



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΔΙΚΤΥΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΜΟΣ, ΑΝΑΠΤΥΞΗ ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΕΙΚΟΝΙΚΟΥ
ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΓΙΑ ΤΗΝ ΠΡΟΣΟΜΟΙΩΣΗ
ΔΙΑΔΙΚΑΣΙΩΝ ΚΑΙ ΚΙΝΗΣΕΩΝ ΣΕ ΠΕΡΙΠΤΩΣΗ ΠΥΡΚΑΓΙΑΣ ΣΕ
ΕΣΩΤΕΡΙΚΟ ΧΩΡΟ**

Χαραλάμπους Βλάσιος

Επιβλέπων: Στύλιος Χρυσόστομος

Καθηγητής

Άρτα, Φεβρουάριος, 2022

**DESIGN AND DEVELOPMENT OF A VIRTUAL EDUCATIONAL
ENVIRONMENT TO SIMULATE MOVEMENTS AND PROCEDURES IN
CASE OF INDOOR FIRE**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Καθηγητής
Χρυσόστομος Στύλιος
2. Επίκουρη καθηγήτρια
Βασιλική Λιάγκου,
3. Αναπληρωτής καθηγητής
Ιωάννης Τσούλος,

© Χαραλάμπους Βλάσιος 2022

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Χαραλάμπους, Βλάσιος



Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θερμές ευχαριστίες στον καθηγητή κ. Χρυσόστομο Στύλιο για την γενική καθοδήγηση και επίβλεψη της εργασίας, καθώς και τον Δρ. κ. Τζέριες Μπεσαρατ για την τεχνική καθοδήγηση στην ανάπτυξη της εργασίας. Τέλος, θερμές ευχαριστίες, επίσης, και στην κ. Μπεσιάνα Τζάνη για την τεχνική υποστήριξη στο κομμάτι του FDS.

ΠΕΡΙΛΗΨΗ

Τα «Σοβαρά Παιχνίδια» (Serious Games), εξελίσσονται εδώ και δεκαετίες σε ψηφιακή μορφή, αλλά και αναλογική. Η προσέγγιση της εκπαίδευσης μέσω μιας διαδικασίας παιχνιδιού, έχει, σταθερά, ανοδική πορεία, ειδικότερα στην αγορά της εικονικής και επαυξημένης πραγματικότητας.

Σκοπός της διπλωματικής εργασίας, είναι η έρευνα και η ανάπτυξη μιας εκπαιδευτικής πλατφόρμας η οποία θα αποτελέσει μέρος της εκπαιδευτικής διδασκαλίας των νέων και ιδιαίτερα μαθητών σε ζητήματα πυρασφάλειας. Συνήθως για τους επαγγελματίες και εθελοντές πυροσβέστες, αυτή η προσομοίωση και εκπαίδευση γίνεται σε φυσικό περιβάλλον, είτε με πραγματική ελεγχόμενη φωτιά, είτε προβάλλοντας εκπαιδευτικό υλικό με τη χρήση πολυμεσικών εφαρμογών, είτε σχεδιάζοντας κάποιες διαδρομές διαφυγής, χωρίς όμως τη χρήση φωτιάς, λόγω του κινδύνου τραυματισμών. Στόχος της διπλωματικής είναι να σχεδιαστεί η πλατφόρμα σε εικονικό περιβάλλον, οπότε να μην υπάρχει κανένας περιορισμός στο σενάριο και οι μαθητές να έχουν αλληλεπίδραση με τη φωτιά -στον εικονικό κόσμο- όπου θα μπορούν είτε να προσπαθούν να σβήσουν τη φωτιά, είτε να διαφύγουν από το χώρο.

Λέξεις-κλειδιά: Εικονική πραγματικότητα, εκπαίδευση, πυρασφάλεια

ABSTRACT

Serious games have been progressing for decades, in digital and nondigital form. Using a game approach to train and educate people is steadily rising, mainly based on the market of virtual and augmented reality.

This master's thesis aims to research and develop an educational platform for young people's educational process, especially students, in fire safety. Usually, for the professional and volunteer firefighters, this simulation and training take part in the real world, either with a real controlled fire, either showing educational content using multimedia technologies, or by designing escaping routes because of the danger of potential injuries. With this thesis, we aim to design the platform in the virtual environment, so there is no limitation in the scenario design, and the students can interact with the fire — in the virtual world— where they can extinguish the fire escaping the room.

Keywords: Virtual Reality, Education, Fire Safety

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ *x*

ABSTRACT *xiii*

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ *xiv*

ΕΙΣΑΓΩΓΗ *Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.*

1	Εκτεταμένη Πραγματικότητα (XR — Extended Reality)	1
1.1	Εισαγωγή	2
1.2	Τι είναι η εκτεταμένη πραγματικότητα (Extended Reality — XR)	3
1.2.1	Εικονική Πραγματικότητα (VR — Virtual Reality)	3
1.3	Επαυξημένη Πραγματικότητα (AR — Augmented Reality)	11
1.4	Εισαγωγή στα Serious Games	16
1.4.1	Serious Games	16
1.4.2	Serious Games with Virtual Reality	19
1.4.3	Fire Safety/ Education with Serious Games/VR	20
1.5	Unity	24
1.5.1	Πλεονεκτήματα της Unity engine	25
1.6	Πως χρησιμοποιείται το Unity	29
1.6.1	Επιθεωρητής (Inspector)	31
1.6.2	Ιεραρχία (Hierarchy)	32
1.6.3	Scripts	32
1.7	Unity Physics	33
1.7.1	RigidBody	34
1.7.2	Colliders	34
1.7.3	Κάμερες (Cameras)	36
1.7.4	Φωτισμός (Lighting)	37
1.7.5	Σύστημα σωματιδίων — Particle System	40
1.8	Asset Store	41
2	Υλοποίηση	43

2.1	Εισαγωγή	43
2.2	FDS	44
2.3	Εργαλειοποίηση του FDS	45
2.4	Ροή ανάπτυξης της προσομοίωσης	46
2.4.1	FDS	46
2.4.2	Unity.....	46
3	Ανάπτυξη της προσομοίωσης στην Unity	46
3.1	Εισαγωγή	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
3.2	Σενάριο	48
3.2.1	Δεδομένα για την απομείωση της μάζας	50
3.2.2	Συνολική μάζα (Total mass)	51
3.2.3	Χρόνος πλήρους καύσης (Full burn time).....	52
3.2.4	Δεδομένα για την θερμοκρασία του φλεγόμενου αντικειμένου	53
3.2.5	Δημιουργία πλέγματος για τον έλεγχο της εξάπλωσης	60
3.2.6	Εκκίνηση σκηνής και εξάπλωση της φωτιάς	63
3.3	Πυρόσβεση	69
3.3.1	Παράμετροι Πυρόσβεσης	69
3.3.2	Κατηγορίες φωτιάς	70
3.3.3	Κατηγορίες πυροσβεστήρων	70
3.3.4	Χρόνος εκκένωσης Πυροσβεστήρων	72
3.3.5	Χωρητικότητα κατάσβεσης Πυροσβεστήρων.....	73
3.3.6	Σύστημα πόντων	75
3.3.7	Αλγόριθμοι πυρόσβεσης	76
4	Συμπερασμα/ Μελλοντική δουλειά	78
	ΒΙΒΛΙΟΓΡΑΦΙΑ	79

1 Εισαγωγή

Στη σύγχρονη εποχή η ανάπτυξη της τεχνολογίας έχει αλλάξει άρδην το περιβάλλον μέσα στο οποίο ζούμε, αλληλεπιδρούμε, μαθαίνουμε και αυτοπροσδιοριζόμαστε. Τα serious games ή αλλιώς τα εκπαιδευτικά παιχνίδια, αποτελούν με τη σειρά τους κομμάτι της τεχνολογικής ανάπτυξης και είναι στη ζωή μας ώστε να εισάγουν ενήλικες και παιδιά στη σύγχρονη εποχή του διαδικτύου. Βιωματική μάθηση και παιχνίδι προσαρμόζονται μέσω σύγχρονων εκπαιδευτικών διαδικασιών και της, πλέον, σύγχρονης τεχνολογίας, με το βλέμμα στραμμένο στο αύριο του νέου τεχνολογικού κόσμου, στην ανατολή του οποίου ζούμε σήμερα.

Μπορεί ο όρος «serious games» να αποτελεί σχήμα οξύμωρο με την πρώτη ανάγνωση, ωστόσο η πράξη έχει αποδείξει πως μάθηση μέσω παιχνιδιού όχι μόνο είναι μία πιθανή μέθοδος διδασκαλίας, αλλά και αρκούντως αποδοτική. Αποτελεί δε, μία συνεχώς εξελισσόμενη τάση στον τομέα της διδασκαλίας που έχει ήδη υιοθετηθεί από δημοτικά σχολεία έως και επιφανή πανεπιστήμια ανά τον κόσμο, όπως επίσης και από μεγάλες πολυεθνικές εταιρείες και ιδρύματα.

1.1 Στόχοι εργασίας και συνεισφορά

Ο σκοπός της εργασίας είναι να δημιουργήσει ένα καινοτόμο σύστημα εκπαίδευσης πυρκαγιάς σε εικονικό περιβάλλον, προκειμένου να εκπαιδεύσει ενδιαφερόμενο προσωπικό στην έγκαιρη και ακριβή απόκρισή τους. Η προσομοίωση μπορεί να πραγματοποιηθεί σε εικονικό περιβάλλον με την χρήση των απαραίτητων συσκευών, αλλά και με απλή χρήση ηλεκτρονικού υπολογιστή, όπως ένα τυπικό video game. Το έργο αναπτύσσει τρισδιάστατο εκπαιδευτικό υλικό με βασικά σενάρια εξάπλωσης πυρκαγιάς μέσω ψηφιακής προσομοίωσης.

Η εργασία εστιάζει στην δημιουργία ενός «serious game» —ένα εικονικό περιβάλλον προσομοίωσης, που έχει στοιχεία ενός video game— το οποίο μπορεί να διδάξει τους εκπαιδευόμενους, κάποιες βασικές αρχές γύρω από την συμπεριφορά της φωτιάς, καθώς η εξάπλωση και η κατάσβεση της φωτιάς εξαρτάται από πολλούς παράγοντες. Ο πιο σημαντικός είναι τα φλεγόμενα υλικά —ειδικότερα, το υλικό που φλέγεται δεν μπορεί να κατασβεστεί με οποιοδήποτε πυροσβεστήρα. Στην παρούσα εργασία χρησιμοποιούνται τέσσερις τύποι

πυροσβεστήρα (νερού, ξηράς σκόνης, διοξειδίου του άνθρακα, αφρού) και ο εκπαιδευόμενος καλείται να επιλέξει το σωστό υλικό για την κατάσβεση της φωτιάς.

Εν κατακλείδι, αυτή η εργασία φιλοδοξεί να συνεισφέρει αναπτύσσοντας ένα εικονικό περιβάλλον, μέσα στο οποίο προσομοιώνεται η κίνηση και εξάπλωση της φωτιάς, ενώ, επίσης, προσομοιώνεται και η συμπεριφορά της φωτιάς κατά την διαδικασία της κατάσβεσης.

Στο πλαίσιο της εργασίας, υπήρχε ανάγκη έρευνας και μελέτης της εξάπλωσης πυρκαγιάς. Αρχικά ήταν απαραίτητο να μελετηθούν οι παράγοντες που επηρεάζουν την ανάφλεξη και την εξάπλωση της φωτιάς. Λόγω της δυσκολίας αποτύπωσης του φαινομένου της φωτιάς με απλές μαθηματικές εξισώσεις, χρειάστηκε να γίνει έρευνα για την αναζήτηση του κατάλληλου εργαλείου όπου παρέχει τα απαραίτητα δεδομένα για την εξάπλωση της φωτιάς.

Τα τελευταία χρόνια είναι ευρέως διαδεδομένη η διαδικασία επίλυσης δυναμικών εξισώσεων με χρήση υπολογιστών, όπου αναφέρεται συνήθως ως Υπολογιστική Ρευστοδυναμικής (Computational Fluid Dynamics - CFD). Ένας CFD κώδικας βασίζεται στην προσέγγιση ότι ένας κλειστός χώρος χωρίζεται σε έναν μεγάλο αριθμό στοιχειωδών όγκων. Ο κώδικας επιλύει τις θεμελιώδεις εξισώσεις οι οποίες περιγράφουν την μεταφορά μάζας, ορμής και ενέργειας μεταξύ των στοιχειωδών αυτών όγκων προκειμένου να προβλεφθεί η ανάπτυξη της φωτιάς σε αυτόν τον χώρο.

Με βάση τη μελέτη αυτή, χρησιμοποιείται το ελεύθερο λογισμικό Fire Dynamics Simulator (FDS), όπου περισσότερες λεπτομέρειες και οδηγίες για την χρήση αυτού αναφέρονται στο Κεφάλαιο 2.2. Για την «οπτικοποίηση» αυτών των δεδομένων, μέσω ενός σεναρίου εικονικού περιβάλλοντος χρησιμοποιήθηκε η Unity 3D Engine, και η γλώσσα προγραμματισμού C#.

2 Εκτεταμένη Πραγματικότητα (Extended Reality - XR)

2.1 Εισαγωγή

Το 1965, σε μια διάλεξή του, ο Ivan Sutherland εξέφρασε ένα όραμα, το οποίο ο Frederick Brooks παραφράζει στη δημοσίευσή του [\[1\]](#):

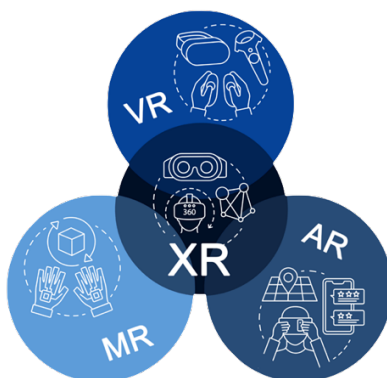
«Μην βλέπετε αυτό το πράγμα, απλά, ως μια οθόνη. Δείτε το ως ένα

παράθυρο· ένα παράθυρο που κάποιος κοιτάει σε ένα εικονικό κόσμο. Η πρόκληση για τα γραφικά των υπολογιστών είναι να κάνουν αυτόν τον εικονικό κόσμο να μοιάζει με τον πραγματικό, να ακούγεται πραγματικός, να αντιδρά στη διάδραση σε πραγματικό χρόνο, ακόμη και να τον αισθάνεσαι πραγματικό».

2.2 Τι είναι η εκτεταμένη πραγματικότητα (Extended Reality — XR)

Η εκτεταμένη πραγματικότητα (XR) είναι ένας όρος που αναφέρεται σε όλα τα πραγματικά/κανονικά ή συνδυαζόμενα περιβάλλοντα, και διαδράσεις ανθρώπου-μηχανής που δημιουργούνται από έναν υπολογιστή, ή φορέσιμες συσκευές. Το X, αναπαριστά την μεταβλητή για κάθε τρέχουσα, ή μελλοντική χωρική υπολογιστική τεχνολογία [2]. Το XR, είναι ένα υπερσύνολο το οποίο περιλαμβάνει τις τρεις επικρατούσες τεχνολογίες εκτεταμένης πραγματικότητας:

- **Εικονική Πραγματικότητα** (Virtual Reality - VR)
- **Μικτή Πραγματικότητα** (Mixed Reality - MR)
- **Επαυξημένη Πραγματικότητα** (Augmented Reality - AR)



Εικόνα 2-1: Τα επίπεδα της εκτεταμένης πραγματικότητας [2]

2.3 Εικονική Πραγματικότητα (Virtual Reality)

Η εικονική πραγματικότητα είναι μια πανίσχυρη τεχνολογία η οποία μας «υπόσχεται» πως θα μας αλλάξει την ζωή όσο καμία άλλη. Διεγείροντας, τεχνητά, τις αισθήσεις μας, τα σώματά μας ξεγελιούνται και αποδέχονται μια διαφορετική εκδοχή της πραγματικότητας. Η εικονική πραγματικότητα είναι σαν ένα όνειρο, το οποίο μπορεί να λάβει χώρα σε ένα κόσμο καρτούν, σε ένα άλλο σημείο της γης, ή σε άλλο σύμπαν [3].

Οι τεχνολογίες Εικονικής Πραγματικότητας άρχισαν να αναπτύσσονται στα

τέλη της δεκαετίας του '80, αλλά δεν κατάφεραν να φανούν «αντάξιος των προσδοκιών», καθώς ο εξοπλισμός ήταν πανάκριβος και αυτό είχε αντίκτυπο στους προγραμματιστές αλλά και τους πελάτες. Τα τελευταία χρόνια βιώνουμε μια αναγέννηση του VR, καθώς η τρομερή ζήτηση στον χώρο των smartphones, ώθησε τις εταιρείες σε καινοτομίες οι οποίες παράγουν, πλέον, χαμηλού κόστους-υψηλών επιδόσεων hardware, το οποίο μπορεί να χρησιμοποιηθεί και σε συσκευές VR, όπως οι συσκευές κεφαλής (VR headsets—Εικόνα 2-2).



Εικόνα 2-2 VR Headset [4]

Από το 2014 ως σήμερα, κορυφαίες εταιρείες του χώρου επενδύουν δισεκατομμύρια σε αυτές τις τεχνολογίες (VR, MR, AR) και ολόκληρο το οικοσύστημα αναπτύσσεται συνεχώς. Ένας ορισμός του VR —σύμφωνα με τον Steven M. LaValle [3]— είναι:

«Η πρόκληση στοχευμένης συμπεριφοράς σε έναν οργανισμό, χρησιμοποιώντας τεχνητή διέγερση των αισθήσεων, όταν ο οργανισμός έχει μικρή ή καθόλου επίγνωση αυτής της παρεμβολής». Η ιδέα της «εξαπάτησης» του οργανισμού μπορεί να αναλυθεί με όρους νευροβιολογίας, καθώς έρευνες έχουν δείξει ο τρόπος που οι νευρικές δομές κωδικοποιούν και αποκωδικοποιούν τον περιβάλλοντα χώρο μπορούν να ενεργοποιηθούν και κατά την διάρκεια του VR —κωδικοποιούν, δηλαδή, κάτι που δεν είναι πραγματικό.»

Στην αλληγορία της σπηλιάς [5] —στην Πολιτεία του Πλάτωνα— περιγράφεται η ιστορία κάποιων ανθρώπων, οι οποίοι από την γέννησή τους ήταν αλυσοδομεμένοι με στραμμένο το πρόσωπό τους σε έναν τοίχο, όπου έβλεπαν όλα τα αντικείμενα ως σκιές μέσω μιας φωτιάς που υπήρχε πίσω από το τοιχίο όπου ήταν

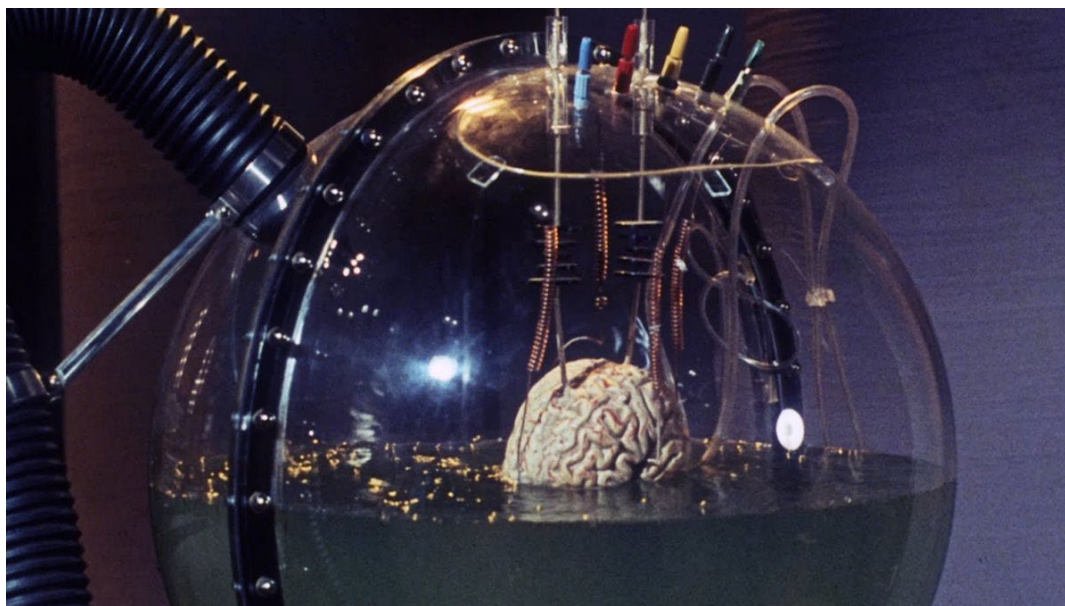
δεμένοι (Εικόνα 2-3). Πίσω από το τοιχίο και μπροστά από τη φωτιά, κάποιοι χρησιμοποιούσαν κάποια αντικείμενα για να δημιουργήσουν κάποιες αναπαραστάσεις αντικειμένων ενώ δημιουργούσαν και τους ήχους που ταίριαζαν στο κάθε αντικείμενο. Ο φιλόσοφος αναρωτιέται τι θα συνέβαινε αν κάποιος έσπαγε τα δεσμά του και έβλεπε τον πραγματικό κόσμο; Πως θα αντιδρούσαν οι υπόλοιποι, αν αυτός επέστρεφε και τους περιέγραφε τον «πραγματικό κόσμο»; Παρόλο που η ιστορία αυτή έχει φιλοσοφική χροιά, δεν μπορούμε να παραβλέψουμε πως είναι, ίσως, η πρώτη —έστω και έμμεση— αναφορά στην εικονική πραγματικότητα.



Εικόνα 2-3: Η αλληγορία του σπηλαίου [6]

Το 1641, ο René Descartes εξέφρασε τη θεωρία ενός δαίμονα [7] ο οποίος είχε ως μοναδικό σκοπό να εξαπατά τους ανθρώπους με την ψευδαίσθηση ενός εξωτερικού φυσικού κόσμου. Κάποιους αιώνες μετά, το 1973, ο Gilbert Hartman εισήγαγε την ιδέα του εγκεφάλου σε δοχείο (brain in a vat), το οποίο είναι ένα πείραμα που δείχνει πως ο δαίμονας του Descartes, θα μπορούσε να το πετύχει αυτό [8]. Ένας εγκέφαλος έχει αποκοπεί από το σώμα ενός ανθρώπου και έχει τοποθετηθεί σε μια γυάλα με υγρό. Αυτός ο εγκέφαλος είναι συνδεδεμένος με ένα έναν υπολογιστή, ο οποίος αναπαράγει τα ερεθίσματα που δέχεται ο εγκέφαλος από τα μάτια μας, ώστε να αναπαραστήσει τον κόσμο. Πάνω σε αυτή την υπόθεση

βασίζεται η θεωρία του Καρτεσιανού σκεπτικισμού.



