεινόμενου διαδραστικού τοίχου. Είναι σε κλίμακα 1:2, αυτοματοποιείται και αναδεικνύει (στον βαθμό που της επιτρέπει το μέγεθός της) την κίνηση της κατασκευής, αλληλεπιδρώντας με τον άνθρωπο.

Στη συνέχεια προτείνονται ενδεικτικές χρήσεις της κατασκευής στον δημόσιο χώρο, που αναδεικνύουν τις δυνατότητες του συστήματος και του τρόπου που αφαιρετικά αναπαριστά διαφορετικές πτυχές δημόσιας ζωής, μέσα από την τοποθέτησή του σε διαφορετικούς χώρους.

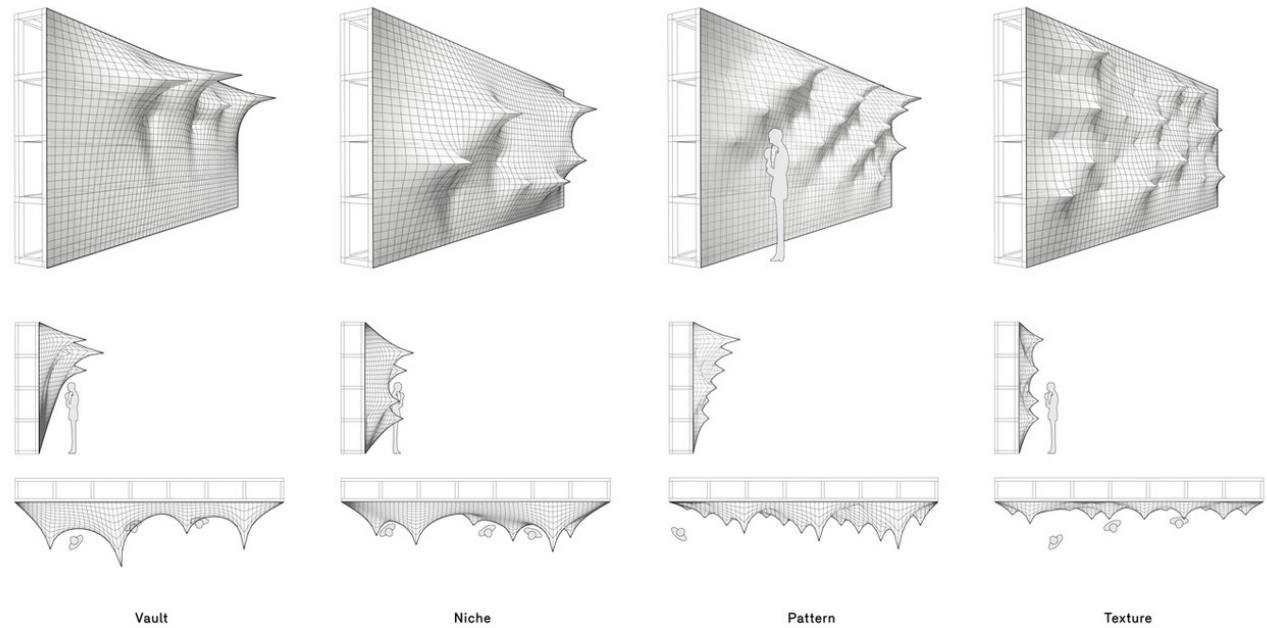
Τέλος, προσομοίωση της ιδέας πραγματοποιείται μέσα από πρόγραμμα σε γλώσσα προγραμματισμού Visual Lisp που τρέχει σε περιβάλλον AutoCAD.

Η μελέτη κατασκευής όλων των προπλασμάτων από τα πρώτα στάδια δοκιμών μέχρι και την υλοποίηση της τελικής μακέτας, καθώς και η αυτοματοποίηση αυτής, έγινε στο εργαστήριο ΠΟΙΩ, το maker space του δήμου Αθηναίων.

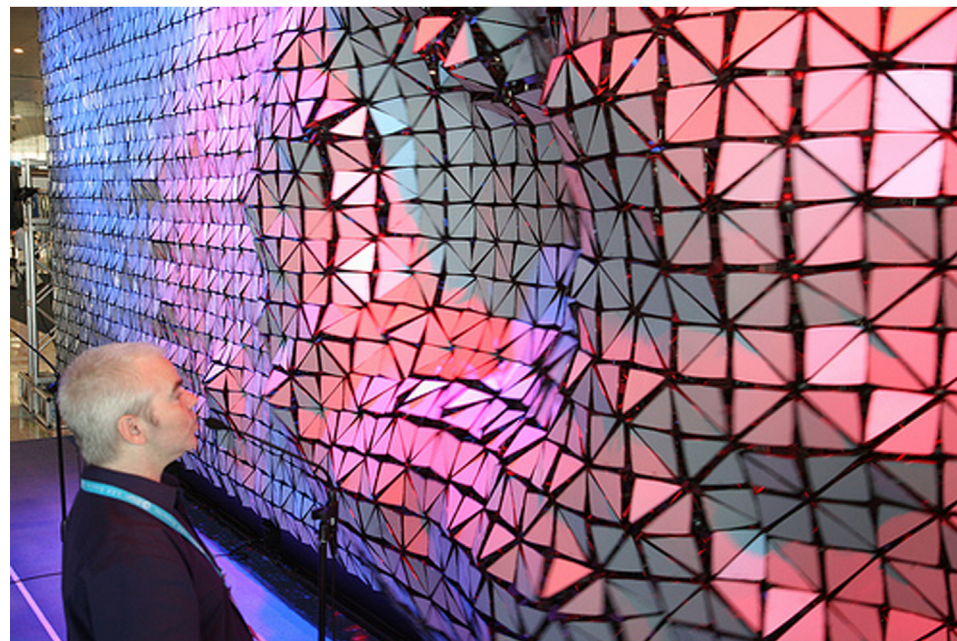
Αναφορές

“Kinetic Wall” των Frank Barkow και Regine Leibinger.

Παρουσιάστηκε στην έκθεση Rem Koolhaas' Elements, Venice Architecture Biennale 2014

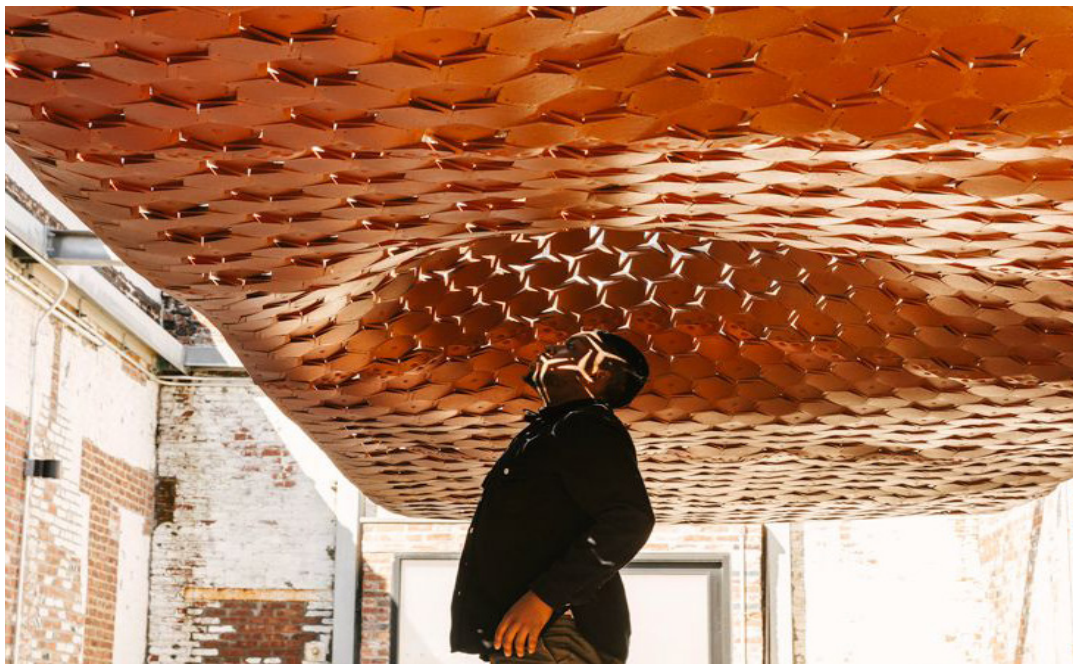


HypoSurface, Mark Goulthorpe - dECOi Architects



<https://archimedespool.wordpress.com/2007/10/05/hyposurface/>
<http://textileatmospheres.blogspot.com/2010/07/aegis-hyposurface.html>

Αστικό αποτύπωμα, εγκατάσταση που σχεδιάστηκε από το Studio INI.
Παρουσιάστηκε στο A/D/O στο Μπρούκλιν το 2019



Εγκατάσταση με το όνομα Disobedience στην αυλή του Somerset House που σχεδιάστηκε από το Studio INI.

London Design Biennale in 2018

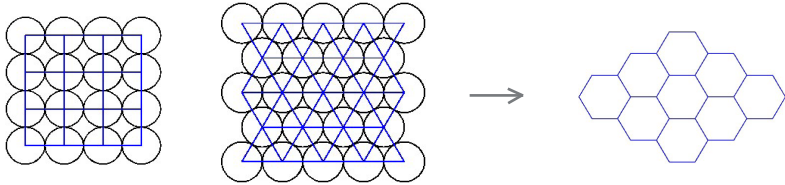


Πορεία δουλειάς

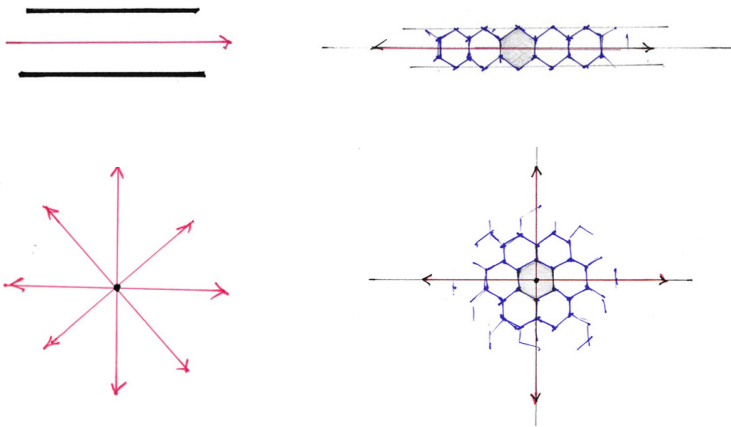
Τρόποι μεταβλητότητας

1. Συνένωση κυττάρων

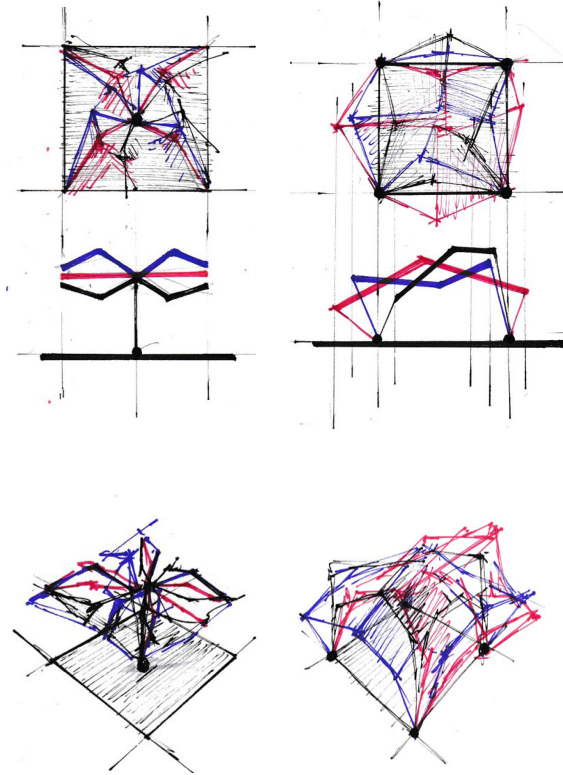
Εύρεση βέλτιστου τρόπου συνένωσης- packing



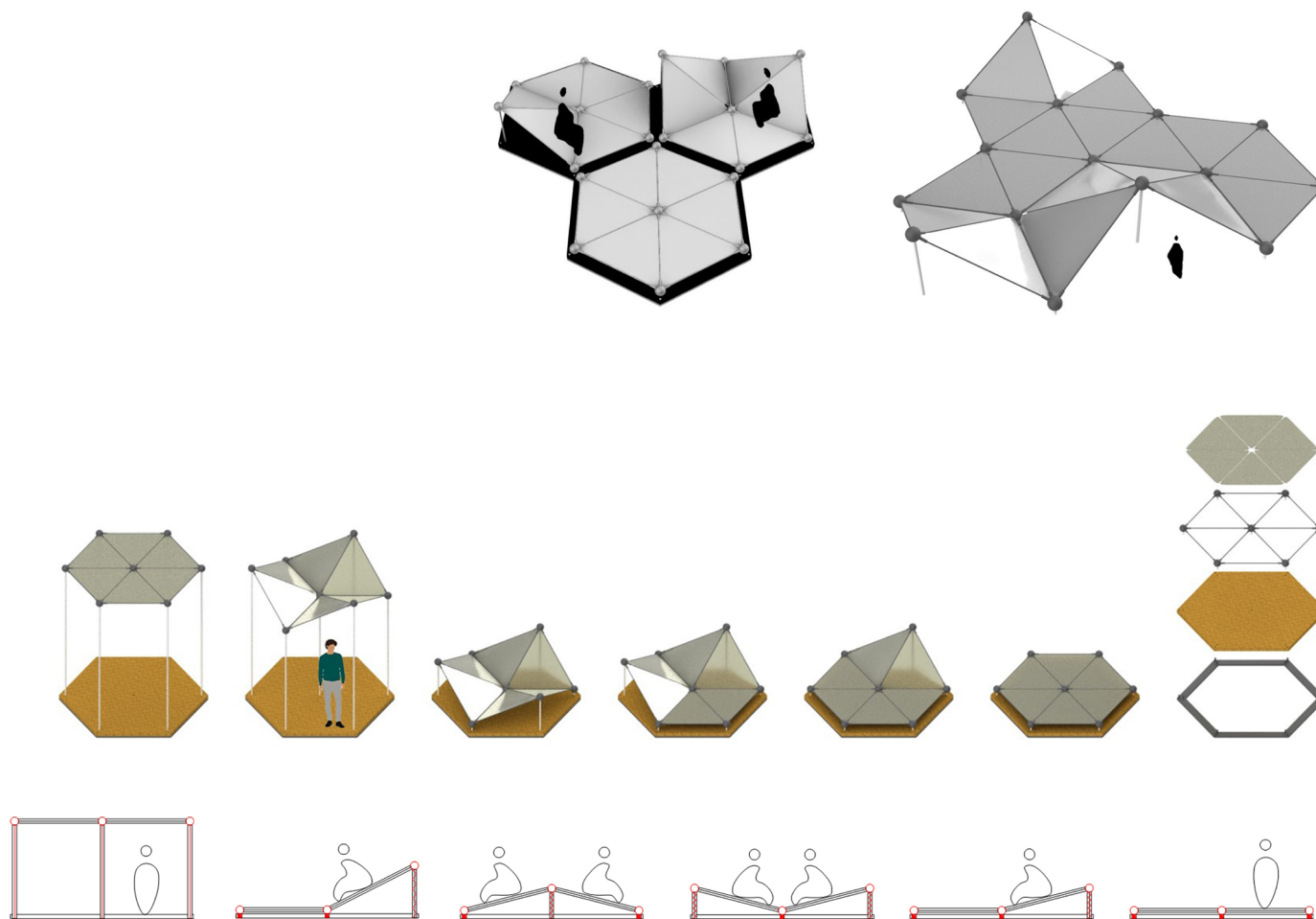
Συνένωση - επεκτασιμότητα ανάλογα με την μορφή του χώρου και τις κινήσεις σε αυτόν



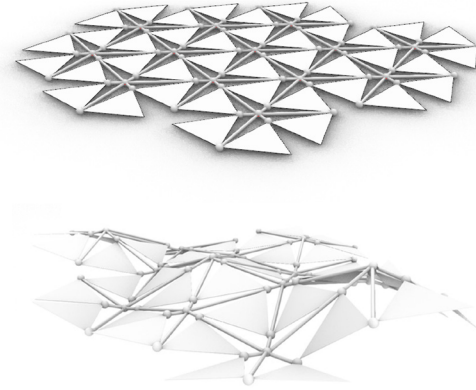
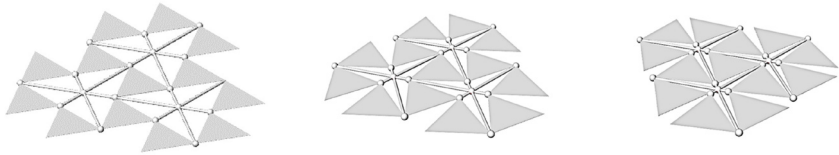
2. Επέκταση κυττάρου



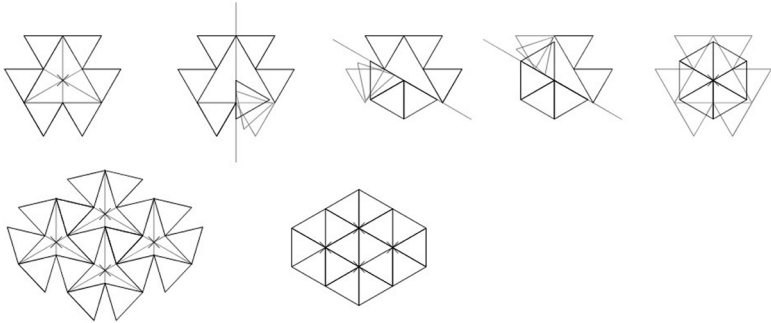
Ξεκινώντας και έχοντας ως σταθερά την αρχή της μεταβλητότητας στο υπο μελέτη σύστημα, διερευνώνται οι πιθανοί τρόποι που μπορεί αυτή να υλοποιηθεί. Διαπιστώνουμε δύο διαφορετικούς τρόπους, πρώτον με τη δυνατότητα συνένωσης πολλών κυττάρων σταθερής μορφής και δεύτερον μέσω διατήρησης σταθερής κάτοψης και μεταβολή σε τομή ή όψη.



