Πανεπιστήμιο Δυτικής Αττικής
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

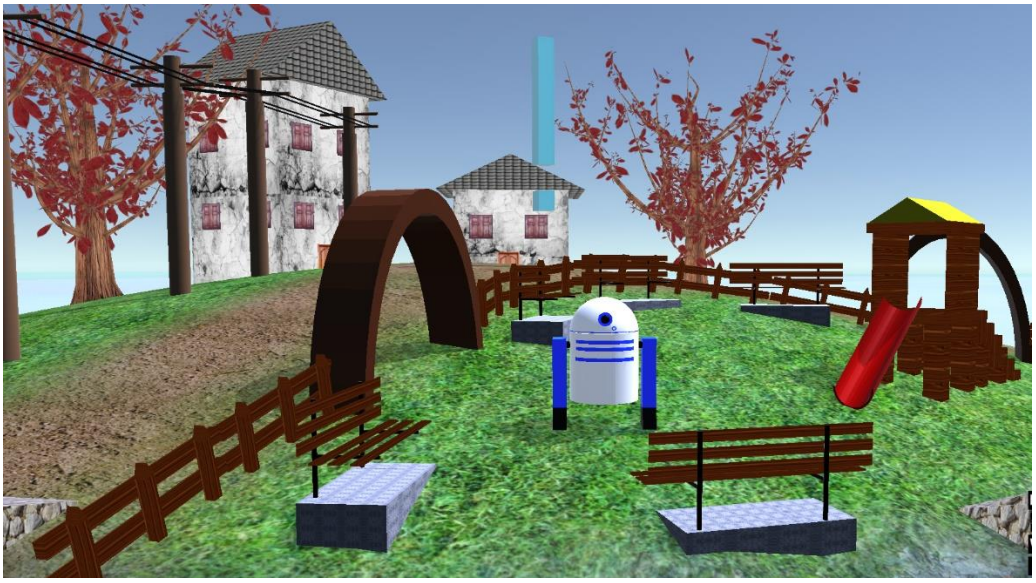
Διπλωματική Εργασία

Σχεδιασμός και υλοποίηση σοβαρού παιχνιδιού (serious game) με χρήση της πλατφόρμας ανάπτυξης παιχνιδιών Unity.

Σκουμπουρδής Πέτρος

ΑΜ: 711121026

Επιβλέπων καθηγητής: Μπαρδής Γεώργιος



Αθήνα 2022

Διπλωματική Εργασία

Σχεδιασμός και υλοποίηση σοβαρού παιχνιδιού (serious game)
με χρήση της πλατφόρμας ανάπτυξης παιχνιδιών Unity.

Σκουμπουρδής Πέτρος

ΑΜ: 711121026

Επιβλέπων καθηγητής: Γεώργιος Μπαρδής

Εξεταστική επιτροπή

Α/Α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ/ΤΜΗΜΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1.	Γεώργιος Μπαρδής	Επίκουρος Καθηγητής	Georgios Bardis Digitally signed by Georgios Bardis Date: 2022.10.26 13:12:30 +03'00'
2.	Χρήστος Τρούσσας	Επίκουρος Καθηγητής	Christos Troussas Digitally signed by Christos Troussas Date: 2022.10.26 19:35:11 +03'00'
3.	Παναγιώτα Τσελέντη	Ε.ΔΙ.Π.	Panagiota Tselenti Digitally signed by Panagiota Tselenti Date: 2022.10.27 09:06:53 +03'00'

Ημερομηνία εξέτασης: 13 / 10 /2022

Δήλωση συγγραφέα

Ο κάτωθι υπογεγραμμένος Σκουμπουρδής Πέτρος του Σοφοκλή, με αριθμό μητρώου 711121026 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

Σκουμπουρδής Πέτρος

(Υπογραφή)

Περιεχόμενα

Πίνακας Εικόνων

Περίληψη

Στην παρούσα εργασία θα περιγράψουμε πως μπορούμε να χρησιμοποιήσουμε την πλατφόρμα Unity για να δημιουργήσουμε ένα σοβαρό παιχνίδι.

Η πλατφόρμα Unity είναι ένα αρκετά δυνατό λογισμικό που επιτρέπει σε καλλιτέχνες, σχεδιαστές και προγραμματιστές να δημιουργήσουν εύκολα απίθανες διαδραστικές εφαρμογές και παιχνίδια. Διαθέτει πλούσιες ηλεκτρονικές βιβλιοθήκες με σεμινάρια (tutorials) και αντικείμενα (assets) που στοχεύουν στο να καθοδηγήσουν και να βοηθήσουν τον σχεδιαστή να δημιουργήσει γρηγορότερα και ευκολότερα την εφαρμογή του. Το Unity διατίθεται για λογισμικό Windows, Mac και Linux και είναι ως επί των πλείστων δωρεάν για εφαρμογές που δεν προορίζονται για εμπορικούς σκοπούς.

Το παιχνίδι που δημιουργήσαμε απευθύνεται σε παιδιά και έχει εκπαιδευτικό περιεχόμενο (serious game). Αποτελείται από 2 κόσμους. Στον πρώτο κόσμο ο στόχος είναι το παιδί να μάθει μέσα από διαδραστικές ερωτήσεις τα μέτρα προστασίας που πρέπει να πάρει κάποιος σε περίπτωση σεισμού. Στον δεύτερο κόσμο ο στόχος είναι το παιδί να εμπλουτίσει τις γνώσεις τους σχετικά με το πως μπορεί κάποιος να επιβιώσει στα χιόνια. Το παιδί καλείται να ολοκληρώσει τον κόσμο αυτό εκτελώντας ορισμένες διαδραστικές ενέργειες, ακολουθώντας οδηγίες και πληροφορίες που του παρέχονται.

Το θεματικό περιεχόμενο του παιχνιδιού δεν επιλέχτηκε τυχαία, αλλά με άξονα την διερεύνηση ορισμένων πτυχών και λειτουργιών της πλατφόρμας Unity.

Εισαγωγή

Στο παρελθόν το να παίζει κάποιο άτομο βιντεοπαιχνίδια ήταν ταμπού. Ο μέσος όρος ηλικίας των ατόμων που παίζουν, είναι 33 ετών βάσει στατιστικών στοιχείων του ιστοτόπου της Entertainment Software Association (E.S.A.). Στατιστικές που προέρχονται από τον ίδιο ιστότοπο, δείχνουν ότι τα βιντεοπαιχνίδια προσφέρουν κυρίως διασκέδαση (93%), νοητική διέγερση (91%) και ανακούφιση από το άγχος (89%). Είναι άξιο να σημειωθεί το πόσο υψηλό είναι το ποσοστό που αφορά την νοητική διέγερση. [1]

Στις μέρες μας, τα βιντεοπαιχνίδια, γίνονται όλο και πιο δημοφιλή. Βλέπουμε να γίνονται θέμα συζήτησης στα μέσα κοινωνικής δικτύωσης, στην τηλεόραση, στις ταινίες και στη μουσική. Εντός της πανδημίας COVID-19 και της υποχρεωτικής καραντίνας, το έτος 2020-2021, η επικοινωνία, η εργασία και η εκπαίδευση διεξάγονταν μόνο ηλεκτρονικά. Ο κόσμος στράφηκε ακόμη περισσότερο στα βιντεοπαιχνίδια ως μέσο ψυχαγωγίας και επικοινωνίας με φίλους και οικογένεια. Εμφανίστηκε έτσι η ανάγκη για δημιουργία νέων εφαρμογών που εξυπηρετούν την τηλεκπαίδευση και την τηλεπικοινωνία. Τέτοιες εφαρμογές είναι και τα εκπαιδευτικά βιντεοπαιχνίδια.

Τα εκπαιδευτικά βιντεοπαιχνίδια είναι πλέον αρκετά ικανά ώστε να εκπαιδεύσουν σωστά τον παίκτη/μαθητή. Μπορούν να κατασκευαστούν με τρόπο τέτοιο ώστε να απευθύνονται σε συγκεκριμένη ηλικιακή ομάδα. Κάποια έχουν τη δυνατότητα να διαμορφώνεται το επίπεδο δυσκολίας τους ώστε ο παίκτης να εξασκείται και να αποκτά μεγαλύτερη εμπειρία στις γνώσεις του. Κάποια άλλα βιντεοπαιχνίδια εκμάθησης μπορεί να επιτρέπουν και την ταυτόχρονη αλληλεπίδραση του παίκτη/μαθητή με το ψηφιακό περιβάλλον (δια δραστικά βιντεοπαιχνίδια). Μάλιστα, στις Η.Π.Α., έχουν εντοπιστεί ορισμένοι τρόποι εκμάθησης, εντός εκπαιδευτικών βιντεοπαιχνιδιών, που θα μπορούσαν να ενσωματωθούν ως τεχνικές στο εκπαιδευτικό σύστημα [2].

Από μικρός μου αρέσαν πολύ τα βιντεοπαιχνίδια. Οι συνδυασμοί χρωμάτων, τα σχέδια, το πως ο παίκτης ο οποίος είναι απλά ένα ψηφιακό αντικείμενο μπορεί να κινείται και να αλληλοεπιδρά με άλλα αντικείμενα στον χώρο. Πάντα αναρωτιόμουν πως μπορεί να δημιουργηθεί ένα τέτοιο αποτέλεσμα. Η εργασία φαινόταν να είναι πολύ επίπονη και χρονοβόρα.

Μεγαλώνοντας, ακολούθησα την επιστήμη των υπολογιστών και του προγραμματισμού. Από τις σπουδές μου, κατάφερα να κατανοήσω πρώτα τη θεωρητική λειτουργία μιας εφαρμογής και στην συνέχεια πως να πραγματοποιήσω μια τέτοια εφαρμογή. Με την ραγδαία εξέλιξη της τεχνολογίας ανακάλυψα ότι ήδη υπάρχουν πανίσχυρα λογισμικά στα οποία μπορεί να βασιστεί κάποιος για να φτιάξει μια εφαρμογή. Τέτοιου είδους λογισμικά αποσκοπούν στο να εξοικονομήσουν χρόνο και κόπο στον χρήστη παρέχοντάς του πλήθος ηλεκτρονικών βιβλιοθηκών και βοηθητικών εργαλείων. Διάσημα λογισμικά που βοηθούν στη δημιουργία ενός βιντεοπαιχνιδιού είναι το Unity, το Unreal Engine, CryEngine, κ.α.

Για την εκπόνηση της εργασίας αυτής, επέλεξα να φτιάξω το δικό μου μικρό εκπαιδευτικό βιντεοπαιχνίδι στην πλατφόρμα Unity, με σκοπό να εξερευνήσω τις δυνατότητες και τις λειτουργείες της πλατφόρμας. Η επιλογή της πλατφόρμας Unity έμοιαζε η καλύτερη γιατί παρέχεται δωρεάν για προσωπική χρήση, θεωρείται η πιο εύκολη για άτομα που μόλις ξεκίνησαν να ασχολούνται με την ανάπτυξη βιντεοπαιχνιδιών, είναι πιο βατή για προγραμματιστές και τέλος υπάρχουν πολλές οδηγίες και μέσα καθοδήγησης (βίντεο, forums, κ.λπ.) για τη δημιουργία διάφορων πραγμάτων (scripts, 3D Model Editing, κ.λπ.).

Μέσα από τη μηχανή Unity και του συγκεκριμένου βιντεοπαιχνιδιού, σκοπός είναι η κατανόηση:

- Της κίνησης του βασικού παίχτη.
- Της 3D μοντελοποίησης αντικειμένων.
- Του τρόπου δημιουργίας του περιβάλλοντος κάθε πίστας.
- Των εφέ (φωτιά, καπνός, σύννεφα, σπίθες, νερό – θάλασσα).
- Αλληλεπίδραση του παίκτη με αντικείμενα και καταστάσεις.
- Της μετάβασης του παίκτη από πίστα σε πίστα.
- Της εμφάνισης κατάλληλων μηνυμάτων στην οθόνη.
- Της πλοήγησης του χρήστη στην εφαρμογή (αλληλεπίδραση με το ποντίκι και το πληκτρολόγιο).
- Της πλοήγησης του χρήστη εντός της εφαρμογής, με χρήση παραθύρων τύπου μενού.

1. Πρώτο κεφάλαιο

1.1 Τι είναι ένα παιχνίδι;

Παιχνίδι, αποκαλείται μία δομημένη δραστηριότητα που διεξάγεται με σκοπό την ψυχαγωγία ή την εκπαίδευση. Τα παιχνίδια υπάρχουν σε διάφορες μορφές όπως επιτραπέζια, ηλεκτρονικά, άυλα, υλικά κ.ά. Κύριοι παράγοντες που αποτελούν ένα παιχνίδι είναι οι κανόνες, τα επιτεύγματα, η πρόκληση και η αλληλεπίδραση. Τα παιχνίδια, γενικά, υποβάλλουν τον παίκτη σε μια ψυχική ή σωματική διέγερση, ή πολλές φορές και τα δύο. Έτσι βοηθούν στην ανάπτυξη πρακτικών δεξιοτήτων. Επίσης προσφέρουν ψυχαγωγία, προωθούν τη φιλία, τη συνεργατικότητα και την επικοινωνία με άλλους ανθρώπους.

Γενικές κατηγορίες παιχνιδιών είναι:

- Αθλήματα
- Παιχνίδια δωματίου (επιτραπέζια, τράπουλα, κ.λπ.)
- Τυχερά παιχνίδια
- Βιντεοπαιχνίδια
- Κ.α. [3]

1.2 Τι είναι ένα εκπαιδευτικό παιχνίδι;

Ένα **εκπαιδευτικό παιχνίδι** είναι ένα παιχνίδι που έχει σχεδιαστεί για να βοηθήσει στην εκμάθηση συγκεκριμένων θεμάτων, όπως για παράδειγμα την κατανόηση ενός ιστορικού γεγονότος ή μιας κουλτούρας, την καλλιέργεια και εξάσκηση μίας δεξιότητας καθώς το άτομο παίζει, επίλυση προβλημάτων φυσικής, μαθηματικών, κ.α.

Χρησιμοποιούνται σαν διαδραστικό εργαλείο στην εκπαίδευση. Παρουσιάζουν στον παίκτη/μαθητή μία ιστορία ή σενάριο, μέσα από την οποία μαθαίνει να προσαρμόζεται σε διάφορα περιβάλλοντα, να ακολουθεί κανόνες, να πετυχαίνει στόχους, να επιλύει προβλήματα, να συνεργάζεται με άλλα άτομα όταν το παιχνίδι το απαιτεί, κ.α. Ταυτόχρονα ικανοποιούνται η ανάγκη για ψυχαγωγία, η ανάγκη για ενεργή συμμέτοχή για κάποιο σκοπό, η ικανοποίηση του «εγώ» με την επίτευξη ενός στόχου, η διέγερση της αδρεναλίνης και της δημιουργικότητας και το ομαδικό πνεύμα. [4]

1.3 Τι είναι ένα βιντεοπαιχνίδι;

Ένα **βιντεοπαιχνίδι** είναι ένα ηλεκτρονικό παιχνίδι που μπορεί να παιχτεί μέσα από μια συσκευή προβολής όπως κινητό, ηλεκτρονικός υπολογιστής και τηλεόραση ή οθόνη. Η αλληλεπίδραση του χρήστη με τη συσκευή προβολής/παιχνίδι πραγματοποιείται μέσω ειδικών διεπαφών, όπως ένα χειριστήριο, ένα πληκτρολόγιο, ένα ποντίκι, έναν αισθητήρα κίνησης κ.λπ.

Τα βιντεοπαιχνίδια μπορούν να διακριθούν σε πολλές κατηγορίες. Κατηγορίες βιντεοπαιχνιδιών με βάσει την **πλατφόρμα** τους:

- Βιντεοπαιχνίδια για Ηλεκτρονικό Υπολογιστή
- Βιντεοπαιχνίδια για κονσόλα στο σπίτι (PlayStation, Xbox, Nintendo Switch)
- Βιντεοπαιχνίδια για κονσόλα τσέπης (Gameboy, PSP, Nintendo Switch)
- Βιντεοπαιχνίδια Arcade (συσκευές εγκατεστημένες σε καταστήματα)
- Βιντεοπαιχνίδια στον Περιηγητή – Browser (Google Chrome, Mozilla Firefox, κ.α.)
- Βιντεοπαιχνίδια για Κινητό τηλέφωνο (Smartphone)
- Βιντεοπαιχνίδια στο Cloud
- Βιντεοπαιχνίδια Εικονικής Πραγματικότητας [2]

Οι πιο διάσημες κατηγορίες βιντεοπαιχνιδιών με βάσει το **είδος** τους:

- Δράσης
- Περιπέτειας
- Μάχης
- Γρίφος
- Αγωνιστικά
- Παιχνίδια Ρόλων
- Όπλων
- Προσομοίωσης
- Αθλητικά
- Στρατηγικής
- Και άλλα. [5]

Κατηγορίες βιντεοπαιχνιδιών με βάσει το **σκοπό** τους:

- Κλασσικά παιχνίδια
- Εκπαιδευτικά παιχνίδια
- Σοβαρά παιχνίδια (Serious Games)
- Δημιουργικά. [2]

1.4 Τι είναι ένα Εκπαιδευτικό Σοβαρό Βιντεοπαιχνίδι (Serious Game);

Σοβαρό Βιντεοπαιχνίδι ή αλλιώς **Serious Game** είναι μια υπερκατηγορία σοβαρών βιντεοπαιχνιδιών που έχουν απώτερο σκοπό να εκπαιδεύσουν. Χρησιμοποιούνται συχνά από βιομηχανίες άμυνας, εκπαίδευσης, επιστημονικών ερευνών, υγείας, εκτάκτων καταστάσεων, πολιτικών θεμάτων, σχεδιασμού πόλεων και στη μηχανολογία. [6]

Συνεπώς, τα εκπαιδευτικά παιχνίδια είναι μια υποκατηγορία που ανήκει στα σοβαρά βιντεοπαιχνίδια και χρησιμοποιούνται στον τομέα της εκπαίδευσης. Ένα εκπαιδευτικό βιντεοπαιχνίδι παρέχει γνώση στον παίκτη – μαθητή, τα μέσα για να την εξασκήσει και ταυτόχρονα τον ψυχαγωγεί. Δεν είναι αυστηρά απαραίτητο να εμπεριέχεται σε κάποιο πρόγραμμα σπουδών ή να υπάρχει επιβλέπων καθηγητής. Παρόλα αυτά, πολλοί καθηγητές, έχουν αναγνωρίσει τη χρησιμότητά τους και προτιμούν να τα χρησιμοποιούν περιστασιακά στο μάθημά τους. Μάλιστα, πολλές πρόσφατες μελέτες έδειξαν ότι τα βιντεοπαιχνίδια, ανεξαρτήτως είδους, ενισχύουν τα παιδιά στις διανοητικές και στις συναισθηματικές δεξιότητές τους, που σχετίζονται με την ακαδημαϊκή τους καριέρα [7]. Άλλες δεξιότητες που ενισχύουν είναι:

- Η γνωσιακή μάθηση
- Οι τρόποι επίλυσης προβλημάτων
- Ο συντονισμός ματιών και χεριών και γενικά δεξιότητες χειρισμού
- Η ταυτόχρονη εκτέλεση πολλών εργασιών
- Η ταχύτερη και πιο σωστή λήψη αποφάσεων.
- Η υψηλή συγκέντρωση
- Η δημιουργική σκέψη [8]

Παραδείγματα εκπαιδευτικών βιντεοπαιχνιδιών είναι:

- *SimCity*. Ο παίκτης καλείται να εξερευνήσει τις κοινωνικές, πρακτικές και οικονομικές διαδικασίες για την κατασκευή μιας πόλης.
- *Civilization*. Ο παίκτης μαθαίνει για την ιστορία, τις πολιτικές, τις οικονομικές και τις στρατιωτικές πτυχές της.
- *PlaceSpotting*. Ο παίκτης μαθαίνει γεωγραφία ψάχνοντας πάνω στον παγκόσμιο χάρτη τις περιοχές που του ζητούνται.
- *Quantum Moves*. Ένα παιχνίδι φυσικής, που προσπαθεί να δώσει μία αίσθηση στον παίκτη για περίπλοκες έννοιες φυσικής, κβαντικής μηχανικής και ειδικής σχετικότητας. [9]

1. 5 Ιστορική Αναδρομή

1.5.1 Ιστορική Αναδρομή βιντεοπαιχνιδιών

Ο πρώτος υπολογιστής κατασκευάστηκε το 1822 από τον J. Presper Eckert και John Mauchly. Του αποδόθηκε το όνομα **ENIAC** και καταλάμβανε χώρο περίπου 168 τετραγωνικά μέτρα. [10]

Μέχρι τότε, υπήρχαν τα **ηλεκτρονικά παιχνίδια**. Αυτά ήταν ογκώδη ηλεκτρονικά συστήματα όπου ο παίκτης μπορούσε να αλληλοεπιδράσει με χειρισμούς που βρίσκονταν ενσωματωμένοι πάνω σε αυτά. Παραδείγματα ηλεκτρονικών παιχνιδιών είναι το φλίπερ, ο κουλοχέρης, μηχανήματα arcade, κ.α. [11]

Περίπου 10 χρόνια αργότερα, στις δεκαετίες του 1950 και 1960, εμφανίστηκαν τα **πρώτα πρωτότυπα βιντεοπαιχνίδια**. Αυτά ήταν απλές επεκτάσεις ηλεκτρονικών παιχνιδιών, δηλαδή υπολογιστές που καταλάμβαναν ολόκληρο δωμάτιο και χρησιμοποιούσαν ως έξοδο, ψηφιακές συσκευές προβολής 2 διαστάσεων, πολύ απλές οθόνες. Ο χρήστης αλληλοεπιδρά με το παιχνίδι μέσω διεπαφών όπως γραφικά, κουμπιά εφαρμογής κ.λπ.

Το πρώτο εμπορικό βιντεοπαιχνίδι δημοσιεύτηκε το 1971, ήταν τύπου arcade και ονομαζόταν *Computer Space*. Το 1972 δημοσιεύτηκε το θρυλικό βιντεοπαιχνίδι *Pong* και η πρώτη κονσόλα για το σπίτι, *Magnavox Odyssey* (Εικόνα 1). Στα τέλη του 1970 και στις αρχές του 1980 έχουμε τη **χρυσή εποχή των arcade βιντεοπαιχνιδιών**, όπου η βιομηχανία τους παρουσίασε ραγδαία ανάπτυξη. Όμως το 1983, η βιομηχανία των arcade κατέρρευσε λόγω της απώλειας του ελέγχου των δημοσιεύσεων τίτλων βιντεοπαιχνιδιών, του κορεσμού της αγοράς στη βιομηχανία των βιντεοπαιχνιδιών στη Βόρεια Αμερική και την εμφάνιση και άνοδο της βιομηχανίας των προσωπικών υπολογιστών [9]. Στη συνέχεια, η βιομηχανία άκμασε ξανά με κυρίαρχους Ιαπωνικές εταιρίες, όπως η Nintendo, η Sega και η Sony, οι οποίες εφαρμόσανε πρακτικές και πολιτικές στην ανάπτυξη και διανομή των βιντεοπαιχνιδιών, ώστε να μην ξανασυμβεί παρόμοια κατάρρευση.

Εικόνα 1 Κονσόλα Magnavox Odyssey 1972. *[12]*

Το 2000 επίκεντρο είναι τα "**AAA**" βιντεοπαιχνίδια. "AAA" είναι τα βιντεοπαιχνίδια που κατασκευάζονται και δημοσιεύονται από πολύ γνωστές μεσαίες και μεγάλες εταιρίες, οι οποίες είναι ικανές να διαθέσουν μεγάλα κεφάλαια [13].

Το 2010, με τη διαθεσιμότητα του Internet και την εμφάνιση νέων τρόπων ψηφιακών πωλήσεων, άνοιξε τον δρόμο για τους ανεξάρτητους δημιουργούς βιντεοπαιχνιδιών, ή αλλιώς **independent video game developers** ή **indie game developers**. Αυτοί ενίσχυσαν ακόμη περισσότερο τη βιομηχανία των βιντεοπαιχνιδιών. Από τότε μέχρι σήμερα η βιομηχανία των βιντεοπαιχνιδιών παρουσιάζει συνεχώς ανάπτυξη. Τον κυρίαρχο ρόλο παίζουν οι Ασιατικές αγορές, αυξάνοντας τα έσοδα εστιάζοντας στο κλασσικά παιχνίδια και προωθώντας τα βιντεοπαιχνίδια ως υπηρεσία. Ενδεικτικά, το 2020, τα ετήσια έσοδα στη βιομηχανία των βιντεοπαιχνιδιών στις Η.Π.Α., ανέρχονται στα 159 δισεκατομμύρια δολάρια, σε υλικό, λογισμικό και υπηρεσίες.

Σήμερα, για να προωθηθεί ένα βιντεοπαιχνίδι στην αγορά, απαιτούνται επιπλέον ενέργειες και ρόλοι όπως προγραμματιστές, εκδότες, διανομείς, καταστήματα λιανικής πώλησης, κατασκευαστές κονσόλων και άλλους τρίτους κατασκευαστές και υπηρεσίες. [2]

Μερικά διάσημα βιντεοπαιχνίδια βάσει του πλήθους πωλήσεων σε αντίγραφα, σήμερα είναι:

- Minecraft, με 238.000.000 πωλήσεις.
- Grand Theft Auto, με 169.000.000 πωλήσεις.
- Tetris (EA), με 100.000.000 πωλήσεις.
- Wii Sports, με 82.900.000 πωλήσεις.
- PUBG: Battlegrounds, με 75.000.000 πωλήσεις.
- Και άλλα. [14]

1.5.2 Ιστορική Αναδρομή εκπαιδευτικών βιντεοπαιχνιδιών

Η χρήση των παιχνιδιών στην εκπαίδευση εμφανίζεται τουλάχιστον την δεκαετία του 1960-1970 και βασίζονταν κυρίως στο χαρτί.

Το πρώτο εκπαιδευτικό βιντεοπαιχνίδι που σχεδιάστηκε για παιδιά δημοτικού, ήταν το *The Sumerian Game*, το 1964. Τα εκπαιδευτικά βιντεοπαιχνίδια έγιναν πιο διάσημα στις αρχές του 1980. Με την κατάρρευση της βιομηχανίας των arcade βιντεοπαιχνιδιών στη Βόρεια Αμερική το 1983 εμφανίστηκε το ηθικό δίλημμα κατά πόσο τα arcade βιντεοπαιχνίδια είναι βίαια και εθιστικά. Έτσι, οι προγραμματιστές εκμεταλλεύτηκαν το γεγονός της ανόδου των προσωπικών υπολογιστών εκείνη την εποχή και εστίασαν στην δημιουργία εκπαιδευτικών βιντεοπαιχνιδιών. Με αυτόν τον τρόπο, ικανοποιούσαν ταυτόχρονα τους γονείς και τους διδάσκοντες.

Την ίδια χρονιά, χρησιμοποιήθηκε ο όρος «**εκπαιδευτική ψυχαγωγία**» ή αλλιώς "**edutainment**" (educational entertainment) για να περιγράψει ένα πακέτο βιντεοπαιχνιδιών φτιαγμένων για τους προσωπικούς υπολογιστές Oric 1[1] και Spectrum Microcomputers[2]. Το πακέτο ήταν διαθέσιμο από την Telford ITEC, ένα εκπαιδευτικό πρόγραμμα χρηματοδοτούμενο από την Κυβέρνηση του Ηνωμένου Βασιλείου. Το 1984 και μετά, εμφανίστηκαν και άλλα εκπαιδευτικά

[1] Το Oric ήταν το όνομα που χρησιμοποίησε η εταιρεία Tangerine Computer Systems με έδρα το Ηνωμένο Βασίλειο, για μια σειρά προσωπικών υπολογιστών που πωλήθηκαν τη δεκαετία του 1980, κυρίως στην Ευρώπη. [48]

[2] Spectrum Microcomputers είναι προσωπικοί μικροϋπολογιστές των 8-bit που κατασκευάστηκαν από την Sinclair Research. Εκδόθηκαν για πρώτη φορά το έτος 1982 στο Ηνωμένο Βασίλειο. [49]

βιντεοπαιχνίδια από άλλες εταιρείες όπως το *Seven Cities of Gold* της *Electronic Arts* (*E.A.*).

Το 1970, ο Clark C. Abt ορίζει ως **σοβαρά βιντεοπαιχνίδια** ή "**serious games**", τα βιντεοπαιχνίδια εκείνα που δίνουν έμφαση, περισσότερο στην εκπαίδευση παρά στην ψυχαγωγία και έχουν έναν σαφή και προσεκτικά μελετημένο εκπαιδευτικό σκοπό. Επίσης τόνισε πως αυτό δεν σημαίνει ότι δεν μπορούν να είναι και ψυχαγωγικά. [6]

Τέλη του 1999 και αρχές της δεκαετίας του 2000 παρατηρήθηκε αύξηση διαφορετικών τύπων εκπαιδευτικών παιχνιδιών, ειδικά εκείνων που προορίζονταν για νεότερους μαθητές. Πολλά από αυτά δεν βασίζονταν σε υπολογιστή, αλλά υιοθετούσαν το μοντέλο παραδοσιακών παιχνιδιών σε κονσόλες για το σπίτι όσο και σε κονσόλες τσέπης. Το 1999, η LeapFrog Enterprises παρουσίασε το LeapPad, το οποίο συνδύαζε ένα διαδραστικό βιβλίο με μια κασέτα και επέτρεπε στα παιδιά να παίζουν παιχνίδια και να αλληλεπιδρούν με το βιβλίο σαν να είναι βασισμένο σε χαρτί. Το 2003, η ίδια εταιρεία την δική της εκπαιδευτική κονσόλα, με όνομα Leapster, η οποία χρησιμοποιούσε κασέτες και συνδύαζε arcade παιχνίδια με εκπαιδευτικό περιεχόμενο. Τα εκπαιδευτικά παιχνίδια επεκτάθηκαν στην *βιώσιμη ανάπτυξη*[3] με τίτλους *Εκπαιδευτική Βιώσιμη Ανάπτυξη* το 2000 και *Κλιματική Πρόκληση* το 2006.

Τα serious games επεκτάθηκαν και πέρα από τον τομέα της εκπαίδευσης με παράδειγμα το παιχνίδι σκόπευσης πρώτου προσώπου, *America's Army* που εκδόθηκε το 2002. Το βιντεοπαιχνίδι αναπτύχθηκε από τον Αμερικανικό στρατό σαν εργαλείο κατάταξης.

