**Use-cases-v0.1**

BLOCKO

**Μέλη Ομάδας:**

**Όνομα**: Κωνσταντίνος

**Επώνυμο**: Καρακίτσος

**ΑΜ**: 1093381

**Όνομα**: Βασίλειος

**Επώνυμο**: Μιχανετζής

**ΑΜ:** 1093433

**Όνομα**: Νυμφοδώρα

**Επώνυμο:** Μπουζιάνη

**ΑΜ**: 1097429

**Όνομα**: Μιχαήλ

**Επώνυμο:** Σοκολάκης

**ΑΜ**: 1072589

**Όνομα**: Σωτήριος

**Επώνυμο**: Χασιώτης

**ΑΜ**: 1093510

**Εργαλεία :**

Τα εργαλεία που χρησιμοποιήσαμε για την σύνταξη των τεχνικών κειμένων είναι το Microsoft Word και για την δημιουργία των διαγραμμάτων το draw.io.

**Use case Diagram:**

****

**Αναλυτική περιγραφή των use cases**

**Use case**: Δήλωση και αρχικοποίηση μεταβλητών

**Editor:** Κωνσταντίνος Καρακίτσος

**Peer reviewer:** Βασίλειος Μιχανετζής, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης, Σωτήριος Χασιώτης

**Βασική ροή:**

**1**. Ο χρήστης επιλέγει το πεδίο δημιουργίας νέας μεταβλητής.

**2**. Το σύστημα εμφανίζει στον χώρο των δηλώσεων των μεταβλητών ένα μπλοκ με δύο κενές θέσεις εισαγωγής.

**3**. Ο χρήστης ορίζει το όνομα της μεταβλητής.

**4**. Το σύστημα ελέγχει την σύνταξη του ονόματος και διαπιστώνει ότι είναι σωστή.

**5**. Το σύστημα ελέγχει αν υπάρχει άλλη μεταβλητή με το ίδιο όνομα και διαπιστώνει ότι δεν υπάρχει.

**6**. Ο χρήστης εισάγει τιμή.

**7**. Το σύστημα ελέγχει αν η τιμή είναι ακέραια και διαπιστώνει ότι είναι.

**8**. Το σύστημα δημιουργεί ένα μπλοκ στον χώρο του προγράμματος για να μπορεί να χρησιμοποιηθεί η μεταβλητή.

**Εναλλακτική Ροή 1:**

**4.1.1.** Το σύστημα διαπιστώνει ότι η σύνταξη είναι λάθος.

**4.1.2.** Το σύστημα κάνει κόκκινο το μπλοκ.

**Εναλλακτική Ροή 2:**

**7.2.1.** Το σύστημα διαπιστώνει ότι η τιμή που έχει δοθεί δεν ανήκει στις τιμές που δέχεται το πρόγραμμα, δηλαδή δεν είναι ακέραια.

