Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Παιχνίδι Παραγωγής Τέλειων Λαβυρίνθων σε Unity3D** **Perfect Maze Generator using Unity3D**
Ονοματεπώνυμο Φοιτητή	**Νικόλαος Στούπας**
Πατρώνυμο	**Γεώργιος**
Αριθμός Μητρώου	**ΜΠΠΛ/ 17053**
Επιβλέπων	**Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής**

Ημερομηνία Παράδοσης **Σεπτέμβριος 2020**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Ευθύμιος Αλέπης Αναπληρωτής καθηγητής	Μαρία Βίρβου Καθηγήτρια	Κων/νος Πατσάκης Επίκουρος Καθηγητής

Πίνακας Περιεχομένων

1. Περίληψη

Δημιουργήθηκε παιχνίδι παραγωγής τέλειων λαβυρίνθων χρησιμοποιώντας αλγοριθμική προσέγγιση. Αναδρομικά οι λαβύρινθοι εμφανίζονται για πρώτη φορά περίπου στο 430 με 440 π.Χ. στα διηγήματα του Ηροδότου τόσο στην Αίγυπτο όσο και στην αρχαία Κνωσσό. Ως τέλειος λαβύρινθος ορίζεται οποιοσδήποτε λαβύρινθος που δεν έχει μη προσβάσιμες περιοχές, δεν έχει κυκλικά μονοπάτια και ανοιχτές περιοχές και όλα του τα σημεία / tiles στο πάτωμά του είναι προσβάσιμα με μόνο ένα μονοπάτι να συνδέει οποιοδήποτε ζευγάρι αυτών. Η υλοποίηση του αλγορίθμου έγινε σε γλώσσα C# σε περιβάλλον Visual Studio 2019. Η σύνδεση ανάμεσα στα γραφικά, τον κώδικα και όλο το υπόλοιπο παιχνίδι ήταν εφικτή μέσω της Unity, μιας μηχανής παραγωγής παιχνιδιών για διάφορες πλατφόρμες. Ακολουθήθηκε απλή τακτική συγγραφής σεναρίων και εισαγωγή τόσο χαρακτήρα με κίνηση μέσα στον λαβύρινθο όσο και UI elements με σκοπό τη διαδραστικότητα με τον χρήστη/παίκτη. Πρόκειται ουσιαστικά για ένα εισαγωγικό επίπεδο με πολύ μεγάλο περιθώριο για επεκτάσεις και προσθήκες μελλοντικά. Η εξαγωγή του τελικού προϊόντος το οποίο για την παρούσα μεταπτυχιακή διατριβή ορίζεται ως demo ενός μελλοντικού ολοκληρωμένου παιχνιδιού έγινε, για λόγους ευκολίας κατά την παρουσίαση, αποκλειστικά για υπολογιστές. Σημειώνονται τέλος και όλα τα στοιχεία που i) αφαιρέθηκαν από το παραγωγικό προϊόν και ii) πρόκειται να προστεθούν σε αυτό σε επερχόμενες εκδόσεις.

1. Abstract

A game capable of generating perfect mazes was created using an algorithmic approach. In retrospect, labyrinths first appear in about 430 to 440 BC. in the stories of Herodotus both in Egypt and in ancient Knossos. A perfect maze is defined as any maze that has no inaccessible areas, no circular paths and open areas, and all its points / tiles on the floor are accessible with only one path connecting any pair of them. The algorithm was implemented in C # language using Visual Studio 2019. The connection between graphics, code and all the rest of the game was made possible through Unity, a cross-platform game engine. A simple scripting tactic was followed while introducing both character navigation through the maze and UI elements for player interaction. This is essentially an introductory level with a great deal of room for future extensions and additions. The export of the final product, which for this postgraduate thesis is defined as a demo of a future integrated game, was made exclusively for computers for ease of presentation. Finally, all the elements that are either i) removed from the product or ii) will be added to it in forthcoming editions are included in the corresponding chapter.

2. Εισαγωγή

Αρχικά θα γίνει προσπάθεια ανάλυσης και διαφοροποίησης των εννοιών maze και labyrinth. Παρόλο που και οι δύο ορισμοί περιγράφουν ένα σύνθετο σύνολο από μονοπάτια που προκαλεί σύγχυση[R1] , οι 2 έννοιες διαφέρουν τόσο σημασιολογικά όσο και ιστορικά. Ως maze ορίζεται ένα περίπλοκο, διακλαδωτό παζλ που περιλαμβάνει επιλογή μονοπατιού και κατεύθυνσης ενώ το labyrinth έχει μόνο ένα μοναδικό μονοπάτι που οδηγεί στο κέντρο. Τόσο επειδή με την πάροδο των χρόνων οι 2 ορισμοί απέκτησαν σιγά σιγά ταυτόσημη σημασία όσο και λόγω του ότι στην ελληνική γλώσσα δεν υπάρχει ακριβής μετάφραση για τον όρο «maze» , αναφέρεται ως λαβύρινθος και επισημαίνεται ότι στο παρόν έγγραφο κάθε φορά που αναφέρεται ο ορισμός «λαβύρινθος» , εννοείται «maze» και όχι «labyrinth». Περεταίρω διαφορές ανάμεσα στα 2 υπάρχουν και είναι οι εξής :

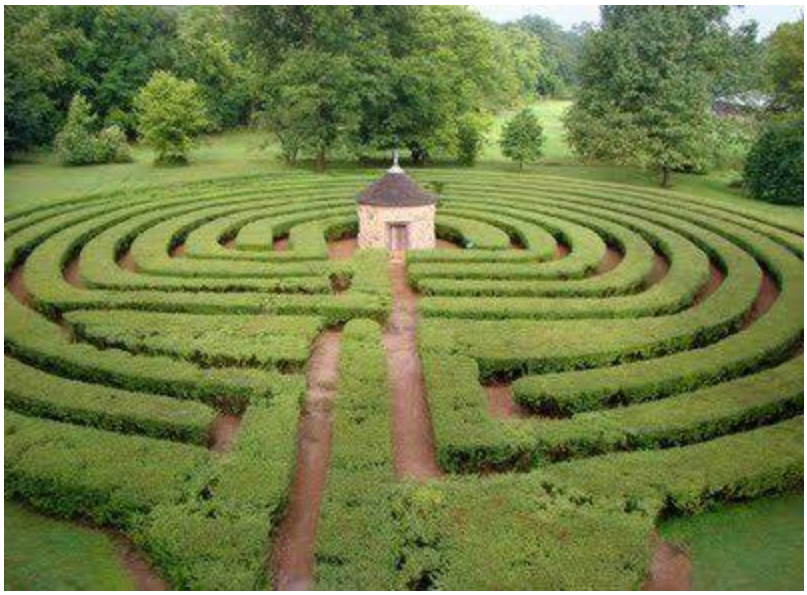
1ον Ένα labyrinth έχει μια μόνο είσοδο και γυρίζει / συστρέφεται χωρίς διακλαδώσεις ενώ ένα maze ουσιαστικά είναι ένα μονοπάτι με πολλές διακλαδώσεις, επιλογές διαδρομής και αδιέξοδα.

2ον Ένα labyrinth δεν είναι σχεδιασμένο για να δυσκολεύει τον πλοηγό. Μπορεί να είναι μακρύ αλλά πάντα υπάρχει μόνο ένα μονοπάτι. Ένα maze αποτελεί ένα περιοδικό παζλ και μπορεί να είναι σχεδιασμένο διαθέτοντας πολλά επίπεδα κλιμακούμενης δυσκολίας και πολυπλοκότητας.

3ον Ένα labyrinth έχει μόνο μια είσοδο που είναι ταυτόχρονα και έξοδος. Υπάρχει επίσης μόνο μια διαδρομή από την είσοδο στο κέντρο. Ένα maze μπορεί να έχει διαφορετική είσοδο και έξοδο.

4ον Ένα labyrinth μπορεί να διαθέτει πνευματική έννοια. Για παράδειγμα αντιπροσωπεύει το σύνθετο και μακρύ μονοπάτι για την συνάντηση με το Θεό. Ένα maze χρησιμοποιείται συνήθως σε επιστημονικά πειράματα για να μελετηθούν ορισμοί όπως χωρική αντίληψη και κάποιες φορές νοημοσύνη.

Ιστορικά η πρώτη αναδρομή παρατηρείται την περίοδο που ο Johann Georg Rapp και οι ακόλουθοί του εγκατέστησαν έναν οικισμό στον ποταμό Wabash στην Ιντιάνα, το 1814.[R2] Μια από τις πρώτες πράξεις τους ήταν να δημιουργήσουν έναν λαβύρινθο από φυτά. Η ιδέα ότι μια πρωτοπόρος κοινότητα θα δαπανούσε πολύτιμη ενέργεια για μια τέτοια δημιουργία φάνταζε απίθανη. Το συγκρότημα των εποίκων του Rapp είχε τις πιεστικές ανάγκες της ερημιάς για να δαμάσει και να φτιάξει τροφή ενώ ταυτόχρονα έπεφταν νεκροί από επιδημία ελονοσίας. Ο λόγος για την επίμονη αφοσίωσή τους στην διακοσμητική κηπουρική έγκειται στις πνευματικές τους πεποιθήσεις αλλά και στη θεμελιώδη (αν και συχνά θολή) διάκριση μεταξύ των ορισμών maze και labyrinth.



Εικόνα 2.1 : Ο λαβύρινθος New Harmony στην Ιντιάνα

Υπάρχουν πολλές διαφορετικές ερμηνείες της σημασίας των λαβυρίνθων σε αρχαίους πολιτισμούς. Οτιδήποτε υποχρέωσε τους προγόνους μας να γράψουν πέτρες με μοτίβα λαβυρίνθου ή να βάλουν ομόκεντρους δακτυλίους πετρωμάτων στην έρημο, οι ενδείξεις τέτοιων αντικειμένων, που χρονολογούνται από την Εποχή του Χαλκού, έχουν βρεθεί σε Ινδία, Ισπανία, Βόρεια Αμερική και Βόρεια Αφρική. Τα αρχαία πετρογλυφικά των λαών Tohono O'odham και Pima, που κατείχαν τη σύγχρονη Αριζόνα και το βόρειο Μεξικό, απεικονίζουν την εικόνα ενός ανθρώπου στην είσοδο ενός λαβυρίνθου. Περισσότεροι από 500 αρχαίοι λαβύρινθοι έχουν εντοπιστεί στη Σκανδιναβία.[R3]

Η περιγραφή του αιγυπτιακού λαβυρίνθου από τον 5ο αιώνα π.Χ. του Ηρόδοτου είναι η πρώτη καταγραφή ενός φυσικού λαβυρίνθου. Θεωρούσε ότι ήταν ένα κατασκεύασμα πιο εντυπωσιακό και από τις Πυραμίδες. Στη βόρεια Ευρώπη, και ιδιαίτερα στα βρετανικά νησιά, υπήρχε μια μακρά παράδοση δημιουργίας λαβυρίνθων, κόβοντας μια πέτρινη διαδρομή σε χλοοτάπητα. Είναι πολύ δύσκολο να προσδιοριστούν τα χαρακτηριστικά τους και απαιτείται τακτική κοπή και διαμόρφωση για να μην εξαφανιστούν. Το Breamore Mizmaze στο Hampshire πιθανολογείται να έχει δημιουργηθεί σύντομα μετά την Νορμανδική κατάκτηση. Στο Wing, Rutland, ένας πολύ καλά διατηρημένος λαβύρινθος χλοοτάπητα χρονολογείται τουλάχιστον από τον 17ο αιώνα. Εξοικειωμένοι με την έννοια λαβύρινθος φαίνονται να ήταν και λόγω Σαίξπηρ: στο Όνειρο της Θερινής Νύχτας (1590-1597) ο Τιτάνια αναφέρει τους λαβύρινθους χλοοτάπητα ως "γραφικούς λαβύρινθους σε βαθύ πράσινο".

Εικόνα 2.2 : Ο λαβύρινθος του Kent Castle στο Λιντς

Το μοτίβο λαβυρίνθου υιοθετήθηκε από τους πρώτους χριστιανούς και έφτασε στο ζενίθ με το πλακόστρωτο παράδειγμα στο πάτωμα του καθεδρικού ναού Chartres της Γαλλίας. Χτισμένο τον 13ο αιώνα, θεωρείται ότι αναφέρεται στην ευσεβής πράξη του προσκυνήματος. Ένας άλλος πλακόστρωτος λαβύρινθος κατασκευάστηκε στον καθεδρικό ναό της Amiens περίπου την ίδια εποχή.

Παραδοσιακά, οι λαβύρινθοι χαρακτηρίζονται από ένα μοναδικό μονοπάτι - μια ενιαία διαδρομή με πολλαπλές στροφές, αλλά χωρίς αδιέξοδο ή ψευδή ανοίγματα - ενώ οι maze λαβύρινθοι βασίζονται σε πολυτροχικές διαδρομές και πολλαπλά τυφλά δρομάκια. Δυστυχώς όπως αναφέρθηκε και στην αρχή της μεθόδου που λαμβάνει χώρα σε αυτήν την μεταπτυχιακή διατριβή για αποσαφήνιση των δύο εννοιών, οι όροι λαβύρινθος (labyrinth) και λαβύρινθος (maze) είναι εναλλάξιμοι εδώ και αιώνες. Όταν ο Πλάτωνας έγραψε για τον μυθολογικό

λαβύρινθο της Κνωσού, το σπίτι του Μινώταυρου, θα μπορούσε να αναφερόταν σε οποιοδήποτε από τα 2. "Είναι μια διαδρομή που προκαλεί σύγχυση, δύσκολο να ακολουθηθεί χωρίς νήμα, αλλά με την προϋπόθεση ότι δεν καταβροχθίζεστε στο μέσον, οδηγεί σίγουρα, παρά τις ανατροπές και τις στροφές, πίσω στην αρχή".[R4]

Γίνεται επίσης αντιληπτό ότι αν ένας λαβύρινθος τύπου labyrinth αντιπροσωπεύει ένα πνευματικό και μυστικιστικό υψηλό σημείο στην αρχιτεκτονική και την κηπουρική, ο λαβύρινθος τύπου maze αποσκοπεί στη διασκέδαση. Οι πρώτοι φυσικοί λαβύρινθοι αναπτύχθηκαν μέσω των κομβικών κήπων που ήταν μοντέρνοι στα τέλη του 16ου και στις αρχές του 17ου αιώνα. Τα περίπλοκα σχέδια των φραχτών που τα σχημάτισαν μεταφέρθηκαν στο «φυτεμένο παζλ» του εκάστοτε λαβυρίνθου.

Εικόνα 2.3 : Ο φυσικός λαβύρινθος Longleat Hedge Maze στο Γουίλτσαϊρ.

Ο παλαιότερος και πιο διάσημος φυσικός λαβύρινθος του Ηνωμένου Βασιλείου είναι εκείνος στο Hampton Court Palace στις όχθες του ποταμού Τάμεση. Ιδέα του William III μεταξύ 1690 και 1700 και σχεδιασμένο από μία από τους κορυφαίους συνεταιρισμούς αρχιτεκτονικής τοπίου της εποχής, ανάμεσα στον George London και τον Henry Wise, ήταν αρχικά φυτεμένος με γκαζόν. Είχε τραπεζοειδές σχήμα και μεταγενέστερα αναδιαμορφώθηκε αποτελούμενο από Ίταμο. Το συγκεκριμένο κωνοφόρο αποτελούσε μια αγαπημένη επιλογή για λαβύρινθους χάρη στο πυκνό, αειθαλές φύλλωμά του. Οι διαδρομές του σήμερα εκτείνονται σε περισσότερο από μισό μίλι.

