

## Αυτόματη ανακάλυψη και σύνθεση υπηρεσιών σε τοπία περιρρέουσας νοημοσύνης

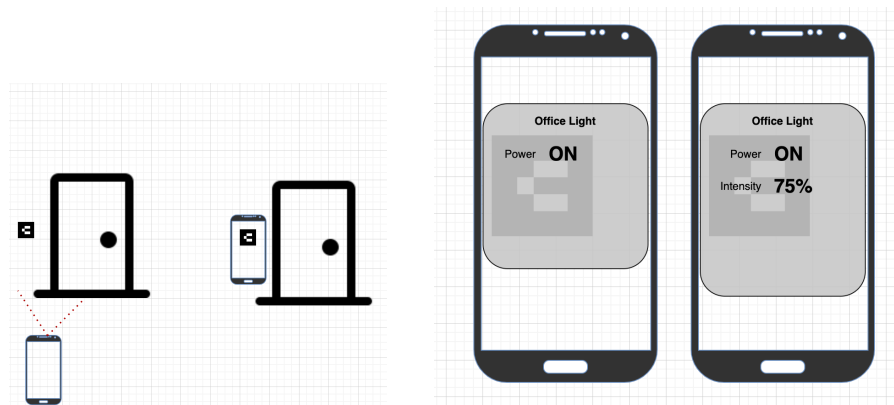
### Περιγραφή Θέματος

Στόχος της άσκησης είναι η ανάπτυξη ενός διάχυτου υπολογιστικού συστήματος, βασισμένο σε τεχνολογίες επαυξημένης πραγματικότητας.

Φανταστείτε ένα περιβάλλον όπου υπάρχουν διάφορες διαθέσιμες συσκευές που δεν έχουν κάποια απτή δυνατότητα χειρισμού, αλλά ο έλεγχός τους μπορεί να γίνει μέσω μιας κινητής συσκευής. Σαν παράδειγμα, φανταστείτε ένα διακόπτη φώτων, ο οποίος δεν έχει κουμπιά, αλλά ένα γραμμωτό κωδικό 2 διαστάσεων. Ο χρήστης πλησιάζει το κινητό του στο γραμμωτό κωδικό, βλέποντάς τον μέσω της κάμερας. Μόλις αυτός αναγνωριστεί, τότε επάνω στον γραμμωτό κωδικό εμφανίζεται ένα user-interface με το οποίο ο χρήστης μπορεί να ελέγξει τη συσκευή **με τρόπο φυσικής αλληλεπίδρασης (χειρονομίες)**.

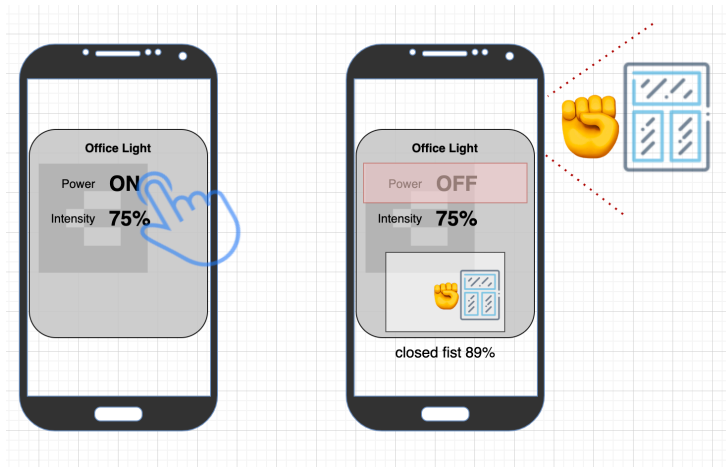
Οι δυνατότητες ελέγχου εξαρτώνται από κάποιους κανόνες. Κάποιοι χρήστες έχουν δικαιώματα πλήρους ελέγχου όλων των δυνατοτήτων, και κάποιοι άλλοι έχουν περιορισμένα δικαιώματα. Το user-interface προσαρμόζεται ώστε να εμφανίζονται μόνο οι δυνατότητες που μπορεί κάθε χρήστης να χρησιμοποιήσει.

Στην ακόλουθη απεικόνιση (Εικόνα 1), ο χρήστης πλησιάζει με το κινητό του το γραμμωτό κωδικό που βρίσκεται σε ένα τοίχο. Μόλις αυτός αναγνωριστεί, εμφανίζεται σε νέα οθόνη το σχετικό UI με περισσότερες ή λιγότερες δυνατότητες ανάλογα με τα δικαιώματα του χρήστη.



Εικόνα 1. Ο χρήστης πλησιάζει ένα γραμμωτό κωδικό με το κινητό του. Το κινητό αναγνωρίζει το γραμμωτό κωδικό και αντλεί πληροφορίες από τον server για τα στοιχεία ελέγχου τα οποία αφορά ο κωδικός. Στη συνέχεια, το κινητό εμφανίζει την κατάσταση των στοιχείων. Στην 1<sup>η</sup> οθόνη ο χρήστης δεν έχει δικαιώματα να δει όλα τα δεδομένα, ενώ αντίθετα στην 2<sup>η</sup> οθόνη, ο χρήστης έχει δικαίωμα να δει περισσότερες πληροφορίες (και στη συνέχεια να ελέγξει την κατάσταση).