**7.2.2.** Το σύστημα κάνει κόκκινο το μπλοκ.

**Use case**: Δημιουργία μπλοκ εντολών

**Editor:** Μιχαήλ Σοκολάκης

**Peer reviewer:** Βασίλειος Μιχανετζής, Σωτήριος Χασιώτης, Κωνσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη

**Βασική Ροή:**

**1.** O χρήστης κάνει κλικ στο κουμπί δημιουργίας του τύπου block που επιθυμεί.

**2**. Το σύστημα δημιουργεί το μπλοκ που επέλεξε ο χρήστης.

**3**. Το σύστημα αλλάζει το χρώμα του μπλοκ σε κόκκινο.

**4**. Ο χρήστης εισάγει τις τιμές και τις μεταβλητές.

**5**. Το σύστημα ελέγχει την μεταβλητή και την τιμή της ως προς την εγκυρότητα.

**6**. Το σύστημα ελέγχει αν η μεταβλητή είναι δηλωμένη.

**7.** Το σύστημα κάνει το χρώμα του μπλοκ πράσινο, ως ένδειξη ότι δεν βρέθηκαν λάθη κατά τον έλεγχο στα βήματα 4 και 5.

**8.** Ο χρήστης παίρνει το νέο του μπλοκ και το τοποθετεί στην βασική ροή του προγράμματος ή του function του.

**Εναλλακτική Ροή 1:**

**6.1.1.** Το σύστημα διαπιστώνει ότι ο χρήστης έχει εισαγάγει μεταβλητή η οποία δεν έχει δηλωθεί.

**6.1.2.** Το σύστημα κάνει κόκκινο το μπλοκ.

**6.1.3**. Ο χρήστης εισάγει εκ νέου μεταβλητή.

**6.1.4.** Συνέχεια από το βήμα 4 της βασικής ροής.

**Εναλλακτική Ροή 2:**

**8.2.1.**Ο χρήστης αντί να τοποθετήσει το νέο του μπλοκ στην βασική ροή του προγράμματος ή του function του, πατάει ξανά το κουμπί δημιουργίας μπλοκ.

**8.2.2.** Το σύστημα ενημερώνει το χρήστη, ότι πρέπει να πάρει το νέο του μπλοκ από το πεδίο δημιουργίας πριν φτιάξει άλλο.

**Use case**: Compile

**Editor:** Σωτήριος Χασιώτης

**Peer reviewer:** Βασίλειος Μιχανετζής, Κωνσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης

**Βασική Ροή:**

**1**. Ο χρήστης πατάει το κουμπί compile εφόσον έχει ολοκληρώσει την δημιουργία του κώδικα του.

**2**. Το σύστημα ελέγχει αν υπάρχουν start και end blocks.

**3.** Το σύστημα διαπερνά την λίστα με τα blocks και ελέγχει αν κάποιο από αυτά περιέχει συντακτικό λάθος στο όνομα του ή δεν έχει δηλωθεί.

**4**. Το σύστημα κάνει έλεγχο για δομές όπως if και βρόγχους while για να βρει τα αντίστοιχα end τους.

**5**. Το σύστημα ελέγχει για function blocks εφόσον υπάρχουν και διαπερνά, με τον ίδιο τρόπο του βήματος 3, την λίστα τους επαναλαμβάνοντας το δεύτερο, το τρίτο και το τέταρτο βήμα .

**6**. Εφόσον όλα είναι έγκυρα το σύστημα εμφανίζει στον χρήστη αντίστοιχο μήνυμα επιτυχίας.

**7**. Το σύστημα εμφανίζει στον χρήστη την επιλογή να τρέξει το πρόγραμμα του.

**Εναλλακτική Ροή 1:**

**2.1.1.** Το σύστημα διαπιστώνει ότι ο χρήστης δεν έχει εισάγει start ή/και end block

**2.1.2.** Το σύστημα ενημερώνει το χρήστη για απουσία start ή/και end block.

**Εναλλακτική Ροή 2:**

**3.2.1.** Το σύστημα διαπιστώνει ότι ο χρήστης έχει κάνει συντακτικό λάθος κατά τη δημιουργία κάποιου block.

**3.2.2.** Το σύστημα ενημερώνει τον χρήστη επισημαίνοντας πού εντοπίζονται τα συντακτικά λάθη καθώς και ποια είναι αυτά.

**Εναλλακτική Ροή 3:**

**4.3.1.** Το σύστημα διαπιστώνει ότι ο χρήστης δεν έχει προσθέσει end if και end while blocks για κάθε while και if block.

**4.3.2.** Το σύστημα ενημερώνει το χρήστη σε ποια σημεία του κώδικα λείπουν τα end blocks.

**Use case**: Εκτέλεση

**Editor:** Σωτήριος Χασιώτης

**Peer reviewer:** Βασίλειος Μιχανετζής, Κωνσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης

**Βασική Ροή:**

**1.** Ο χρήστης κάνει κλικ στο κουμπί του run.

**2**. Το σύστημα ελέγχει αν έχει γίνει compile μετά από την τελευταία αλλαγή του κώδικα.

**3.** Το σύστημα ξεκινάει την εκτέλεση του κώδικα.

**4.** Το σύστημα εμφανίζει φωτεινό περίγραμμα γύρω από την εντολή προς εκτέλεση.

**5.** Το σύστημα κατά την εκτέλεση του κάθε μπλοκ πριν πάει στο επόμενο μπλοκ θα υποστεί μια μικρή αναμονή .

**6.** Το σύστημα εκτελεί την εντολή του προγράμματος που βρίσκεται εκείνη την στιγμή, γίνεται εκτύπωση των μεταβλητών στην οθόνη εξόδου, ο δείκτης προχωράει κατά ένα και πάμε στο βήμα 4.