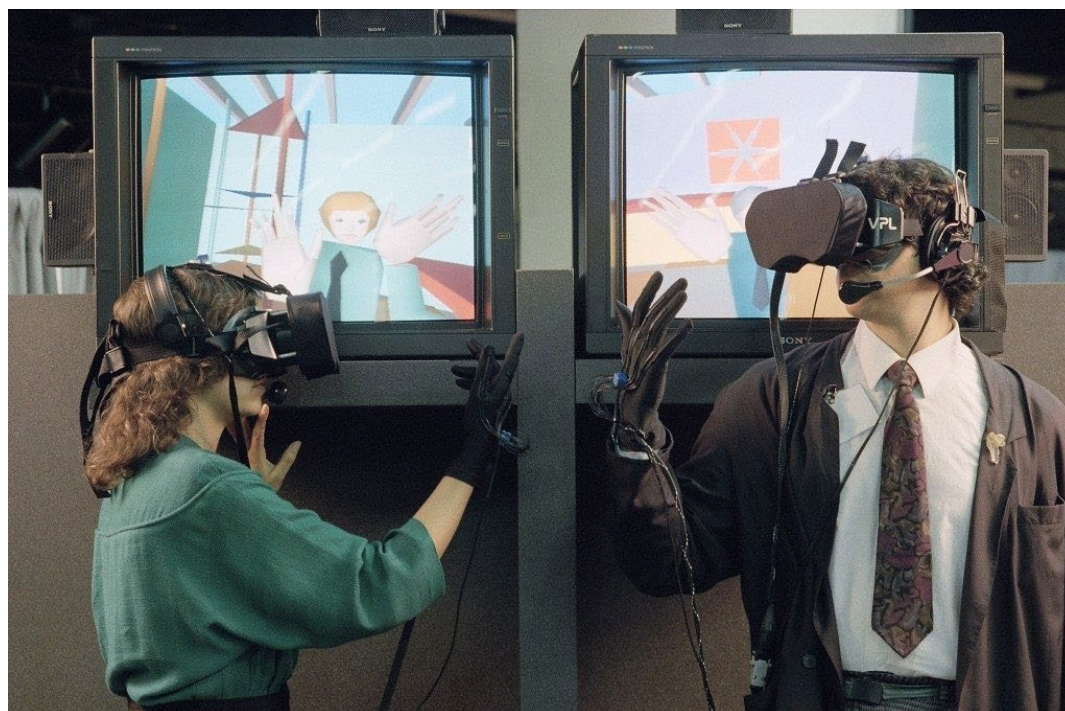
Εικόνα 2-4: Brain in a vat [9]

Πάνω σε αυτή τη θεωρία βασίστηκε και η διάσημη σειρά ταινιών Matrix, όπου οι μηχανές —οι οποίες έχουν δημιουργηθεί με την χρήση τεχνητής νοημοσύνης— έχουν εξαπατήσει το ανθρώπινο είδος, έχοντας συνδέσει τον εγκέφαλό τους σε μια προσομοίωση του κόσμου. Ο Νιο, ο κεντρικός χαρακτήρας, καλείται να διαλέξει μεταξύ δύο χαπιών —του μπλε, όπου θα διαγραφόταν η μνήμη του και θα ζούσε στην προσομοίωση χωρίς να το γνωρίζει, ή του κόκκινου, όπου θα καλούταν να αντιμετωπίσει την νέα πραγματικότητα.

Ο όρος εικονική πραγματικότητα εμφανίστηκε για πρώτη φορά, πριν από, περίπου, 200 χρόνια από τον Immanuel Kant, ο οποίος όμως δεν ενέπλεξε την τεχνολογία [3]. Ο Kant εισήγαγε τον όρο εικονική πραγματικότητα ως μια κατάσταση, όπου μια «πραγματικότητα» υπάρχει στο μυαλό κάποιου ανθρώπου— η οποία είναι διαφοροποιημένη από τον πραγματικό κόσμο— η οποία είναι, εξίσου, μια πραγματικότητα.

Ωστόσο, τον όρο εικονική πραγματικότητα με την τεχνολογική της έννοια, τον εισήγαγε για πρώτη φορά ο Jaron Lanier, ο οποίος θεωρείται ο «πατέρας» του VR. Η VPL Research ήταν από τις πρώτες εταιρείες που ανέπτυξαν VR προϊόντα [10], όπου, μεταξύ άλλων, ανέπτυξε γάντια (Εικόνα 2-5) τα οποία συνδεόταν με τον ηλεκτρονικό υπολογιστή και κατέγραφαν κινήσεις και προσανατολισμό. Αυτή η πληροφορία καταγράφονταν και επέστρεφε, ως είσοδος, στον υπολογιστή και αυτή μπορούσε να αποδοθεί εικονικά, με σκοπό το χειρισμό και την αλλαγή

προσανατολισμού των εικονικών αντικειμένων.



Εικόνα 2-5: Το EyeGlove, ένα από τα προϊόντα της VPL [11]

Επίσης, ανέπτυξαν HMD (Head Mount Display), συσκευές προβολής κεφαλής καθώς και ολόσωμη στολή με αισθητήρες κίνησης, προϊόντα τα οποία ήταν πρωτοποριακά για την δεκαετία του '80. Στις μέρες μας με την επανάσταση της τεχνολογίας και το, συνεχώς, αναβαθμισμένο hardware που αναπτύσσεται, πλέον είναι πολύ προσιτό για όλους να επενδύσουν σε αυτές τις τεχνολογίες.

Οι πιο πολλές εμπειρίες VR, προϋποθέτουν και την διαδραστικότητα. Όταν η διέγερση των αισθήσεων εξαρτάται από ενέργειες του χειριστή, το VR σύστημα ονομάζεται closed-loop, αλλιώς ονομάζεται open-loop [3]. Στην περίπτωση του closed-loop, ο χειριστής έχει μερικό έλεγχο στην διέγερση αισθήσεων, το οποίο ποικίλει λόγω των κινήσεων του σώματος, των ματιών, ακόμη και της θερμοκρασίας, ή της αγωγιμότητας του σώματος. Πριν αναφερθούμε στις εφαρμογές του VR στις μέρες μας, αξίζει να αναφερθεί πως, σε κάποιες περιπτώσεις, οι προσομοιώσεις σε VR μπορεί να δημιουργήσουν προβλήματα στους χρήστες. Η υπερδιέγερση των αισθήσεων δημιουργεί μια αίσθηση κόπωσης και ασθένειας, φαινόμενο που μπορεί να προκληθεί από κακή ανάπτυξη του λογισμικού, ή λόγω χρήσης ακατάλληλου hardware. Μία άλλη περίπτωση που παρατηρείται, είναι κατά την χρήση VR εξαρτημάτων τα οποία απαιτούν μυϊκή δύναμη, ή ακόμη, περιπτώσεις που λόγω του σεναρίου προσομοίωσης απαιτείται διαρκής κίνηση και

μπορεί να προκαλέσουν σωματική κόπωση στον χρήστη.

2.3.1.1 Εφαρμογές *Virtual Reality*

Στην εποχή μας όλο και περισσότερες εφαρμογές VR εμφανίζονται, καθώς το συνεχώς εξελισσόμενο hardware δημιουργεί νέες δυνατότητες, τις οποίες όλοι προσπαθούν να εκμεταλλευτούν για να φτιάξουν κάτι καινοτόμο.

Video Games

Ο πιο δημοφιλής τομέας του VR είναι τα video games. Οι χρήστες παιχνιδιών με την χρήση κάποιων εμπορικών headsets βιώνουν πολύ ρεαλιστικές εμπειρίες μέσω ενός avatar, είτε χρησιμοποιώντας ένα headset και κάποια τηλεχειριστήρια για πλοήγηση, αλλά κι άλλων εξαρτημάτων όπως συσκευές που προσομοιώνουν σε όπλα (Εικόνα 2-6).



Εικόνα 2-6: VR video game [\[12\]](#)

Τηλεπαρουσία

Το Google Street View ήταν το πρώτο βήμα της αίσθησης πως βρισκόμαστε κάπου αλλού, το οποίο επετεύχθη μέσω λήψης πανοραμικών εικόνων από αμέτρητες τοποθεσίες σε όλο τον κόσμο. Με την χρήση VR τεχνολογιών είναι δυνατή η δημιουργία αίσθησης στον χρήστη πως στέκεται μέσα σε αυτές τις τοποθεσίες, ενώ ανάλογες τεχνολογίες έχουν εφαρμοστεί για την τηλεπαρουσία σε συναυλίες (Εικόνα 2-7), δίνοντας την αίσθηση σε κάποιον τηλεθεατή πως βρίσκεται

στην σκηνή με τον καλλιτέχνη που παίζει ζωντανά.



Εικόνα 2-7: VR συναυλία του Paul McCartney [13]

Εικονικές κοινωνίες

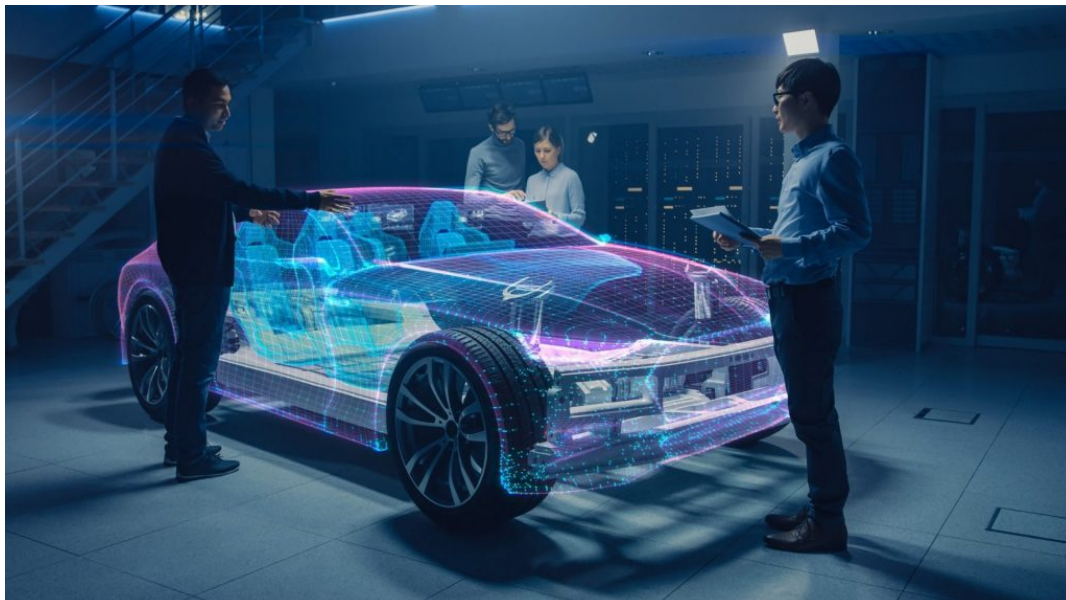
Σε συνέχεια της τηλεπαρουσίας, υπάρχει η δυνατότητα δημιουργίας εικονικών κοινωνιών οι οποίες μας θυμίζουν τις πραγματικές κοινωνίες, με τη διαφορά πως η αναπαράσταση γίνεται με τη χρήση avatar, τα οποία συνδέονται με τους πραγματικούς ανθρώπους. Η συμμετοχή σε εικονικές κοινωνίες μπορεί να γίνει ένα πανίσχυρο εργαλείο ώστε να προκαλέσεις την ενσυναίσθηση των ανθρώπων, δημιουργώντας εικονικές κοινωνίες στις οποίες κάποιοι άνθρωποι μπορούν να βιώσουν τις δυσκολίες κάποιων άλλων ανθρώπων. Το *Clouds Over Sidra* [14] είναι μια VR ταινία μικρού μήκους που ακολουθά τη ζωή της Sidra —μιας πρόσφυγα από τη Συρία— σε μια κατασκήνωση 84.000 προσφύγων στη Συρία (Εικόνα 2-8), με σκοπό να μεγαλώσει την ενσυναίσθηση των ανθρώπων ως προς το προσφυγικό ζήτημα.



Εικόνα 2-8: Clouds Over Sidra [\[15\]](#)

Εικονική δημιουργία πρωτοτύπων

Η δημιουργία και η εκτύπωση τρισδιάστατων πρωτοτύπων είναι μια διαδικασία που μπορεί να διευκολυνθεί με τη χρήση του VR, όπως η κατασκευή και ανάπτυξη ενός αυτοκινήτου (Εικόνα 2-9). Υπάρχει η δυνατότητα ανάπτυξης ενός εικονικού αυτοκινήτου, και πάνω σε αυτό να δοκιμαστούν τα πρωτότυπα.



Εικόνα 2-9: VR Prototyping [\[16\]](#)

Υγεία

Η Υγεία είναι ένας κρίσιμος τομέας εφαρμογής VR τεχνολογιών, όπου γιατροί μπορούν να εκπαιδευτούν στον εικονικό κόσμο, όπως για παράδειγμα ένα χειρουργείο (Εικόνα 2-10).



Εικόνα 2-10: VR προσομοίωση χειρουργείου [17]

2.4 Επαυξημένη Πραγματικότητα (Augmented Reality — AR)

Επαυξημένη πραγματικότητα είναι μια διαδραστική εμπειρία σε ένα περιβάλλον πραγματικού κόσμου, όπου τα αντικείμενα του πραγματικού κόσμου, ενισχύονται με ψηφιακά αντικείμενα [18]. Ο στόχος του AR, ήταν η ελάττωση του κόστους και η βελτίωση της αποδοτικότητας σε πολλές από τις διαδικασίες που σχετιζόταν ο άνθρωπος, κατά την κατασκευή αεροσκαφών. Ο όρος εισήχθη, για πρώτη φορά, από δύο ερευνητές της Boeing, τον Tom Caudell και τον David Mizell, το 1992 [18]. Είναι από τις πλέον αναπτυσσόμενες τεχνολογίες της εποχής μας, και χρησιμοποιείται, κατά κύριο λόγο, σε φορητές συσκευές (smartphones, tablets), καθώς αυτές οι συσκευές έχουν χαρακτηριστικά που χρειάζονται οι AR εφαρμογές, όπως τα γυροσκόπια, κάμερες, επιταχυνσιόμετρο, οθόνες υψηλής ανάλυσης, GPS, κ.α.

Η υψηλή ζήτηση σε smartphones, ώθησε την μαζική παραγωγή αυτών των μερών, το οποίο οδήγησε σε μεγάλες καινοτομίες αλλά και πτώση των τιμών. Η πραγματική αξία του AR είναι ο τρόπος με τον οποίο στοιχεία του πραγματικού

κόσμου συνδυάζονται στην αντίληψη ενός ατόμου για τον πραγματικό κόσμο, όχι ως μια απλή αναπαράσταση δεδομένων, αλλά μέσω της ενσωμάτωσης αισθήσεων εμπύθισης, οι οποίες —μέσω της διαδικασίας αυτής— γίνονται αντιληπτές ως φυσικά μέρη του πραγματικού κόσμου (Εικόνα 2-11).



Εικόνα 2-11: Εφαρμογή AR σε αποθήκη [19]

Η τεχνολογία αυτή, έχει εισχωρήσει σε διάφορους κλάδους του σύγχρονου τρόπου ζωής, όπως εκπαίδευση, επικοινωνίες, ιατρική, διασκέδαση, βιομηχανίες, logistics κ.α. Με την βοήθεια του AR, η πληροφορία από τον περιβάλλοντα πραγματικό κόσμο του χρήστη, γίνεται διαδραστική και ψηφιακά χειραγωγίσιμη. Η βασική διαφορά του με την εικονική πραγματικότητα (Virtual Reality — VR) είναι πως στο VR, η αντίληψη του χρήστη είναι βασισμένη στην εικονική πληροφορία, ενώ στην επαυξημένη πραγματικότητα (AR), η εικονική πληροφορία προβάλλεται στον πραγματικό κόσμο.

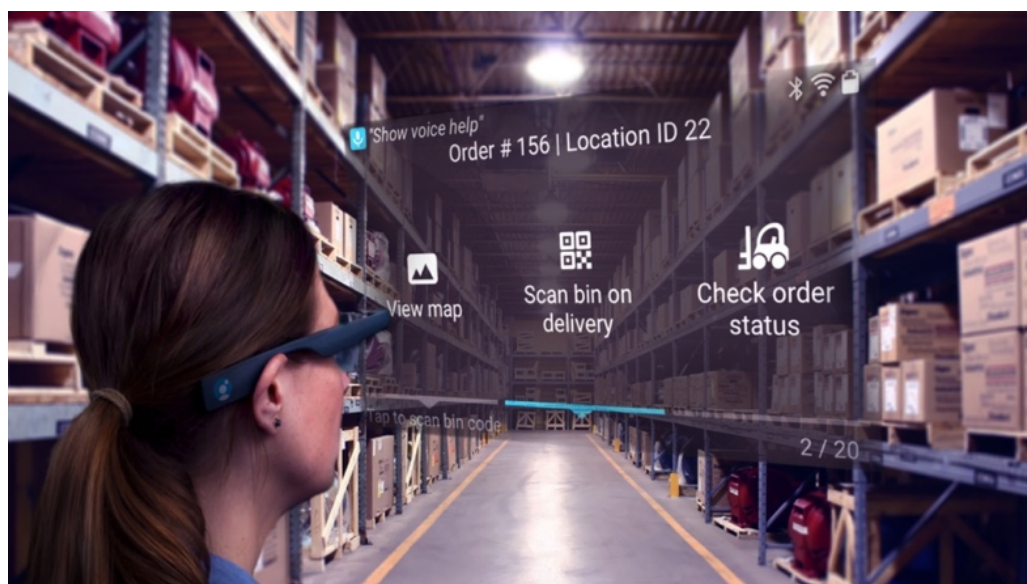
Οι συσκευές που απαιτούνται για την επαυξημένη πραγματικότητα είναι επεξεργαστές, οθόνες, αισθητήρες, και συσκευές εισόδου. Για οθόνη μπορούν να χρησιμοποιηθούν διάφορες τεχνολογίες, όπως απλές συμβατικές οθόνες (κινητά, tablets), συσκευές που εφαρμόζουν στο κεφάλι (Head-mounted Display — HMD) αλλά και ειδικού τύπου γυαλιά, τα οποία δίνουν τη δυνατότητα στον χρήστη να βλέπει το επαυξημένο περιβάλλον χωρίς να κρατάει επιπλέον συσκευές, ενώ είναι σε διαδικασία ανάπτυξης φακοί επαφής επαυξημένης πραγματικότητας. Δεδομένης της αλματώδους ανάπτυξης της τεχνολογίας, νέες τεχνολογίες, αλλά και συσκευές

εμφανίζονται καθημερινά. Στο τέλος του 2020, υπολογίζεται πως υπήρχαν 598 εκατομμύρια συσκευές AR, οι οποίες αναμένεται να φτάσουν στις 1.73 δισεκατομμύρια έως το 2024 [20].

2.4.1.1 Εφαρμογή των τεχνολογιών επαυξημένης πραγματικότητας — AR

Logistics – Αποθήκες

Τα logistics και οι αποθήκες είναι από τους κλάδους που επωφελούνται πρώτοι σε αυτή την ψηφιακή μεταμόρφωση του τρόπου εργασίας. Πλέον οι εργαζόμενοι μπορούν να εργαστούν πιο αποδοτικά, καθώς, για παράδειγμα, ένας εργαζόμενος δεν χρειάζεται να έχει δεσμευμένα τα χέρια του με μια συσκευή για να σκανάρει τα προϊόντα. Φορώντας τα γυαλιά AR, θα αναγνωρίζεται το barcode του προϊόντος και οι πληροφορίες του θα εμφανίζονται μπροστά του (Εικόνα 2-12).



Εικόνα 2-12: Αναζήτηση παραγγελίας με γυαλιά AR [21]

Βιομηχανία

Υπάρχουν διάφορες χρήσεις του AR στον βιομηχανικό κλάδο, όπως η εκπαίδευση νέων υπαλλήλων, ή η καθοδήγηση έμπειρων υπαλλήλων σε τομείς που χρειάζεται βοήθεια (Εικόνα 2-13). Μια άλλη περίπτωση χρήσης είναι η αναγνώριση αντικειμένων· μια πολύτιμη βοήθεια για έναν εργαζόμενο ο οποίος μπορεί να χρειαστεί να αναζητήσει ένα αντικείμενο ανάμεσα σε εκατοντάδες, ή και χιλιάδες. Μέσω του AR, μπορεί το αντικείμενο να αναγνωριστεί και να επισημανθεί. Στην εποχή μας, εν καιρώ πανδημίας, όλο και περισσότεροι εργαζόμενοι αδυνατούν να δώσουν φυσική παρουσία για να βοηθήσουν σε ένα πρόβλημα που υπάρχει. Πλέον

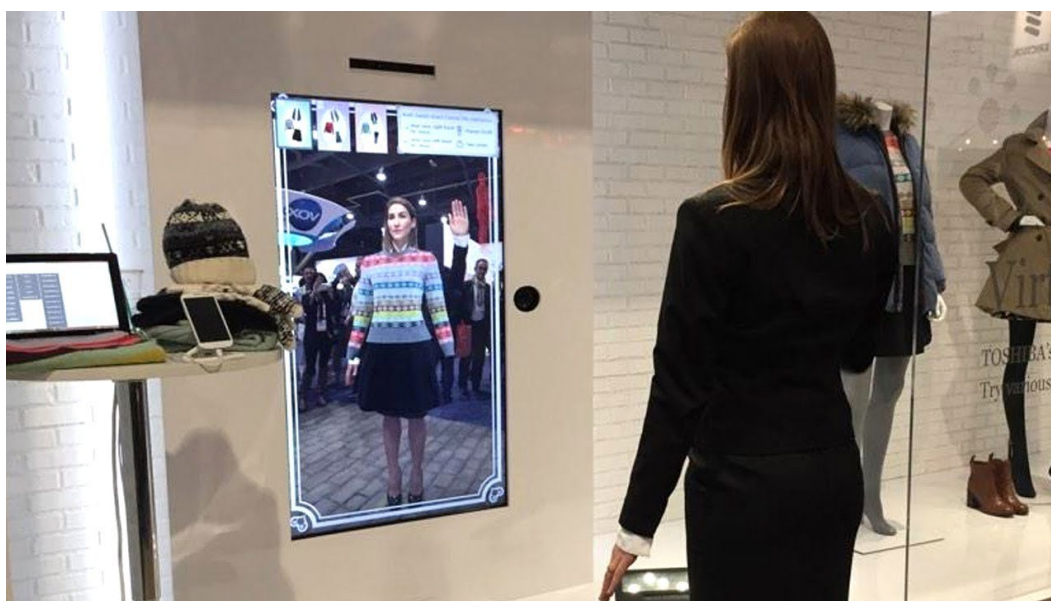
με το AR, η συνεργασία και η καθοδήγηση των συνεργατών είναι πιο εύκολη.



Εικόνα 2-13: Διαδικασία συντήρησης εξαρτήματος με AR [22]

Λιανική-Ψώνια

Κάποιες εταιρείες, έχουν αναπτύξει εφαρμογές εικονικών δοκιμαστηρίων (virtual fitting) όπου οι χρήστες μπορούν να δοκιμάσουν τα προϊόντα χωρίς να είναι αναγκαία η φυσική παρουσία, κάτι που εν μέσω πανδημίας φάνηκε πολύ σημαντικό. Παρόμοιου τύπου εφαρμογές είναι οι εφαρμογές όπου ο χρήστης δοκιμάζει αν τα έπιπλα ταιριάζουν στον χώρο του.



