



(ΕΠΑνΕΚ, κωδικός έργου: Τ1ΕΔΚ-02454)

Π4.1

eVision Unity Project Documentation

Εισαγωγή

Στην παρούσα έκθεση θα παρουσιαστεί η μεθοδολογία και ο τρόπος δημιουργίας της διεπαφής της εφαρμογής eVision, στην πλατφόρμα Unity3D. Πιο συγκεκριμένα θα αναλυθούν τα plugins που χρησιμοποιήθηκαν και ο τρόπος εισαγωγής τους στο project, το σύνολο της ιεραρχίας στην σκηνή (scene) καθώς και τα components, game objects και scripts που χρησιμοποιήθηκαν για το τελικό αποτέλεσμα.

Το σύνολο του project είναι μορφοποιημένο με UI (User Interface) components τα οποία παρέχει η πλατφόρμα του Unity. Κατά κύριο λόγο χρησιμοποιήθηκαν Canvas Components και ένα Camera Component. Για την τελική υλοποίηση της διεπαφής εγκαταστάθηκαν τα 3rd party plugins, Playmaker και UAP (UI Accessibility Plugin).

Το Playmaker είναι ένα plugin οπτικού προγραμματισμού που διευκολύνει την ανάπτυξη στην πλατφόρμα Unity προσφέροντας μια μεγάλη γκάμα εντολών και μεθόδων οι οποίες προσάπτονται στα game objects δίνοντας την δυνατότητα προσαρμογής της λειτουργίας τους. Επιπλέον, διευκολύνει την παραμετροποίηση της λειτουργίας των game objects από μέλη της σχεδιαστικής ομάδας, τα οποία δεν έχουν ανεπτυγμένες γνώσεις προγραμματισμού.

Το UAP διευκολύνει την ανάπτυξη μιας διεπαφής για άτομα με προβλήματα όρασης. Παρέχει αυτοματοποιήσεις για την εισαγωγή τεχνικών πλοήγησης και χειρισμού μιας διεπαφής από άτομα με προβλήματα όρασης.



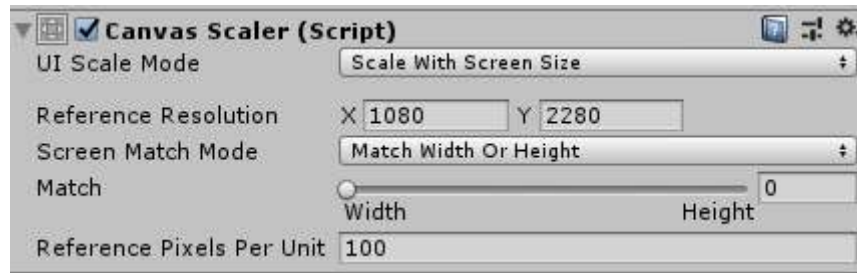
1 Μορφοποίηση σκηνής

Πρώτο βήμα είναι η ακριβής μεταφορά του σχεδιασμού της διεπαφής από τα σχεδιαστικά λογισμικά, στο Unity. Για την υλοποίηση αυτού του βήματος χρησιμοποιήθηκε μια πληθώρα UI components. Πριν αρχίσει η παραμετροποίηση των UI components εισάγαμε στο scene το component που επιβλέπει την λειτουργία του UAP plugin. Για την εισαγωγή του, επιλέγουμε από την καρτέλα “Tools” την επιλογή “UAP Accessibility” και στην συνέχεια “Add Accessibility Manager to Scene”. Η επιλογή αυτή εισάγει στο scene ένα game object με όνομα Accessibility Manager. Για την υλοποίηση της διεπαφής έγινε συνδυαστική χρήση UI components, scripts και των plugin UAP και Playmaker. Στις επόμενες ενότητες θα ακολουθήσει ανάλυση των βασικών στοιχείων που χρησιμοποιήθηκαν καθώς και ο συνδυασμός αυτών.

2 Μεθοδολογία χρήσης UI Components

Σαν βασικό αντικείμενο (parent object) χρησιμοποιήθηκε το Canvas Component, το οποίο είναι ο κορμός οποιουδήποτε σχεδιασμού διεπαφής 2D στην πλατφόρμα Unity. Κάθε μενού και υπο-μενού της διεπαφής αποτελείται από ένα Canvas Component σαν parent object και διάφορα άλλα UI components (Image Component, Text Component, Button Component κτλ) σαν children game objects του Canvas.