Ο αρχιτέκτονας Girolamo Frigimelica σχεδίασε το φυσικό λαβύρινθο στη Villa Pisani στο Stra, κοντά στη Βενετία, το 1722, στο πλαίσιο της δουλειάς του για τη βίλα των 114 δωματίων του Δόγη Alvise Pisani. Ο λαβύρινθος, γνωστός ως «Il Labirinto», βασίζεται σε 12 ομόκεντρους δακτυλίους φραχτών με έναν πέτρινο πύργο στο κέντρο. Αυτό είναι επίσης σχεδιασμένο για να συγχέεται, με μια διπλή ελικοειδή εξωτερική σκάλα. Ο Ναπολέων Βοναπάρτης κατέλαβε τη βίλα για λίγο και, σύμφωνα με μαρτυρίες, κατάφερε να χαθεί στο Il Labirinto.[R5]

Το Longleat Hedge Maze, που δημιουργήθηκε στη δεκαετία του 1970 στο αρχοντικό με το ίδιο όνομα, λέγεται ότι είναι ένα από τα μακρύτερα στον κόσμο, με μονοπάτια που εκτείνονται σε σχεδόν 1,7 μίλια και φράκτες αποτελούνται από 16,000 φυτά Ιτάμου. Το Maze Pineapple Garden στο Wahiawa της Χαβάη, είναι ο μεγαλύτερος λαβύρινθος στον κόσμο σε τρία στρέμματα και αποτελείται από αυτόχθονα φυτά όπως Heliconia, Croton και, αναρίθμητα φυτά ανανά. Στο Baan Teelanka στο Πουκέτ της Ταϊλάνδης, ένας φυσικός λαβύρινθος ύψους 1.000 τετραγωνικών μέτρων έχει προκαλέσει σύγχυση με τη μορφή ενός ανάποδου σπιτιού ως κεντρικού χαρακτηριστικού του.

Από τα μέσα του 18ου αιώνα οι λαβύρινθοι παρήλθαν από τη μόδα στην Αγγλία με την έλευση του κινήματος Τοπίου, και πολλοί εκριζώθηκαν. Ωστόσο ο άνθρωπος ακόμα γοητεύεται από την έννοια του να βρίσκεται κάπου νιώθοντας ότι έχει χαθεί. Έτσι λοιπόν οι λαβύρινθοι συνεχίζουν να δημιουργούνται, με μια ραγδαία εμφάνιση νέων που δημιουργούνται τα τελευταία 50 χρόνια.

Στην ελληνική μυθολογία ο λαβύρινθος ήταν μια περίπλοκη, συγκεχυμένη δομή σχεδιασμένη και κατασκευασμένη από τον θρυλικό τεχνίτη Δαίδαλο για τον βασιλιά Μίνωα της Κρήτης στην Κνωσό. Η λειτουργία του ήταν να κρατήσει μέσα τον Μινώταυρο, το τέρας που τελικά σκοτώθηκε από τον ήρωα Θησέα. Ο Δαίδαλος είχε σχεδιάσει έτσι τον Λαβύρινθο που με δυσκολία κατάφερε να διαφύγει μόλις το έκτισε.

Αν και τα πρώιμα κρητικά νομίσματα εμφανίζουν συχνά διακλαδιστικά (multicursal) σχέδια, ο "κλασικός" σχεδιασμός ενός μονοπατιού με επτά κατευθύνσεις χωρίς διακλάδωση ή αδιέξοδα συνδέθηκε με τον Λαβύρινθο σε νομίσματα ήδη από το 430 π.Χ. και παρόμοια σχέδια χωρίς διακλάδωση χρησιμοποιήθηκαν ευρέως ως οπτικές αναπαραστάσεις του Λαβύρινθου (παρόλο που τόσο οι προφορικές όσο και οι γραπτές περιγραφές καθιστούν σαφές ότι ο Μινώταυρος παγιδεύτηκε σε ένα σύνθετο λαβύρινθο διακλάδωσης). Ακόμα και όταν τα σχέδια έγιναν πιο περίτεχνα, οι οπτικές απεικονίσεις του μυθολογικού λαβυρίνθου από τους Ρωμαϊκούς χρόνους μέχρι την Αναγέννηση είναι σχεδόν πάντα μονοδρομείς (unicursal) . Η διακλάδωση επανεμφανίζεται μόνο όταν οι φυσικοί λαβύρινθοι γίνονται δημοφιλείς κατά την Αναγέννηση.

Εικόνα 2.4 : Ασημένιο νόμισμα από την Κνωσσό που απεικονίζει την κλασσική μορφή των 7 μονοπατιών του λαβυρίνθου

Όσον αφορά τα χαρακτηριστικά του παιχνιδιού που απασχολεί η συγκεκριμένη μεταπτυχιακή διατριβή: Αναπτύχθηκε εφαρμογή/παιχνίδι παραγωγής τέλειων λαβυρίνθων σε περιβάλλον Unity3D. Τόσο οι τοίχοι όσο και τα μονοπάτια στις περισσότερες περιπτώσεις είναι συγκεκριμένα όμως υπάρχουν παιχνίδια λαβυρίνθων που τα μονοπάτια και οι τοίχοι αλλάζουν κατά τη διάρκεια του παιχνιδιού. Η εφαρμογή είναι σε θέση να παράγει κατά το runtime , τέλειους λαβυρίνθους μεγέθους ορισμένου από τον χρήστη και να τους απεικονίζει στο σύνολο της οθόνης ανεξάρτητα από τον αριθμό tiles (κομμάτια στο πάτωμα) από τα οποία αποτελείται. Ως τέλειος λαβύρινθος ορίζεται οποιοσδήποτε λαβύρινθος που δεν έχει μη προσβάσιμες περιοχές, δεν έχει κυκλικά μονοπάτια και ανοιχτές περιοχές και όλα του τα σημεία , tiles στο πάτωμα είναι προσβάσιμα με ένα μονοπάτι να συνδέει οποιοδήποτε ζευγάρι αυτών. Ταυτόχρονα με την δημιουργία του λαβυρίνθου που δημιουργείται κατά το runtime , εμφανίζεται ο χαρακτήρας του παίκτη που έχει την ιδιότητα της κίνησης , της περιστροφής και της εισόδου εξόδου από τον λαβύρινθο μέσω αόρατων colliders τοποθετημένων ακριβώς κάτω από τις περιοχές εισόδου / εξόδου ώστε κατά την επαφή με τον παίκτη να επανεκκινείται η εφαρμογή από την αρχή.

Για τη δημιουργία του λαβυρίνθου χρησιμοποιήθηκε ο αλγόριθμος Hunt n' Kill. Ένας αλγόριθμος που σε καμία περίπτωση δεν είναι βίαιος, όπως προδιαθέτει το όνομα του. Τα βήματα του αλγορίθμου περιληπτικά υλοποιούνται ως εξής :

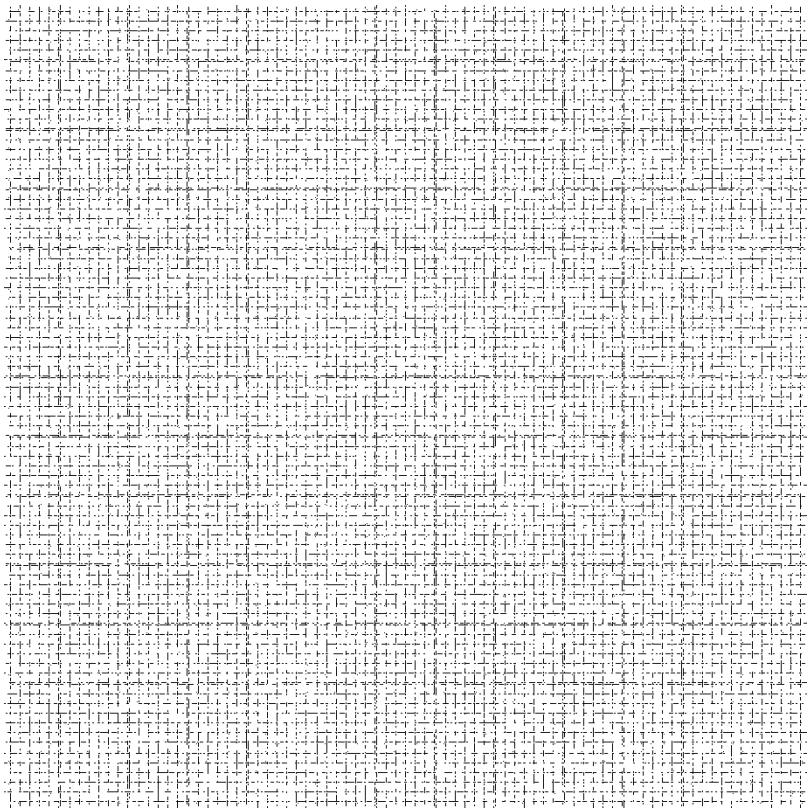
1. Επιλογή θέσης εκκίνησης
2. Τυχαίο βάδισμα χαράζοντας μονοπάτια στους μη επισκεπτόμενους γείτονες έως ότου το τρέχον κελί να μην έχει άλλους μη επισκεπτόμενους γείτονες
3. Ενεργοποίηση κατάστασης κυνηγιού (hunt mode) όπου ελέγχονται (scanning) όλα τα tiles του πίνακα (grid πατώματος) ψάχνοντας κελί που να μην έχει επισκεφθεί και να είναι δίπλα σε τέτοιο που να είναι επισκεπτόμενο. Εάν βρεθεί τέτοιο, τότε χαράζεται μονοπάτι , ανάμεσα στα δύο και το τελευταίο μη επισκεπτόμενο κελί γίνεται αρχική θέση.
4. Επανάληψη βημάτων 2 και 3 έως ότου η κατάσταση κυνηγιού ελέγξει (scanning) όλον τον πίνακα/grid και δεν βρει επισκεπτόμενα κελιά

Το παιχνίδι που αναπτύχθηκε ναι μεν διαθέτει τα χαρακτηριστικά ενός τελικού προϊόντος αλλά σε καμία περίπτωση δεν είναι ένα standalone product με την έννοια ότι δεν θα μπορούσε να αποτελεί μια ολοκληρωμένη λύση ψυχαγωγίας. Αυτό συμβαίνει κυρίως για τους εξής 2 λόγους : Αρχικά για την υλοποίηση και τελειοποίηση ενός ολοκληρωμένου παιχνιδιού (με όλα τα συστατικά του να προέρχονται from scratch) χρειάζονται όχι μόνο πολλοί μήνες / χρόνια ενασχόλησης αλλά συνήθως και ομάδα περισσότερων ατόμων για σωστότερη κατανομή των ρόλων. Αυτοί οι ρόλοι καταλαμβάνουν ένα ευρύτερο φάσμα κατηγοριών-τομέων όπως είναι ο κώδικας, τα γραφικά, η μουσική, το σενάριο κλπ. Λόγω συγκεκριμένης οριοθέτησης στόχων, το συγκεκριμένο παιχνίδι δημιουργήθηκε καθαρά για την αποτύπωση της σωστής λειτουργίας του αλγορίθμου παραγωγής λαβυρίνθων και την παρουσίαση των δυνατοτήτων της μηχανής Unity, σε συνδυασμό με προγραμματισμό στη γλώσσα C#. Επίσης οτιδήποτε εικαστικό χρησιμοποιήθηκε για το παιχνίδι, προέρχεται σε μεγάλο βαθμό από το asset store της Unity.

Ακολουθήθηκε αυτή η τακτική γιατί το art and design κομμάτι δεν καλύπτεται αυτήν την μεταπτυχιακή διατριβή και γιατί θα χρειαζόταν σημαντικά επιπλέον χρόνος (και για ένα πιο επαγγελματικό look, παραπάνω εξοπλισμός π.χ ψηφιακή γραφίδα) για την δημιουργία custom art. Ταυτόχρονα με αυτήν την προσέγγιση γίνεται εύκολα και διακριτά ο διαχωρισμός μεταξύ game art and design και game development. Εύκολα από τα παραπάνω βγαίνει συμπέρασμα ότι ουσιαστικά πρόκειται για ένα demo ενός προϊόντος εν εξέλιξη. Περισσότερα για μελλοντικές επεκτάσεις και τον τρόπο προσέγγισης της συνέχειας της ανάπτυξης της εφαρμογής αναφέρονται στην ενότητα 6 «Συμπεράσματα και μελλοντικές επεκτάσεις».

3. Ανασκόπηση πεδίου

Ο αλγόριθμος Hunt n’ Kill είναι ένας αλγόριθμος διαδικαστικής παραγωγής και έχει πολλές ομοιότητες με τον αναδρομικό αναστροφέα (Recursive Backtracker). Η Μόνη τους διαφορά έγκειται στον τρόπο με τον οποίο διαχειρίζονται τα αδιέξοδα. Επειδή ο Hunt n’ Kill δεν περιλαμβάνει αναδρομικότητα αποφεύγονται υπερχειλίσεις στοίβας (stack overflow), κάτι που αποτελεί ένα από τα μεγάλα προβλήματα του αναδρομικού αναστροφέα σε μεγάλους λαβυρίνθους. Ο αλγόριθμος Hunt n’ Kill επιλέγει μια τυχαία θέση και ξεκινάει την, κίνηση προς μια τυχαία κατεύθυνση. Συνεχίζει την κίνηση μέχρι να βρει αδιέξοδο. Σε αυτό το σημείο ο αναδρομικός αναστροφέας θα έκανε ένα βήμα πίσω αλλά ο Hunt n’ Kill κάνει κάτι διαφορετικό : Αντί να κάνει βήμα πίσω , ψάχνει στον λαβύρινθο για κελί που δεν έχει επισκεφθεί και επαναλαμβάνει τη διαδικασία σε αυτήν την τοποθεσία. Αυτή η διαδικασία συνεχίζεται μέχρι να μην υπάρχουν τέτοιου είδους κελιά. Λόγω αυτής της ιδιαιτερότητας , ο αλγόριθμος μπορεί να είναι λίγο αργός στα τελευταία βήματα και ιδιαίτερα σε μεγάλου μεγέθους λαβυρίνθους όπου πρέπει να προσπελαστούν ολόκληροι για την αναζήτηση κελιού που δεν έχει επισκεφθεί.