Εικόνα 2-12: Δοκιμαστήριο με AR [23]

Πλοήγηση

Ίσως η πιο συνήθης κατηγορία εφαρμογών AR στις μέρες μας, είναι οι εφαρμογές πλοήγησης. Εφαρμογές που βοηθούν τους χρήστες να βρουν τη σωστή διαδρομή σε αεροδρόμια, μετρό (Εικόνα 2-13), εμπορικά κέντρα, και άλλες περιοχές.



Εικόνα 2-13: AR εφαρμογή πλοήγησης για μετρό [24]

Μικτή Πραγματικότητα (MR — Mixed Reality)

Η μικτή πραγματικότητα —αναφέρεται και ως Υβριδική Πραγματικότητα (Hybrid Reality)— συνδυάζει στοιχεία από τις δύο προαναφερθείσες τεχνολογίες (VR, AR), και αφορά το ενδιάμεσο στάδιο μεταξύ της εικονικής και επαυξημένης πραγματικότητας [21] (Εικόνα). Υπάρχουν πολλές περιπτώσεις χρήσης, η πλειονότητα των οποίων, επικαλύπτεται από τις προαναφερθείσες περιπτώσεις των AR και VR τεχνολογιών. Αυτό που διαφοροποιείται είναι ο τρόπος εφαρμογής και

συνδυασμός αυτών των τεχνολογιών, το οποίο επιδρά στο τελικό αποτέλεσμα.



Εικόνα 2-14: Προβολή μακέτας με MR τεχνολογίες

3 Serious Games

3.1 Εισαγωγή στα serious games

Τα serious games ή αλλιώς, τα εκπαιδευτικά παιχνίδια, θέτουν από την αρχή τους πως πρωταρχικός τους στόχος είναι η μάθηση και η πρακτική εξάσκηση κάποιων δεξιοτήτων, παρά, αμιγώς, η διασκέδαση. Η συνεχής εξέλιξη τους και τα αδιαμφισβήτητα πλεονεκτήματα που προσφέρουν σε μία εκπαιδευτική διαδικασία, τα έχουν καταστήσει ιδιαίτερα δημοφιλή σε διάφορους τομείς όπως η εκπαίδευση, οι επιστήμες υγείας, η χημεία, τα μαθηματικά, η μηχανική, οι κλάδοι άμυνας και ασφάλειας.

Οι σκοποί για τους οποίους μπορούν να χρησιμοποιηθούν τα serious games είναι πολλοί και διαφορετικοί: από το να διδάξεις μία επιστήμη όπως εκείνη της χημείας, μέχρι να χρησιμοποιηθούν από κάποιον που ενδιαφέρεται να μάθει μία ξένη γλώσσα και από το να εκπαιδεύσεις μία ομάδα πωλητών, έως να εκπαιδεύσεις σώματα ασφαλείας και πυροσβεστικά σώματα.

Η χρήση τους στην εκπαιδευτική διαδικασία φαίνεται να έχει πρακτική εφαρμογή, τουλάχιστον, από τον 20ο αιώνα. Προάγοντας τη συνεργασία και βοηθώντας, καταλυτικά, στην οργάνωση της σκέψης, η βιωματική μάθηση μέσω των serious games, με τη συνδρομή, πάντοτε, των νέων τεχνολογιών και

εξαρτημάτων πληροφορικής, φέρνει έναν νέο αέρα στην εκπαιδευτική διαδικασία.

Η λογική της game-based εκπαιδευτικής διαδικασίας (χρήση των video games στην διαδικασία της εκπαίδευσης), βασίζεται στο γεγονός πως η γνώση που θέλεις να μεταφέρεις και οι τεχνικές γνώσεις που θέλεις να διδάξεις, δεν έχουν ως διδακτικό μέσο κάποιο βιβλίο ή την πρόσωπο με πρόσωπο —μέσα σε μία αίθουσα, πιο κλασσικού τύπου— εκπαιδευτική διαδικασία, αλλά ένα video game. Οι θιασώτες αυτής της εκπαιδευτικής μεθόδου, υποστηρίζουν πως η εκπαιδευτική διαδικασία μέσω των video games μπορεί να αποδειχθεί αποτελεσματική και διασκεδαστική ταυτόχρονα, προσφέροντας πολλαπλά πρακτικά πλεονεκτήματα όπως το μειωμένο οικονομικό κόστος των εκπαιδευτικών προγραμμάτων, καθώς και ψυχολογικά πλεονεκτήματα για τους μαθητευόμενους, όπως η διευκόλυνση της άμεσης πρακτικής διαδικασίας που αμέσως προσφέρει και μία νέα σειρά κινήτρων.

Τα πεδία στα οποία βρίσκουν εφαρμογή τα serious games είναι πολλά και διαφορετικά. Ο ιατρικός κλάδος, ενθουσιωδώς, ανταποκρίθηκε σε αυτήν την καινοτόμο μέθοδο διδασκαλίας, με μία συνεχώς αυξανόμενη χρήση των serious games στα ιατρικά εκπαιδευτικά προγράμματα. Ιατρικό και νοσηλευτικό προσωπικό έχουν πλέον τη δυνατότητα να δοκιμάσουν τις θεωρητικές τους γνώσεις και να κάνουν πρακτική μέσω των «παιχνιδιών» αυτών, αποκτώντας τη γνώση και την εμπειρία διαχείρισης καταστάσεων βασισμένες σε πραγματικά γεγονότα. Στον τομέα της άθλησης και άσκησης συναντούμε παιχνίδια που ξεπερνώντας τυπικά ή μη εμπόδια, όπως η ηλικία ή φυσικές δυσλειτουργίες, προσφέρουν στον χρήστη τη δυνατότητα κίνησης ή την αίσθηση αυτής παρότι βρίσκεται σε εσωτερικούς χώρους.

Οι τομείς άμυνας και ασφάλειας, κάνοντας χρήση των serious games, έχουν τη δυνατότητα να εκπαιδεύσουν το προσωπικό τους στην αντιμετώπιση πιθανών κινδύνων. Μέσω της ανάπτυξης εικονικών σεναρίων και με την αποφυγή λελογισμένων κινδύνων έχουν τη δυνατότητα να εκπαιδεύσουν το προσωπικό τους σε καταστάσεις αντιμετώπισης φυσικών καταστροφών, τρομοκρατικών ενεργειών, αποτροπή οποιασδήποτε κατάστασης κινδύνου, ανταπόκριση και παροχή άμεσης βοήθειας, κ.α..

Άλλοι τομείς όπως εκείνοι της πολιτικής, του πολιτισμού, των τεχνών ή της διαφήμισης, προσπαθούν να επικοινωνήσουν τους σκοπούς των έργων τους κάνοντας χρήση των serious games. Κλάδοι όπως περιβαλλοντικές οργανώσεις και φορείς ή φορείς που σχετίζονται με τον τουρισμό ή τον αστικό σχεδιασμό, έχει παρατηρηθεί πως κάνουν χρήση των παιχνιδιών ώστε να επικοινωνήσουν νέες ιδέες,

στρατηγικές και έναν πιο μοντέρνο τρόπο σκέψης και αντιμετώπισης του σύγχρονου κόσμου μέσω των βιωματικών εκπαιδευτικών εμπειριών που έχουν τη δυνατότητα να προσφέρουν αυτά τα παιχνίδια.

Καθοριστικό ρόλο φαίνεται να επιτελούν και στον αμιγώς εκπαιδευτικό κλάδο, τόσο των παιδιών όσο και των ενηλίκων. Αναφορικά με την εκπαίδευση ενηλίκων, μέσω εκπαιδευτικών προγραμμάτων προσομοίωσης, τους δίνεται η δυνατότητα να αποκτήσουν εμπειρία πάνω στο αντικείμενο του ενδιαφέροντος τους, κάνοντας χρήση της διαδικασίας συνεχόμενων δοκιμών έως ότου έρθουν στο αποτέλεσμα που επιθυμούν, εφαρμόζοντας τις θεωρητικές τους γνώσεις στο εικονικό πεδίο. Όσον αφορά τη βιωματική μάθηση μέσω του παιχνιδιού στα παιδιά (Εικόνα 3-1), διαμέσου μεθοδικών και σταδιακών βημάτων, παρέχουν τη δυνατότητα μέσω μικρών «αποστολών» κάθε φορά στο παιχνίδι να ανακαλύπτουν τη γνώση και να λύνουν τους γρίφους της. Η σωστή «ποσότητα» θεωρητικής γνώσης στη σωστή χρονική στιγμή, μέσα σε ένα σύγχρονο τεχνολογικά περιβάλλον προσδίδει ένα ακόμα πλεονέκτημα στη διαδικασία της μάθησης για τα μικρά παιδιά.



Εικόνα 3-1: *Serious Games* στην εκπαίδευση[26]

Αδιαμφισβήτητο, λοιπόν, είναι το γεγονός ότι τα ψηφιακά παιχνίδια διαθέτουν μία, συνεχώς, αυξανόμενη δημοτικότητα και αυτό προκύπτει από μία σειρά χαρακτηριστικών που διαθέτουν. Παρά το γεγονός πως υπάρχει μία τεράστια ποικιλία παιχνιδιών εκπαιδευτικού χαρακτήρα, τα περισσότερα διαθέτουν κάποια κοινά γνωρίσματα όπως η πλοκή —μια κύρια ιστορία γύρω από την οποία

διακλαδώνεται η δομή του παιχνιδιού. Όσο πιο προσεγμένο και δομημένο το περιβάλλον του παιχνιδιού και οι χαρακτήρες των εικονικών παικτών, τόσο πιο ευχάριστο, διασκεδαστικό και κινητροδοτικό είναι για τον φυσικό χρήστη-παίκτη. Οι παίκτες αποκτούν τη δυνατότητα μέσα σε ένα δομημένο περιβάλλον να αλληλοεπιδρούν και να ασχολούνται ενεργά. Τα παιχνίδια αυτά διαθέτουν τη δυνατότητα να προσαρμόζονται ώστε ο παίκτης να αποκτήσει τη μέγιστη δυνατή αίσθηση ικανοποίησης, δημιουργώντας συνθήκες νίκης με στόχο την τόνωση της αυτοπεποίθησης των παικτών και συνθήκες έντασης με στόχο την αύξηση της αδρεναλίνης. Παρέχουν κίνητρα στον παίκτη, ώστε να ενισχύσουν τη δημιουργικότητα του μέσω προβλημάτων προς επίλυση, θέτουν ως στόχο την πρόκληση συναισθημάτων των παικτών μέσω του σεναρίου και του εικονικού περιβάλλοντος μέσα στο οποίο αυτό αναπτύσσεται, καθώς επίσης ενισχύουν και ενθαρρύνουν τη δημιουργία ομάδων και την ανάπτυξη της επικοινωνίας ανάμεσα στους παίκτες.

Παρέχοντας γνώση και ενημέρωση, εκπληρώνουν το στόχο τους που είναι αυτή καθαυτή η απόκτηση γνώσης και η πρακτικής εξάσκηση του παίκτη στο πεδίο ενδιαφέροντος του, μέσω ασφαλών πρακτικών.

3.1.1 Serious Games with Virtual Reality

Η Εικονική Πραγματικότητα ενσωματώνεται μέσω εφαρμογών και νέων τεχνολογιών στην εκπαιδευτική διαδικασία. Μέσω της εμβύθισης (immersion) ο χρήστης έχει τη δυνατότητα να αποκτήσει εμπειρίες σε πρώτο πρόσωπο, καθώς η Εικονική Πραγματικότητα προσομοιώνει ένα περιβάλλον τριών διαστάσεων, παρέχοντας κατ' αυτόν τον τρόπο υψηλό βαθμό αλληλεπίδρασης και ενίσχυσης της αμεσότητας επαφής του χρήστη με αυτό.

Κάποια χρόνια πριν, η high immersive (υψηλού βαθμού εμβύθισης) Εικονική Πραγματικότητα αναδύθηκε παράλληλα με την ανάπτυξη των τρισδιάστατων ευρείας κλίμακας συσκευών όπως το Head Mounted Display (HMD) και το Cave Automatic Virtual Environment (CAVE). Η ανάπτυξη αυτών των εφαρμογών και μέσων, συνοδεύτηκε από την ανάπτυξη των πρώτων περιβαλλόντων Εικονικής Πραγματικότητας με σκοπό την εφαρμογή σε εκπαιδευτικά πεδία διάφορων επιστημών όπως τα μαθηματικά, η οικονομία και η διοίκηση επιχειρήσεων, η πληροφορική, η εκμάθηση ξένων γλωσσών, κ.α. Η είσοδος τους στην εκπαιδευτική διαδικασία στέφθηκε με επιτυχία καθώς οι κριτικές

από τις πρώτες κιόλας χρήσεις, των εκπαιδευτικών περιβαλλόντων Εικονικής Πραγματικότητας, υπερτόνισαν δύο βασικές αρχές της· το ότι οι νέες τεχνολογίες προκαλούν το ενδιαφέρον των νέων, και παράλληλα αυξάνουν το βαθμό εμπλοκής τους στη μαθησιακή διαδικασία. Το γεγονός ότι η Εικονική Πραγματικότητα μπορεί μέσω της οπτικοποίησης και του βιωματικού χαρακτήρα της εμπειρίας να κάνει ακόμη πιο κατανοητές δυσνόητες έννοιες για τους μαθητές, αποτελεί ένα ακόμη κέρδος για την εκπαιδευτική διαδικασία.

Παρά το γεγονός ωστόσο πως η Εικονική Πραγματικότητα διαθέτει ένα ισχυρό χαρακτηριστικό, αυτό της εμπύθισης, υπάρχει κι ένα δεύτερο στοιχείο εξίσου απαραίτητο, η διαδραστικότητα του χρήστη με το εικονικό περιβάλλον. Τα Serious Games βασίζονται στο σχεδιασμό τους και θέτουν ως στόχο ο χρήστης όχι μόνο να ψυχαγωγηθεί αλλά και να εκπαιδευτεί επί ενός αντικειμένου, κάνοντας χρήση τους. Θέτοντας τον χρήστη στο κέντρο αυτής της εκπαιδευτικής διαδικασίας, ο ίδιος αισθάνεται πως έχει τον έλεγχο αυτής της διαδραστικής διδασκαλίας, αυξάνοντας την ενεργό συμμετοχή του και βελτιώνοντας την κριτική σκέψη και ικανότητα. Από τις πιο σημαντικές συνεισφορές στην εκπαιδευτική διαδικασία είναι εκείνες που σχετίζονται με την αναπαράσταση δυνητικών καταστάσεων και περιβαλλόντων που σε διαφορετική περίπτωση δε θα είχαν τη δυνατότητα οι χρήστες να αντιμετωπίσουν και να εκπαιδευτούν αποτελεσματικά, σε πραγματικές συνθήκες. Αυτό περιλαμβάνει μία ποικιλία από επικίνδυνες —ακόμη και αδύνατον να αναπαραχθούν στον πραγματικό κόσμο— καταστάσεις.

3.1.2 *Fire Safety/ Education with Serious Games/VR*

Η Εικονική Πραγματικότητα, όπως έχει ήδη αναφερθεί, έχει κερδίσει τη θέση της στην εργαλειοθήκη των εκπαιδευτικών προγραμμάτων, μεταξύ των οποίων βρίσκεται και η εκπαίδευση στην πυρασφάλεια. Παρά τα μέτρα αποτροπής της πυρκαγιάς που λαμβάνονται είτε σε δασικές περιοχές είτε σε αστικές, το φαινόμενο της φωτιάς και οι καταστρεπτικές του συνέπειες είναι εδώ, συχνά, και μας υπενθυμίζει πως στη μεθοδολογία αντιμετώπισης της, τα «όπλα» δεν είναι ποτέ αρκετά. Οι συνέπειες της πυρκαγιάς είναι πολύ μεγάλες —είτε αφορά δασικές περιοχές, είτε βρίσκεται εντός του αστικού ιστού— και ποικίλουν αναφορικά με το μέγεθος της καταστροφής που αφήνει πίσω της σε όλες τις διαστάσεις της· περιβαλλοντικές, οικονομικές ή το απεχθέστερο όλων, απώλειες σε ανθρώπινες ζωές. Τα πυροσβεστικά σώματα ή οι υπεύθυνοι πυρασφάλειας των κτιρίων θα

κληθούν να αντιμετωπίσουν αυτό το ισχυρό και επικίνδυνο φαινόμενο και για αυτόν ακριβώς το λόγο πρέπει να είναι όσο το δυνατόν καλύτερα εκπαιδευμένοι, φέροντας το καλύτερο δυνατό αποτέλεσμα και μειώνοντας τα μη απαραίτητα ρίσκα.

Η εκπαίδευση πυρασφάλειας, κάνοντας χρήση εργαλείων Εικονικής Πραγματικότητας και Serious Games, απαρτίζεται από το θεωρητικό κομμάτι της γνώσης και στη συνέχεια από το πρακτικό στο οποίο οι εκπαιδευόμενοι θα μάθουν, σε ελεγχόμενο περιβάλλον, να αντιμετωπίζουν μικρής κλίμακας πυρκαγιές, να φορούν και να κάνουν χρήση με ορθό τρόπο του πυροσβεστικού εξοπλισμού σε περιβάλλοντα πολύ υψηλών θερμοκρασιών και υπό τη συναισθηματική και ψυχολογική φόρτιση έντονου στρες.

Η Εικονική Πραγματικότητα μέσω του περιβάλλοντος προσομοίωσης, αποτελεί τεχνολογία «κλειδί» που, βάσει της εφαρμογής πολλών και διαφορετικών σεναρίων πυρκαγιάς, δίνει τη δυνατότητα στους εκπαιδευόμενους να αποκτούν όσο το δυνατόν περισσότερες εμπειρίες αντιμετώπισης σε ένα απολύτως ασφαλές περιβάλλον, εξαλείφοντας έτσι μη αναγκαία ρίσκα από τη διαδικασία εκπαίδευσης. Οι αλγόριθμοι και τα μοντέλα που χρησιμοποιούνται, έχουν σχεδιαστεί έτσι ώστε να παράγουν —κατά προσέγγιση και όσο το δυνατόν με μεγαλύτερη ταχύτητα εναλλαγής— αποτελέσματα που αντιστοιχούν σε προσομοιωμένες περιπτώσεις πυρκαγιάς και που μπορούν να χρησιμοποιηθούν στην εκπαίδευση πυροσβεστών και υπευθύνων πυρασφάλειας.

Η χρήση εκπαιδευτικών χώρων προσομοίωσης εκπαίδευσης στην πυρασφάλεια, παρέχει πολλαπλά οφέλη και σε μια πιο συγκεκριμένη περίπτωση, όπως η εκπαίδευση στην αντιμετώπιση του κινδύνου πυρκαγιάς, εκτός των πυροσβεστών και των υπευθύνων πυρασφάλειας και τα μικρά παιδιά. Τα παιδιά, δεν θα κληθούν να σβήσουν τη φωτιά σε μια πραγματική κατάσταση, αλλά κρίνεται χρήσιμο να ξέρουν πως συμπεριφέρεται η φωτιά και πως μπορεί μια μικρή φλόγα, με μια λάθος κίνηση να γίνει μια μεγάλη φωτιά.

Οι μέχρι τώρα μελέτες δείχνουν πως οι συμπεριφορικές ικανότητες, αναφορικά με την αντιμετώπιση ενός ενδεχόμενου περιστατικού πυρκαγιάς, βελτιώνονται μέσω των προγραμμάτων που προσομοιώνουν συνθήκες τέτοιου τύπου μέσω εικονικών περιβαλλόντων. Οι ικανότητες που αποκτώνται μέσω της εκπαίδευσης σε περιβάλλον Εικονικής Πραγματικότητας μπορούν να βρουν πεδίο εφαρμογής σε πραγματικές συνθήκες αυξάνοντας σημαντικά τα ποσοστά μιας

επιτυχημένης αντιμετώπισης σε συνθήκες έκτακτης ανάγκης.

3.2 Το πρόβλημα της εκπαίδευσης στον τομέα της πυρασφάλειας

Η εκπαίδευση του επαγγελματικού προσωπικού, αλλά και απλών πολιτών, στο κομμάτι της πυρασφάλειας είναι ένα πρόβλημα, στο οποίο εδώ και χρόνια αναζητούνται οι βέλτιστες λύσεις και προτάσεις. Το πρόβλημα αυτό εντείνεται τα τελευταία χρόνια, καθώς λόγω της κλιματικής αλλαγής, το ρίσκο εκδήλωσης και εξάπλωσης της φωτιάς έχει αυξηθεί. Η παρούσα εργασία αφορά σενάριο εκδήλωσης σε κλειστό χώρο, ωστόσο έχει αναπτυχθεί με τέτοιο τρόπο ώστε να μπορεί να προσαρμοστεί σε διάφορα σενάρια.

Η χρήση προσομοιωτών εικονικής πραγματικότητας για την εκπαίδευση σε σενάρια πυρασφάλειας, έχει αρκετά πλεονεκτήματα:

Αποφυγή τραυματισμών

Το πιο σημαντικό πλεονέκτημα των προσομοιώσεων είναι η αποφυγή πιθανών τραυματισμών από μια εκπαίδευση πυρόσβεσης φωτιάς σε κανονικές συνθήκες.

Σενάρια

Η δημιουργία σεναρίων στον εικονικό κόσμο δεν έχει κανέναν περιορισμό, σε αντίθεση με τον πραγματικό κόσμο, όπου πρέπει να βρεθεί ο χώρος, τα υλικά, αλλά και να μεταφερθούν οι εκπαιδευόμενοι στο σημείο εκπαίδευσης.

Κόστος

Το κόστος για την δημιουργία ενός σεναρίου εκπαίδευσης, αλλά και τα αναλώσιμα υλικά που θα καούν, σε επαναλαμβανόμενες εκπαιδευτικές συνεδρίες μπορεί να εκτοξεύσουν το κόστος για έναν οργανισμό.

Απομακρυσμένη εκπαίδευση

Η εκπαίδευση μέσω προσομοίωσης, παρέχει τη δυνατότητα —εφόσον το επιτρέπουν οι τεχνολογίες που αναπτύχθηκε— να πραγματοποιηθεί απομακρυσμένα.

Μαθησιακά αποτελέσματα

Όπως αναλύθηκε και στην ενότητα [1.4](#), η επανάληψη των προσομοιώσεων, η ρεαλιστικότητά τους, αλλά και η αίσθηση του παιχνιδιού, δημιουργούν υψηλά μαθησιακά αποτελέσματα στους εκπαιδευόμενους.