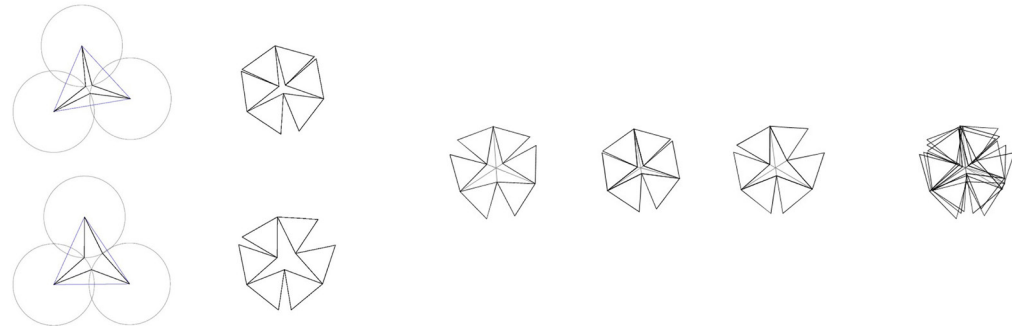
Αξιοποιώντας τον δεύτερο τρόπο μεταβλητότητας και διατηρώντας δηλαδή σταθερή την κάτοψη, σχεδιάστηκε μία πρώτη προσέγγιση μεταβλητής κατασκευής που αποτελείται από τρίγωνα τα οποία παραμένουν ισόπλευρα σε κάτοψη και αλλάζοντας το ύψος των κορυφών τους, δημιουργούν διαφορετικές χωρικές ποιότητες ικανές να υποδεχτούν και διαφορετικές λειτουργίες, όπως καθίσματα ή στέγαστρα.



Μελέτη κίνησης των τριγώνων γύρω από τις τηλεσκοπικές ράβδους



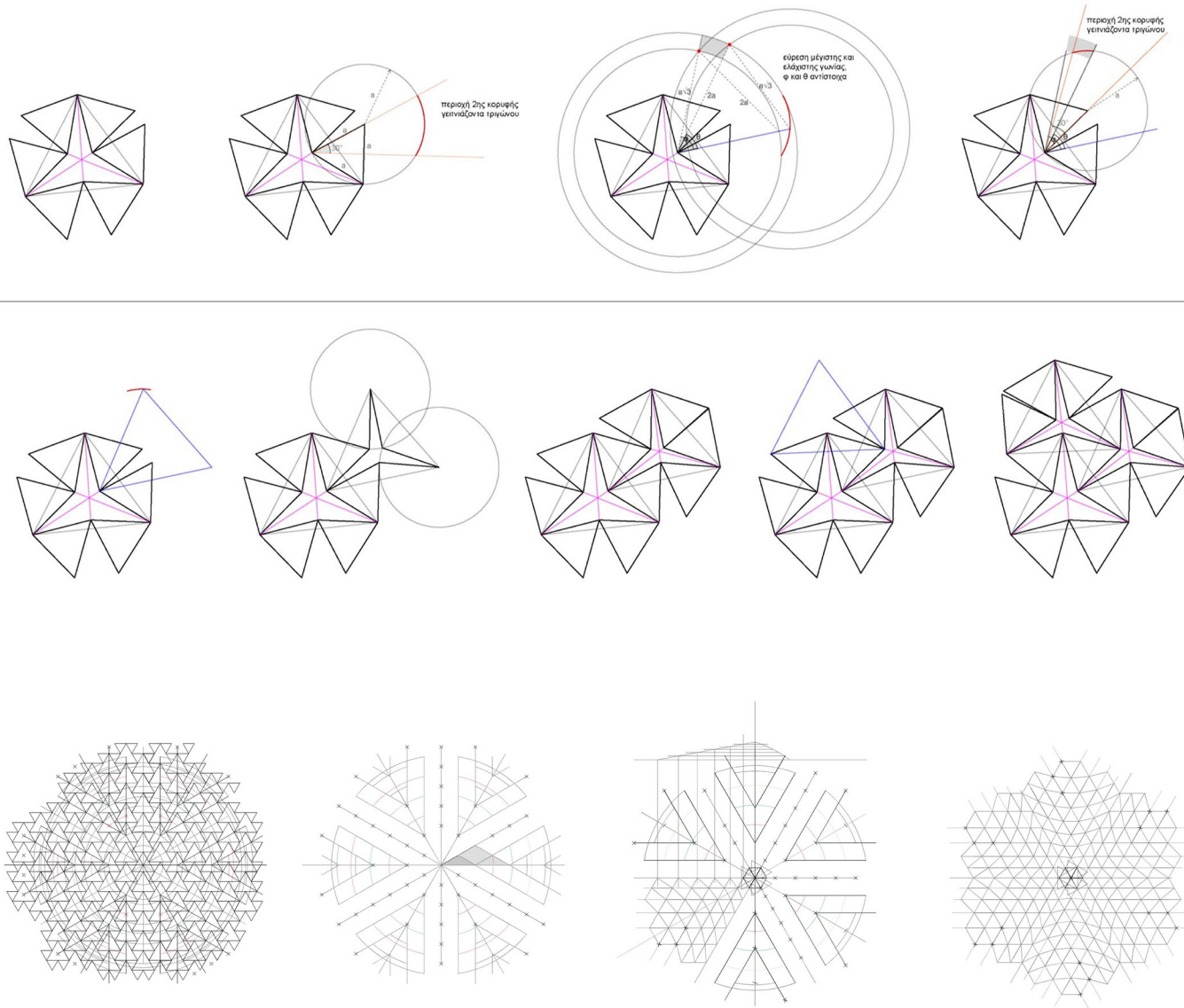
Συμμετρικό άνοιγμα τριγώνων-
δυνατότητα ομοιόμορφης
αναδίπλωσης σε όλη την κατασκευή



Μη συμμετρικό άνοιγμα
τριγώνων- δυνατότητα
ποικιλομορφίας στην
κατασκευή

Με βάση το υπόβαθρο αυτό, η μεταβλητότητα σε μία δεύτερη προσέγγιση στηρίχθηκε στην διατήρηση σταθερής γεωμετρίας στην επιφάνεια των τριγώνων και την προσθήκη τηλεσκοπικών ράβδων στις μεταξύ τους συνδέσεις. Ο τρόπος με τον οποίο ανοίγουν οι τηλεσκοπικές ράβδοι, μπορεί να παράξει συμμετρικές ή μη γεωμετρίες. Πιο συγκεκριμένα, με το ομοιόμορφο άνοιγμα των ράβδων επαναλαμβάνεται το ίδιο γεωμετρικό μοντέλο σε όλη την επιφάνεια, ενώ το μη συμμετρικό άνοιγμα των εξαγώνων δημιουργεί ποικιλομορφία.

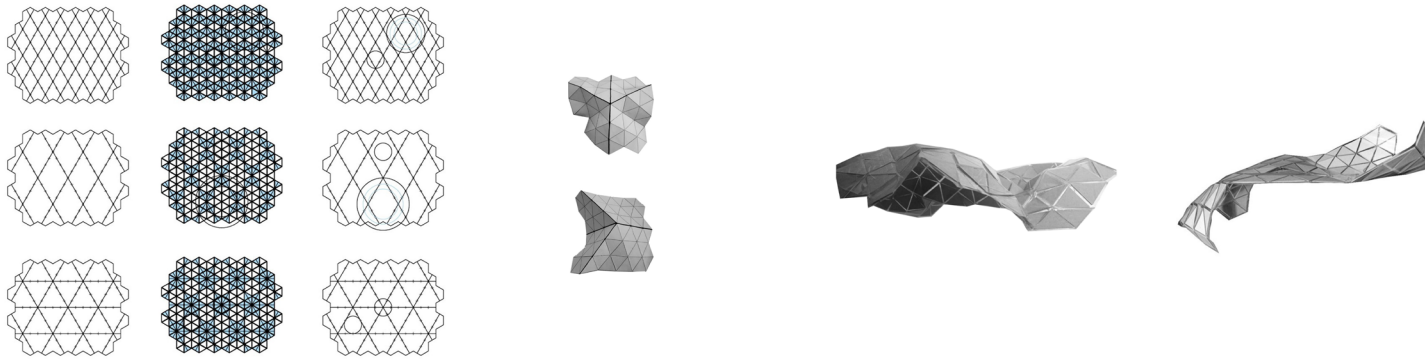




Προσεγγίζοντας μορφολογικά τις ποιότητες που παραγει το μη συμμετρικό άνοιγμα των τριγώνων, έγινε μία γεωμετρική μελέτη αποκωδικοποίησης του τρόπου με τον οποίο διατάσσονται σε σειρά τρίγωνα με διαφορετικές γωνίες στροφής. Ταυτόχρονα, μελετώντας την γεωμετρία αυτή σε μία μεγαλύτερη κλίμακα, έγιναν κάποιες σκέψεις για την μορφή του συνολικού μοντέλου με βάση την θέση των κέντρων γύρω από τα οποία στρέφονται τα τρίγωνα και συγκεκριμένα θεωρώντας ένα κεντρικό το οποίο «έλκει» τα υπόλοιπα γύρω από τον εαυτό του. Οι διατάξεις που προκύπτουν παρουσιάζουν ενδιαφέρον.

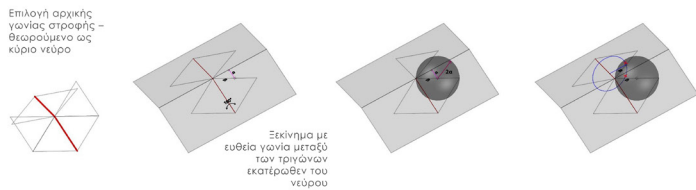
Σύστημα διαμόρφωσης της επιφάνειας

Χρήση “νεύρων” που διατρέχουν την επιφάνεια, την τσακίζουν και αποτελούν τον οδηγό που της δίνει τις επιθυμητές μορφές.



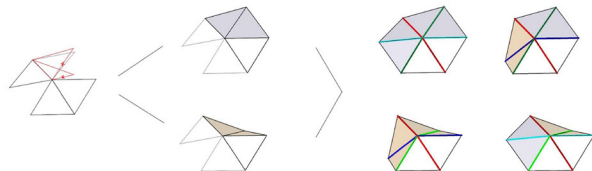
Γεωμετρική μελέτη αναδίπλωσης τριγώνων – προκύπτουσες δυνατές γωνίες σε κάθε κόμβο

Μελέτη της συμπεριφοράς του μοντέλου, όπως διαμορφώνεται με βάση μια δεδομένη μορφή ενός νεύρου.

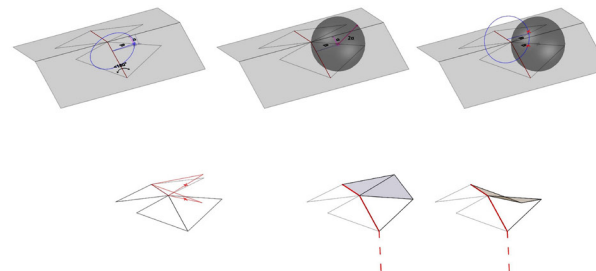


Επιλογή αρχικής γωνίας στροφής – θεωρούμενο ως κύριο νεύρο

Εκκίνηση με ευθεία γωνία μεταξύ των τριγώνων εκατέρωθεν του νεύρου

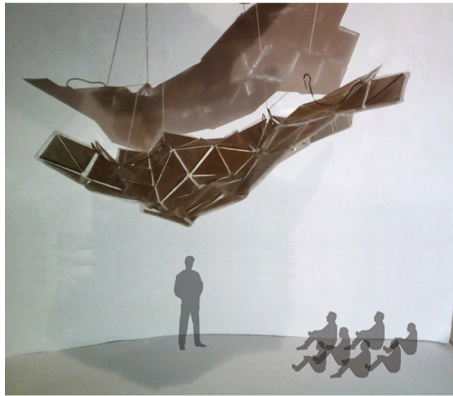


Δυνατοί συνδυασμοί γωνιών αναδίπλωσης, ανάλογα με τις προθέσεις επέκτασης της γεωμετρίας της κάσας κελύφ. Παραμένει πάντα σταθερή η επιλεγμένη ως αρχική γωνία του κύριου νεύρου (κόκκινη)



Η περιπλοκότητα ωστόσο της προηγούμενης μορφής, ειδικά όσον αφορά την μετάφρασή της σε χωρικές ποιότητες τριών διαστάσεων, οδήγησε στην απλοποίησή της με σκοπό την επίτευξη μίας πιο καθαρής γεωμετρίας. Έτσι λοιπόν, η επιφάνεια αναδιαμορφώνεται και συνίσταται από ίδια ισόπλευρα τρίγωνα παντού. Σε αυτό το σύστημα, για την παραγωγή χώρων διπλής καμπυλότητας, προτείνεται η χρήση αξόνων ή αλλιώς «νεύρων» που διατρέχουν την επιφάνεια, την τσακίζουν και αποτελούν τον οδηγό που της δίνει τις επιθυμητές μορφές. Στα πλαίσια αναζήτησης των πιθανών τελικών μορφών όλης της επιφάνειας, έγινε μία ακόμα γεωμετρική μελέτη που στόχευε στον προσδιορισμό των προκύπτουσων δυνατών γωνιών σε κάθε σημείο τσακίσης των νεύρων.

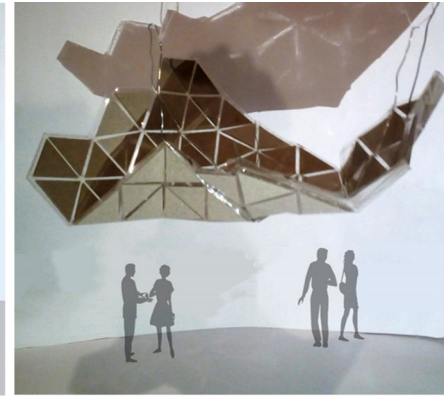
Αναζήτηση τυπολογιών χώρων ανάλογα με τις χωρικές σχέσεις



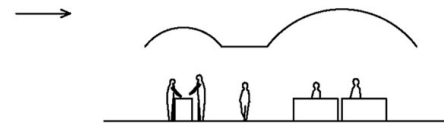
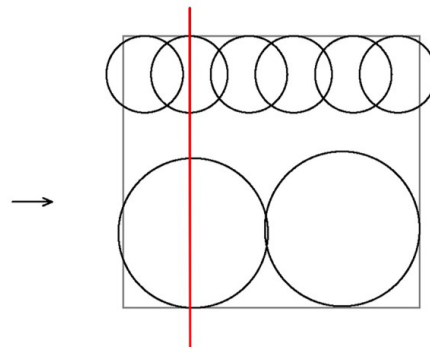
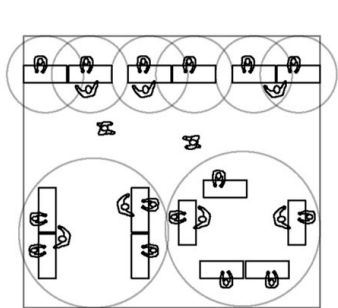
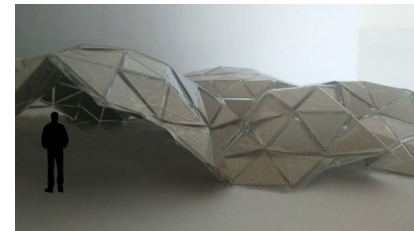
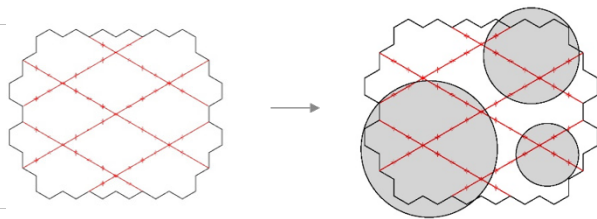
Μετωπικότητα



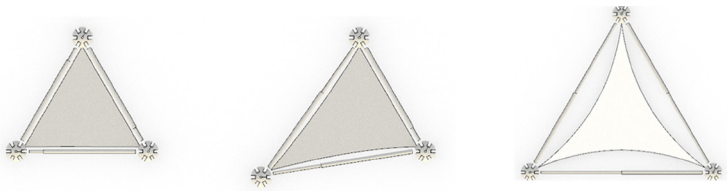
Κεντρικότητα



Γραμμικότητα



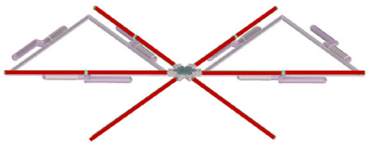
Ο τρόπος που το σύστημα αυτό θα μπορούσε να διαμορφώσει χωρικές εντάσεις, αρχικά προσδιορίστηκε αφαιρετικά αναπαριστώντας διαφορετικές τυπολογίες χώρων, όπως μετωπικότητα, κεντρικότητα ή γραμμικότητα. Ύστερα, η παραμόρφωση συνδέθηκε με τις χρήσεις που εκτυλίσσονται κάθε φορά και έτσι νοητοί κύκλοι που πλαισιώνουν την κάθε χρήση «μεταφέρονται» στην κατασκευή προκαλώντας την αντίστοιχη μεταβολή στην μορφή της, μέσα από την κατάλληλη γεωμετρία που παίρνουν τα νεύρα. Έτσι η αλληλεπίδραση με τον δημόσιο χώρο βασίζεται στις χωρικές ποιότητες που δημιουργούνται και πλαισιώνουν τις εκάστοτε χρήσεις.