Η εφαρμογή eVision έχει σκοπό να χρησιμοποιηθεί από διαφορετικές φορητές συσκευές με διαφορετικές αναλύσεις οθόνης. Αυτό δημιουργεί ένα σχεδιαστικό πρόβλημα το οποίο είναι αναγκαίο να λυθεί. Ο σχεδιασμός της διεπαφής θα πρέπει να είναι ίδιος σε όλες τις διαφορετικές συσκευές. Για την λύση αυτού του προβλήματος έπρεπε να γίνουν σημαντικές παραμετροποιήσεις στο Canvas Component. Το Canvas Component αποτελείται από ένα σύνολο άλλων Component που είναι υπεύθυνα στην διαμόρφωση της λειτουργικότητας του. Για να δημιουργήσουμε μια διεπαφή συμβατή σε διαφορετικές αναλύσεις είναι αναγκαία η ρύθμιση του Canvas Scaler, που χρησιμοποιείται για τον έλεγχο της συνολικής κλίμακας και της πυκνότητας των pixel των στοιχείων UI στον καμβά. Επηρεάζει τα πάντα κάτω από το Canvas Component, συμπεριλαμβανομένων των μεγεθών των γραμματοσειρών και των ορίων εικόνας.



Εικόνα 1: Canvas Scaler Component

Στο Component αυτό αλλάζουμε την επιλογή UI Scale Mode σε “Scale With Screen Size” και επιλέγουμε μια ενδεικτική ανάλυση. Για την ανάπτυξη της εφαρμογής επιλέχθηκε η ενδεικτική ανάλυση 1080x2280 η οποία λειτουργεί ως οδηγός της σχεδίασης, ο λόγος αυτής της επιλογής εξαρτάται από την ανάλυση της συσκευής που γίνεται ο έλεγχος της σχεδίασης της διεπαφής. Στην επιλογή Screen Match Mode, επιλέγουμε “Match Width Or Height”, στο Match επιλέγουμε “0” και στην επιλογή Reference Pixels Per Unit επιλέγουμε “100”.

Το κάθε Canvas Component περιέχει επίσης ένα Initialize Navigations Script Component, Accessible UI Group Root Script Component και ένα Play Maker FSM Script Component. Το Accessible UI Group Root Script Component και το Play Maker FSM Script Component είναι απαραίτητα Components για την χρήση των plugins που αναφέραμε παραπάνω και θα αναλυθούν στην ενότητα 3 και 4 αντίστοιχα. Το Initialize Navigation Script Component δημιουργήθηκε σαν συνδετικός κρίκος των δύο plugins, για να ρυθμίζει την λειτουργία πλοήγησης για κάθε μενού. Περιέχει μια μέθοδο (InitializeNavs) που καλείται από το Playmaker, όταν ενεργοποιείται το εκάστοτε Canvas Component και καθορίζει τις ενέργειες που θα λαμβάνουν χώρα κατά την κάθε χειρονομία που υποστηρίζει το UAP. Με αυτόν το τρόπο μπορεί να καθοριστεί διαφορετική πλοήγηση ανάλογα με το μενού που ενεργοποιείται. Στη μέθοδο αυτή αφού γίνει εκκαθάριση των events που αφορούν τις χειρονομίες πλοήγησης χρησιμοποιώντας ειδικές μεθόδους στην κλάση του UAP_AccessibilityManager, δημιουργούνται καινούρια events για κάθε χειρονομία, και συνδέονται με τα αντίστοιχα events του Play Maker FSM Script Component.

Σαν child objects του Canvas τοποθετήθηκαν τα ανάλογα UI Components για την δημιουργία της διεπαφής. Τα κύρια στοιχεία είναι UI Image και UI Text που εξυπηρετούν την αισθητική πλευρά της διεπαφής, ένα game object που περιέχει ένα UAP Accessible Label Script Component, στοιχείο απαραίτητο για την προσφώνηση του ονόματος του μενού στο οποίο βρίσκεται ο χρήστης, η λειτουργία του οποίου θα αναλυθεί στην ενότητα 3. Επίσης χρησιμοποιούνται UI Button Components, απαραίτητα στοιχεία κάθε μενού της διεπαφής, αφού επιτρέπουν την διάδραση του χρήστη με την διεπαφή και την σύνδεση μεταξύ των μενού και υπο-μενού. Στο component αυτό επίσης εισάγαμε και δύο component απαραίτητα για την χρήση των UAP και Playmaker plugins, το Accessible Button Script Component και το