Το 2010, τα serious games εξελίχθηκαν σε βαθμό που ενσωμάτωναν την πραγματική οικονομία. Παράδειγμα είναι το Second Life, στο οποίο οι χρήστες μπορούν να δημιουργήσουν πραγματικές επιχειρήσεις που παρέχουν εικονικά εμπορεύματα και υπηρεσίες με δολάρια Linden, τα οποία ανταλλάσσονται με το νόμισμα των ΗΠΑ.

Το 2015, κυκλοφόρησε το *Project Discovery* ως ένα σοβαρό παιχνίδι. Το *Project Discovery* εμπεριείχε ένα όχημα με το οποίο γενετιστές και αστρονόμοι του Πανεπιστημίου της Γενεύης μπορούσαν να έχουν πρόσβαση στα δεδομένα του κοινού που έπαιζε, μέσω ενός μικρού παιχνιδιού, που άνηκε μέσα σε ένα μεγαλύτερο διαδικτυακό βιντεοπαιχνίδι ρόλων για πολλούς παίκτες, το *Eve Online*. Οι παίκτες έπαιρναν ρόλο ως

[3] Η βιώσιμη ή αειφόρος ανάπτυξη αναφέρεται στην οικονομική ανάπτυξη που σχεδιάζεται και υλοποιείται λαμβάνοντας υπόψιν την προστασία του περιβάλλοντος και τη βιωσιμότητα. [50]

πολίτες επιστήμονες και αξιολογούσαν πραγματικά γενετικά δείγματα ή αστρονομικά δεδομένα. Αυτά τα δεδομένα στη συνέχεια μελετιούνταν από τους ερευνητές του Πανεπιστημίου της Γενεύης. [6]

1.5.3 Ιστορική Αναδρομή των Game Engines

Μια **μηχανή παιχνιδιών** γνωστή ως **Game Engine**, είναι ένα πακέτο λογισμικών πάνω στο οποίο αναπτύσσεται το βιντεοπαιχνίδι. Για τον σκοπό αυτό, περιλαμβάνει βιβλιοθήκες και υποστηρικτικά προγράμματα. Ο όρος “Engine”, ή μηχανή, είναι αντίστοιχος του όρου μηχανής λογισμικού.

Εικόνα 2 Κονσόλα ATARI 2600. Είχε μόλις 4 μοχλούς. [15]

Στο παρελθόν, πριν από τις μηχανές παιχνιδιών, τα βιντεοπαιχνίδια αναπτύσσονταν από την αρχή τους ως αυτόνομες οντότητες. Κάθε βιντεοπαιχνίδι, έπρεπε να σχεδιαστεί από την αρχή μέχρι το τέλος του για να δουλέψει σωστά σε μία συγκεκριμένη μόνο κονσόλα (π.χ. για την κονσόλα *Atari 2600*, το 1977, Εικόνα 2). Η διαδικασία για την προβολή του βιντεοπαιχνιδιού στην αντίστοιχη πλατφόρμα ονομάζεται από τους developers, *kernel*, για τα παλαιότερα συστήματα. Όμως, πολλές πλατφόρμες/κονσόλες, μπορεί να μην είχαν πρόβλημα στον τρόπο προβολής, αλλά υστερούσαν σε χωρητικότητα μνήμης για να μπορέσει να φτιαχτεί σε αυτές μία μηχανή παιχνιδιού με πολλά δεδομένα. Ακόμη και σε πιο εύκολες πλατφόρμες, λίγα ήταν τα δεδομένα που μπορούσαν να επαναχρησιμοποιηθούν ανάμεσα στα παιχνίδια.

Την **χρυσή εποχή των arcade βιντεοπαιχνιδιών** το 1980, η ραγδαία ανάπτυξη του νέου υλικού για arcade βιντεοπαιχνίδια σήμανε πως τα μεγαλύτερα κομμάτια κώδικα των μέχρι τότε βιντεοπαιχνιδιών θα αχρηστεύονταν γιατί τα νεότερα βιντεοπαιχνίδια θα έπρεπε να ακολουθήσουν τελείως διαφορετικό σχεδιασμό για να μπορέσουν να χρησιμοποιηθούν στους νέους πόρους υλικού. Ο νέος σχεδιασμός τους ήταν βασισμένος σε πολύ συγκεκριμένο σύνολο κανόνων κώδικα, λίγες πίστες και λίγα δεδομένα γραφικών. Οι εταιρίες βιντεοπαιχνιδιών άρχισαν να κατασκευάζουν τότε δικές τους μηχανές παιχνιδιών χρησιμοποιώντας δικό τους λογισμικό.

Ένα αξιοσημείωτο αποτέλεσμα μιας μηχανής βιντεοπαιχνιδιών σε μία κονσόλα για το σπίτι στα μέσα του 1980, ήταν η ομαλή κίνηση του παίκτη προς τα δεξιά, κατά βούληση, και σχεδιάστηκε από την Nintendo. Με τη βοήθεια αυτής της μηχανής, έφτιαξαν το αγωνιστικό βιντεοπαιχνίδι *Excitebike* (1984) και αργότερα ενσωματώθηκε στο *Super Mario Bros.* (1985) όπου ο παίκτης μπορούσε να επιταχύνει προς τα δεξιά όποτε ήθελε, πέρα από το να κινείται με μια σταθερή ταχύτητα.

Ακόμη, το 1980 είχαν παραχθεί πολλά συστήματα για την δημιουργία βιντεοπαιχνιδιών σε 2 διαστάσεις (2D), χωρίς ο δημιουργός να εξαρτάται από κάποια μηχανή βιντεοπαιχνιδιών κάποιας τρίτης εταιρείας. Πάνω σε αυτά, φτιάχτηκαν τα παιχνίδια *Pinball Construction Set* (1983), *Garry Kitchen's GameMaker* (1985), *Arcade Game Construction Kit* (1988), κ.α.

Το 1990 έγιναν γνωστές οι μηχανές παιχνιδιών για υπολογιστές 3D γραφικών. Ο όρος «game engine» παγιώθηκε στα μέσα του 1990 και συνδεόταν κυρίως με τα 3D βιντεοπαιχνίδια, όπως τα βιντεοπαιχνίδια σκόπευσης πρώτου προσώπου που κατασκευάστηκαν σε δική τους μηχανή. Η θρυλική μηχανή βιντεοπαιχνιδιών *Klik & Play* του 1994 είναι διαθέσιμη μέχρι και σήμερα. Οι πιο διάσημες μηχανές βιντεοπαιχνιδιών ήταν οι ***RPG Maker*** της **ASCII** και εμφανίστηκαν το 1998 και μετά. Την ίδια χρονιά, η εταιρία *Epic Games*, με ιδρυτή τον Tim Sweeney, πρωτοεμφάνισε την μηχανή ***Unreal Engine***. Παραδείγματα που φτιάχτηκαν με την *Unreal Engine* είναι τα βιντεοπαιχνίδια *Doom* και *Quake* της εταιρείας Id Software, που δεν κατασκευάστηκαν από την αρχή αλλά συνέβαλαν πολλοί δημιουργοί, ο καθένας σε διαφορετικό κομμάτι (λογισμικό, γραφικά, χαρακτήρες, όπλα και πίστες).

Κατά τη δημιουργία 3D βιντεοπαιχνιδιών, εμφανίστηκαν προβλήματα σχετικά με τους κανόνες του παιχνιδιού (όπως προβλήματα φυσικής) και διαχείρισης δεδομένων που σήμανε την ανάγκη για διαχωρισμό των δημιουργών σε ομάδες, αύξηση του ανθρώπινου δυναμικού και εξειδίκευση σε συγκεκριμένους τομείς. Στα επόμενα βιντεοπαιχνίδια, όπως το *Quake III Arena* που φτιάχτηκε με την Unreal το 1998, έγινε διαχωρισμός ρόλων σε 2 ομάδες. Η πρώτη απασχολήθηκε με το περιεχόμενο του παιχνιδιού και η δεύτερη με το λογισμικό της μηχανής Unreal. Στη συνέχεια και μέχρι σήμερα, έχει γίνει περεταίρω διαχωρισμός ρόλων σε ομάδες:

- Ομάδα για την ορθή απόδοση του παιχνιδιού (rendering).
- Προγραμματιστές.

- Καλλιτέχνες/σχεδιαστές.
- Σχεδιαστές πιστών/επιπέδων.
- Κ.α.

Οι επαναχρησιμοποιούμενες μηχανές βιντεοπαιχνιδιών έκαναν την διαδικασία ανάπτυξης βιντεοπαιχνιδιών ευκολότερη και γρηγορότερη. Έτσι, γύρω στο 2000, εμφανίστηκαν στην αγορά μεγάλο πλήθος βιντεοπαιχνιδιών που απέδωσε μεγάλα έσοδα στην βιομηχανία τους, αλλά αύξησε τον ανταγωνισμό.

Σήμερα, οι σύγχρονες μηχανές παιχνιδιών είναι πολύπλοκες σύνθετες εφαρμογές, που διαθέτουν δεκάδες αρκετά καλώς συντονισμένα συστήματα που αλληλεπιδρούν μεταξύ τους, για να εξασφαλίσουν έναν ακριβή χειρισμό στην εμπειρία του χρήστη. Λόγω των πολυπύρηνων σύγχρονων συστημάτων, στις μηχανές βιντεοπαιχνιδιών εφαρμόζεται συχνά η μέθοδος **Threading**, δηλαδή ο διαμερισμός των εργασιών σε: απόδοση (rendering), μετάδοση (streaming), ήχο (audio) και φυσική (physics).

Οι μηχανές βιντεοπαιχνιδιών πλέον μπορούν να εξάγουν την εφαρμογή – βιντεοπαιχνίδι για πολλά λειτουργικά συστήματα όπως:

- Για κινητά τηλέφωνα (Android, iOS)
- Για Ηλεκτρονικούς Υπολογιστές (Windows, MacOS, Linux)
- Για κονσόλες (PlayStation, Xbox)
- Web Browsers

Επίσης, είναι χτισμένες πάνω σε γλώσσες υψηλού επιπέδου όπως:

- Java
- C#
- .NET
- Python
- Lua
- Κ.α.

Ενώ τα 3D περισσότερα βιντεοπαιχνίδια βασίζονται στη κάρτα γραφικών που έχουν κάποιο όριο στις δυνατότητες τους, η μετάφραση από τις γλώσσες υψηλού επιπέδου σε γλώσσα μηχανής δεν προκαλεί κολλήματα ή καθυστερήσεις καθώς ο παίκτης παίζει. Παράλληλα, οι γλώσσες υψηλού επιπέδου διευκολύνουν πολύ τους προγραμματιστές λόγω της ευχρηστίας τους. Έτσι ενθαρρύνονται οι ανεξάρτητοι προγραμματιστές (indie

developers) να δημιουργήσουν και να προωθήσουν τα δικά τους μοναδικά παιχνίδια. [16]

Μερικές διάσημες μηχανές βιντεοπαιχνιδιών που υπάρχουν στην αγορά και προσφέρονται φθηνά ή δωρεάν είναι:

- Unity
- Unreal Engine
- CryEngine
- Coco’s Creator
- Amazon Lumberyard
- Godot
- GameMaker: Studio
- Κ.α. [17]

2. Περιεχόμενο βιντεοπαιχνιδιού *The Two Worlds*

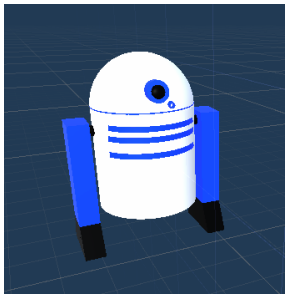
Στο κεφάλαιο αυτό, θα περιγραφεί ο σχεδιασμός και το περιεχόμενο του βιντεοπαιχνιδιού με τίτλο *The Two Worlds*, της παρούσας εργασίας.

Το βιντεοπαιχνίδι *The Two Worlds*, ανήκει στην κατηγορία των serious και εκπαιδευτικών βιντεοπαιχνιδιών. Είναι ένα διαδραστικό 3D βιντεοπαιχνίδι, με την έννοια ότι ο παίκτης μπορεί να αλληλεπιδράσει με το τρισδιάστατο περιβάλλον του. Απευθύνεται σε παιδιά ηλικίας μέχρι και 15 ετών. Σαν βαθμό δυσκολίας είναι εύκολο και συγχωρητικό, προσφέροντας έτσι ψυχαγωγία στον παίκτη.

Το βιντεοπαιχνίδι αυτό, υλοποιήθηκε στο **game engine Unity**, σε συνδυασμό με άλλα βοηθητικά πακέτα λογισμικού που θα αναλυθούν στο επόμενο κεφάλαιο.

2.1 Ιδέα του παιχνιδιού

Το παιχνίδι αποτελείται από δύο κόσμους. Στόχος του παιχνιδιού είναι ο παίκτης να ολοκληρώσει με επιτυχία και τους δύο κόσμους. Ο βασικός παίκτης – χαρακτήρας είναι ένα ρομποτάκι εμπνευσμένο από τη διάσημη σειρά ταινιών ***Star Wars*** (Εικόνα 3).



Εικόνα 3 Ο βασικός χαρακτήρας/παίκτης του παιχνιδιού The Two Worlds.

Στον 1ο κόσμο καλείται να απαντήσει υποχρεωτικά 3 συγκεκριμένες ερωτήσεις εκ των οποίων, τουλάχιστον μία από αυτές πρέπει να απαντηθεί σωστά για να μπορέσει να μεταβεί στον 2ο κόσμο.

Στον 2ο κόσμο καλείται να ακολουθήσει πιστά μία σειρά οδηγιών, με κεντρικό στόχο την επιβίωση του βασικού χαρακτήρα στα χιόνια.

2.1.1 Το αρχικό μενού

Με την έναρξη του παιχνιδιού, εμφανίζεται το λογότυπο της game engine Unity. Στη συνέχεια ακολουθεί το βασικό μενού με τις εξής επιλογές/κουμπιά (Εικόνα 4):

- *Παίξε!*
  Με αυτό το κουμπί ξεκινάει το παιχνίδι στον κόσμο **1**.
- *Πηγές*
  Αυτό το κουμπί μεταφέρει τον παίκτη σε ένα νέο μενού με τις πηγές που χρησιμοποιήθηκαν για το παιχνίδι και με τους υπερσυνδέσμους τους.
- *Έξοδος*
  Αυτό το κουμπί εμφανίζει ερώτηση αν ο παίκτης επιθυμεί να κλείσει το παιχνίδι.

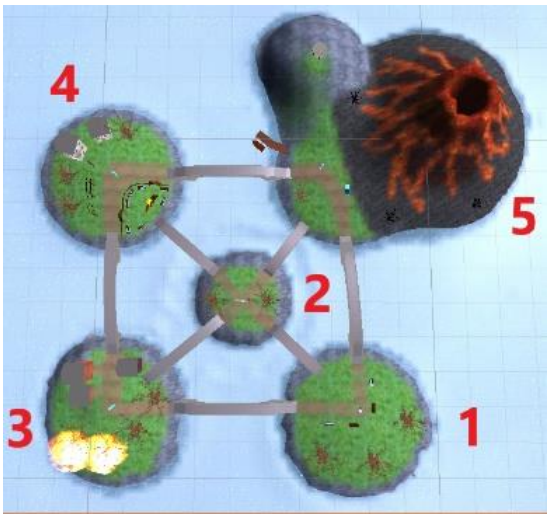
Εικόνα 4 Το αρχικό μενού στο παιχνίδι The Two Worlds.

2.1.2 Ο κόσμος 1

Κύριο θέμα του **κόσμου 1** είναι τα **μέτρα προστασίας σε περιπτώσεις φυσικών καταστροφών**. Οι φυσικές καταστροφές που παρουσιάζονται στο παιχνίδι είναι ο σεισμός, η φωτιά και η έκρηξη ηφαιστείου.

Ο κόσμος 1 αποτελείται από 5 μικρά νησιά (Εικόνα 5):

1. Το πρώτο νησί είναι η αφετηρία του παίκτη.
2. Το δεύτερο νησί έχει τη πύλη μετάβασης για τον κόσμο 2.
3. Το τρίτο νησί περιέχει την ερώτηση για την αντιμετώπιση της φωτιάς.
4. Το τέταρτο νησί περιέχει την ερώτηση για τα μέτρα προστασίας σε περίπτωση σεισμού.
5. Το πέμπτο νησί περιέχει την ερώτηση για τα μέτρα προστασίας σε περίπτωση έκρηξης ηφαιστείου.



Εικόνα 5 Ο χάρτης του κόσμου 1 (κάτοψη).

Τα νησιά συνδέονται μεταξύ τους με γέφυρες. Ο παίκτης έχει την επιλογή να περιηγηθεί στα νησιά και να απαντήσει τις ερωτήσεις με όποια σειρά θέλει.

*Νησί Αφετηρίας (**1**)*

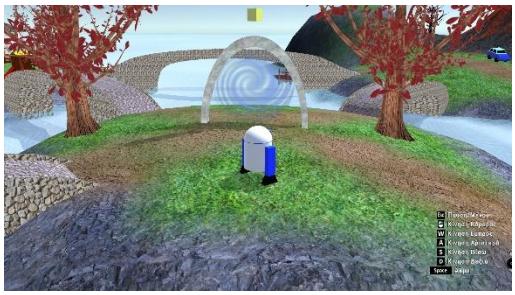
Είναι το νησί εκκίνησης του παίκτη (Εικόνα 6).


*Εικόνα 6 Νησί Αφετηρίας στον κόσμο **1**.*

*Νησί με τη πύλη μετάβασης για τον κόσμο 2 (**2**)*

Είναι το νησί, στο οποίο, υπάρχει η πύλη που θα μεταφέρει τον παίκτη στον κόσμο 2 (Εικόνα 7). Η πύλη δεν του επιτρέπει τη μεταφορά σε περίπτωση που κάποια ερώτηση δεν έχει απαντηθεί ή αν όλες οι ερωτήσεις έχουν απαντηθεί λάθος. Τα πιθανά σενάρια είναι τα εξής:

Σε περίπτωση που ο παίκτης περάσει μέσα ή κοντά από την πύλη ενώ δεν έχουν απαντηθεί όλες οι ερωτήσεις, τότε του εμφανίζεται κατάλληλο μήνυμα για το ποιες ερωτήσεις δεν έχουν απαντηθεί.



*Εικόνα 7 Το νησί με την πύλη μεταφοράς από τον κόσμο **1** για τον κόσμο **2**.*

Σε περίπτωση που ο παίκτης περάσει μέσα ή κοντά από την πύλη και δεν έχει απαντήσει καμία ερώτηση σωστά, τότε του εμφανίζεται κατάλληλο μήνυμα να ξαναπαίξει από την αρχή την πίστα.

Σε κάθε άλλη περίπτωση, εμφανίζεται στον παίκτη μήνυμα για την πρόοδο του και οι επιλογές να παραμείνει ή να προχωρήσει στον επόμενο κόσμο. Αν απαντήθηκαν 2 σωστές από τις τρεις, εμφανίζεται το μήνυμα «Τα πήγες καλά, όχι τέλεια.». Αν απαντήθηκαν και οι 3 ερωτήσεις σωστά, εμφανίζεται το μήνυμα «Τα πήγες τέλεια.».

*Νησί με την ερώτηση για τη φωτιά (**3**)*

Το νησί αυτό αναπαριστά ένα μικρό κατοικημένο χωριό (Εικόνα 8). Το ένα μέρος του νησιού, έχει ένα μικρό δάσος που έχει πιάσει φωτιά. Στο κέντρο του νησιού υπάρχει ένα θαυμαστικό που περιστρέφεται γύρω από τον εαυτό του. Ο παίκτης περνώντας πολύ κοντά ή μέσα από το θαυμαστικό αυτό, περιληπτικά, του εμφανίζεται η εξής ερώτηση: «Το δάσος έχει πιάσει φωτιά, ποια τα μέτρα προστασίας για να μην καεί το χωριό;». Στο μήνυμα υπάρχουν τρεις διαθέσιμες επιλογές – απαντήσεις στη μορφή κουμπιών. Περιληπτικά, αυτές είναι:

Εικόνα 8 Το νησί με την ερώτηση για τη φωτιά.

- Να καλέσουμε το 199, δηλαδή την πυροσβεστική. (Σωστή)
- Να ειδοποιήσουμε τους κατοίκους της περιοχής. (Λάθος)
- Να επιχειρήσουμε να σβήσουμε τη φωτιά μόνοι μας. (Λάθος)

Αν ο παίκτης απαντήσει σωστά, η φωτιά σβήνει, το δάσος και το χωριό σώζονται. Σε οποιαδήποτε άλλη περίπτωση, το δάσος καίγεται ολοκληρωτικά, το διπλανό σπίτι καίγεται και καταστρέφεται και ο παίκτης βγάζει από το κεφάλι του καπνούς που τον ακολουθούν μέχρι να ολοκληρώσει τον κόσμο 1 (Εικόνα 9).

Εικόνα 9 Επιπτώσεις στο νησί, αν ο παίκτης απαντήσει λάθος.

Νησί με την ερώτηση για το σεισμό (4)

Το νησί αυτό αναπαριστά μια κατοικημένη περιοχή με παλιά σπίτια (Εικόνα 10). Στη μία μεριά του νησιού, υπάρχουν κολώνες ρεύματος, δίπλα από τις κολώνες υπάρχουν τα παλιά σπίτια και στην άλλη μεριά του νησιού, υπάρχει ένα πάρκο.

Στο κέντρο του νησιού, υπάρχει ένα θαυμαστικό. Ο παίκτης περνώντας πολύ κοντά ή μέσα από το θαυμαστικό αυτό, περιληπτικά, του εμφανίζεται η εξής ερώτηση: «Καθώς περνάμε από αυτήν την περιοχή, ξεσπάει ένας δυνατός σεισμός. Που πρέπει να πάμε για να προφυλαχθούμε;». Με το κλείσιμο του μηνύματος, ο παίκτης αισθάνεται τον σεισμό. Σε κάθε περιοχή του νησιού, εμφανίζεται από ένα θαυμαστικό. Ο παίκτης καλείται να κινηθεί προς το θαυμαστικό της περιοχής που θεωρεί κατάλληλη για να προφυλαχθεί. Τα πιθανά αποτελέσματα είναι τα εξής:

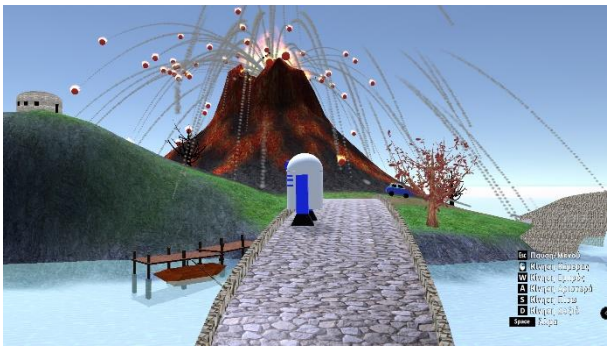
Εικόνα 10 Το νησί με την ερώτηση για το σεισμό.

- Ο παίκτης κινείται προς το πάρκο. (Σωστό)
  Σε αυτή την περίπτωση του εμφανίζεται μήνυμα ότι η επιλογή του ήταν η σωστή.
- Ο παίκτης κινείται προς τα παλιά σπίτια. (Λάθος)
  Σε αυτή την περίπτωση ο παίκτης βγάζει σπίθες από το κεφάλι του και τα πόδια του δείχνουν να είναι ραγισμένα. Το εφέ παραμένει μέχρι να ολοκληρώσει τον κόσμο 1.

- Ο παίκτης κινείται προς τις κολώνες ρεύματος. (Λάθος)
  Σε αυτή την περίπτωση ο παίκτης βγάζει σπίθες από το κεφάλι του και τα πόδια του δείχνουν να είναι ραγισμένα. Το εφέ παραμένει μέχρι να ολοκληρώσει τον κόσμο 1.

Νησί με την ερώτηση για το ηφαίστειο (5)

Στο νησί αυτό αναπαρίστανται ένα ηφαίστειο, ένα όχημα, ένα καταφύγιο ψηλά σε λόφο και μία μικρή βάρκα δίπλα στη θάλασσα (Εικόνα 11).



Εικόνα 11 Το νησί με το ηφαίστειο.

Στο κέντρο του νησιού, υπάρχει ένα θαυμαστικό. Ο παίκτης περνώντας πολύ κοντά ή μέσα από το θαυμαστικό αυτό, περιληπτικά, του εμφανίζεται η εξής ερώτηση: «Καθώς περνάμε από αυτήν την περιοχή, εκρήγνυται το ηφαίστειο. Που πρέπει να πάμε για να προφυλαχθούμε;». Με το κλείσιμο του μηνύματος, ο παίκτης αισθάνεται έναν πολύ δυνατό σεισμό. Επίσης, μέσα από το ηφαίστειο εκσφενδονίζονται μπάλες φωτιάς. Δίπλα στο όχημα, δίπλα στη μικρή βάρκα και δίπλα στο καταφύγιο, εμφανίζεται και από ένα θαυμαστικό. Ο παίκτης καλείται να κινηθεί προς το θαυμαστικό της περιοχής που θεωρεί κατάλληλη για να προφυλαχθεί. Τα πιθανά αποτελέσματα είναι τα εξής:

- Ο παίκτης κινείται προς το καταφύγιο ψηλά στο λόφο. (Σωστό)
  Σε αυτή την περίπτωση του εμφανίζεται μήνυμα ότι η επιλογή του ήταν η σωστή.
- Ο παίκτης κινείται προς τη μικρή βάρκα. (Λάθος)
  Σε αυτή την περίπτωση ο παίκτης καίγεται και το σώμα του γίνεται κόκκινο. Το εφέ παραμένει μέχρι να ολοκληρώσει τον κόσμο 1.
- Ο παίκτης κινείται προς το όχημα. (Λάθος)
  Σε αυτή την περίπτωση ο παίκτης καίγεται και το σώμα του γίνεται κόκκινο. Το εφέ παραμένει μέχρι να ολοκληρώσει τον κόσμο 1.

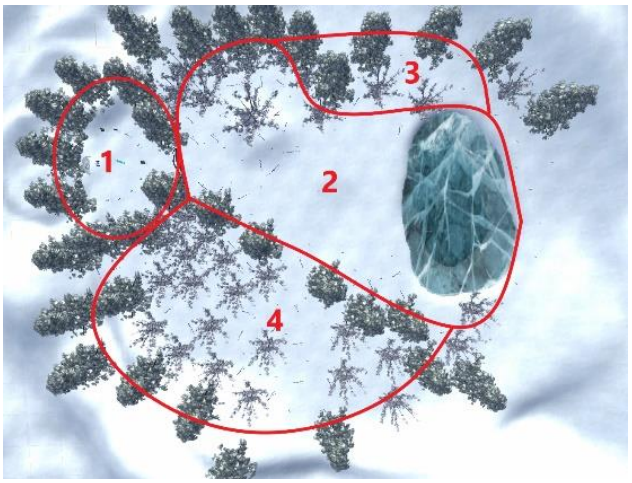
Σημείωση: Σε περίπτωση που ο παίκτης απαντά κάποια ερώτηση λάθος, τα προηγούμενα εφέ πάνω στον παίκτη από προηγούμενες λανθασμένες απαντήσεις, αν υπάρχουν, παραμένουν.

2.1.3 Ο κόσμος 2

Κύριο θέμα του **κόσμου 2** είναι η **επιβίωση στα χιόνια**. Η ιστορία εξελίσσεται με ξεκάθαρες οδηγίες που δίνονται στον παίκτη. Σε κάθε φάση της ιστορίας, υπάρχει ένα βοηθητικό θαυμαστικό με μορφή κουμπιού κάτω δεξιά της οθόνης, που διευκρινίζει στον παίκτη κάθε φορά τι πρέπει να κάνει, σε περίπτωση που ο παίκτης χαθεί.

Ο κόσμος 2 χωρίζεται σε 4 μεγάλες περιοχές (Εικόνα 12):

1. Η περιοχή εκκίνησης.
2. Η περιοχή επιβίωσης. Εδώ εξελίσσεται όλη η επιβίωση του παίκτη.
3. και **4.** Είναι κομμάτια δάσους που ο παίκτης μπορεί να περιηγηθεί και να μαζέψει υλικά από το έδαφος (πέτρες και ξύλα).

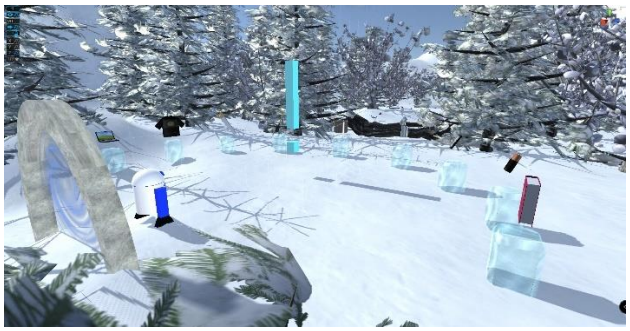


Εικόνα 12 Ο χάρτης του κόσμου 2 (κάτοψη).

*Περιοχή εκκίνησης (**1**)*

Αυτή η περιοχή είναι το σημείο εκκίνησης για τον παίκτη στον κόσμο 2 (Εικόνα 13). Εδώ υπάρχει η πύλη που θεωρητικά μετέφερε τον παίκτη από τον κόσμο 1. Ο παίκτης δεν μπορεί να αλληλεπιδράσει περεταίρω με την πύλη.

Στο κέντρο της περιοχής υπάρχει ένα θαυμαστικό. Χωρίς να είναι υποχρεωτικό για την εξέλιξη της ιστορίας, αν ο παίκτης περάσει μέσα από αυτό, του εμφανίζονται οδηγίες για το τι πρέπει να κάνει.