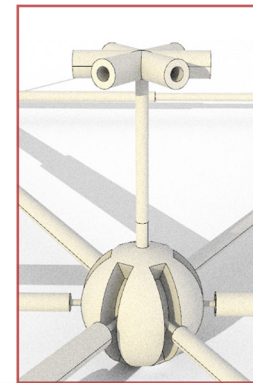
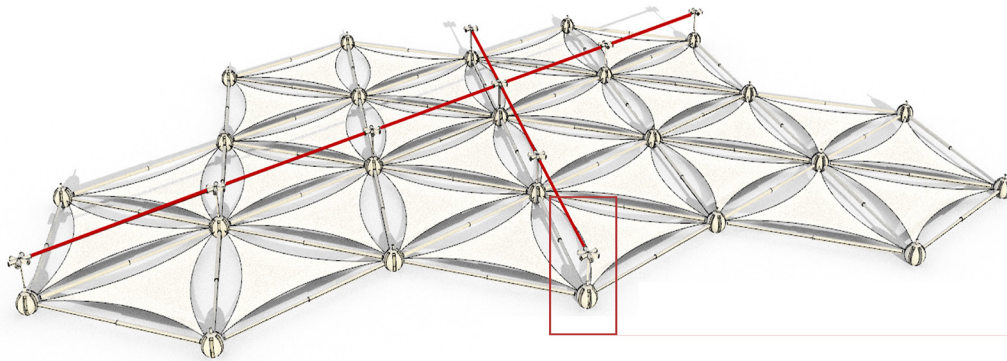
Τρίγωνα με τηλεσκοπικές ράβδους



“Νεύρα” σε ξεχωριστό επίπεδο πάνω από την επιφάνεια

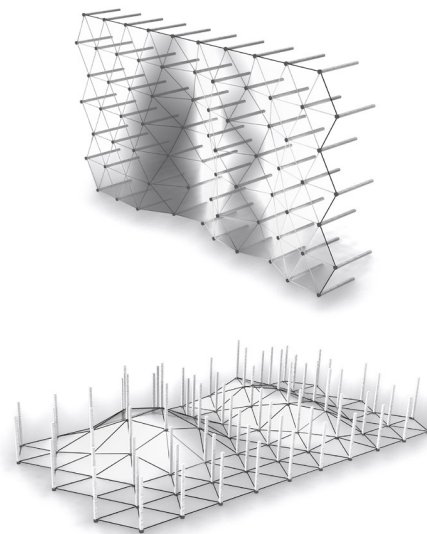
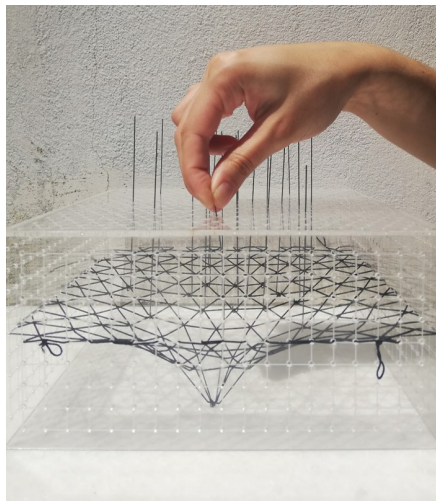


Ελέγχουν την θέση των κορυφών των τριγώνων μέσω τηλεσκοπικών ράβδων

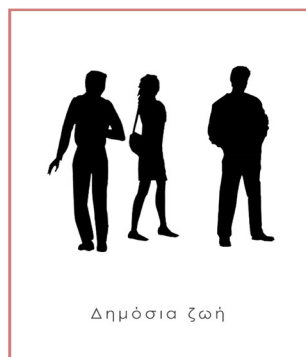


Επαναπροσδιορίζοντας την περιπλοκότητα δημιουργίας μορφών διπλής καμπυλότητας με τρίγωνα σταθερής διάστασης που αναδιπλώνονται, τα τελευταία αντικαθιστούνται με πλευρές μεταβλητής διάστασης. Έτσι η επιφάνεια του εκάστοτε τριγώνου μεταβάλλεται μέσα από τηλεσκοπικές ράβδους στις πλευρές του και ελαστικά υφάσματα για πλήρωση. Καθώς τα τρίγωνα έχουν αυξομειώμενες διαστάσεις, τα νεύρα διαχωρίζονται από αυτά και τοποθετούνται ψηλότερα από το κέλυφος, έχοντας σταθερό μήκος. Συνεχίζουν να αποτελούν τα σημεία ελέγχου της κατασκευής μέσω κατακόρυφων τηλεσκοπικών ράβδων, που καθορίζουν την καθ' ύψος θέση των κορυφών των τριγώνων που ελέγχουν.

Τελική μορφή:



Μεταβλητότητα



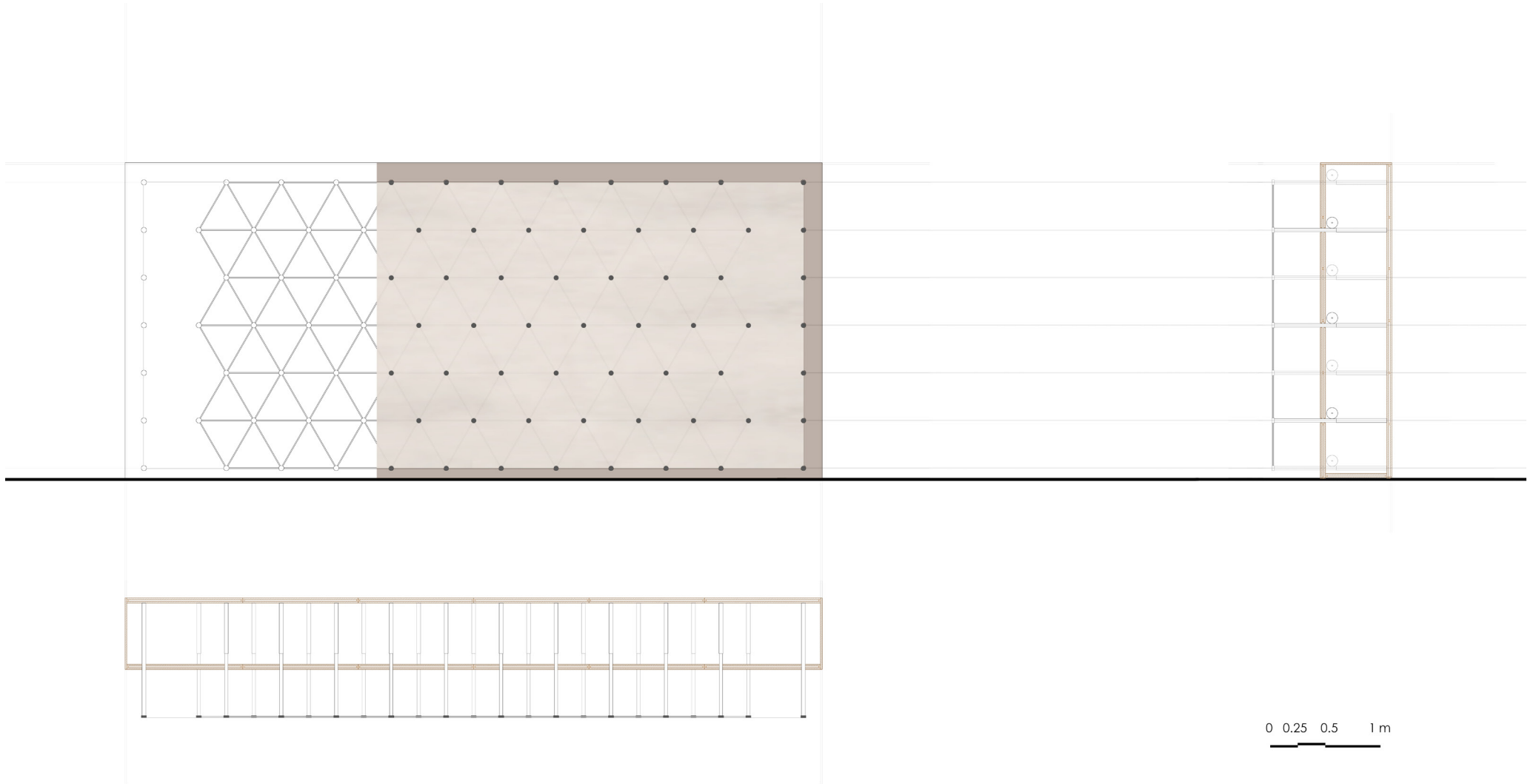
Δημόσια ζωή

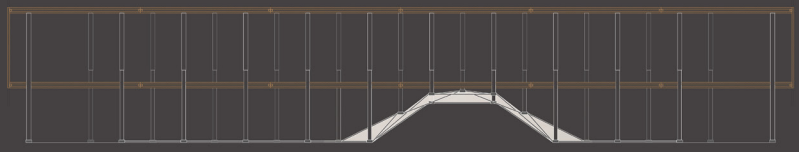
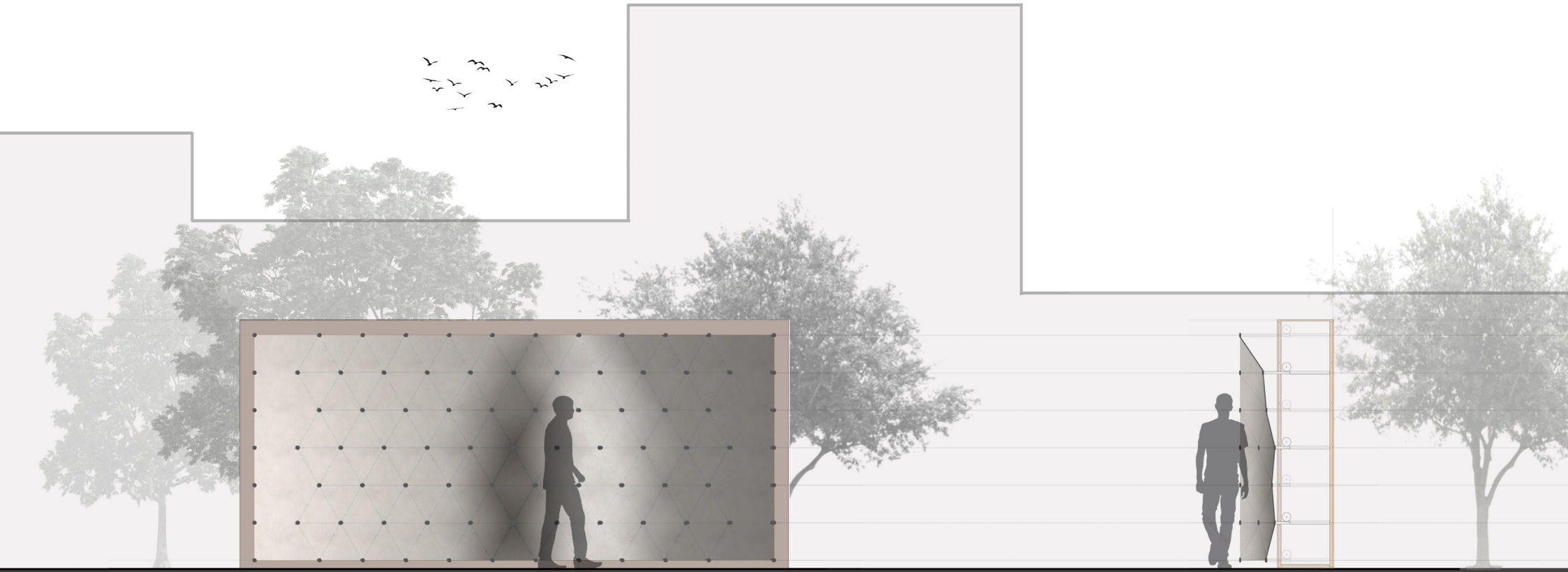
Τα «νεύρα» που χρησιμοποιήθηκαν στις τελευταίες λύσεις, δεν ελέγχουν όλα τα τρίγωνα, αλλά μόνο μερικές κορυφές τριγώνων, και συνεπώς η επιφάνεια δεν μπορεί να ελεγχθεί πλήρως και να παραμορφωθεί στα σημεία που δεν διαπερνώνται από αυτά. Έτσι λοιπόν, τοποθετούνται τηλεσκοπικές ράβδοι πάνω από κάθε κορυφή των τριγώνων και όχι κατά μήκος συγκεκριμένων νεύρων. Άρα το σύστημα καμπυλώνει και παραμορφώνεται σε οποιοδήποτε σημείο χρειαστεί.

Επίσης, μεταξύ άλλων παραγόντων, επιλέγεται ο άνθρωπος ως το μέσο που θα διαμορφώνει τις χωρικές εντάσεις, ώστε να συνδεθεί η μεταβλητότητα με διαφορετικές πτυχές του δημόσιου βίου.