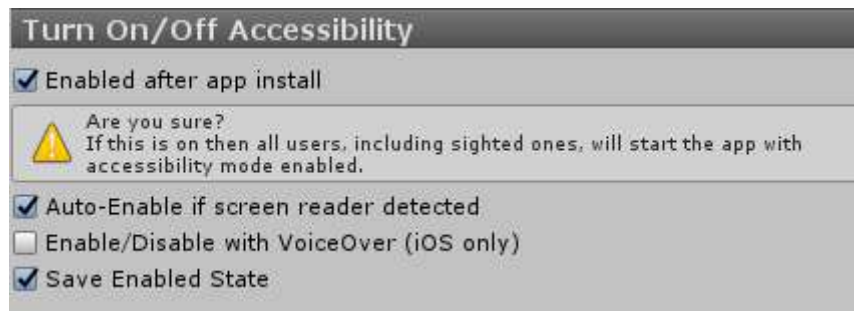
PlayMaker FSM Script Component, η χρήση των οποίων θα αναλυθεί στις ενότητες 3 και 4 αντίστοιχα. Τέλος, χρησιμοποιήθηκαν επίσης UI Toggle Components και UI Slider Components για την παραμετροποίηση των ρυθμίσεων των δυνατοτήτων της διεπαφής.

3 Μεθοδολογία χρήσης UAP plugin

Επιλέγοντας το game object “Accessibility Manager” μπορούμε να ρυθμίσουμε την λειτουργικότητα του UAP plugin. Πλήρης καταγραφή των δυνατοτήτων και της λειτουργίας του UAP plugin υπάρχει διαθέσιμη στην ιστοσελίδα του plugin: <http://www.metalpopgames.com/assetstore/accessibility/doc/index.html>. Στην καρτέλα Inspector βλέπουμε τις ρυθμίσεις που μπορούν να γίνουν κατηγοριοποιημένες σε 5 ενότητες:

1. Turn On/Off Accessibility

Στην ενότητα επιλέγουμε να ενεργοποιείται το UAP plugin κατά την εγκατάσταση της εφαρμογής.



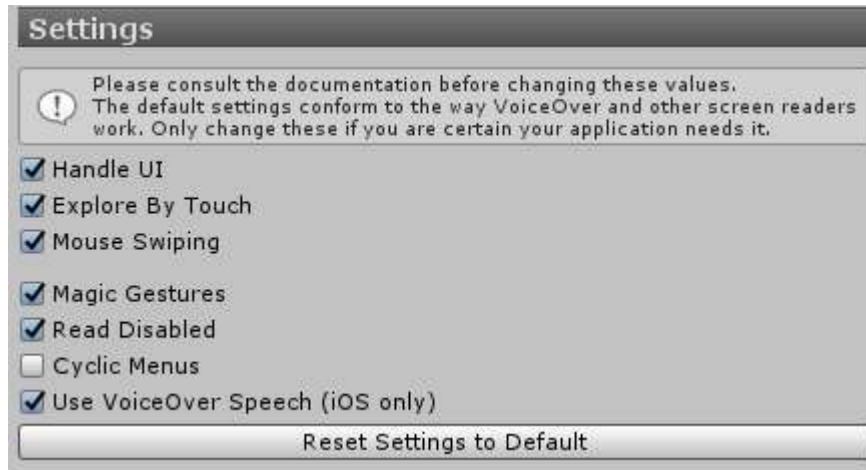
Εικόνα 2: Accessibility Manager Component

Ενεργοποιούμε τις επιλογές “Enabled after app install”, “Auto-Enable if screen reader is detected” και “Save Enabled State”. Επίσης αν το η εφαρμογή μας προορίζεται για iOS ενεργοποιούμε επίσης την επιλογή “Enable/Disable with VoiceOver (iOS only)”

2. Testing and Debugging

Στην ενότητα αυτή επιλέγουμε “On” κατά την διάρκεια της ανάπτυξης της εφαρμογής.

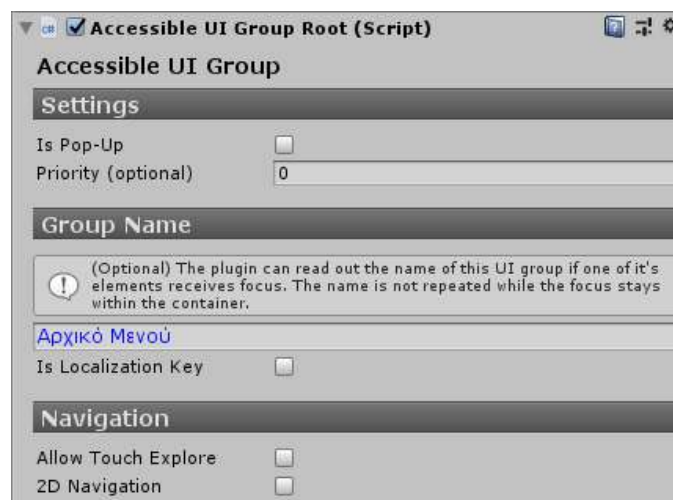
3. Στην ενότητα Settings ενεργοποιούμε όλες τις επιλογές εκτός αυτής του “Cyclic Menu” για να διατηρήσουμε την γραμμικότητα της σχεδίασης των μενού