3.3 Συναφείς εργασίες

Οι προσομοιώσεις εκπαίδευσης πυρκαγιάς ή οι προσομοιώσεις εξάπλωσης πυρκαγιάς, γενικά, χρησιμοποιούνται ευρέως τα τελευταία χρόνια και είναι ένα δημοφιλές πεδίο έρευνας. Οι Moreno και συν. [27], πρότειναν αλγόριθμους εξάπλωσης πυρκαγιάς για δασικά και αστικά περιβάλλοντα, που παρήγαγαν κατά προσέγγιση, αλλά καλά αποτελέσματα που θα μπορούσαν να χρησιμοποιήσουν οι πυροσβέστες και οι διαχειριστές. Πρότειναν αλγόριθμους χαμηλής πολυπλοκότητας για να λειτουργούν αποτελεσματικά, με χαμηλή υπολογιστική ισχύ. Επίσης, προσάρμοσαν την προσέγγιση του πλέγματος όπου η κυψέλη κάθε δικτύου περιέχει τις βασικές πληροφορίες, όπως τον τύπο, την κατάσταση, το καύσιμο υλικό και άλλα. Η αξιολόγηση των αλγορίθμων πραγματοποιήθηκε με τη χρήση της εφαρμογής FARSITE, την έξοδο του σεναρίου ανάλυσης Zhao [28], καθώς και με τη συμβουλή επαγγελματιών πυροσβεστών.

Οι Fathima και συν. [29] παρουσίασαν ένα εκπαιδευτικό περιβάλλον (εμβύθισης) πυρασφάλειας για πυροσβεστήρες, όπου η βασική ιδέα ήταν το P.A.S.S. (Pull, Aim, Squeeze, Sweep). Αυτό το VR serious game διδάσκει τους εκπαιδευόμενους πώς να εφαρμόζουν ορισμένες ασκήσεις πυρασφάλειας που είχαν διδαχθεί νωρίτερα, σε πραγματική κατάσταση. Ολόκληρη η εκπαιδευτική διαδικασία εφάρμοσε μοτίβα παιχνιδιού για να είναι πιο φιλικά για τους εκπαιδευόμενους. Η σκηνή ήταν ένα δωμάτιο γραφείου, όπου ξεσπά φωτιά και ο χρήστης έχει συγκεκριμένο χρόνο για να σώσει ένα άτομο, οδηγώντας τον έξω από το γραφείο.

Οι Dechamplain και συν. [30], παρουσίασαν το Blaze, ένα serious game που προορίζεται να εκπαιδεύσει τους χρήστες για τους κινδύνους πυρκαγιάς στο σπίτι. Η σκηνή ξεκινά με μια μικρή εστία μαγειρέματος και ο χρήστης πρέπει να επιλέξει το κατάλληλο εργαλείο ή συστατικό για να σβήσει τη φωτιά, και μπορεί να αποφασίσει ανάλογα με τις περιστάσεις. Για το σύστημα διάδοσης της φωτιάς, χρησιμοποίησαν την έννοια του μπλοκ όπου κάθε μπλοκ έχει μια τιμή υλικού και θερμοκρασίας. Η φωτιά εξαπλώνεται θερμαίνοντας παρακείμενα τετράγωνα, τα οποία πυροδοτούν την εξάπλωση της φωτιάς όταν περάσουν ένα όριο. Αυτό το όριο εξαρτάται από το υλικό κάθε μπλοκ.

Οι Schlager και συν. [31], πρότειναν μια λύση για την εκπαίδευση μη επαγγελματιών χρησιμοποιώντας το HTC VIVE ως πυροσβεστικό μέσο. Το εικονικό περιβάλλον έχει αναπτυχθεί με την Unity 3D και μια προσέγγιση

πλέγματος προσαρμοσμένη για το μοντέλο εξάπλωσης πυρκαγιάς. Αναπτύχθηκε ένα τρισδιάστατο πλέγμα voxel (μια μονάδα γραφικών πληροφοριών που ορίζει ένα σημείο στον τρισδιάστατο χώρο) όπου κάθε voxel περιέχει τις πληροφορίες του αντικειμένου: θερμοκρασία, κατάσταση και φυσικές ιδιότητες. Κάθε voxel είναι υπεύθυνο για την έναρξη ή τη διακοπή της εξάπλωσης της πυρκαγιάς, σύμφωνα με τα όρια αυτανάφλεξης που ορίστηκαν για κάθε αντικείμενο. Κάθε αντικείμενο είχε έξι θερμομπομπούς — σε κάθε πλευρά — που λάμβαναν ή εξέπεμπαν θερμότητα στη σκηνή.

4 Πλατφόρμες ανάπτυξης περιβάλλοντος εικονικής πραγματικότητας

Ενδεικτικά, οι πιο δημοφιλείς πλατφόρμες ανάπτυξης περιβάλλοντος VR:

- Unity
- Unreal
- Amazon Sumerian
- Google VR

Στην παρούσα εργασία επιλέχθηκε η Unity engine.

4.1 Unity

Η Unity είναι μια μηχανή ανάπτυξης παιχνιδιών, που υποστηρίζει διάφορες πλατφόρμες (cross-platform), η οποία κυκλοφόρησε τον Ιούνιο του 2005, στο παγκόσμιο συνέδριο της Apple. Παρουσιάστηκε ως μια αποκλειστική μηχανή παιχνιδιών (game engine) της Apple, και έπειτα από λίγους μήνες, επέκτεινε την συμβατότητά της ώστε να υποστηρίζει το λειτουργικό σύστημα των Windows, και το 2008, επεκτάθηκε στο iPhone. Το 2010 έγινε συμβατό με το λειτουργικό σύστημα κινητών συσκευών Android, και τα τελευταία χρόνια προστέθηκε συμβατότητα με WebGL —το νέο framework για τρισδιάστατη υποστήριξη στον περιηγητή (browser)— καθώς και υποστήριξη σε πλατφόρμες εικονικής πραγματικότητας (Virtual Reality — VR), όπως το Oculus Rift και το Vive. Το 2018, επέκτεινε την υποστήρισή της σε περισσότερες από 25 πλατφόρμες [32]. Η μηχανή αυτή, μπορεί

να χρησιμοποιηθεί για να αναπτυχθούν, μεταξύ άλλων, παιχνίδια ή περιβάλλοντα:

- Τρισδιάστατα (3D)
- Δισδιάστατα (2D)
- Εικονικής πραγματικότητας (VR)
- Επαυξημένης πραγματικότητας (AR)
- Προσομοιώσεις (Simulations)

Έπειτα, η λύση του Unity υιοθετήθηκε από πολλές εταιρείες, και εκτός της βιομηχανίας των παιχνιδιών, όπως ο κινηματογράφος, η αυτοκινητοβιομηχανία, η αρχιτεκτονική, η μηχανική, ο κατασκευαστικός κλάδος κ.α.

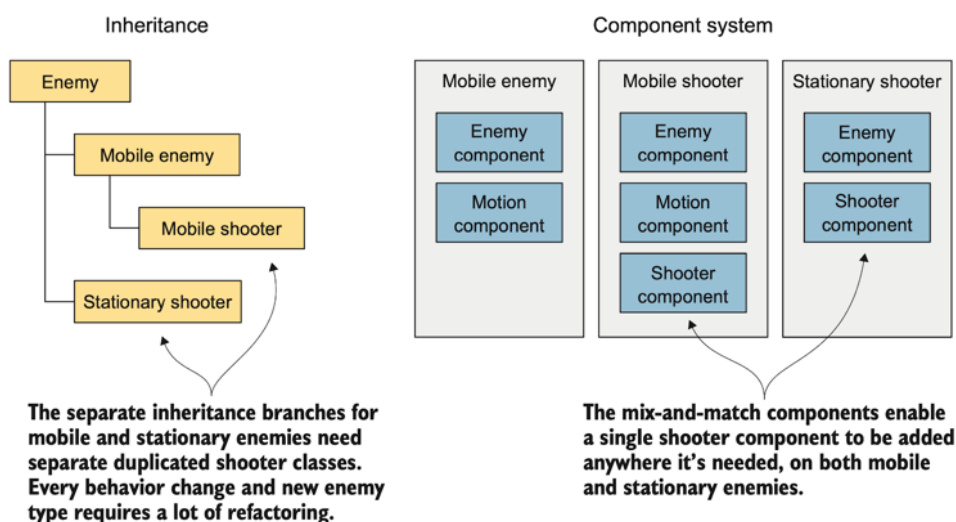
4.1.1 Πλεονεκτήματα της Unity engine

Οι μηχανές ανάπτυξης παιχνιδιών προσφέρουν μια πληθώρα χαρακτηριστικών, τα οποία είναι χρήσιμα κατά της διαδικασία της ανάπτυξης. Η Unity παρέχει προσομοίωση φυσικών φαινομένων (physics simulation), χάρτες, δυναμικές σκιές και πολλά άλλα. Πολλές μηχανές παιχνιδιών, έχουν αυτά τα χαρακτηριστικά, αλλά η Unity, έχει δύο κύρια πλεονεκτήματα σε σχέση με άλλες μοντέρνες ανταγωνιστικές μηχανές παιχνιδιών. Έχει ένα εξαιρετικά φιλικό περιβάλλον ανάπτυξης, που ενισχύει την παραγωγικότητα των προγραμματιστών, και ένα πολύ υψηλό βαθμό συμβατότητας με διάφορες πλατφόρμες. Η ροή εργασίας με τη χρήση της γραφικής διεπαφής χρήστη, έχει μοναδικό σχεδιασμό σε σχέση με παρόμοιες μηχανές του χώρου. Ενώ άλλα εργαλεία ανάπτυξης παιχνιδιών είναι, συνήθως, ένα μίγμα μερών που πρέπει να συνδυάσεις και να βάλεις σε τάξη, ή ίσως, μια βιβλιοθήκη προγραμματισμού, η οποία απαιτεί να στήσεις το δικό σου ενσωματωμένο περιβάλλον ανάπτυξης (Integrated Development Environment — IDE), η ροή ανάπτυξης στην Unity συνοδεύεται από ένα εκλεπτυσμένο περιβάλλον σύνταξης κώδικα (editor). Ο editor χρησιμοποιείται για να συνδέσεις κομμάτια κώδικα (scripts) με αντικείμενα (gameobjects) στην σκηνή του Unity, με σκοπό την αλληλεπίδραση των δύο μερών.

Αυτός ο editor ενισχύει το rapid iteration —την δημιουργία μικρών εκδόσεων της εφαρμογής επαναληπτικά σε μικρό χρονικό διάστημα— σύμφωνα με τις ανάγκες του προγραμματιστή· ακόμη και όταν η σκηνή «τρέχει». Τέλος, η Unity δίνει τη δυνατότητα να στον προγραμματιστή να γράψει τα δικά του scripts, ώστε να προσθέσει νέα μενού, ή χαρακτηριστικά στη γραφική διεπαφή χρήστη (GUI – Graphical User Interface). Εκτός της φιλικότητας του περιβάλλοντος ανάπτυξης ως

προς τον προγραμματιστή, η συμβατότητα της Unity με μεγάλο αριθμό πλατφορμών, είναι άλλο ένα μεγάλο πλεονέκτημα. Η Unity δεν είναι συμβατή με διάφορα λογισμικά μόνο κατά το στάδιο του deployment (δημοσίευση, ή «ανέβασμα» της εφαρμογής), αλλά και κατά τη διάρκεια της ανάπτυξης, καθώς είναι συμβατή με Windows, Mac OS, και Linux.

Επιπλέον των παραπάνω πλεονεκτημάτων, μια ακόμη μεγάλη δυνατότητα του Unity, είναι το σύστημα μερών (component system). Σε ένα σύστημα μερών, τα μέρη (components), ανταλλάσσουν πακέτα λειτουργικότητας, και τα αντικείμενα είναι χτισμένα, ως μια συλλογή από μέρη, παρά μια αυστηρή ιεραρχία κατηγοριών. Αυτό το σύστημα μερών, ενισχύει την αντικειμενοστραφή μέθοδο προγραμματισμού, όπου τα αντικείμενα παιχνιδιού (game objects) είναι αντικείμενα, κατασκευασμένα μέσω σύνθεσης, και όχι, μέσω κληρονομικότητας



Εικόνα 4-1: Κληρονομικότητα vs Σύστημα Μερών [33]

Σε ένα σύστημα μερών, τα αντικείμενα είναι σε μια επίπεδη ιεραρχία, και τα διάφορα αντικείμενα έχουν διαφορετικές συλλογές από μέρη (components), αντίθετα με μια ιεραρχική δομή, όπου διαφορετικά αντικείμενα είναι σε εντελώς διαφορετικά κλαδιά (branches) του δέντρου μερών (component tree). Αυτό ενισχύει την ταχύτητα ανάπτυξης (rapid prototyping), καθώς μπορείς, πολύ γρήγορα, να συνθέτεις μεταξύ τους πολλά μέρη χωρίς να εμπλέκεσαι με χρονοβόρες διαδικασίες όπως το refactoring.

Εκτός από πλεονεκτήματα, το Unity, έχει και κάποια μειονεκτήματα. Το σύστημα μερών της Unity, σε πολύπλοκες σκηνές, με πάρα πολλά αντικείμενα,

μπορεί να οδηγήσει στο χάσιμο της επαφής με το ποια αντικείμενα, σε μια σκηνή, έχουν συγκεκριμένα μέρη συνημμένα. Παρόλο που παρέχει μηχανή αναζήτησης όλων των μερών, κάποιες φορές μπορεί να καθυστερήσει την ανάπτυξη. Ένα ακόμη μειονέκτημα είναι η μη υποστήριξη σύνδεσης με εξωτερικές βιβλιοθήκες. Οι απαιτούμενες βιβλιοθήκες πρέπει να αποθηκεύονται χειροκίνητα, σε κάθε πρότζεκτ που χρησιμοποιούνται, εν αντιθέσει, με την αναφορά (referencing) σε μια κεντρική τοποθεσία. Η έλλειψη κεντρικής διαχείρισης βιβλιοθηκών, μπορεί να κάνει τον διαμοιρασμό λειτουργικότητας, μεταξύ διαφόρων πρότζεκτ, προβληματικό.

Ενδεικτικά, κάποια γνωστά παιχνίδια που αναπτύχθηκαν με τη Unity Engine:

Cuphead

Ένα από τα πιο επιτυχημένα παιχνίδια της τελευταίας δεκαετίας, το οποίο αναπτύχθηκε και δημοσιεύθηκε από την Studio MDHR. Δισδιάστατο παιχνίδι πλευρικής κύλισης, με μια αισθητική η οποία θυμίζει κινούμενα σχέδια της Walt Disney, σε συνδυασμό με την αισθητική της εποχής του βωβού κινηματογράφου (Εικόνα 4-2)



Εικόνα 4-2: Στιγμιότυπο από το παιχνίδι Cuphead [34]

Pillars of Eternity

Ένα, διάσημο παιχνίδι ρόλων (Εικόνα 4-3), που αναπτύχθηκε από την Obsidian Entertainment, και δημοσιεύθηκε από την Paradox Interactive.



Εικόνα 4-3: Στιγμιότυπο του παιχνιδιού Pillars Of Eternity [35]

Ori And The Blind Forest

Ένα παιχνίδι (Εικόνα 4-4) το οποίο αναπτύχθηκε από την Moon Studios, και δημοσιεύθηκε από την Microsoft Studios. Είναι ένα παιχνίδι, τύπου Metroidvania —δηλαδή, ο παίκτης, συχνά, χρειάζεται να ξαναεπισκεφτεί περιοχές, οι οποίες, προηγουμένως, εξαναγκάστηκε από το παιχνίδι να της προσπεράσει καθώς δεν είχε ξεκλειδώσει τα απαραίτητα χαρακτηριστικά ή ικανότητες για να μπορέσει να εισέλθει σε αυτές.

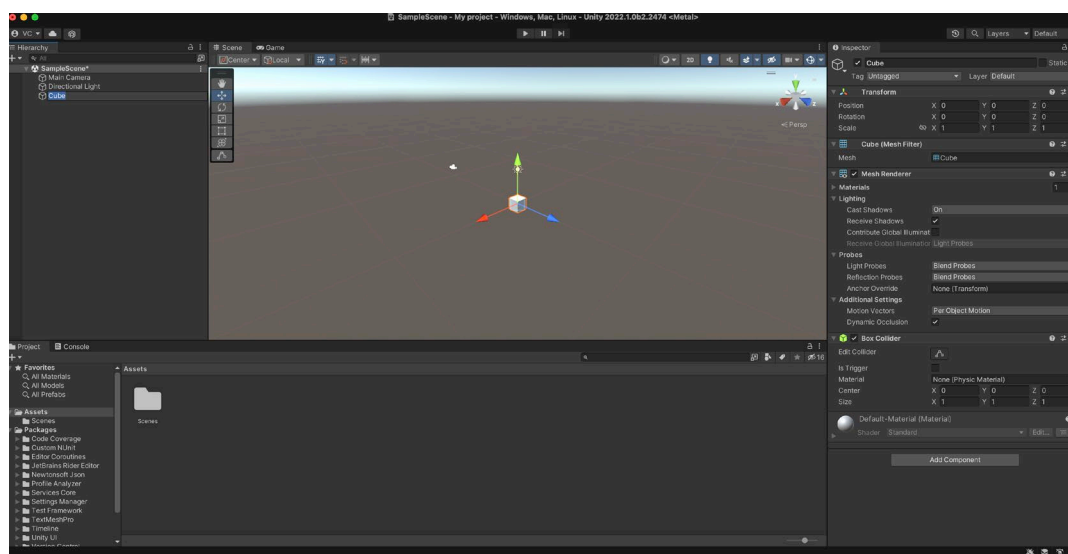


Εικόνα 4-4: Στιγμιότυπο του παιχνιδιού Ori and the Blind Forest [36]

4.2 Πως χρησιμοποιείται το Unity

Η γλώσσα ανάπτυξης της Unity, είναι η C#. Η C# είναι μια μοντέρνα, αντικειμενοστραφής και type-safe (ασφαλής ως προς τους τύπους) γλώσσα προγραμματισμού. Έχει τις ρίζες της στην C και μπορεί να είναι οικεία σε προγραμματιστές που έχουν χρησιμοποιήσει, τις C, C++, Java, JavaScript και TypeScript. [37]

Η διεπαφή χρήστη (User Interface/ UI) της Unity είναι οργανωμένη σε παράθυρα, όπου σε καθένα από αυτό ο χρήστης μπορεί να κάνει διαφορετικά πράγματα (Εικόνα 4-5).



Εικόνα 4-5: Το παραθυρικό περιβάλλον της Unity

Στην πρώτη εκκίνηση της Unity, η διάταξη της διεπαφής χρήστη έχει μια προεπιλεγμένη μορφή, η οποία ωστόσο, μπορεί να αλλάξει σύμφωνα με τις προτιμήσεις του χρήστη. Ένα από τα βασικά κομμάτια της διεπαφής χρήστη, είναι η γραμμή εργαλείων (Toolbar). Βρίσκεται πάντα στην κορυφή της οθόνης και είναι το μόνο κομμάτι που δεν μπορεί να αλλάξει θέση (εικόνα 4-6).

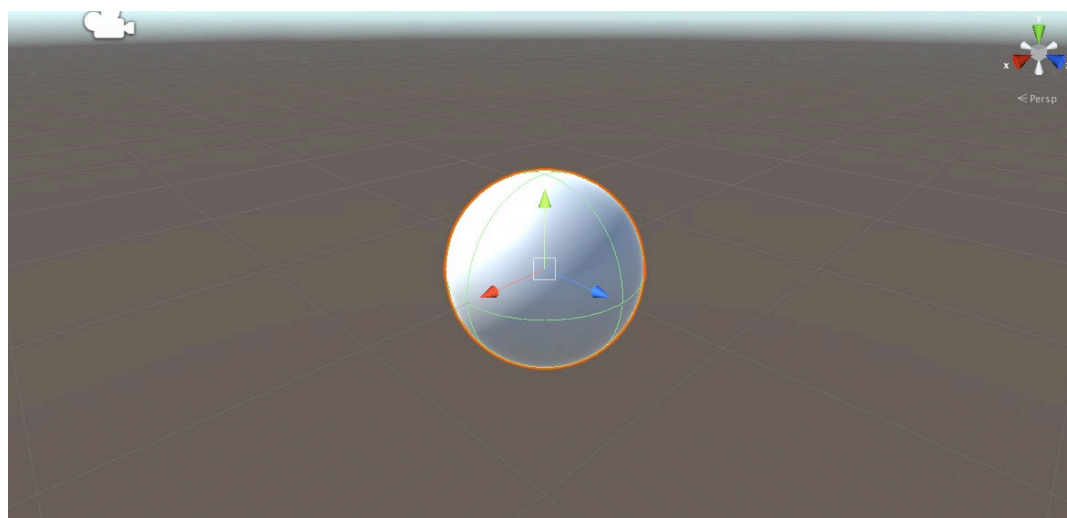
- **Εργαλείο χεριού (Hand tool):** Με αυτό το εργαλείο μπορείς να περιηγηθείς στη σκηνή, κάνοντας ένα κλικ και «τραβώντας» (click & drag).
- **Εργαλείο μετακίνησης (Move tool):** Μετακίνηση αντικειμένων. Επιλέγοντας κάποιο αντικείμενο, μπορείς να το μετακινήσεις οπουδήποτε στη σκηνή.
- **Εργαλείο περιστροφής (Rotate tool):** Περιστροφή αντικειμένων. Επιλέγοντας κάποιο αντικείμενο, μπορείς να το περιστρέψεις.

- **Εργαλείο κλιμάκωσης (Scale tool):** Κλιμάκωση αντικειμένων. Με αυτό το εργαλείο, μπορείς να αλλάξεις την κλίμακα ενός αντικειμένου.
- **Κουμπί εκκίνησης (Play button):** Με αυτό το κουμπί ξεκινά η σκηνή και σταματά.
- **Κουμπί παύσης (Pause button):** Με αυτό το κουμπί γίνεται παύση της σκηνής.

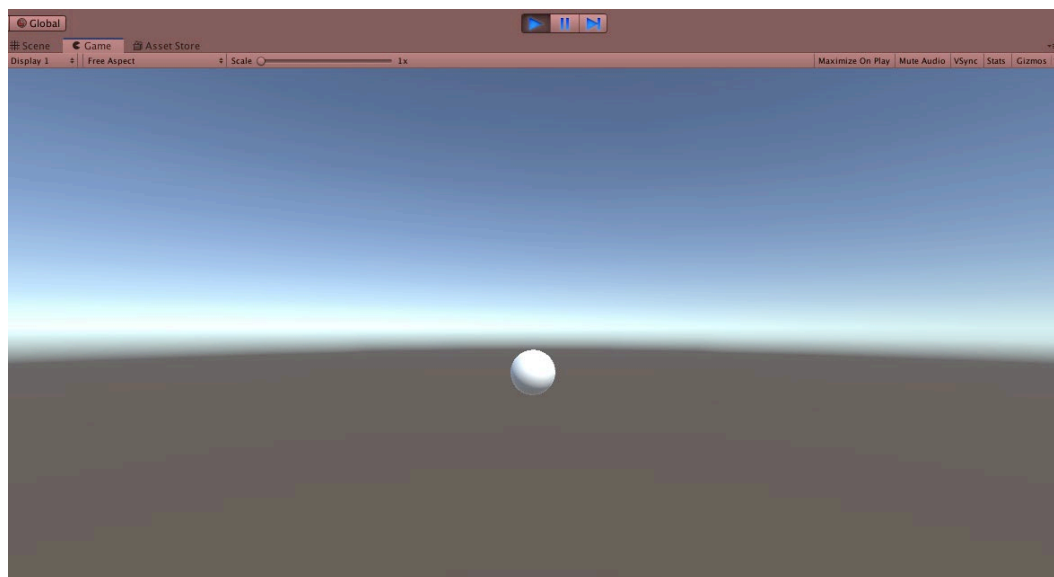


Εικόνα 4-6: Γραμμή εργαλείων

Υπάρχουν δύο προβολές στην Unity· η προβολή σκηνής (Scene View) και η προβολή παιχνιδιού (Game View). Η προβολή σκηνής χρησιμοποιείται, κυρίως, κατά τη διάρκεια της ανάπτυξης, καθώς επιτρέπεται να επιλέξεις οποιοδήποτε αντικείμενο, και να χρησιμοποιήσεις οποιοδήποτε από τα εργαλεία είναι διαθέσιμα, όπως για παράδειγμα, τα προαναφερθέντα στη γραμμή εργαλείων. Αυτές οι αλλαγές που θα πραγματοποιηθούν στην προβολή σκηνής, αποθηκεύονται στο project, εν αντιθέσει με την προβολή παιχνιδιού, όπου οι αλλαγές παραμένουν για όσο είσαι στην συγκεκριμένη προβολή. Μόλις βγεις από αυτή, αναιρούνται, και επιστρέφεις στις αλλαγές που είχαν γίνει στην σκηνή, κατά την προβολή σκηνής.