*Εικόνα 13 Κόσμος **2**, επιλογή των 4 αντικειμένων από τα 9.*

Μπροστά από την πύλη υπάρχουν 9 παγωμένοι στύλοι με περιστρεφόμενα αντικείμενα από πάνω τους. Ο παίκτης καλείται να επιλέξει 4 από αυτά κάνοντας κλικ πάνω τους, που θα τον βοηθήσουν στη μετέπειτα επιβίωση του. Μόνο 4 από αυτά είναι τα σωστά. Το πέρασμα από την περιοχή 1 στην περιοχή 2 είναι

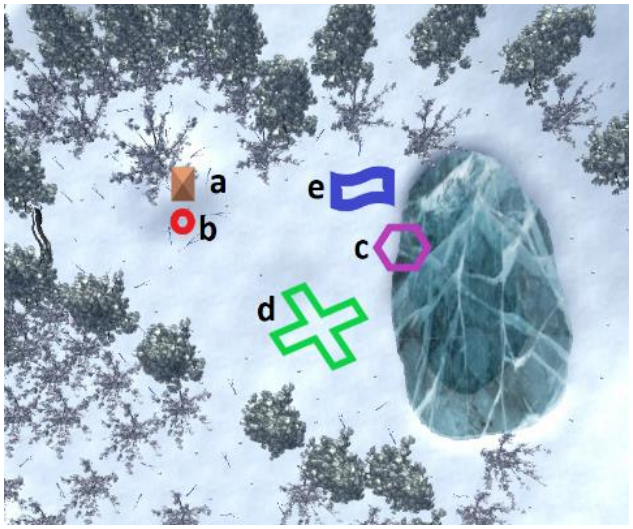
αρχικά μπλοκαρισμένο και δεν επιτρέπει στον παίκτη να περάσει, εμφανίζοντας κατάλληλο μήνυμα.

Τα 9 αντικείμενα πάνω στους παγωμένους στύλους είναι: ένα laptop, μία κοντομάνικη μπλούζα, πριόνι, φτυάρι, μεταλλικός κουβάς, αναπτήρας, κινητό τηλέφωνο, μπαταρίες, βιβλίο ιστορίας. Τα 4 σωστά αντικείμενα είναι: το πριόνι, το φτυάρι, ο μεταλλικός κουβάς και ο αναπτήρας.

Μόλις ο παίκτης επιλέξει τα σωστά αντικείμενα, του εμφανίζεται κατάλληλο μήνυμα για να περάσει από την περιοχή 1 στην περιοχή 2 και τα εμπόδια του περάσματος εξαφανίζονται.

Περιοχή επιβίωσης (2)

Η επιβίωση του παίκτη πραγματοποιείται σε φάσεις. Κάθε φάση λαμβάνει χώρα στα σημεία **a**, **b**, **c**, **d**, **e** της περιοχής **2**, όπως φαίνεται στην Εικόνα 14. Οι φάσεις είναι:



Εικόνα 14 Η περιοχή επιβίωσης (2). Στα σημεία a, b, c, d, e πραγματοποιείται η κάθε φάση επιβίωσης.

- **a.** Η φάση χτισίματος της σκηνής.
- **b.** Η φάση δημιουργίας της φωτιάς.
- **c.** Η φάση λήψης νερού.
- **d.** Η φάση δημιουργίας S.O.S.
- **e.** Η φάση αποχώρησης από τον κόσμο **2** με ελικόπτερο.

Από αυτό το σημείο και έπειτα, είναι υποχρεωτικό, ο παίκτης να περνάει μέσα από κάθε θαυμαστικό που του εμφανίζεται. Για να συλλέξει οτιδήποτε (πέτρες, ξύλα) ή να αλληλεπιδράσει με οποιοδήποτε σημείο (χτίσιμο σκηνής, φωτιάς, κ.λπ.), πρέπει να πλησιάσει πολύ κοντά σε αυτό και να κάνει κλικ πάνω του.

Η φάση χτισίματος της σκηνής

Τη στιγμή που ο παίκτης μπει στην περιοχή **2**, υπάρχει ήδη ένα θαυμαστικό στο κέντρο της. Εκεί του δίνονται οι οδηγίες για το που να

φτιάξει τη σκηνή. Τότε, εμφανίζεται ένα θαυμαστικό πάνω από το σημείο **a**.

Το πρώτο βήμα είναι να σκάψει τη βάση της σκηνής για να διώξει τα χιόνια. Αυτό πραγματοποιείται πατώντας το κουμπί «Σκάψε!» που του εμφανίζεται.

Το δεύτερο βήμα είναι να κατασκευάσει τη στέγη της σκηνής. Εκεί καλείται να συλλέξει ξύλα από το έδαφος κάνοντας κλικ πάνω τους. Ο παίκτης μπορεί να κρατάει πάνω του μέχρι και 3 ξύλα κάθε φορά. Στα δεξιά της οθόνης εμφανίζονται βοηθητικά μηνύματα για το πόσα ξύλα κρατάει ακριβώς πάνω του και πόσα ξύλα περίπου χρειάζεται για να ολοκληρώσει τη στέγη. Ανά πάσα στιγμή μπορεί να αφήσει τα ξύλα που κρατάει κάνοντας κλικ πάνω στη σκηνή.

Το τρίτο βήμα σε αυτή τη φάση είναι να προσθέσει χιόνι στη στέγη της σκηνής. Αυτό ολοκληρώνεται, κάνοντας κλικ πάνω στο κουμπί «Πρόσθεσε Χιόνι!» που του εμφανίζεται (Εικόνα 16). Ακολουθούν σχετικές οδηγίες για την επόμενη φάση.

Η φάση δημιουργίας της φωτιάς

Με την ολοκλήρωση της σκηνής, εμφανίζεται ένα θαυμαστικό πάνω στο σημείο **b**, όπου θα δημιουργηθεί η φωτιά. Μόλις ο παίκτης περάσει μέσα από το θαυμαστικό, του δίνονται οι σχετικές οδηγίες.

Το πρώτο βήμα είναι να σκάψει τη βάση της φωτιάς για να διώξει τα χιόνια. Αυτό πραγματοποιείται πατώντας το κουμπί «Σκάψε!» που του εμφανίζεται.

Το επόμενο βήμα είναι να μαζέψει πέτρες και ξύλα από το έδαφος κάνοντας κλικ πάνω τους. Ο παίκτης μπορεί να κρατάει πάνω του μέχρι 3 ξύλα και 3 πέτρες ταυτόχρονα. Στα δεξιά της οθόνης εμφανίζονται βοηθητικά μηνύματα για το πόσα ξύλα και πόσες πέτρες κρατάει ακριβώς πάνω του και πόσα ξύλα και πέτρες περίπου χρειάζεται για να ολοκληρώσει τη φωτιά. Ανά πάσα στιγμή μπορεί να αφήσει τα ξύλα και τις πέτρες που κρατάει κάνοντας κλικ πάνω στη φωτιά.

Το τρίτο βήμα είναι να ανάψει τη φωτιά. Αυτό ολοκληρώνεται πατώντας το κουμπί «Άναψε!» που του εμφανίζεται (Εικόνα 16). Ακολουθούν σχετικές οδηγίες για την επόμενη φάση.

Η φάση λήψης νερού

Με την ολοκλήρωση της φωτιάς εμφανίζεται ένα θαυμαστικό πάνω από την παγωμένη λίμνη, στο σημείο **c**. Ταυτόχρονα, στην πλάτη του παίκτη εμφανίζεται ένας μεταλλικός κουβάς (ένα από τα σωστά αντικείμενα που επέλεξε ο παίκτης στην περιοχή **1**).

Εικόνα 15 Συλλογή νερού από τη λίμνη.

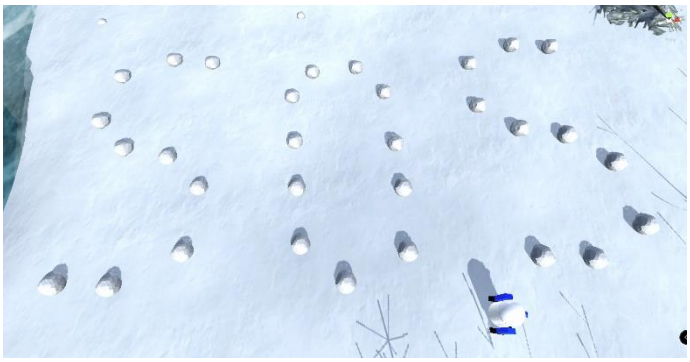
Το πρώτο βήμα είναι να πάει στο θαυμαστικό της παγωμένης λίμνης και να συλλέξει νερό πατώντας το κουμπί «Συνέχεια» που του εμφανίζεται (Εικόνα 15). Σε αυτό το σημείο, ο παίκτης χρησιμοποιεί θεωρητικά το φτυάρι που επέλεξε στην περιοχή **1** για να σπάσει τον πάγο. Στην συνέχεια, οδηγίες τον πληροφορούν να κινηθεί προς το θαυμαστικό που εμφανίζεται στο σημείο **b** της φωτιάς. Πηγαίνοντας προς τη φωτιά, ο κουβάς στην πλάτη του παίκτη δείχνει να είναι γεμάτος νερό.

Εικόνα 16 Η σκηνή, η φωτιά και ο κουβάς με το νερό.

Το επόμενο βήμα είναι να αφήσει τον κουβά στο σημείο **b**, αφού πλησιάσει το θαυμαστικό στο σημείο εκείνο, και να πατήσει το κουμπί «Συνέχεια» που του εμφανίζεται (Εικόνα 16). Σε αυτό το σημείο, θεωρητικά, ο παίκτης ζεσταίνει το νερό που είναι απαραίτητο για την επιβίωσή του. Ακολουθούν σχετικές οδηγίες για την επόμενη φάση.

Η φάση δημιουργίας S.O.S.

Αμέσως μετά τη φάση λήψης νερού, εμφανίζεται ένα θαυμαστικό κοντά στη λίμνη, στο σημείο **d**. Εκεί θα πρέπει ο παίκτης να δημιουργήσει ένα μεγάλο **S.O.S.** από πέτρες. Το μόνο που πρέπει να κάνει είναι να μαζέψει πέτρες από το έδαφος κάνοντας κλικ πάνω τους. Μπορεί να κρατάει πάνω του μέχρι και 5 πέτρες. Στα δεξιά της οθόνης εμφανίζονται βοηθητικά μηνύματα για το πόσες πέτρες κρατάει ακριβώς πάνω του και πόσες πέτρες περίπου χρειάζεται ακόμη για να ολοκληρώσει το **S.O.S.**



Εικόνα 17 Το S.O.S. ολοκληρωμένο.

Ανά πάσα στιγμή μπορεί να αφήσει τις πέτρες κάνοντας κλικ πάνω στο **S.O.S.** στο σημείο **d**. Αφού ολοκληρώσει το **S.O.S**. (Εικόνα 17). ακολουθούν σχετικές οδηγίες για την επόμενη φάση.

Η φάση αποχώρησης από τον κόσμο **2** με ελικόπτερο

Μόλις ολοκληρωθεί το S.O.S., εμφανίζεται σχετικό μήνυμα να κινηθεί στο σημείο **e**. Στο σημείο **e** εμφανίζεται ένα θαυμαστικό και ένα ελικόπτερο να πετάει ψηλά. Ο παίκτης μόλις πλησιάσει το ελικόπτερο, αυτό προσγειώνεται στο έδαφος και το θαυμαστικό εξαφανίζεται (Εικόνα 18). Ο παίκτης, όσο είναι κοντά στο ελικόπτερο, έχει την επιλογή να αναχωρήσει ή να παραμείνει στον κόσμο **2**.

Εικόνα 18 Η άφιξη του ελικοπτέρου.

2.1.4 Τελική Σκηνή

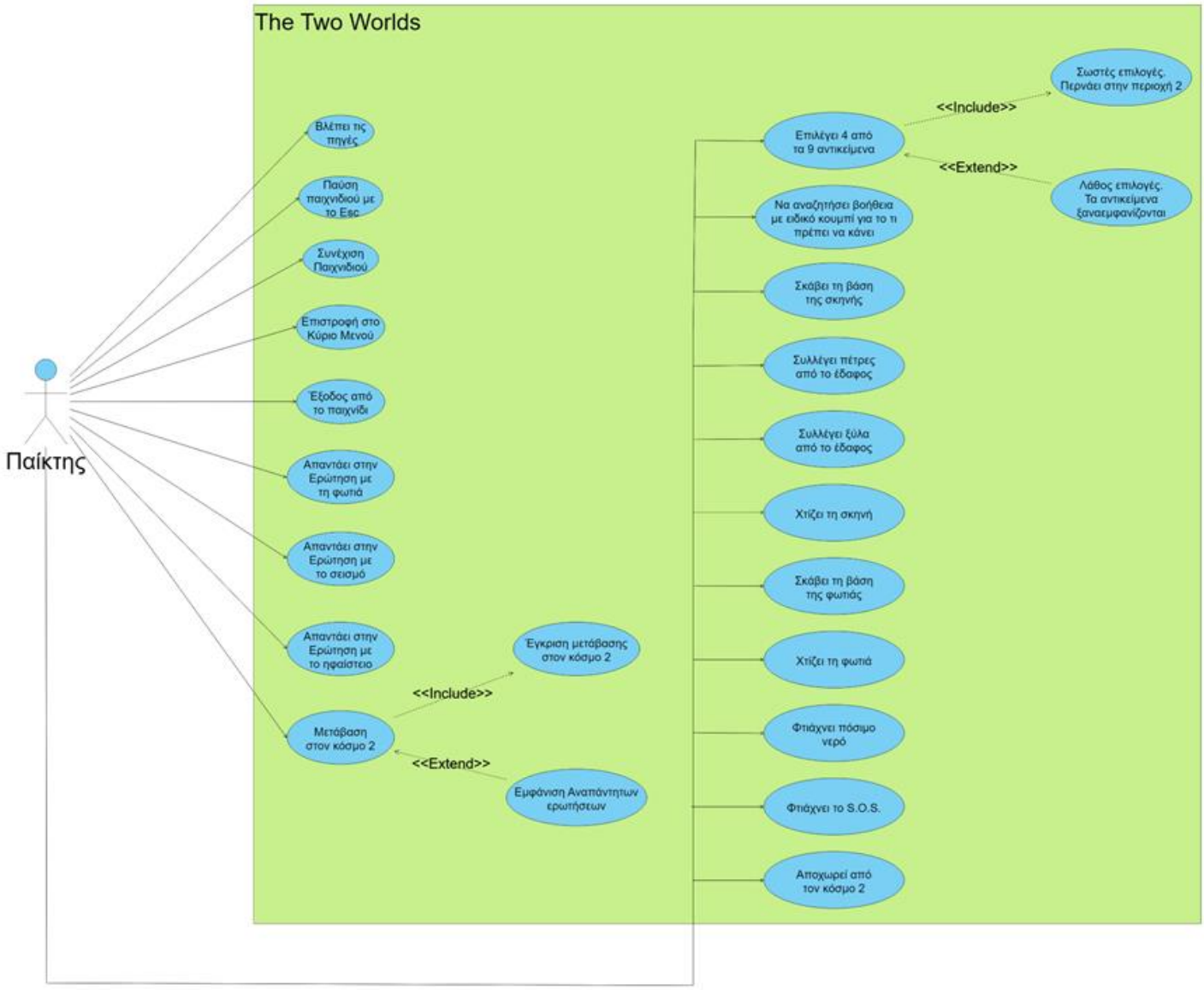
Με την αναχώρηση του παίκτη από τον κόσμο 2 εμφανίζεται η τελική σκηνή του παιχνιδιού (Εικόνα 19). Η μόνη επιλογή της τελικής σκηνής είναι η επιστροφή στο αρχικό μενού με σχετικό κουμπί στο κάτω μέρος της οθόνης. Επίσης, εμφανίζεται στην οθόνη η επίδοση του παίκτη σε κάθε κόσμο:

- Στον κόσμο **1** αξιολογούνται ποιες ερωτήσεις απαντήθηκαν σωστά και ποιες λάθος.
- Στον κόσμο **2** αναγράφεται ο χρόνος που χρειάστηκε ο παίκτης για να τερματίσει τον κόσμο **2**. Το χρονόμετρο αυτό είναι κρυφό και άγνωστο στον παίκτη όταν παίζει για πρώτη φορά. Ο βασικότερος λόγος είναι για να μην προκληθεί άγχος στον παίκτη και να εστιάσει στη νέα γνώση που του προσφέρεται. Ένας δεύτερος λόγος είναι η δημιουργία πρόκλησης στον παίκτη για να εξερευνήσει ξανά το παιχνίδι.

Εικόνα 19 Στιγμιότυπο της τελικής σκηνής.

2.2 Πιθανά σενάρια χρήσης (Use Case)

Στην Εικόνα 20 εμφανίζονται τα πιθανά σενάρια χρήσης του βιντεοπαιχνιδιού *The Two Worlds*, με τη βοήθεια του διαγράμματος Use Case.



Εικόνα 20 Use Case Diagram του παιχνιδιού The Two Worlds

3. Υλοποίηση βιντεοπαιχνιδιού *The Two Worlds*

Σε αυτό το κεφάλαιο θα αναλυθεί, πως υλοποιήθηκε το παιχνίδι *The Two Worlds* στο game engine Unity. Συγκεκριμένα θα περιγραφεί κάθε τεχνικό κομμάτι των:

- Ελάχιστων απαιτήσεων του Unity και του παιχνιδιού.
- Η δημιουργία των παραθύρων και μενού.
- Τα έξτρα βοηθητικά πακέτα του Unity και τρίτα λογισμικά.
- Δημιουργία του 3D περιβάλλοντος.
- Δημιουργία των 3D μοντέλων.
- Scripts.
- Game testing και rendering για την απόδοση του παιχνιδιού.
- Ειδικών εφέ που χρησιμοποιηθήκαν για το παιχνίδι.

Ορισμοί

Συχνά θα χρησιμοποιούνται οι παρακάτω όροι, γι' αυτό είναι χρήσιμο να εξηγηθούν:

Asset

Είναι ένα αντικείμενο ή αρχείο που μπορεί να χρησιμοποιηθεί μέσα στο project του Unity και υποστηρίζεται από το Unity.

Τέτοια μπορεί να είναι:

- Αρχείο που δημιουργήθηκε μέσα ή έξω από το Unity
- 3D ή 2D μοντέλα
- Animations, ρυθμιστής animation (animation controller)
- Αρχείο ήχου, μείκτης ήχου (audio mixer)
- Εικόνα
- Υφές (texture)
- Βοηθητικά έξτρα εργαλεία (tools and kits)
- Πακέτα με έτοιμο υλικό, π.χ. έτοιμες σκηνές με αντικείμενα
- Σκηνές
- Scripts
- Εφέ
- Και άλλα. [18]

Texture

Texture είναι η **υφή** ενός αντικειμένου. Textures στο Unity είναι εικόνες τύπου (**.png**) που μπορούν να προστεθούν μέσα σε ένα material. Για το Unity, είναι αδύνατο να τοποθετηθεί απευθείας ένα texture πάνω σε ένα 3D αντικείμενο της σκηνής. Απαιτείται πρώτα να δημιουργηθεί material και μετά να προστεθεί η υφή μέσα σε αυτό.

Material

Το **material** είναι το **υλικό** ενός 3D αντικειμένου. Δημιουργείται από την κλάση *material*. Το υλικό αυτό κληρονομεί ιδιότητες από την κλάση, όπως το χρώμα, η λάμψη, η υφή, κλπ. Η προσθήκη υφής (texture) σε ένα material δεν είναι υποχρεωτική. Για παράδειγμα, για την αναπαράσταση ενός μεταλλικού αντικειμένου, πρέπει να φτιαχτεί ένα material με χρώμα γκρι και μεταλλική λάμψη.

Collider – Mesh Collider

Ένας **Collider** στο Unity ορίζει το σχήμα ενός αντικειμένου για τους σκοπούς της φυσικής σύγκρουσης με άλλα αντικείμενα. Ένας Collider, συνήθως είναι αόρατος στο μάτι του παίκτη και δεν είναι απαραίτητο να έχει το ακριβές σχήμα του αντικειμένου. Τα συνήθη είδη 3D Collider είναι κυβικό (Box Collider), σφαιρικό (Sphere Collider), κάψουλα (Capsule Collider).

Ένας Collider που έχει το ακριβές σχήμα – πλέγμα υπόστασης του αντικειμένου λέγεται **Mesh Collider**.

Prefab

Το **prefab** είναι ένα επαναχρησιμοποιούμενο asset που κατασκευάζει ο ίδιος ο δημιουργός. Στο asset αυτό αποθηκεύονται όλες οι ιδιότητες του, οι τιμές στις μεταβλητές του και τα εξαρτώμενα αντικείμενα που ανήκουν σε αυτό. Το asset αυτό λειτουργεί σαν πρότυπο με το οποίο είναι δυνατό να δημιουργηθούν νέα αντίγραφά του μέσα στη σκηνή.

Shaders

Οι Shaders είναι μικρά scripts που περιλαμβάνουν μαθηματικούς υπολογισμούς και αλγόριθμους που υπολογίζουν το χρώμα του κάθε pixel σε ένα υλικό. Οι υπολογισμοί εφαρμόζονται σε τιμές φωτισμού που αποδίδει ο χρήστης, σε συνδυασμό με τη διαμόρφωση ενός material. Με άλλα λόγια, είναι μία διεπαφή που επιτρέπει την επεξεργασία ενός material για την προσθήκη εφέ πάνω του [19].

Shader Graphs και VFX Graphs

Είναι **editors** με οπτικό περιβάλλον, με τους οποίους ο χρήστης μπορεί να φτιάξει materials με εφέ, μέσω ενός συστήματος κόμβων (Nodes) και να τα παρακολουθεί ταυτόχρονα το αποτέλεσμα τους, χωρίς να χρησιμοποιήσει καθόλου κώδικα. Οι κόμβοι είναι έτοιμες συναρτήσεις, οι οποίες δέχονται τιμές εισόδου και βγάζουν κάποιες τιμές εξόδου. Η σύνδεση των κόμβων μεταξύ τους πραγματοποιείται με ακμές σχηματίζοντας έτσι έναν γράφο. Ο χρήστης φτιάχνει έναν γράφο για να επιτύχει το επιθυμητό αποτέλεσμα σε material (π.χ. Εικόνα 33).

3.1. Ενδεικτικές ελάχιστες απαιτήσεις για το Unity και για το βιντεοπαιχνίδι

Ενδεικτικές ελάχιστες απαιτήσεις συστήματος για το *Unity editor*, έκδοση 2020.1.1f1 [20] :

Ελάχιστες Απαιτήσεις Συστήματος	Windows	macOS	Linux
Λειτουργικό Σύστημα	Windows 7 με SP1 ή νεότερο και Windows 10, μόνο 64-bit έκδοση	High Sierra 10.13+	Ubuntu 16.04, Ubuntu 18.04 και CentOS 7
Επεξεργαστής	Αρχιτεκτονικής x64 με υποστήριξη εντολών SSE2	Αρχιτεκτονικής x64 με υποστήριξη εντολών SSE2	Αρχιτεκτονικής x64 με υποστήριξη εντολών SSE2
Κάρτα γραφικών API	DX10, DX11 και DX12 και δυνατή κάρτα γραφικών	Δυνατούς Intel ή AMD κάρτες γραφικών	OpenGL 3.2+ Nvidia ή AMD GPUs.
Επιπλέον απαιτήσεις	Αυθεντικούς οδηγούς προγραμμάτων υλικού κατασκευαστή	Αυθεντικούς οδηγούς προγραμμάτων της Apple	Υπολογιστές Gnome οι οποίοι τρέχουν πάνω σε X11 σύστημα παραθύρων. Αυθεντικούς οδηγούς προγραμμάτων για Nvidia κάρτες γραφικών ή Mesa οδηγό προγράμματος για AMD.

Σημείωση: είναι σημαντικό να τονιστεί ότι ένα project που υλοποιείται σε μία συγκεκριμένη έκδοση του Unity editor, υπάρχει πιθανότητα να μην μπορεί να λειτουργήσει σε παλαιότερη ή νεότερη έκδοση. Για το λόγο αυτό είναι σύνηθες, ο δημιουργός, να ξεκινάει και να ολοκληρώνει το project του σε μία μόνο έκδοση.

Ενδεικτικές ελάχιστες απαιτήσεις συστήματος του παιχνιδιού *The Two Worlds*:

Λειτουργικό Σύστημα	Windows 7 ή νεότερη έκδοση
Επεξεργαστής	2.8 GHz
Μνήμη RAM	3 GB RAM
Κάρτα Γραφικών	Κάρτα γραφικών με τουλάχιστον 1 GB εικονική μνήμη (για 1080p)
Αποθηκευτικός χώρος	150 MB ελεύθερος χώρος

3.2. Πλοήγηση του παίκτη

Ο παίκτης (Εικόνα 3) κινείται με σταθερή ταχύτητα και έχει βάρος που υπακούει στους νόμους της φυσικής και έτσι μπορεί να προσγειώνεται στο έδαφος όταν κάνει άλμα. Μπορεί να παίξει με τον συνήθη τρόπο χειρισμού που εμφανίζεται πλέον στα σύγχρονα βιντεοπαιχνίδια και αποκαλείται "**WASD**". Αυτός είναι τα εξής κουμπιά (Εικόνα 21):

- Μπροστά, με το κουμπί **W** ή το βέλος πάνω «↑» .
- Πίσω, με το κουμπί **S** ή το βέλος πάνω «↓» .
- Δεξιά, με το κουμπί **D** ή το βέλος πάνω «→» .
- Αριστερά, με το κουμπί **A** ή το βέλος πάνω «←» .
- Διαγώνια με τον συνδυασμό των παραπάνω κουμπιών.
- Άλμα, με το κουμπί «**Space**».
- Να κοιτάει τον χώρο γύρω του (ουσιαστικά γίνεται περιστροφή της κάμερας), κρατώντας πατημένο το **δεξί κλικ** του ποντικιού.
- Παύση του παιχνιδιού, με το κουμπί «**Esc**» και εμφάνιση διεπαφής με τις επιλογές: «*Συνέχεια*», «*Κύριο Μενού*», «*Έξοδος*».

Εικόνα 21 Στιγμιότυπο των βασικών χειρισμών/κουμπιών στο παιχνίδι.

Σε περίπτωση που ο παίκτης δεν είναι εξοικειωμένος με τα κουμπιά, υπάρχει πάντα μια βοηθητική περιγραφή στη κάτω δεξιά γωνία της οθόνης του παιχνιδιού (Εικόνα 21).

Το script για τον κώδικα για την κίνηση του παίκτη περιγράφεται στο κεφάλαιο κεφ. 3.11.1.

3.3. Σκηνές (Scenes)

Είναι assets που περιέχουν μέρος ή ολόκληρο το παιχνίδι. Κάθε σκηνή έχει το δικό της περιβάλλον, χαρακτήρες, εμπόδια, αντικείμενα και τη δική του διεπαφή (User Interface). Το βιντεοπαιχνίδι είναι χωρισμένο σε **7** σκηνές:

- Πρώτη σκηνή: το αρχικό μενού

- Δεύτερη σκηνή: η σκηνή με όλες τις κατηγορίες πηγών
- Τρίτη σκηνή: η σκηνή με τις πηγές του κόσμου **1**
- Τέταρτη σκηνή: η σκηνή με τις πηγές του κόσμου **2**
- Πέμπτη σκηνή: ο κόσμος **1**
- Έκτη σκηνή: ο κόσμος **2**
- Έβδομη σκηνή: η τελική σκηνή με την απόδοση του παίκτη

3.4. Η κάμερα του χώρου (main camera)

Η κάμερα είναι ένα αντικείμενο της σκηνής μέσα από το οποίο βλέπει ο παίκτης (το τι θα βλέπει ο χρήστης). Έχει συγκεκριμένο τρισδιάστατο οπτικό πεδίο στον 3D χώρο. Η κάμερα καταγράφει την προβολή των αντικειμένων που βρίσκονται μέσα στο οπτικό της πεδίο.

3.5. Ο φωτισμός του χώρου

Σε κάθε νέα σκηνή, το Unity τοποθετεί αυτόματα μία πηγή κατευθυνόμενου φωτισμού, ***directional light***. Σε όλες τις σκηνές του παιχνιδιού *The Two Worlds*, έχει χρησιμοποιηθεί το προκαθορισμένο asset φωτός και έχει ρυθμιστεί η θέση του και η έντασή του για να φωτίζει τον 3D χώρο ισότροπα.

Δεν χρησιμοποιήθηκε άλλο είδος πηγής γιατί το Unity παρουσιάζει αδυναμία στη σωστή τοποθέτηση ενός ambient (περιβάλλον) φωτός.

3.6. Η δημιουργία των παραθύρων και μενού

Κάθε παράθυρο, είναι σχεδιασμένο πάνω σε ένα ειδικό asset τύπου καμβά (**Canvas**) που διαθέτει το Unity. Ο καμβάς τοποθετείται αυτόματα στο κέντρο του οπτικού πεδίου της κάμερας. Ο καμβάς όμως δεν έχει μέγεθος για να εφαρμόζει ακριβώς στην ανάλυση κάθε οθόνης.

Σε όλες τις σκηνές του παιχνιδιού *The Two Worlds*, για να ταιριάζει ο καμβάς στις περισσότερες οθόνες (hardware), πρέπει στις ιδιότητες του καμβά να επιλεγούν οι παρακάτω ρυθμίσεις:

- «**Canvas Scaler**» → «**Ui Scale Mode**» → «**Scale with Screen Size**»

- «**Resolution Reference**» → X: 1920 Y: 1080 [21]

Πάνω στον καμβά έχουν τοποθετηθεί διεπαφές – κουμπιά τύπου **Button – TextMeshPro**. Επιλέχθηκε το πακέτο **TextMeshPro** (κεφ. 3.7) γιατί ήταν πιο πλούσιο σε ιδιότητες και δεχόταν Ελληνικές γραμματοσειρές [22]. Ως font, χρησιμοποιήθηκε η γραμματοσειρά **RobotoCondensed-Regular SDF**, η οποία προστέθηκε από εξωτερική πηγή, την ιστοσελίδα *Google Fonts* [23].

Χρησιμοποιήθηκε μία μόνο φωτογραφία για το background του κάθε καμβά και των διεπαφών, η οποία και τροποποιήθηκε σε κάθε σκηνή (πηγή φωτογραφίας: [24]).

Το **TextMeshPro** επιτρέπει να χρωματιστούν τα κουμπιά με material που έχει φτιάξει ο δημιουργός. Για τη γραμματοσειρά του project, χρησιμοποιήθηκε material με χρώμα πράσινο.