**7**. Όταν φτάσει στο end block το σύστημα κρατάει τις τελευταίες τιμές στην οθόνη εξόδου και ενεργοποιεί κουμπί για την εκκαθάριση της.

**8.** Ο χρήστης πατάει το κουμπί εκκαθάρισης.

**9.** Το σύστημα επαναφέρει την οθόνη εξόδου στην αρχική της κατάσταση.

**Εναλλακτική Ροή 1:**

**2.1.1.** Το σύστημα διαπιστώνει ότι δεν έχει γίνει compile.

**2.1.2.**Εμφανίζεται μήνυμα λάθους στον χρήστη.

**Use case**: Μετακίνηση και ένωση block εντολών

**Editor:** Βασίλειος Μιχανετζής

**Peer reviewer:** Κωνσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης, Σωτήριος Χασιώτης

**Βασική Ροή:**

**1.** Ο χρήστης επιλέγει το μπλοκ που επιθυμεί να μετακινήσει μέσω drag.

**2.** Ο χρήστης μετακινεί το block πάνω από την θέση που θέλει να το τοποθετήσει.

**3.** Το σύστημα ελέγχει αν το block που μετακινεί ο χρήστης βρίσκεται σε κοντινή απόσταση από κάποιο άλλο.

**4.** Το σύστημα χρωματίζει την θέση στην οποία αναμένεται να προστεθεί το block.

**5.** Το σύστημα μετακινεί τα block που βρίσκονται δεξιά της θέσης αυτής κατά μία θέση δεξιά.

**6.** Ο χρήστης αφήνει το block στη θέση που επιθυμεί.

**7.** Το σύστημα επισυνάπτει το μετακινούμενο block στο κοντινό του block, δημιουργώντας την επιθυμητή σύνδεση.

**8.** Το σύστημα ελέγχει αν το block που μετακινήθηκε μπορεί να ενταχθεί στην κύρια ροή του κώδικα ή αν ανήκει σε κάποια υπορουτίνα, βάσει της θέσης του και της δομής του προγράμματος.

**9.** Ανάλογα με τον έλεγχο, το σύστημα προσθέτει το id του μετακινούμενου μπλοκ στην κατάλληλη θέση της λίστας.

**Εναλλακτική Ροή 1:**

**2.1.1.** Ο χρήστης προσπαθεί να μετακινήσει το μπλοκ εκτός ορίων οθόνης.

**2.1.2.** Το σύστημα κρατάει το block στα όρια του επιτρεπτού πεδίου.

**Use case**: Διαγραφή block εντολών

**Editor:** Βασίλειος Μιχανετζής

**Peer reviewer:** Κωνσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης, Σωτήριος Χασιώτης

**Βασική Ροή:**

1. O χρήστης πατάει το κουμπί διαγράφης για το block που θέλει να διαγράψει.

2. Το σύστημα εμφανίζει παράθυρο ερώτησης επιβεβαίωσης της διαγραφής.

3. O χρήστης πατάει το κουμπί "Yes".

4. Το σύστημα κλείνει το παράθυρο.

5. Το σύστημα ελέγχει αν το block που διαγράφτηκε αποτελεί μέλος εντολής δυαδικού block και το επισημάνει στον χρήστη με κόκκινο.

6. Το σύστημα αφαιρεί από τη λίστα το id του block διαγραφής .

7. Tο σύστημα διαγραφεί το block.

8. Το σύστημα συνδέει το επόμενο μπλοκ από αυτό που διαγράφτηκε στο προηγούμενο και επαναλαμβάνει μέχρι το τέλος των εντολών.