Εικόνα 4-7: Εργαλείο μετακίνησης επιλεγμένο (Προβολή σκηνής — Scene View)

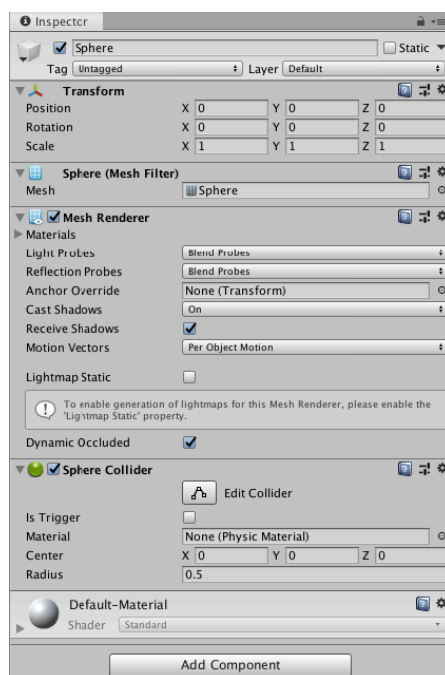


Εικόνα 4-8: Τρισδιάστατο (Προβολή παιχνιδιού — Game View)

Στην προβολή παιχνιδιού (Game view) υπάρχει και ένας ολισθητής (slider) ο οποίος μας βοηθά να μεγεθύνουμε (zoom) την προβολή μας.

4.2.1 Επιθεωρητής (Inspector)

Ο επιθεωρητής είναι από τα πιο σημαντικά μέρη της διεπαφής χρήστη, καθώς μας προβάλλει τις ρυθμίσεις αλλά και τις πληροφορίες του αντικειμένου, ή των αντικειμένων που έχουν επιλεγεί.

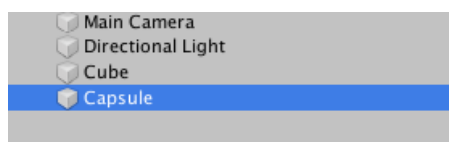


Εικόνα 4-9: Επιθεωρητής με επιλεγμένη μια σφαίρα

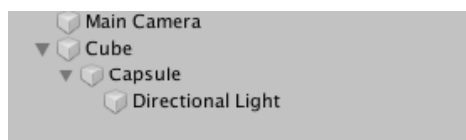
Στην κορυφή του επιθεωρητή (Εικόνα 4-9), υπάρχει ένα πεδίο με το οποίο μπορούμε να αλλάξουμε το όνομα ενός αντικειμένου, καθώς και να το ενεργοποιήσουμε/ απενεργοποιήσουμε χρησιμοποιώντας το checkbox που βρίσκεται αριστερά του. Επίσης, από τον επιθεωρητή μπορείς να αλλάξεις την ετικέτα (tag) και το επίπεδο (layer) του αντικειμένου. Πιο κάτω, υπάρχει η δυνατότητα να αλλάξεις τη θέση, την περιστροφή και το μέγεθος του αντικειμένου —ρυθμίσεις που, όπως προαναφέρθηκε, μπορούν να αλλάξουν και από την γραμμή εργαλείων. Παρακάτω είναι όλα τα μέρη (components) που υπάρχουν στο επιλεγμένο αντικείμενο, καθώς επίσης, υπάρχει η δυνατότητα να προστεθούν νέα μέρη.

4.2.2 Ιεραρχία (Hierarchy)

Το παράθυρο ιεραρχίας, είναι το παράθυρο που εμφανίζονται όλα τα αντικείμενα που βρίσκονται στην σκηνή, όπως τα αντικείμενα, τα φώτα, οι κάμερες και πολλά άλλα. Αυτό το παράθυρο μπορεί να χρησιμοποιηθεί για να δημιουργήσουμε, να αφαιρέσουμε, να αλλάξουμε την ιεραρχία των αντικειμένων και πολλά άλλα. Όσον αφορά την ιεραρχία των αντικειμένων, μπορούμε να έχουμε τα αντικείμενα σε μια οριζόντια ιεραρχία (siblings — αδέρφια, Εικόνα 4-10) ή μπορεί να τα έχουμε σε κάθετη ιεραρχία [38] (parent/child — γονέα/παιδιού, Εικόνα 4-10).



Εικόνα 4-10: Οριζόντια ιεραρχία



Εικόνα 4-11: Κάθετη ιεραρχία

4.2.3 Scripts

Με την χρήση των scripts εισάγεται λογική στο παιχνίδι. Τα scripts αυτά πρέπει να γράφονται σε γλώσσα προγραμματισμού C# και για να χρησιμοποιηθούν οι δυνατότητες της Unity, πρέπει να επεκταθούν με την κλάση MonoBehaviour. Παρακάτω, στον αλγόριθμο 4-1, είναι ένα script στην αρχική του μορφή, μετά τη

δημιουργία του. Οι μέθοδοι, `start` και `update` είναι οι μέθοδοι που ορίζουν πότε θα εκτελεστεί μια μέθοδος που δημιουργείται. Για παράδειγμα, αν μια μέθοδος τοποθετηθεί μέσα στην `start`, τότε θα εκτελεστεί μια φορά, κατά την εκκίνηση της σκηνής —πιο συγκεκριμένα, ακριβώς μετά την ενημέρωση του πρώτου frame. Ενώ, αν τοποθετηθεί μέσα στην `Update`, θα εκτελείται σε κάθε frame —δηλαδή σε κάθε ανανέωση της οθόνης.

```
public class ExampleClass : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        //
    }

    // Update is called once per frame
    void Update()
    {
        //
    }
}
```

Αλγόριθμος 4-1: Start, Update μέθοδοι

Τα scripts έχουν απεριόριστες δυνατότητες, καθώς σου επιτρέπουν να δημιουργήσεις, ό,τι δημιουργείς «με το χέρι» στην Unity, να το αναπτύξεις και με πηγαίο κώδικα.

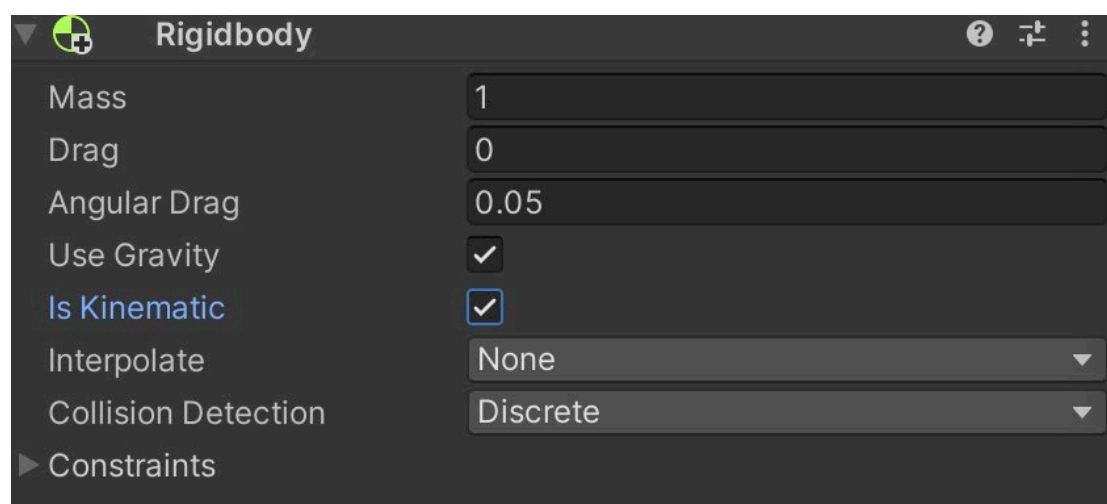
4.3 Unity Physics

Ένα από τα πιο σημαντικά χαρακτηριστικά της Unity, είναι τα Unity Physics. Αυτή η λειτουργικότητα της Unity, έχει χρησιμοποιηθεί, διαχρονικά, σε διάφορους τομείς όπως σε επιστήμες, στη μελέτη και το λογισμικό. Χωρίς την χρήση των Physics, δεν θα ήταν δυνατό να αναπτυχθούν παιχνίδια —ακόμη και τα πιο απλά, όπως ένα παιχνίδι που αναπηδά μια μπάλα.

Παρακάτω αναφέρονται τα κύρια μέρη των Unity Physics που χρησιμοποιήθηκαν στην παρούσα εργασία.

Rigidbody

Το Rigidbody είναι το κύριο μέρος (component) το οποίο δίνει δυνατότητες Φυσικής (Unity Physics) σε ένα αντικείμενο. Η προεπιλεγμένη και βασική συμπεριφορά ενός αντικειμένου με Rigidbody είναι να υπακούει στο νόμο της βαρύτητας. Μέσω του Rigidbody υπάρχει η δυνατότητα να εφαρμοστούν δυνάμεις σε ένα αντικείμενο, ώστε να το ελέγξεις με ρεαλιστικό τρόπο. Αν σε ένα σενάριο απαιτείται να υπάρχει Rigidbody —μόνο για την χρήση Colliders (αναλύεται παρακάτω)— αλλά να μην υπακούει στους νόμους των Unity Physics, υπάρχει η δυνατότητα να ενεργοποιηθεί η Κινηματική ιδιότητα (is Kinematic, Εικόνα 4-12). Με την ενεργοποίηση της Κινηματικής, πλέον το αντικείμενο μπορεί να μετακινηθεί μέσω του script.



Εικόνα 4-12: Ιδιότητα Is Kinematic

Colliders

Οι colliders καθορίζουν τη συμπεριφορά ενός αντικειμένου, με σκοπό την προσομοίωση συγκρούσεων [39]. Οι συγκρούσεις είναι ένα πολύ βασικό στοιχείο της ανάπτυξης σκηνών στην Unity, καθώς μέσω αυτών ενεργοποιούνται διάφορα συμβάντα. Για παράδειγμα, αν το παιχνίδι είναι ένας αγώνας μπάσκετ, τότε το συμβάν όπου η μπάλα θα χτυπήσει εκτός των επιτρεπτών ορίων του γηπέδου, είναι μια σύγκρουση με μια περιοχή της σκηνής, όπου θα πυροδοτήσει ένα συμβάν· τη διακοπή του παιχνιδιού, την διακοπή της χρονομέτρησης, καθώς και το σφύριγμα από τον διαιτητή μεταξύ άλλων. Το σχήμα του collider, μπορεί να είναι ακριβώς ίδιο με αυτό του αντικειμένου που έχει ανατεθεί, αλλά αυτό δεν είναι αναγκαία

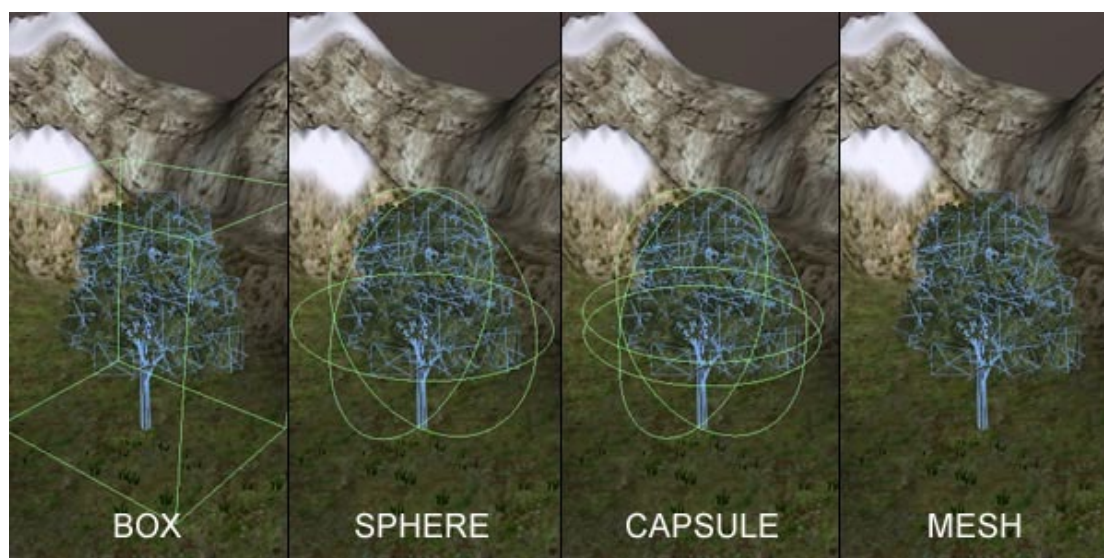
προϋπόθεση.

Στο τρισδιάστο πεδίο, υπάρχουν οι

- Box Colliders
- Sphere Colliders
- Capsule Colliders
- Mesh Colliders

4.3.1.1 Box/ Sphere/ Capsule/ Circle Colliders

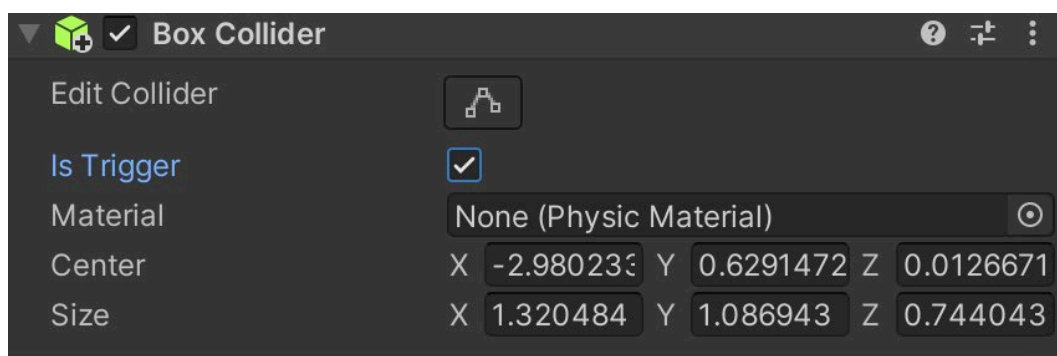
Αυτοί οι colliders περιβάλλουν με έναν αόρατο σχήμα (Εικόνα 4-13) ένα αντικείμενο και αυτό καθορίζει την επιφάνεια με την οποία το αντικείμενο θα συγκρούεται με τα υπόλοιπα αντικείμενα της σκηνής. Επειδή δεν έχουν μεγάλη ακρίβεια στην κάλυψη του αντικειμένου σε σχέση με το σχήμα του, είναι χρήσιμοι γιατί δεν απαιτούν πολλούς υπολογιστικούς πόρους. Είναι ιδανικά για απλά γεωμετρικά σχήματα, αλλά και σε περιπτώσεις όπου δεν απαιτείται ακριβής αναπαράσταση του σχήματος για τον έλεγχο μιας σύγκρουσης. Οι Mesh Colliders αναπαριστούν ακριβώς το σχήμα του αντικειμένου και, παράλληλα, είναι απαιτητικοί ως προς την κατανάλωση υπολογιστικών πόρων.



Εικόνα 4-13: Unity colliders [\[35\]](#)

Για την ανίχνευση των συγκρούσεων και στο script, χρησιμοποιούνται οι μέθοδοι OnCollisionEnter, και OnCollisionExit. Η πρώτη ανιχνεύει την στιγμή που ο collider έρχεται σε επαφή με ένα άλλο αντικείμενο, ενώ η δεύτερη, όταν αυτή η επαφή πάψει να υφίσταται. Σε περίπτωση που χρειάζεται να ανιχνευθεί η επαφή,

αλλά να μην συγκρουστεί το αντικείμενο και εκτραπεί από την πορεία του, πρέπει να επιλεγθεί η ιδιότητα `isTrigger` (Εικόνα 4-14). Με αυτή την ιδιότητα και τις μεθόδους, `OnTriggerEnter` και `OnTriggerExit`, έχουμε το αντίστοιχο αποτέλεσμα με τις παραπάνω μεθόδους, μόνο που τώρα πλέον το αντικείμενο που έρχεται σε επαφή με τον collider δεν συγκρούεται, αλλά το διαπερνά [39, 40].

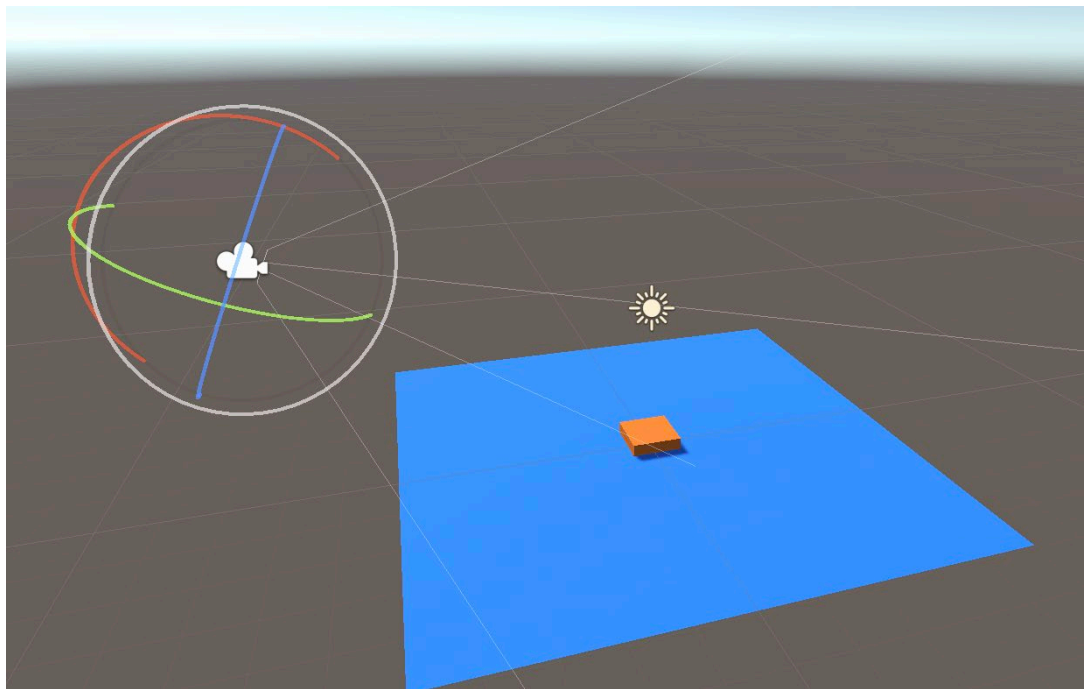


Εικόνα 4-14: Ιδιότητα *Is Trigger*

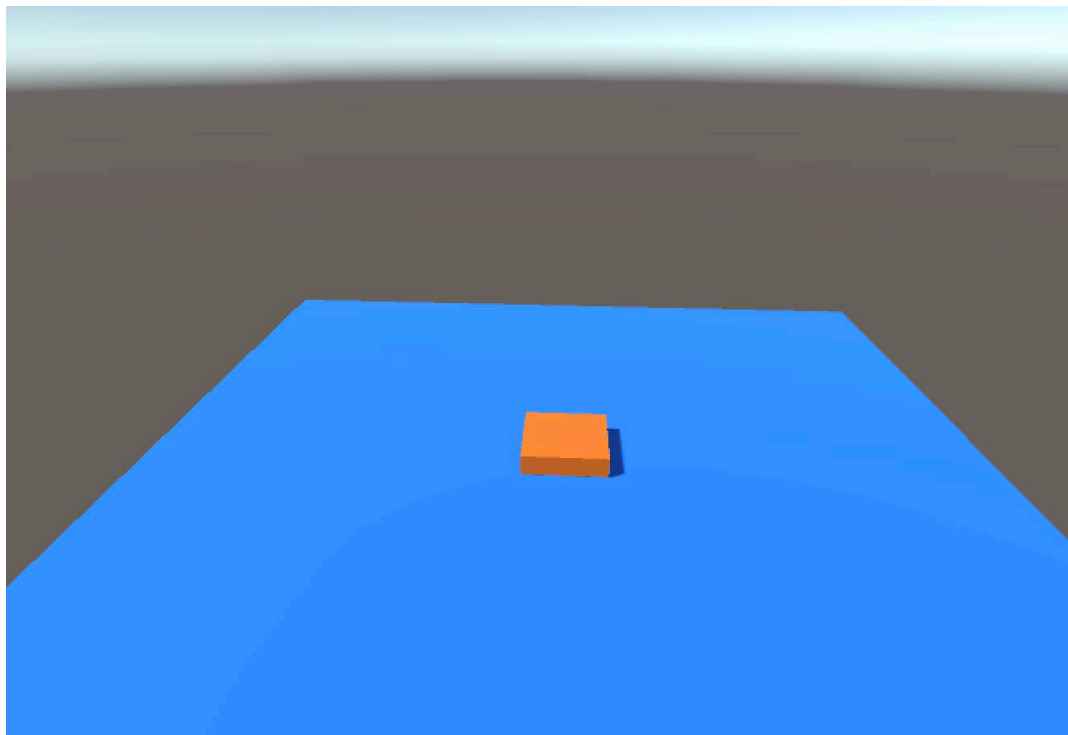
4.3.2 Κάμερες (Cameras)

Η Unity αναπαριστά δεδομένα στον τρισδιάστατο χώρο και καθώς η οθόνη του χρήστη είναι δισδιάστατη, με την χρήση των καμερών δημιουργεί ένα τρισδιάστατο περιβάλλον [41, 42]. Επίσης, παρέχει τη δυνατότητα για χρήση κάμερας με ορθογραφική προβολή (ιδανική για δισδιάστατα παιχνίδια), κάμερες προοπτικής προβολής (οι κλασικές κάμερες που βλέπουμε σε τρισδιάστατα παιχνίδια). Η κάμερα είναι υπεύθυνη για την προοπτική του χρήστη καθώς η σκηνή

είναι σε λειτουργία Play (Εικόνα 4-16).