4. Στην ενότητα Sounds έχουμε την δυνατότητα να εισάγουμε αρχεία ήχου για διάφορες ενέργειες πλοήγησης. Το UAP plugin περιέχει ήδη εγκατεστημένα αρχεία ήχου που κρίθηκαν επαρκή για την δημιουργία της διεπαφής.
5. Η ενότητα NGUI αναφέρεται σε παλαιότερες εκδόσεις του Unity για αυτό δεν ενεργοποιείται.

Αφού ρυθμιστεί ο Accessibility Manager του UAP plugin, είναι αναγκαίο να εισαχθούν ειδικά διαμορφωμένα components στο Canvas Component αλλά και σε κάθε άλλο UI component που θα έχει προσφέρει κάποιου είδους διάδρασης στον χρήστη.

Το κάθε Canvas Component λειτουργεί όπως είπαμε σαν μενού της διεπαφής. Το UAP προσφέρει την δυνατότητα ομαδοποίησης επιλογών για την καλύτερη πλοήγηση. Για την ενεργοποίηση αυτής της δυνατότητας είναι αναγκαίο να προσθέσουμε το Accessibility UI Group Root Script Component στο εκάστοτε Canvas Component.

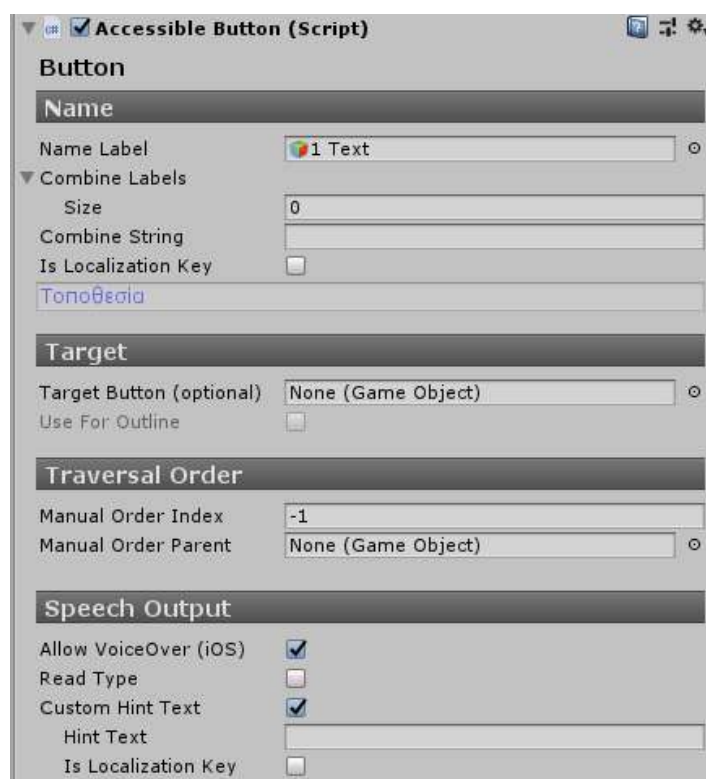


Εικόνα 3: Accessible UI Group Root Script Component

Οι ρυθμίσεις του component αυτού χωρίζονται σε 3 ενότητες. Στην ενότητα Settings επιλέγουμε αν αυτό το Canvas game object δρα ως κάποιου είδους pop-up στην διεπαφή και αν χρειάζεται να δοθεί προτεραιότητα στο συγκεκριμένο μενού. Στην ενότητα Group Name πληκτρολογούμε το όνομα του μενού που θέλουμε να προφέρει κατά την ενεργοποίηση το UAP. Η ενότητα Navigation διαθέτει σημαντικές ρυθμίσεις όπου πρέπει να μείνουν απενεργοποιημένες. Η επιλογή “Allow Touch Explore” απομονώνει την διάδραση του χρήστη μόνο στις χειρονομίες που εισάγουμε προστατεύοντας τον χρήστη από την λάθος επιλογή στο εκάστοτε μενού. Η επιλογή “2D Navigation” αναφέρεται στην περίπτωση που το μενού ακολουθεί σχεδίαση πλέγματος.