**Εναλλακτική Ροή 1:**

**3.1.1.** Ο χρήστης πατάει το κουμπί "No".

**3.1.2.** Το σύστημα επιστρέφει το χρήστη στην κύρια οθόνη.

**Use case**: Αποθήκευση

**Editor:** Νυμφοδώρα Μπουζιάνη

**Peer reviewer:** Βασίλειος Μιχανετζής, Μιχαήλ Σοκολάκης, Σωτήριος Χασιώτης, Κωνσταντίνος Καρακίτσος

**Βασική Ροή:**

**1.** Ο χρήστης πατάει το κουμπί Save .

**2.** Το σύστημα εμφανίζει ένα παράθυρο με προεπιλεγμένο όνομα αρχείου και τοποθεσία αποθήκευσης και δύο κουμπιά cancel και save.

**3.** Ο χρήστης εισάγει το όνομα του αρχείου.

**4.** Το σύστημα ελέγχει το όνομα του αρχείου ως προς την εγκυρότητα.

**5.** Εφόσον το όνομα του αρχείου είναι έγκυρο και ο χρήστης επιλέγει τοποθεσία αποθήκευσης.

**6.** Ο χρήστης πατάει το κουμπί Save As .

**7.** Το σύστημα αποθηκεύει το αρχείο στην επιλεγμένη τοποθεσία.

**8.** Το σύστημα εμφανίζει μήνυμα επιτυχούς αποθήκευσης.

**Εναλλακτική Ροή 1:**

**4.1.1.** Το σύστημα διαπιστώνει ότι το όνομα που εισήγαγε ο χρήστης δεν είναι έγκυρο.

**4.1.2.** Το σύστημα εμφανίζει στον χρήστη μήνυμα λάθους.

**4.1.3.** Η περίπτωση χρήσης συνεχίζεται από το βήμα 3 της βασικής ροής.

**Εναλλακτική Ροή 2:**

**2.2.1.** Ο χρήστης πατάει το κουμπί cancel.

**2.2.2.** Το σύστημα κλείνει το παράθυρο χωρίς να γίνεται η αποθήκευση του αρχείου.

**2.2.3.** Το σύστημα επιστρέφει τον χρήστη στην κύρια οθόνη.

**Use case**: Undo

**Editor:** Νυμφοδώρα Μπουζιάνη

**Peer reviewer:** Βασίλειος Μιχανετζής, Μιχαήλ Σοκολάκης, Σωτήριος Χασιώτης, Κωνσταντίνος Καρακίτσος

**Βασική Ροή:**

**1.** Ο χρήστης κάνει αλλαγές στο πρόγραμμά του.

**2.** Το σύστημα αποθηκεύει τις τελευταίες δύο σε μορφή στιγμιότυπου στο ιστορικό-undo.

**3.** Ο χρήστης πατάει το κουμπί Undo.

**4.** Το σύστημα ελέγχει την ύπαρξη ενεργειών στο ιστορικό-undo.

**5.** Το σύστημα διατρέχει το ιστορικό-undo και ανακτά την προηγούμενη κατάσταση του project.

**6.** Το σύστημα διαγράφει την ενέργεια αυτή από το ιστορικό-undo.

**7.** Το σύστημα αποθηκεύει την ενέργεια που θα αναιρεθεί στο ιστορικό-redo.

**8.** Το σύστημα ενεργοποιεί το κουμπί Redo.

**9.** Το σύστημα δημιουργεί την προηγούμενη κατάσταση με βάση τα δεδομένα που πήρε από το ιστορικό-undo.

**10.** Ο χρήστης κάνει νέα αλλαγή στο πρόγραμμα του και το σύστημα καθαρίζει το ιστορικό-redo.

**Εναλλακτική Ροή 1:**

**4.1.1.** Το σύστημα ελέγχει την ύπαρξη ενεργειών στο history-undo και ανακαλύπτει ότι δεν υπάρχει καμία ενέργεια στο ιστορικό.

**4.1.2.**Το κουμπί undo απενεργοποιείται και ο χρήστης παραμένει με την ίδια κατάσταση του προγράμματος του.

**Use case**: Υπορουτίνα

**Editor:** Μιχαήλ Σοκολάκης

**Peer reviewer:** Βασίλειος Μιχανετζής, Σωτήριος Χασιώτης, Κωνσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη

**Βασική Ροή:**

**1.** Ο χρήστης κάνει κλικ στο κουμπί "New Function".

**2.** Το σύστημα εμφανίζει ένα κενό πεδίο στο οποίο ζητά από το χρήστη να εισάγει όνομα .

**3.** Ο χρήστης πληκτρολογεί όνομα για τη νέα function και έπειτα πατά το κουμπί "Continue".

**4.** Το σύστημα εμφανίζει τα block "Start Function" και "End Function" μεταξύ των οποίων ο χρήστης πρέπει να τοποθετήσει μπλοκ εντολών και το ίδιο το μπλοκ κλήσης του function.