Εικόνα 4-15: Χρήση κάμερας στην Unity (Ανάπτυξη)



Εικόνα 4-16: Χρήση κάμερας στην Unity (Play mode)

4.3.3 Φωτισμός (Lighting)

Ο φωτισμός στην Unity, «συμπεριφέρεται», σχεδόν, όπως ο φωτισμός στον

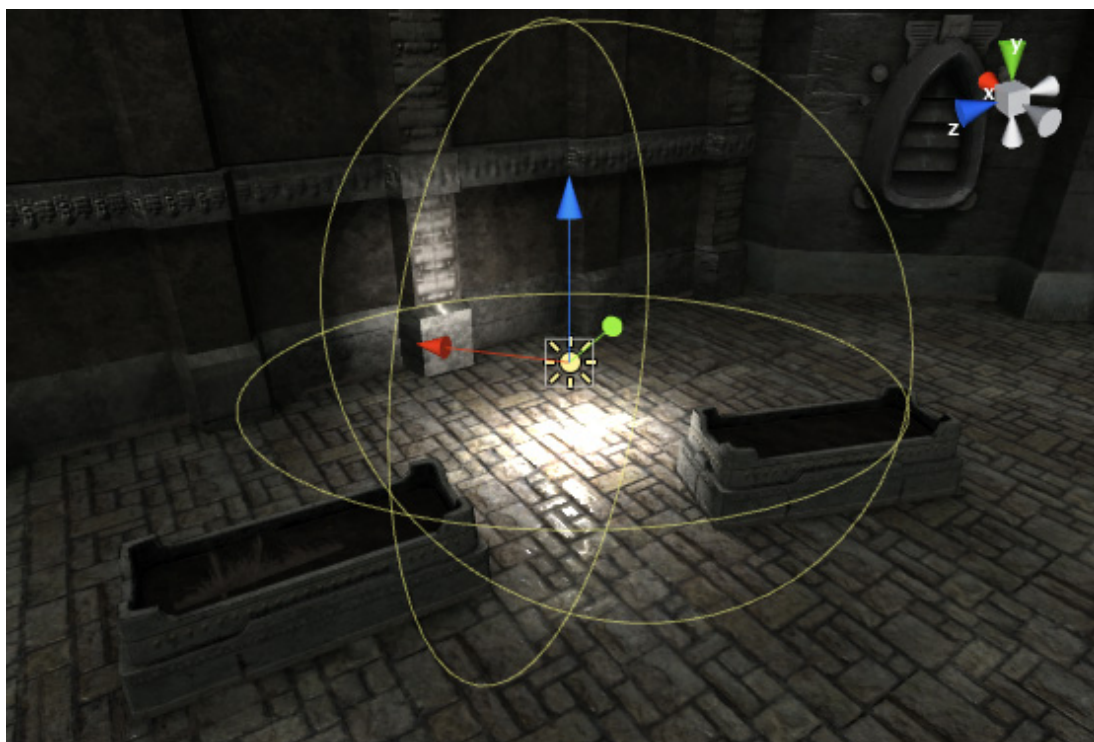
φυσικό κόσμο και είναι ένα από τα βασικά στοιχεία της Unity, για την επίτευξη μιας ελκυστικής αλλά και ρεαλιστικής σκηνής. Παρέχονται τέσσερα είδη φωτισμού:

Point light: βρίσκεται σε ένα σημείο της σκηνής και εκπέμπει, εξίσου, προς όλες τις κατευθύνσεις (Εικόνα 4-17).

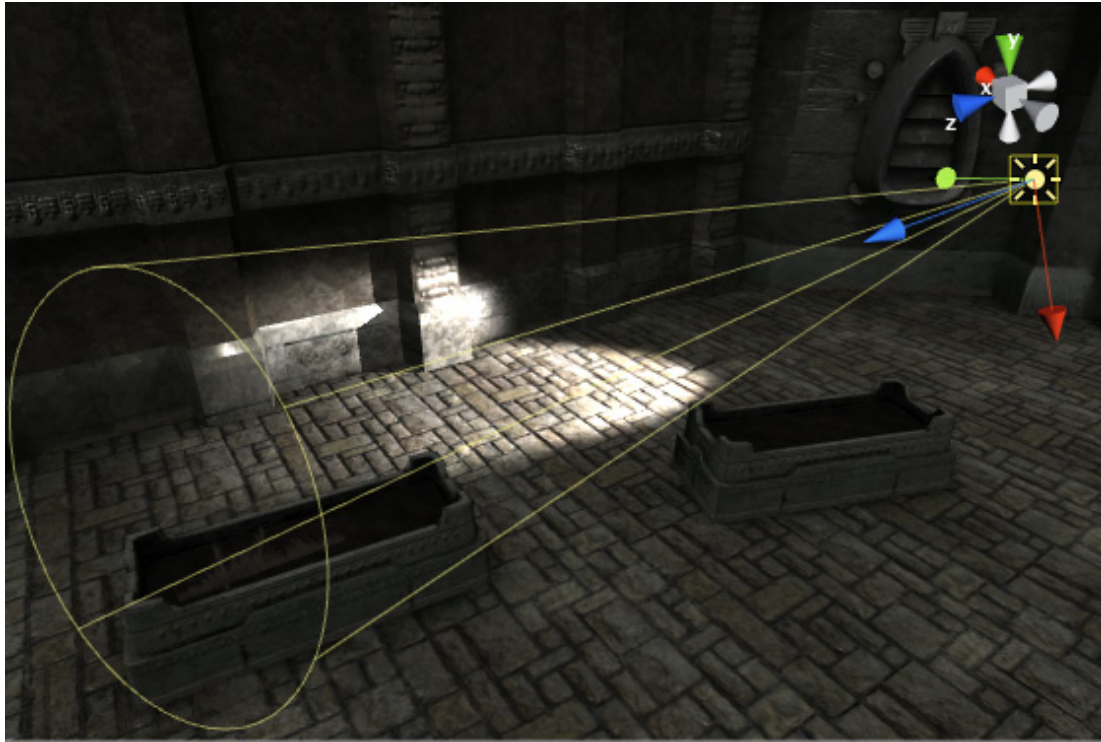
Spot light: βρίσκεται σε ένα σημείο της σκηνής και εκπέμπει με κωνικό σχήμα (Εικόνα 4-18).

Directional (Κατευθυνόμενο): βρίσκεται σε ένα σημείο πολύ μακριά από τη σκηνή και εκπέμπει προς μια κατεύθυνση (Εικόνα 4-19).

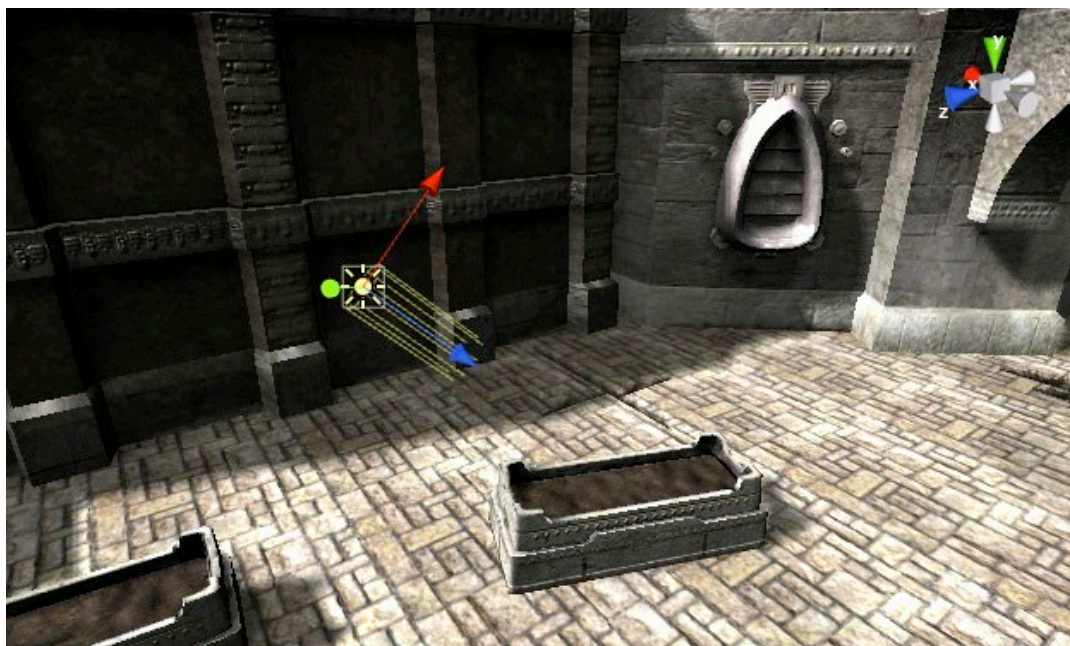
Area Light (Περιοχή φωτός): ορίζεται με ένα παραλληλόγραμμο στη σκηνή και εκπέμπει φως, ομοιόμορφα, προς όλες τις κατευθύνσεις σε όλη την επιφάνειά του, αλλά μόνο από τη μία πλευρά του παραλληλόγραμμου (Εικόνα 4-20) [43, 44].



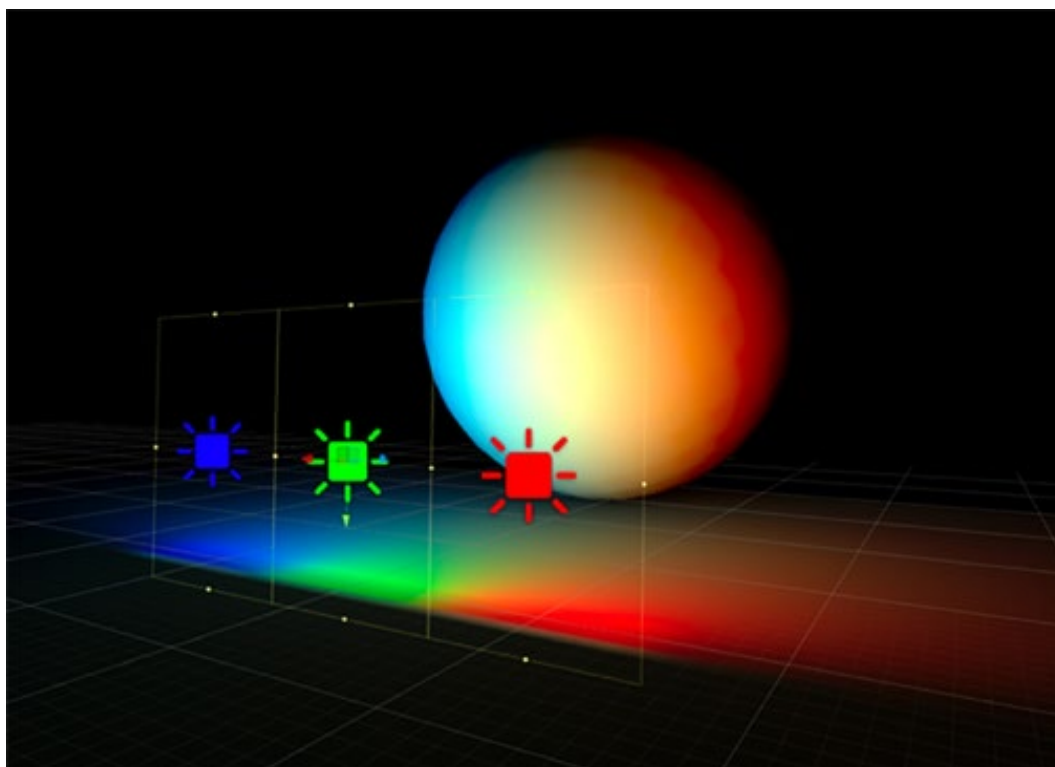
Εικόνα 4-17: Point light [36]



Εικόνα 4-18: Spot light [\[36\]](#)



Εικόνα 4-19: Directional light [\[36\]](#)

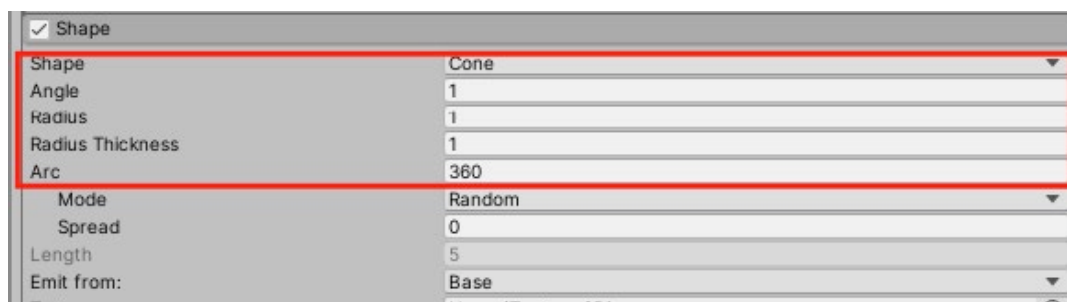


Εικόνα 4-20: Area light [36]

4.3.4 Σύστημα σωματιδίων — Particle System

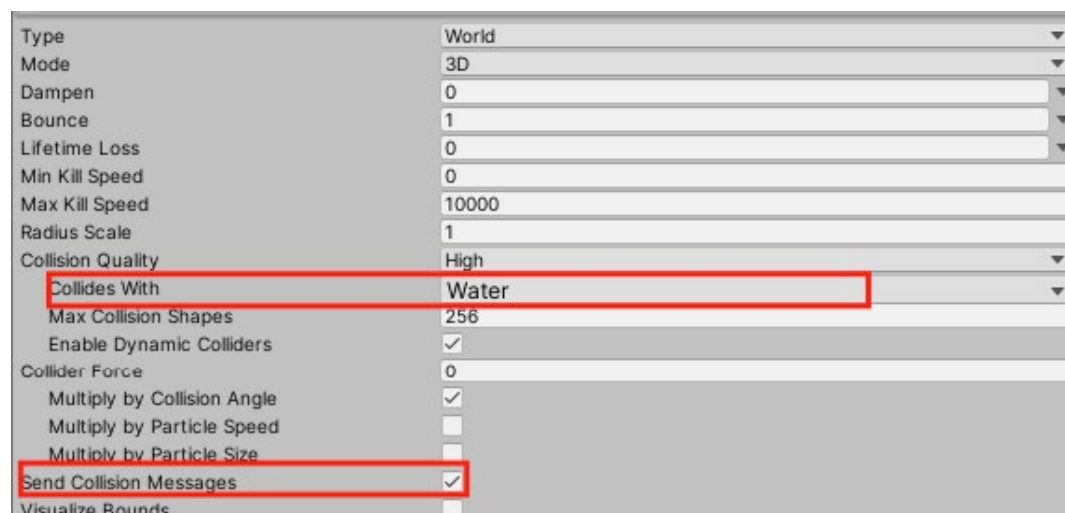
Με το Particle System, η Unity μας επιτρέπει να δημιουργήσουμε διάφορα εφέ, όπως φωτιά, καπνό, υγρά, σύννεφα κ.α. Είναι ένα μέρος της Unity, το οποίο έχει βασιστεί και μεγάλο μέρος της παρούσας εργασίας, καθώς χρησιμοποιείται το Particle System για την φωτιά, τον καπνό και τον κατασβεστικό υλικό του πυροσβεστήρα. Το Particle System έχει πάρα πολλές δυνατότητες παραμετροποίησης· κάποιες από αυτές είναι:

Σχήμα (Shape): Με αυτή τη ρύθμιση καθορίζεται το σχήμα εκπομπής των σωματιδίων. Όπως φαίνεται στην παρακάτω εικόνα (εικόνα 32), υπάρχει η δυνατότητα ρύθμισης, μεταξύ άλλων, το σχήμα, η γωνία εκπομπής, η ακτίνα εκπομπής, το πάχος της ακτίνας κ.α.



Εικόνα 4-21: Ρύθμιση σχήματος των Particles

Σύγκρουση (Collision): Σε αυτή την ενότητα υπάρχει η δυνατότητα να προστεθεί δυνατότητα σύγκρουσης στα σωματίδια (particles). Όπως φαίνεται στην εικόνα 4-22, τα σωματίδια ρυθμίζονται ώστε να συγκρούονται με σωματίδια νερού. Αντιστοίχως ρυθμίζονται και τα σωματίδια του νερού και του καπνού.



Εικόνα 4-22: Ρύθμιση συγκρούσεων των Particles

4.4 Asset Store

Η Unity παρέχει και ένα ηλεκτρονικό κατάστημα, το Asset Store, το οποίο δίνει πάρα πολλές επιλογές για τους προγραμματιστές να αγοράσουν δωρεάν, ή επί πληρωμή, έτοιμα μέρη για να χρησιμοποιηθούν σε μια σκηνή της Unity, όπως:

- Animations
- Χαρακτήρες
- Αυτοκίνητα
- Βιβλιοθήκες
- Σκηνές

Το Unity Asset Store, επιτρέπει στον προγραμματιστή να αγοράσει κάποια έτοιμα μέρη προς χρήση, ελαχιστοποιώντας, με αυτόν τον τρόπο, το χρόνο

ανάπτυξης, καθώς η μοντελοποίηση αντικειμένων είναι μια χρονοβόρα διαδικασία.

Vehicles x

On Sale Only (2)

Realistic Car HD 02
POLYCRAFT
★★★★★ (17)
€31.27
warandoy ★★★★★ 2 years ago
Amazing quality
Deserves 5 stars. Really beautiful model
[Read more](#)

POLYGON - Street Racer
SYNTY STUDIOS
(not enough ratings) | ♥ (102)
€116.12

Mi-8MT Mi-17MT Panels ...
PUKAMAKARA
(not enough ratings) | ♥ (4)
€1,071.11

Mi-8MT Mi-17MT Panels ...
PUKAMAKARA
(not enough ratings) | ♥ (4)
€1,071.11

Low Poly 3D Vehicles Ga...
AAANIMATORS
(not enough ratings) | ♥ (22)
€892.44

Low Poly Mega Vehicle Pa...
LINDER MEDIA
(not enough ratings) | ♥ (69)
€335

The Big Spaceship Builder
TRIPLEBRICK
★★★★☆ (3) | ♥ (56)

"TITANIC"
AGLOBEX
★★★★★ (4) | ♥ (121)
€133.11

ARCade: Ultimate Vehicle...
MENA
★★★★★ (8) | ♥ (170)
€44.67

All Categories
3D (38912)
Animations (782)
Characters (11341)
Environments (7806)
GUI (16)
Props (11369)
Vegetation (1064)
☒ Vehicles (5056)
2D (8508)
Add-Ons (59)
Audio (6576)
Essentials (45)
Templates (2719)
Tools (9087)
VFX (2636)

Pricing +
Unity Versions +
Publisher +
Ratings +

Εικόνα 4-23: Επιλογή οχημάτων στο Asset Store

Home > 3D > Characters > Creatures > RPG Character Complete Bundle - 11 Pack Bundle!

RPG Character Complete Bundle - 11 Pack Bundle!
Infinity PBR
★★★★★ 5 | 36 Reviews
€311.77
Taxes/VAT calculated at checkout
171 views in the past week
License type: Single Entity
Refund policy
Add to Cart
Secure checkout: VISA, Mastercard, PayPal, etc.
haohaorena ★★★★★ 20 days ago
good but not enough
good packs, only shortage is not have the cloth material armor pack. it will best to have mental, leather, cloth armor pack.
[Read more reviews](#)

Human Character with Armor i
28/29

Εικόνα 4-24: Επιλογή χαρακτήρων στο Asset Store

5 Υλοποίηση

5.1 Εισαγωγή

Η συγκεκριμένη υλοποίηση αποτελεί ένα ολοκληρωμένο σύστημα προσομοίωσης εξάπλωσης φωτιάς και πυρόσβεσης. Με αυτή την προσομοίωση προσδοκούμε να επιτύχουμε υψηλά μαθησιακά αποτελέσματα· ως εκ τούτου προσπαθήσαμε να πετύχουμε όσο το δυνατόν ρεαλιστικότερο αποτέλεσμα.

Η μοντελοποίηση πυρκαγιάς μπορεί να κατηγοριοποιηθεί σε εμπειρική, ημι-εμπειρική και φυσική [45]. Η εμπειρική μοντελοποίηση χρησιμοποιεί στατιστικές σχέσεις και η φυσική μοντελοποίηση χρησιμοποιεί συναγωγή και μεταφορά θερμότητας. Η ημι-εμπειρική προσέγγιση χρησιμοποιεί τόσο εμπειρική, όσο και φυσική για τη δημιουργία υβριδικής μοντελοποίησης.

Σε αυτή την έρευνα, αρχικά, χρησιμοποιήσαμε μαθηματικές εξισώσεις που είναι ευρέως αποδεκτές στη βιβλιογραφία που σχετίζονται με τη φωτιά. Λάβαμε υπόψη πολλές πτυχές του υλικού ενός αντικειμένου, όπως η πυκνότητα, το μέγεθος, ο όγκος, η γεωμετρία κ.λπ. και αναπτύξαμε έναν νέο αλγόριθμο που μπορούσε να υπολογίσει τη μείωση της μάζας και τη διάδοση της φωτιάς. Ωστόσο αποδείχθηκε μια πολύπλοκη λύση και με, πιθανόν, αναξιόπιστα αποτελέσματα, με αποτέλεσμα, να στραφούμε στη λύση FDS, η οποία είναι ένα καθιερωμένο λογισμικό που έχει δοκιμαστεί σε πολλά έργα και χρησιμοποιείται ευρέως για μοντελοποίηση πυρκαγιάς [46-51]. Μέσω των δεδομένων που εξάγει ο FDS, εξάγουμε την «πληροφορία» την οποία εισάγουμε στο Unity, ώστε να οπτικοποιήσουμε την εξάπλωση, βάσει των δεδομένων αυτών.

5.2 Αρχιτεκτονική του συστήματος

Αυτή η έρευνα στοχεύει στη διερεύνηση και τη μοντελοποίηση μιας πυρκαγιάς σε εσωτερικούς χώρους, και στη συνέχεια να δημιουργήσει ένα περιβάλλον εμπύθισης, με την εξέλιξη της φωτιάς καθώς και την κατάσβεση. Έχει σχεδιαστεί, και αναπτυχθεί, μια ελκυστική σκηνή δωματίου με την Unity 3D.

Τα αντικείμενα της σκηνής έχουν επισημανθεί ως εύφλεκτα, αυτόματα, από τα scripts που έχουν αναπτυχθεί. Αντίστοιχα, η φωτιά έχει την αντίστοιχη ετικέτα, ώστε να διαχωρίζεται η συμπεριφορά της από τη συμπεριφορά των εύφλεκτων αντικειμένων. Όλες οι γεωμετρίες των εύφλεκτων αντικειμένων εισήχθησαν στο

FDS για τον υπολογισμό της επίδρασης της φωτιάς σε αυτά. Πριν ξεκινήσει η προσομοίωση, ορίζουμε ένα χρονικό βήμα που είναι ο ρυθμός που εκτελείται η προσομοίωση. Μετά την εκτέλεση της προσομοίωσης, το FDS εξάγει πολλά αρχεία με τα αποτελέσματα του μοντέλου.