Σε κάθε Canvas game object όπως αναφέρθηκε παραπάνω υπάρχουν εμφωλευμένα διάφορα UI components ανάλογα με τον σχεδιασμό του εκάστοτε μενού. Για κάθε, διαφορετικό τύπο UI Component το UAP plugin διαθέτει το αντίστοιχο UAP component το οποίο διευκολύνει την διάδραση του με άτομα με προβλήματα όρασης. Για αυτό το λόγο κατά την σχεδίαση της διεπαφής, σε κάθε εισαγωγή UI Components πρέπει να εισάγεται και το αντίστοιχο UAP component.

Στα UI Component τύπου Button είναι απαραίτητο να εισάγουμε το UAP Component Accessible Button Script.



Εικόνα 4: Accessible Button Script Component

Στην ενότητα ρυθμίσεων Name τοποθετούμε ένα Text Game Object ή πληκτρολογούμε το όνομα που θέλουμε να προφέρει το UAP όταν ο χρήστης πλοηγηθεί στο Button component αυτό. Στην ενότητα Traversal Order δίνεται η δυνατότητα αλλαγής της σειράς που προφέρονται τα Components μέσα σε ένα μενού, αφήνοντάς τα απενεργοποιημένα η σειρά καθορίζεται από την θέση τους στην σκηνή. Στην ενότητα Speech Output, ενεργοποιούμε την επιλογή “Allow VoiceOver (iOS)” για χρήση της εφαρμογής σε λογισμικά iOS, επίσης μπορούμε να ενεργοποιήσουμε την επιλογή “Read Type” εφόσον θέλουμε το UAP να προφέρει και τον τύπο του Component, στην συγκεκριμένη περίπτωση ο τύπος είναι “Button”. Επίσης υπάρχει η δυνατότητα εισαγωγής σχολίου ενεργοποιώντας την επιλογή “Custom Hint Text” και πληκτρολογώντας την λέξη ή φράση την οποία θα προφέρει το UAP.

Στην περίπτωση χρήσης του UI Toggle Component εισάγουμε και το αντίστοιχο UAP Accessible Toggle Script Component.



Εικόνα 5: Accessible Toggle Script Component

Η ενότητα ρυθμίσεων Name είναι ίδια με αυτή του Accessible Button Script Component που περιγράφηκε παραπάνω, το ίδιο και η ενότητα Traversal Order. Η ενότητα Target αφορά την περίπτωση που το UAP component τοποθετηθεί σε διαφορετικό game object από αυτό που περιέχει το UI Toggle component. Η ενότητα Speech Output περιέχει τις ίδιες επιλογές με το Accessible Button Script Component καθώς και τρεις επιπλέον επιλογές, “Custom, Status Text, “Custom Checked Text” και “Custom Unchecked Text” που αφορούν την προσφώνηση προσωποποιημένων ηχητικών μηνυμάτων κατά την χρήση του UI Toggle component.

Εκτός των UAP component που εισάγαμε στα αναγκαία Canvas game objects και στα child objects τους, ήταν απαραίτητο να γίνουν κάποιες μορφοποιήσεις στον πηγαίο κώδικα του UAP plugin έτσι ώστε να εξυπηρετηθεί η λειτουργία της διεπαφής.

Δημιουργήθηκαν μέθοδοι στην κλάση UAP_AccesiblityManager οι οποίοι αποσκοπούσαν στην εκκαθάριση των events που ήταν συνδεδεμένα με τις χειρονομίες που αναγνωρίζει το UAP plugin και οι αντίστοιχοι μέθοδοι που συνδέουν τα events αυτά με τις χειρονομίες.

Πιο συγκεκριμένα δημιουργήθηκαν οι εξής μέθοδοι για εκκαθάριση:

- ResetTwoFingerSwipeLeftHandler()
- ResetTwoFingerSwipeRightHandler()
- ResetOneFingerSwipeLeftHandler()
- ResetOneFingerSwipeRightHandler()

Και οι παρακάτω μέθοδοι για σύνδεση των events, οι οποίοι σαν input παίρνουν το delegate "OnTapEvent":

- SetOneFingerSwipeRightHandler(OnTapEvent func)
- SetOneFingerSwipeLeftHandler(OnTapEvent func)
- SetTwoFingerSwipeRightHandler(OnTapEvent func)
- SetTwoFingerSwipeLeftHandler(OnTapEvent func)

Οι παραπάνω μέθοδοι καλούνται από το script Initialize Navigations Script Component που περιγράφηκε στην ενότητα δύο κάθε φορά που ενεργοποιείται ένα μενού και δίνουν την δυνατότητα αλλαγής των ενεργειών της κάθε χειρονομίας.