Εικόνα 3.1 : Λαβύρινθος μεγέθους 500*500, παραγμένος από τον αλγόριθμο Hunt n' Kill

3.1 Αναλυτική παρουσίαση αλγορίθμου :

Η υλοποίηση του αλγορίθμου είναι σχετικά απλή σε σύγκριση με τους πολλοστούς που επίσης έχουν υλοποιηθεί [υπ1]. Διαθέτει επίσης κάποια optimizations τα οποία εισάγονται εξίσου εύκολα εύκολα , που δεν αποτελούν όμως αντικείμενο ασχολίας της παρούσας πτυχιακής[υπ2]. Η υλοποίηση που επιλέχθηκε εδώ ξεκινάει με την επιλογή ενός τυχαίου αρχικού σημείου και το loop ανάμεσα σε δύο διαφορετικές φάσεις : «περπάτημα» και «κυνήγι» έως ότου η φάση κυνηγιού τερματίσει χωρίς να βρεθεί νέα θέση. Οι υλοποιήσεις περπατήματος και κυνηγιού επιστρέφουν είτε έναν δυαδικό πίνακα (ο οποίος δείχνει τις συντεταγμένες της επόμενης αρχικής θέσης) είτε null (το οποίο δείχνει ότι η φάση που είναι υπό διερεύνηση έχει τερματιστεί).

```
1 x, y = rand(width), rand(height)
2
3 loop do
4   x, y = walk(grid, x, y)
5   x, y = hunt(grid) unless x
6   break unless x
7 end
```

Η υλοποίηση της συνάρτησης walk είναι απλή. Πραγματοποιείται προσπέλαση σε μία τυχαία φτιαγμένη λίστα από κατευθύνσεις και επιστρέφεται null εάν καμία από τις κατευθύνσεις δεν είναι έγκυρη.

```
def walk(grid, x, y)
  [N, S, E, W].shuffle.each do |dir|
    # ...
  end

  nil
end
```

Για κάθε κατεύθυνση υπολογίζεται το γειτονικό κελί σε αυτήν την κατεύθυνση και μετά γίνεται έλεγχος για να διαπιστωθεί εάν το κελί είναι εντός ορίων του λαβυρίνθου και ταυτόχρονα δεν έχει επισκεφθεί.

```
nx, ny = x + DX[dir], y + DY[dir]
if nx >= 0 && ny >= 0 && ny < grid.length && nx < grid[ny].length && grid[ny][nx] == 0
  # ..
end
```

Με το που βρεθεί ένα γειτονικό κελί που πληροί τις παραπάνω προϋποθέσεις τότε συνδέεται με το τρέχον κελί και επιστρέφεται ως νέο τρέχον κελί.

```
grid[y][x] |= dir
grid[ny][nx] |= OPPOSITE[dir]
return [nx, ny]
```

Αντίθετα, η φάση κυνηγιού είναι λίγο πιο σύνθετη. Κάνει iteration σε κάθε κελί στο πλέγμα, ανά γραμμή, αγνοώντας όσα ήδη έχουν επισκεφθεί και επιστρέφοντας null αν δεν βρεθούν κελιά που δεν έχουν επισκεφθεί.

```
def hunt(grid)
  grid.each_with_index do |row, y|
    row.each_with_index do |cell, x|
      next unless cell == 0

      # ...
    end
  end

  nil
end
```

Στη συνέχεια , μέσα σε αυτόν τον εσωτερικό βρόγχο , εάν το κελί δεν έχει επισκεφθεί , υπολογίζεται μια λίστα από κελιά τα οποία είναι ήδη μέρος του λαβυρίνθου.

```
neighbors = []
neighbors << N if y > 0 && grid[y-1][x] != 0
neighbors << W if x > 0 && grid[y][x-1] != 0
neighbors << E if x+1 < grid[y].length && grid[y][x+1] != 0
neighbors << S if y+1 < grid.length && grid[y+1][x] != 0
```

Μετά επιλέγεται τυχαία ένα από αυτά (προχωρώντας στο επόμενο κελί εάν δεν υπάρχει διαθέσιμο γειτονικό κελί για να επιλεγεί από τα προηγούμενα).

```
direction = neighbors[rand(neighbors.length)] or next
```

Και τέλος υπολογίζονται οι συντεταγμένες του γειτονικού κελιού στην επιλεγμένη κατεύθυνση , χαράσσεται το μονοπάτι και επιστρέφονται οι νέες συντεταγμένες.

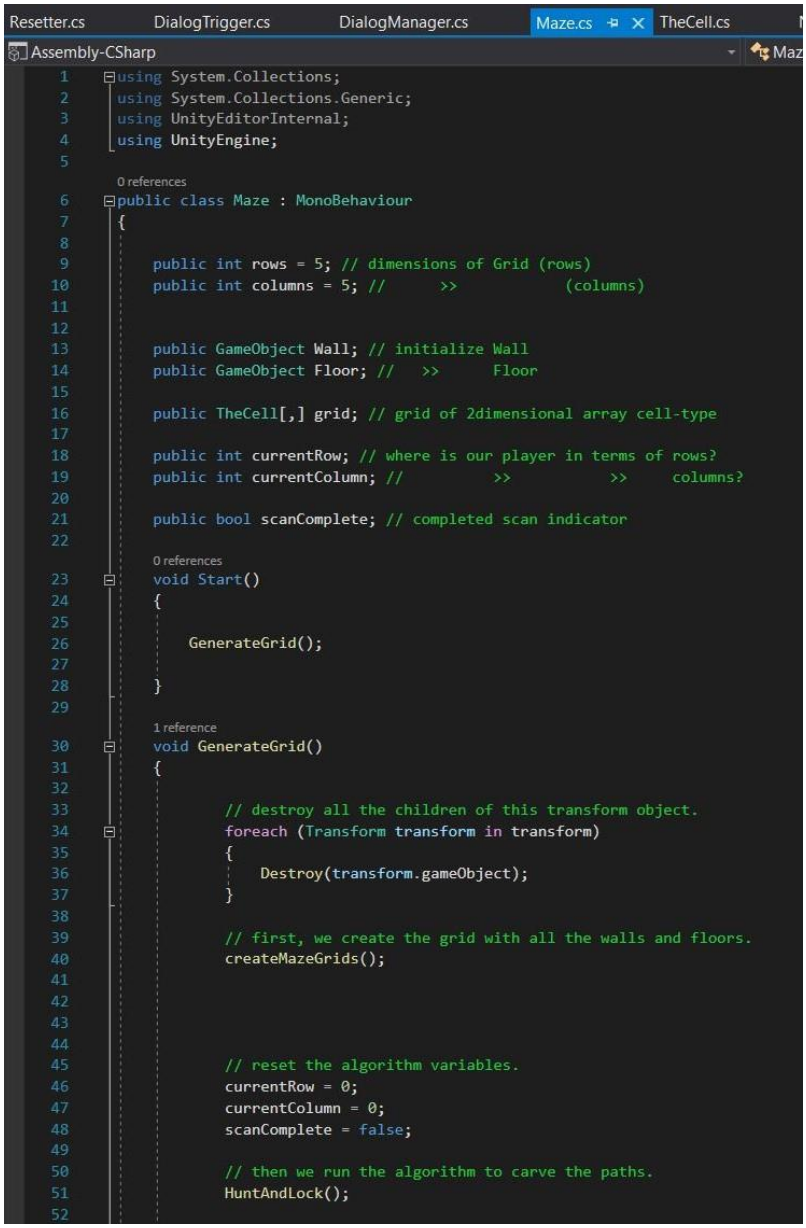
```
nx, ny = x + DX[direction], y + DY[direction]

grid[y][x] |= direction
grid[ny][nx] |= OPPOSITE[direction]

return [x, y]
```

Και έτσι ολοκληρώνονται τα βήματα του αλγορίθμου hunt n kill.

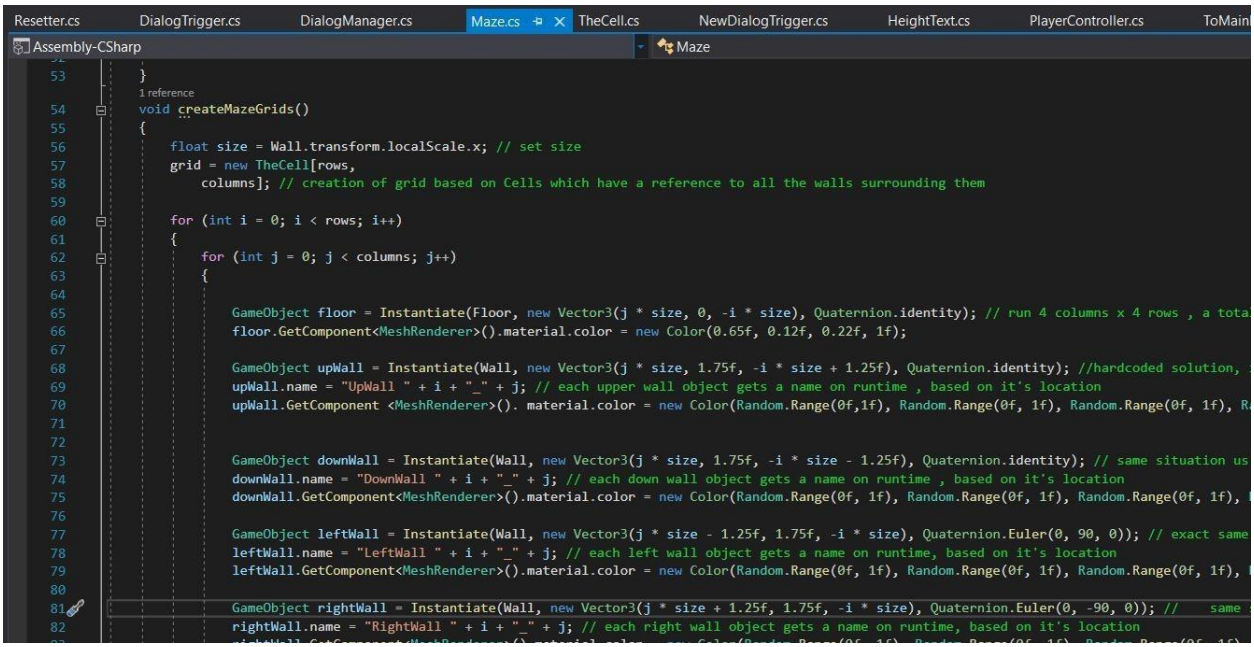
υπ1 Τα screenshots κώδικα είναι σε objective C, από την αρχική υλοποίηση του αλγόριθμου, στη συνέχεια ακολουθούν αποσπάσματα από την υλοποίηση στο script , σε C#.

υπ2 Ένας από τους πολλούς τρόπους για βελτιστοποίηση της λειτουργίας του αλγορίθμου είναι η επιτάχυνση της φάσης κυνηγιού μέσω απομνημόνευσης της πρώτης γραμμής που περιέχει κελιά τα οποία δεν έχουν επισκεφθεί.

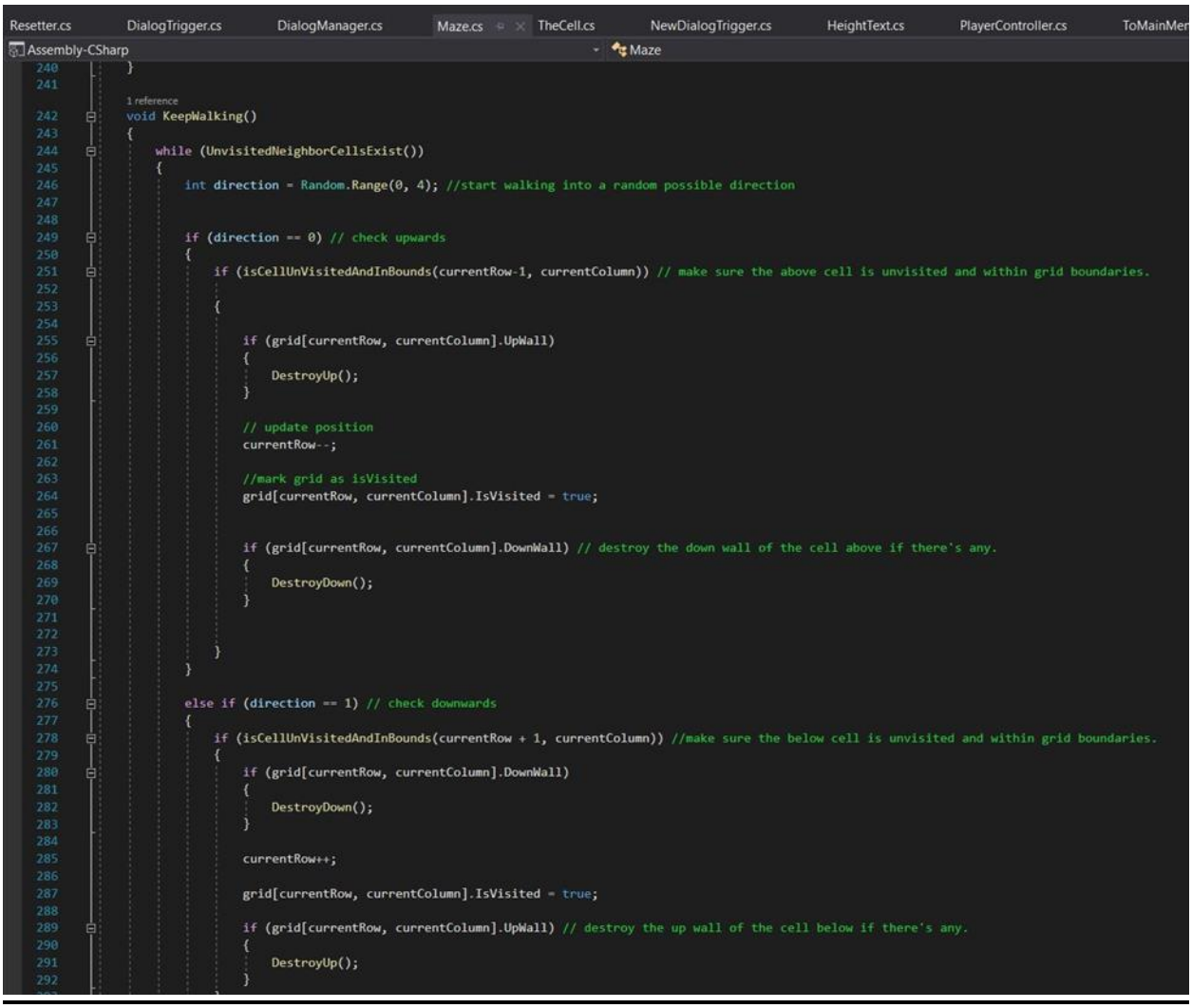
υπ3 Σε πολλά από τα παραπάνω αποσπάσματα κειμένου αναφέρεται ο όρος «null» ενώ στον κώδικα παρατηρείται ο όρος «nil». Η μοναδική διαφορά είναι ότι το «nil» είναι απλά ένας null pointer σε ένα αντικείμενο στην Objective-C και όπως θα φανεί και στο απόσπασμα της C# το αποτέλεσμα είναι το ίδιο, ότι δηλαδή η φάση υπό διερεύνηση έχει τερματιστεί.