**5.** Ο χρήστης τοποθετεί τα block εντολών που επιθυμεί μεταξύ των μπλοκ του βήματος 4 "Start Function" και "End Function".

**6.** Το σύστημα αποθηκεύει τους δείκτες τους σε νέα λίστα από αυτή της 'main' ροής του προγράμματος.

**7.** O χρήστης βάζει το block κλήσης της function στην main.

**8.** Kατά την εκτέλεση του προγράμματος το σύστημα συναντά το block κλήσης, μεταφέρεται στην λίστα του function εκτελεί ό,τι έχει μέσα και επιστρέφει στην main.

**Εναλλακτική Ροή 1:**

**3.1.1.** Ο χρήστης δεν πληκτρολογεί όνομα στο κενό πεδίο και κάνει κλικ στο κουμπί "Continue".

**3.1.2.** Το σύστημα εμφανίζει μήνυμα λάθους.

**3.1.3.** Συνέχεια από το βήμα 2 της βασικής ροής.

**Εναλλακτική Ροή 2:**

**5.2.1.** Ο χρήστης αντί να τοποθετήσει κατάλληλα τα μπλοκ εντολών που επιθυμεί ξανακάνει κλικ το κουμπί "New function".

**5.2.2.** Το σύστημα εμφανίζει κατάλληλο μήνυμα λάθους.

**5.2.3.** Το σύστημα επιστρέφει το χρήστη στην κύρια οθόνη.

**Εναλλακτική Ροή 3:**

**3.3.1**. Ο χρήστης αντί να κάνει κλικ στο κουμπί "continue", κάνει κλικ στο κουμπί "cancel".

**3.3.2.** Το σύστημα επιστρέφει το χρήστη στην κύρια οθόνη.

**Use case**: Debug Mode

**Editor:** Κωνσταντίνος Καρακίτσος

**Peer reviewer:** Βασίλειος Μιχανετζής, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης, Σωτήριος Χασιώτης

**Βασική ροή:**

**1.** Ο χρήστης κάνει κλικ στο κουμπί του debug και επιλέγει τα blocks που επιθυμεί να γίνει break της εκτέλεσης.

**2.** Tο σύστημα αλλάζει το χρώμα των επιλεγμένων blocks.

**3.** Εφόσον έχουν μπει breakpoints, το σύστημα ενεργοποιεί το debug mode.

**4.** Ο χρήστης πατάει το κουμπί run.

**5.** Το σύστημα ελέγχει αν έχει γίνει compile από την τελευταία αλλαγή.

**6.** Το σύστημα ξεκινάει την εκτέλεση σε λειτουργία debug και εκτελεί μέχρι το πρώτο breakpoint.

**7.** Tο σύστημα κάνει ενημέρωση των νέων τιμών στο πεδίο εξόδου.

**8**. Tο σύστημα ελέγχει αν υπάρχει επόμενο breakpoint και περιμένει μέχρι ο χρήστης πατήσει run και επαναληφθούν τα βήματα 4,5,6,7,8 της βασικής ροής.

**9**. Μόλις το πρόγραμμα τερματίσει, το σύστημα απενεργοποιεί το debug mode.

**Εναλλακτική Ροή 1**

**5.1.1.** Το σύστημα διαπιστώνει ότι δεν έχει γίνει compile .

**5.1.2.**Εμφανίζεται μήνυμα λάθους στον χρήστη.

**Εναλλακτική Ροή 2**

**8.2.1**. Ο χρήστης αντί να πατήσει το κουμπί run, πατάει το κουμπί stop.

**8.2.2**. Το σύστημα τερματίζει τη λειτουργία debug.

**8.2.3.**Απενεργοποιεί τα κουμπιά.

**8.2.4.**Επαναφέρει τον δείκτη και καθαρίζει την οθόνη εξόδου.