Μια ακόμη μορφοποίηση που έγινε στον κώδικα του UAP plugin είναι αυτή της λειτουργίας Long Press, η οποία δίνει την δυνατότητα στον χρήστη να ενημερωθεί σε ποιο μενού της διεπαφής βρίσκεται, διατηρώντας το δάχτυλο του στην οθόνη για τρία δευτερόλεπτα. Η μέθοδος CheckForLongPress() καλείται στην Update() loop της κλάσης UAP_AccessibilityManager και είναι πάντα ενεργοποιημένη.

Επιπλέον στο base class UAP_BaseElement από το οποίο κληρονομούν κοινές λειτουργίες τα UAP components, τοποθετήθηκαν μέθοδοι για την σύνδεση των UAP component με τα αντίστοιχα Playmaker component σε συγκεκριμένα σημεία αλληλεπίδρασης, όπως είναι η επιλογή ενός button, η λειτουργία highlighted και η λειτουργία Idle. Οι μέθοδοι αυτοί είναι οι εξής:

-
- CallPlayMakerHighlighted()

Καλείτε από την κλάση AccessibleButton στην μέθοδο HoverHighlight() και από την κλάση AccessibleToggle στην μέθοδο HoverHighlight()

- CallPlayMakerPressed()

Καλείτε από την κλάση AccessibleButton στην μέθοδο Interact() και από την κλάση AccessibleToggle στην μέθοδο Interact()

- CallPlayMakerUnSelected()

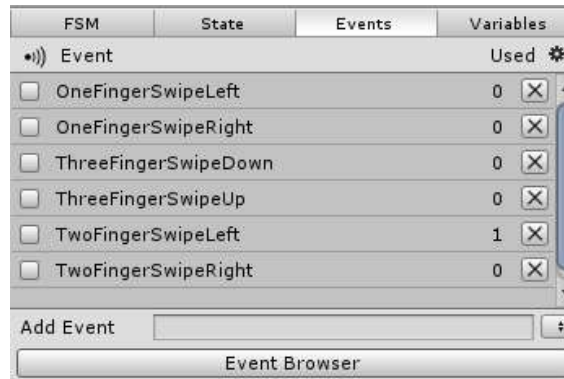
Καλείτε από την κλάση AccessibleButton στην μέθοδο HoverHighlight(), από την κλάση AccessibleToggle στην μέθοδο HoverHighlight() καθώς και από την κλάση UAP_BaseElement από την μέθοδο OnDisable().

Οι μορφοποιήσεις του πηγαίου κώδικα μπορούν να εντοπιστούν κάνοντας εύρεση της λέξης “Tetragon”, αφού έχει τοποθετηθεί ειδικό σχόλιο για την διευκόλυνση της εύρεσής τους.

4 Μεθοδολογία χρήσης Playmaker plugin

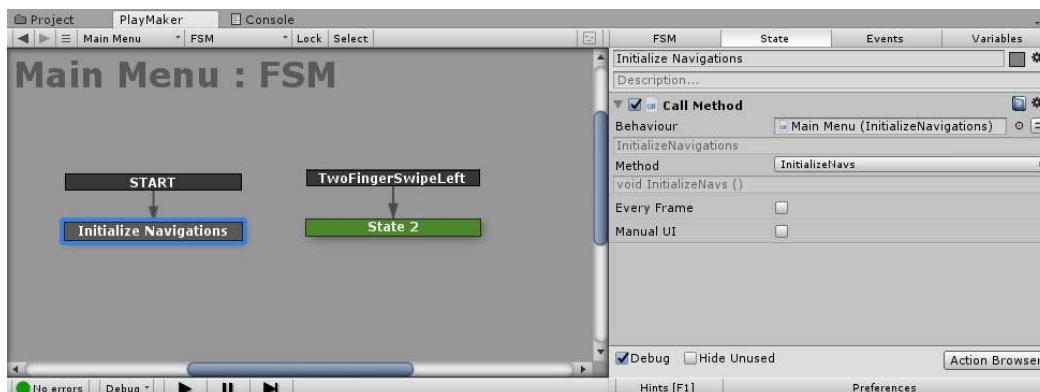
Το Playmaker plugin παρέχει μια βιβλιοθήκη Actions τα οποία μπορούν να προστεθούν σε οποιοδήποτε game object και να ευρίνουν την λειτουργικότητά τους. Επίσης, το Playmaker plugin παρέχει και αυτό την δυνατότητα δημιουργίας events τα οποία καλούνται οποιαδήποτε στιγμή της χρήσης της διεπαφής. Για αυτό το λόγο τοποθετήθηκαν Play Maker FSM Script components σε κάθε Canvas game object καθώς και σε άλλα game object που ήταν υπεύθυνα για την αλληλεπίδραση της διεπαφής με τον χρήστη.