Εικόνα 3.2 : Script παραγωγής λαβυρίνθου (αρχικοποίηση)



Εικόνα 3.3 : Script παραγωγής λαβυρίνθου (spawning κελιών και χρωματισμός)



Εικόνα 3.4 : Script παραγωγής λαβυρίνθου (σκανάρισμα κελιών και καταστροφή τοίχων)

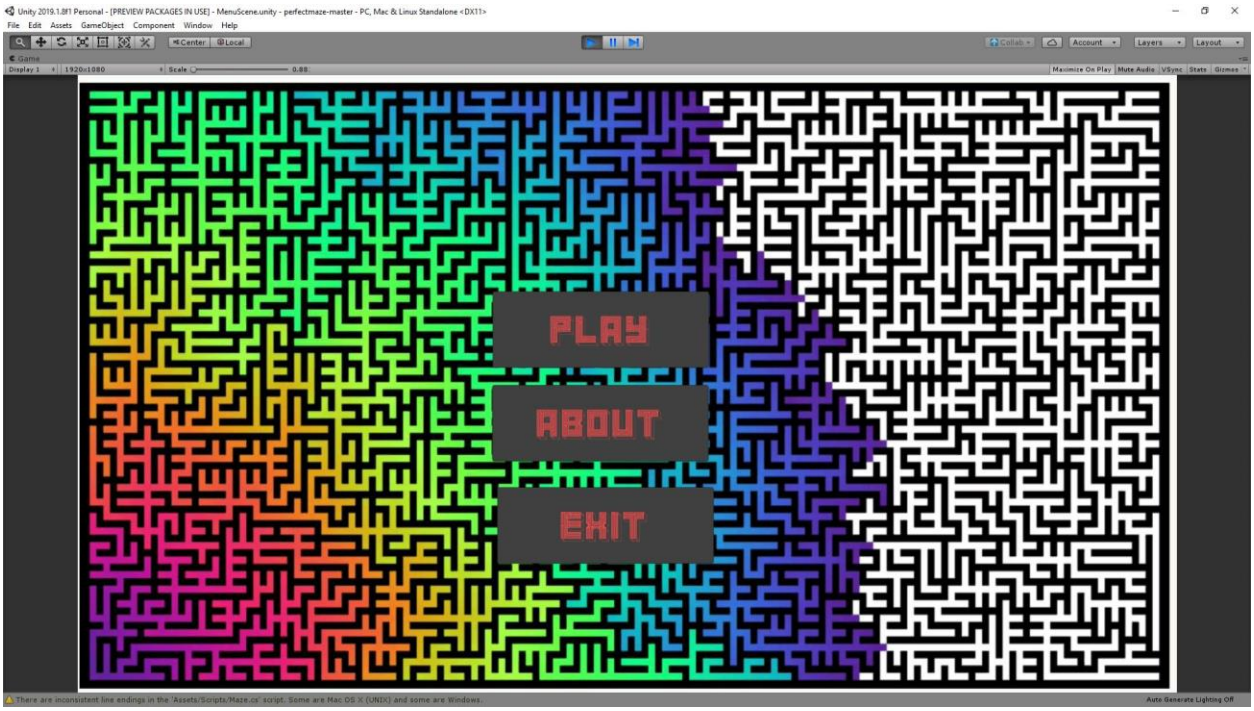
4. Game Manual

Perfect Maze Generator (Ελληνική Περιγραφή) :

Διασκεδάστε με τον παραγωγό τυχαίων λαβυρίνθων. Ο αλγόριθμος που δουλεύει στο παρασκήνιο επιβεβαιώνει ότι ΚΑΝΕΝΑΣ λαβύρινθος δεν είναι ίδιος με τον προηγούμενο. Μπορείς να βρεις την έξοδο? Αν ναι , πέσε και ξαναπροσπάθησε από την αρχή! Η διασκέδαση δεν τελειώνει ποτέ με το PMG!

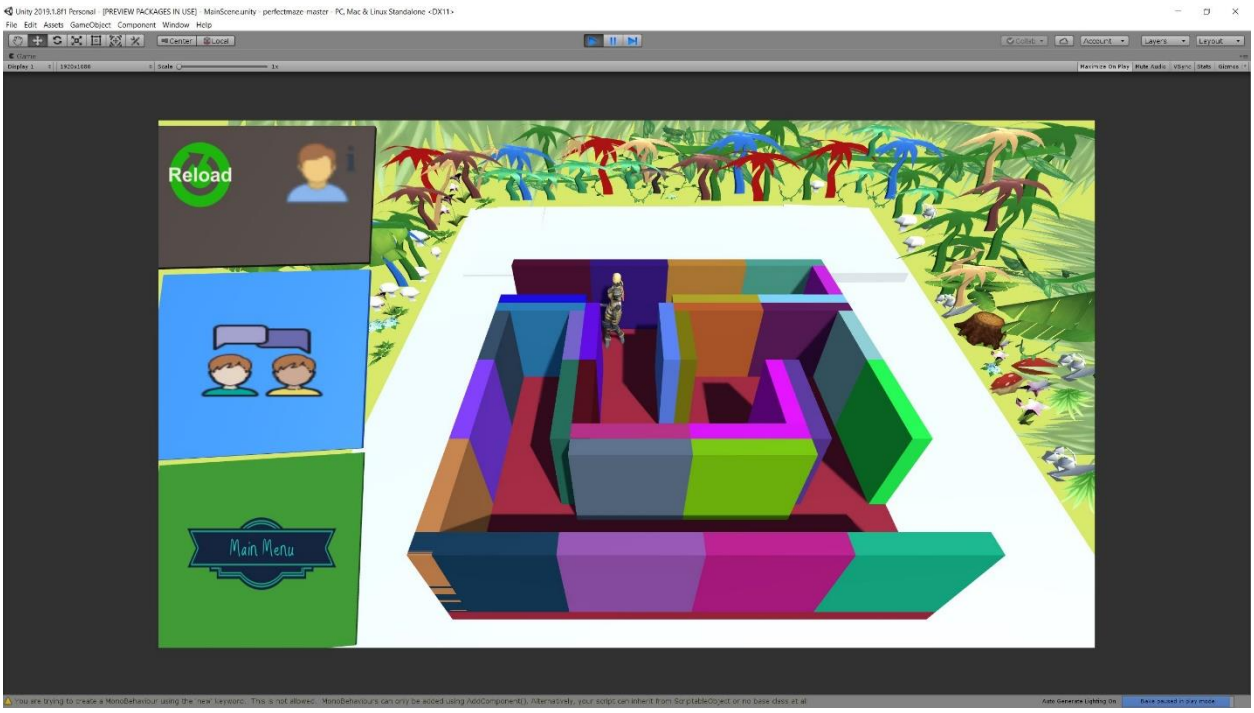
Perfect Maze Generator (English description) :

Have fun while navigating in randomly generated mazes! The underlying algorithm will make sure that NO maze is the same as the one generated beforehand. Can you find the exit? If yes , spawn again and try anew! The fun is never ending in PMG!



Εικόνα 4.1 : Η οθόνη Main Menu

Ως υποδοχή για τον παίκτη που εκκινεί το παιχνίδι, έχει οριστεί η σκηνή που φαίνεται στην εικόνα 4.1. Η πλοήγησή του για τις απαραίτητες επιλογές (μετάβαση στο παιχνίδι (Play) , πλαίσιο πληροφοριών (Infos) , έξοδος από το παιχνίδι (Exit) πραγματοποιείται με το ποντίκι.



Εικόνα 4.2 : Η οθόνη παιχνιδιού

Ως game scene έχει οριστεί η οθόνη που φαίνεται στην εικόνα 4.2. Αυτή αποτελεί και την μοναδική σκηνή του παιχνιδιού. Όλες οι ενέργειες και το παιχνίδι λαμβάνουν χώρα εδώ.

4.1 UI Elements

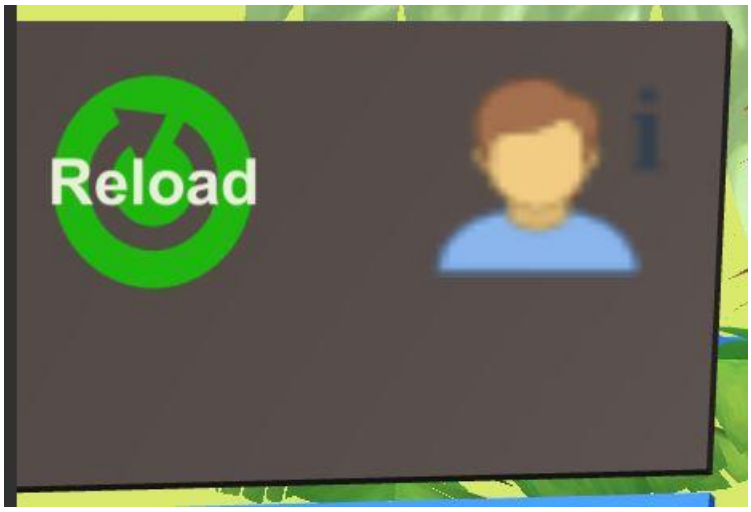
Στην αριστερή πλευρά της οθόνης βρίσκονται τα UI elements του παιχνιδιού. Είναι προσβάσιμα με την χρήση του ποντικιού και διαχωρίζονται σε 3 ευδιάκριτα, διαφορετικού χρώματος panes (τζάμια).

1 ο pane :

Το πρώτο pane, χρώματος μαύρο που διακρίνεται επάνω αριστερά περιλαμβάνει 2 κουμπιά.

Κουμπί Reload : Το πράσινο κουμπί με το γυριστό βέλος χρησιμοποιείται για την επαναφόρτιση της σκηνής που έχει ως αποτέλεσμα τόσο την δημιουργία νέου λαβυρίνθου όσο και την επαναφορά του χαρακτήρα στο αρχικό σημείο spawn. Δεν υπάρχει ίδιος λαβύρινθος κατά το reload, ο αλγόριθμος κάθε φορά δημιουργεί νέα μονοπάτια.

Κουμπί Info : Το μπλε κουμπί με τον βοηθό χρησιμοποιείται για την εμφάνιση του παραθύρου πληροφοριών. Ο παίκτης μπορεί να βγάλει χρήσιμα συμπεράσματα για το παιχνίδι αλλά και να μάθει περισσότερα για το πως χτίζεται ο λαβύρινθος πατώντας το κουμπί αυτό.

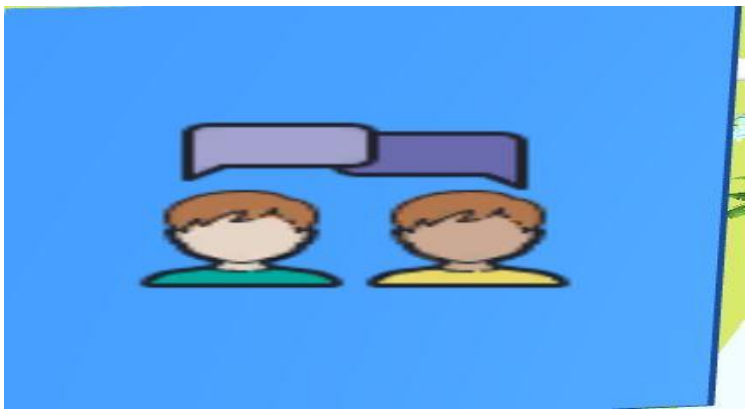


Εικόνα 4.1.1 : Η αποτύπωση του 1ου pane

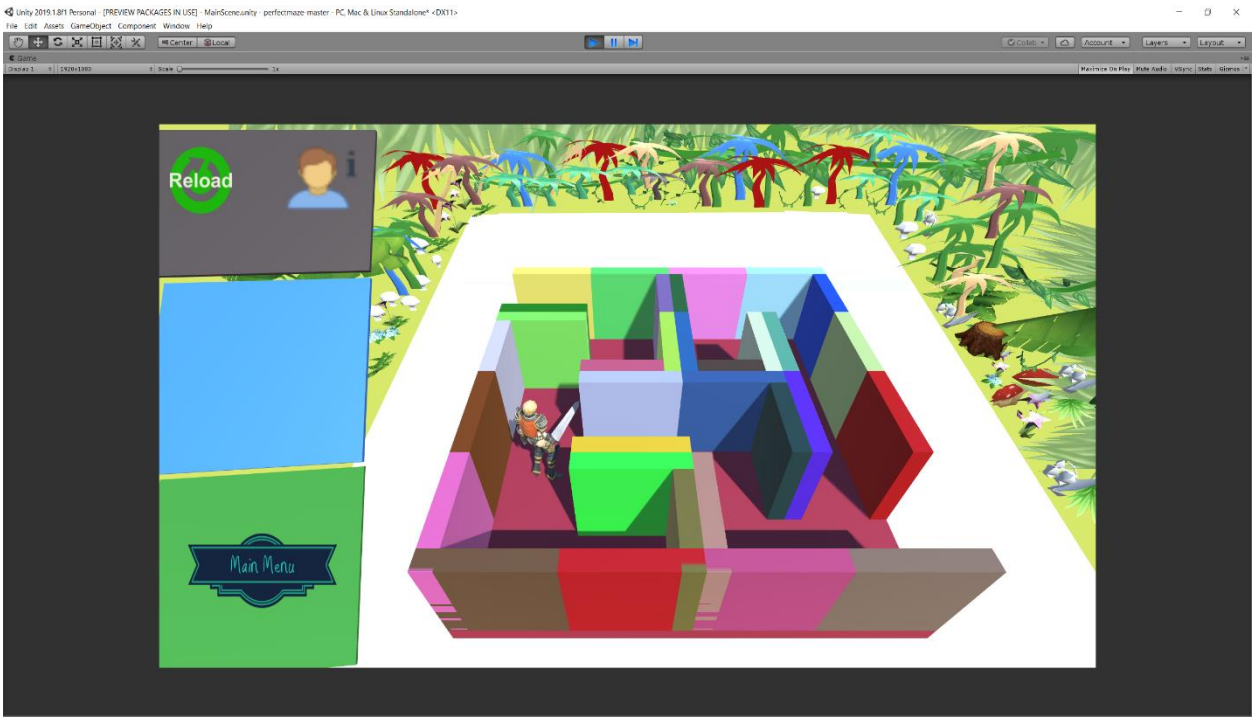
2ο pane :

Το δεύτερο pane στη μέση χρώματος μπλε περιέχει το κουμπί διαλόγου.

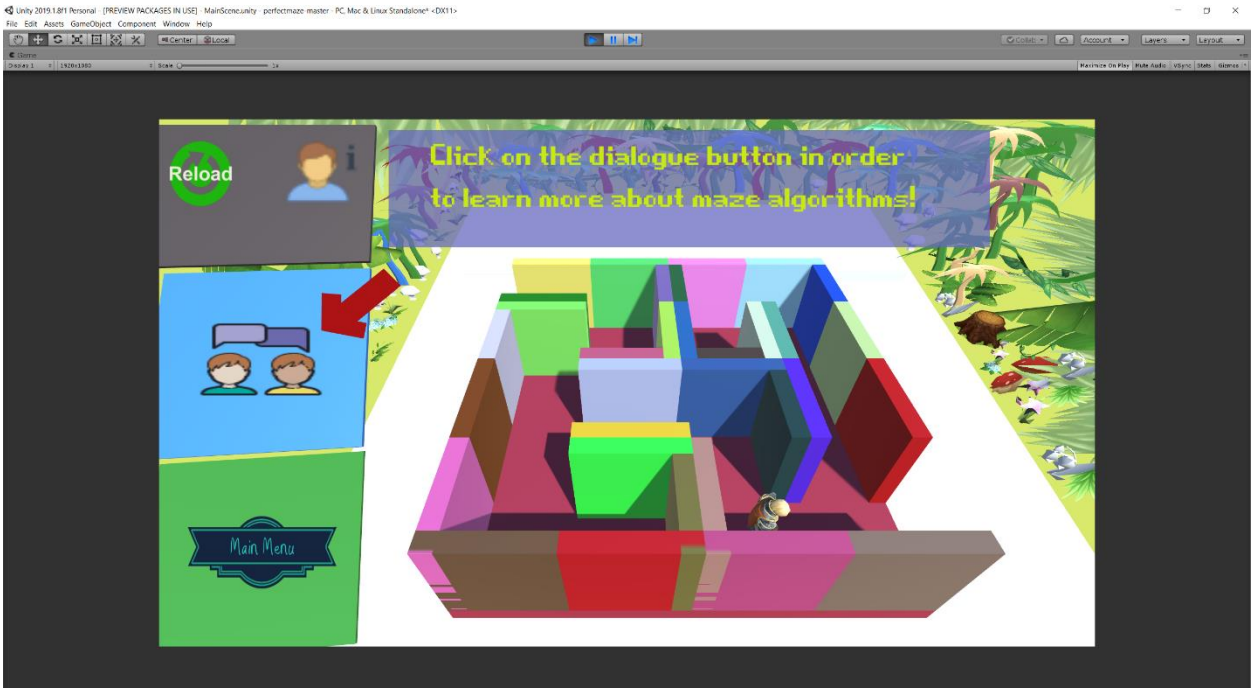
Κουμπί Dialog : Ουσιαστικά με αυτό το κουμπί ο παίκτης είναι σε θέση να εκκινεί μια συνομιλία με το σύστημα που ως περιεχόμενο έχει την αποτίμηση των γνώσεων του παίκτη για τους αλγορίθμους και τους λαβυρίνθους γενικότερα. [υπ1]