Αυτά τα αποτελέσματα περιγράφουν την εξάπλωση της φωτιάς στη συγκεκριμένη γεωμετρία με βάση διάφορους παράγοντες όπως το υλικό, το μέγεθος, τον όγκο αυτής της γεωμετρίας και άλλα. Με βάση αυτό το σύνολο δεδομένων, εστιάζουμε στις τιμές του ρυθμού απώλειας μάζας και στο χρόνο που χρειάζεται για να κάψει η φωτιά τη γεωμετρία. Έπειτα, δημιουργήσαμε μεθόδους για να ανακτήσουμε αυτά τα δεδομένα για να χρησιμοποιηθούν στον αλγόριθμο.

5.3 FDS

Η φωτιά είναι ένα από τα πιο περίπλοκα φαινόμενα, καθώς ενσωματώνει μια μεγάλη ποικιλία φυσικών και χημικών διεργασιών. Αυτές οι διεργασίες περιλαμβάνουν καύση, ορμή, μάζα και μεταφορά θερμότητας, που αλληλοεπιδρούν για να σχηματίσουν ένα εξαιρετικά περίπλοκο φυσικό και χημικό φαινόμενο. Η εξαιρετικά περίπλοκη φύση της φωτιάς ήταν ένας από τους λόγους για τους οποίους η λεπτομερής επιστημονική έρευνα καθυστέρησε μέχρι τη δεκαετία του 1950 [52].

Τα τελευταία χρόνια, έχουν σημειωθεί σημαντικά βήματα στη μελέτη τόσο της πυρκαγιάς όσο και του τρόπου αντιμετώπισής της (πυρασφάλεια). Καθοριστική συμβολή είχε η ταχεία ανάπτυξη των υπολογιστών σε συνδυασμό με την πρόοδο της Υπολογιστικής Μηχανικής Ρευστών. Επομένως, η υπολογιστική προσομοίωση του φαινομένου της φωτιάς, για αυτό το έργο, στηρίχθηκε στο διαδομένο και ελεύθερο λογισμικό υπολογιστικής ρευστομηχανικής FDS (Fire Dynamics Simulator) [53, 54], το οποίο αναπτύχθηκε από το Εθνικό Ινστιτούτου Προτύπων και Τεχνολογίας (NIST-National Institute for Standards and Technology) των ΗΠΑ, το οποίο εξειδικεύεται στην υπολογιστική προσομοίωση φωτιάς.

Πρόκειται για ένα πρόγραμμα Fortran που διαβάζει παραμέτρους εισόδου από ένα αρχείο πηγαίου κώδικα (script.fds), υπολογίζει μια αριθμητική λύση στις εξισώσεις που διέπουν και γράφει δεδομένα εξόδου που καθορίζονται από τον χρήστη σε αρχεία (.csv). Το συνοδευτικό πρόγραμμα που διαβάζει αρχεία εξόδου FDS και παράγει την οπτικοποίηση του φαινομένου στην οθόνη του υπολογιστή είναι το Unity. Όπως αναφέρουν οι δημιουργοί του, «το μοντέλο επιλύει αριθμητικά μια μορφή εξισώσεων Navier-Stokes κατάλληλη για ροές θερμικής

κίνησης χαμηλής ταχύτητας με έμφαση στη μεταφορά καπνού και θερμότητας από πυρκαγιές» [54].

Ο FDS επιτυγχάνει μια ρεαλιστική και ακριβής προσέγγιση στην περιγραφή εξάπλωσης πυρκαγιάς, λόγω των προηγμένων μαθηματικών τύπων και μοντέλων. Επομένως η οπτικοποίηση της εξάπλωσης πυρκαγιάς σε ένα σύστημα VR είναι πιο ρεαλιστική εισάγοντας ακριβή δεδομένα, τα οποία παρέχονται από το FDS.

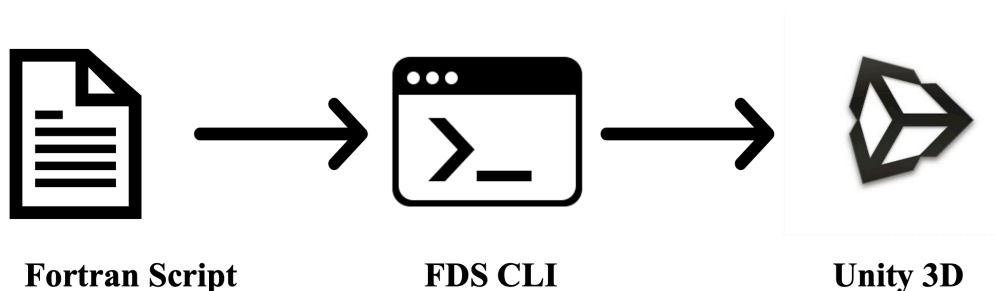
Το μοντέλο μας επικεντρώθηκε κυρίως σε λειτουργίες FDS και στα δεδομένα που αυτό μας παρέχει ως εξόδους (CSV datasets). Οι έξοδοι του FDS χρησιμοποιούνται ως είσοδοι στο Unity 3D. Για το σχεδιασμό επιλέξαμε την Unity καθώς εμφανίζει σημαντικά πλεονεκτήματα στην οπτικοποίηση ενός τέτοιου μοντέλου και μας παρέχει λειτουργικότητα για τη δημιουργία ενός διαδραστικού περιβάλλοντος.

5.4 Εργαλειοποίηση του FDS

Ο FDS χρησιμοποιεί το μοντέλο κλάσμα μείγματος. Αυτό το μοντέλο «υποθέτει» πως η καύση ελέγχεται από φαινόμενα ανάμιξης καυσίμων υλικών και οξείδωσης αλλά και πως η αντίδραση του καυσίμου υλικού με το οξειδωτικό συμβαίνει πολύ γρήγορα. Επιπρόσθετα μας δίνει την δυνατότητα να μοντελοποιήσουμε μια γεωμετρία, αλλά και ένα σύνολο γεωμετριών. Εμείς επιλέξαμε να μοντελοποιήσουμε την κάθε γεωμετρία, που θα χρησιμοποιήσουμε στη σκηνή, ξεχωριστά. Η είσοδος του FDS είναι ένα FORTRAN script, στο οποίο πρέπει να δηλωθούν οι επιθυμητές ιδιότητες της γεωμετρίας. Για συντόμευση της διαδικασίας ανάπτυξης, χρησιμοποιήθηκαν κάποια πρότυπα scripts του FDS, σε συγκεκριμένες γεωμετρίες επίπλων, τα οποία μετετράπησαν στις επιθυμητές γεωμετρίες, αλλά και με τις απαραίτητες ιδιότητες.

5.5 Ροή ανάπτυξης της προσομοίωσης

5.5.1 FDS



Εικόνα 5-1: Ροή ανάπτυξης της σκηνής

5.5.2 Unity

Στην Unity, έχουν αναπτυχθεί τα κατάλληλα scripts τα οποία διαβάζουν τα csv, κάνουν προεπεξεργασία των δεδομένων (data preprocessing), και αποθηκεύουν τα απαιτούμενα δεδομένα, τα οποία και θα αναλυθούν στην επόμενη ενότητα.

5.6 Ανάπτυξη της προσομοίωσης στην Unity

5.6.1 Εισαγωγή

Η σκηνή του εκπαιδευτικού περιβάλλοντος αναπαριστά μια σχολική μονάδα. Πιο συγκεκριμένα, μια σχολική αίθουσα διδασκαλίας (εικόνα 5-2), ένα διάδρομο (εικόνα 5-3) ενώ στην μια πλευρά του διαδρόμου υπάρχει μια έξοδος κινδύνου (εικόνα 5-4). Είναι μια προσομοίωση τύπου βιντεοπαιχνίδι-βολής-πρώτου προσώπου (FPS — First Person Shooter) όπου ο χρήστης-εκπαιδευόμενος κρατά ένα πυροσβεστήρα (εικόνα 5-5). Επιπρόσθετα, στη σκηνή —και πιο συγκεκριμένα στο διάδρομο— υπάρχουν τοποθετημένοι πυροσβεστήρες διαφορετικών τύπων.



Εικόνα 5-2: Αίθουσα διδασκαλίας



Εικόνα 5-3: Διάδρομος



Εικόνα 5-4: Έξοδος κινδύνου

5.7 Σενάριο

Η προσομοίωση ξεκινά με τα θρανία της αίθουσας να έχουν πιάσει φωτιά. Όσο η φωτιά εξαπλώνεται, ο χρήστης-εκπαιδευόμενος καλείται να κατασβέσει τη φωτιά, διαλέγοντας τον κατάλληλο πυροσβεστήρα για το φλεγόμενο υλικό. Γι' αυτό το λόγο, στον διάδρομο υπάρχουν τοποθετημένοι πυροσβεστήρες (Εικόνες 5-5, 5-6),



Εικόνα 5-5: Πυροσβεστήρες τοποθετημένοι στον διάδρομο — επιγραφές



Εικόνα 5-6: Πυροσβεστήρες τοποθετημένοι στον διάδρομο

έτσι ώστε ο χρήστης να έχει δυνατότητα επιλογής. Επιπλέον, τοποθετήθηκαν και κάποιες μεγάλες επιγραφές (Εικόνες 5-6, 5-7) που περιγράφουν τις «δυνατότητες» του πυροσβεστήρα: πιο συγκεκριμένα, τι τύπου φωτιές μπορεί να σβήσει. Καθώς ο χρήστης πλησιάζει στην περιοχή του πυροσβεστήρα, ένα μενού αναδύεται και ο χρήστης ενημερώνεται για τον πυροσβεστήρα που έχει επιλέξει (Εικόνα 5-7).



Εικόνα 5-7: Αναδύόμενο παράθυρο που ενημερώνει ποιος πυροσβεστήρας επιλέχθηκε

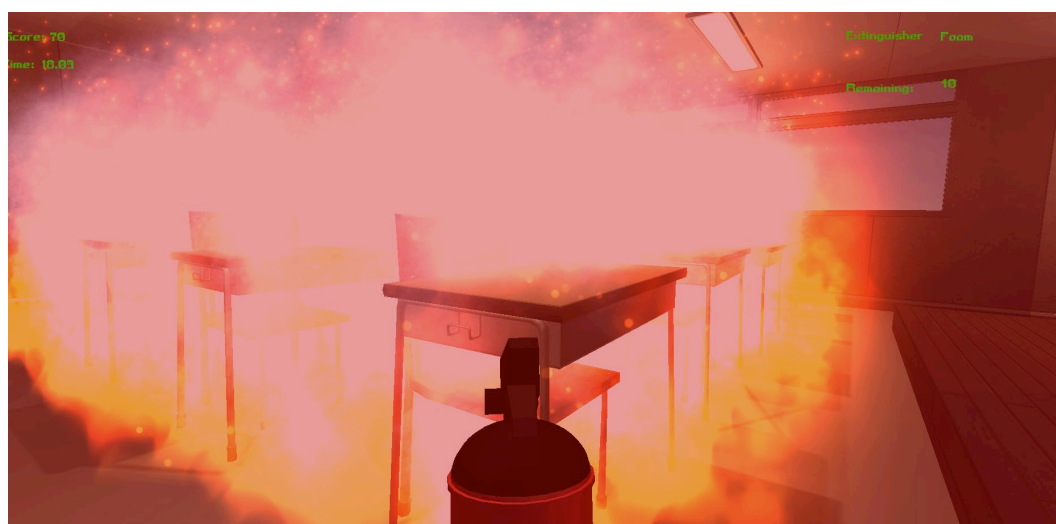
Το παιχνίδι έχει ένα σύστημα πόντων, όπου ο χρήστης ξεκινά με 100 πόντους. Όσο σβήνει τη φωτιά (Εικόνα 5-8), έχοντας επιλέξει το σωστό

πυροσβεστήρα, αυξάνονται οι πόντοι, ενώ όσο προσπαθεί να σβήσει τη φωτιά με λάθος πυροσβεστήρα, μειώνονται οι πόντοι του.



Εικόνα 5-8: Διαδικασία πυρόσβεσης

Όπως φαίνεται στην εικόνα 5-8, στην οθόνη του χρήστη φαίνεται το score που έχει συγκεντρώσει, ο χρόνος που έχει περάσει, ο επιλεγμένος πυροσβεστήρας, καθώς και το υπόλοιπο του κατασβεστικού υλικού που περιέχει ο πυροσβεστήρας. Επίσης, όταν ο χρήστης έρθει σε επαφή με τη φωτιά, η οθόνη κοκκινίζει —ως ένδειξη ότι έχει «πληγωθεί»— δίνοντας μια αίσθηση βιντεοπαιχνιδιού (Εικόνα 5-9).



Εικόνα 5-9: Ο χρήστης έρχεται σε επαφή με τη φωτιά

5.7.1 Δεδομένα για την απομείωση της μάζας

Όπως προαναφέρθηκε, για κάθε γεωμετρία που πρέπει να μοντελοποιηθεί η καύση της, εκτελείται μια προσομοίωση στον FDS. Ο FDS εξάγει πάρα πολλά

δεδομένα, ωστόσο για την απομείωση της μάζας χρησιμοποιείται ένα συγκεκριμένο dataset, με την κατάληξη `hrr.csv`. Από αυτό το dataset, εξάγονται δύο συγκεκριμένες στήλες —ο ρυθμός απομείωσης της μάζας και τα χρονικά σημεία που συμβαίνει αυτό (timestamps). Από αυτό το dataset εξάγουμε δύο σημαντικές πληροφορίες για το κάθε αντικείμενο:

- Τη συνολική μάζα του αντικειμένου
- Τον χρόνο πλήρους καύσης

Εν συνεχεία, στην Unity, αναπτύσσονται οι κατάλληλες μέθοδοι για την προεπεξεργασία των δεδομένων (data preprocessing) —στην περίπτωσή μας, την μετατροπή και ανάθεση των δεδομένων του csv (comma separated value) αρχείου, σε δομές δεδομένων όπως οι λίστες (Αλγόριθμος 5-1).

```
private void FillingMassLossDataLists(IReadOnlyList<CsvRecord> result) {  
    for (var i = 2; i < result[MlrIndex].Data.Length; i++) {  
        timestampsList.Add(result[TimeIndex].Data[i]);  
        massLossRateList.Add(result[MlrIndex].Data[i]);  
    }  
  
    int index = timestampsList.FindIndex(j => j > 0);  
    step = timestampsList[index + 1] - timestampsList[index];  
    CalculateTotalMass();  
}
```

Αλγόριθμος 5-1: Ανάθεση των στηλών του dataset σε λίστες και εύρεση του βήματος (step)

Σε αυτή τη μέθοδο, επίσης, υπολογίζεται το βήμα της προσομοίωσης (step) το οποίο εξάγεται αφαιρώντας τον χρόνο του κάθε βήματος προσομοίωσης, στον FDS, με το προηγούμενο του. Σύμφωνα με αυτό το βήμα τρέχει και ο αλγόριθμος που απομειώνει την μάζα.

5.7.2 Συνολική μάζα (Total mass)

Για να υπολογιστεί η συνολική μάζα, εκτελείται μια επανάληψη στην λίστα (for loop), η οποία περιέχει τις ποσότητες μάζας που αφαιρούνται κατά την καύση, οσότου μηδενιστεί η μάζα. Αυτά τα κομμάτια μάζας που αφαιρούνται κατά την καύση, αθροίζονται μέσω της επανάληψης και ανατίθενται στην ιδιότητα `flammableProps.totalMass`, όπως φαίνεται και στον παρακάτω αλγόριθμο (Αλγόριθμος 5-2). Το `flammableProps` είναι ένα στιγμιότυπο της κλάσης `FlammableProps`, η οποία περιγράφει τις ιδιότητες του κάθε φλεγόμενου αντικειμένου, ενώ η μεταβλητή `totalMassPreview` είναι υπεύθυνη για την προβολή της συνολικής μάζας του αντικειμένου στον επιθεωρητή (Inspector) της Unity

(Αλγόριθμος 5-3).

```
private void CalculateTotalMass() {  
    for (var i = 0; i < massLossRateList.Count; i++) {  
        flammableProps.totalMass += massLossRateList[i];  
    }  
  
    totalMassPreview = flammableProps.totalMass;  
}
```

Αλγόριθμος 5-1: Υπολογισμός συνολικής μάζας

5.7.3 Χρόνος πλήρους καύσης (Full burn time)

Για να υπολογιστεί ο συνολικός χρόνος καύσης, εκτελείται μια επανάληψη στη λίστα της απομείωσης μάζας (massLossRateList), και γίνεται έλεγχος εωσότου βρεθεί το σημείο της λίστας όπου η αφαιρούμενη μάζα είναι μηδενική, όπου και θεωρείται πως το αντικείμενο έχει καεί πλήρως (Αλγόριθμος 5-3).

Με αυτά τα δεδομένα ελέγχουμε την πορεία της μάζας του αντικειμένου καθώς καίγεται, και βάσει αυτού, μόλις η μάζα μηδενιστεί, η φωτιά σβήνει και μένει μόνο το στοιχείο του καπνού. Το στοιχείο της φωτιάς θα περιγραφεί σε επόμενη ενότητα.

```
for (var i = 2; i < massLossRateList.Count; i++) {  
    if (massLossRateList[i] == 0) {  
        fullBurnIndexList.Add(i);  
    }  
}  
  
fullBurnIndex = fullBurnIndexList[0];  
fullBurnTime = timestampsList[fullBurnIndex];
```

Αλγόριθμος 5-2: Υπολογισμός σημείου πλήρους καύσης

Ο σκοπός του υπολογισμού της συνολικής μάζας του αντικειμένου, είναι να προσομοιωθεί η αντίστροφη διαδικασία· η διαδικασία απομείωσης της μάζας καθώς το αντικείμενο φλέγεται. Παρακάτω παρουσιάζεται ο αλγόριθμος (Αλγόριθμος 5-4)

για την απομείωσης της μάζας:

```
private void ReduceMass()
{
    if (flammableProps.totalMass > 0 && flammableProps.IsOnFire)
    {
        var temp = Decimal.Parse(massLossRateList[0].ToString(), System.Globalization.NumberStyles.Any);
        flammableProps.totalMass -= (float) temp;
        totalMassPreview = flammableProps.totalMass;
        massLossRateList.RemoveAt(0);
    }

    if (flammableProps.totalMass ≤ 1)
    {
        flammableProps.IsOnFire = false;
        dictCtrl.objectsOnFire.Remove(gameObject);
    }
}
```

Αλγόριθμος 5-3: Μέθοδος απομείωσης μάζας

Αρχικά, αυτή η μέθοδος ξεκινά μόλις ένα αντικείμενο πιάσει φωτιά και τρέχει επαναληπτικά με ρυθμό, ίσο με το βήμα που έχει υπολογιστεί στον αλγόριθμο 5-1 (μεταβλητή step). Όσο η μάζα του φλεγόμενου αντικειμένου είναι μεγαλύτερη του μηδέν, η μέθοδος «διαβάζει» το πρώτο στοιχείο από την massLossRateList (την λίστα που περιέχει τις ποσότητες προς αφαίρεση) αφαιρεί την ποσότητα που αντιστοιχεί σε αυτό το στοιχείο από τη συνολική μάζα του αντικειμένου και, τέλος, αφαιρεί αυτό το πρώτο στοιχείο. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να μηδενιστεί η μάζα. Επιπρόσθετα, όταν η μάζα είναι κάτω από το 1, κάνουμε μια παραδοχή πως το αντικείμενο δεν είναι πλέον onFire —δηλαδή έχει πάψει να φλέγεται— και αφαιρούμε το αντικείμενο από τη λίστα των φλεγόμενων αντικειμένων που έχει προστεθεί. Αυτή η παραδοχή έγινε προς αποφυγή απρόβλεπτων εξαιρέσεων για την εφαρμογή, που θα οδηγούσε σε κρίσιμα σφάλματα.

5.7.4 Δεδομένα για την θερμοκρασία του φλεγόμενου αντικειμένου.

Με την χρήση του fds2ascii (εντολή του FDS) μπορούμε να πάρουμε πληροφορία σχετικά με την θερμοκρασία της φωτιάς κατά τη διάρκεια της καύσης ενός αντικειμένου. Στην δική μας προσομοίωση έχουμε επιλέξει να έχουμε 5 datasets για κάθε αντικείμενο, και το κάθε ένα από αυτά ορίζει την θερμοκρασία της φωτιάς για 10 δευτερόλεπτα. Επομένως παρακολουθείται η πορεία της θερμοκρασίας της φωτιάς για 50 δευτερόλεπτα (το χρονικό εύρος παρακολούθησης μπορεί να επεκταθεί όσο θέλουμε). Για παράδειγμα, μπορούμε να τρέξουμε ένα

simulation 50 δευτερολέπτων σε πέντε παράθυρα (0-10, 11-20, 21-30, 31-40, 41-50), το οποίο εξάγει datasets με την θερμοκρασία για κάθε συντεταγμένη του υπολογιστικού χώρου, που έχει δημιουργηθεί στον FDS. Η προσομοίωση παρέχει δεδομένα για τη θερμοκρασία, ανά ζεύγη (x-y, x-z, y-z), ωστόσο, εμείς λαμβάνουμε υπόψη μόνο το ζεύγος συντεταγμένων x-z, διότι ελέγχουμε την εξάπλωση της φωτιάς μέσω πλέγματος που ελέγχει το πάτωμα (περιγράφεται παρακάτω, λεπτομερώς, στην ενότητα 5.7.5).