3.7. Τα έξτρα βοηθητικά πακέτα του Unity και τρίτα λογισμικά

3.7.1. Τα έξτρα πακέτα εντός του Unity που χρησιμοποιηθήκαν

Cinemachine

Είναι ένα πακέτο που ενισχύει την ρυθμίσεις της κάμερας. Με την ιδιότητα **Follow** μπορεί να ακολουθεί ένα 3D αντικείμενο, π.χ. τον παίκτη. Έχει έτοιμες μεθόδους για να ρυθμιστεί το υψηλότερο, το μεσαίο και το χαμηλότερο σημείο που μπορεί να φτάσει η κάμερα, ακολουθώντας τον παίκτη. Η ρύθμιση αυτή, βοηθάει τον παίκτη να κοιτάει τον γύρω χώρο. Άλλη μέθοδος, δίνει την ιδιότητα στην κάμερα να προσπερνάει αντικείμενα και εμπόδια χωρίς να περνάει από μέσα τους. Για παράδειγμα, δεν μπορεί να μπει μέσα στο έδαφος, σε ένα σπίτι ή σε ένα δέντρο. Δίνεται επίσης η δυνατότητα να τεθεί ένα σημείο εκκίνησης για την κάμερα, όταν ξεκινάει το παιχνίδι. Ακόμη, με την ιδιότητα **Look At**, η κάμερα μπορεί να εστιάζει – κοιτάει ένα συγκεκριμένο σημείο που της ορίζεται, ενώ παράλληλα ακολουθεί τον παίκτη. [25]

FBX Exporter

Το πακέτο αυτό, εξοικονομεί χρόνο για την εξαγωγή αρχείων από το Unity, π.χ. γεωμετρικά σχήματα, μαζί με τα scripts τους, σκηνές, assets

και οτιδήποτε άλλο συσχετίζεται με αυτά, σε μορφοποίηση **FBX**. Αυτή η μορφοποίηση είναι αναγνωρίσιμη από τα λογισμικά *Autodesk Maya*, *Autodesk Maya LT* και *Autodesk 3ds Max* τα οποία χρησιμοποιούν διεπαφές για καλλιτεχνικές δημιουργίες. Τα εξαγόμενα αρχεία **FBX** μπορούν να εισαχθούν στα προηγούμενα λογισμικά και να δεχθούν περαιτέρω επεξεργασία και μορφοποιήσεις. Η διαδικασία μπορεί να γίνει και αντίστροφα. Δηλαδή, αρχεία **FBX** εξαγόμενα από τα προηγούμενα λογισμικά, μπορούν να εισαχθούν στο Unity και να επεξεργαστούν εκεί. [26]

ProBuilder

Με το πακέτο αυτό είναι δυνατή η κατασκευή τρισδιάστατης περίπλοκης γεωμετρίας στο Unity. Το **ProBuilder**, φτιάχνει αυτόματα τον Mesh Collider του τρισδιάστατου αντικειμένου.

Αξιοσημείωτο είναι το γεγονός πως διαθέτει προσχεδιασμένες πόρτες και σκάλες, ακόμη και κυκλικές σκάλες (σπιράλ). Μετά το πρώτο τους «χτίσιμο», μπορούν να δεχθούν περαιτέρω αλλαγές.

Εξελιγμένες ιδιότητες του πακέτου είναι η επεξεργασία της υφής ή χρώματος των εδρών (UV editing) του 3D σχήματος, η παραμόρφωση των ακμών του και η μετατόπιση των κορυφών του.

TextMeshPro

Το **TextMeshPro** ή αλλιώς **TMP**, είναι το εξελιγμένο πακέτο για την επεξεργασία κειμένου, που αντικαθιστά το κλασσικό UI Text του Unity. Είναι εμπλουτισμένο με ιδιότητες μορφοποίησης, στυλ και χρωματισμού κειμένου. Έχει τη δυνατότητα να υποστηρίζει και άλλες γραμματοσειρές που δεν υπάρχουν στο Unity (π.χ. το RobotoCondensed-Regular SDF, κεφ. 3.6). Είναι λιγότερο απαιτητικό σε επιδόσεις απ’ ότι είναι το κλασσικό UI Text του Unity.

Universal Render Pipeline (URP)

Το **URP** είναι ένα πακέτο που στοχεύει στη καλύτερη απόδοση γραφικών σε ένα μεγάλο εύρος πλατφόρμων (από κινητά τηλέφωνα, μέχρι κονσόλες και υπολογιστές). Προσφέρει στο δημιουργό ένα φιλικό περιβάλλον εργασίας, με λύσεις για τον άμεσο ή έμμεσο φωτισμό εντός

της σκηνής του παιχνιδιού, συνδυασμένο με κάποιο material (real – time shadows, Spot και Point Lights, Light Cookies, Reflection Probe Blending).

Το URP προσφέρει ένα νέο σετ από shaders, αυτοί είναι [27]:

- Lit
- Simple Lit
- Baked Lit
- Unlit
- Particles Lit
- Particles Simple Lit
- Particles Unlit
- SpeedTree
- Autodesk Interactive
- Autodesk Interactive Transparent
- Autodesk Interactive Masked

Βασικό μειονέκτημα του πακέτου URP είναι, λόγω της ύπαρξης του shader SpeedTree, η δημιουργία – επεξεργασία ενός δέντρου είναι αρκετά δύσκολη. Αυτό συμβαίνει γιατί στο URP [28]:

- Δεν υπάρχει ενιαίος φωτισμός στα δέντρα (Global Illumination).
- Δεν δημιουργούνται σκιές στα δέντρα.
- Ο χρήστης ρυθμίζει αν ο φωτισμός θα είναι ανά ακμή ή ανά pixel.

Visual Effect Graph (VFX)

Το **VFX** επιτρέπει στον χρήστη να φτιάξει οπτικά εφέ νέας γενιάς όπου το Unity τα φορτώνει στην κάρτα γραφικών. Από αυτό το πακέτο, δημιουργηθήκανε τα εξής εφέ στο παιχνίδι *The Two Worlds*:

- Φωτιά
- Καπνός
- Σπίθες

3.7.2. Τα τρίτα λογισμικά που χρησιμοποιήθηκαν

- *Visual Studio Enterprise 2022*. Χρησιμοποιήθηκε για τη γραφή του κώδικα σε C# των scripts. [29]

- *AUTODESK tinkercad*. Χρησιμοποιήθηκε για τη κατασκευή του 3D βασικού χαρακτήρα. [30]
- *Microsoft Paint 3D*. Χρησιμοποιήθηκε για την επεξεργασία εικόνων για την κατασκευή των textures. Ειδικότερα, χρησιμοποιήθηκε για τη δημιουργία διαφάνειας (**transparency**) σε συγκεκριμένα σημεία ή χρώματα πάνω στην εικόνα και αντίστοιχα στο Texture [31].
- *Microsoft Paint*. Χρησιμοποιήθηκε για την επεξεργασία εικόνων για την κατασκευή των textures. Ειδικότερα, για περικοπή εικόνας, αλλαγή χρωμάτων, προσθήκη σχεδίων κλπ. [32].

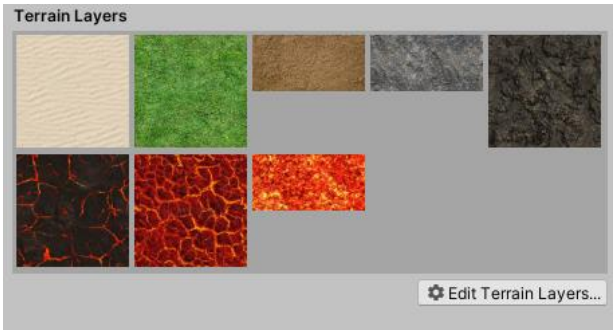
3.8. Δημιουργία 3D περιβάλλοντος σκηνής

Για να δημιουργηθεί ένα 3D περιβάλλον, βασικά στοιχεία είναι η ύπαρξη του εδάφους και ο ουρανός. Το Unity χρησιμοποιεί ειδικά assets, το **terrain** για το έδαφος και το **skybox** για τον ουρανό, για την ευκολότερη και πιο άμεση μορφοποίηση τους.

3.8.1. Terrain

Το terrain είναι ένα επίπεδο αντικείμενο στον τρισδιάστατο χώρο, με ενσωματωμένα εργαλεία που βοηθούν το χρήστη να το διαμορφώσει ώστε να μοιάζει με έδαφος.

Τοποθετείται αυτόματα στο επίπεδο yz της 3D σκηνής. Έχει εργαλεία του στυλ «βούρτσες» (**brushes**), με τα οποία ο χρήστης μπορεί να διαμορφώσει το ανάγλυφο του εδάφους (βουνά, τρύπες), όπως και να το βάψει.



*Εικόνα 22 Η παλέτα των textures του terrain του κόσμου **1**.*

Για το βάψιμο του εδάφους, ο χρήστης μπορεί να δημιουργήσει και να χρησιμοποιήσει μια παλέτα με textures της επιλογής του, ως βαφές, αντί για απλά χρώματα (π.χ. Εικόνα 22 και Εικόνα 23).

*Εικόνα 23 Η παλέτα των textures του terrain του κόσμου **2**.*

Μια ακόμη ιδιότητα του terrain είναι να αναγνωρίζει αντικείμενα που έχουν την ιδιότητα δέντρου ή φυτού και τα τοποθετεί με τη βάση τους ακριβώς πάνω του.

Και τέλος, το terrain έχει το δικό του collider (Terrain Collider), ούτως ώστε τα αντικείμενα που βρίσκονται πάνω του, να μην περνάνε μέσα από το έδαφος με αποτέλεσμα να πέφτουν στο κενό.

3.8.2. Skybox

Το **skybox** είναι ο ουρανός της σκηνής. Τα όρια μιας σκηνής περικλείονται από το skybox.

Με τη δημιουργία μιας νέας σκηνής, το Unity έχει τοποθετήσει ένα προεπιλεγμένο skybox σε αυτή. Στις σκηνές των κόσμων **1** και **2**, έχει χρησιμοποιηθεί το προεπιλεγμένο skybox (τύπου procedural), ενώ στις υπόλοιπες σκηνές, σαν background χρησιμοποιείται μία φωτογραφία (κεφ. 3.6).

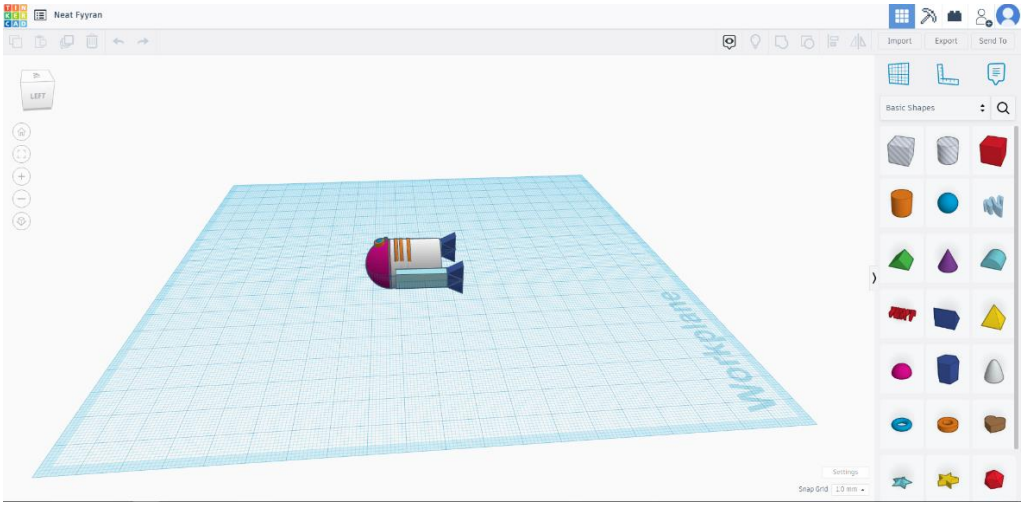
Το skybox είναι ένα τρισδιάστατο αντικείμενο που αποτελείται από την ένωση διαδοχικών textures, σχηματίζοντας έτσι μια ενιαία εικόνα. Μπορεί να έχει τις παρακάτω μορφές [33]:

- Έξι εδρών (6 Sided). Είναι ορθογώνιο παραλληλεπίπεδο, με texture την ένωση 6 διαδοχικών textures, ένα σε κάθε έδρα.
- Κυβικό (Cube Map). Είναι κύβος, με texture την ένωση 6 διαδοχικών textures, ένα σε κάθε έδρα.
- Πανοραμικό (Panoramic). Είναι σφαιρικό, και αποτελείται από μόνο ένα texture.
- Διαδικαστικό (Procedural). Είναι ένα είδος skybox που παράγεται χρησιμοποιώντας τις ιδιότητες ενός material αντί για ένα texture.

3.9. Δημιουργία 3D μοντέλων

3.9.1. Δημιουργία του παίκτη

Ο παίκτης φτιάχτηκε στην ιστοσελίδα της AUTODESK tinkercad της Autodesk (Εικόνα 24) [30].



Εικόνα 24 Η δημιουργία του βασικού παίκτη στην ιστοσελίδα της AUTODESK, www.tinkercad.com

Αποτελείται από έναν συνδυασμό κυλίνδρων, ημισφαιρίων, ορθογωνίων παραλληλεπιπέδων, κ.α. Για να αποδοθεί σωστά η κίνηση του παίκτη και να βαδίζει όρθιος, πρέπει στο σχεδιασμό στην ιστοσελίδα να τοποθετηθεί στο σημείο (0,0,0) και «ξαπλωμένο» όπως φαίνεται στην Εικόνα 24. Τέλος, το 3D αντικείμενο εξάγεται σε μορφή **.obj**, για να αναγνωρίζεται από το Unity.

Μόλις το 3D μοντέλο προστεθεί στο Unity, εκεί αποκτάει τα επόμενα φυσικά χαρακτηριστικά που του δίνουν ζωή: έχει βάρος (ιδιότητα από το Component του **Rigidbody** του Unity), μπορεί να κινείται και να αλλάζει η ταχύτητα του μέσω script, δεν μπορεί να διαπεράσει μέσα από αντικείμενα – εμπόδια. Επίσης, μέσω script, μπορεί να κάνει άλμα.

Στον κόσμο **1**, σε περίπτωση που ο παίκτης απαντήσει λάθος, στην ερώτηση της φωτιάς, τότε βγαίνουν από το κεφάλι του καπνοί. Σε περίπτωση που ο παίκτης απαντήσει την ερώτηση του σεισμού λάθος, τότε, βγαίνουν σπίθες από το κεφάλι του και τα πόδια του αντικαθίστανται με ραγισμένα. Τα νέα ραγισμένα πόδια του δημιουργήθηκαν από γεωμετρικά σχήματα μέσα από το Unity και χρησιμοποιήθηκε material με υφή ραγισμένου. Σε περίπτωση που απαντήσει λάθος την ερώτηση με το ηφαίστειο, το άσπρο σώμα του αντικαθίσταται με κόκκινο σώμα. Το κόκκινο σώμα, επίσης δημιουργήθηκε με γεωμετρικά σχήματα του Unity και material με χρώμα κόκκινο (Εικόνα 25).

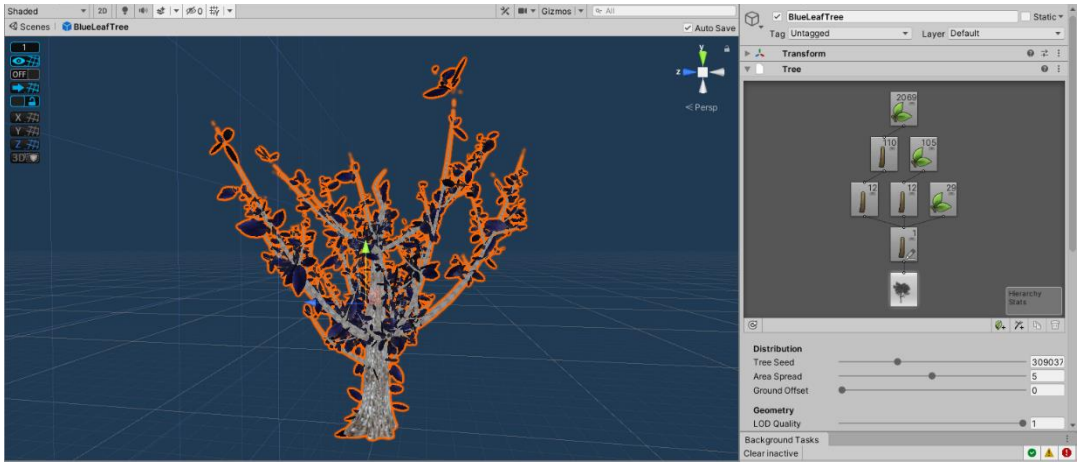
Εικόνα 25 Οι αρνητικές συνέπειες πάνω στον παίκτη σε περίπτωση που απαντήσει λάθος όλες τις ερωτήσεις του πρώτου κόσμου.

3.9.2. Η δημιουργία δέντρων και κλαδιών

Τα δέντρα δημιουργήθηκαν με το προκατασκευασμένο 3D αντικείμενο *Tree* που υπάρχει ήδη στο Unity. Σε αυτό το αντικείμενο είναι ενσωματωμένο ένα σετ από ρυθμίσεις με τις οποίες είναι εφικτό να ρυθμιστούν ο κορμός ενός δέντρου, τα κλαδιά ενός δέντρου, τα κλαδιά των κλαδιών, κ.λπ. Με τις ίδιες ρυθμίσεις, πάνω στα κλαδιά μπορούν να προστεθούν φύλλα.

Τα αντικείμενα τύπου Tree του Unity υπακούν στους νόμους της φύσης, για παράδειγμα, λυγίζουν με το φύσημα του αέρα. Βέβαια, στους κόσμους του παιχνιδιού *The Two Worlds* δεν έχει προστεθεί τέτοιος μηχανισμός.

Το URP προκάλεσε πολλά προβλήματα σε όλα τα materials των δέντρων. Γι' αυτό, σε πρώτη φάση, τα δέντρα φτιαχτήκαν με το πακέτο URP ανενεργό. Στη συνέχεια, μετά την ενεργοποίηση του URP, τα materials προσαρμόστηκαν χειροκίνητα σε κάποιο URP shader.



Εικόνα 26 Στιγμιότυπο κατασκευής ενός από τα δέντρα του παιχνιδιού The Two Worlds, με το πακέτο URP ανενεργό.

Ένα παράδειγμα δημιουργίας ενός από τα δέντρα του παιχνιδιού *The Two Worlds*, φαίνεται στην Εικόνα 26. Στα δεξιά της εικόνας, φαίνεται η συσχέτιση των κορμών και των κλαδιών. Κάτω κάτω έχουμε (τον έναν) κορμό. Ακριβώς από πάνω, είναι 2 ομάδες κλαδιών και τα φύλλα που ανήκουν στον κορμό. Πάνω από τις 2 ομάδες κλαδιών, υπάρχει άλλη μια υποομάδα κλαδιών καθώς και τα φύλλα της μίας από τις δύο προηγούμενες ομάδες κλαδιών. Τέλος, στην κορυφή, υπάρχουν τα φύλλα της προηγούμενης υποομάδας κλαδιών. Λόγω του μεγάλου πλήθους κλαδιών σε αυτήν την υποομάδα, αποτέλεσμα είναι να της ανήκουν 2069 φύλλα.

Σε αυτό το σημείο, το game engine Unity, απορροφά πολλούς πόρους από το υπολογιστικό σύστημα για να καταφέρει να σχεδιάσει όλα αυτά τα φύλλα μαζί με τα textures τους.

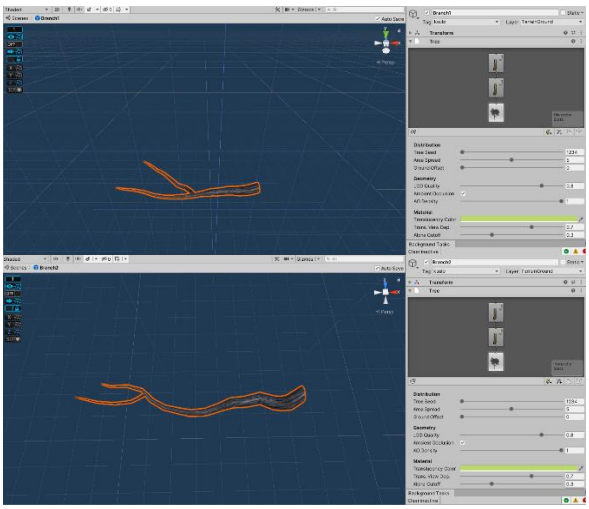
Εικόνα 27 Παράδειγμα του texture ενός φύλλου. Το μαύρο χρώμα αντιστοιχίζεται στην αόρατη περιοχή.

Τα φύλλα αποτελούνται από 2D επίπεδα στα οποία προστίθεται ειδικό texture, το οποίο στη μέση του έχει το σχήμα και το χρώμα του φύλλου. Γύρω από το φύλλο, το texture είναι transparent (αόρατο). Ένα παράδειγμα texture ενός φύλλου, φαίνεται στην Εικόνα 27. Το μαύρο χρώμα της εικόνας, αντιστοιχίζεται στην αόρατη περιοχή. Η ιδιότητα της διαφάνειας φτιάχτηκε με τη βοήθεια του Microsoft Paint 3D.

Με παρόμοιο τρόπο σχεδιαστήκαν όλα τα δέντρα του παιχνιδιού *The Two Worlds* και παρουσιάζονται στην Εικόνα 28.

*Εικόνα 28 Όλα τα δέντρα που φτιάχτηκαν στον κόσμο **1** και **2** του The Two Worlds.*

Με μεγαλύτερη ευκολία, σχεδιάστηκαν και τα κλαδιά που αναζητεί ο παίκτης στον κόσμο **2** στο έδαφος, με τη διαφορά ότι, δεν υπάρχουν πάνω τους φύλλα και σαν κύριο σώμα κλαδιού χρησιμοποιήθηκε ο κορμός στην ιδιότητα Tree. Τα κλαδιά του κόσμου **2** φαίνονται στην Εικόνα 29.

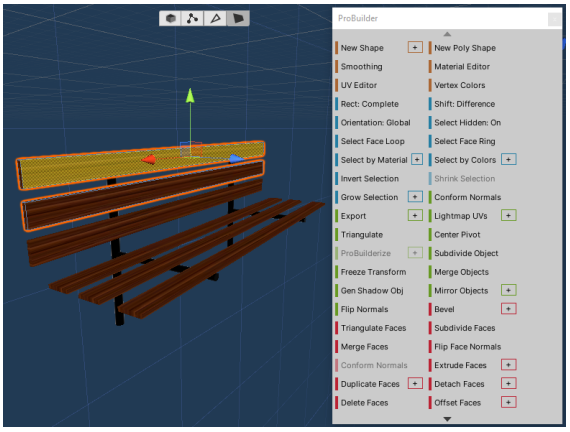


*Εικόνα 29 Τα κλαδιά που πρέπει να μαζέψει ο παίκτης στον κόσμο **2**.*

3.9.3. Η δημιουργία των υπόλοιπων 3D μοντέλων

Όλα τα 3D μοντέλα των κόσμων 1 και 2, φτιάχτηκαν με το έξτρα πακέτο του Unity, ProBuilder (κεφ. 3.7.1).

Ακολουθεί παράδειγμα χρήσης του ProBuilder για το πως φτιάχτηκε το παγκάκι στον κόσμο **1**.



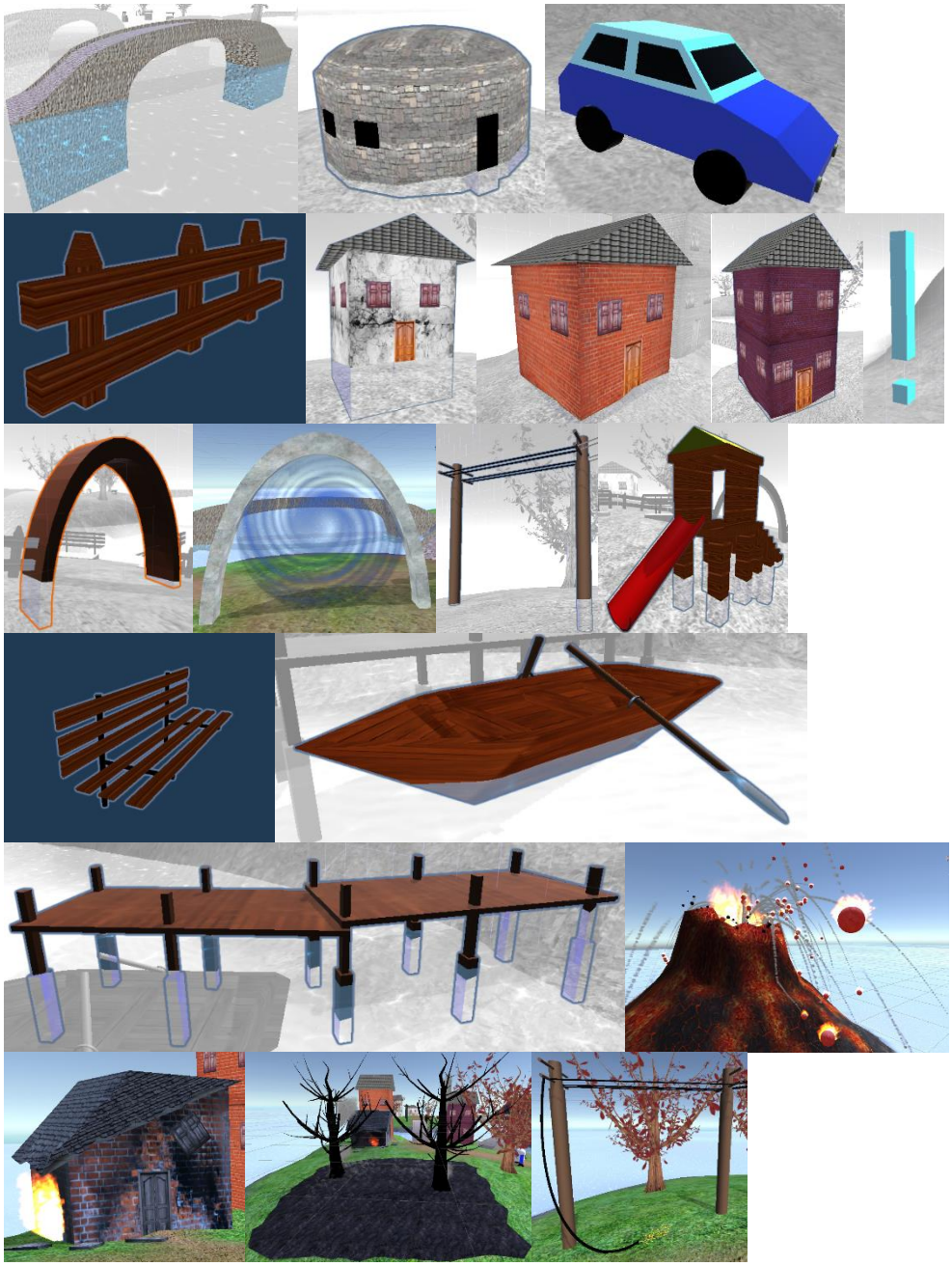
*Εικόνα 30 Η δημιουργία για το παγκάκι στον κόσμο **1**, με το ProBuilder.*

Αρχικά, με την επιλογή **new shape** στο ProBuilder, φτιάχτηκε ένα ορθογώνιο παραλληλεπίπεδο. Αυτό το ορθογώνιο, «βάφτηκε» με ένα material τύπου URP. Κάθε έδρα του βάφτηκε ξεχωριστά προσαρμόζοντας το material πάνω σε αυτή και όχι σε ολόκληρο το 3D αντικείμενο.

Με την ολοκλήρωση της πρώτης τάβλας, εκτελείται το Duplicate που μας παρέχει το Unity, και δημιουργούνται άλλα 5 αντίγραφα. Τοποθετούνται 2 αντίγραφα σε τέτοιο μέρος, ώστε να δημιουργηθεί τριάδα για την πλάτη του καθίσματος. Οι υπόλοιπες τάβλες, τοποθετούνται παράλληλα στο έδαφος, ώστε να δημιουργηθεί το υπόλοιπο κάθισμα.

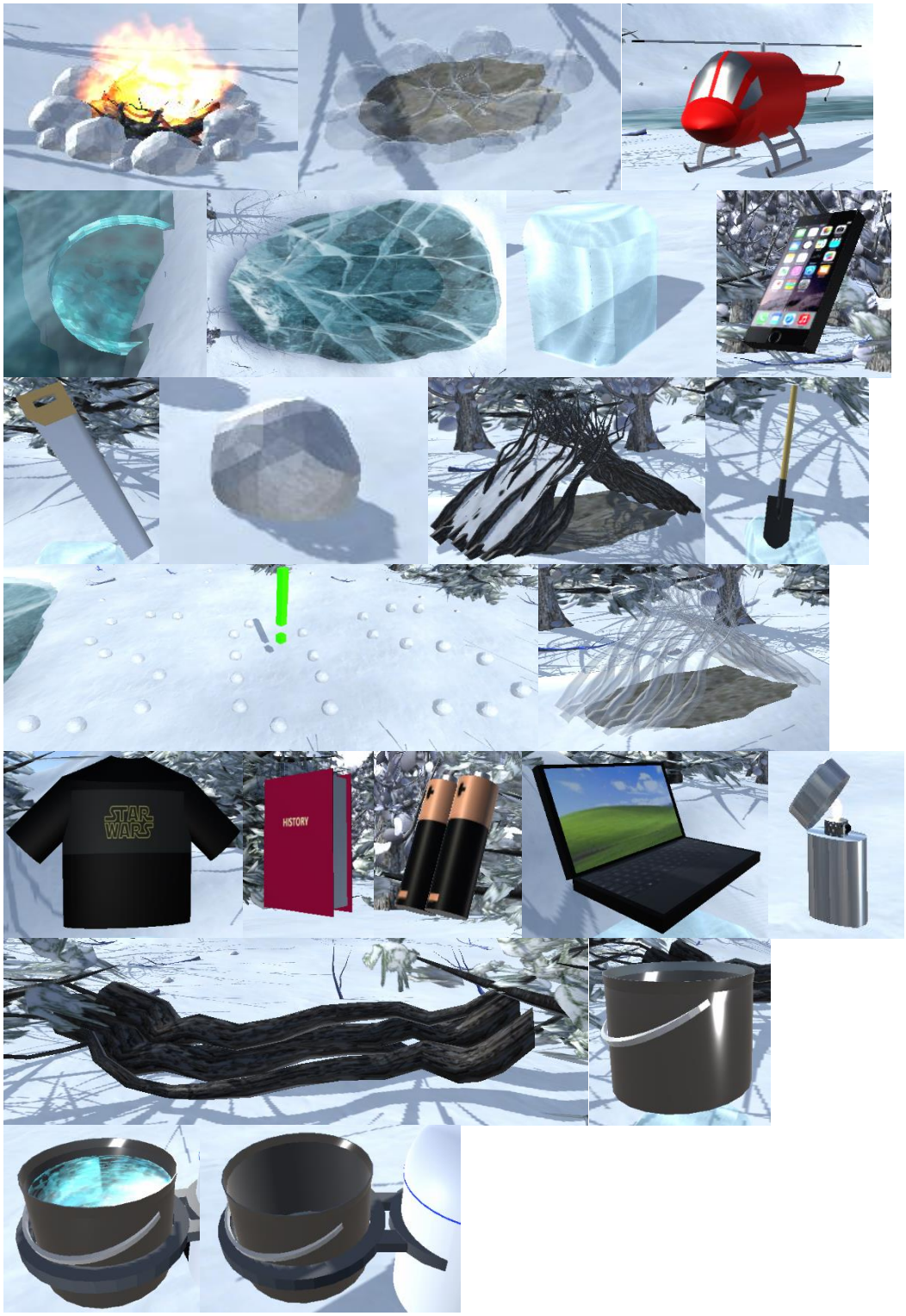
Στη συνέχεια, με την ίδια επιλογή new shape στο ProBuilder, δημιουργούνται 2 κύλινδροι που θα στηρίζουν τη μία μεριά του καθίσματος. Αυτά θα είναι τα μεταλλικά στηρίγματα για το παγκάκι. Ο ένας κύλινδρος είναι όρθιος (90^{o} από τη γη), ενώ ο άλλος τοποθετείται παράλληλα στη γη. «Βάφονται» με material που έχει υφή μετάλλου (μαύρο χρώμα, με λάμψη) και γίνονται Duplicate για να «στηρίζουν» το παγκάκι και στην άλλη μεριά του.