Εικόνα 4.1.2 : Η αποτύπωση του 2ου pane



Εικόνα 4.1.3: Το περιβάλλον παιχνιδιού πριν ο χαρακτήρας εισέλθει στη νότια περιοχή με το collider



Εικόνα 4.1.4: Το περιβάλλον παιχνιδιού αφότου ο χαρακτήρας εισέλθει στη νότια περιοχή με το collider

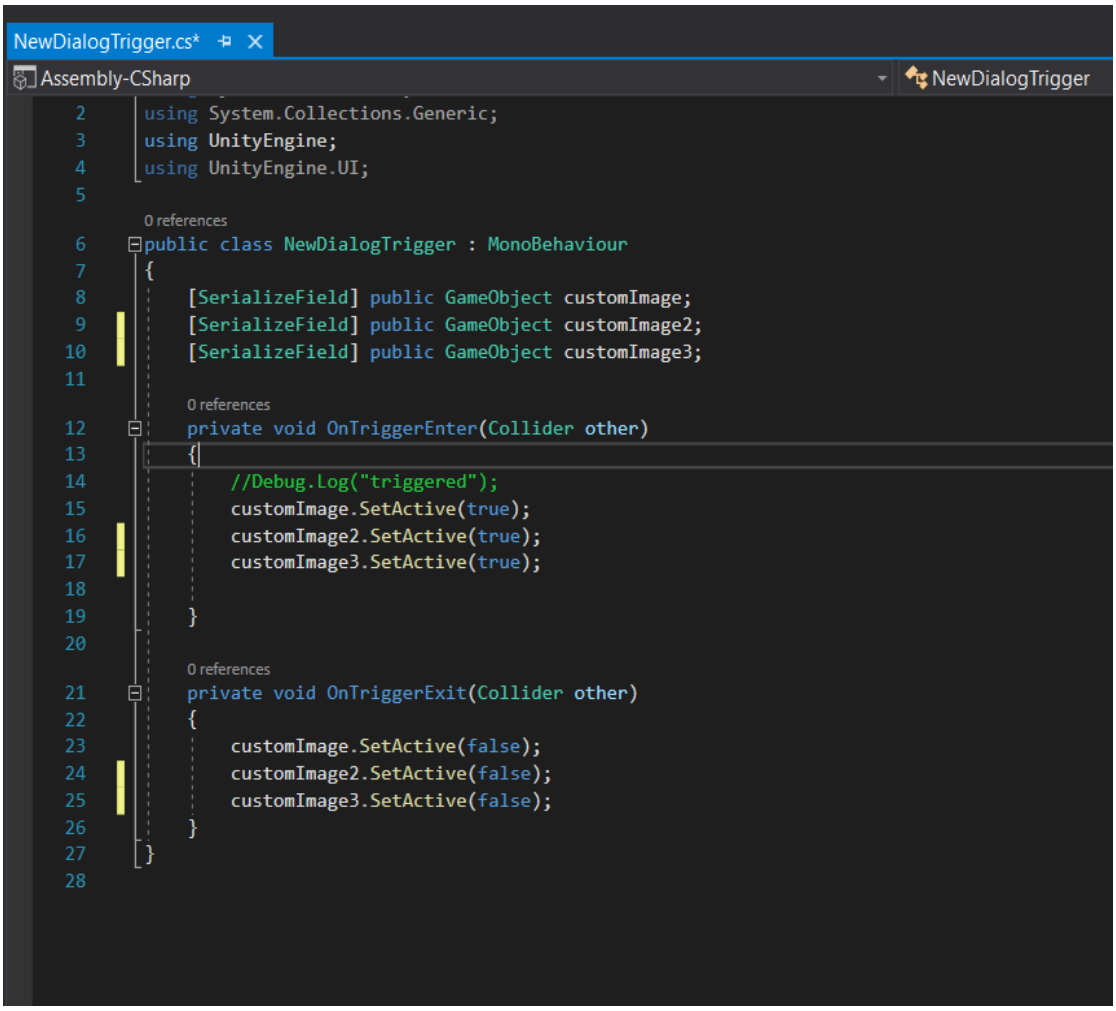
3 ο pane :

Το τρίτο pane κάτω αριστερά χρώματος πράσινο περιέχει το κουμπί «Main Menu»

Κουμπί Main menu : Πατώντας αυτό το κουμπί ο χρήστης επιστρέφει στην αρχική οθόνη.

Εικόνα 4.1.5 : Η αποτύπωση του 3ου pane

υπ1 Για την ορθή κατανόηση του χρονικού σημείου που θα ήταν βέλτιστο να πατήσει ο παίκτης το κουμπί και επειδή ήταν πιο σύνθετη μια υλοποίηση με αυτόματη εκτέλεση των λειτουργιών του κουμπιού διαλόγου, εσκεμμένα το κουμπί αυτό μαζί με τον διάλογο και το βέλος που κάνει prompt τον παίκτη να το πατήσει εμφανίζεται μόνο όταν ο χαρακτήρας πλοηγείται σε ένα συγκεκριμένο σημείο στον λαβύρινθο και συγκεκριμένα στο νοτιότερο μέρος του. Και αυτή η λειτουργία όπως και η επανεκκίνηση στην περίπτωση που ο χαρακτήρας βγει εκτός ορίων λαβυρίνθου ή τερματίσει το επίπεδο, είναι εφικτή με τη χρήση colliders.



Εικόνα 4.1.6 : Η χρήση collider (μέσω script) για την εμφάνιση/απόκρυψη στοιχείων του διαλόγου

5. Αρχιτεκτονική Συστήματος

Όσον αφορά τη διαδικασία που ακολουθήθηκε για τη δημιουργία, εισαγωγή και ορθή εναρμονισμένη λειτουργία των assets και prefabs στο περιβάλλον Unity3D ακολουθήθηκε η εξής διαδικασία :

Δημιουργία Τοίχων και Πλέγματος (οι διαστάσεις του πίνακα μπορούν να τεθούν τόσο από το ίδιο το script τόσο και από τα fields που έχουν γίνει serialized)

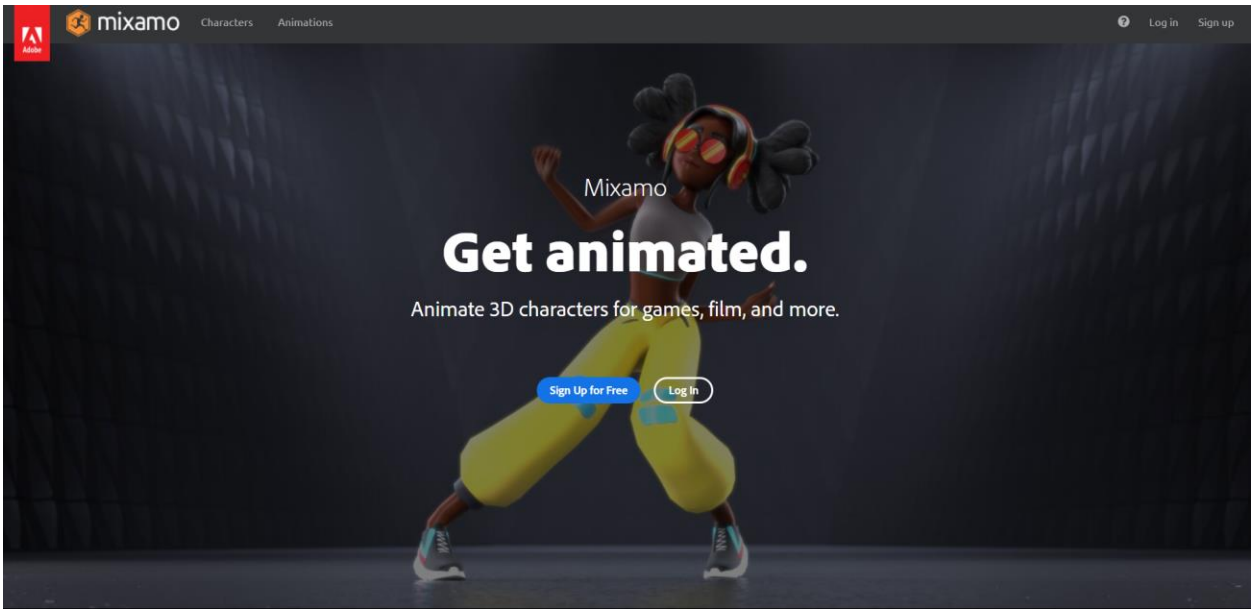
1. Δημιουργία κύβου 3 x 3 x 0.5 και εισαγωγή του ως prefab
2. Δημιουργία C# script
3. Δημιουργία πατώματος (3 x 0.5 x 3)
4. Αρχικοποίηση αντικειμένων παιχνιδιού (GameObjects) στο script Maze και εισαγωγή references στη Unity.

Όσον αφορά τον κώδικα για τη δημιουργία της εφαρμογής ακολουθήθηκε συγκεκριμένη προσέγγιση για την επίλυση του προβλήματος.

Μέσω των παρακάτω συναρτήσεων και την αλληλεπίδραση αυτών επιτεύχθηκε η σωστή δημιουργία τέλειου λαβυρίνθου κάθε φορά που είτε το παιχνίδι ξεκινάει από την αρχή είτε ο χρήστης πατάει το κουμπί reload κατά το runtime. Επίσης τόσο με το UI element reload maze όσο και με την παραδοσιακή επανεκκίνηση του παιχνιδιού , ο λαβύρινθος που δημιουργείται κάθε φορά έχει διαφορετικά μονοπάτια και ανταποκρίνεται στο μέγεθος που έχει θέσει ο προγραμματιστής μέσα στο script δημιουργίας του λαβυρίνθου πριν την εκκίνηση της εφαρμογής.

5.1 Χαρακτήρας και κίνηση

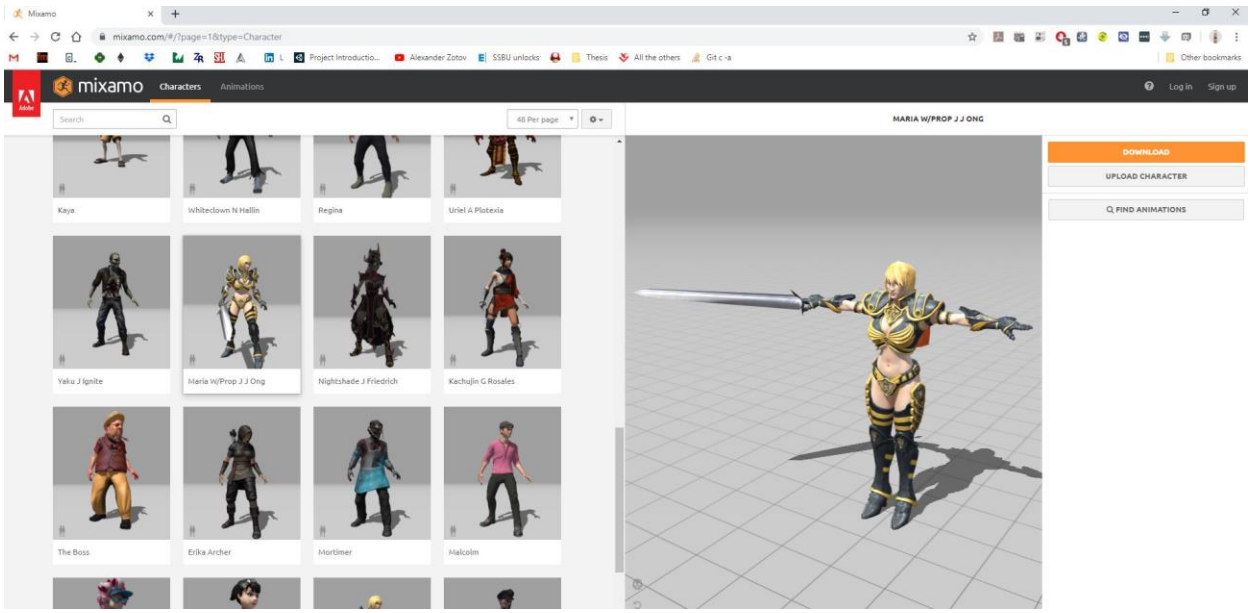
Η εισαγωγή μοντέλου χαρακτήρα που να διαθέτει κίνηση ανάλογα με το input που δίνει ο παίκτης από το πληκτρολόγιο χωρίς κολλήματα και bugs κατά την πλοήγηση στο λαβύρινθο ήταν μια αρκετά απαιτητική διαδικασία. Λόγω έλλειψης εγγενούς design custom χαρακτήρα και γνώσεων για το πως αυτό θα ήταν εφικτό να πραγματοποιηθεί (ανάγκη για third party applications όπως Blender, Harmony ή Photoshop) ο χαρακτήρας που επιλέχθηκε για τις ανάγκες της πτυχιακής (Maria Paladin) προέρχεται από την ιστοσελίδα mixamo.com . Σε αυτόν τον ιστότοπο διατίθενται δωρεάν μοντέλα χαρακτήρων σε μορφές αρχείων που υποστηρίζονται από τη Unity, έτσι ώστε εύκολα οι ενδιαφερόμενοι να είναι σε θέση να τα εισάγουν στο παιχνίδι τους.



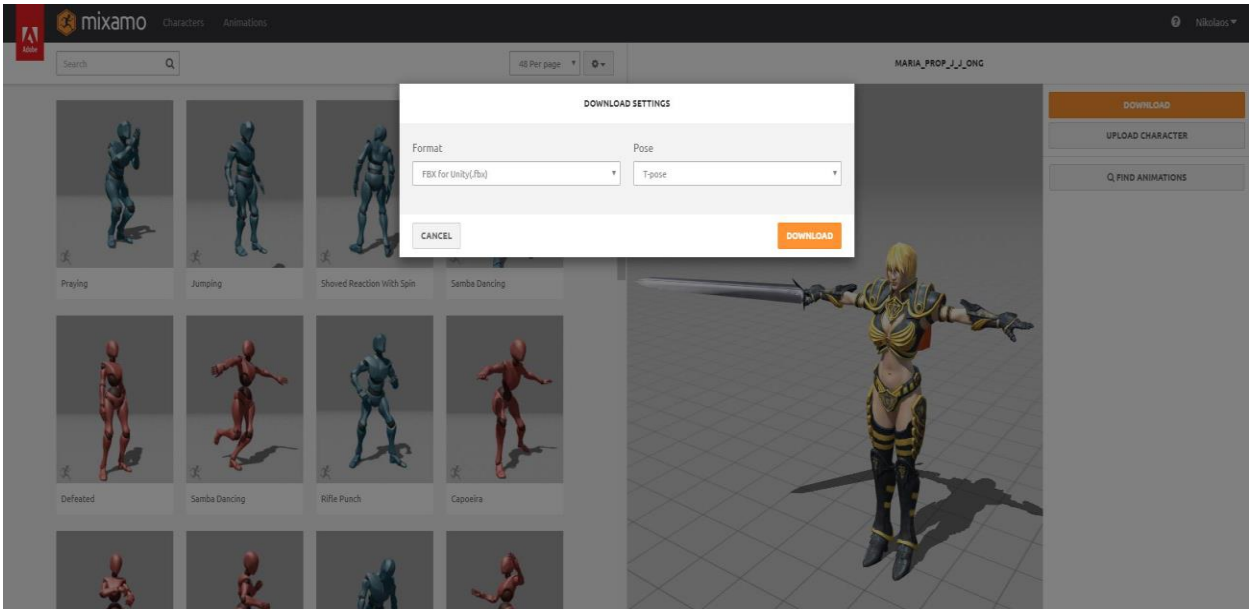
Εικόνα 5.1.1 : Η εισαγωγική σελίδα στο mixamo.com

Αυτή η διαδικασία για να γίνει ολοκληρωμένα και χωρίς πιθανότητες λανθασμένης εισαγωγής , απαιτεί συγκεκριμένα βήματα.