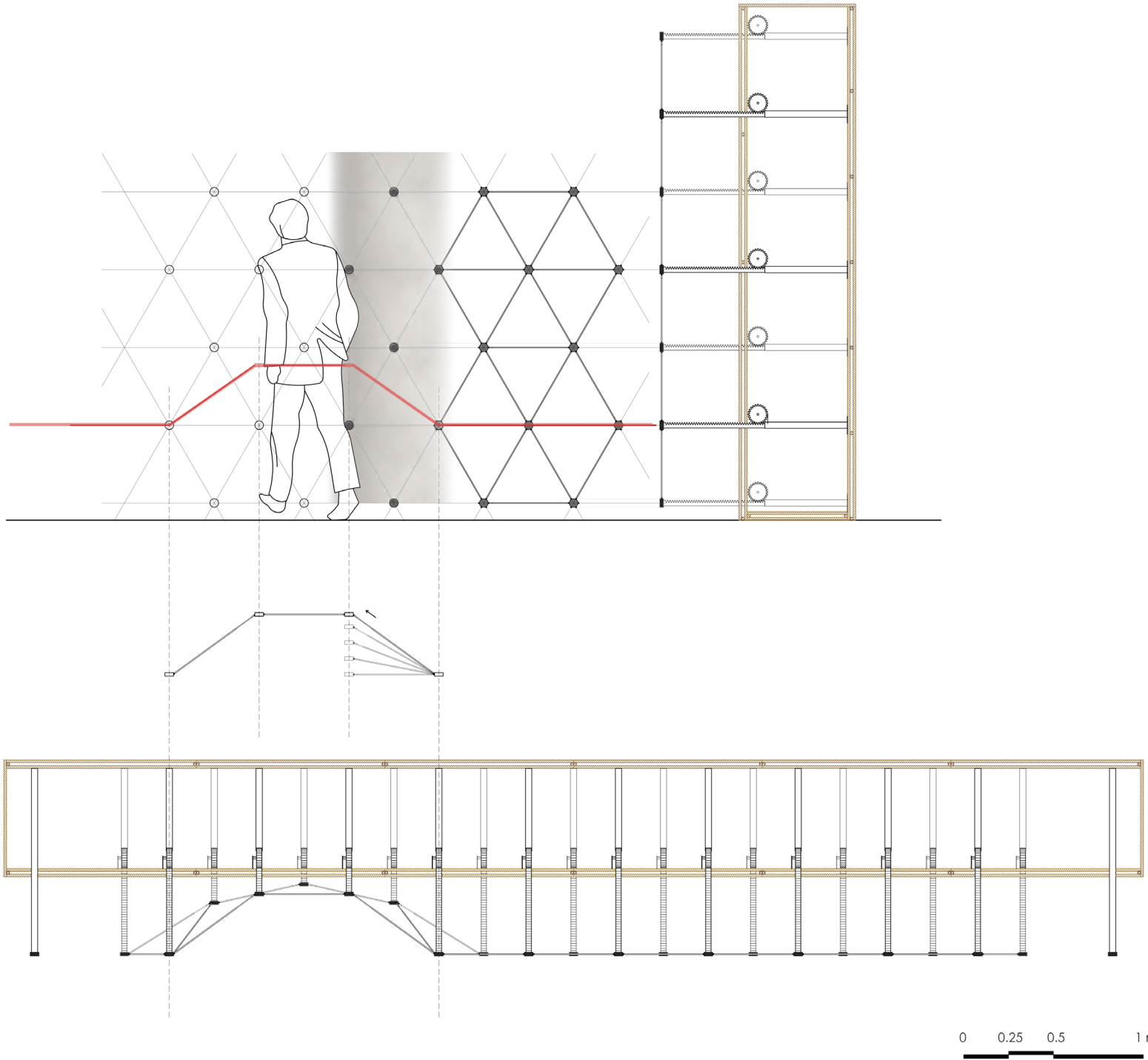
Τελική πρόταση

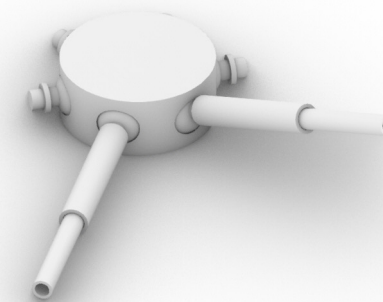
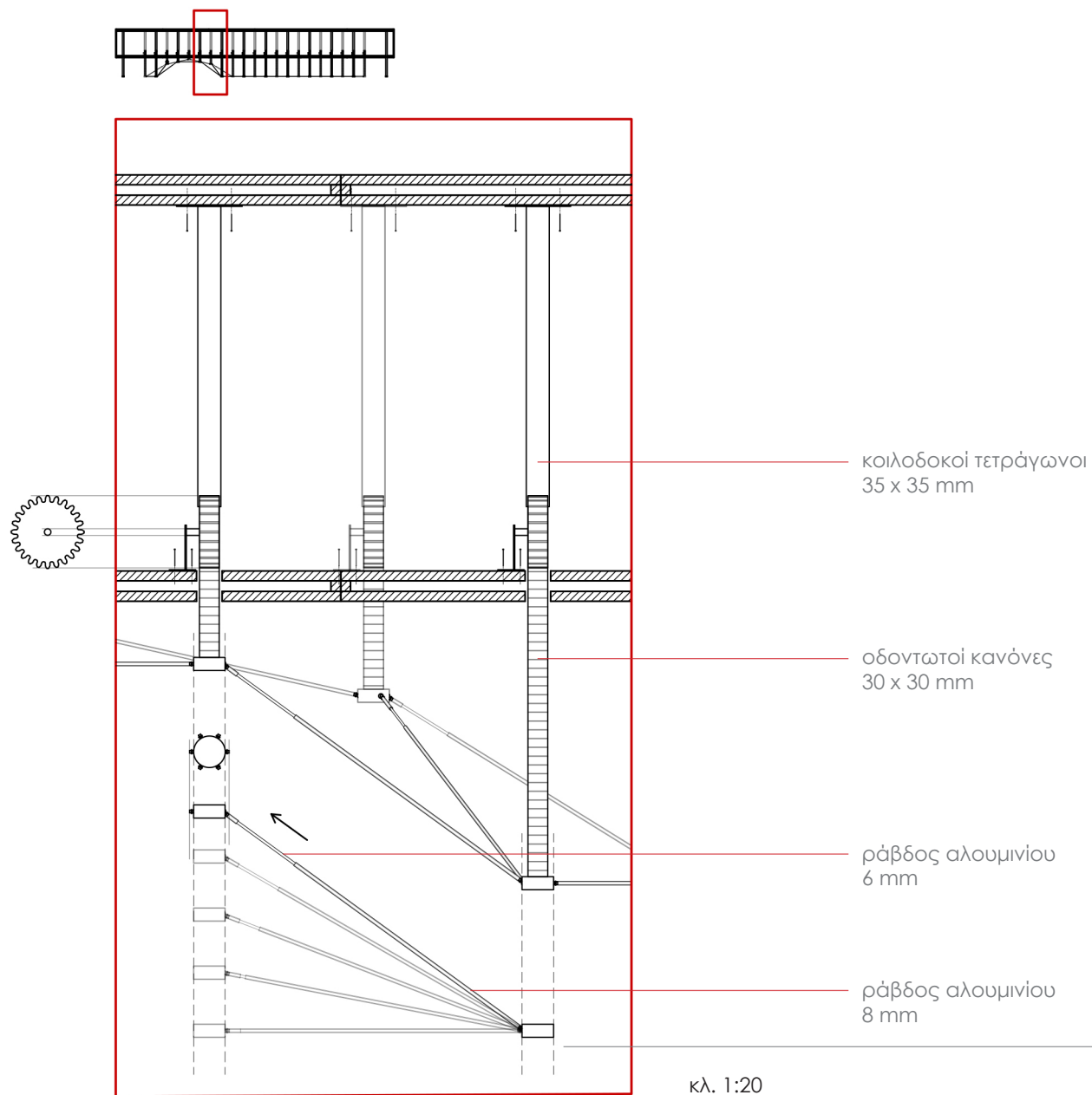
Ο διαδραστικός τοίχος



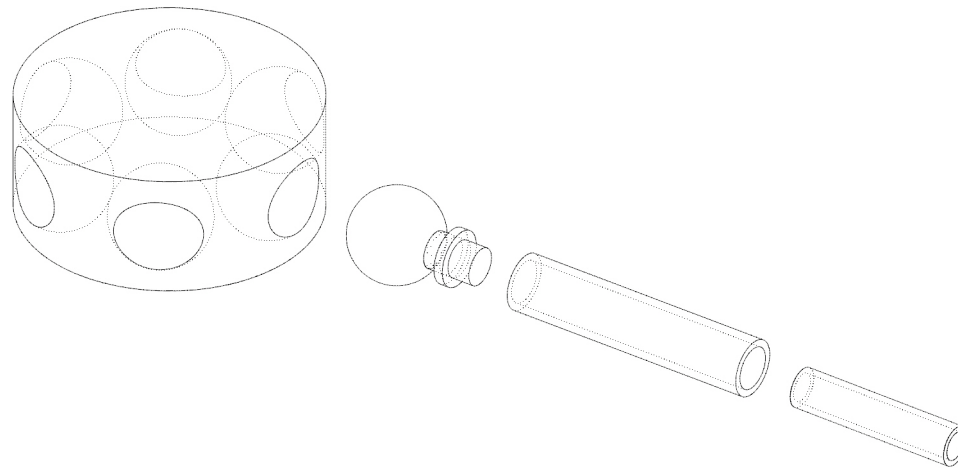
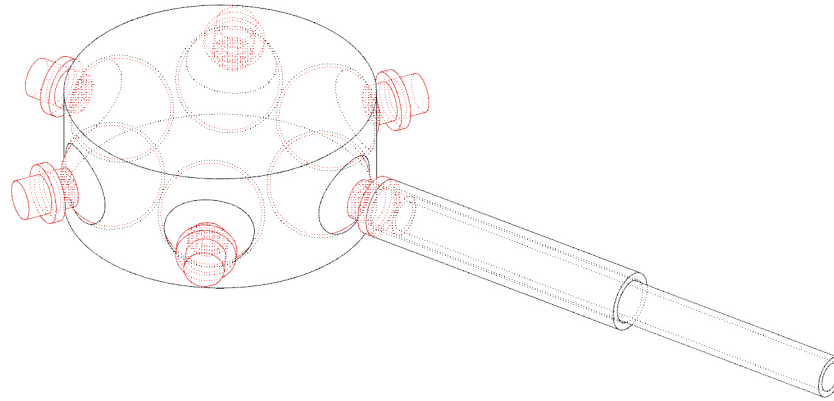


0 0.25 0.5 1 m

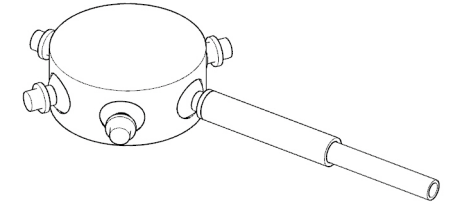




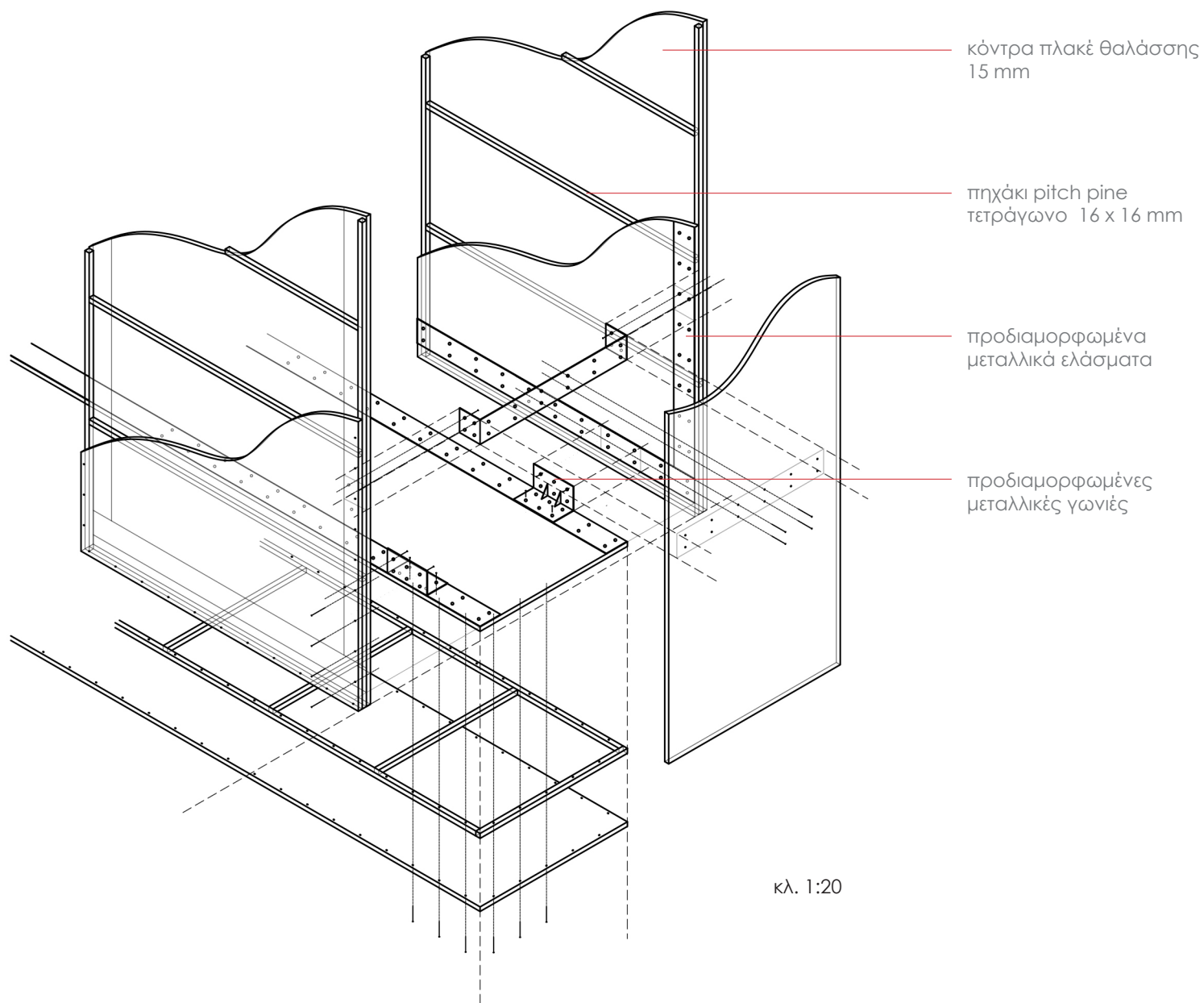
Κόμβοι με 3d εκτύπωση και ράβδοι αλουμινίου