Στην Εικόνα 31 παρουσιάζονται όλα τα υπόλοιπα 3D μοντέλα που δημιουργηθήκαν με το ProBuilder, για τον κόσμο **1**. Σε κάποια από αυτά τα αντικείμενα έχει προστεθεί κίνηση ή εφέ (θαυμαστικά, μαγική πόρτα, βάρκα) που θα περιγραφεί στο επόμενο κεφ. 3.11.



*Εικόνα 31 Τα 3D μοντέλα του κόσμου **1**.*

Στην Εικόνα 32 παρουσιάζονται όλα τα 3D μοντέλα που δημιουργηθήκαν με το ProBuilder, για τον κόσμο **2**. Σε κάποια από αυτά τα αντικείμενα έχει προστεθεί κίνηση ή εφέ: φωτιά, σπασμένος πάγος, ελικόπτερο, στύλος πάγου, ο κουβάς με το νερό και τα 9 αντικείμενα στην περιοχή **1** του κόσμου **2** (Εικόνα 12). Τα εφέ και η κίνησή τους θα περιγραφούν στο επόμενο κεφ. 3.11.



*Εικόνα 32 Τα 3D μοντέλα του κόσμου **2**.*

Σημείωση: Το URP προκάλεσε πολλά προβλήματα σε όλα τα materials των 3D μοντέλων. Οπότε, κάθε material, έπρεπε χειροκίνητα να προσαρμοστεί σε κάποιο shader που να ανήκει στο URP (κεφ. 3.7.1).

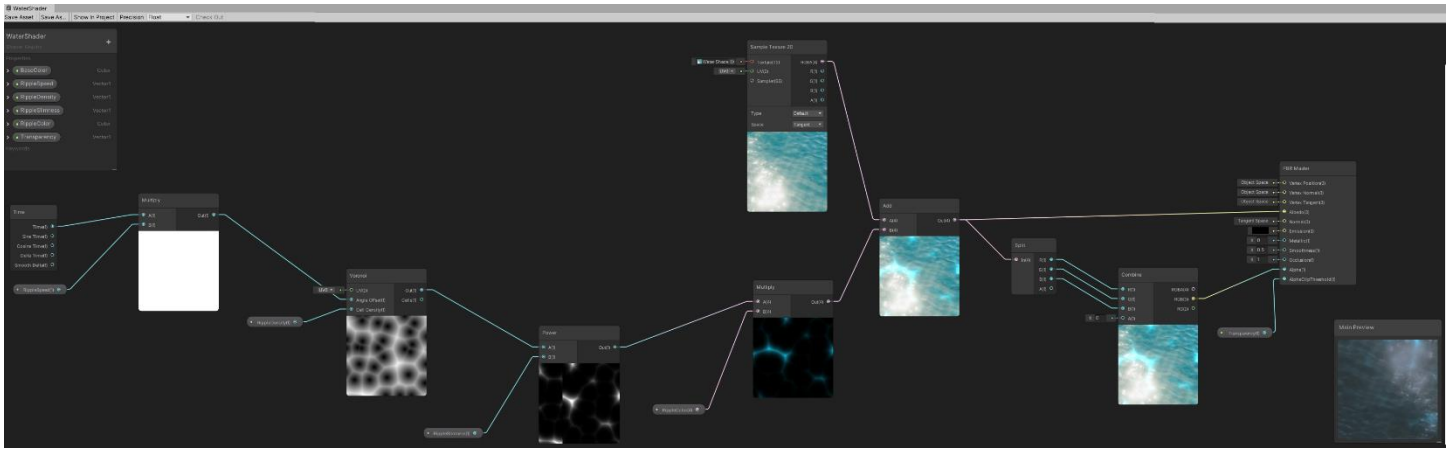
3.10. Ειδικά εφέ που χρησιμοποιηθήκαν για το παιχνίδι

Στους κόσμους **1** και **2** προστέθηκαν τα εφέ:

- Η κίνηση της θάλασσας
- Τα εφέ στη μαγική πόρτα
- Ο πάγος που λιώνει στις βάσεις με τα 9 αντικείμενα στον κόσμο **2**
- Σπίθες
- Φωτιά
- Καπνός
- Μπάλες ηφαιστείου
- Σύννεφα
- Σεισμός

Κίνηση της θάλασσας

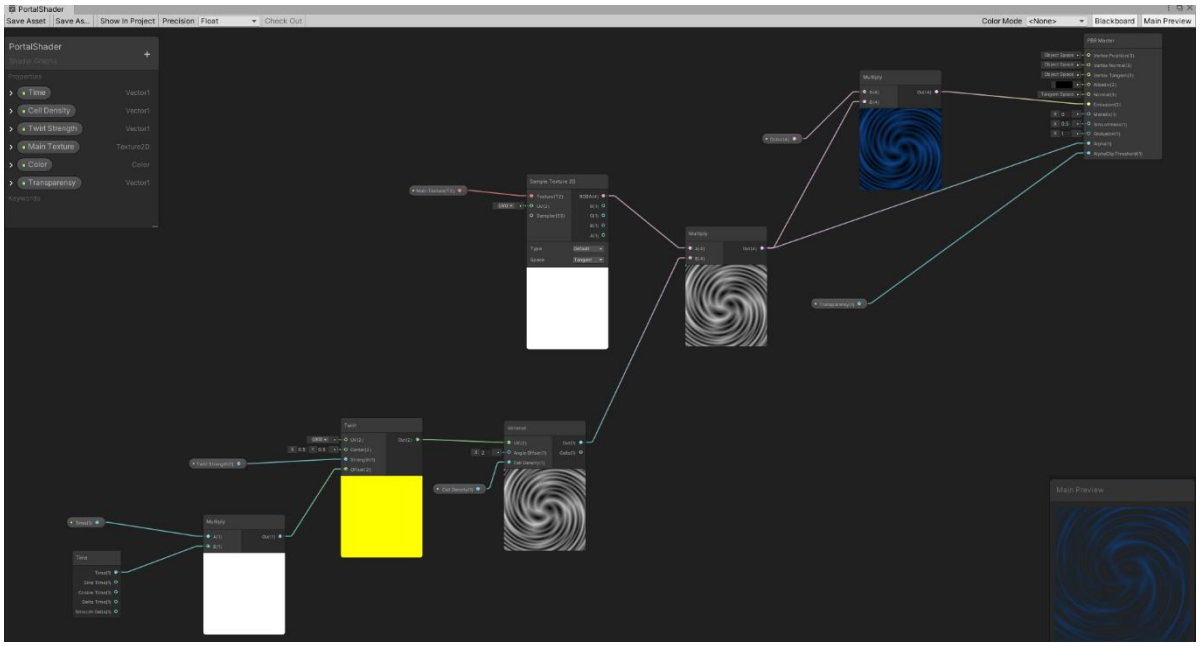
Για την θάλασσα, αρχικά χρησιμοποιήθηκε ένα 2D επίπεδο. Πάνω σε αυτό το επίπεδο τοποθετήθηκε material που φτιάχτηκε με τη βοήθεια του **Shader Graph**, όπως φαίνεται στην Εικόνα 33. Στον γράφο, χρησιμοποιήθηκε ως texture 2D ένα στιγμιότυπο από μια επιφάνεια θάλασσας. Συνδυασμένο με τον κόμβο **Voronoi** και τον κόμβο του χρόνου **Time**, δημιουργήθηκαν τα κύματα πάνω στο texture. Το Voronoi είναι ένα είδους θορύβου, που μοιάζει με κουκίδες που κινούνται στον διαθέσιμο χώρο που υπάρχει γύρω τους. Ο κόμβος του χρόνου (Time) είναι τιμές που μεταβάλλονται συναρτησιακά με το χρόνο. Η μεταβολή των τιμών αυτών δίνουν την κίνηση των κουκίδων στο θόρυβο Voronoi. Χωρίς τον κόμβο του χρόνου δεν μπορούμε να έχουμε κίνηση στο εφέ. Στον τελικό κόμβο του γράφου, αποδόθηκε διαφάνεια πάνω στο texture.



*Εικόνα 33 Το shader graph για το material με εφέ, για την θάλασσα του κόσμου **1**.*

Το εφέ στη μαγική πόρτα

Για την μαγική πόρτα, όμοια με τη θάλασσα, αρχικά χρησιμοποιήθηκε ένα 2D επίπεδο. Πάνω σε αυτό το επίπεδο τοποθετήθηκε material που φτιάχτηκε με τη βοήθεια του **Shader Graph**, όπως φαίνεται στην Εικόνα 34. Στον γράφο, χρησιμοποιήθηκε ως texture 2D ένα απλό λευκό χρώμα. Το texture συνδυάστηκε με τον κόμβο **Twirl**, **Voronoi** και **Time**. Ο κόμβος Twirl, είναι ένα σπιράλ πάνω σε επίπεδο. Ο κόμβος Time, έδωσε κίνηση στο σπιράλ για να περιστρέφεται στο επίπεδο αυτό. Ο κόμβος Voronoi, πρόσθεσε μερικές παραμορφώσεις και κινήσεις πάνω στους έλικες του σπιράλ. Στη συνέχεια, με τον κόμβο color, δόθηκε το χρώμα μπλε. Στον τελικό κόμβο του γράφου, αποδόθηκε διαφάνεια πάνω στο texture του περιστρεφόμενου σπιράλ.



*Εικόνα 34 Το shader graph για το material με εφέ, για τη μαγική πόρτα των κόσμων **1** και **2**.*

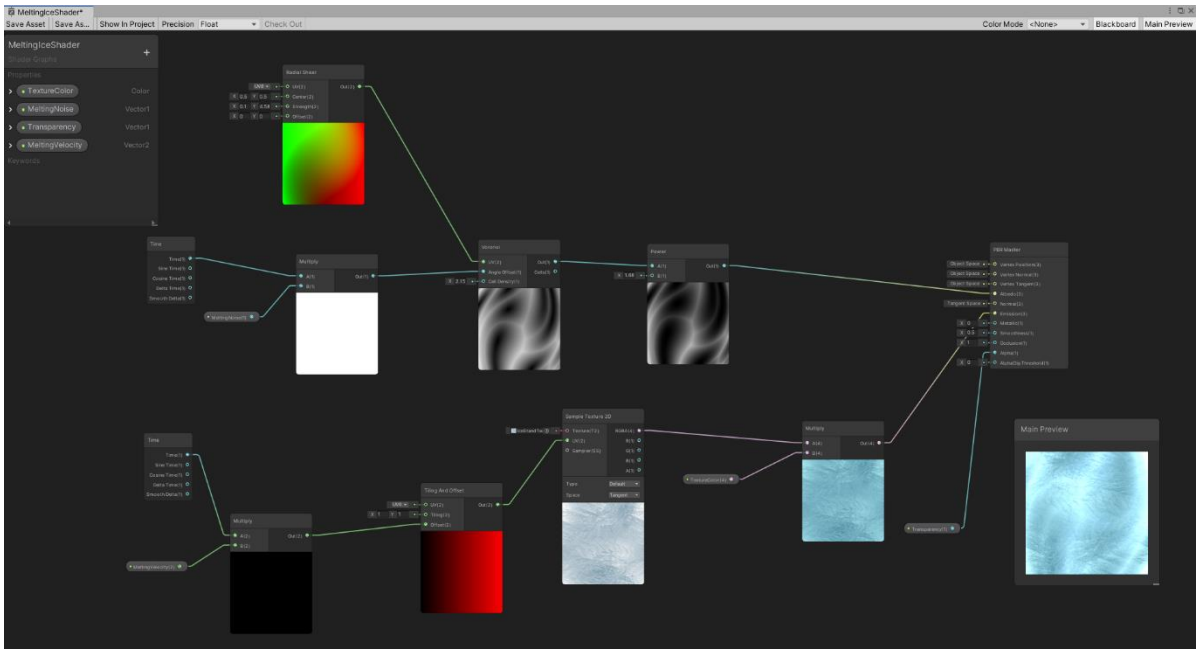
Ο πάγος που λιώνει στις βάσεις με τα 9 αντικείμενα στον κόσμο 2

Σε όλους τους παγωμένους στύλους, κάτω από τα 9 αντικείμενα στον κόσμο **2**, έχει προστεθεί εφέ πάγου που λιώνει. Το εφέ είναι ανεπαίσθητο όπως θα συνέβαινε σε έναν πραγματικό πάγο στα χιόνια.

Το εφέ δημιουργήθηκε με τη βοήθεια ενός **Shader Graph** (Εικόνα 35). Χρησιμοποιήθηκε 2D texture με την υφή πάγου. Συνδυασμένο με τους κόμβους **Time**, **MeltingVelocity** και **Tiling And Offset**[4] δημιουργήθηκε η κίνηση του texture προς στον κατακόρυφο άξονα.

Για την παραμόρφωση του πάγου, χρησιμοποιήθηκε ο θόρυβος **Voronoi** (κινούμενες κουκίδες) συνδυασμένος με το εφέ **Radial Shear** (ακτινική παραμόρφωση που ξεκινάει από το κέντρο του UV texture του γράφου, μοιάζει με ρουφήχτρα) και τον χρόνο **Time**.

Τέλος, στον τελικό κόμβο προστέθηκε διαφάνεια στο συνολικό υλικό.



*Εικόνα 35 Το shader graph για το material με εφέ του πάγου που λιώνει, στους στύλους με τα 9 αντικείμενα στον κόσμο **2**.*

[4] Tiling And Offset είναι σαν να λέμε το πλήθος αντιγράφων ενός texture, πάνω σε μία επιφάνεια.

Σπίθες

Με την εγκατάσταση του πακέτου **Visual Effect Graph (VFX)**, εγκαταστάθηκε και ένας φάκελος με δείγματα κάποιων εφέ. Ένα από αυτά ήταν και οι σπίθες (sparks).

Φωτιά

Όλα τα εφέ φωτιάς δημιουργήθηκαν με το πακέτο **Visual Effect Graph (VFX)**, με βάσει τις οδηγίες από ένα ερασιτεχνικό κανάλι στο youtube [34]. Αλλαγές έγιναν στο μέγεθος και στην γωνία εκπομπής της φωτιάς σε κάθε σημείο που χρησιμοποιήθηκε η φωτιά. Το texture που χρησιμοποιήθηκε, λήφθηκε από την ακόλουθη πηγή [35]. Στο texture αυτό χρειάστηκε να προστεθεί διαφάνεια με τη βοήθεια του **3D Paint**.

Καπνός

Όλα τα εφέ καπνού δημιουργήθηκαν με το πακέτο **Visual Effect Graph (VFX)**, με βάσει τις οδηγίες από το επίσημο κανάλι του Unity στο youtube [36]. Η λογική για τη δημιουργία του καπνού, είναι η ίδια με αυτή για τη φωτιά. Αλλαγές έγιναν στο μέγεθος και στην γωνία εκπομπής του καπνού σε κάθε σημείο που χρησιμοποιήθηκε καπνός.

Μπάλες Ηφαιστείου

Τα εφέ για τις μπάλες φωτιάς είναι τα εξής: η φωτιά και ο καπνός σε όλες τις στιγμές. Τη στιγμή της κρούσης της οποιασδήποτε μπάλας, με το έδαφος, ενεργοποιούνται τα εφέ έκρηξης, τα οποία δημιουργήθηκαν με τη βοήθεια της ακόλουθης πηγής [37]. Λεπτομέρειες για το πώς ενεργοποιούνται, περιγράφονται αναλυτικά στο κεφάλαιο με τα scripts, 3.11.3/Scripts για τα εφέ του Ηφαιστείου.

Σύννεφα

Τα σύννεφα δημιουργήθηκαν με το βασικό **Particle System** του Unity. Η πηγή εκπομπής έχει σχήμα Donut και περιβάλει όλη την περιοχή που μπορεί να κινηθεί ο παίκτης στον κόσμο **2** (Εικόνα 12). Το texture που χρησιμοποιήθηκε, δημιουργήθηκε στο ***Microsoft Paint***, και προστέθηκε διαφάνεια στο ***Microsoft 3D Paint***.

Σεισμός

Για τη δημιουργία σεισμού, στον κόσμο **1**, στην ερώτηση με τον σεισμό έχει προστεθεί ένα **Component** με όνομα: **Cinemachine Impulse Source**. Όταν ο παίκτης εκκινήσει την ερώτηση του σεισμού, ξεσπάει ο σεισμός με επίκεντρο το θαυμαστικό στο κέντρο του νησιού.

Για τη δημιουργία σεισμού, στην ερώτηση με τον ηφαίστειο έχει προστεθεί ένα **Component** με όνομα: **Cinemachine Impulse Source**. Όταν ο παίκτης εκκινήσει την ερώτηση του ηφαιστείου, ξεσπάει ο σεισμός με επίκεντρο το θαυμαστικό στο κέντρο του νησιού.

Η ενεργοποίηση των 2 διαφορετικών σεισμών αναλύεται στο κεφάλαιο των scripts 3.11.3/Ερώτηση Σεισμού και Ερώτηση Ηφαιστείου. [38]

3.11. Scripts

Σε αυτό το κεφάλαιο θα αναλυθούν οι κώδικες εντός των scripts, όπως και οι συνδέσεις – σειρά μεταξύ τους. Θα αναλυθούν πρώτα τα scripts τα οποία δουλεύουν ή χρειάζονται για ολόκληρη την ανάπτυξη του παιχνιδιού. Στη συνέχεια θα αναλυθούν τα script που απαιτούνται για την ομαλή λειτουργία του βασικού μενού. Με τη σειρά τους, τα scripts του **1ου** και του **2ου** κόσμου. Και τέλος το script της τελικής σκηνής με το σκορ του παίκτη.

3.11.1. Γενικά Scripts

Τα scripts που είναι όμοια στην σκηνή **1**, στην σκηνή **2** ή εξυπηρετούν άλλους σκοπούς, είναι τα επόμενα:

- ClearDevLog.cs
- CustomSceneController.cs
- PlayerEscapeMenu.cs
- RotateQuestMarker.cs
- CameraRotationRightClick.cs
- ThirdPersMove.cs

Ανάλυση γενικών scripts

ClearDevLog.cs

Το script αυτό χρησιμοποιείται από τον δημιουργό για να καθαρίζει την κονσόλα μηνυμάτων του Unity, ώστε να υπάρχει ευκρίνεια του **debugging** και να μην προκαλείται σύγχυση.

Αυτό που κάνει εσωτερικά του script, είναι να δημιουργεί μια μεταβλητή τύπου assembly, να τη «γεμίζει» με τα δεδομένα της κονσόλας, δηλαδή τα προηγούμενα κείμενα, **Logs** ή **Warnings** που είχανε μείνει, και στην συνέχεια τα αδειάζει.

Το script διαθέτει μία συνάρτηση τύπου "**public static void**", **clearDeveloperLog()**. Αυτή η συνάρτηση είναι:

- "**Public**" για να μπορεί ο δημιουργός να την καλέσει γενικά από την ίδια την κλάση – script.
- "**Static**" για να είναι ορατή από όλα τα scripts του Project.
- Και τέλος "**void**" γιατί δεν επιστρέφει κάποιον τύπο δεδομένων, απλά «καθαρίζει» την κονσόλα.

Σημείωση: το εσωτερικό κομμάτι της συνάρτησης **clearDeveloperLog()** είναι εντός σχολίων γιατί τη στιγμή του «χτισίματος» (**Build**) του Project, το Unity παρουσιάζει σφάλμα και δεν επιτρέπει τη τελική δημιουργία των εκτελέσιμων αρχείων του παιχνιδιού. Οπότε, όταν ο δημιουργός αποφασίσει πως χρειάζεται η κονσόλα να «καθαρίζεται», απενεργοποιεί τα σχόλια του εσωτερικού κομματιού της συνάρτησης **clearDeveloperLog()** και την καλεί κατά βούληση. [39]

CustomSceneController.cs

Το script αυτό δημιουργήθηκε για περαιτέρω ευχρηστία στην μετάβαση μεταξύ σκηνών. Τη στιγμή που το Project γίνεται **Build**, το Unity κωδικοποιεί όλες τις υπάρχουσες σκηνές που έχει ορίσει ο χρήστης να είναι ενεργοποιημένες. Αυτές οι σκηνές είναι κωδικοποιημένες με έναν μοναδικό ακέραιο αριθμό.

Για τη διευκόλυνση του δημιουργού, τη στιγμή που αποφασίσει να ενσωματώσει στον κώδικα μία εντολή μετάβασης σκηνής, το μόνο που απαιτείται είναι αυτός ο μοναδικός ακέραιος αριθμός.

Όμως, κατά τη διαδικασία του **Build**, υπάρχει η πιθανότητα να αλλαχτεί η σειρά των σκηνών, οπότε και να αλλάξει ο μοναδικός αυτός αριθμός. Το πρόβλημα που εμφανίζεται εδώ λοιπόν, είναι πως ο δημιουργός πρέπει να ψάξει όλα τα script του Project και να αλλάξει τον αριθμό της σκηνής στον νέο μοναδικό ακέραιο αριθμό.

Με το script **CustomSceneController.cs**, δεν χρειάζεται ο δημιουργός να αναζητήσει τις εντολές μετάβασης σκηνών και να αλλάξει τον ακέραιο αριθμό. Απλά, προαπαιτούμενο είναι, να ενημερώνεται η κλάση – script **CustomSceneController.cs,** με τους νέους ακέραιους αριθμούς που αντιστοιχούνται σε κάθε σκηνή. Η αντιστοίχιση αυτή φαίνεται ξεκάθαρα τη στιγμή που ο δημιουργός κάνει **Build** στο Project.

Σημείωση 1: Όταν καλεί ο δημιουργός τη συνάρτηση «**SceneManager.LoadScene([…])**;», αντί να προσθέσει τον ακέραιο αριθμό της σκηνής που επιθυμεί να μεταβεί ο παίκτης σε εκείνο το στάδιο, εντός των παρενθέσεων, θα τοποθετήσει την εντολή " **CustomSceneController.get** …" και θα επιλέξει την αντίστοιχη σκηνή.

Σημείωση 2: Οι μεταβλητές σε αυτή τη κλάση είναι όλες "**private**" για να μην μπορούν να είναι ορατές έξω από αυτή, με πιθανό αποτέλεσμα να «πειραχτούν» από τρίτους. Μπορούν να αξιοποιηθούν μόνο με τη κλήση των αντίστοιχων «**public static**» μεθόδων τους.

PlayerEscapeMenu.cs

Αυτό το script διαχειρίζεται το κουμπί "**Esc**" του παίκτη στον κόσμο **1** και στον κόσμο **2**, ώστε να κάνει παύση στο παιχνίδι και να του εμφανιστεί το παράθυρο παύσης. Έχουν οριστεί τα 2 αντικείμενα τύπου **GameObjects**[5]:

- Το πρώτο GameObject είναι το **EscapeWindow**, το οποίο είναι το κύριο παράθυρο παύσης. Σε αυτό το παράθυρο, ο παίκτης έχει την επιλογή να επιστρέψει ξανά στο παιχνίδι, να επιστρέψει στο κύριο μενού ή να αποχωρήσει τελείως από το παιχνίδι.
- Το δεύτερο GameObject είναι το **exitQuestionWindow**, το οποίο είναι ένα μικρότερο παράθυρο, εμφανίζεται μόνο αν ο παίκτης πατήσει το κουμπί «Έξοδος» και ρωτά τον παίκτη αν επιθυμεί να αποχωρήσει από το παιχνίδι.

[5] **GameObject** είναι μία κλάση, που κληρονομούν από αυτή όλα τα αντικείμενα (3D ή 2D), μέσα σε μια σκηνή του Unity.

Με τις επόμενες 2 μεταβλητές τύπου "**bool**" που εμφανίζονται, επιτυγχάνεται ο έλεγχος για το αν το παιχνίδι *The Two Worlds* έχει κάνει παύση επειδή έχει εμφανιστεί στον παίκτη παράθυρο κάποιας ερώτησης ή αν το παιχνίδι έχει κάνει παύση από τον ίδιο τον παίκτη.

Αν αφαιρεθούν αυτές οι 2 μεταβλητές, σε περίπτωση που το παιχνίδι έχει κάνει παύση επειδή ο παίκτης απαντά σε κάποια ερώτηση, θα παρουσιαστεί πρόβλημα αν πατήσει ο παίκτης 2 φορές το κουμπί "**Esc**".

Το πρώτο πρόβλημα που θα παρουσιαστεί είναι πως θα εμφανιστεί το βασικό παράθυρο **EscapeWindow** πάνω στο παράθυρο της ερώτησης, την πρώτη φορά που θα πατηθεί το "**Esc**", το οποίο είναι αντιαισθητικό. Το δεύτερο πρόβλημα είναι πως με το δεύτερο πάτημα του ίδιου κουμπιού ή του κουμπιού «Συνέχεια», το παράθυρο **EscapeWindow**, θα εξαφανιστεί, το παιχνίδι δεν θα είναι σε παύση πια, με αποτέλεσμα ο παίκτης να κινείται ακόμη, και το παράθυρο της ερώτησης να παραμένει ακόμη ανοικτό χωρίς να έχει απαντήσει στην ερώτηση. Για αυτό λοιπόν, με τις 2 αυτές "**bool**" μεταβλητές διαχωρίζεται το είδος της παύσης.

Χάρης σε αυτές τις 2 μεταβλητές, ο παίκτης δεν μπορεί να έχει ένα ανοικτό παράθυρο ερώτησης, να κινείται και να κατευθύνεται σε άλλη ή άλλες ερωτήσεις και να ενεργοποιήσει και το παράθυρο εκείνων των ερωτήσεων πάνω στο ήδη υπάρχον παράθυρο ερώτησης.

Επίσης, η μεταβλητή "**gameisPaused**", έχει πρακτική λειτουργία για τον δεύτερο κόσμο. Αν ο παίκτης κάνει παύση, δηλαδή η μεταβλητή αυτή γίνεται "**true**" και εμφανίζεται το παράθυρο **EscapeWindow**, ο παίκτης δεν μπορεί να σηκώσει αντικείμενα από το έδαφος.

Δύο ακόμη λειτουργίες που υπάρχουν μέσα σε αυτό το script, είναι σε μορφή 3 συναρτήσεων.

Η μία λειτουργία είναι μέσα στην συνάρτηση **goToMainMenu**(), όπου ο παίκτης, δεν μεταβαίνει μόνο στη σκηνή του κύριου μενού, αλλά αρχικοποιεί τους 2 κόσμους, ώστε όταν και αν, ξανά περιηγηθεί ο παίκτης μέσα στους κόσμους **1** ή **2**, να είναι οι κόσμοι στις αρχικές τους καταστάσεις – οι ερωτήσεις να είναι αναπάντητες και έτοιμες προς απάντηση.

Η άλλη λειτουργία είναι μέσα στις συναρτήσεις **pauseGame**() και **resumeGame**(), οι οποίες καλούνται από οπουδήποτε μέσα στο Project λόγω του τύπου τους "**public static**". Αυτές οι συναρτήσεις υλοποιούνται μία φορά εδώ και επαναχρησιμοποιούνται συνεχώς κατά τη διάρκεια του παιχνιδιού, χωρίς να ξανά γράφονται.

RotateQuestMarker.cs

Αυτό το script, περιστρέφει αντικείμενα γύρω από τον εαυτό τους. Χρησιμοποιείται για να περιστρέφει όλα τα θαυμαστικά που υπάρχουν μέσα στο παιχνίδι.

Σημείωση : Το Unity από μόνο του, όταν υπάρχουν εντολές όπως "**Transform**" (κίνηση) ή "**Rotate**" (περιστροφή), οι ταχύτητες κίνησης ή περιστροφής αντίστοιχα, ακολουθούν τον αριθμό καρέ που ανανεώνεται η υπάρχουσα οθόνη ανά δευτερόλεπτο (Frames Per Second). Επειδή υπάρχουν πολλά ήδη συχνοτήτων οθονών, πολλαπλασιάζουμε τον αντίστοιχο τρισδιάστατο πίνακα με τη μεταβλητή **Time.deltaTime** για να κινούνται και να περιστρέφονται τα αντικείμενα με την ίδια ταχύτητα σε συνάρτηση με τον χρόνο.

CameraRotationRightClick.cs

Αυτό το script είναι υπεύθυνο για την λειτουργία της κάμερας γύρω από τον παίκτη στον κόσμο **1** και **2**. Όταν μετατοπίζεται το ποντίκι, το πακέτο **Cinemachine** που παρέχεται από το Unity, εντοπίζει την κίνηση και την αποθηκεύει στις μεταβλητές "**Mouse X**" "**Mouse Y**", αντίστοιχα.