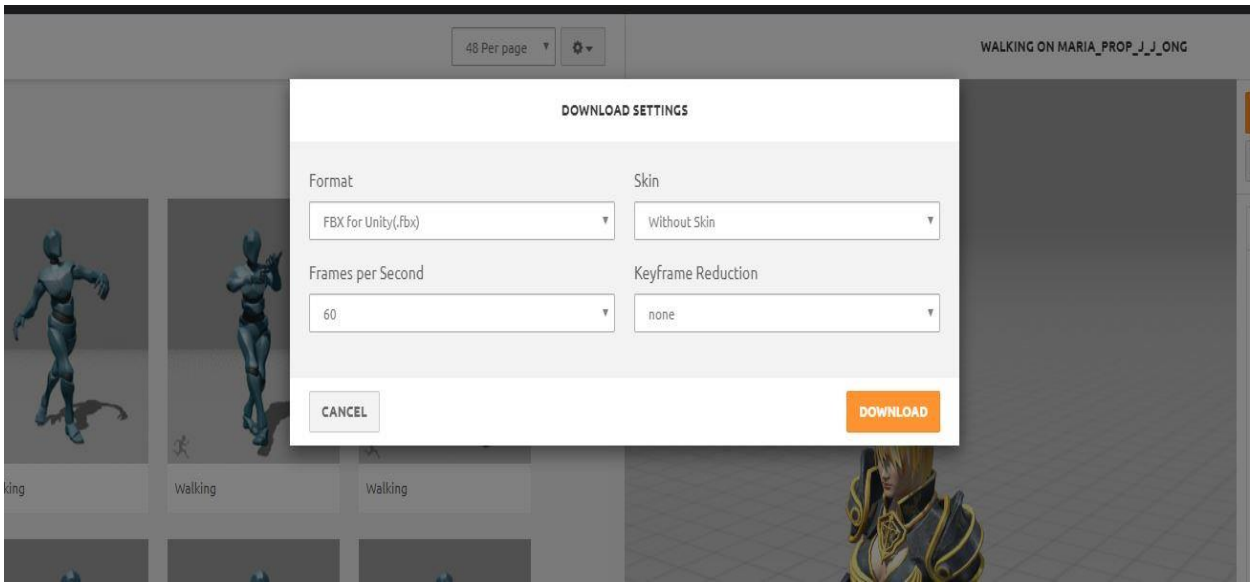
Το πρώτο pack που θα πρέπει να κατεβάσει ο χρήστης από το site είναι το μοντέλο μαζί με το skin σε μορφή fbx for unity. Για το συγκεκριμένο παιχνίδι επιλέχθηκε framerate 60fps αλλά αυτό είναι στην κρίση του ενδιαφερόμενου ανάλογα με τις ανάγκες τις εκάστοτε εφαρμογής και τις απαιτήσεις συστήματος. Για αρχή επιλέγεται η t-pose στάση που δεν περιέχει κίνηση. Έπειτα με αλλαγή στο tab Character / Animations επιλέγονται οι κινήσεις που χρειάζονται ανάλογα με τις προδιαγραφές της κίνησης που έχει ορίσει ο κάθε προγραμματιστής.



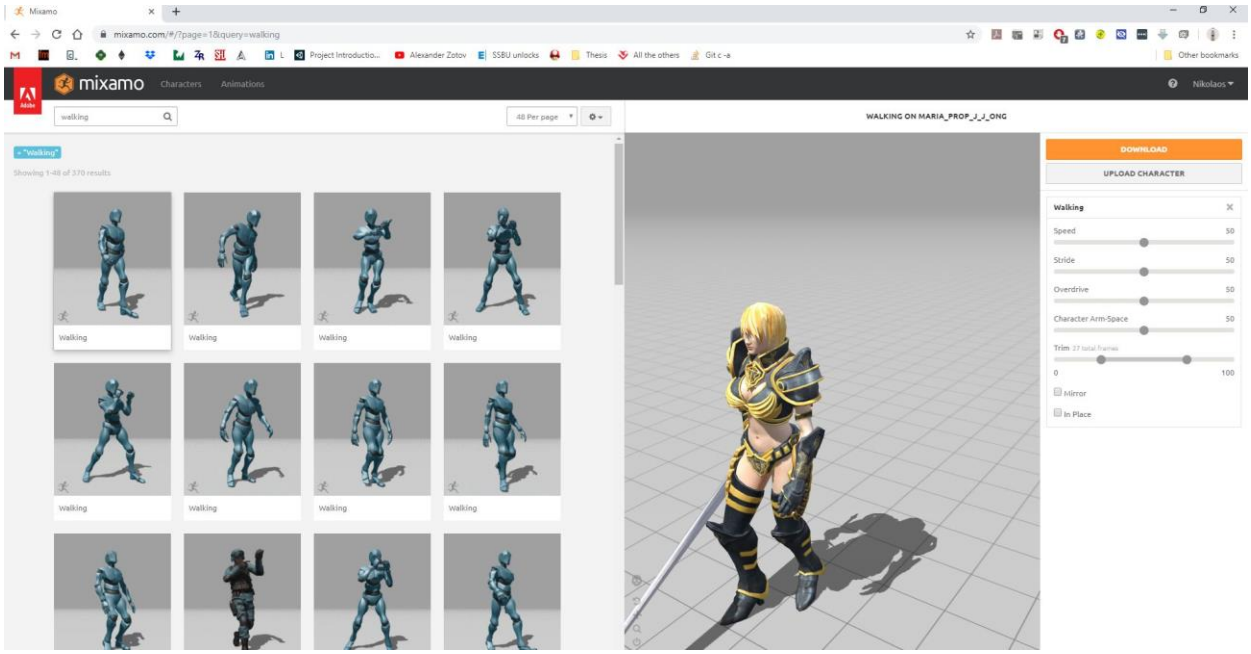
Εικόνα 5.1.2 : Η σελίδα επιλογής μοντέλου χαρακτήρα στο mixamo.com



Εικόνα 5.1.3 : Η σελίδα επιλογής τύπου μοντέλου στο mixamo.com



Εικόνα 5.1.4 : Η σελίδα επιλογής τύπου μοντέλου και framerates στο mixamo.com

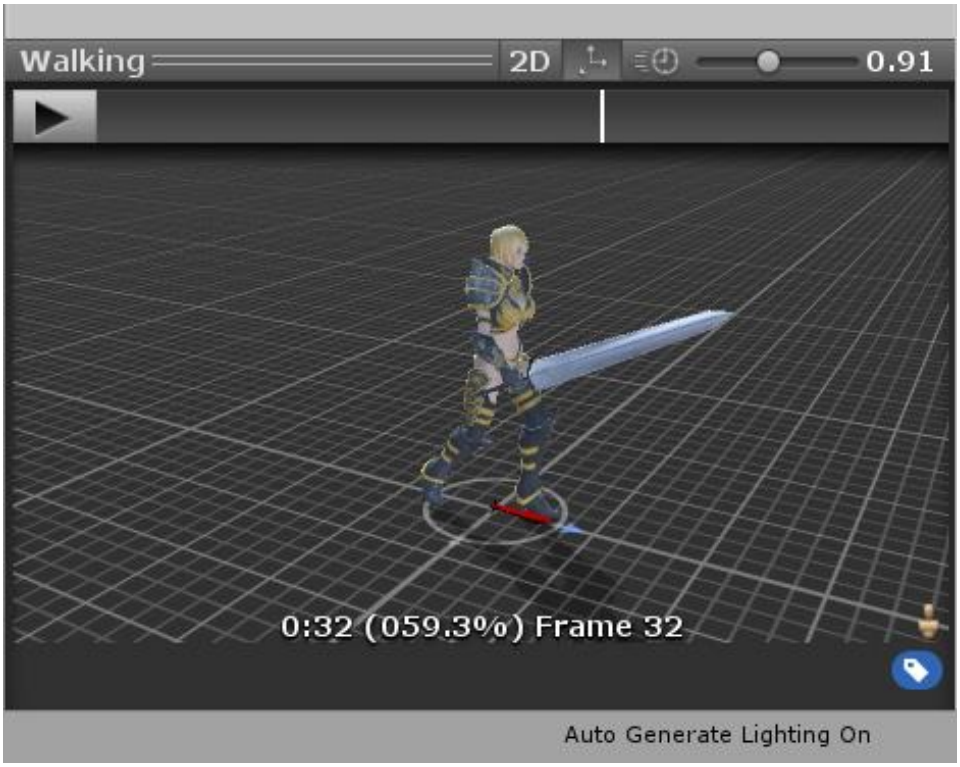


Εικόνα 5.1.5 : Η σελίδα επιλογής πακέτων κινήσεων μοντέλου στο mixamo.com

Από τον file explorer της Unity και μετά τη δημιουργία του αντίστοιχου φακέλου , με δεξί κλικ – Import Asset επιλέγεται το αρχείο του μοντέλου χωρίς κίνηση που κατέβηκε από το mixamo. Στο παράθυρο που εμφανίζεται τις περισσότερες φορές και ζητάει διορθώσεις γίνεται κλικ στο κουμπί fix now. Στη συνέχεια είναι απαραίτητο να γίνει extract features και extract textures κατά προτίμηση σε διαφορετικούς φακέλους για καλύτερη ταξινόμηση. Έτσι το μοντέλο του χαρακτήρα μπορεί άμεσα να εισαχθεί στην σκηνή που δουλεύεται την εκάστοτε στιγμή στο περιβάλλον της Unity.

Εικόνα 5.1.6 : Η περιοχή επεξεργασίας κίνησης χαρακτήρα στη Unity

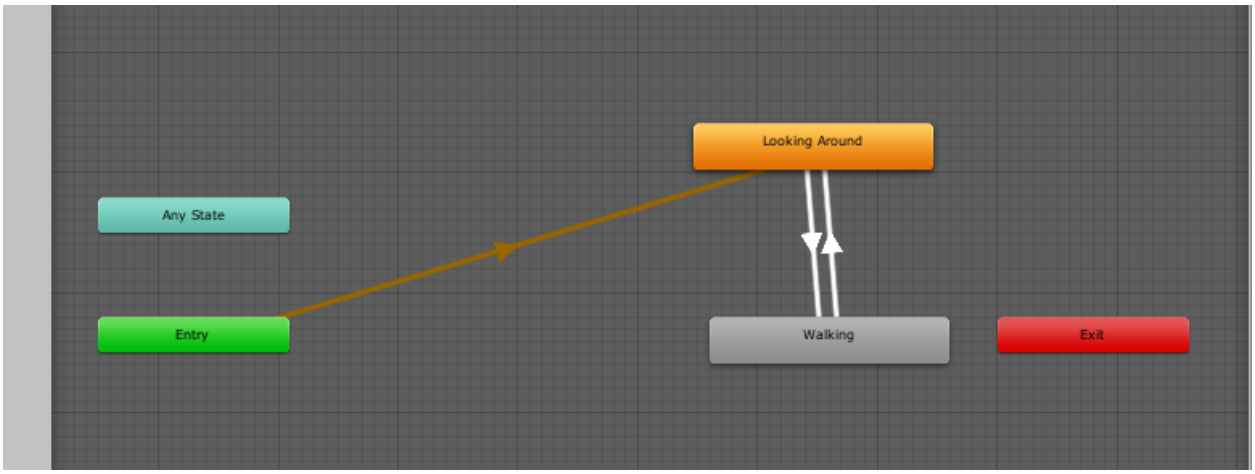
Στη συνέχεια σε διαφορετικό φάκελο (στην προκειμένη, φάκελος animations) με την ίδια διαδικασία εισάγονται και οι κινήσεις του χαρακτήρα.



Εικόνα 5.1.7 : Η περιοχή επεξεργασίας κίνησης χαρακτήρα στη Unity

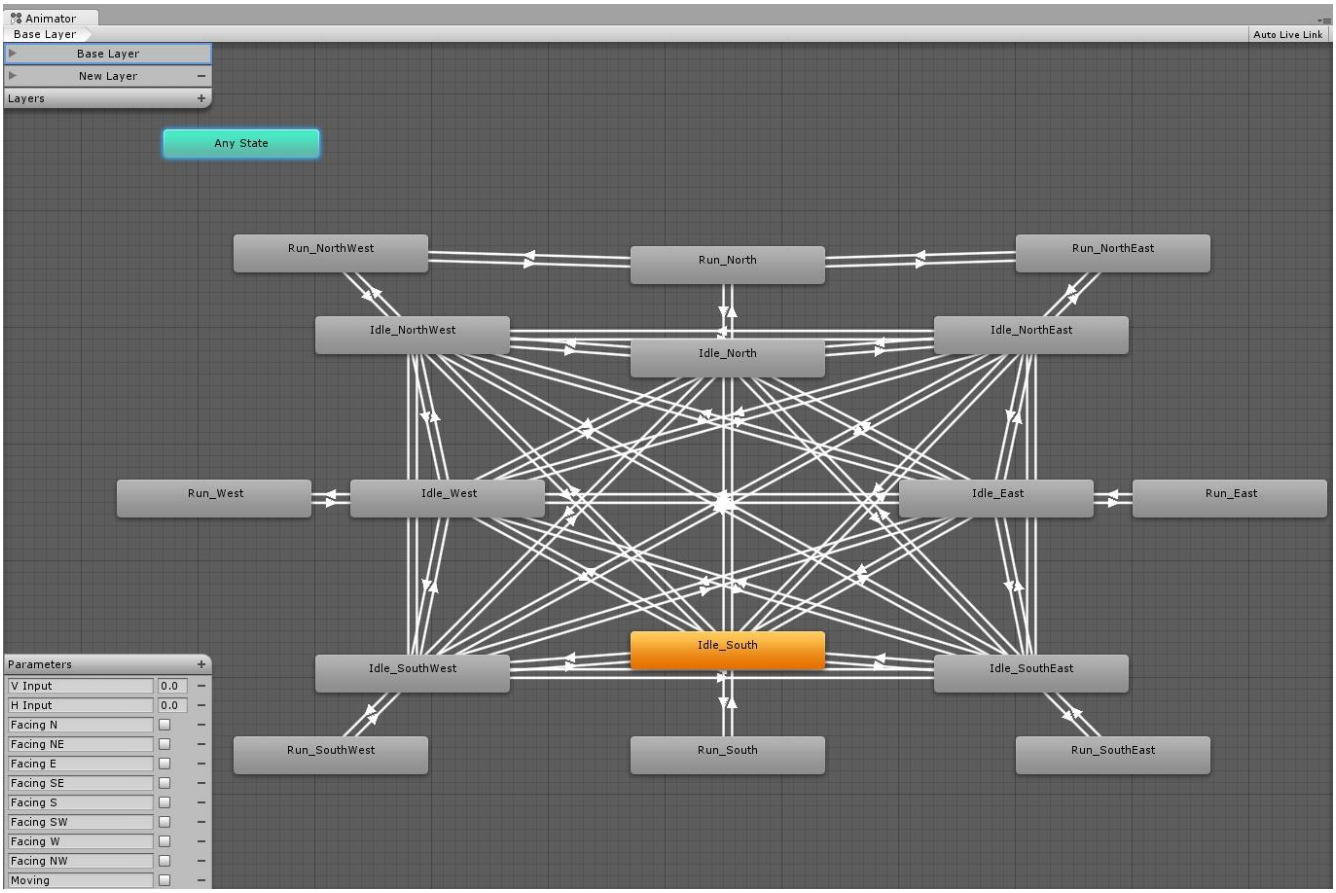
Δημιουργία animator states :

Για την κίνηση του χαρακτήρα δημιουργήθηκε ένα πλάνο με animations και transitions μεταξύ αυτών. Αυτό το πλάνο αντικατοπτρίζει τις εναλλαγές μεταξύ των states του χαρακτήρα κατά την κίνηση του.