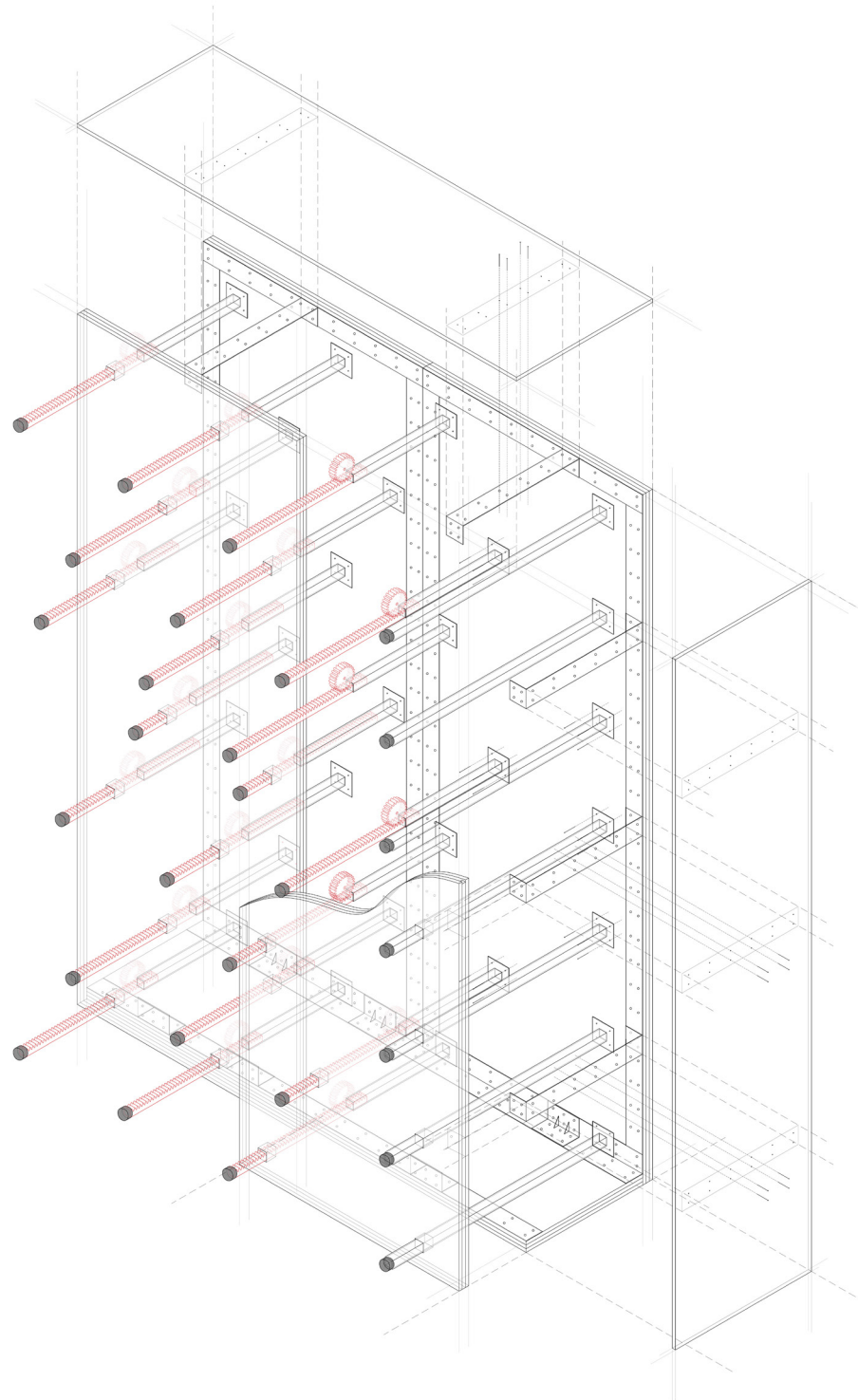


κλ. 1:1

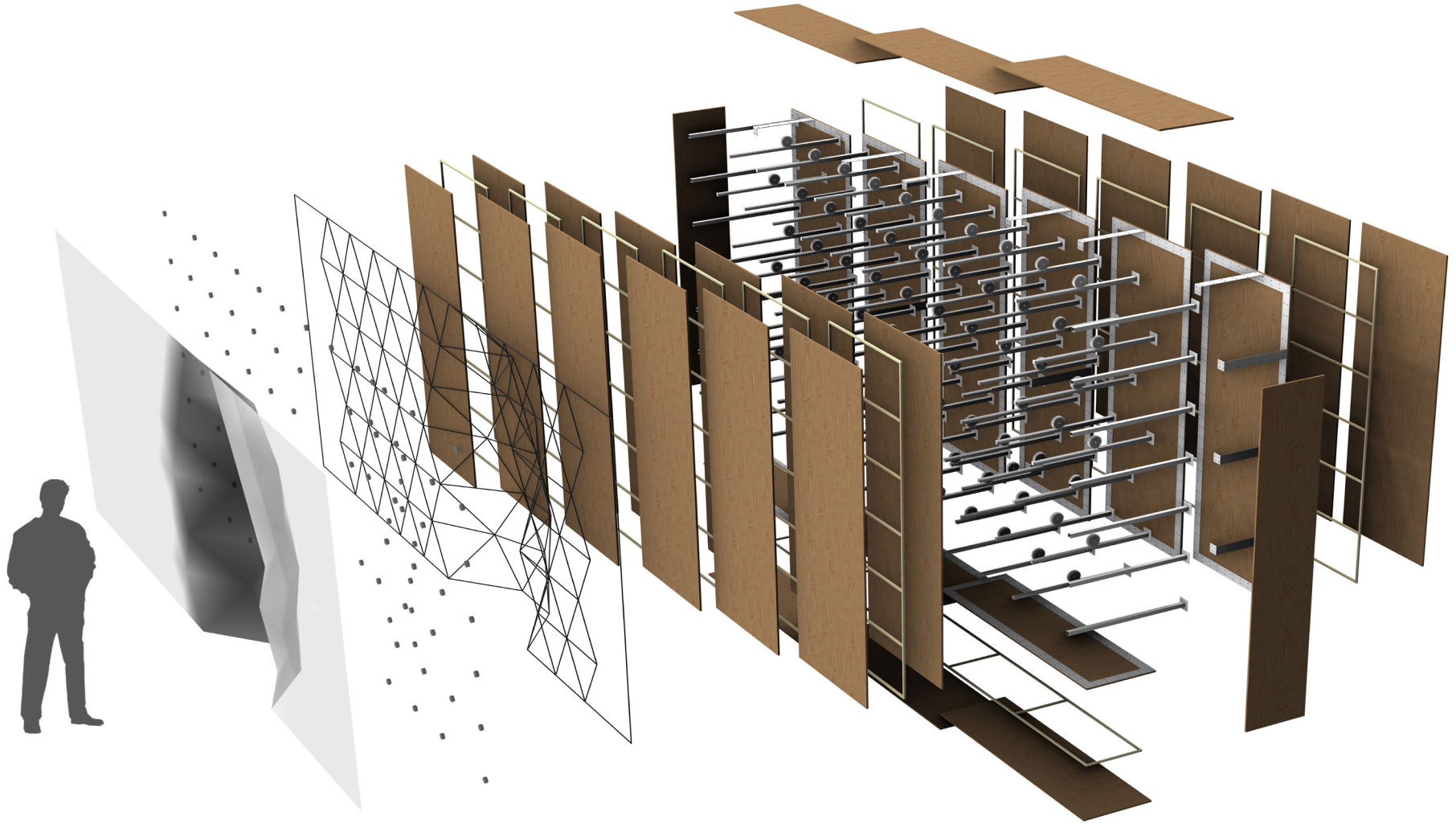


κλ. 1:2

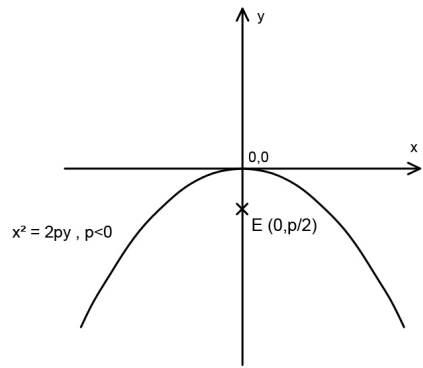




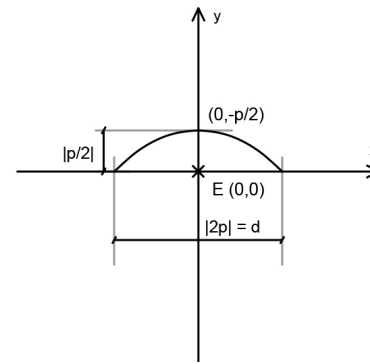
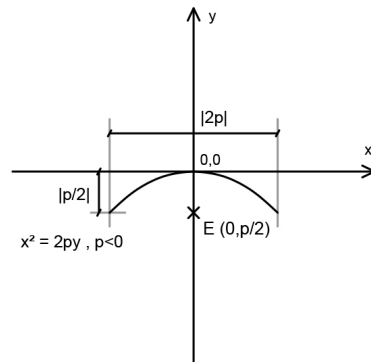
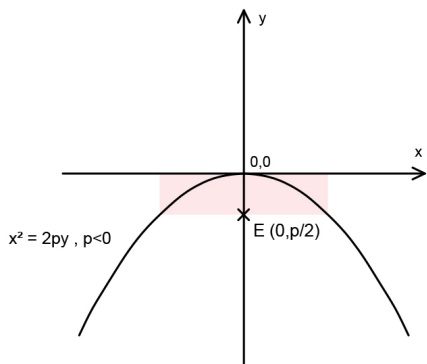
0 0.25 0.5 m



Γεωμετρική μελέτη



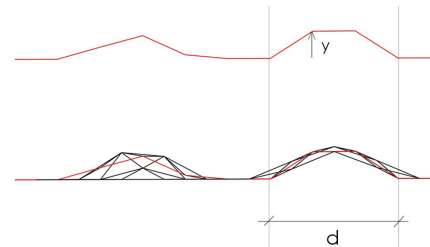
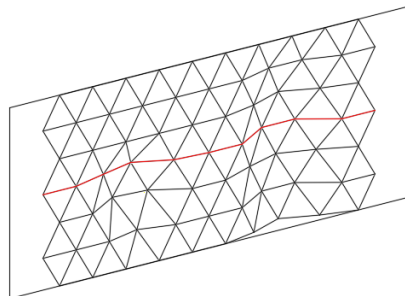
$$x^2 = 2py \Leftrightarrow y = x^2/2p, p < 0$$



Αν θεωρήσουμε ότι η παραμορφούμενη επιφάνεια έχει πλάτος $d = |2p| = -2p$ (αφού $p < 0$) τότε:

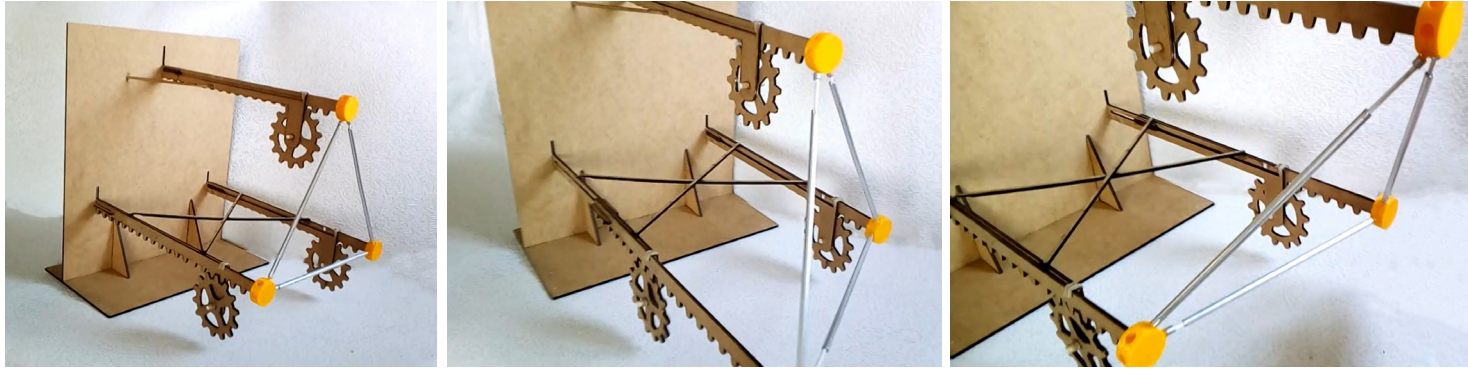
$$(1) \Rightarrow y = -x^2/|2p| + |p/2| \Leftrightarrow y = -x^2/d + d/4, \text{ όπου}$$

y είναι η οριζόντια μετατόπιση σε σχέση με την αρχική θέση του κάθε σημείου-κόμβου που βρίσκεται σε απόσταση x από το μέσο του παραμορφούμενου τμήματος (πλάτους d).

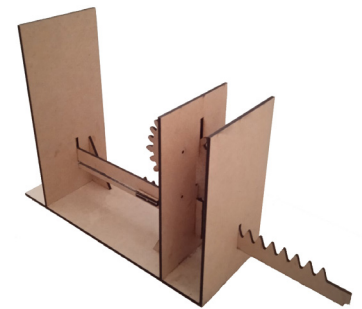
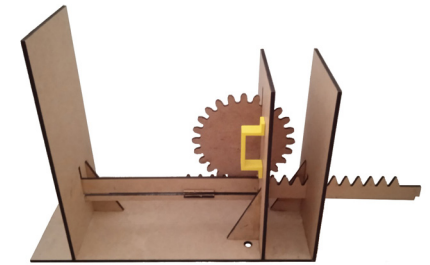
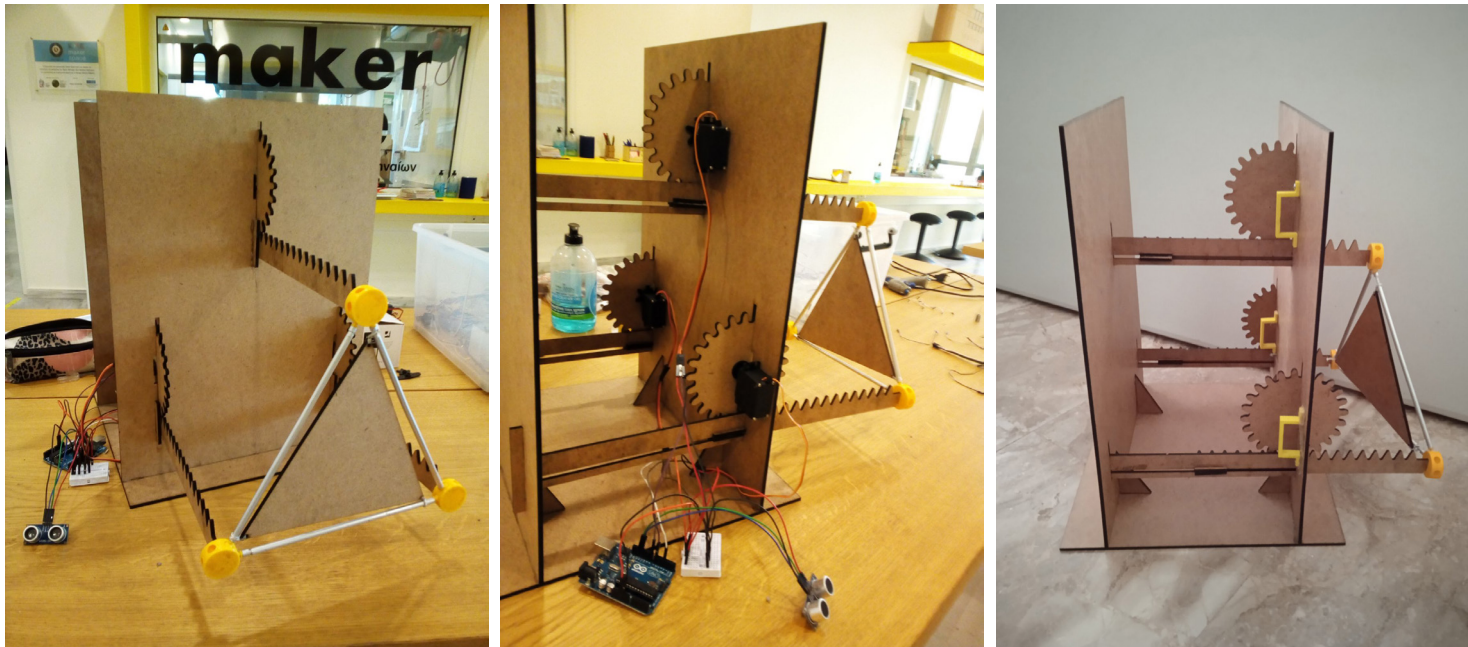


Ο τρόπος με τον οποίο θα παραμορφώνεται η κατασκευή όταν υπάρξει κίνηση ή συγκεντρωση ανθρώπων, θα μπορούσε να ακολουθεί οποιαδήποτε γεωμετρία, καθώς και να βασίζεται σε εξίσωση κάποιας καμπύλης. Εδώ μελετήθηκε η περίπτωση της παραβολής και προσαρμόστηκε η εξίσωση που την περιγράφει με τρόπο ώστε να δίνει την μετατόπιση κάθε κόμβου, εντός του τμήματος της επιφάνειας που παραμορφώνεται.

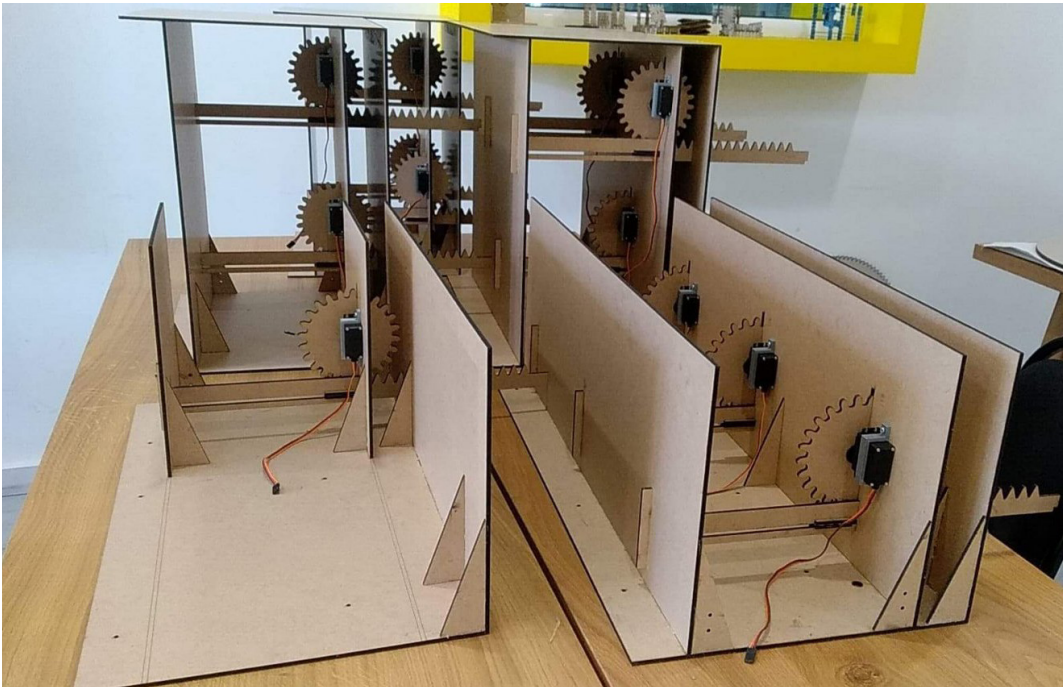
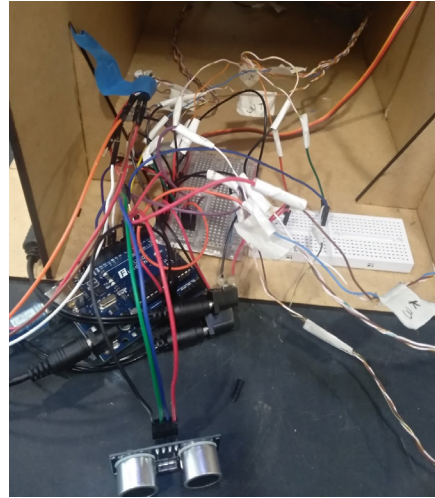
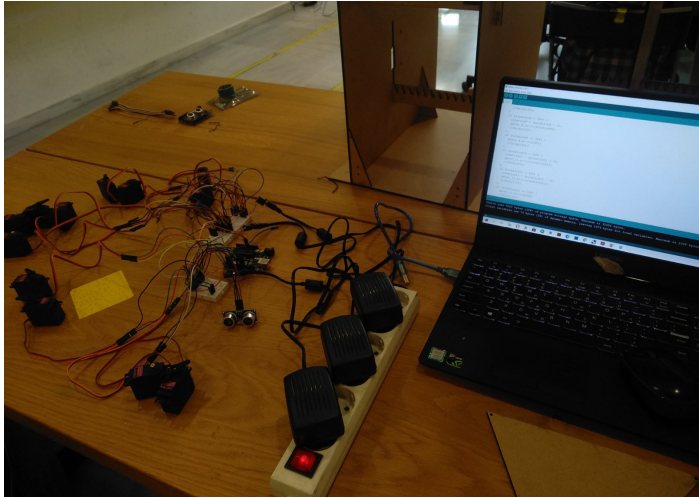
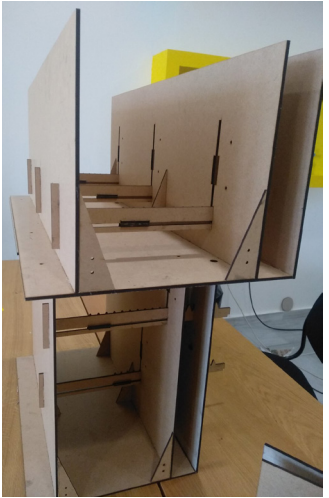
Κατασκευή