Στη συνέχεια, για να μοιάζει ο χειρισμός της κάμερας περισσότερο με τα σύγχρονα βιντεοπαιχνίδια που κυκλοφορούν, επιτρέπεται να γυρίσει η κάμερα μόνο όταν ο παίκτης πατήσει δεξί κλικ [40].

Τέλος, το script αυτό, κάθε φορά που πατιέται το δεξί κλικ εντός των κόσμων **1** και **2**, εξαφανίζει τον κέρσορα (**mouse cursor**) για να υπάρχει μια πιο καθαρή εικόνα. Όταν απελευθερώνεται το δεξί κλικ, το script επαναφέρει τον κέρσορα ξανά στην μέση της οθόνης. [41]

ThirdPersMove.cs

Αυτό το script είναι από τα πιο σημαντικά scripts του παιχνιδιού. Σε αυτό το script, πραγματοποιείται η κίνηση του παίκτη. Με τις εξής μεταβλητές: **speed**, **gravity**, **jumpHeight**, επιτυγχάνεται η κίνηση του παίκτη και το άλμα του. Το **speed** ορίζει την βασική ταχύτητα του παίκτη, το **gravity** τη δύναμη της βαρύτητας και το **jumpHeight** το πόσο ψηλά μπορεί να κάνει άλμα ο παίκτης [42].

Οι εντολές **Input.GetAxisRaw("Vertical")**; και **Input.GetAxisRaw("Horizontal")**; ορίζουν την κίνηση του παίκτη. Αν

δηλαδή ο παίκτης πατήσει τα κουμπιά «**βελάκι δεξιά**», ή το κουμπί "**D**", τότε κινείται ενεργοποιείται το "**Horizontal**". Με το «**βελάκι πάνω**» ή το κουμπί "**W**", ο παίκτης κινείται στον κάθετο άξονα και εκτελείται η εντολή **Input.GetAxisRaw("Vertical")**; κ.λπ.

Η μεταβλητή **isGrounded**, γίνεται **true**, όταν ο παίκτης είναι στο έδαφος. Το **groundMask**, εντός του inspector του Unity, πρέπει να οριστεί από τον δημιουργό σε ποια αντικείμενα θα υπακούει ως «έδαφος», επιλέγοντας μία υπάρχουσα «**μάσκα**». Η μεταβλητή **groundDistance** είναι ένας αριθμός τύπου «κατώφλι». Δηλαδή, αν η απόσταση μεταξύ του 3D αντικειμένου και μεταξύ ενός αντικειμένου με την καθορισμένη μάσκα **groundMask** είναι μικρότερη από **groundDistance**, τότε επιτρέπεται ο παίκτης να εκτελέσει άλμα.

Επίσης, η κάμερα αλλάζει άκρο, μαζί και οι άξονες της κίνησης ώστε, όπου και να κοιτάει η κάμερα, η κίνηση προσαρμόζεται στην οπτική της γωνία. Π.χ. με το βελάκι εμπρός ή κουμπί "**W**", ο παίκτης κινείται στην κατεύθυνση που κοιτάει η κάμερα.

Τέλος, έχει τεθεί στις ρυθμίσεις του script εντός του Unity, το **groundMask** να είναι τύπου "**TerrainGround**", ώστε ο παίκτης να έχει διαθέσιμο το άλμα μόνο αν πατάει σε 3D αντικείμενα με τέτοια μάσκα πάνω τους. [43]

3.11.2. Scripts του βασικού μενού

Τα scripts που χρησιμοποιούνται στο βασικό μενού είναι:

- MainMenuGroup.cs
- SourcesMenu.cs
- SourcesWorld1.cs

Ανάλυση των scripts του βασικού μενού

MainMenuGroup.cs

Στο αρχικό μενού, με το που ανοίγει το παιχνίδι *The Two Worlds*, εμφανίζονται κάποια κουμπιά. Το «**Παίξε!**», «**Πηγές**», «**Έξοδος**». Το script αυτό είναι υπεύθυνο για την ομαλή λειτουργία τους.

Στην αρχή του script έχει οριστεί ένα **GameObject** με όνομα **exitQuestionWindow**. Όταν πατηθεί το κουμπί «**Έξοδος**», ενεργοποιείται αυτό το GameObject το οποίο είναι μια διεπαφή που ρωτά τον παίκτη για το αν θέλει να αποχωρήσει από το παιχνίδι, αυτό επιτυγχάνεται με την **ExitAppButton**(). Από εκεί, αν ο χρήστης επιλέξει «όχι» για να παραμείνει στο παιχνίδι, απλά το παράθυρο **exitQuestionWindow**, κλείνει. Αυτό επιτυγχάνεται με την συνάρτηση **noExitButton**(). Διαφορετικά, ο χρήστης μπορεί να πατήσει «ναι» και να κλείσει το παιχνίδι. Η λειτουργία αυτή περιγράφεται στην συνάρτηση **yesExitButton**().

Αν ο παίκτης επιλέξει το κουμπί «**Πηγές**», θα τον κατευθύνει το script στη σκηνή με τις πηγές. Η μέθοδος της κλάσης **CustomSceneController**, **getSourcesMenuNumber**() θα επιστρέψει τον ακέραιο που ανήκει στη σκηνή των κατηγοριών πηγών, οπότε θα γίνει η μετάβαση κανονικά. Η λειτουργία φαίνεται στην μέθοδο του script **GotoSources**().

Αν ο παίκτης επιλέξει το κουμπί «**Παίξε!**», θα κατευθυνθεί στον κόσμο **1** και θα γίνει επαναφορά των ερωτήσεων στον κόσμο **1** σε περίπτωση που ο παίκτης έχει ξανά μπει στον κόσμο **1** να παίξει χωρίς να κλείσει το παιχνίδι.

SourcesMenu.cs

Το script αυτό, διαχειρίζεται τη διεπαφή στη σκηνή «**Πηγές**».

Σε περίπτωση που πατηθεί το “Esc” κουμπί, ο παίκτης κατευθύνεται στο κύριο μενού. Αυτό είναι φανερό στο “**if**()” της **Update**(). Εκτελείται η **GotoMainMenu**().

Αν ο παίκτης επιλέξει το κουμπί «**Κόσμος 1**», θα κατευθυνθεί στις πληροφορίες των πηγών για τον κόσμο **1**, χάρης της **GotoSource1**() μεθόδου. Το αντίστοιχο θα συμβεί με το επόμενο κουμπί «**Κόσμος 2**», με τη βοήθεια της μεθόδου **GotoSource2**(), θα πλοηγηθεί ο παίκτης στη πηγή του κόσμου **2**.

Με το κουμπί «**Επιστροφή**», ο παίκτης επιστρέφει πίσω στο αρχικό μενού με την υποστήριξη της μεθόδου **GotoMainMenu**().

SourcesWorld1.cs

Όπως και στο προηγούμενο script, με το “Esc” ή με το κάτω κάτω κουμπί «**Επιστροφή**» πλοηγείται ο παίκτης ένα βήμα πίσω, στην σκηνή με τις κατηγορίες πηγών.

Με το πάνω πάνω κουμπί, που αναφέρει για τις δασικές πυρκαγιές, αν επιλεγεί, ανοίγει ο προεπιλεγμένος περιηγητής του λειτουργικού συστήματος με την ιστοσελίδα της πολιτικής προστασίας που αναφέρει για τις δασικές πυρκαγιές. Συνάρτηση: **goToFireSource**().

Με το αμέσως επόμενο κουμπί που αναφέρει για σεισμούς, αν επιλεγεί, ανοίγει ο προεπιλεγμένος περιηγητής και οδηγείται ο παίκτης στην ιστοσελίδα της πολιτικής προστασίας, που έχει πληροφορίες για τους σεισμούς. Συνάρτηση: **goToEarthquakeSource**().

Τέλος, στην περίπτωση που ο χρήστης πατήσει το κουμπί που αναφέρει για τα ηφαίστεια, θα μεταβεί στην αντίστοιχη ιστοσελίδα της πολιτικής προστασίας που έχει πληροφορίας για τα ηφαίστεια.

3.11.3. Scripts Κόσμου **1**

Στον κόσμο **1** χρησιμοποιούνται τα επόμενα scripts κατηγοριοποιημένα με βάσει την κάθε ερώτηση:

- Η ερώτηση φωτιάς έχει μόνο ένα script:
    1. FireOnTrigger.cs
- Η ερώτηση σεισμού λειτουργεί με τα εξής scripts:
    1. EarthquakeOnTrigger.cs
    2. ParkOnTrigger.cs
    3. SpitiaOnTrigger.cs
    4. WiresOnTrigger.cs
- Η ερώτηση ηφαιστείου λειτουργεί με τα εξής scripts:
    1. LavaOnTrigger.cs
    2. CarOnTrigger.cs
    3. BunkerOnTrigger.cs
    4. BoatOnTrigger.cs

Με την ενεργοποίηση της ερώτησης ηφαιστείου, ενεργοποιούνται ένα σετ από scripts για τα εφέ της έκρηξης του ηφαιστείου:

1. Cube.cs

2. CubeSpawner.cs
3. ExplodeOnGround.cs
4. IPooledObject.cs
5. ObjectPooler.cs

Άλλα σημαντικά scripts είναι:

1. PortalOnTrigger.cs
2. WorldOneFlagsAnswered.cs
3. BoatMovement.cs

*Ανάλυση των scripts του κόσμου **1***

Ερώτηση φωτιάς

FireOnTrigger.cs

Αυτό το script είναι υπεύθυνο να ενεργοποιήσει την ερώτηση της φωτιάς. Είναι επίσης και υπεύθυνο να ενεργοποιήσει και τις απαντήσεις σε μορφή κουμπιών ενός παραθύρου.

Οπότε, τη στιγμή που ο παίκτης πλησιάζει τον **collider** του θαυμαστικού, το παιχνίδι κάνει παύση και εμφανίζεται το **answerWindow**. Η διεπαφή αυτή έχει και την ερώτηση για το τι πρέπει να κάνει ο παίκτης σε περίπτωση που εντοπίσει φωτιά. Επίσης γίνεται έλεγχος για το αν έχει ξανά απαντηθεί η ερώτηση της φωτιάς. Συνάρτηση: **OnTriggerEnter** ().

Αν ο παίκτης επιλέξει το Α, τότε έχει απαντήσει σωστά. Τα εφέ της φωτιάς εξαφανίζονται, η φόρμα της ερώτησης για τη φωτιά εξαφανίζεται, εμφανίζεται το αντικείμενο **correctWindow**, η "**bool**" μεταβλητή για το αν απαντήθηκε η φωτιά γίνεται "**true**", διαλύεται το ερωτηματικό από τη περιοχή, εξαφανίζονται οι φωτιές, σώζεται το δάσος και αυξάνονται οι σωστά απαντημένες ερωτήσεις κατά 1. Επίσης, σε αυτό το στάδιο, η ερώτηση φωτιάς μαρκάρεται ως σωστή.

Σε οποιαδήποτε άλλη περίπτωση, για αρχή εξαφανίζεται το παράθυρο της ερώτησης για τη φωτιά, εμφανίζεται ένα παράθυρο που λέει πως ο παίκτης έκανε λάθος και ο καπνός στο κεφάλι του παίκτη ενεργοποιείται. Αντικαθίστανται επίσης τα καλά δέντρα με καμένα, το σπίτι δίπλα φαίνεται καμένο, το έδαφος εμφανίζεται καμένο και

ενεργοποιούνται οι καπνοί στο δάσος και πάνω στα δέντρα. Και τέλος η ερώτηση της φωτιάς μαρκάρεται ως απαντημένη.

Ερώτηση Σεισμού

EarthquakeOnTrigger.cs

Το script αυτό ενεργοποιεί την ερώτηση για τον σεισμό. Όταν ο παίκτης μπει στον **Collider** του θαυμαστικού, ενεργοποιείται η ερώτηση του σεισμού και εμφανίζεται το παράθυρο της περιγραφής της ερώτησης σεισμού.

Την ίδια στιγμή, το καλώδιο της δίπλα κολώνας εξαφανίζεται, και ενεργοποιείται το καλώδιο το οποίο είναι κομμένο και βγαίνουν από την άκρη του σπίθες. Ενεργοποιείται και η συνάρτηση που προκαλεί τον σεισμό. Η κάμερα **Cinemachine** κουνιέται με δύναμη **earthquakeStrength**, για μερικά δευτερόλεπτα. Ο σεισμός επιτυγχάνεται με την συνάρτηση **shake**() [44].

Όταν ο παίκτης πατήσει «**Συνέχεια**», εξαφανίζεται το θαυμαστικό και ο **Collider** του. Κλείνει το παράθυρο της ερώτησης του σεισμού, ενεργοποιείται μια ομάδα 3D αντικειμένων με όνομα **afterCollision**. Η ομάδα αυτή είναι τα 3 θαυμαστικά, και πρέπει ο παίκτης να πάει δίπλα σε ένα από αυτά, ώστε να απαντήσει στην ερώτηση με το σεισμό. Με το πάτημα του ίδιου κουμπιού, το παιχνίδι βγαίνει από παύση. Αυτό επιτυγχάνεται με τη συνάρτηση **continueFromQuestionWindowButton**()

Σε κάθε περίπτωση, σε όποια από τις 3 περιοχές και να πάει ο παίκτης ώστε να απαντήσει στην ερώτηση του σεισμού, θα εμφανιστεί μια διεπαφή η οποία θα έχει ένα μοναδικό κουμπί «**Συνέχεια**». Είτε απάντησε ο παίκτης σωστά, είτε λάθος, με αυτό το κουμπί θα εκτελεστούν οι εξής διαδικασίες. Θα εξαφανιστεί η ομάδα **afterCollision** γιατί η ερώτηση έχει απαντηθεί, οπότε τα θαυμαστικά δεν είναι αναγκαίο να φαίνονται πια. Στο ίδιο στάδιο, για κάθε απάντηση διατηρείται ένα ξεχωριστό παράθυρο για τον σχολιασμό της απάντησης (πχ. Σωστή Απάντηση ή Λάθος απάντηση). Και τα 3 αυτά παράθυρα απενεργοποιούνται, γιατί δεν χρειάζονται πλέον – η ερώτηση έχει απαντηθεί. Το παιχνίδι βγαίνει από την κατάσταση παύσης και τέλος, η ερώτηση σεισμού μαρκάρεται ως απαντημένη.

ParkOnTrigger.cs

Το script αυτό ενεργοποιείται τη στιγμή που ο παίκτης θα πλησιάσει το θαυμαστικό εντός του πάρκου. Τη στιγμή που ο παίκτης θα ενεργοποιήσει τις λειτουργίες αυτού του script, στο παιχνίδι γίνεται παύση.

Επίσης, ενεργοποιείται το παράθυρο που περιγράφει αν ο παίκτης έδρασε σωστά. Επειδή τυγχάνει το πάρκο να είναι η σωστή απάντηση – κίνηση, το πλήθος σωστών απαντήσεων αυξάνεται κατά ένα και η ερώτηση σεισμού μαρκάρεται ως σωστά απαντημένη.

SpitiaOnTrigger.cs

Αν ο παίκτης πλησιάσει το θαυμαστικό δίπλα στα ραγισμένα σπίτια, ενεργοποιείται αυτό το script.

Το παιχνίδι μπαίνει σε παύση και εμφανίζεται η αντίστοιχη διεπαφή. Επειδή αυτή η απάντηση είναι λανθασμένη, εμφανίζονται πάνω από το κεφάλι του παίκτη σπίθες και, τα υγιή πόδια του αντικαθίστανται με ραγισμένα.

WiresOnTrigger.cs

Αν ο παίκτης πλησιάσει το θαυμαστικό δίπλα στις κολώνες ρεύματος, ενεργοποιείται αυτό το script.

Όπως και στο προηγούμενο script **SpitiaOnTrigger.cs**, ο παίκτης βγάζει σπίθες και αποκτά σπασμένα πόδια. Η μόνη διαφορά είναι πως ενεργοποιείται η αντίστοιχη διεπαφή.

Ερώτηση Ηφαιστείου

LavaOnTrigger.cs

Αυτό το script είναι υπεύθυνο για την ενεργοποίηση της ερώτησης του ηφαιστείου, τη στιγμή που ο παίκτης περνάει κοντά από το θαυμαστικό στο νησί με το ηφαίστειο.

Κατά την ενεργοποίηση της ερώτησης, συμβαίνουν τα εξής: ενεργοποιείται η μεταβλητή (εφέ) **activateVolcanoMouth**, ενεργοποιείται ένα σετ από 3D αντικείμενα με όνομα **afterCollisionVolcano**.

Εξαφανίζεται ο **Collider** που ενεργοποιεί αυτό το script, εμφανίζεται η διεπαφή που περιγράφει το ηφαίστειο και τις πιθανές του απαντήσεις – κινήσεις, με το κουμπί «**Συνέχεια**». Το παιχνίδι κάνει παύση και 1 δευτερόλεπτο αργότερα, αφού πατηθεί το κουμπί «**Συνέχεια**» και βγει το παιχνίδι από παύση, δημιουργείται ένας μεγάλος σεισμός με δύναμη **earthquakeStrength** [44]. Ο σεισμός αυτός, δημιουργείται με τη συνάρτηση **shake**().

Τα εφέ **activateVolcanoMouth**, αποτελείται από τα εξής κομμάτια:

- Μια μεγάλη φωτιά μέσα στο στόμα του ηφαιστείου.
- Μόνιμες – επαναλαμβανόμενες εκρήξεις γύρω από το στόμα του ηφαιστείου.
- Πολλές σφαίρες φωτιάς που εκτοξεύονται από το στόμα του ηφαιστείου.

Τα script των εφέ που αναφέρονται, περιγράφονται παρακάτω.

Τα 3D αντικείμενα που ανήκουν στην ομάδα **afterCollisionVolcano**, είναι τα 3 θαυμαστικά με τους **Colliders** τους που επιτρέπουν στον παίκτη να απαντήσει την ερώτηση του ηφαιστείου. Η λειτουργία τους θα αναλυθεί αμέσως.

Αφού ο παίκτης διαβάσει την περιγραφή και τις 3 πιθανές απαντήσεις, από τη διεπαφή του ηφαιστείου και πατήσει «**Συνέχεια**», θα εκτελεστούν οι εξής διαδικασίες: θα εξαφανιστεί το θαυμαστικό που εκκινεί την ερώτηση ηφαιστείου, θα εξαφανιστεί η διεπαφή περιγραφής ερώτησης ηφαιστείου και το παιχνίδι θα βγει από την κατάσταση παύσης. Η λειτουργία πραγματοποιείται με την διαδικασία **continueFromQuestionWindowButton**().

Αφού απαντηθεί η ερώτηση του ηφαιστείου, κινώντας τον παίκτη σε 1 οποιοδήποτε από τα 3 θαυμαστικά, θα εμφανιστεί μία διεπαφή με ένα κουμπί «**Συνέχεια**». Η λειτουργία αυτού του κουμπιού φαίνεται στην συνάρτηση **continueFromAnswerWindowButton**(). Αυτή η συνάρτηση, θα εξαφανίσει όλα τα θαυμαστικά πιθανών απαντήσεων, θα εξαφανίσει όλες τις διεπαφές απάντησης, θα βγάλει το παιχνίδι από την κατάσταση παύσης και θα μαρκαριστεί η ερώτηση του ηφαιστείου ως απαντημένη.

CarOnTrigger.cs

Αυτό το script ενεργοποιείται, όταν ο παίκτης απαντήσει στην ερώτηση ηφαιστείου πηγαίνοντας στο αυτοκίνητο. Ενεργοποιεί την διεπαφή που περιγράφει αν η απάντηση ήταν σωστή ή λάθος, θέτει το παιχνίδι σε παύση και επειδή η ερώτηση ήταν λάθος, απενεργοποιεί το λευκό σώμα του παίκτη μέσω της ομάδας 3D αντικειμένων **lavaDeactivatedAfterWrong** και ενεργοποιεί την ομάδα 3D αντικειμένων **lavaActivatedAfterWrong** που κοκκινίζει το σώμα του παίκτη.

BunkerOnTrigger.cs

Αυτό το script ενεργοποιείται, όταν ο παίκτης απαντήσει στην ερώτηση ηφαιστείου πηγαίνοντας στο οχηρό ψηλά στο λόφο. Ενεργοποιεί την διεπαφή που περιγράφει αν η απάντηση ήταν σωστή ή λάθος, θέτει το παιχνίδι σε παύση και επειδή αυτή η ερώτηση είναι σωστή αυξάνει το πλήθος των σωστά απαντημένων ερωτήσεων κατά ένα. Τέλος μαρκάρει την ερώτηση ηφαιστείου ως σωστά απαντημένη.

BoatOnTrigger.cs

Αυτό το script ενεργοποιείται, όταν ο παίκτης απαντήσει στην ερώτηση ηφαιστείου πηγαίνοντας στη βάρκα, δίπλα στη προβλήτα. Ενεργοποιεί την διεπαφή που περιγράφει αν η απάντηση ήταν σωστή ή λάθος, θέτει το παιχνίδι σε παύση και επειδή η ερώτηση ήταν λάθος απενεργοποιεί το λευκό σώμα του παίκτη μέσω της ομάδας 3D αντικειμένων **lavaDeactivatedAfterWrong** και ενεργοποιεί την ομάδα 3D αντικειμένων **lavaActivatedAfterWrong** που κοκκινίζει το σώμα του παίκτη.

Scripts για τα εφέ του Ηφαιστείου

Για την δημιουργία του κώδικα, ώστε να πετάγονται οι σφαίρες από το στόμα του ηφαιστείου, χρησιμοποιήθηκε η μέθοδος **Object Pooling**. Αυτή η τεχνική, διατηρεί πάντα έναν συγκεκριμένο αριθμό από 3D Objects από το **pool**. Δουλεύει, όπως λέει το όνομα της, ως «πισίνα». Κάθε τέτοια πισίνα, έχει ένα **tag** (ετικέτα), ένα είδος **GameObject** και ένα **μέγεθος**.

Δουλεύει ως εξής: έστω ότι χρειάζεται σε έναν χώρο να βγαίνουν συνεχώς **15 μπλε κύβοι**. Από εδώ θα προκύψει μία «πισίνα» που το **tag** θα είναι π.χ. "**BlueCube**", για **GameObject** θα τοποθετηθεί ένα **prefab μπλε κύβου** που θα έχει προσαρμόσει ο δημιουργός και το **size** της «πισίνας» θα είναι **15**. Αυτό σημαίνει πως θα έχουμε μια πισίνα με 15 μπλε κύβους που θα «κοιμούνται».

Όταν κληθεί να ενεργοποιηθεί το **pool** με το tag "**BlueCube**", θα εμφανίζονται συνεχόμενα στην οθόνη **15 μπλε κύβοι** εξαγόμενοι από αυτό το **pool**.

Τη χρονική στιγμή που πάει να βγει **μπλε κύβος** και το **pool** είναι άδειο, δηλαδή χρησιμοποιούνται και οι 15 μπλε κύβοι, ο 1ος κύβος που εμφανίστηκε, διαλύεται και «γεννιέται» ένας νέος **μπλε κύβος**. Πάντα διαλύεται ο κύβος που είναι «ζωντανός» πιο πολύ ώρα και βγαίνει ένας καινούριος.

Η ουσία είναι πως θα έχουμε πάντα **15 μπλε κύβους** να «γεννιούνται» ασταμάτητα και ποτέ περισσότερους. Η διαδικασία του **Object Pooling** θεωρείται ότι είναι πολύ οικονομική και ελαφριά για το σύστημα. [45]

Cube.cs

Αυτό το script διαχειρίζεται τη συμπεριφορά της κάθε φλεγόμενης σφαίρας τη στιγμή που «ζωντανεύει» μέσα στο στόμα του ηφαιστείου. Πιο συγκεκριμένα, με πόση δύναμη προς στον άξονα **X**, πόση δύναμη στον άξονα **Y**, με πόση δύναμη στον άξονα **Z**. Και τέλος το αποτέλεσμα το εκχωρεί στην ιδιότητα του **Rigidbody**, **velocity** της φλεγόμενης σφαίρας, όπου είναι πάνω «κολλημένο» όλο το script **Cube.cs**.

Το script αυτό «κληρονομεί» χαρακτηριστικά, από το **MonoBehaviour** και το **IPooledObject**. Την μέθοδο **OnObjectSpawn**() την κληρονομεί από το **Interface**: **IPooledObject**.

CubeSpawner.cs

Το script αυτό είναι ενσωματωμένο μέσα στα components ενός **κενού 3D αντικειμένου** που βρίσκεται μέσα στο στόμα του ηφαιστείου.

Δημιουργεί ένα αντικείμενο τύπου **Objectpooler** και το ενεργοποιεί στην μέθοδο **Start**(). Η **FixedUpdate**() καλείται να δημιουργήσει μια

«πισίνα» με **tag** "**Cube**". Το script αυτό, τοποθετείται στις συντεταγμένες που βρίσκεται το παραπάνω **κενό 3D αντικείμενο** (το στόμα του ηφαιστείου) και παίρνει τις πληροφορίες του Rotation αυτού του κενού **3D αντικειμένου**. Σε αυτό το σημείο ενεργοποιείται πλήρως η «πισίνα», δηλαδή εκτοξεύονται οι σφαίρες.

ExplodeOnGround.cs

Σε αυτό το script περιγράφεται το τι θα γίνει σε περίπτωση που οι σφαίρες φωτιάς ακουμπήσουν το έδαφος, πιο συγκεκριμένα, με αντικείμενα που έχουν **tag** τύπου "**Terrain**". Στην προκειμένη περίπτωση, «σκάνε» και εμφανίζεται μία μικρή έκρηξη όταν ακουμπήσουν το **terrain** που έχει ως **tag** "**Terrain**". Με την μέθοδο .**Play**() των αντικειμένων **ParticleSystem**, ενεργοποιείται το αντίστοιχο **Particle** για τη μικρή έκρηξη [46].

IPooledObject.cs

Το script αυτό έχει απλά τη λειτουργία ενός **Interface**. Τα Interfaces, έχουν κυρίως την ιδιότητα να υπενθυμίζουν στον δημιουργό να τοποθετούν μεθόδους στις κύριες κλάσεις. Σε αυτή την περίπτωση θα υπενθυμίζεται στον δημιουργό να τοποθετήσει μέσα στην καινούρια κλάση, την μέθοδο **OnObjectSpawn**().

ObjectPooler.cs

Σε αυτό το script δηλώνεται η δομή ενός **pool**. Όποτε καλείται το script αυτό, φτιάχνεται ένα καινούριο «λεξικό» με ένα **tag** και με μία **σειρά** από **GameObjects**. Αυτή η σειρά γεμίζει βάζοντας στη σειρά τα prefabs που έχουν οριστεί με την εντολή "**Enqueue**", δηλαδή τις φλεγόμενες σφαίρες. Όταν η σειρά αυτή γεμίσει, προστίθεται μέσα στο λεξικό των «πισίνων».

Να τονιστεί πως σε αυτό το λεξικό μπορούν να προστεθούν και άλλες «πισίνες» με – υποχρεωτικά – άλλα **tags**, με άλλο ή ίδιο 3D **prefab** και με άλλα ή όμοια **μεγέθη**.

Εντός του Unity, δηλώνεται να «γεννιούνται» ασταμάτητα από το στόμα του ηφαίστειου, το **prefab** της φλεγόμενης σφαίρας και το μέγεθος του **pool** αυτού έχει οριστεί στο **180**, ώστε να φαίνονται πολλές.

Άλλα σημαντικά scripts

PortalOnTrigger.cs

Το script αυτό είναι υπεύθυνο για τη μετάβαση του παίκτη από τον κόσμο **1** στον κόσμο **2.** Όταν ο παίκτης περάσει μέσα από τη μαγική πόρτα, αν έστω και μία ερώτηση δεν έχει απαντηθεί, τότε εμφανίζεται αντίστοιχο μήνυμα. Πάντα εμφανίζεται κατάλληλο μήνυμα για οποιοδήποτε συνδυασμό μη απαντημένων ερωτήσεων. Αυτό επιτυγχάνεται με ένα αντικείμενο τύπου **string**, το οποίο κάθε φορά που εντοπίζεται πως μία ερώτηση δεν έχει απαντηθεί, προστίθεται μπροστά στο **string** το αντίστοιχο μήνυμα. Όταν αυτό το string, δημιουργηθεί πλήρως, τοποθετείται σε ειδικό παράθυρο.

Διαφορετικά, αν δηλαδή ο παίκτης έχει απαντήσει και στις 3 ερωτήσεις, εμφανίζεται κατάλληλο μήνυμα για την αντίστοιχη περίπτωση. Π.χ. Αν δεν απάντησε καμία σωστή, «Προσπάθησε Ξανά.», ενώ αν τις απάντησε όλες, «Τα πήγες τέλεια.». Αυτά εκτελούνται στην μέθοδο **OnTriggerEnter**()

Στην περίπτωση που απομακρυνθεί ο παίκτης από την μαγική πόρτα, εκτελείται η **OnTriggerExit**() και ότι παράθυρο ήταν ενεργό, εξαφανίζεται. Παρόμοια αποτελέσματα υπάρχουν όταν ο παίκτης πατάει «**Παραμονή**» στον κόσμο **1**. Αυτό πραγματοποιείται στην **continueButtonAndStayToWorldOneWindow**().

Στην περίπτωση που πατήσει ο παίκτης «**ταξίδεψε!**», εκτελείται η συνάρτηση μετάβασης **goToWorld2**().

Στην περίπτωση που πατήσει ο παίκτης «**Ξαναπαίξε**» για να διορθώσει το σκορ στον κόσμο **1**, εκτελείται η συνάρτηση **replayWorld1**(). Επίσης ο κόσμος **1** σε αυτό το σημείο αρχικοποιείται.

WorldOneFlagsAnswered.cs

Είναι ένα script το οποίο για τον κόσμο **1**, αποθηκεύει: το αν απαντήθηκαν όλες οι ερωτήσεις, το πόσες ερωτήσεις απαντηθήκαν σωστά και τέλος ποιες ερωτήσεις απαντήθηκαν ή όχι ακόμη. Όλες οι μεταβλητές είναι ορατές μόνο μέσα από τις ειδικές μεθόδους **public static**.

Υπάρχει και η συνάρτηση **resetAll**() η οποία επαναφέρει όλες τις ερωτήσεις του κόσμου **1** στην αρχική τους κατάσταση ώστε ο παίκτης αν

θέλει να μπει και να ξαναπαίξει χωρίς να χρειάζεται να κλείσει το παιχνίδι και να το ξανά ανοίξει.