Στη συνέχεια (Εικόνα 2), ο χρήστης επιλέγει στην οθόνη το στοιχείο ελέγχου με το οποίο επιθυμεί να αλληλεπιδράσει, και εκτελεί με το χέρι του χειρονομίες μπροστά στο κινητό του, που επιτρέπουν τον έλεγχο της κατάστασης της συσκευής.



Εικόνα 2. Ο χρήστης επιλέγει στην οθόνη (κλικ) το στοιχείο ελέγχου που θέλει να μεταβάλλει. Στη συνέχεια, η συσκευή με την κάμερα του πίσω μέρους της συσκευής, αναγνωρίζει τις φυσικές χειρονομίες του χρήστη που εκτελούνται με το χέρι του, και προσαρμόζει την κατάσταση κατάλληλα (εδώ, μεταβολή της κατάστασης από on σε off). Στο κάτω μέρος της οθόνης απεικονίζεται το αποτέλεσμα της αναγνώρισης της χειρονομίας (με ποσοστό εμπιστοσύνης 89%).

**Σημειώνεται ότι οι ανωτέρω απεικονίσεις είναι απολύτως ενδεικτικές και θα πρέπει ΜΟΝΟΙ σας να αποφασίσετε τη ροή της αλληλεπίδρασης μέσα στην εφαρμογή.**

Για την υλοποίηση, θα κατασκευαστεί ένα web-based σύστημα. Η εφαρμογή του χρήστη (client) είναι ένα web-app που στηρίζεται στη χρήση των βιβλιοθηκών:

- Zebra Crossing (ZXing) για την αναγνώριση των γραμμωτών κωδικών δύο διαστάσεων <https://github.com/zxing-js/library>
- MediaPipe για την αναγνώριση των χειρονομιών [https://developers.google.com/mediapipe/solutions/vision/gesture\\_recognizer/web\\_js](https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer/web_js)

Η βιβλιοθήκη MediaPipe έχει την ικανότητα by default να αναγνωρίζει λίγες μόνο προκαθορισμένες χειρονομίες, οι οποίες είναι:

- "None" (καμία γνωστή χειρονομία)
- "Closed\_Fist", 🤞
- "Open\_Palm", 🤚
- "Pointing\_Up", 🙌
- "Thumb\_Down", 👎
- "Thumb\_Up", 👍
- "Victory", ✌️
- "ILoveYou", 🙏

Προτείνεται λοιπόν να σκεφθείτε το widget ελέγχου που ταιριάζει «φυσικά» με την κατάλληλη χειρονομία. Για παράδειγμα, ένα toggle switch (on/off) μπορεί να συσχετίζεται με τις χειρονομίες Open Palm (ON) και Closed Fist (OFF). Αντίστοιχα ένας ρυθμιστής κλίμακας (slider) μπορεί να συσχετίζεται με τις χειρονομίες Thumb Up (increase value by 1) ή Thumb Down (decrease value by 1), ή ένα button να συσχετίζεται με τη χειρονομία Pointing Up.

Επιπρόσθετα, θα πρέπει να υπάρχει μια back-end υποδομή η οποία διατηρεί τα γνωρίσματα των συσκευών, τους λογαριασμούς χρηστών και τα δικαιώματά τους, και αναλαμβάνει τη σύνθεση του user interface με δυναμικό τρόπο. Κάθε συσκευή συσχετίζεται με κάποιον marker που την ταυτοποιεί μοναδικά. Στη συνέχεια, οι δυνατότητες ελέγχου κάθε συσκευής αναπαρίστανται στον server με κατάλληλη δομή δεδομένων. Κατά την αναγνώριση ενός marker, η εφαρμογή client αποστέλλει το σχετικό αίτημα στον server ώστε να λάβει πίσω μια κατάλληλη δομή δεδομένων (JSON) με τις δυνατότητες ελέγχου που έχει ο χρήστης.

Στη συνέχεια, ανάλογα με το είδος του κάθε χειριστηρίου (π.χ. toggle, slider κ.α.) ο client συνθέτει και απεικονίζει το κατάλληλο UI στοιχείο, δημιουργώντας παράλληλα και τον κατάλληλο κώδικα ώστε όταν ο χρήστης αλλάζει το state μιας λειτουργίας, αυτή να αποθηκεύεται κατάλληλα στο backend.

## Παραδοτέα και χρονοδιάγραμμα

### Εξεταστική Ιουνίου 2023

Τα παραδοτέα του project θα είναι σε 3 φάσεις.