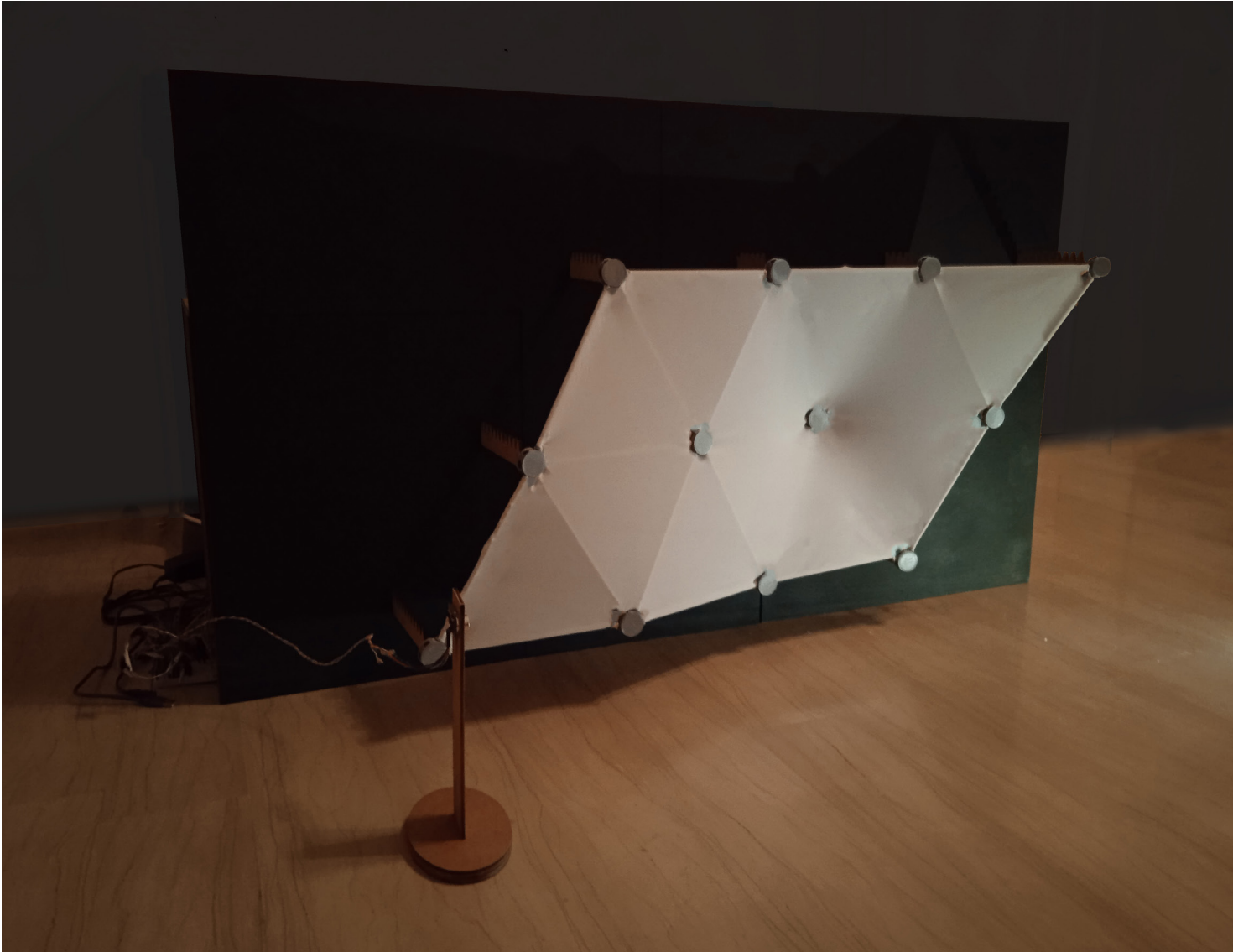


Μακέτες εργασίας για την
εύρεση του μηχανισμού

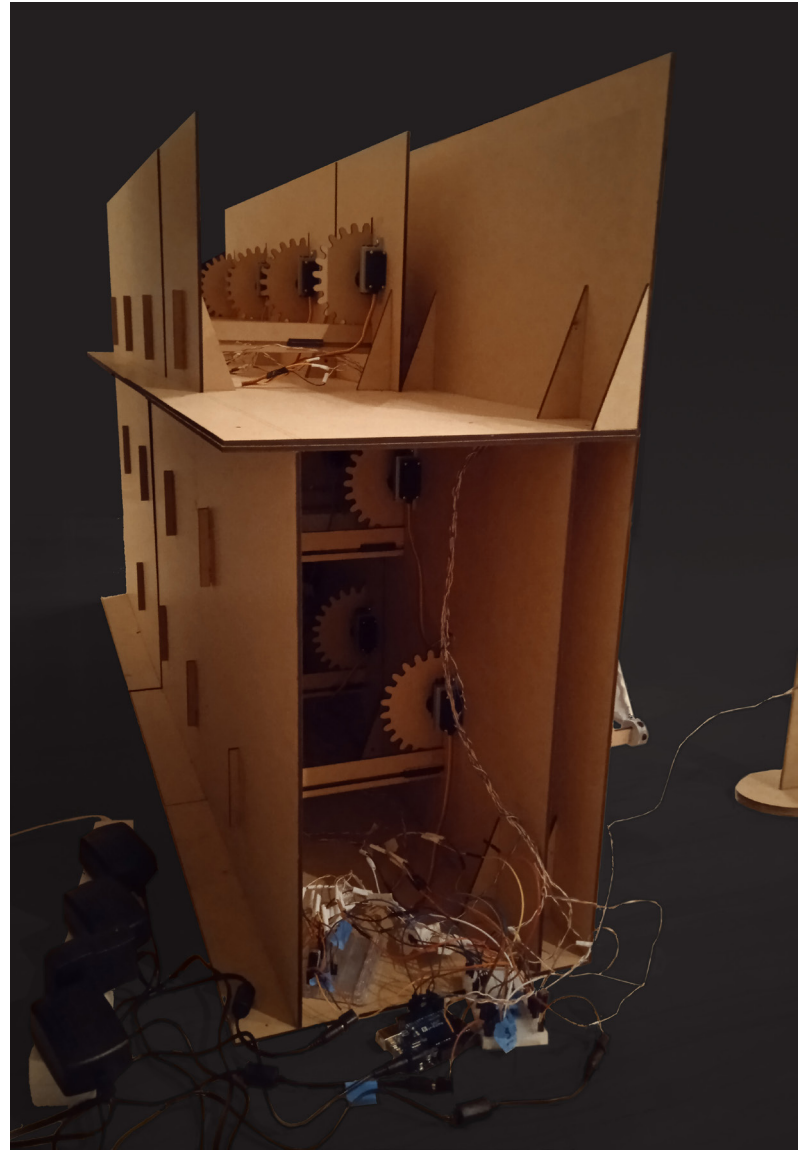
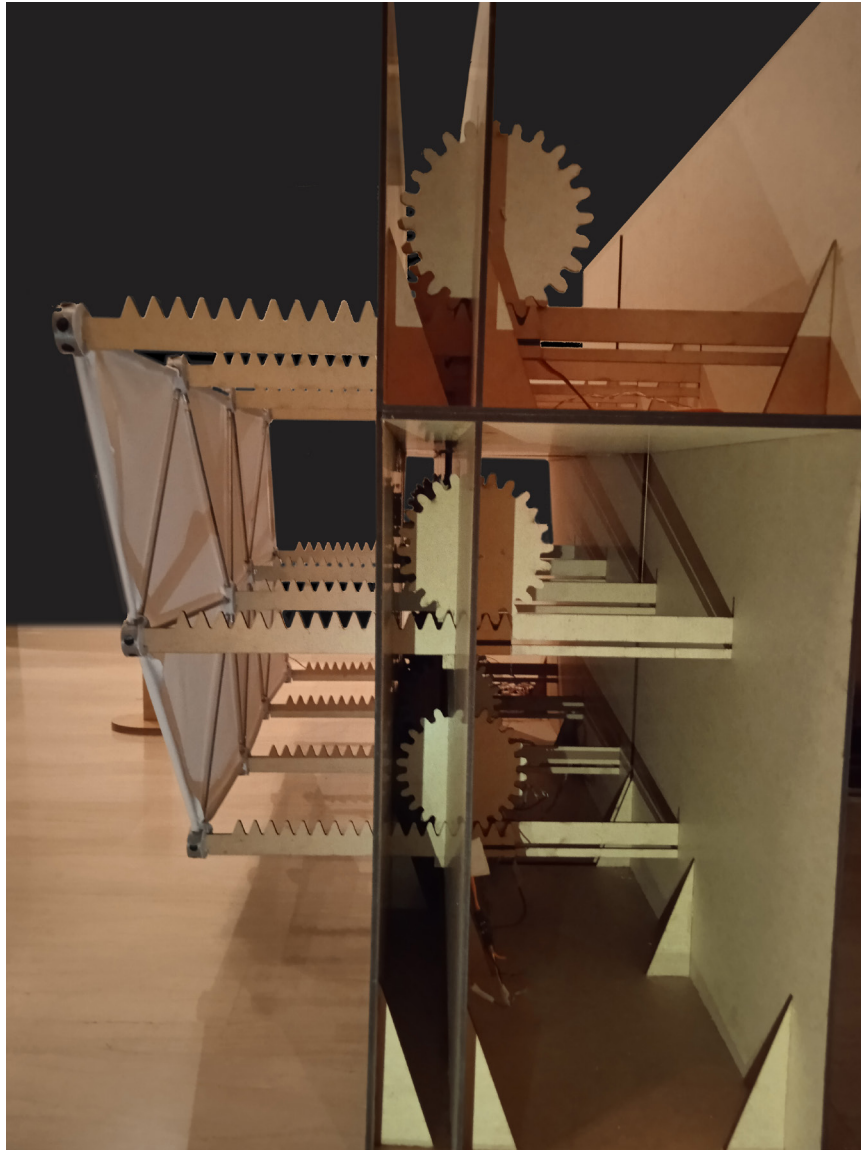


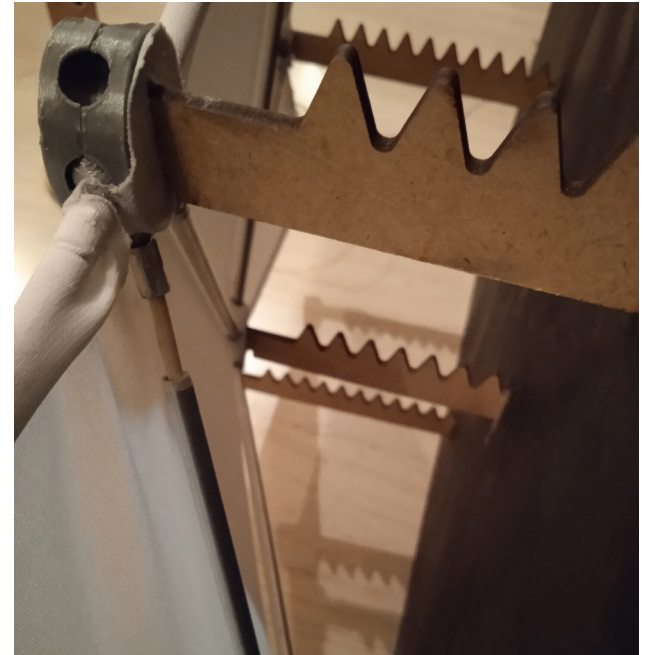
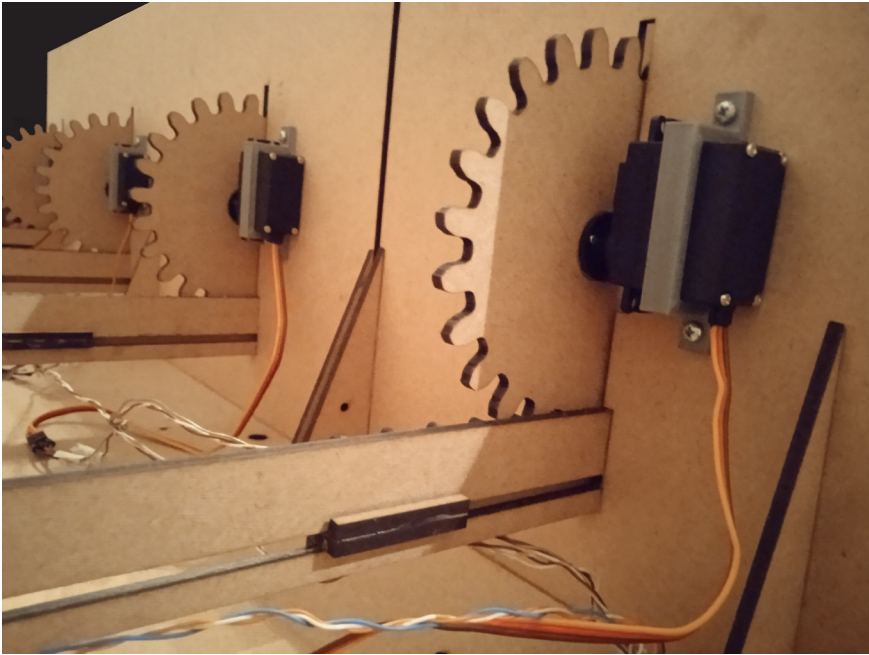
Στάδια τελικής κατασκευής





Τελική κατασκευή





Υλικά τελικής κατασκευής



Ξύλο mdf, πάχος 4mm



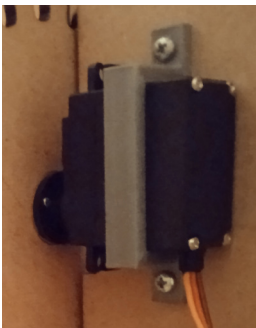
Βίδες M3



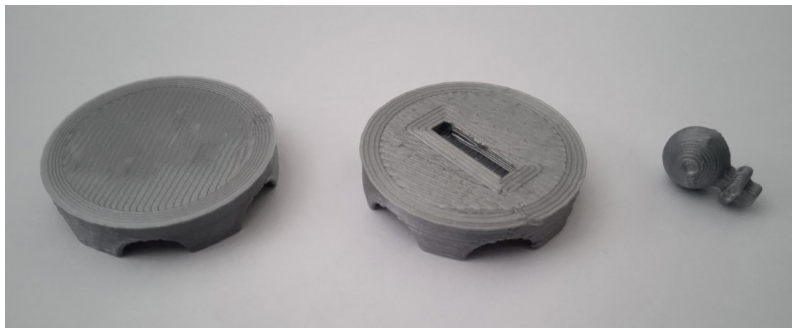
Ράβδος αλουμινίου 6mm και ξύλινο καλαμάκι 3,5mm για προσομοίωση της τηλεσκοπικής ράβδου



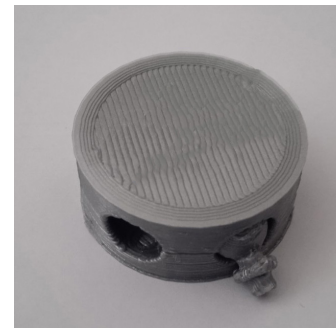
Ελαστικό ύφασμα λύκρα



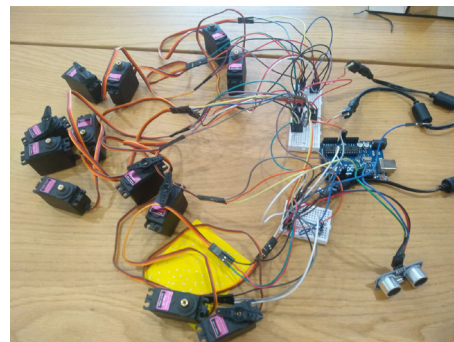
Θήκες στερέωσης των μοτέρ από εκτύπωση 3d printing