BoatMovement.cs

Με αυτό το script είναι διαχειρίσιμη η μικρή και ελαφριά κίνηση της βάρκας πάνω στη θάλασσα, με ημιτονοειδή ταχύτητα. Μπορεί να ρυθμιστεί η συχνότητα επαναλήψεων, η απόσταση που μπορεί να φτάσει η βάρκα, και το αν θα ξεκινάει από άλλο σημείο. [47]

3.11.4. Scripts Κόσμου **2**

Τα scripts του κόσμου **2** είναι τα ακόλουθα:

- Για την περιοχή **1**, όπου ο παίκτης καλείται να επιλέξει 4 από τα 9 αντικείμενα που του παρουσιάζονται, τα scripts είναι:
    1. ChoiceOfFourObjsPhaseOne.cs
    2. ChooseFourObjectsTrigger.cs
    3. OnTriggerDontAllowPlayer.cs
- Για την περιοχή **2**, το σημείο που ο παίκτης καλείται να κατασκευάσει την σκηνή και τη βάση της, υπάρχουν τα εξής scripts:
    1. WorldTwoTentCreateTrigger.cs
    2. BaseTrigger.cs
    3. BuildTentPhaseTwo.cs
- Στο κομμάτι που ο παίκτης πρέπει να φτιάξει την φωτιά, τα scripts που υλοποιούν τις αντίστοιχες λειτουργίες είναι:
    1. BuildFirePhaseThree.cs
    2. BuildFireReal.cs
- Για την αμέσως επόμενη φάση όπου ο παίκτης πρέπει να βρει από κάπου νερό, τα scripts είναι:
    1. WaterCollect.cs
    2. LeaveTheBucketToFire.cs
- Για το επόμενο σημείο που ο παίκτης θα πρέπει να σχηματίσει το σήμα S.O.S. στο πάτωμα, ακουμπώντας πέτρες, υπάρχει μόνο ένα script:
    1. CreateSOS.cs

- Η τελευταία φάση αυτού του κόσμου που έρχεται το ελικόπτερο δίνοντας την επιλογή να αποχωρήσει ή να παραμείνει ο παίκτης αποτελείται από τα scripts:
    1. HelicopterPhase.cs
    2. RotateBigHelix.cs
    3. RotateSmallHelix.cs
- Άλλα scripts που χρησιμεύουν σε αυτόν τον κόσμο είναι:
    1. OnMouseHoverInfoButton.cs
    2. TimeClock.cs

*Ανάλυση των scripts του κόσμου **2***

Περιοχή **1** – επιλογή τεσσάρων από τα 9 αντικείμενα

ChoiceOfFourObjsPhaseOne.cs

Στο script αυτό, ρυθμίζεται η επιλογή των τεσσάρων αντικειμένων που επέλεξε ο παίκτης. Στην μέθοδο **Start**() αρχικοποιείται ο πίνακας με τα 3D μοντέλα των 9 διαθέσιμων αντικειμένων για να επιλέξει ο παίκτης, στην μεταβλητή **Allobj**[]. Το πόσα αντικείμενα έχει επιλέξει σύνολο ο παίκτης στη μεταβλητή **chosenItemsPhaseOne** που αρχικοποιείται με 0. Και τέλος, δημιουργείται μία λίστα από αντικείμενα τύπου string για να διατηρούνται τα αντικείμενα που έχει επιλέξει ο παίκτης στην μεταβλητή **chosenByNames**(). Είναι χρήσιμος ο πίνακας **Allobj**[] για την περίπτωση λάθος επιλογής αντικείμενων και εξηγείται παρακάτω.

Αν ο παίκτης, έχει επιλέξει λιγότερα από 4 αντικείμενα πάνω του, τότε ενεργοποιείται η διαδικασία της επιλογής του.

Αν ο παίκτης δεν έχει κάνει παύση το παιχνίδι και πατήσει το αριστερό κλικ πάνω στα αντικείμενα, τότε μπορεί να τα επιλέξει. Αυτό συμβαίνει γιατί θα ήταν αντιαισθητικό να πατούσε ο παίκτης "**Esc**", να φαινόταν το μενού παύσης του παιχνιδιού και ο παίκτης να μπορεί ακόμη να επιλέγει αντικείμενα.

Αμέσως μετά, δημιουργείται μία αόρατη ακτίνα από την κάμερα ως το 3D αντικείμενο επιλογής. Ο βασικός παίκτης θεωρείται 3D αντικείμενο, οπότε είναι αναγκαίο να «προσπερνάει» η ακτίνα επιλογής από μέσα του, αυτό επιτυγχάνεται με την μεταβλητή **ignoreMask** [48]. Κρίθηκε

κουραστικό να σταματάει η ακτίνα πάνω στον παίκτη, αντί να επιλέγεται το αντίστοιχο αντικείμενο που κρύβεται από πίσω του.

Σε επόμενη φάση καθαρίζεται το Log από μηνύματα, για λόγους debugging. Και στη συνέχεια ελέγχεται το τι «χτύπησε» ακτίνα επιλογής στην συνάρτηση **phase1Function**(). Αν η ακτίνα δεν ήταν πάνω από κάποια διεπαφή (εξηγείται σε επόμενο script), ελέγχεται το **tag** του 3D αντικειμένου που επιλέχθηκε. Αν η ακτίνα «βρήκε» σε ένα από τα 9 3D αντικείμενα προς επιλογή, τότε αυτό το αντικείμενο απενεργοποιείται, μετά αυξάνεται ο αριθμός των επιλεγμένων αντικειμένων που έχει ο παίκτης πάνω του κατά **1** και στην λίστα **chosenByNames**() προστίθεται το αντίστοιχο όνομα αντικειμένου.

Αν ο παίκτης έχει επιλέξει 4 αντικείμενα, ελέγχεται η περίπτωση της σωστής επιλογής του παίκτη, μέσω της συνάρτησης **ChosenTheRightObjects**() η οποία γίνεται **true** αν ο παίκτης επέλεξε τα σωστά. Αν έστω και ένα αντικείμενο, δεν ταιριάζει με το όνομα ενός από τα σωστά αντικείμενα, επιστρέφει **false**.

Στην περίπτωση που η παραπάνω συνάρτηση γίνει **true**, δηλαδή ο παίκτης επέλεξε τα σωστά αντικείμενα, το κουμπί βοήθειας στα δεξιά απενεργοποιείται, το θαυμαστικό και οι κορμοί δέντρων – τα εμπόδια εξαφανίζονται, εμφανίζεται η διεπαφή ότι η φάση της επιλογής αντικειμένων έχει τελειώσει και το παιχνίδι βγαίνει από την κατάσταση παύσης.

Αφού τελειώσει η φάση της επιλογής αντικειμένων με επιτυχία, η διεπαφή που θα εμφανιστεί, θα έχει μόνο το κουμπί «**Συνέχεια**». Αυτό το κουμπί εκτελεί τις εξής λειτουργίες: ξανά επιτρέπει την συλλογή αντικειμένων, κλείνει τη διεπαφή επιτυχίας, βγάζει το παιχνίδι από παύση, ξεκινάει να μετράει ο χρόνος (θα αναλυθεί σε επόμενο script) και εμφανίζεται το θαυμαστικό στη μέση της επόμενης περιοχής. Αυτά όλα εκτελούνται στην συνάρτηση **continueToPhaseTwoSuccess**().

Στην περίπτωση που ο παίκτης δεν επέλεξε τα σωστά αντικείμενα, το παιχνίδι κάνει παύση και εμφανίζεται η διεπαφή λάθους με το κουμπί «**Προσπάθεια Ξανά**». Όταν πατηθεί αυτό το κουμπί, εκτελούνται οι εξής διαδικασίες: το πλήθος επιλεγόμενων αντικειμένων γίνεται 0, ώστε ο παίκτης να μπορέσει να ξαναμπεί στη διαδικασία επιλογής, η λίστα των **string** που γέμιζε με τα ονόματα των επιλεγόμενων αντικειμένων αδειάζει ώστε να γεμίσει με τα ονόματα των νέων επιλεγόμενων αντικειμένων, η λίστα με όλα τα 3D αντικείμενα προς επιλογή ξανά ενεργοποιείται για να

γίνουν όλα τα αντικείμενα φανερά και να τα δει ο παίκτης, η επιλογή να μαζεύει ο χρήστης αντικείμενα ξανά ενεργοποιείται, το παιχνίδι βγαίνει από την κατάσταση παύσης και η διεπαφή των λανθασμένων επιλογών κλείνει. Οι λειτουργίες αυτές εκτελούνται στην **συνάρτησηresetForPhaseOne**().

Στην περίπτωση που ο παίκτης έχει χαθεί και δεν γνωρίζει που πρέπει να πάει, υπάρχει ένα κουμπί – θαυμαστικό στα δεξιά της οθόνης. Αν ο παίκτης το πατήσει, θα του εμφανιστεί βοηθητικό παράθυρο με οδηγίες. Οι οδηγίες αυτές του παρουσιάζουν το πόσα αντικείμενα πρέπει να επιλεγούν ακόμη και το ποια αντικείμενα έχει επιλέξει ο παίκτης από την συνάρτηση **showMeWhatToDo**(). Αυτό το παράθυρο, έχει ένα κουμπί «**Συνέχεια**», αν πατηθεί, το παράθυρο θα κλείσει. Αυτό συμβαίνει με τη βοήθεια της συνάρτησης **simpleContinue**().

ChooseFourObjectsTrigger.cs

Με την μετάβαση του παίκτη από τον κόσμο **1** στον κόσμο **2**, ο παίκτης βλέπει 1 θαυμαστικό μπροστά του. Αν το πλησιάσει, το εμφανίζεται μία διεπαφή με την περιγραφή επιλογής αντικειμένων, όπως περιγράφεται στην **OnTriggerEnter**().

Αφού ο παίκτης διαβάσει την περιγραφή, μπορεί με το κουμπί «**Συνέχεια**» να κλείσει αυτό το παράθυρο. Όταν πατηθεί αυτό το κουμπί επιτρέπεται ξανά η συλλογή αντικειμένων, εμφανίζεται δεξιά της οθόνης ένα μικρό κουμπί με ένα θαυμαστικό, κλείνει η διεπαφή περιγραφής επιλογής των τεσσάρων αντικειμένων και εξαφανίζεται το θαυμαστικό στη μέση αυτής της περιοχής. Οι εντολές αυτές πραγματοποιούνται εντός της **closeWindow**().

OnTriggerDontAllowPlayer.cs

Αν ο παίκτης επιχειρήσει να μεταβεί στην επόμενη περιοχή, πλησιάζοντας τους κορμούς δέντρων – εμπόδια, θα του εμφανιστεί μια διεπαφή η οποία θα του πει πως πρέπει πρώτα να επιλεγούν τα 4 αντικείμενα. Αν πατήσει συνέχεια, του επιτρέπεται η συλλογή αντικειμένων ξανά και κλείνει η διεπαφή αυτή.

WorldTwoTentCreateTrigger.cs

Σε αυτό το script, ο παίκτης εκκινεί την διαδικασία κατασκευής της σκηνής. Όταν ο παίκτης πλησιάσει το θαυμαστικό, το παιχνίδι κάνει παύση, εμφανίζεται η διεπαφή για τη δημιουργία της σκηνής, το θαυμαστικό εξαφανίζεται, εμφανίζεται η διάφανη – άκτιστη βάση της σκηνής και από πάνω από την βάση το νέο θαυμαστικό. Όλα αυτά περιγράφονται στην συνάρτηση **OnTriggerEnter**().

Αφού ο παίκτης διαβάσει την περιγραφή και είναι έτοιμος να συνεχίσει, πατώντας «**Συνέχεια**», θα εκτελεστούν οι εντολές της συνάρτησης **closeTentCreationInfoWindow**(). Θα κλείσει η διεπαφή περιγραφής, θα ενεργοποιηθεί το βοηθητικό κουμπί στα δεξιά της οθόνης, θα επιτρέπεται στον παίκτη να επιλέγει αντικείμενα και το παιχνίδι θα βγει από την παύση.

Αν ο παίκτης χαθεί, με το πάτημα του κουμπιού στο δεξιό μέρος της οθόνης, θα ανοίξει η διεπαφή βοήθειας. Με το πάτημα του κουμπιού «**Συνέχεια**», το βοηθητικό παράθυρο κλείνει.

BaseTrigger.cs

Όταν ο παίκτης θα πλησιάσει το θαυμαστικό πάνω από τη βάση της σκηνής, θα συμβούν εντός της **OnTriggerEnter**() τα εξής: θα εμφανιστεί το παράθυρο πληροφοριών για το σκάψιμο της βάσης, το βοηθητικό κουμπί στα δεξιά θα εξαφανιστεί και το παιχνίδι θα κάνει παύση.

Όταν ο παίκτης πατήσει «**Σκάψε**», θα εκτελεστεί ο κώδικας της **DigBaseButton**(). Δηλαδή, επιτρέπεται ξανά στον παίκτη να συλλέγει αντικείμενα από το έδαφος, το παιχνίδι βγαίνει από παύση, εξαφανίζεται η διάφανη βάση και εμφανίζεται η κανονική, εξαφανίζεται το παράθυρο περιγραφής και το θαυμαστικό, εμφανίζεται το παράθυρο για συλλογή αντικειμένων, εμφανίζεται η δεξιά και η αριστερή πλευρά της στέγης της σκηνής, ενεργοποιείται ο **Collider** της σκηνής ώστε ο παίκτης να μην μπορεί να περάσει από μέσα και εμφανίζεται ένα νέο θαυμαστικό πάνω από την σκηνή για να γνωρίζει ο παίκτης, ακόμη και από απόσταση που πρέπει να πάει.

Στην διεπαφή συλλογής αντικειμένων, όταν πατηθεί το κουμπί «**Συνέχεια**» εκτελούνται μέσω της συνάρτησης

continueToCollectBranches() τα εξής: εμφανίζεται στα δεξιά το πόσα ξύλα κουβαλάει πάνω του ο παίκτης, ο παίκτης μπορεί να συλλέγει αντικείμενα από το έδαφος, κλείνει η διεπαφή για τη στέγη, εμφανίζεται στα δεξιά το βοηθητικό κουμπί και ο παίκτης μπορεί να συλλέγει ξύλα για τη κατασκευή της σκηνής. Το τελευταίο δηλώνεται μέσω της **bool** μεταβλητής **collectBranchesForTent** που γίνεται **true**.

Αν ο παίκτης χαθεί, με το πάτημα του κουμπιού στο δεξιό μέρος της οθόνης, θα ανοίξει η διεπαφή βοήθειας. Με το πάτημα του κουμπιού «**Συνέχεια**», το βοηθητικό παράθυρο κλείνει.

BuildTentPhaseTwo.cs

Στην αρχή αυτού του script, στην μέθοδο **Start**(), αρχικοποιούνται 2 πίνακες, οι **AllRealSides**() και **AllTransparentSides**(). Ο Πρώτος πίνακας είναι υπεύθυνος να έχει μέσα τα **6** κομμάτια της **πραγματικής στέγης** και ο δεύτερος τα **6** κομμάτια της **διάφανης** – **άκτιστης στέγης**. Επίσης, αρχικοποιείται το πόσα ξύλα κρατάει ο παίκτης πάνω του, το μήνυμα για το πόσα ξύλα θέλει ακόμη, πόσος είναι ο μέγιστος αριθμός ξύλων που μπορεί να κουβαλήσει πάνω του, αρχικοποιείται το μήνυμα στα δεξιά για το πόσα ξύλα θέλει ακόμη, αρχικοποιείται το πόσο έχει χτιστεί η σκηνή, ρυθμίζεται η απόσταση από την κάμερα (για την ακτίνα επιλογής των ξύλων) και απενεργοποιείται η δυνατότητα να μαζέψει ο παίκτης οποιοδήποτε υλικό για την επόμενη φάση που είναι το χτίσιμο της φωτιάς.

Αν η φάση που ο παίκτης φτιάχνει τη σκηνή και η φάση της φωτιάς δεν είναι ενεργοποιημένη, τότε αν ο παίκτης κάνει κλικ και του επιτρέπεται να συλλέγει αντικείμενα από το έδαφος, εκτοξεύεται μία, αόρατη από τον παίκτη, ακτίνα από την κάμερα έως εκεί που έκανε κλικ ο παίκτης.

Αν αυτή η ακτίνα χτυπήσει μακριά ή το παιχνίδι είναι σε παύση από τον παίκτη, δεν γίνεται τίποτα. Σε κάθε άλλη περίπτωση, αν η ακτίνα καταφέρει να ακουμπήσει πάνω σε 3D αντικείμενο με **tag** «**ksulo**» και δεν κουβαλάει μέγιστο αριθμό ξύλων πάνω του, αυτό το αντικείμενο το αποθηκεύουμε προσωρινά σε μία μεταβλητή **GameObject**.

Το **GameObject** αυτό, το διαχειρίζεται μετά η **collectWood**(). Αυτή η συνάρτηση είναι υπεύθυνη για να μαζεύει ο παίκτης ξύλα. Το ξύλο εξαφανίζεται από το έδαφος, το πόσα ξύλα κρατάμε αυξάνεται κατά **1**, το TextBox στα δεξιά της οθόνης, ενημερώνεται με τον νέο αριθμό π.χ. 1/3, και το κείμενο για το αν ο παίκτης θέλει πολλά ή λίγα ξύλα συνολικά για να τελειώσει με την κατασκευή της σκηνής.

Αν ο παίκτης κάνει κλικ πάνω σε αντικείμενο με **tag** «**tent**», δηλαδή την σκηνή, τότε εκτελείται η συνάρτηση **buildTent**() και γίνεται η πρόσθεση του ποσού που έχει χτιστεί ήδη η σκηνή με τα ξύλα που κρατάει ο παίκτης.

Αν το ποσό αυτό είναι μικρότερο του 6, τότε εμφανίζονται – χτίζονται τόσα κομμάτια της σκηνής από όσα ξύλα κουβαλάει πάνω του ο παίκτης.

Αν το ποσό αυτό είναι μεγαλύτερο ή ίσο του 6, τότε αυτό σημαίνει πως ο παίκτης μπορεί να χτίσει πλήρως τη σκηνή. Και για να μην προκληθεί «Εξαίρεση εκτός ορίων Πίνακα» (Out Of bounds Exception), από τους 2 δηλωμένους πίνακες της στέγης, απλά χτίζεται ολόκληρη η υπόλοιπη σκηνή.

Ο όρος χτίζεται, σημαίνει πως εξαφανίζονται **X** κομμάτια διάφανης – άκτιστης σκηνής και εμφανίζονται **X** κτισμένα κομμάτια της σκηνής.

Αν η σκηνή θέλει κι άλλα ξύλα, τότε το πόσο έχει χτιστεί η σκηνή, προστίθεται με το πόσα ξύλα άφησε τώρα ο παίκτης, ώστε στην επόμενη φορά που θα αφήσει ξύλα ο παίκτης, να συνεχιστεί το χτίσιμο από εκεί που είχε μείνει η σκηνή. Τα ξύλα που κρατάει πλέον ο παίκτης πάνω του γίνονται 0 και ενημερώνεται στα δεξιά το TextBox.

Αν η σκηνή χτίστηκε πλήρως, τότε εξαφανίζεται το θαυμαστικό πάνω από τη σκηνή, εξαφανίζεται το βοηθητικό κουμπί στα δεξιά της οθόνης, ενεργοποιείται η **bool** μεταβλητή για να μπορέσει ο παίκτης να μαζέψει αντικείμενα για τη δημιουργία της φωτιάς, «κλείνει» η **bool** μεταβλητή για την κατασκευή της σκηνής, συνεπώς, αν ο παίκτης κάνει κλικ πάνω στην σκηνή δεν θα γίνει τίποτα και τέλος, εμφανίζεται η διεπαφή ολοκλήρωσης κατασκευής της σκηνής με το κουμπί «**Πρόσθεσε χιόνι!**».

Η συνάρτηση **showBackpack**(), χρησιμοποιείται από τον δημιουργό για λόγους ελέγχου – Debug. Δεν έχει κάποια οπτική αξία.

Στην διεπαφή ολοκλήρωσης της κατασκευής της σκηνής, αν πατηθεί το κουμπί «**Πρόσθεσε χιόνι!**», ενεργοποιείται η κάλυψη χιονιού στην δεξιά και αριστερή στέγη της σκηνής, κλείνει η διεπαφή της ολοκλήρωσης της στέγης, ανοίγει η διεπαφή δημιουργίας της φωτιάς και εξαφανίζεται το κείμενο και ο αριθμός ξύλων στα δεξιά της οθόνης.

BuildFirePhaseThree.cs

Στην εκκίνηση αυτού του script, στην **Start**(), δεν επιτρέπεται στον παίκτη να σηκώνει πέτρες και ξύλα από το έδαφος από το έδαφος.

Από την προηγούμενη φάση (κατασκευή σκηνής), στο τέλος εμφανίστηκε μία διεπαφή που περιέγραφε την αρχή δημιουργίας φωτιάς. Με το κουμπί «**Συνέχεια**» θα εκτελεστούν τα εξής: θα κλείσει αυτή η διεπαφή, θα εμφανιστεί πάνω από τη φωτιά ένα κόκκινο θαυμαστικό, θα εμφανιστεί η διάφανη – άκτιστη βάση της φωτιάς και στα δεξιά της οθόνης θα εμφανιστεί το βοηθητικό κουμπί. Αυτά υλοποιούνται στην **continueFromFireStartWindow**().

Όταν ο παίκτης πλησιάσει το κόκκινο θαυμαστικό θα εκτελεστεί η **OnTriggerEnter**(), θα εμφανιστεί η διεπαφή περιγραφής για τη βάση της φωτιάς, το παιχνίδι θα κάνει παύση και το κουμπί βοήθειας στα δεξιά θα εξαφανιστεί. Αυτή η διεπαφή θα έχει μόνο το κουμπί «**Σκάψε**».

Όταν θα πατηθεί το κουμπί «**Σκάψε**», θα εκτελεστεί η συνάρτηση **digBaseFire**(). Αυτή συνάρτηση, υλοποιεί τα εξής: ανοίγει τη διεπαφή περιγραφής δημιουργίας της φωτιάς, εξαφανίζει από το έδαφος την διάφανη – άκτιστη φωτιά, εμφανίζεται η αληθινή βάση της φωτιάς και κλείνει η προηγούμενη διεπαφή περιγραφής για τη βάση της φωτιάς. Η νέα διεπαφή περιέχει το κουμπί «**Συνέχεια**».

Όταν πατηθεί το κουμπί αυτό, εκτελείται η **continueToCollectStonesAndBranches**() η οποία: κλείνει την διεπαφή περιγραφής δημιουργίας φωτιάς, εξαφανίζει τον **Collider** που εκκινεί την δημιουργία της φωτιάς, ενεργοποιεί το βοηθητικό κουμπί στα δεξιά, ενεργοποιεί τα διάφανα – άκτιστα ξύλα και τις διάφανες – άκτιστες πέτρες, ενεργοποιεί πάνω από το βοηθητικό κουμπί τις πληροφορίες για το πόσες πέτρες και ξύλα κρατάει ο παίκτης πάνω του καθώς και τις πληροφορίες για το πόσες πέτρες ή ξύλα θέλει ακόμη, ενεργοποιείται η ικανότητα να μαζεύει ξύλα και πέτρες από το έδαφος, ενεργοποιείται γενικά η ιδιότητα να μαζεύει αντικείμενα και το παιχνίδι βγαίνει από παύση.

Αν ο παίκτης χαθεί, με το πάτημα του κουμπιού στο δεξιό μέρος της οθόνης, θα ανοίξει η διεπαφή βοήθειας. Με το πάτημα του κουμπιού «**Συνέχεια**», το βοηθητικό παράθυρο κλείνει.

BuildFireReal.cs

Σε αυτό το στάδιο ενεργοποιείται η δημιουργία της φωτιάς. Στη μέθοδο **Start**() αρχικοποιούνται: το πόσα ξύλα κρατάει ο παίκτης με 0, τον μέγιστο αριθμό ξύλων που μπορεί να κρατήσει ο παίκτης, το πόσες πέτρες μπορεί να κρατάει ο παίκτης με 0, τον μέγιστο αριθμό από πέτρες που μπορεί να κρατήσει ο παίκτης, το αν θέλει ακόμη πέτρες ή αν θέλει ακόμη ξύλα με 2 μεταβλητές **bool** οι οποίες είναι **true**, ορίζεται με άλλη μεταβλητή **bool** με τιμή **false** ότι η φωτιά δεν έχει χτιστεί, ορίζεται με 2 μεταβλητές το πόσο έχει χτιστεί η φωτιά από πέτρες και πόσο από ξύλα με 0 γιατί τώρα ξεκινάει η αποστολή αυτή, αρχικοποιούνται τα 2 μηνύματα για το πόσες πέτρες και πόσα ξύλα θέλει ακόμη η φωτιά, ορίζεται η απόσταση της κάμερας, αρχικοποιούνται τα 8 κομμάτια της διάφανης – άκτιστης φωτιάς (4 για τις πέτρες, 4 για τα ξύλα) και τέλος αρχικοποιούνται τα 8 κομμάτια της αληθινής φωτιάς (4 για τις πέτρες, 4 για τα ξύλα).

Στην μέθοδο **Update**(): από το προηγούμενο script έχει δοθεί το σήμα πως μπορεί ο παίκτης να συλλέξει ξύλα και πέτρες, οπότε μπορεί να εκτελεστεί ο κώδικας συλλογής ξύλων και πετρών.

Αν ο παίκτης πατήσει το κλικ, του επιτρέπεται να συλλέγει αντικείμενα από το έδαφος και αν η φωτιά δεν έχει χτιστεί πλήρως, τότε συνεχίζεται η διαδικασία.

Δημιουργείται, με το κλικ, μία ακτίνα από την κάμερα που εκτοξεύεται πάνω στον κόσμο **2** αγνοώντας αν θα χτυπήσει τον παίκτη, η οποία, οτιδήποτε και αν χτυπήσει, γίνονται οι εξής έλεγχοι: αν το 3D αντικείμενο που χτύπησε η ακτίνα, είναι εντός της απόστασης που ορίστηκε και το παιχνίδι δεν είναι σε παύση τότε ακολουθεί ο επόμενος έλεγχος.

Αν η ακτίνα χτύπησε ένα αντικείμενο με **tag** «**ksulo**» και ο παίκτης δεν κρατάει μέγιστο αριθμό ξύλων και η μεταβλητή **bool** ότι ο παίκτης χρειάζεται ξύλα είναι **true**, τότε εκτελείται η **collectWoodForFire**().

Η **collectWoodForFire**(), αυξάνει τα ξύλα που κουβαλάει ο παίκτης, ενημερώνεται στα δεξιά της οθόνης ο αριθμός ξύλων που κουβαλάει ο παίκτης, το ξύλο που πατήθηκε στο έδαφος, εξαφανίζεται και διαμορφώνεται κατάλληλα το μήνυμα για το πόσα ξύλα συνολικά θέλει ο παίκτης ακόμη.

Αν η ακτίνα χτύπησε ένα αντικείμενο με **tag** «**petra**» και ο παίκτης δεν κρατάει μέγιστο αριθμό από πέτρες και η μεταβλητή **bool** ότι ο παίκτης χρειάζεται πέτρες είναι **true**, τότε εκτελείται η **collectStonesForFire**().

Η **collectStonesForFire**(), αυξάνει τις πέτρες που κουβαλάει ο παίκτης, ενημερώνεται το μήνυμα για τις πέτρες που κουβαλάει ο παίκτης στα δεξιά της οθόνης, η πέτρα που πατήθηκε στο έδαφος εξαφανίζεται και διαμορφώνεται κατάλληλα το μήνυμα για το πόσες πέτρες συνολικά θέλει ο παίκτης ακόμη.

Αν η ακτίνα χτύπησε ένα αντικείμενο με **tag** «**fire**», δηλαδή τη φωτιά που προσπαθεί να χτίσει, εκτελείται η συνάρτηση **buildFire**(). Αυτή η συνάρτηση, ελέγχει αν δεν έχει χτιστεί η φωτιά πλήρως από ξύλα ή/και πέτρες και παίρνει από τον παίκτη τις πέτρες του (αν έχει) και τα ξύλα του (αν έχει) και εκτελούνται αντίστοιχα οι: **xtiseMePetres**() και **xtiseMeKsula**(). Στο τέλος αυτής της συνάρτησης, εκτελείται πάντα μία ακόμη συνάρτηση, η **checkIfFireIsComplete**().

Η συνάρτηση **xtiseMePetres**(), προσθέτει όσες πέτρες έχει ο παίκτης πάνω του και το πόσο έχει χτιστεί η φωτιά από πέτρες. Αν το ποσό αυτό είναι μικρότερο του 4, τότε η φωτιά χτίζεται κατά όσες πέτρες κουβαλάει ο παίκτης πάνω του. Αν το ποσό αυτό είναι μεγαλύτερο ή ίσο του 4, τότε αυτό σημαίνει ότι ο παίκτης κουβαλάει περισσότερες πέτρες απ' ότι χρειάζεται η φωτιά, οπότε χτίζεται ολόκληρη από πέτρες. Με τον όρο χτίζεται, σημαίνει πως τα διάφανα – άκτιστα κομμάτια πετρών εξαφανίζονται και εμφανίζονται τα αντίστοιχα πραγματικά κομμάτια. Αν η φωτιά έχει χτιστεί πλήρως από πέτρες, εξαφανίζονται οι πληροφορίες για το πόσες πέτρες χρειάζονται ακόμη και το πόσες πέτρες κουβαλάει ο παίκτης πάνω του, στα δεξιά της οθόνης. Επίσης, η μεταβλητή **bool** για το αν χρειάζονται πέτρες γίνεται **false**.

Η συνάρτηση **xtiseMeKsula**(). προσθέτει όσα ξύλα έχει ο παίκτης πάνω του και το πόσο έχει χτιστεί η φωτιά από ξύλα. Αν το ποσό αυτό είναι μικρότερο του 4, τότε η φωτιά χτίζεται κατά όσα ξύλα κουβαλάει ο παίκτης πάνω του. Αν το ποσό αυτό είναι μεγαλύτερο ή ίσο του 4, τότε αυτό σημαίνει ότι ο παίκτης κουβαλάει περισσότερα ξύλα απ' ότι χρειάζεται η φωτιά, οπότε χτίζεται ολόκληρη από ξύλα. Με τον όρο χτίζεται, σημαίνει πως τα διάφανα – άκτιστα κομμάτια ξύλων εξαφανίζονται και εμφανίζονται τα αντίστοιχα πραγματικά κομμάτια. Αν η φωτιά έχει χτιστεί πλήρως από ξύλα, εξαφανίζονται οι πληροφορίες για το πόσα ξύλα χρειάζονται ακόμη και το πόσα ξύλα κουβαλάει ο παίκτης

πάνω του, στα δεξιά της οθόνης. Επίσης, η μεταβλητή **bool** για το αν χρειάζονται ξύλα γίνεται **false**.