Πιο συγκεκριμένα, σε κάθε Canvas Play Maker FSM component δημιουργήθηκαν έξι events που αφορούσαν τις χειρονομίες πλοήγησης.

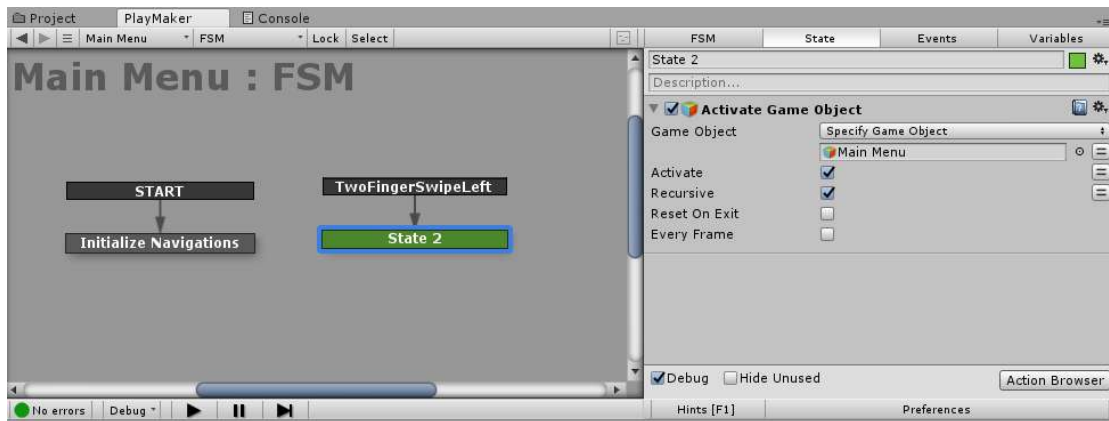


Εικόνα 6: Τα events του κάθε Canvas FSM component στην διεπαφή του PlayMaker Plugin

Αφού δημιουργήθηκαν τα events στο περιβάλλον του PlayMaker plugin, συνδέθηκαν με τα κατάλληλα Actions. Κάθε PlayMaker FSM Component έχει επίσης μια υποδοχή Actions που ονομάζεται "Start" και ενεργοποιείται κάθε φορά που ενεργοποιείται και το αντίστοιχο game object στο οποίο είναι προσαρτημένο. Στο "Start" του κάθε Canvas game object τοποθετήθηκε το Action "Call Method" το οποίο καλεί την μέθοδο "InitializeNavs" της κλάσης InitializeNavigations που περιγράφηκε στην ενότητα δύο. Η μέθοδος αυτή με τη σειρά της αντιστοιχεί τα events των χειρονομιών πλοήγησης που περιγράφονται στην ενότητα τρία με τα events του PlayMaker FSM Component σε κάθε Canvas Game Object. Η μεθοδολογία αυτή επιτρέπει τους σχεδιαστές να τοποθετούν τα Actions που χρειάζεται στο περιβάλλον του Play Maker χωρίς χρειάζονται γνώσεις προγραμματισμού.