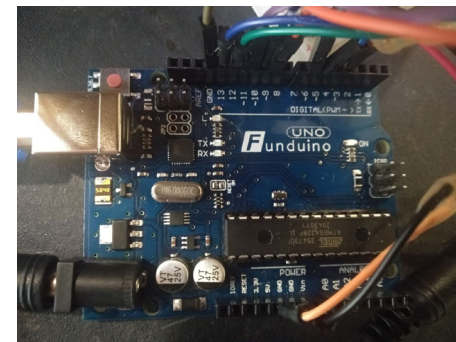
Κόμβοι από εκτύπωση 3d printing



Αισθητήρας απόστασης



12 κινητήρες servo



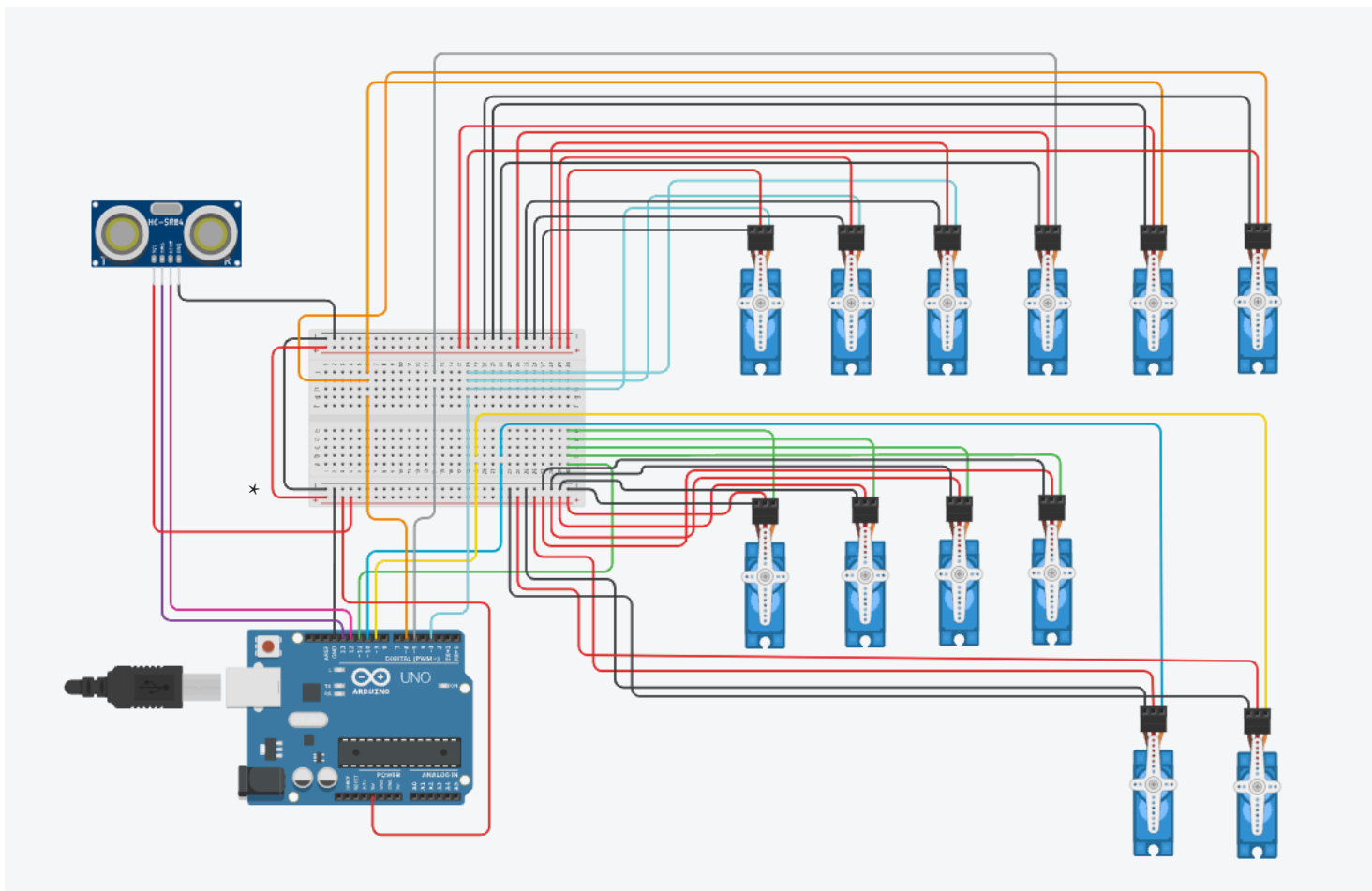
Πλακέτα UNO συμβατή με Arduino IDE

Αυτοματοποίηση

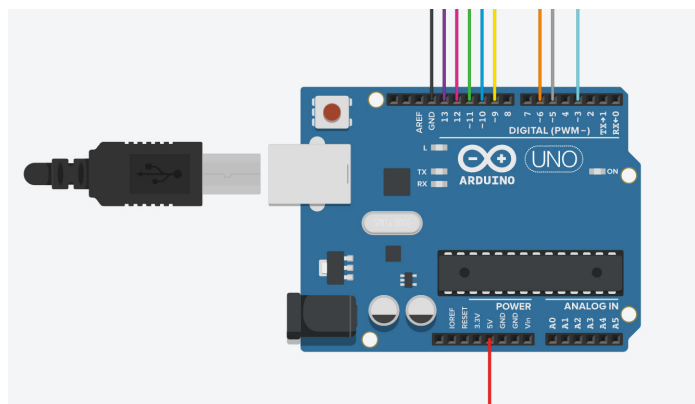
Όπως αναφέρθηκε και στην εισαγωγή, σκοπός του συστήματος που μελετήθηκε ήταν μέσα από την μεταβλητότητα της μορφής του να επιτυγχάνεται η αλληλεπίδραση με τη δημόσια ζωή. Για το λόγο αυτό, συσχετίστηκε η μεταβλητότητα της κατασκευής με το ανθρώπινο σώμα και την κίνηση αυτού, ώστε να αποτυπώνεται με αυτό τον τρόπο η ανθρώπινη δραστηριότητα στον δημόσιο χώρο. Έτσι ήταν απαραίτητη η μελέτη ενός συστήματος αυτοματοποίησης, ώστε η κίνηση του ανθρώπου να μεταγράφεται σε αντίστοιχη παραμόρφωση στην κατασκευή.

Χρησιμοποιήθηκε ο μικροελεγκτής Arduino, καθώς βασική επίσης επιδίωξη ήταν η χρήση ανοιχτού λογισμικού, το οποίο να δίνει δυνατότητα παρέμβασης και εξέλιξης του συστήματος από τον κάθε χρήστη.

Στην μακέτα κλίμακας 1:2 που κατασκευάστηκε, αποτυπώθηκε μέρος του διαδραστικού τοίχου, αποτελούμενο από 12 τρίγωνα και 12 κόμβους. Η προσομοίωση της παραμόρφωσης προσασμόστηκε στις διαστάσεις της κατασκευής. Επομένως, το τμήμα του διαδραστικού τοίχου που κατασκευάστηκε, μπορεί να λάβει 4 διαφορετικές μορφές ανάλογα με τη θέση του ανθρώπου που αντιλαμβάνεται ο αισθητήρας.



* Για την τροφοδοσία της πλακέτας και των κινητήρων χρησιμοποιήθηκαν 4 τροφοδοτικά ρεύματος, τα οποία δεν απεικονίζονται στο διάγραμμα.



Για την αυτοματοποίηση της κατασκευής χρησιμοποιήθηκαν 12 κινητήρες servo, οι οποίοι περιστρέφουν 12 γρاناζία αντίστοιχα. Με την περιστροφή των γρاناζιών, οι οδοντωτοί κανόνες κινούνται στον οριζόντιο άξονα και έτσι η κυκλική κίνηση μετατρέπεται σε ευθύγραμμη. Για την ανίχνευση της θέσης του ανθρώπου χρησιμοποιείται ένας αισθητήρας απόστασης.

Η επιθυμητή θέση των κόμβων στον οριζόντιο άξονα, ανάλογα κάθε φορά με τα δεδομένα που δίνει ο αισθητήρας, μεταφράζεται σε μοίρες για την περιστροφή των αντίστοιχων μοτέρ.

Κώδικας Arduino

```

#include <Servo.h>

int distance = 0;
int rotation3 = 0;
int rotation5 = 0;
int rotation6 = 0;
int rotation9 = 0;
int rotation10 = 0;
int rotation11 = 0;

Servo servo_3;
Servo servo_5;
Servo servo_6;
Servo servo_9;
Servo servo_10;
Servo servo_11;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  servo_3.attach(3, 500, 2500);
  servo_5.attach(5, 500, 2500);
  servo_6.attach(6, 500, 2500);
  servo_9.attach(9, 500, 2500);
  servo_10.attach(10, 500, 2500);
  servo_11.attach(11, 500, 2500);
}

void loop()
{
  rotation3 = servo_3.read();
  rotation5 = servo_5.read();
  rotation6 = servo_6.read();
  rotation9 = servo_9.read();
  rotation10 = servo_10.read();
  rotation11 = servo_11.read();
  distance = 0.01723 * readUltrasonicDistance(13, 12);
}

```

Χρήση 6 εντολών για τα servo

Η γωνία των servo την δεδομένη στιγμή

Για θέση ανθρώπου σε απόσταση μικρότερη από 30 cm από τον αισθητήρα:

στροφή servo 3 στις 180 μοίρες ->
οι κόμβοι 3 μετατοπίζονται 18 cm από την αρχική θέση

στροφή servo 5 στις 180 μοίρες ->
ο κόμβος 5 μετατοπίζεται 18 cm από την αρχική θέση

στροφή servo 10 στις 180 μοίρες ->
ο κόμβος 10 μετατοπίζεται 18 cm από την αρχική θέση

στροφή servo 6 στις 100 μοίρες ->
οι κόμβοι 6 μετατοπίζονται 10 cm από την αρχική θέση

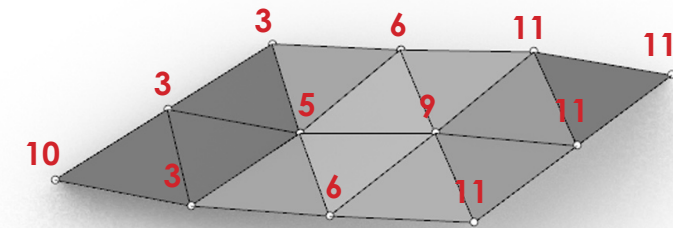
στροφή servo 9 στις 60 μοίρες ->
ο κόμβος 9 μετατοπίζεται 6 cm από την αρχική θέση

στροφή servo 11 στις 0 μοίρες ->
οι κόμβοι 11 επανέρχονται στην αρχική θέση

```

if (distance <= 30) {
  if (rotation3 < 180) {
    rotation3 = (rotation3 + 1);
    servo_3.write(rotation3);
  }
  if (rotation3 == 180) {
    servo_3.write(180);
  }
  if (rotation5 < 180) {
    rotation5 = (rotation5 + 1);
    servo_5.write(rotation5);
  }
  if (rotation5 == 180) {
    servo_5.write(180);
  }
  if (rotation10 < 180) {
    rotation10 = (rotation10 + 1);
    servo_10.write(rotation10);
  }
  if (rotation10 == 180) {
    servo_10.write(180);
  }
  if (rotation6 < 100) {
    rotation6 = (rotation6 + 1);
    servo_6.write(rotation6);
  }
  if (rotation6 > 100) {
    rotation6 = (rotation6 - 1);
    servo_6.write(rotation6);
  }
  if (rotation6 == 100) {
    servo_6.write(100);
  }
  if (rotation9 < 60) {
    rotation9 = (rotation9 + 1);
    servo_9.write(rotation9);
  }
  if (rotation9 > 60) {
    rotation9 = (rotation9 - 1);
    servo_9.write(rotation9);
  }
  if (rotation9 == 60) {
    servo_9.write(60);
  }
  if (rotation11 > 0) {
    rotation11 = (rotation11 - 5);
    servo_11.write(rotation11);
  }
  if (rotation11 == 0) {
    servo_11.write(0);
  }
}

```

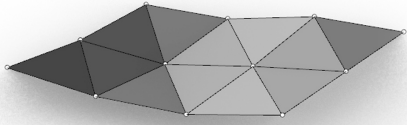


-Αν επιθυμητή γωνία περιστροφής < γωνία τη δεδομένη στιγμή -->
αυξάνει κατά 1 μοίρα τη γωνία του μοτέρ

-Αν επιθυμητή γωνία περιστροφής > γωνία τη δεδομένη στιγμή -->
μειώνει κατά 1 μοίρα την γωνία του μοτέρ

-Αν επιθυμητή γωνία περιστροφής = γωνία τη δεδομένη στιγμή -->
στρέφει το μοτέρ στην γωνία αυτή

Για θέση ανθρώπου σε απόσταση μεγαλύτερη από 30 cm και μικρότερη από 55 cm από τον αισθητήρα:

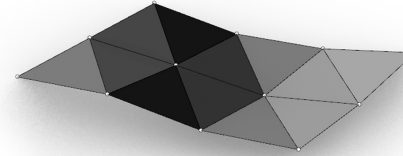


```

else if (distance > 30 && distance <= 55) {
  if (rotation3 < 100) {
    rotation3 = (rotation3 + 1);
    servo_3.write(rotation3);
  }
  if (rotation3 > 100) {
    rotation3 = (rotation3 - 1);
    servo_3.write(rotation3);
  }
  if (rotation3 == 100) {
    servo_3.write(100);
  }
  if (rotation10 < 40) {
    rotation10 = (rotation10 + 1);
    servo_10.write(rotation10);
  }
  if (rotation10 > 40) {
    rotation10 = (rotation10 - 1);
    servo_10.write(rotation10);
  }
  if (rotation10 == 40) {
    servo_10.write(40);
  }
  if (rotation5 < 180) {
    rotation5 = (rotation5 + 1);
    servo_5.write(rotation5);
  }
  if (rotation5 == 180) {
    servo_5.write(180);
  }
  if (rotation6 < 100) {
    rotation6 = (rotation6 + 1);
    servo_6.write(rotation6);
  }
  if (rotation6 > 100) {
    rotation6 = (rotation6 - 1);
    servo_6.write(rotation6);
  }
  if (rotation6 == 100) {
    servo_6.write(100);
  }
  if (rotation9 < 80) {
    rotation9 = (rotation9 + 1);
    servo_9.write(rotation9);
  }
  if (rotation9 > 80) {
    rotation9 = (rotation9 - 1);
    servo_9.write(rotation9);
  }
  if (rotation9 == 80) {
    servo_9.write(80);
  }
  if (rotation11 > 0) {
    rotation11 = (rotation11 - 5);
    servo_11.write(rotation11);
  }
  if (rotation11 == 0) {
    servo_11.write(0);
  }
}

```

Για θέση ανθρώπου σε απόσταση μεγαλύτερη από 55 cm και μικρότερη από 80 cm από τον αισθητήρα:

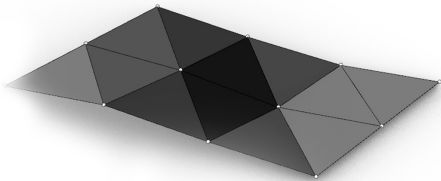


```

if (distance > 55 && distance <= 80)
  if (rotation3 > 0) {
    rotation3 = (rotation3 - 5);
    servo_3.write(rotation3);
  }
  if (rotation3 == 0) {
    servo_3.write(0);
  }
  if (rotation10 > 0) {
    rotation10 = (rotation10 - 5);
    servo_10.write(rotation10);
  }
  if (rotation10 == 0) {
    servo_10.write(0);
  }
  if (rotation5 < 80) {
    rotation5 = (rotation5 + 1);
    servo_5.write(rotation5);
  }
  if (rotation5 > 80) {
    rotation5 = (rotation5 - 1);
    servo_5.write(rotation5);
  }
  if (rotation5 == 80) {
    servo_5.write(80);
  }
  if (rotation6 < 100) {
    rotation6 = (rotation6 + 1);
    servo_6.write(rotation6);
  }
  if (rotation6 > 100) {
    rotation6 = (rotation6 - 1);
    servo_6.write(rotation6);
  }
  if (rotation6 == 100) {
    servo_6.write(100);
  }
  if (rotation9 < 180) {
    rotation9 = (rotation9 + 1);
    servo_9.write(rotation9);
  }
  if (rotation9 == 180) {
    servo_9.write(180);
  }
  if (rotation11 < 100) {
    rotation11 = (rotation11 + 1);
    servo_11.write(rotation11);
  }
  if (rotation11 > 100) {
    rotation11 = (rotation11 - 1);
    servo_11.write(rotation11);
  }
  if (rotation11 == 100) {
    servo_11.write(100);
  }
}

```

Για θέση ανθρώπου σε απόσταση μεγαλύτερη από 80 cm και μικρότερη από 110 cm από τον αισθητήρα:



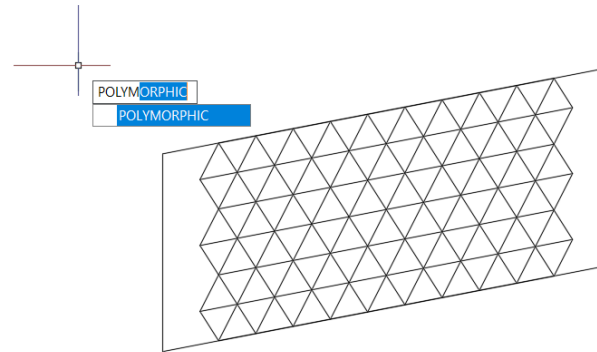
```
if (distance > 80 && distance <= 110) {  
  if (rotation3 > 0) {  
    rotation3 = (rotation3 - 5);  
    servo_3.write(rotation3);  
  }  
  if (rotation3 == 0) {  
    servo_3.write(0);  
  }  
  if (rotation10 > 0) {  
    rotation10 = (rotation10 - 5);  
    servo_10.write(rotation10);  
  }  
  if (rotation10 == 0) {  
    servo_10.write(0);  
  }  
  if (rotation5 < 60) {  
    rotation5 = (rotation5 + 1);  
    servo_5.write(rotation5);  
  }  
  if (rotation5 > 60) {  
    rotation5 = (rotation5 - 1);  
    servo_5.write(rotation5);  
  }  
  if (rotation5 == 60) {  
    servo_5.write(60);  
  }  
  if (rotation6 < 100) {  
    rotation6 = (rotation6 + 1);  
    servo_6.write(rotation6);  
  }  
  if (rotation6 > 100) {  
    rotation6 = (rotation6 - 1);  
    servo_6.write(rotation6);  
  }  
  if (rotation6 == 100) {  
    servo_6.write(100);  
  }  
  if (rotation9 < 180) {  
    rotation9 = (rotation9 + 1);  
    servo_9.write(rotation9);  
  }  
  if (rotation9 == 180) {  
    servo_9.write(180);  
  }  
  if (rotation11 < 180) {  
    rotation11 = (rotation11 + 1);  
    servo_11.write(rotation11);  
  }  
  if (rotation11 == 180) {  
    servo_11.write(180);  
  }  
}
```

Για θέση ανθρώπου έξω από το πεδίο της κατασκευής:

```
else if (distance > 110) {  
  if (rotation3 > 0) {  
    rotation3 = (rotation3 - 1);  
    servo_3.write(rotation3);  
  }  
  if (rotation3 == 0) {  
    servo_3.write(0);  
  }  
  if (rotation10 > 0) {  
    rotation10 = (rotation10 - 1);  
    servo_10.write(rotation10);  
  }  
  if (rotation10 == 0) {  
    servo_10.write(0);  
  }  
  if (rotation5 > 0) {  
    rotation5 = (rotation5 - 1);  
    servo_5.write(rotation5);  
  }  
  if (rotation5 == 0) {  
    servo_5.write(0);  
  }  
  if (rotation6 > 0) {  
    rotation6 = (rotation6 - 1);  
    servo_6.write(rotation6);  
  }  
  if (rotation6 == 0) {  
    servo_6.write(rotation6);  
  }  
  if (rotation9 > 0) {  
    rotation9 = (rotation9 - 1);  
    servo_9.write(rotation9);  
  }  
  if (rotation9 == 0) {  
    servo_9.write(rotation9);  
  }  
  if (rotation11 > 0) {  
    rotation11 = (rotation11 - 1);  
    servo_11.write(rotation11);  
  }  
  if (rotation11 == 0) {  
    servo_11.write(rotation11);  
  }  
}
```