Εικόνα 5.1.8 : Το set κινήσεων που αναπτύχθηκε για τον χαρακτήρα

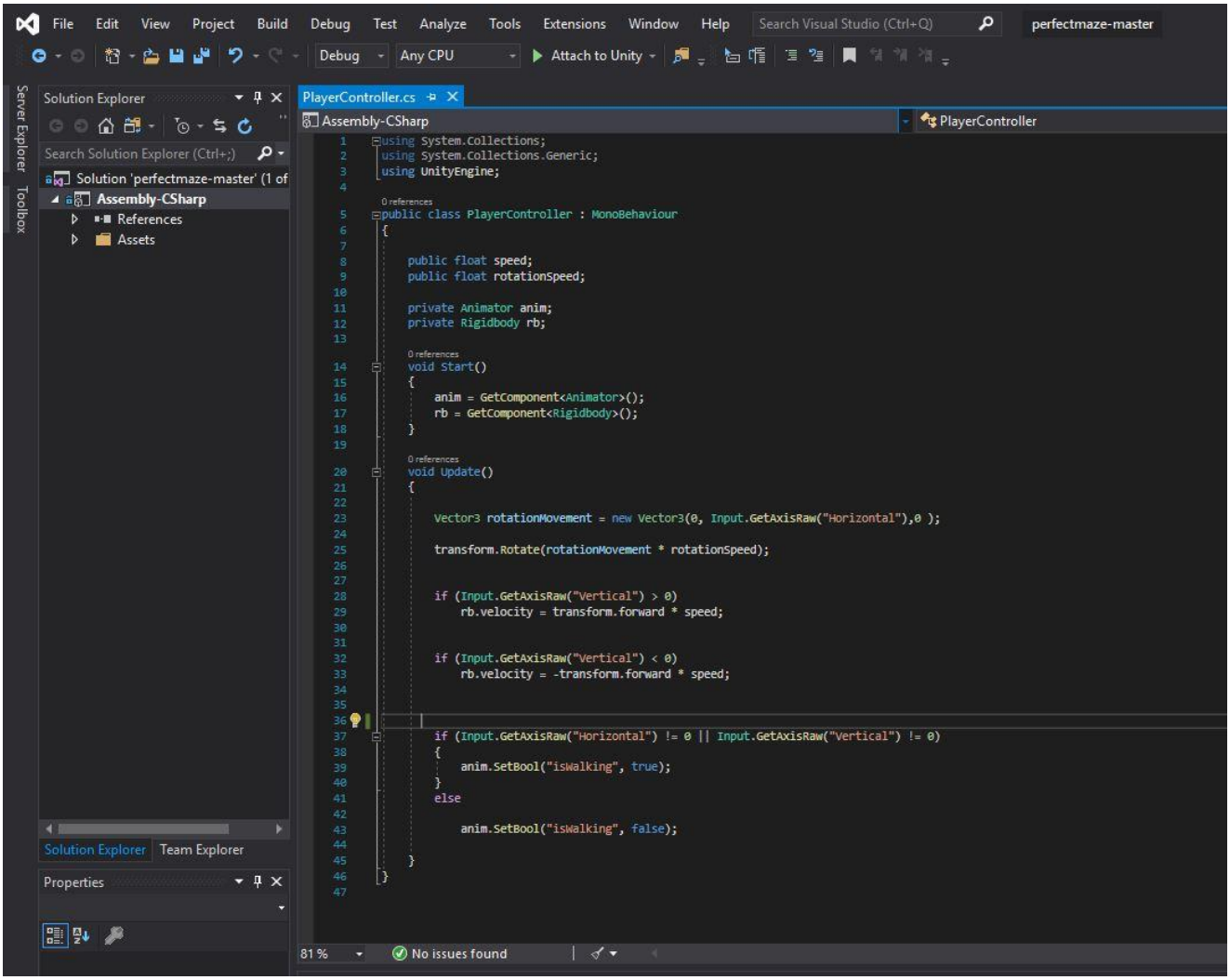
Γίνεται αρκετά κατανοητό από την εικόνα ότι πρόκειται για έναν animator ο οποίος είναι πολύ απλός σε υλοποίηση. Αυτό συμβαίνει γιατί πέρα από την εναλλαγή από στάση (idle/ looking around) σε περπάτημα (walking) ο χαρακτήρας δεν έχει προγραμματιστεί να κάνει απολύτως τίποτε άλλο. Τόσο ο στόχος για απλότητα όσο και το περιβάλλον του λαβυρίνθου που λόγω κατασκευής περιορίζει λοιπές κινήσεις όπως jumping λόγω των τοίχων και του ύψους του χαρακτήρα οδήγησαν σε αυτήν την απλή και bug free υλοποίηση. Παρακάτω ακολουθεί ένα παράδειγμα σύνθετου animator χαρακτήρα από παιχνίδι που αναπτύχθηκε από εταιρεία για σύγκριση της πολυπλοκότητας.



Εικόνα 5.1.9 : Σύνθετο παράδειγμα set κινήσεων

Δημιουργία controller :

Για την ορθή κίνηση του χαρακτήρα όμως δεν αρκεί απλά η εισαγωγή των animations και του animator αλλά και η δημιουργία ενός controller που θα είναι υπεύθυνος τόσο για τα physics όσο και για τις εναλλαγές μεταξύ των διάφορων κινήσεων που δημιουργήθηκαν στο παράθυρο του animator. Όπως και με πολλές άλλες προαναφερθείσες λειτουργίες το controller για το μοντέλο του χαρακτήρα είναι ένα script γραμμένο σε C#.

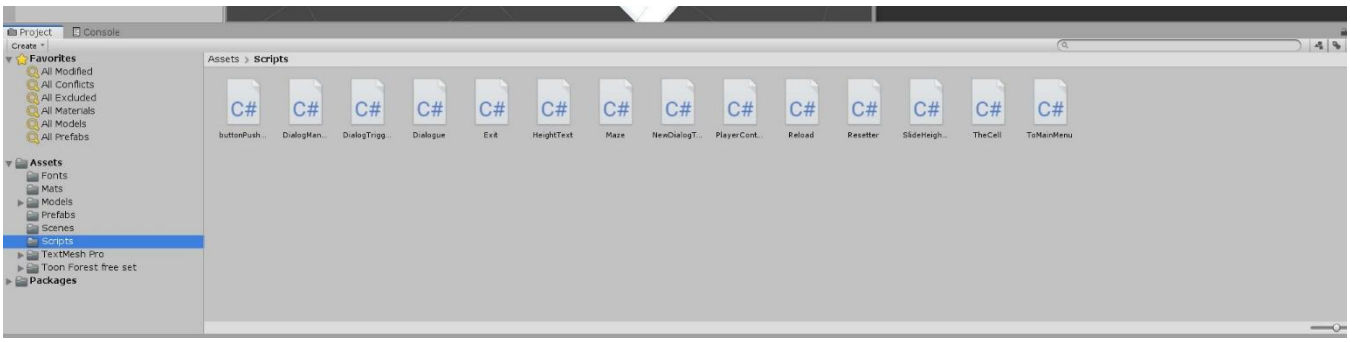


Εικόνα 5.1.10 : Το script του controller του χαρακτήρα

5.2 Scripts

Για την δημιουργία του λαβυρίνθου, τη δημιουργία όλων των χαρακτηριστικών της κίνησης και της συμπεριφοράς του χαρακτήρα, τη λειτουργία των κουμπιών και των διαλόγων και την πλοήγηση ανάμεσα στις 2 οθόνες χρησιμοποιήθηκαν σύντομα και εύχρηστα σενάρια (scripts). Τα σενάρια είναι αρχεία κατάληξης .cs, τα οποία ο προγραμματιστής επεξεργάζεται μέσω Visual Studio ή editor της επιλογής του, διαθέτουν εγγενή υποστήριξη λειτουργιών στο περιβάλλον της Unity και είναι βασικό συστατικό για τη δημιουργία μιας ολοκληρωμένης λύσης παιχνιδιού. Το scripting μπορεί να χαρακτηριστεί ως βασικό συστατικό σε όλες τις εφαρμογές που αναπτύσσονται σε Unity. Οι περισσότερες, αν όχι όλες οι εφαρμογές εκεί, χρειάζονται σενάρια σε τομείς όπως :

- ανταπόκριση στο input του παίκτη
- και να κανονίσει τα συμβάντα στο gameplay να συμβαίνουν όταν πρέπει.
- Πέρα από αυτό, τα σενάρια μπορούν να χρησιμοποιηθούν για τη δημιουργία γραφικών αποτελεσμάτων,
- τον έλεγχο της φυσικής συμπεριφοράς αντικειμένων
- ή ακόμα και την εφαρμογή ενός προσαρμοσμένου συστήματος AI για χαρακτήρες στο παιχνίδι.



Εικόνα 5.1.11 : Η συλλογή των scripts στον inspector

6. Συμπεράσματα και μελλοντικές επεκτάσεις

Μια από τις μελλοντικές επεκτάσεις του παιχνιδιού πρόκειται να είναι η εναλλαγή χρήσης αλγόριθμου για τη δημιουργία του λαβυρίνθου κατά το runtime. Μέσω ενός UI element και συγκεκριμένα ενός switch , ο χρήστης θα είναι σε θέση να επιλέγει με ποιόν αλγόριθμο θα δημιουργείται ο λαβύρινθος. Έτσι θα είναι εμφανές ποιος αλγόριθμος λειτουργεί οπτικά καλύτερα ή το script του καθενός θα είναι έτσι διαμορφωμένο ώστε οι τύποι λαβυρίνθων που παράγονται να έχουν διαφορετικά χαρακτηριστικά. Εδώ θα πρέπει να σημειωθεί ότι λόγω της μεγάλης παραμετροποίησης που ήδη υπάρχει στα οπτικά χαρακτηριστικά του λαβύρινθου, αυτό με την μελλοντική επέκταση μπορεί να αναβαθμιστεί ακόμα περισσότερο και ο παίκτης να είναι σε θέση κατά τη δημιουργία του λαβυρίνθου να επιλέξει πολλά και διαφορετικά attributes όπως χρώμα, σχήμα , μέγιστο αριθμό τοίχων και έκταση λαβυρίνθου.

Ταυτόχρονα θα δίνεται η επιλογή στον παίκτη να επιλέγει ανάμεσα από διαφορετικά μοντέλα χαρακτήρα για την πλοήγηση στον λαβύρινθο. Συγκεκριμένα , όχι μόνο θα παρέχονται έτοιμα προ εγκατεστημένα μοντέλα τα οποία θα είναι σε θέση να εναλλάσσονται κατά βούληση επίσης με τη χρήση UI κουμπιών αλλά θα υπάρχει και η επιλογή ο παίκτης να είναι σε θέση να δημιουργεί το δικό του χαρακτήρα με επιλογές για τα χαρακτηριστικά του προσώπου και του σώματος του. Αυτή η δυνατότητα αρχικά θα υπάρχει με την μορφή προεγκατεστημένων μοντέλων σε αντίστοιχο UI pane επιλογής χαρακτήρα. Αντίστοιχα η δημιουργία custom χαρακτήρα θα γίνεται μέσω πατήματος κουμπιού που θα οδηγεί τον παίκτη σε νέα οθόνη για τον συγκεκριμένο σκοπό.

Τέλος ως μελλοντική επέκταση στο παιχνίδι προβλέπεται το τυχαίο spawn αντικειμένων στον λαβύρινθο και ο στόχος για τον χαρακτήρα να τα μαζέψει όλα ώστε να είναι σε θέση να προβεί στο επόμενο επίπεδο. Τόσο η τυχαία θέση όσο και ο περιορισμός στην εξέλιξη των επιπέδων είναι κάτι σχετικά απλό ως υλοποίηση καθώς το τυχαίο στοιχείο είναι accessible μέσω random number generator κάτι που έχει αντίστοιχα απλή υλοποίηση σε script , και ο περιορισμός υλοποιείται επίσης με απλό τρόπο μέσω ελέγχου συνθηκών στον κώδικα.

6.1 Το asset store tool «amaze»

Κατά την έρευνα στον τομέα για αναζήτηση τόσο παρόμοιων προσεγγίσεων όσο και διαφορετικών αλγορίθμων , βρέθηκε και το έτοιμο εργαλείο παραγωγής λαβυρίνθων «amaze». Η συγκεκριμένη υλοποίηση επιλέγει τον αλγόριθμο recursive backtrack για να είναι σε θέση να παράγει τέλειους λαβυρίνθους. Το συγκεκριμένο tool είναι διαθέσιμο στο asset store έναντι ενός μικρού αντίτιμου. Αυτό αποτελεί ένα καλό παράδειγμα ότι εάν κάποιος προγραμματιστής καταφέρει να είναι ο πρώτος που παράγει ένα χρήσιμο εργαλείο μπορεί να έχει κέρδος απο την έκδοση του.

Συμπέρασμα αποτελεί επίσης ότι υπάρχει δυνατή συνεργασία μεταξύ ανάπτυξης αλγορίθμων στη C# κατά την ανάπτυξη παιχνιδιών στη Unity. Επιπλέον δεν περιορίζεται σε αλγορίθμους για λαβυρίνθους αλλά ουσιαστικά σε οτιδήποτε μπορεί να αποκτήσει γραφική αναπαράσταση ενώ δημιουργείται στο παρασκήνιο μέσω μιας αλγοριθμικής διαδικασίας.