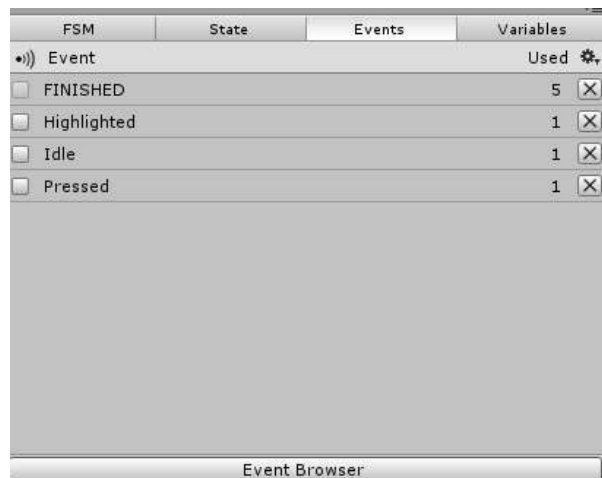


Εικόνα 7: Περιβάλλον Play Maker, τοποθέτηση Action στην υποδοχή Start

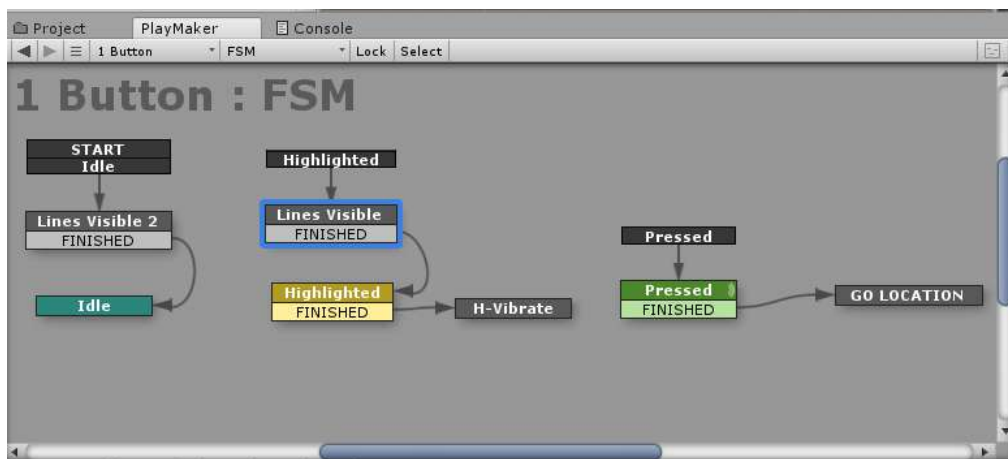


Εικόνα 8: Περιβάλλον Play Maker, τοποθέτηση Action στο event χειρονομίας πλοήγησης “TwoFingerSwipeLeft”

Στα UI Components Button και Toggle τοποθετήθηκαν PlayMaker FSM Components για την διαμόρφωση της λειτουργικότητάς τους κατά την πλοήγηση της διεπαφής. Δημιουργήθηκαν τα κατάλληλα events και έγινε η αντιστοίχιση με τα Actions όπου χρειαζόταν.

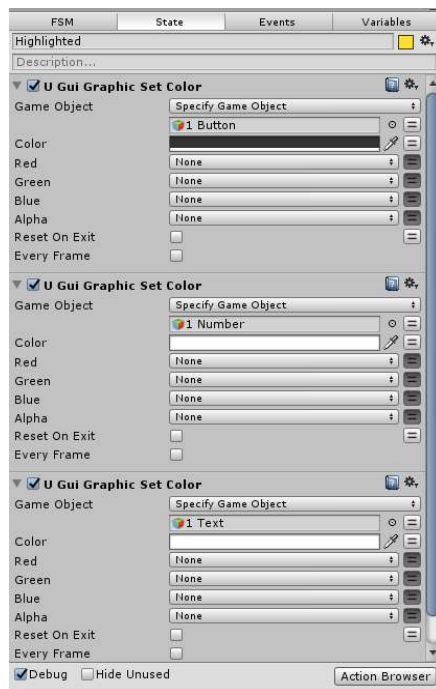


Εικόνα 9: Δημιουργία PlayMaker events



Εικόνα 10: Διαμόρφωση λειτουργικότητας UI Button Component στο περιβάλλον PlayMaker

Κατά την διαμόρφωση της λειτουργικότητας των UI Button Components χρησιμοποιήθηκαν PlayMaker Actions που αφορούσαν την αισθητική πλευρά της διεπαφής καθώς και τις ενέργειες που λαμβάνουν χώρα κατά την πλοήγηση της διεπαφής. Ένα παράδειγμα φαίνεται στην παρακάτω εικόνα όπου καλείται το event “Highlighted” όταν ο χρήστης πλοηγείται σε εκείνο το σημείο. Παρατηρούμε ότι οι ενέργειες που γίνονται είναι να αλλάξουν χρώμα γραμματοσειρές και φόντο για να δοθεί έμφαση στην επιλογή αυτή και να γίνει περισσότερο αντιληπτή από τα άτομα με προβλήματα όρασης.



Εικόνα 11: PlayMaker Actions κατά την λειτουργία Highlighted ενός UI Button

Χρησιμοποιώντας τα events και τις υποδοχές για Actions του PlayMaker διευκολύνεται η περαιτέρω ανάπτυξη της εφαρμογής eVision. Με αυτόν τον τρόπο επίσης γίνεται ευκολότερη και η διασύνδεση της διεπαφής με τους αισθητήρες του συστήματος καθώς έχουν δημιουργηθεί συγκεκριμένες υποδοχές, στις οποίες η προγραμματιστική ομάδα θα τοποθετήσει τις κατάλληλες κλίσεις στους αλγορίθμους της επεξεργασίας των δεδομένων που απορρέουν από τους αισθητήρες.