Η συνάρτηση **checkIfFireIsComplete**() ελέγχει αν δεν χρειάζονται ούτε πέτρες, ούτε ξύλα. Αν αυτή η συνθήκη είναι αληθής, τότε μπορεί ο παίκτης να πάει στην επόμενη φάση. Δηλαδή, στα δεξιά της οθόνης, εξαφανίζονται όλες οι πληροφορίες για το αν απομένουν ξύλα ή πέτρες για την ολοκλήρωση της φωτιάς, εξαφανίζεται το βοηθητικό κουμπί, εμφανίζεται η διεπαφή για το άναμμα της φωτιάς με τον αναπτήρα, η **bool** μεταβλητή που αποθηκεύει αν η φωτιά έχει χτιστεί γίνεται **true**, δεν επιτρέπεται να μαζέψει ο παίκτης άλλα ξύλα και άλλες πέτρες και τέλος το παιχνίδι κάνει παύση.

Στην πρώτη διεπαφή, ζητείται από τον παίκτη, με τη χρήση του αναπτήρα να ανάψει τη φωτιά. Όταν ο παίκτης πατήσει το κουμπί «**Άναψε**», εκτελείται η συνάρτηση **lightTheBonfire**() και συμβαίνουν τα εξής: το θαυμαστικό πάνω από τη φωτιά εξαφανίζεται, το εφέ της φωτιάς ενεργοποιείται, η διεπαφή κλείνει και εμφανίζεται μια νέα που περιγράφει για τη λήψη του νερού.

Στην νέα διεπαφή με τη περιγραφή του νερού, όταν πατηθεί «**Συνέχεια**», εκτελείται η **waterStartWindowContinue**() και εκτελούνται τα εξής: εμφανίζεται στην άκρη της λίμνης ένα θαυμαστικό, εμφανίζεται στα δεξιά της οθόνης το βοηθητικό κουμπί, κλείνει το παράθυρο με τη περιγραφή του νερού, εμφανίζεται στον παίκτη ο κουβάς και το παιχνίδι βγαίνει από την παύση.

Λήψη νερού

WaterCollect.cs

Όταν ο παίκτης πλησιάσει δίπλα στο νερό, ενεργοποιείται η **OnTriggerEnter**() του script αυτού. Αυτή η συνάρτηση εμφανίζει το παράθυρο με τη συλλογή νερού από τη λίμνη και κάνει το παιχνίδι παύση.

Από αυτήν τη διεπαφή, αν πατηθεί συνέχεια, εκτελείται η **collectWaterWindowContinue**() το θαυμαστικό πάνω από τη λίμνη εξαφανίζεται, το πάνω κομμάτι του πάγου της λίμνης εξαφανίζεται και εμφανίζεται ο σπασμένος πάγος, το νερό μέσα στον κουβά εμφανίζεται, εξαφανίζεται το βοηθητικό κουμπί, κλείνει το παράθυρο με τη συλλογή

νερού και ανοίγει η επόμενη διεπαφή που καθοδηγεί τον παίκτη να πάει στη φωτιά (το παιχνίδι είναι ακόμη σε παύση).

Όταν ο παίκτης είναι έτοιμος και πατήσει συνέχεια, εκτελείται η **goToFireAgainContinue**(). Αυτή η συνάρτηση με τη σειρά της, κλείνει την τελευταία διεπαφή, ενεργοποιεί νέο βοηθητικό κουμπί, ενεργοποιεί ένα θαυμαστικό πάνω από τη φωτιά και βγάζει το παιχνίδι από παύση.

Αν ο παίκτης χαθεί, με το πάτημα του κουμπιού στο δεξιό μέρος της οθόνης, θα ανοίξει η διεπαφή βοήθειας. Με το πάτημα του κουμπιού «**Συνέχεια**», το βοηθητικό η διεπαφή κλείνει.

LeaveTheBucketToFire.cs

Στην αρχή αυτού του script, εντός της **Start**() συνάρτησης απενεργοποιείται η επόμενη φάση (η δημιουργία του S.O.S.)

Όταν ο παίκτης πλησιάσει τη φωτιά για να αφήσει τον μεταλλικό κουβά με το νερό, εκτελείται η **OnTriggerEnter**() και συμβαίνουν τα εξής: εμφανίζεται ένα παράθυρο επιτυχίας, ο κουβάς στην πλάτη του παίκτη εξαφανίζεται, το βοηθητικό κουμπί στα δεξιά της οθόνης εξαφανίζεται, εμφανίζεται ο κουβάς με το νερό πάνω στη φωτιά και τέλος, το παιχνίδι κάνει παύση.

Όταν στο παράθυρο επιτυχίας, πατηθεί «**Συνέχεια**», εκτελείται η **leaveTheBucketWindowContinue**(). Αυτή η συνάρτηση κλείνει τη διεπαφή επιτυχίας, εμφανίζει τη διεπαφή για τη δημιουργία του S.O.S. και τέλος εξαφανίζει το θαυμαστικό πάνω από τη φωτιά.

Με το πάτημα «**Συνέχεια**» στην διεπαφή της δημιουργίας του S.O.S., εκτελείται η **continueToSOS**(). Αυτή η μέθοδος, κλείνει το τελευταίο παράθυρο, ενεργοποιεί τα γράμματα του S.O.S. στο έδαφος (μόνο τα διάφανα – άκτιστα είναι ορατά), εμφανίζεται στα δεξιά της οθόνης το βοηθητικό κουμπί, ενεργοποιείται η ιδιότητα συλλογής αντικείμενων για τη δημιουργία του S.O.S. , εμφανίζεται στα δεξιά της οθόνης το πόσες πέτρες χρειάζεται ο παίκτης ακόμη και το πόσες πέτρες κρατάει πάνω του εκείνη τη στιγμή, επιτρέπεται η συλλογή αντικειμένων από το έδαφος, και το παιχνίδι βγαίνει από την κατάστασης παύσης.

CreateSOS.cs

Στην αρχή του Script, στην συνάρτηση **Start**(), εκχωρούνται σε έναν πίνακα 9 θέσεων, τα διάφανα – άκτιστα κομμάτια του S.O.S. και σε έναν άλλον, τα κανονικά κομμάτια του S.O.S., ενημερώνεται το μήνυμα για το πόσες πέτρες υπολείπονται στα δεξιά της οθόνης, ορίζεται η απόσταση που μπορεί ο παίκτης να σηκώνει αντικείμενα, αρχικοποιείται το πόσο έχει χτιστεί το S.O.S., αρχικοποιείται το πόσες πέτρες κρατάει ο παίκτης πάνω του και τέλος, αρχικοποιείται το πόσες πέτρες του επιτρέπεται να κουβαλάει.

Στην συνάρτηση **Update**() γίνεται ο εξής έλεγχος, αν ο παίκτης έχει αφήσει τον κουβά και έχει ενεργοποιηθεί η φάση της δημιουργίας του S.O.S. , τότε μπορούν να εκτελεστούν και τα επόμενα: αν ο παίκτης πατήσει το κλικ και του επιτρέπεται να συλλέγει αντικείμενα, εκτοξεύεται μία αόρατη ακτίνα, η οποία αγνοεί – διαπερνάει μέσα από τον παίκτη. Αν αυτή η ακτίνα χτυπήσει κάτι εντός της απόστασης και το παιχνίδι δεν είναι σε παύση, τότε γίνεται ο εξής έλεγχος.

Αν ο παίκτης έκανε κλικ στο αντικείμενο με **tag** «**petra**», και δεν κουβαλάει πάνω του τον μέγιστο αριθμό πετρών, τότε εκτελείται η **collectPetres**(). Οι πέτρες που κουβαλάει ο παίκτης αυξάνονται κατά **1**, το ποσό πετρών που κουβαλάει ο παίκτης στα δεξιά της οθόνης αυξάνεται κατά **1**, η πέτρα που έγινε κλικ πάνω της εξαφανίζεται και τέλος ενημερώνεται στα δεξιά της οθόνης το μήνυμα για το πόσες πέτρες χρειάζονται συνολικά.

Αν ο παίκτης έκανε κλικ στο αντικείμενο με **tag** «**SOS**», τότε εκτελείται η **buildSOS**(). Αυτή η συνάρτηση προσθέτει όσες πέτρες έχει ο παίκτης πάνω του και το πόσο έχει χτιστεί το S.O.S. από πέτρες. Αν το ποσό αυτό είναι μικρότερο του 9, τότε το S.O.S. χτίζεται κατά όσες πέτρες κουβαλάει ο παίκτης πάνω του. Αν το ποσό αυτό είναι μεγαλύτερο ή ίσο του 9, τότε αυτό σημαίνει ότι ο παίκτης κουβαλάει ακριβώς των αριθμό πετρών ή περισσότερες πέτρες απ’ ότι χρειάζεται το S.O.S., οπότε χτίζεται ολόκληρο από πέτρες. Με τον όρο χτίζεται, σημαίνει πως τα διάφανα – άκτιστα κομμάτια πετρών εξαφανίζονται και εμφανίζονται τα αντίστοιχα πραγματικά κομμάτια. Στην συνέχεια αποθηκεύεται το πόσο έχει χτιστεί το S.O.S. συνολικά, οι πέτρες που κουβαλάει ο παίκτης γίνονται 0, και

ενημερώνεται το κείμενο στα δεξιά. Αν το πόσο έχει χτιστεί είναι μεγαλύτερο ή ίσο από 9, τότε εκτελείται η **completedSOS**().

Η **completedSOS**(), είναι η μέθοδος που εκτελείται όταν ολοκληρώνεται το S.O.S. και υλοποιεί: την εξαφάνιση του θαυμαστικού πάνω από το S.O.S., την εξαφάνιση του βοηθητικού κουμπιού, την εξαφάνιση όλων των πληροφοριών για την ολοκλήρωση του S.O.S. στα δεξιά της οθόνης, την εμφάνιση του παραθύρου ότι ο παίκτης τα κατάφερε, το παιχνίδι κάνει παύση και του επιτρέπεται να σηκώσει αντικείμενα από το έδαφος. Αν πατήσει να ξαναμπεί στο παιχνίδι και φτάσει στον 2ο κόσμο, χωρίς αυτήν την συνάρτηση, είναι ανέφικτο για τον παίκτη να επιλέξει ή συλλέξει αντικείμενα.

Τελική φάση – άφιξη ελικοπτέρου

Όταν ο παίκτης καταφέρει να ολοκληρώσει το S.O.S., έρχεται δίπλα στη λίμνη ένα ελικόπτερο. Οι λειτουργίες αυτής της φάσης θα αναλυθούν μέσω των επομένων scripts.

HelicopterPhase.cs

Στην αρχή αυτού του script, στην συνάρτηση **Start**(), δηλώνεται πως ο παίκτης δεν έχει πλησιάσει το ελικόπτερο. Είναι βασικό αυτό το κομμάτι γιατί αν ο παίκτης παίξει για 2η φορά το παιχνίδι, το ελικόπτερο κατεβαίνει προς τα κάτω χωρίς να το πλησιάσει ο παίκτης.

Στην διεπαφή ολοκλήρωσης του S.O.S. (στην προηγούμενη φάση αυτού του κόσμου), αν ο παίκτης πατήσει «**Πήγαινε στο ελικόπτερο**», εκτελείται η **victoryWindowContinue**(). Αυτή η συνάρτηση κλείνει την διεπαφή, εμφανίζει όλα τα 3D αντικείμενα που έχουν να κάνουν με το ελικόπτερο, εμφανίζεται στα δεξιά της οθόνης βοηθητικό κουμπί και τέλος το παιχνίδι βγαίνει από παύση.

Στην συνάρτηση **Update**(), αν ο η συνάρτηση **getPlayerEntered**() επιστρέψει **true**, το ελικόπτερο προσγειώνεται με ταχύτητα **landSpeed** στο έδαφος. Αν η θέση του ελικοπτέρου στον άξονα **Y** είναι μικρότερη από **7.70** (θέση εδάφους), το ελικόπτερο σταματάει εκεί.

Αν ο παίκτης πλησιάσει το ελικόπτερο, από το έδαφος κατά ύψος, τότε ενεργοποιείται η **OnTriggerEnter**(). Μετά εκτελούνται τα

ακόλουθα: η **setPlayerEntered**() ενεργοποιεί την **bool** μεταβλητή σε **true** ώστε να κατέβει το ελικόπτερο στη γη, εμφανίζεται το παράθυρο αναχώρησης και εξαφανίζεται το θαυμαστικό κάτω από το ελικόπτερο.

Αν ο παίκτης απομακρυνθεί από το ελικόπτερο ενεργοποιείται η **OnTriggerExit**(), η οποία κλείνει το παράθυρο επιλογής για αναχώρηση. Ο παίκτης πρέπει να είναι κοντά στο ελικόπτερο για να μπορέσει να αποχωρήσει με αυτό.

Αν ο παίκτης είναι δίπλα στο ελικόπτερο και πατήσει παραμονή, εκτελείται η **stayMountainButton**() και ο παίκτης απλά παραμένει στο βουνό (κόσμο **2**).

Αν ο παίκτης πατήσει αποχώρηση για να φύγει από τον κόσμο 2 με το ελικόπτερο, ο χρόνος σταματάει να μετράει και φορτώνεται η τελευταία σκηνή του παιχνιδιού που δείχνει τα σκορ του παίκτη.

Αν ο παίκτης χαθεί, με το πάτημα του κουμπιού στο δεξιό μέρος της οθόνης, θα ανοίξει η διεπαφή βοήθειας. Με το πάτημα του κουμπιού «**Συνέχεια**», το βοηθητικό παράθυρο κλείνει.

RotateBigHelix.cs

Το script που περιστρέφεται ο μεγάλος έλικας. Όσο η ταχύτητα του ελικοπτέρου δεν είναι 0, αυτό σημαίνει ότι το ελικόπτερο πετάει ή προσγειώνεται και ο μεγάλος έλικας γυρίζει ακόμη.

RotateSmallHelix.cs

Το script που περιστρέφεται ο μικρός έλικας. Όσο η ταχύτητα του ελικοπτέρου δεν είναι 0, αυτό σημαίνει ότι το ελικόπτερο πετάει ή προσγειώνεται και ο μικρός έλικας γυρίζει ακόμη.

Άλλα χρήσιμα scripts για τον κόσμο **2**

OnMouseHoverInfoButton.cs

Αν ο παίκτης φέρει τον κέρσορα του ποντικιού πάνω από οποιοδήποτε παράθυρο/διεπαφή απενεργοποιείται η δυνατότητα να κάνει κλικ πάνω σε αντικείμενα του κόσμου **2** για να τα συλλέξει. Αυτό το script

τοποθετήθηκε και ενεργοποιήθηκε μέσα σε όλα τα αντικείμενα τύπου παραθύρων – διεπαφών ώστε να μην μπορούν να διαπερνάνε οι αόρατες ακτίνες επιλογής μέσα από τα παράθυρα/διεπαφές και να συλλέγει ή επιλέγει ο παίκτης αντικείμενα. Αυτή η λειτουργία εκτελείται στην συνάρτηση **OnPointerEnter**().

Όταν το ποντίκι του παίκτη είναι εκτός από παράθυρα/διεπαφές, μπορεί να ξανά επιλέξει – συλλέξει αντικείμενα. Αυτό επιτυγχάνεται με την συνάρτηση **OnPointerExit**().

TimeClock.cs

Το script αυτό, είναι υπεύθυνο να χρονομετρεί τον παίκτη από τη στιγμή που θα επιλέξει 4 από τα 9 αντικείμενα στον κόσμο **2**. Όταν ο χρόνος σταματήσει, αποθηκεύεται στη μεταβλητή **lastTime**, ο χρόνος ολοκλήρωσης του κόσμου **2** και ο χρόνος μηδενίζεται για την περίπτωση που ο παίκτης θελήσει να ξανά παίξει.

3.11.5. Τα script της τελικής σκηνής

Τα script της τελικής σκηνής αναλύονται σε αυτό το κεφάλαιο και είναι:

- AnswerChecker.cs
- RotateCameraTelikhSkhnh.cs

AnswerChecker.cs

Το script αυτό ελέγχει το ποιες ερωτήσεις στον κόσμο **1** απαντήθηκαν σωστά, ποιες λάθος, τον χρόνο που ολοκληρώθηκε ο κόσμος **2** και τα παρουσιάζει όλα στην τελική σκηνή.

Η συνάρτηση **checkFireQuestion**() ελέγχει αν η ερώτηση φωτιάς απαντήθηκε σωστά ή λάθος, ενεργοποιεί και απενεργοποιεί τα αντίστοιχα **TextBoxes** (Σωστό ή Λάθος).

Η συνάρτηση **checkEarthquakeQuestion**() ελέγχει αν η ερώτηση σεισμού απαντήθηκε σωστά ή λάθος, ενεργοποιεί και απενεργοποιεί τα αντίστοιχα **TextBoxes** (Σωστό ή Λάθος).

Η συνάρτηση **checkVolcanoQuestion**() ελέγχει αν η ερώτηση ηφαιστείου απαντήθηκε σωστά ή λάθος, ενεργοποιεί και απενεργοποιεί τα αντίστοιχα **TextBoxes** (Σωστό ή Λάθος).

Η συνάρτηση **showTheTime**(), εμφανίζει τον χρόνο σε ένα **TexBox** στο δεξιό μέρος της οθόνης.

Στην Τελική σκηνή, το μόνο διαθέσιμο κουμπί, είναι το «**Επιστροφή στο κύριο μενού**». Όταν πατηθεί, εκτελείται η **goToMainMenu**(), η οποία με τη σειρά της εκτελεί τα εξής: βγάζει το παιχνίδι από παύση σε περίπτωση που είχε μπει, την **resetAll**() και τη μετάβαση στο αρχικό μενού.

Η **resetAll**() επαναφέρει όλες τις **bool** μεταβλητές στην αρχική τους κατάσταση ώστε αν ξαναμπεί ο παίκτης να παίξει από την αρχή, να βρει όλες τις ερωτήσεις αναπάντητες.

RotateCameraTelikhSkhnh.cs

Το script αυτό, στην τελική σκηνή, γυρίζει την κάμερα γύρω γύρω για να δίνει την ψευδαίσθηση στον παίκτη ότι κοιτώντας τον ουρανό, η Γη γυρίζει.

4. Μελλοντικές προκλήσεις

Άλλες πρόσθετες ιδιότητες που θα μπορούσαν να ενσωματωθούν στο βιντεοπαιχνίδι *The Two Worlds* θα μπορούσαν να είναι:

- Μία ή περισσότερες καινούρια/ες σκηνή/ές με καινούριες ερωτήσεις ή προκλήσεις.
- Η δυνατότητα να μπορούν να μπουν 2 ή περισσότεροι παίκτες. Και μέσω συνεργασίας να προσπαθούν να απαντήσουν σε ερωτήσεις ή προκλήσεις. Π.χ. ο ένας παίκτης πατάει συνεχόμενα ένα κουμπί, ένας άλλος κρατάει ένα τιμόνι, και ένας άλλος φέρνει καύσιμα.
- Η ερώτηση στο νησί της φωτιάς να μην απαντάται σε ένα παράθυρο, αλλά όπως και στις ερωτήσεις του σεισμού και του ηφαιστείου, να πρέπει ο παίκτης να επιλέξει να πάει δίπλα σε κάποιο χώρο.
- Υποστήριξη χειριστηρίου.
- Να μπορεί το βιντεοπαιχνίδι να παίξει σε κινητά τηλέφωνα.
- Καλύτερα γραφικά και εφέ.
- Με τη βοήθεια του scroll wheel, να γίνεται η κάμερα πρώτου προσώπου.
- Προσθήκη ήχου.
- Φωτισμός για καλύτερες σκιές.
- Αλλαγή ημέρας και νύχτας και τη νύχτα να ανάβουν φώτα στις περιοχές.
- Αέρας στα δέντρα.
- Περισσότερες επιπτώσεις στον κόσμο ανάλογα με τις απαντήσεις του παίκτη.

5. Πηγές

[1] "2022 Essential Facts About the Video Game Industry," Entertainment Software Association ESA, 2022. [Online]. Available: https://www.theesa.com/resource/2022-essential-facts-about-the-video-game-industry/. [Accessed 22 September 2022].

[2] "Video Game," Wikipedia, 7 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Video_game#Effects_on_society. [Accessed 22 September 2022].

[3] "Παιχνίδι (δραστηριότητα)," Βικιπαίδεια, 12 Σεπτέμβριος 2022. [Online]. Available: https://el.wikipedia.org/wiki/%CE%A0%CE%B1%CE%B9%CF%87%CE%BD%CE%AF%CE%B4%CE%B9_(%CE%B4%CF%81%CE%B1%CF%83%CF%84%CE%B7%CF%81%CE%B9%CF%8C%CF%84%CE%B7%CF%84%CE%B1). [Accessed 22 Σεπτέμβριος 2022].

[4] "Educational Game," Wikipedia, 12 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Educational_game. [Accessed 22 September 2022].

[5] "Video game genre," Wikipedia, 11 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Video_game_genre. [Accessed 22 September 2022].

[6] "Serious game," Wikipedia, 19 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Serious_game. [Accessed 23 September 2022].

[7] M. e. a. Chang, Learning by Playing. Game-based Education System Design and Development, Canada: Springer Berlin, Heidelberg, 2009.

[8] "11 Positive Effects of Video Games," Game Quitters, [Online]. Available: https://gamequitters.com/positive-effects-of-video-games/. [Accessed 22 September 2022].

[9] "Educational Video Game," Wikipedia, 21 June 2022. [Online]. Available: https://en.wikipedia.org/wiki/Educational_video_game. [Accessed 22 September 2022].

[10] "When was the first computer invented?," Computer Hope, 30 December 2021. [Online]. Available: https://www.computerhope.com/issues/ch000984.htm. [Accessed 23 September 2022].

[11] "Electronic Games," Wikipedia, 23 July 2022. [Online]. Available: https://en.wikipedia.org/wiki/Electronic_game. [Accessed 23 September 2022].

[12] "Video Games Day," Pinterest, [Online]. Available: https://gr.pinterest.com/pin/736408976564691538/. [Accessed March 2022].

[13] "AAA (video game industry)," Wikipedia, 23 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/AAA_(video_game_industry). [Accessed 23 September 2022].

[14] "List of best-selling video games," Wikipedia, 19 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_best-selling_video_games. [Accessed 23 September 2022].

[15] "Atari 2600," Wikipedia, 29 August 2022. [Online]. Available: https://en.wikipedia.org/wiki/Atari_2600. [Accessed March 2022].

[16] "Game engine," Wikipedia, 20 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Game_engine#History. [Accessed 24 September 2022].

[17] "List of game engines," Wikipedia, 15 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_game_engines. [Accessed 25 September 2022].

[18] "Quick guide to the Unity Asset Store," Unity, [Online]. Available: https://unity3d.com/quick-guide-to-unity-asset-store . [Accessed 27 September 2022].

[19] "Materials, Shaders & Textures," Unity, 12 July 2017. [Online]. Available:

https://docs.unity3d.com/560/Documentation/Manual/Shaders.html. [Accessed march 2022].

[20] "System requirements for Unity 2020.1," Unity, 24 February 2021. [Online]. Available: https://docs.unity3d.com/2020.1/Documentation/Manual/system-requirements.html. [Accessed 27 September 2021].

[21] "Unity 5 UI Fill Any Screen," Unity, 24 April 2015. [Online]. Available: https://answers.unity.com/questions/953327/unity-5-ui-fill-any-screen.html. [Accessed March 2021].

[22] "START MENU in Unity," YouTube, 29 November 2017. [Online]. Available: https://www.youtube.com/watch?v=zc8ac_qUXQY. [Accessed March 2022].

[23] "Roboto Condensed," Google Fonts, [Online]. Available: https://fonts.google.com/specimen/Roboto+Condensed?subset=greek-ext#standard-styles. [Accessed march 2022].

[24] "Plain Color Wallpaper Backgrounds," Wallpaper-House.com, 2021. [Online]. Available: https://wallpaper-house.com/group/plain-color-wallpaper-background/index.php. [Accessed March 2022].

[25] "THIRD PERSON MOVEMENT in Unity," YouTube, 24 May 2020. [Online]. Available: https://www.youtube.com/watch?v=4HpC--2iowE. [Accessed March 2021].

[26] "Export FBX from Unity - Version 2020.2.1f1," YouTube, 15 February 2021. [Online]. Available: https://www.youtube.com/watch?v=UwNC6VYHkjI&ab_channel=Jaflanson. [Accessed March 2022].

[27] "Shaders and Materials," Unity, [Online]. Available: https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@7.1/manual/shaders-in-universalrp.html. [Accessed March 2022].

[28] "URPSpeedTree," Unity, [Online]. Available: https://docs.unity3d.com/Packages/com.unity.render-

pipelines.universal@7.1/manual/speedtree.html. [Accessed March 2022].

[29] "Visual Studio Communtiy," Microsoft, [Online]. Available: https://visualstudio.microsoft.com/vs/community/. [Accessed March 2022].

[30] "AUTODESK Tinkercad," AUTODESK, [Online]. Available: www.tinkercad.com. [Accessed March 2022].

[31] "How to Make Background Transparent in Paint (Windows 10)," YouTube, 11 July 2020. [Online]. Available: https://www.youtube.com/watch?v=WYp851hyDuU&ab_channel=AddictiveTipsTV. [Accessed March 2022].

[32] "Paint," Microsoft, [Online]. Available: https://apps.microsoft.com/store/detail/paint/9PCFS5B6T72H?hl=en-us&gl=us. [Accessed March 2022].

[33] "Skybox shaders," Unity, [Online]. Available: https://docs.unity3d.com/Manual/skybox-shaders.html. [Accessed March 2022].

[34] "Creating Fire, Smoke & Mist Effects with VFX Graph in Unity! (Tutorial)," YouTube, 31 August 2020. [Online]. Available: https://www.youtube.com/watch?v=OCzGXcdyqnQ&t=94s&ab_channel=Unity. [Accessed March 2022].

[35] "HIGH QUALITY TEXTURES," HIGH QUALITY TEXTURES, 19 November 2006. [Online]. Available: https://www.highqualitytextures.com/textures/displayimage.php?album=148&pid=1047. [Accessed March 2022].

[36] "FIRE AND SMOKE with Unity VFX Graph!," YouTube, 27 January 2019. [Online]. Available: https://www.youtube.com/watch?v=R6D1b7zZHHA&t=269s&ab_channel=Brackeys. [Accessed March 2022].

[37] "EZ EXPLOSIONS | UNITY," YouTube, 16 April 2021. [Online]. Available: https://www.youtube.com/watch?v=dOnQY0t3TBM&ab_channel=RAZDOLBAYS. [Accessed March 2022].

[38] "Unity: Camera Shake using Cinemachine Impulse [Tutorial(better quality)]," YouTube, 19 March 2021. [Online]. Available: https://www.youtube.com/watch?v=AhLnbRw0SuQ&ab_channel=AntnL. [Accessed March 2022].

[39] "Clear Editor Console logs from script," stackoverflow, January 2017. [Online]. Available: https://stackoverflow.com/questions/40577412/clear-editor-console-logs-from-script. [Accessed March 2022].

[40] "Cinemachine input axis camera movement,cinemachine input axis control with script," Unity, 11 August 2020. [Online]. Available: https://answers.unity.com/questions/1759654/cinemachine-input-axis-camera-movementcinemachine.html. [Accessed March 2022].

[41] "Cursor.visible," Unity, 23 September 2022. [Online]. Available: https://docs.unity3d.com/ScriptReference/Cursor-visible.html. [Accessed March 2022].

[42] "Making a Character Jump (Unity Tutorial)," YouTube, 6 March 2021. [Online]. Available: https://www.youtube.com/watch?v=ynh7b-AUSPE&ab_channel=KetraGames. [Accessed March 2022].

[43] "THIRD PERSON MOVEMENT in Unity," YouTube, 24 May 2020. [Online]. Available: https://www.youtube.com/watch?v=4HpC--2iowE&ab_channel=Brackeys. [Accessed March 2022].

[44] "Unity: Camera Shake using Cinemachine Impulse [Tutorial(better quality)]," YouTube, 19 March 2021. [Online]. Available: https://www.youtube.com/watch?v=AhLnbRw0SuQ. [Accessed March 2022].

[45] "OBJECT POOLING in Unity," YouTube, 11 February 2018. [Online]. Available: https://www.youtube.com/watch?v=tdSmKaJvCoA&ab_channel=Brackeys. [Accessed March 2022].

[46] "ParticleSystem.Play," Unity, 23 September 2022. [Online]. Available:

https://docs.unity3d.com/ScriptReference/ParticleSystem.Play.html. [Accessed March 2022].

[47] "How To Move Obstacles Or Even Enemy In Sine Wave Way Up And Down With Simple C# Script In Unity Game," YouTube, 10 September 2020. [Online]. Available: https://www.youtube.com/watch?v=UVqH2XS5E2M&ab_channel=AlexanderZotov. [Accessed march 2022].

[48] "Click Objects In Unity - How To Click Objects in Unity 5 | Unity Mouse Tutorial," YouTube, 20 July 2017. [Online]. Available: https://www.youtube.com/watch?v=EANtTI6BCxk&ab_channel=Omnirift. [Accessed March 2022].

[49] "Oric," Wikipedia, 5 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/Oric. [Accessed 24 September 2022].

[50] "ZX Spectrum," Wikipedia, 4 September 2022. [Online]. Available: https://en.wikipedia.org/wiki/ZX_Spectrum. [Accessed 24 September 2022].

[51] "Αειφόρος ανάπτυξη," Βικιπαίδεια, 15 Σεπτέμβριος 2022. [Online]. Available: https://el.wikipedia.org/wiki/%CE%91%CE%B5%CE%B9%CF%86%CF%8C%CF%81%CE%BF%CF%82_%CE%B1%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7. [Accessed 24 Σεπτέμβριος 2022].