Στροφή όλων servo στις 0 μοίρες -> όλοι οι κόμβοι επανέρχονται στην αρχική θέση

Προσομοίωση



```
(defun C:POLYMORPHIC (/ sset kwd enm els lst pn1 pn2 dst obj vc1 vc2 cnt ssnew lsnew name UcsCrt)
  (setq olderr *error* *error* polymorph-error)
  (princ "\nSelect objects (lines or points) to transform them.")
  (setq sset (ssget '((-4 . "<OR")(-4 . "<AND")(-4 . "<OR")(0 . "LINE")(0 . "POINT")(-4 . "OR>")(8 . "SURF-OBJECTS")(-4 . "AND>")(-4 . "<AND")(0 . "LWPOLYLINE")(8 . "SURFACE")
  (if sset
    (progn
      (regapp "EIRINI")
      (vl-load-com)
      (setq ssnew (ssget "X" '((-4 . "<AND")(0 . "LWPOLYLINE")(8 . "SURFACE")(-4 . "AND>"))))
      (if ssnew
        (progn
          (setq cnt 0)
          (while (setq enm (ssname ssnew cnt))
            (if (ssmemb enm sset)
              (setq lsnew (append lsnew (list enm))))
            (setq cnt (1+ cnt))
          )
          (cond
            ((> (length lsnew) 1)
              (princ "\nAmbiguous selection (many surfaces).")
              (exit)
            )
            ((= (length lsnew) 1)
              (ssdel (car lsnew) sset)
              (setq name (cdr (assoc 1000 (cdr (cdr (assoc -3 (entget (car lsnew) '("EIRINI"))))))))
              (setvar "CMDECHO" 0)
              (vl-cmdf "UCS" "Named" "Restore" name)
              (setq UcsCrt T)
            )
          )
        )
      )
    )
  )
)
```

```
(defun polymorph-error (msg)
  (setq *error* olderr olderr nil)
  (if (and (/= msg "Function cancelled")(/= msg "console break")(/= msg "quit / exit abort"))
    (princ (strcat "\nError: " msg))
  )
  (if UcsCrt
    (command-s "UCS" "")
  )
  (setvar "CMDECHO" 1)
  (gc)
  (princ)
)
```

Με βάση την παραμόρφωση σύμφωνα με την εξίσωση της παραβολής που περιγράφηκε, γράφτηκε κώδικας στη Vllsp που προσομοιώνει αυτή την μεταβλητότητα σε περιβάλλον AutoCAD. Αρχικά με την εκτέλεση ενός πρώτου κώδικα σχεδιάζεται η επιθυμητή επιφάνεια (εδώ έχει σχεδιαστεί η μορφή της κατασκευής) και η παραμόρφωση υλοποιείται με το πρόγραμμα που τρέχει σαν εντολή "polymorphic" στην γραμμή εντολών.

```

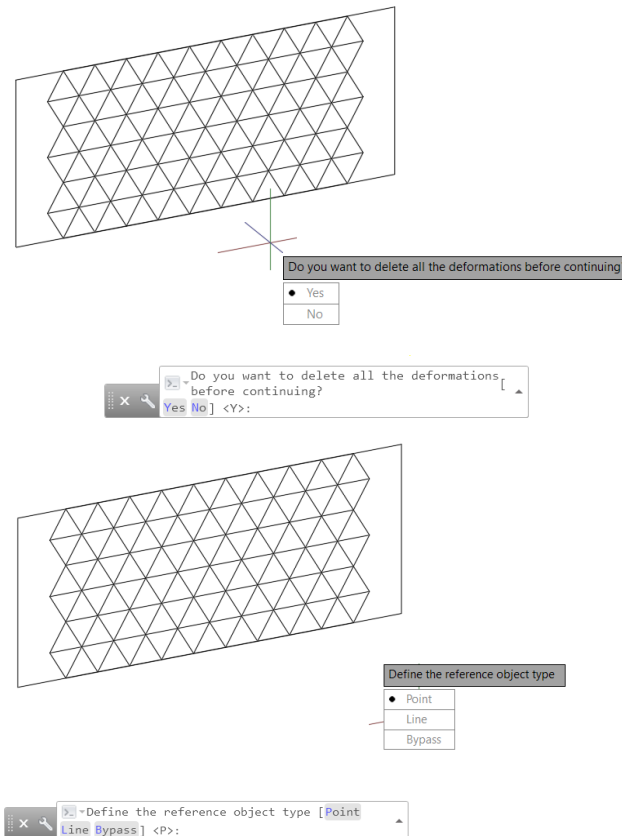
(initget "Yes No")
(setq kwd (getkword "\nDo you want to delete all the deformations before continuing? [Yes/No] <Y>: "))
(if (not kwd)(setq kwd "Yes"))
(reset)
(initget "Point Line Bypass")
(setq kwd (getkword "\nDefine the reference object type [Point/Line/Bypass] <P>: "))
(if (not kwd)(setq kwd "Point"))
(cond
  ((= kwd "Point")
   (initget 1)
   (setq pn1 (getpoint "\nInsert the reference point: ") pn1 (list (car pn1)(cadr pn1) 0))
   (entmake (list '(0 . "POINT")'(8 . "REF-OBJECTS")'(62 . 2)(cons 10 (trans pn1 1 0))))
  )
  ((= kwd "Line")
   (initget 1)
   (setq pn1 (getpoint "\nInsert the first end point of reference line: ") pn1 (list (car pn1)(cadr pn1) 0))
   (initget 1)
   (setq pn2 (getpoint pn1 "\nInsert the second end point of reference line: ") pn2 (list (car pn2)(cadr pn2) 0))
   (entmake (list '(0 . "LINE")'(8 . "REF-OBJECTS")'(62 . 2)(cons 10 (trans pn1 1 0))(cons 11 (trans pn2 1 0))))
  )
)
)

```

```

(defun reset (/ enm els pt01 pt02 pt1 pt2)
  (setq cnt 0)
  (while (setq enm (ssname sset cnt))
    (setq els (entget enm))
    pt01 (cdr (assoc 10 els))
    pt1 (trans pt01 0 1)
    pt02 (cdr (assoc 11 els))
    pt2 (if pt02 (trans pt02 0 1) nil)
    cnt (+ cnt))
  (if (= kwd "Yes")
    (progn
      (if (/= (last pt1) 0)
        (setq pt1 (trans (list (car pt1)(cadr pt1) 0) 1 0)
              els (subst (cons 10 pt1)(cons 10 pt01) els))
        )
      (if (and pt2 (/= (last pt2) 0))
        (setq pt2 (trans (list (car pt2)(cadr pt2) 0) 1 0)
              els (subst (cons 11 pt2)(cons 11 pt02) els))
        )
      (setq lst (append lst (list (last pt1)(if pt2 (last pt2) nil))))
    )
  )
  (entmod els)
)
)

```



Αρχικά, δίνεται η δυνατότητα να ανααιρεθεί όποια παραμόρφωση υπάρχει ήδη στην επιφάνεια.

Ύστερα, το τμήμα της επιφάνειας που θα παραμορφωθεί καθορίζεται από ένα σχήμα αναφοράς το οποίο σχεδιάζεται οπουδήποτε και μπορεί να είναι σημείο ή γραμμή. Μπορεί επίσης να επιλεγεί ήδη σχεδιασμένο σχήμα και να μετακινηθεί.

```

(if (/= kwd "Bypass")
  (progn
    (initget 7)
    (setq dst (getdist (if pn2 pn2 pn1) "\nDefine the radius of influence: "))
    (entmod (append (entget (entlast))(list (list -3 (list "EIRINI" (cons 1040 dst))))))
    (if (= kwd "Point")
      (entmd-all pn1 nil)
      (entmd-all pn1 pn2)
    )
  )
)

```

```

(defun entmd-all (pn1 pn2 / cnt ent el s ptst ptfn)
  (setq cnt 0)
  (while (setq ent (ssname sset cnt))
    (setq els (entget ent)
          ptst (cdr (assoc 10 els))
          ptfn (cdr (assoc 11 els))
          ptst (pnew-all ptst pn1 pn2)
          els (subst (cons 10 ptst)(assoc 10 els) els)
          cnt (1+ cnt))
    (if ptfn
      (setq ptfn (pnew-all ptfn pn1 pn2)
            els (subst (cons 11 ptfn)(assoc 11 els) els))
      )
    (entmod els)
  )
)

```

```

(defun pnew-all (pnt pt1 pt2 / zn pt0)
  (setq pnt (trans pnt 0 1))
  (if (= (last pnt) 0)
    (setq zn (if pt2 (lines pnt pt1 pt2)(points pnt pt1))
          pnt (list (car pnt)(cadr pnt) zn))
    )
  (setq pt0 (list (car pnt)(cadr pnt) 0)
        zn (if pt2 (lines pt0 pt1 pt2)(points pt0 pt1))
        pnt (if (> zn (last pnt))(list (car pnt)(cadr pnt) zn) pnt)
  )
  (trans pnt 1 0)
)

```

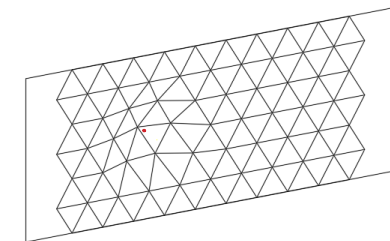
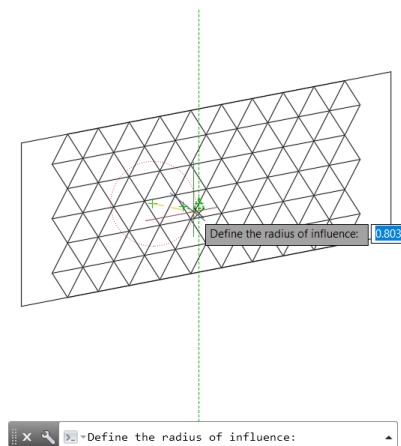
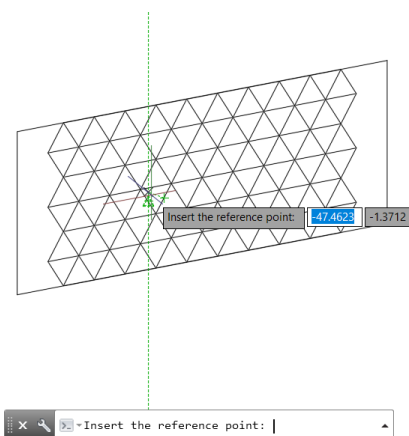
εξίσωση παραβολής

```

(defun points (pnt ptref)
  (if (< (distance pnt ptref) dst)
    (- (/ dst 2)/( expt (distance pnt ptref) 2) dst 2))
    (last pnt)
  )
)

```

Όταν το σχήμα αναφοράς είναι ένα σημείο, το τμήμα της επιφάνειας που παραμορφώνεται βρίσκεται μέσα σε έναν νοητό κύκλο, με κέντρο το σημείο αυτό και ακτίνα το μήκος που δίνει ο χρήστης ως ακτίνα επιρροής. Σε αυτό το τμήμα η επιφάνεια καμπυλώνει σύμφωνα με την εξίσωση της παραβολής.



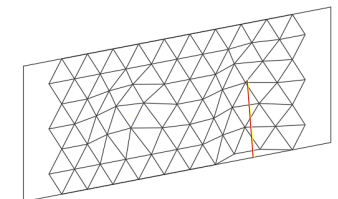
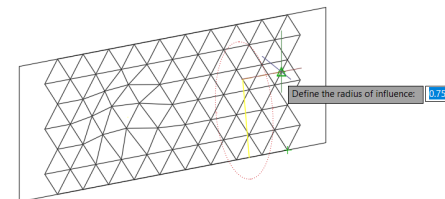
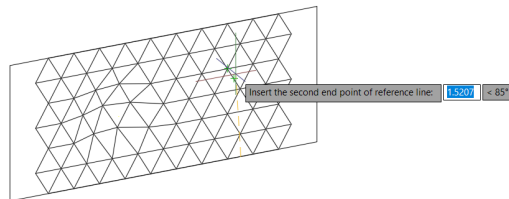
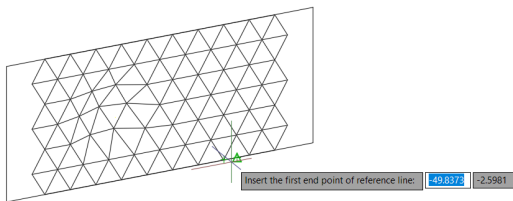
Όταν το σχήμα αναφοράς είναι μία γραμμή, το τμήμα της επιφάνειας που παραμορφώνεται βρίσκεται μέσα σε μία νοητή έλλειψη, γύρω από τη γραμμή αυτή και σύμφωνα με την ακτίνα επιρροής που δίνεται. Σε αυτό το τμήμα η επιφάνεια καμπυλώνει σύμφωνα με την εξίσωση της παραβολής.

μικρός και μεγάλος άξονας νοητής έλλειψης

εξίσωση έλλειψης

εξίσωση παραβολής

```
(defun lines (pnt pn1 pn2 / mid a b pt0 Hpt ex x y z)
  (setq mid (midpt pn1 pn2)
        a (+ (distance pn1 mid) dst)
        b dst
        pt0 (list (car pnt)(cadr pnt) 0)
        Hpt (polar pt0 (+ (angle pn1 pn2)(* 0.5 pi)) 100)
        Hpt (inters pn1 pn2 pt0 Hpt nil)
        x (distance Hpt mid)
        ex (+ (/ (* x x) a a)/( (expt (distance pt0 Hpt) 2) b b))
  )
  (if (< ex 1)
    (setq y (sqrt (- (* b b)/( * b b x x) a a)))
          z (- (/ y 2)/( (expt (distance pnt Hpt) 2) y 2))
    )
  (last pnt)
  )
)
```



Insert the first end point of reference line: |

Insert the first end point of reference line:
Insert the second end point of reference line: |

Define the radius of influence: |

Continue moving a reference object? [Yes No]

Προτάσεις εφαρμογής



Ως σκηνικό



Σε δημόσιο κτίριο



Στο μετρό



Σε πλατεία

Τέλος, το σύστημα αυτό μπορεί να αποτελέσει ένα δίκτυο που μεταφέρει την κίνηση και την δράση με τρόπο σύγχρονο ή ασύγχρονο. Το γεγονός αυτό μπορεί να το καταστήσει ικανό να δικτυώσει διαφορετικούς δημόσιους χώρους μεταξύ τους. Με άλλα λόγια, μπορεί να μετατρέψει την διάδραση σε έναν κώδικα επικοινωνίας, δίνοντας στην τελευταία μορφή και σχήμα μέσα από μια συμβολική διάσταση. Έτσι προσδίδεται ένας νέος τρόπος επικοινωνίας στο αστικό περιβάλλον, όπως για παράδειγμα σε μία πλατεία, όταν σε αυτήν ο κινούμενος τοίχος αποτυπώνει την δράση που εξελίσσεται σε μία άλλη πλατεία ακόμα και σε άλλη πόλη.

Επίλογος

Η διπλωματική αυτή δεν είχε ως τελικό αποτέλεσμα τον σχεδιασμό μίας πρότασης, αλλά ενός εργαλείου που μπορεί να οδηγήσει στον σχεδιασμό μίας σειράς προτάσεων, ανάλογα με το πώς θα αξιοποιηθεί από τον εκάστοτε συνθέτη. Έτσι, τα παραδείγματα εφαρμογών που παρουσιάστηκαν ήταν ενδεικτικά και δεν είχαν σκοπό να περιορίσουν την κατασκευή σε αυτές μόνο τις χρήσεις, αλλά να αναδείξουν τις πολλαπλές δυνατότητες του εργαλείου που προτείνεται.

Βασικό μέλημα ήταν η χρήση ανοιχτού λογισμικού, ώστε να δίνεται δυνατότητα εύκολης πρόσβασης από τον κάθε χρήστη με σκοπό την εξέλιξη του συστήματος. Επίσης, μέσα από το πρόγραμμα που γράφτηκε στην Vllisp επιχειρήθηκε η προσομοίωση της μεταβλητότητας της επιφάνειας σύμφωνα και με την επιλεγμένη εξίσωση καμπύλης, σε ένα περιβάλλον φιλικό προς τον χρήστη όπως είναι το AutoCAD.

Ως προς τον σχεδιασμό της κατασκευής, η χρήση απλών υλικών και η δυνατότητα εύκολης συναρμολόγησης αυτών σε οποιοδήποτε σημείο δημόσιου χώρου χρειαστεί, αποτέλεσε κατευθυντήρια αρχή.

Τέλος, η μεταβαλλόμενη μορφή που μελετήθηκε μπορεί να αποτελέσει τη βάση για μελλοντική έρευνα, μέσα από την εξέλιξη του συστήματος σε πεδία που δεν μελετήθηκαν στα πλαίσια της διπλωματικής αυτής. Για παράδειγμα, θα μπορούσαν να ενταχθούν και φυσικοί παράγοντες (φως, ήχος) στον τρόπο λειτουργίας του συστήματος. Συνεπώς, η προτεινόμενη κατασκευή αποτελεί την αφετηρία για πολλαπλά πεδία μελέτης, που αφορούν είτε την τοποθέτησή της σε ένα δημόσιο χώρο (μελέτη συγκεκριμένης πρότασης εφαρμογής) είτε την επέκτασή της με σκοπό την προσθήκη νέων παραμέτρων και την κάλυψη περισσότερων λειτουργιών.