7. Βιβλιογραφία

7.1 Ιστότοποι

- Financial Times The puzzling difference between mazes and labyrinths : https://www.ft.com/content/fd3175fa-ff41-11e6-8d8e-a5e3738f9ae4

- What's the difference between a maze and a labyrinth? : http://blog.english-heritage.org.uk/whats-the-difference-between-a-maze-and-a-labyrinth/

  https://www.quora.com/What-is-the-difference-between-a-maze-and-a-labyrinth

- Maze generation algorithms : https://en.wikipedia.org/wiki/Maze_generation_algorithm

- Hunt and Kill Algorithm : http://people.cs.ksu.edu/~ashley78/wiki.ashleycoleman.me/index.php/Hunt_and_Kill_Algorithm.html

- Mazes : http://datagenetics.com/blog/november22015/index.html

7.2 Βιβλία

- Mazes for Programmers: Code Your Own Twisty Little Passages 1st Edition – Jamis Buck, ISBN-13: 978-1680500554

- Algorithmic Puzzles 1st Edition – Anany Levitin, ISBN-13: 978-0199740444

- Gardens of Imagination/Programming 3d Maze Games in C/C++/Book – Christopher Lampton , ISBN-13: 978-1878739599

8. Αναφορές

- R1 Kern, Through the Labyrinth, 2000, item 43, p. 53.
- R2 Robert Paul Sutton, Communal Utopias and the American Experience: Religious Communities (2003) p. 38
- R3 Adrian Fisher & Georg Gerster, The Art of the Maze, Weidenfeld & Nicolson (1990), ISBN 0-297-83027-9
- R4 Herodotus ,The Histories, translated by Aubrey de Sélincourt, Book II, pp. 160–61.
- R5 Las independencias iberoamericanas en su laberinto: Controversias, cuestiones, interpretaciones - Manuel Chust Calero Universitat de València, ISBN 9788437083193

9. Παραρτήματα

Εδώ θα μπουν : τα προβλήματα που αντιμετωπίσθηκαν (κίνηση χαρακτήρα) , τα UI elements που υπήρχαν για διαφορετικό σκοπό (height/ width) και που αντικαταστάθηκαν από τα κουμπιά info, dialog και main menu και θα επεξηγηθεί για ποιο λόγο ο παραγόμενος λαβύρινθος είναι στον αέρα.

Χαρακτήρας και κίνηση :

Κατά τη δημιουργία του controller της κίνησης του χαρακτήρα προέκυψαν τα εξής ζητήματα : ο χαρακτήρας ενώ αρχικά εκτελούσε κανονικά την κίνηση και άλλαζε θέση στο χώρο , στη συνέχεια κατά τη διακοπή του animation του περπατήματος αυτόματα επανερχόταν στην αρχική θέση από όπου ξεκίνησε το πρώτο animation. Αυτό είχε ως αποτέλεσμα ολόκληρη η κίνηση όχι μόνο να μην εκτελείται σωστά αλλά ουσιαστικά να αποτελεί bug. Ο λόγος που συνέβαινε αυτό ήταν ένας συνδυασμός από συγκεκριμένες ρυθμίσεις στον inspector των animations αλλά και μια όχι αποδεκτή προσέγγιση στον τρόπο που είχε γραφτεί το controller script. Για τη δημιουργία τόσο της αλληλουχίας κινήσεων στο animation pane όσο και του controller που ελέγχει την κίνηση του χαρακτήρα αρχικά είχε επιλεχθεί μια υλοποίηση η οποία ναι μεν είναι καθολικά αποδεκτή στον τρόπο προσέγγισης , όμως όπως φάνηκε κατά την δοκιμή της αντιστοιχούσε τόσο σε παλαιότερες εκδόσεις της Unity , όσο και σε διαφορετικά μοντέλα. Σημειώνεται ότι η εφαρμογή ενός λειτουργικού movement controller από C# script δεν είναι απλή διαδικασία και είναι άμεσα συνδεδεμένη τόσο με το origin όσο και με τον τύπο του μοντέλου στο οποίο εφαρμόζεται. Συνεπώς μετά από αρκετές προσπάθειες και δοκιμές δημιουργήθηκε συνδυασμός script και animation pane που μεταφράστηκε στην σωστή κίνηση του χαρακτήρα για την πλοήγηση του μέσα στον λαβύρινθο.

Deprecated (removed) UI Elements :

Τα αρχικά UI elements στο πλαϊνό πολύχρωμο pane του λαβυρίνθου και τα οποία στην τελική (2.1) έκδοση του παιχνιδιού δεν υπάρχουν ήταν τα εξής :

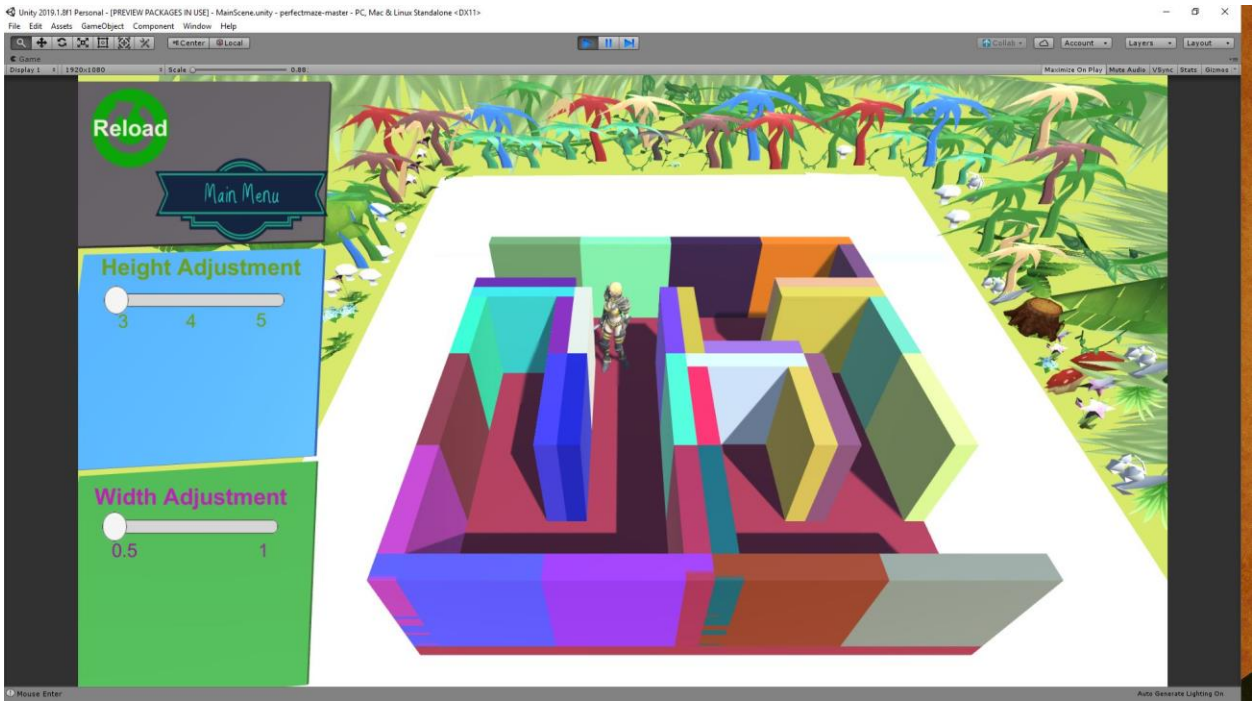
1. Height adjustment pane
2. Width adjustment pane

Το height adjustment pane, όπως φαίνεται και στην εικόνα 9.1 , περιείχε μια μπάρα κύλισης με scale για την ρύθμιση των τιμών του ύψους των τοίχων του λαβυρίνθου οι οποίες μετά την επιλογή εφαρμόζονται κατά την επαναφόρτιση του λαβυρίνθου.

Με τον ίδιο ακριβώς τρόπο λειτουργούσε και το width adjustment pane που βρίσκεται ακριβώς από κάτω από το height adjustment pane. Αντίστοιχα διαθέτει (διαφορετικών τιμών scales) και με αυτό ο παίκτης μπορούσε να ρυθμίσει το πάχος των τοίχων. Σημειώνεται ότι στις μεγαλύτερες τιμές η πλοήγηση του χαρακτήρα στον λαβύρινθο γινόταν σχεδόν αδύνατη λόγω του περιορισμού του διαθέσιμου χώρου για πλοήγηση και ασυμβατότητας με των ήδη προσαρμοσμένων colliders στο μοντέλο του χαρακτήρα.

Ως collider ορίζεται το περίγραμμα του χαρακτήρα που ευθύνεται για τη σύγκρουση του χαρακτήρα με τα φυσικά εμπόδια (στην προκειμένη περίπτωση τοίχους). Σημειώνεται ότι στην περίπτωση που τα παραπάνω colliders ήταν μικρότερα , ο χαρακτήρας θα μπορούσε να κινείται σε περιορισμένο χώρο, με επίδραση όμως στην οπτική εμπειρία του παίκτη και εάν ήταν ανύπαρκτα, ο χαρακτήρας δεν θα μπορούσε να σταθεί στον λαβύρινθο λόγω της βαρύτητας.

Τα παραπάνω panes έχουν αφαιρεθεί από την τελική έκδοση του παιχνιδιού καθώς αποτελούσαν features που είχαν ζητηθεί ως task για project σε εργασιακό χώρο. Λόγω του ότι υπάρχουν πολλά πιο ενδιαφέροντα features για εισαγωγή , επιλέχθηκε το height adjustment pane να αντικατασταθεί με το κουμπί που ενεργοποιεί την έναρξη του διαλόγου του συστήματος με τον παίκτη και αντίστοιχα αντί για width pane εισάχθηκε κουμπί που κατά το πάτημα οδηγεί πίσω στην αρχική οθόνη του παιχνιδιού.



Εικόνα 9.1 Η αρχική μορφή των panes

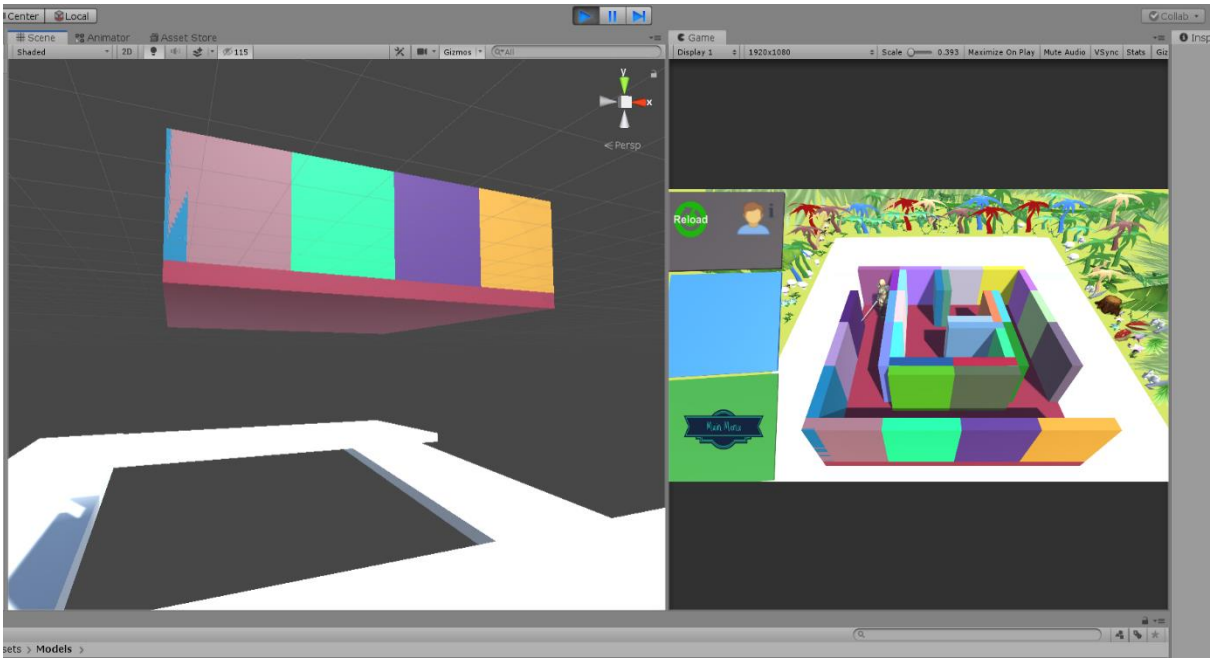
Flying maze (Γιατί είναι στον αέρα ο λαβύρινθος)?

Εύλογη απορία του παίκτη και οποιουδήποτε παρατηρητή του παιχνιδιού (πάντοτε μέσα από τον editor της Unity και όχι στο τελικό προϊόν) αποτελεί ο λόγος που ο λαβύρινθος βρίσκεται στον αέρα. Συμβαίνει κυρίως για τους εξής λόγους :

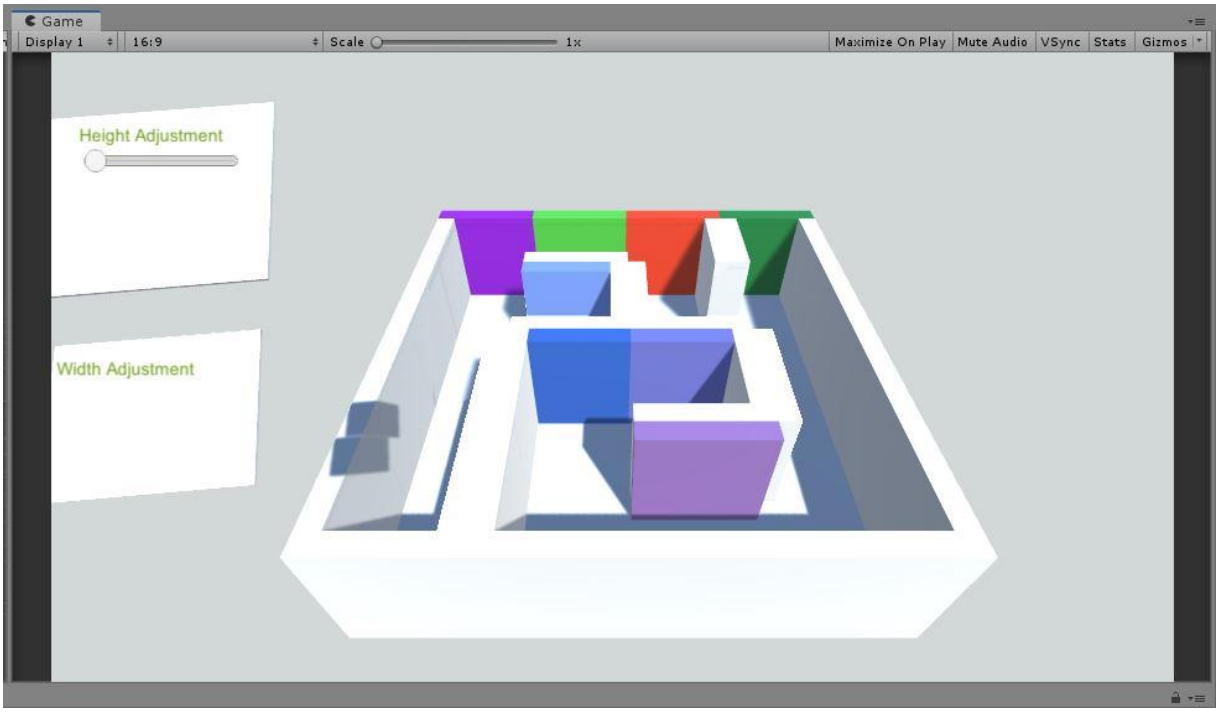
1. Το object από το οποίο προκύπτει ολόκληρος ο λαβύρινθος δεν εμφανίζεται παρα μόνο κατά το runtime καθώς δημιουργείται δυναμικά μέσω C# script. Συνεπώς δεν έχει βρεθεί ακόμα λύση για την μετακίνηση του.

2. Τα υπόλοιπα assets και features του παιχνιδιού εισάχθηκαν αφού δημιουργήθηκε ο τρόπος κατασκευής του λαβυρίνθου

3. Υπάρχουν αόρατα colliders που αιωρούνται κάτω από τον λαβύρινθο και ευθύνονται για την επανεκκίνηση του παιχνιδιού κατά την κρούση.



Εικόνα 9.2 Το παράδοξο του ιπτάμενου λαβυρίνθου



Εικόνα 9.3 Προεπισκόπηση ενός πολύ πρώιμου σταδίου ανάπτυξης του παιχνιδιού