#### Φάση 1 – Σχεδιασμός εφαρμογής (ως 22.5.2023)

- Αναφορά σχεδιασμού εφαρμογής (τεκμηρίωση σχεδιαστικών επιλογών, αρχιτεκτονική εφαρμογής)
- Πρωτότυπος / δοκιμαστικός κώδικας για το backend και την client εφαρμογή

#### Φάση 2 – Ανάπτυξη εφαρμογής (ως 19.6.2023)

- Πλήρης κώδικας εφαρμογής
- Αναφορά τεκμηρίωσης της εφαρμογής
  - Τελικός σχεδιασμός εφαρμογής
  - Τεκμηρίωση τελικής εφαρμογής με
    - Screenshots
    - Παραδείγματα κώδικα
    - Αποτελέσματα δοκιμών λειτουργίας

#### Φάση 3 – Αναφορά δοκιμαστικής λειτουργίας (3.7.2022)

- Ενημέρωση της αναφοράς σχεδιασμού εφαρμογής ώστε να περιέχει τυπική αξιολόγηση με τουλάχιστον 5 χρήστες
  - Πρωτόκολλο αξιολόγησης
  - Ευρήματα
  - Προτάσεις για βελτίωση και μελλοντικές επεκτάσεις

### Εξέταση Σεπτεμβρίου 2023

Τα παραδοτέα του project θα είναι σε μία φάση.

#### Φάση 1 – Υποβολή τελικού παραδοτέου (23/9/2023)

Θα υποβάλλετε μία εκτενή αναφορά που περιέχει:

- Αρχικός σχεδιασμός εφαρμογής (τεκμηρίωση σχεδιαστικών επιλογών, αρχιτεκτονική εφαρμογής)
- Τεκμηρίωσης της τελικής εφαρμογής
- Λειτουργικές δοκιμές τελικής εφαρμογής
- Αξιολόγηση με 5 τουλάχιστον χρήστες

#### Εκπόνηση

Κάθε φοιτητής μπορεί να εργαστεί μόνος του, ή να εργαστείτε σε ομάδες το πολύ 2 ατόμων. Σε περίπτωση ομαδικής εργασίας, θα αναφέρετε ρητά στο τελικό παραδοτέο πλήρη κατάλογο με τις δραστηριότητες με τις οποίες ασχολήθηκε το κάθε μέλος.

Κάθε άτομο ή ομάδα θα πρέπει να υποβάλλει τα σχετικά παραδοτέα εγκαίρως, όπως περιγράφονται ανωτέρω.

## Χρήσιμες πληροφορίες

Όσον αφορά το back-end, θα χρησιμοποιήσετε την πλατφόρμα ThingsBoard. Μπορείτε να το στήσετε σε δικό σας υπολογιστή, ή να ζητήσετε από το διδάσκοντα να στηθεί κατάλληλος λογαριασμός στο server του εργαστηρίου μας.

Η φιλοσοφία ολοκλήρωσης ακολουθεί την παρακάτω λογική:

- 1) Μέσω του client, ο χρήστης ταυτοποιείται προς το server (login).
- 2) Ο client αναγνωρίζει τον 2D κωδικό στον οποίο εμπεριέχεται το id μιας συσκευής ελέγχου, και στέλνει σαν request στον back-end server το id του κωδικού και το userid του χρήστη.

- 3) Ο server απαντά στον client με δομημένα δεδομένα (π.χ. JSON) που περιέχουν όλες τις πληροφορίες που χρειάζονται για να χτίσει ο client το user interface. Αυτές περιέχουν το είδος του widget και πληροφορίες για το τρέχον state του (τρέχουσα τιμή).
- 4) Ο client δημιουργεί δυναμικά όλο το user interface και το δείχνει στην οθόνη.
- 5) Ο χρήστης επιλέγει ένα widget με το οποίο θα αλληλεπιδράσει
- 6) Η συσκευή του χρήστη αναγνωρίζει τις σχετικές φυσικές αλληλεπιδράσεις και προσαρμόζει την κατάσταση (τιμή) του widget.
- 7) Σε κάθε αλληλεπίδραση με το χρήστη (π.χ. ενεργοποίηση ενός toggle), ο client στέλνει στον server το σχετικό request για την αλλαγή της τρέχουσας τιμής. Αν η αλλαγή αποτύχει (δηλαδή αν ο client δεν λάβει κάποιο OK πίσω), το state του UI επιστρέφει στην προηγούμενη τιμή.

#### Bonus

Η βιβλιοθήκη MediaPipe αναγνωρίζει μόνο 7 κλάσεις χειρονομιών. Αν επιθυμείτε, μπορείτε να εκπαιδεύσετε μόνοι σας νέα μοντέλα χειρονομιών για να προσφέρετε νέες ικανότητες χειρισμού στο σύστημα<sup>1</sup>.

Οι ομάδες που θα επιτύχουν κάτι τέτοιο, θα έχουν + 2 μονάδες στον τελικό βαθμό του μαθήματος ανεξάρτητα από την επίδοσή τους σε project και γραπτή εξέταση, με μέγιστο πάντα το 10.

---

<sup>1</sup> [https://developers.google.com/mediapipe/solutions/vision/gesture\\_recognizer/customize](https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer/customize)