Στο κομμάτι της ανάπτυξης, έπρεπε να αναπτυχθεί ο αντίστοιχος αλγόριθμος που θα αλλάζει το dataset που αναλύεται, κάθε 10 δευτερόλεπτα. Με την εκκίνηση της σκηνής ένας χρονομετρητής ξεκινά για να βοηθήσει στην ανάθεση των παραθύρων (μέθοδος GetTimeFrame στον αλγόριθμο 5-5). Θέτουμε ένα αρχικό χρόνο (initialTime) καθώς και τον χρόνο προσομοίωσης (simulationTime) και καθ' όλη τη διάρκεια της προσομοίωσης, ο αλγόριθμος συγκρίνει, σε ποιο χρονικό σημείο βρίσκεται, υπολογίζει το τρέχον παράθυρο προσομοίωσης (currentTimeFrame) και αναθέτει το αντίστοιχο dataset προς ανάγνωση.

```
private void GetTimeFrame() {  
    if (flammableProps.IsOnFire) {  
        var timer = flammableProps.Timer;  
        const int initialTime = FlammableProperties.initialTime;  
        var simulationTimeFrame = flammableProps.SimulationTimeFrame;  
        if (timer ≥ initialTime + simulationTimeFrame && timer < initialTime + simulationTimeFrame * 2) {  
            flammableProps.CurrentTimeFrame = 2;  
        }  
  
        if (timer ≥ initialTime + simulationTimeFrame * 2 && timer < initialTime + simulationTimeFrame * 3) {  
            flammableProps.CurrentTimeFrame = 3;  
        }  
  
        if (timer ≥ initialTime + simulationTimeFrame * 3 && timer < initialTime + simulationTimeFrame * 4) {  
            flammableProps.CurrentTimeFrame = 4;  
        }  
  
        if (timer ≥ initialTime + simulationTimeFrame * 4 && timer < initialTime + simulationTimeFrame * 5) {  
            flammableProps.CurrentTimeFrame = 5;  
        }  
    }  
}
```

Αλγόριθμος 5-4: Μέθοδος εύρεσης του παραθύρου προσομοίωσης

```
private void GetDatasetFileName() {  
    var furniturname = gameObject.name;  
    if (flammableProps.CurrentTimeFrame == 1) {  
        datasetFileName = "temperature/" + furniturname + "/" + furniturname + "-T0-" +  
            flammableProps.SimulationTimeFrame;  
    }  
  
    if (flammableProps.CurrentTimeFrame == 2) {  
        datasetFileName = "temperature/" + furniturname + "/" + furniturname + "-T" +  
            (flammableProps.SimulationTimeFrame + 1) + "-" +  
            (flammableProps.SimulationTimeFrame * 2);  
    }  
  
    if (flammableProps.CurrentTimeFrame == 3) {  
        datasetFileName = "temperature/" + furniturname + "/" + furniturname + "-T" +  
            ((flammableProps.SimulationTimeFrame * 2) + 1) + "-" +  
            (flammableProps.SimulationTimeFrame * 3);  
    }  
  
    if (flammableProps.CurrentTimeFrame == 4) {  
        datasetFileName = "temperature/" + furniturname + "/" + furniturname + "-T" +  
            ((flammableProps.SimulationTimeFrame * 3) + 1) + "-" +  
            (flammableProps.SimulationTimeFrame * 4);  
    }  
  
    if (flammableProps.CurrentTimeFrame == 5) {  
        datasetFileName = "temperature/" + furniturname + "/" + furniturname + "-T" +  
            ((flammableProps.SimulationTimeFrame * 4) + 1) + "-" +  
            (flammableProps.SimulationTimeFrame * 5);  
    }  
}
```

Αλγόριθμος 5-5: Μέθοδος ανάκτησης ενός dataset

Στο αλγόριθμο 5-5 φαίνεται η διαδικασία ανάγνωσης του κατάλληλου dataset σχετικά με το χρονικό σημείο της προσομοίωσης. Στο project υπάρχουν τα datasets που έχουν εξαχθεί από το FDS, με το όνομά των αρχείων αυτών να ακολουθεί ένα μοτίβο. Στο δικό μας σενάριο, εισάγοντα πέντε datasets, από τον FDS, στον φάκελο Assets/Resources/temperature/{object name} του Unity. Όπου object name το όνομα του κάθε εύφλεκτου αντικειμένου. Δημιουργήθηκαν δύο ξεχωριστοί φάκελοι για τα datasets για καλύτερη οργάνωση του project: ένας φάκελος περιέχει τα datasets για την μάζα, ενώ ο άλλος τα datasets για την θερμοκρασία. Για να μπορέσει το plugin να διαβάσει δυναμικά όλα τα datasets, πρέπει να ακολουθηθεί ένα συγκεκριμένο μοτίβο (naming convention). Επομένως, τα ονόματα των csv πρέπει να τηρούν τους παρακάτω κανόνες:

Πρώτο συνθετικό: Το όνομα του αντικειμένου στο Unity, μέσω του

οποίου το Unity συνδέει το αντικείμενο με το dataset.

Κατάληξη: Μετά το όνομα του αντικειμένου, ακολουθεί η κατάληξη «-T» ακολουθούμενη, χωρίς κενό από το διάστημα του παραθύρου (**Bed-T0-10.csv**).



Εικόνα 5-2: Desk_chair.csv στον φάκελο Resources -> temperature

Έπειτα, με την μέθοδο GetDatasetFileName υπολογίζεται το όνομα του αρχείου (dataset) που πρέπει να προσπελαστεί. Όπως έχει ήδη προαναφερθεί, το DataAndMassController script, που περιέχει αυτή τη μέθοδο, βρίσκεται σε όλα τα εύφλεκτα αντικείμενα, και καλείται να υπολογίσει, μέσω των παραπάνω μεθόδων, το όνομα του dataset που καλείται να προσπελάσει, για την απομείωση της μάζας, όσο και για την παρακολούθηση της θερμοκρασίας.

Στο ίδιο script και στην μέθοδο OnTriggerEnter, ελέγχονται οι συγκρούσεις των αντικειμένων, και βάσει αυτών, «τροφοδοτούνται» τα λεξικά (dictionaries) — που έχουν δημιουργηθεί γι' αυτό το λόγο— με δεδομένα που θα βοηθήσουν στην παρακολούθηση της εξάπλωσης της φωτιάς. Η OnTriggerEnter, είναι μια μέθοδος της Unity, η οποία εκτελείται όταν δύο αντικείμενα συγκρούονται, όπως και η OnCollisionEnter, ή οποία, ωστόσο, προσομοιώνει μια πιο «φυσική» σύγκρουση, καθώς στην OnTriggerEnter, το ένα αντικείμενο διαπερνά το άλλο, παρόλο που συγκρούονται.

Στον παρακάτω Αλγόριθμο 5-6, πραγματοποιείται έλεγχος, μέσα στην OnTriggerEnter, με ποια εύφλεκτα αντικείμενα έρχονται σε επαφή τα κελιά του πλέγματος και αυτή η πληροφορία αποθηκεύεται στις ιδιότητες του αντικειμένου,

και ειδικότερα, στην λίστα `objectCells`.

```
if (gameObject.CompareTag("Flammable") && triggered.CompareTag("cell")) {

    if (!dictCtrl.cellsInObjectDict.ContainsKey(gameObject.name)) {
        dictCtrl.cellsInObjectDict.Add(gameObject.name, new List<GameObject>());
    }

    if (dictCtrl.cellsInObjectDict.ContainsKey(gameObject.name)) {
        dictCtrl.cellsInObjectDict[gameObject.name].Add(triggered.gameObject);
    }

    if (!flammableProps.objectCells.Contains(triggered.gameObject)) {
        flammableProps.ObjectCells.Add(triggered.gameObject);
    }

}
```

Αλγόριθμος 5-6: Προσθήκη των εύφλεκτων αντικειμένων στη λίστα `ObjectCells`

Η λίστα αυτή, ενδεικτικά, θα μπορούσε να έχει την μορφή:

```
flammableProps.objectsCells = ["cell-11", "cell-12", "cell-15", "cell-16",]
```

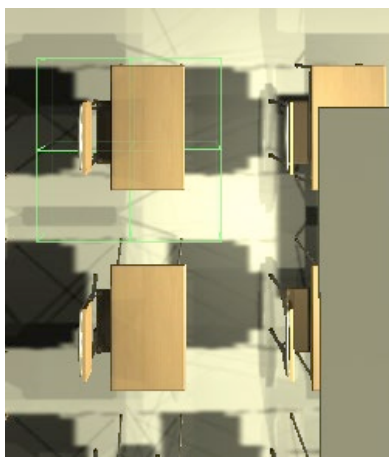
Έπειτα, ελέγχεται ποιό αντικείμενο ακουμπά τη φωτιά για να προστεθεί στη λίστα των `objectsOnFire`, και να τεθεί η ιδιότητά του `isOnFire`, σε `true`. Στον Αλγόριθμο 5-7 ελέγχεται, αν η ετικέτα (tag) του αντικειμένου που φλέγεται, είναι `flammable` (εύφλεκτο) και, επίσης, αν το αντικείμενο το οποίο έρχεται σε σύγκρουση είναι κελί (κελί του πλέγματος — κεφάλαιο 5.7.5).

```
if (triggered.CompareTag("Fire") && flammableProps != null) {
    flammableProps.IsOnFire = true;
    dictCtrl.objectsOnFire.Add(gameObject);
}
```

Αλγόριθμος 5-7: Έλεγχος των φλεγόμενων αντικειμένων

Επίσης γεμίζουμε την λίστα με τα κελιά που «περιέχει» το κάθε αντικείμενο. Όπως φαίνεται στην εικόνα 5-10, το αντικείμενο `Desk` (θρανίο), περιέχει τέσσερα κελιά —τα τέσσερα κελιά που έρχεται σε σύγκρουση το θρανίο θα μούνε στην λίστα `objectCells` του θρανίου, κι έτσι θα έχουμε την πληροφορία αυτή για την

διαχείριση της εξάπλωσης της φωτιάς.



Εικόνα 5-10: Εύρεση των κελιών που περιέχει το κάθε αντικείμενο

Επίσης, αν ένα αντικείμενο έρχεται σε επαφή με τη φωτιά (Αλγόριθμος 5-7), τότε θέτουμε την ιδιότητα του αντικειμένου `IsOnFire` σε `true` και προσθέτουμε το φλεγόμενο αντικείμενο στη λίστα `objectsOnFire` — τα αντικείμενα που έχουν πιάσει φωτιά.

Παράλληλα, ο αλγόριθμος —δεδομένου πως τα δεδομένα της θερμοκρασίας μεταβάλλονται συνεχώς— ενημερώνει τα αντίστοιχα λεξικά (dictionaries) με τις νέες θερμοκρασίες των αντικειμένων, εφόσον το αντικείμενο φλέγεται.

```
private void UpdatingTemperatureDicts() {  
    StartCoroutine(SpreadFireInObjectCells());  
  
    foreach (var cell in flammableProps.objectCells) {  
  
        if (!flammableProps.IsOnFire) continue;  
  
        if (!dictCtrl.cellCurrentTemp.ContainsKey(cell.name) && dictCtrl.cellsOnFireList.Contains(cell)) {  
            dictCtrl.cellCurrentTemp.Add(cell.name, flammableProps.currentTemperature);  
        }  
  
        if (dictCtrl.cellCurrentTemp.ContainsKey(cell.name)) {  
            dictCtrl.cellCurrentTemp[cell.name] =  
                flammableProps.currentTemperature;  
        }  
  
    }  
}
```

Αλγόριθμος 5-8: Ενημέρωση των δεδομένων της θερμοκρασίας των επίπλων

Έπειτα, έπρεπε να δημιουργηθούν οι κατάλληλες μέθοδοι για την ανάπτυξη κλώνων φωτιάς στη σκηνή, βάσει του πλέγματος που αναπτύχθηκε. Αρχικά δημιουργείται μια προσωρινή λίστα, η `fireList`, στην οποία ανατίθενται τα

αντικείμενα της `cellsOnFireList` στο script `DictionaryController`. Η προσωρινή λίστα αυτή, δημιουργείται χάριν αναγνωσιμότητας του κώδικα. Βάσει σεναρίου, πρέπει να αναφλεχθεί το πρώτο αντικείμενο και έπειτα η φωτιά να εξαπλωθεί στα υπόλοιπα αντικείμενα. Αρχικά γίνεται έλεγχος αν αυτό το αντικείμενο είναι το πρώτο (Αλγόριθμος 5-9), καθώς σε αυτό το αρχικό στάδιο μας ενδιαφέρει να εξαπλωθεί η φωτιά, πρώτα στο έπιπλο αυτό.

```
if (gameObject == firstObject) {  
    foreach (var cell in flammableProps.ObjectCells) {  
        fireList.Add(cell);  
    }  
}
```

Αλγόριθμος 5-9: Έλεγχος αν φλέγεται το πρώτο αντικείμενο

Έπειτα, αν δεν είναι το πρώτο έπιπλο, εκτελείται μια σειρά από ελέγχους ώστε να εξασφαλιστεί η μεταφορά της φωτιάς σε γειτονικά κελιά. Οπότε, εκτελείται μια επανάληψη στη λίστα με τα κελιά που φλέγονται και αναζητούνται τα γειτονικά κελιά των ήδη φλεγόμενων κελιών. Εφόσον υπάρχουν τέτοια, και για κάθε ένα από αυτά, εκτελείται μια άλλη μέθοδος η `GenerateAdjacentClones` (Αλγόριθμος 5-10) η οποία δημιουργεί κλώνους φωτιάς στα αμέσως γειτονικά κελιά ενός εύφλεκτου αντικειμένου και όχι σε οποιοδήποτε κελί αυτού του αντικειμένου.

```
private List<GameObject> GenerateAdjacentClones(GameObject cellOnFire) {  
    var adjacentList = new List<GameObject>();  
    foreach (var onFireAdj in fireCellCtrl.findAdjacents(cellOnFire.transform.position)) {  
        if (dictCtrl.positionToObjMap.ContainsKey(onFireAdj)) {  
            var onFireAdjacent = dictCtrl.positionToObjMap[onFireAdj];  
            if (!dictCtrl.cellsOnFireList.Contains(onFireAdjacent)) {  
                adjacentList.Add(onFireAdjacent);  
            }  
        }  
    }  
    return adjacentList;  
}
```

Αλγόριθμος 5-10: Δημιουργία γειτονικών κελιών

```

if (flammableProps.IsOnFire) {

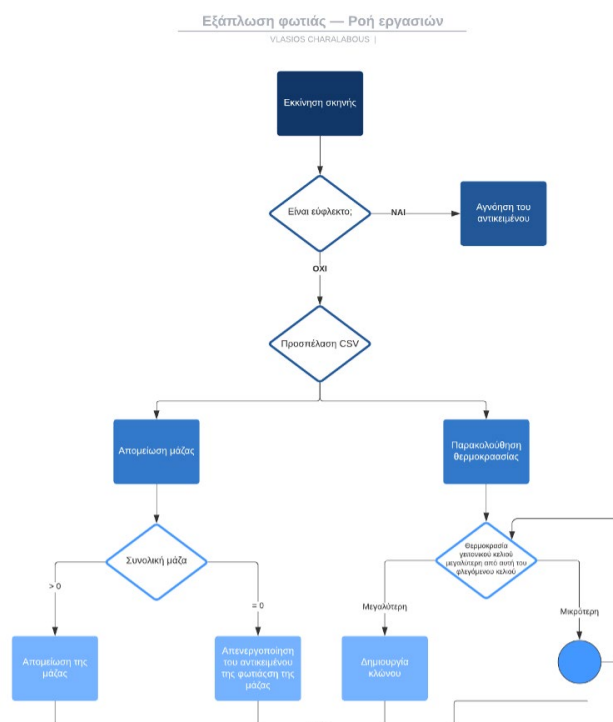
    var fireList = dictCtrl.cellsOnFireList;

    if (gameObject == firstObject) {
        foreach (var cell in flammableProps.ObjectCells) {
            fireList.Add(cell);
        }
    }
    else {
        foreach (var fireCell in fireList.ToList()) {
            if (flammableProps.objectCells.Contains(fireCell)) {
                foreach (var cell in GenerateAdjacentClones(fireCell).ToList()) {
                    if (flammableProps.objectCells.Contains(cell)) {
                        fireList.Add(cell);
                    }
                }
            }
        }
    }
}
}

```

Αλγόριθμος 5-11: Αλγόριθμος εξάπλωσης φωτιάς

Συνοψίζοντας, ο αλγόριθμος θα αναλύει για κάθε έπιπλο ένα dataset για την απομείωση της μάζας, και πέντε datasets για την παρακολούθηση της θερμοκρασίας. Βάσει αυτών, ελέγχεται ο τρόπος της εξάπλωσης της φωτιάς στον χώρο, με τρόπο που θα αναλυθεί λεπτομερώς στις επόμενες ενότητες.



5.7.5 Δημιουργία πλέγματος για τον έλεγχο της εξάπλωσης

Για την εξάπλωση της φωτιάς αναπτύχθηκε ένα μοντέλο πλέγματος (grid

system). Αναπτύχθηκε αλγόριθμος που ανιχνεύει τις γωνίες του δαπέδου της σκηνής και βάσει αυτού δημιουργεί κελιά 1x1 που καλύπτουν όλη τη σκηνή. Αρχικά, αναπτύχθηκε μια μέθοδος για τον υπολογισμό των διαστάσεων της σκηνής. Με την βοήθεια του MeshFilter —το οποίο είναι ένα μέρος το οποίο κρατά μια αναφορά του Mesh— υπολογίστηκαν οι γωνίες της σκηνής (Αλγόριθμος 5-12).

```
private void CalcPlaneCorners(MeshFilter meshFilter, Transform planeTransform) {
    Vector3 extends = meshFilter.mesh.bounds.extents;
    Vector3 bottomLeftPosition = new Vector3(-extends.x, 0, -extends.z);
    Vector3 bottomRightPosition =
        new Vector3(extends.x, 0, -extends.z);
    Vector3 topLeftPosition = new Vector3(-extends.x, 0, extends.z);
    Vector3 topRightPosition = new Vector3(extends.x, 0, extends.z);

    var bottomLeftWorldSpacePosition = planeTransform.TransformPoint(bottomLeftPosition);
    startingPointX = bottomLeftWorldSpacePosition.x;
    startingPointZ = bottomLeftWorldSpacePosition.z;
}
```

Αλγόριθμος 5-12 Υπολογισμός διαστάσεων σκηνής

Έπειτα, έπρεπε να αναπτυχθεί μια μέθοδος η οποία θα δημιουργούσε τους κύβους που θα αποτελούσαν τα κελιά του πλέγματος. Για την δημιουργία του πλέγματος, χρησιμοποιήθηκε το Raycast της Unity, το οποίο είναι μια μέθοδος η οποία με τη βοήθεια μιας ακτίνας (ray) μας επιτρέπει να αναγνωρίσουμε την απόσταση ή την θέση ενός αντικειμένου. Αυτό επιτυγχάνεται με την επιστροφή μιας μεταβλητής Boolean (true, false) η οποία μας ενημερώνει αν η ακτίνα συγκρούστηκε με τον αντικείμενο, ή όχι. Χρησιμοποιήθηκαν οι δυνατότητες αυτής της μεθόδου, για την δημιουργία αυτού του πλέγματος (Εικόνα 5-11) το οποίο μας βοηθά να ελέγξουμε την εξάπλωση της φωτιάς.

```
private void GenerateGrid() {
    RaycastHit hitInfo;
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    var x = grid.startingPointX;
    var z = grid.startingPointZ;
    for (var i = 0; i < cellNumbersInXAxis; i++) {
        for (var j = 0; j < cellNumbersInZAxis; j++) {
            if (Physics.Raycast(ray, out hitInfo) && counter <= (cellNumbersInXAxis * cellNumbersInZAxis)) {
                PlaceCubeNear(new Vector3(x, 0.474f, z));
                counter++;
            }
            else {
                return;
            }
        }
        z += cellsSize;
    }

    z = grid.startingPointZ;
    x += cellsSize;
}
```

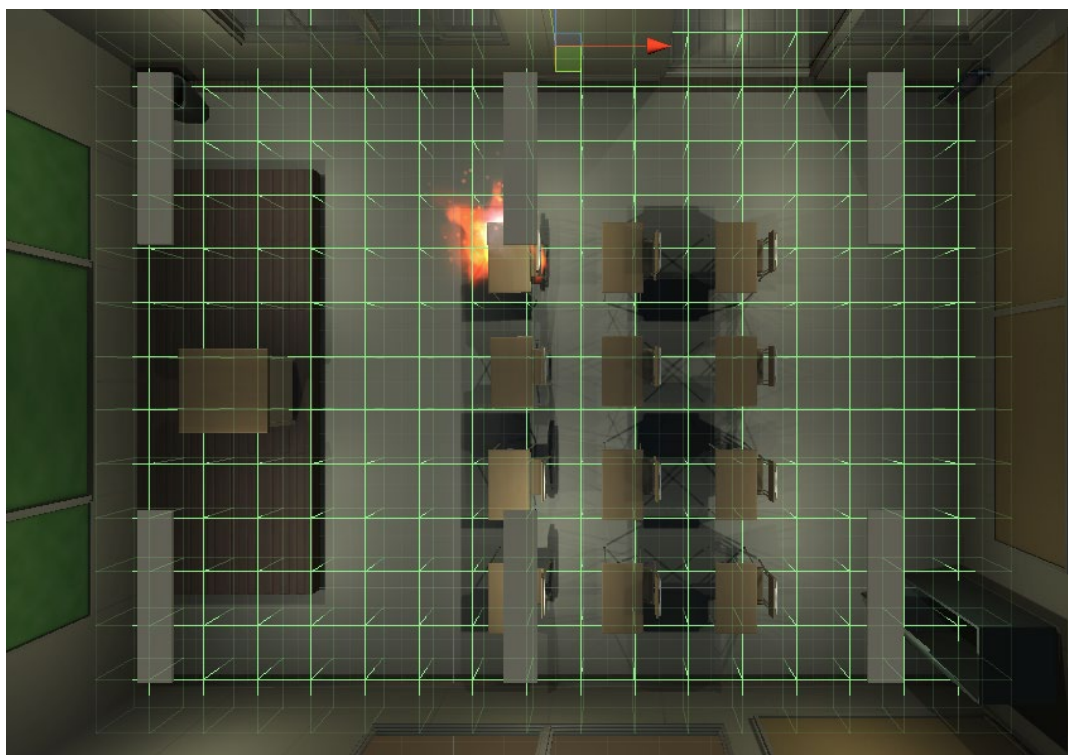
Αλγόριθμος 5-13: Μέθοδος δημιουργίας πλέγματος

Επίσης, σε κάθε κελί που δημιουργείται, προστίθενται και κάποιες απαραίτητες ιδιότητες, όπως τα μέρη RigidBody, Collider, και CollisionController, που είναι απαραίτητα για την ανίχνευση σύγκρουσης, καθώς επίσης, ορίζεται και η θέση του κελιού, η οποία έχει υπολογιστεί από προηγούμενη μέθοδο. Τέλος, στο κελί δίνονται όνομα με αύξοντα αριθμό, και ετικέτα.

```
private void AddCellProperties(GameObject cube, Vector3 finalPosition) {  
    cube.AddComponent<Rigidbody>();  
    cube.AddComponent<CollisionController>();  
    cube.GetComponent<Rigidbody>().isKinematic = true;  
    cubeRenderer = cube.GetComponent<Renderer>();  
    cube.GetComponent<Collider>().isTrigger = true;  
    cube.transform.position = finalPosition;  
    cube.name = "cell-" + counter;  
    cube.tag = "cell";  
    cubeRenderer.enabled = false;  
}
```

Αλγόριθμος 5-14: Προσθήκη ιδιοτήτων στα κελιά

Μέσω αυτών των κελιών μπορούμε να ελέγξουμε όλα τα αντικείμενα σε ποια θέση βρίσκονται, ποια είναι τα γειτονικά τους αντικείμενα, αλλά και να αποθηκεύσουμε τις ιδιότητες των αντικειμένων αυτών στα κελιά.



Εικόνα 5-11: Το πλέγμα στην σκηνή του Unity

Για την υλοποίηση της προσέγγισης του πλέγματος χρησιμοποιήθηκαν οι Unity Colliders (ενότητα 1.7.2).



Εικόνα 5-12: Box colliders στις περιοχές επιλογής πυροσβεστήρα

Μια ακόμη δυνατότητα του Collider, είναι η επιλογή `isTrigger`. Με την επιλογή αυτή, μπορούμε να διαπεράσουμε τον collider, και να εντοπίσουμε αυτή τη στιγμή της σύγκρουσης, με την μέθοδο `OnTriggerEnter`. Ανατίθενται σε όλα τα κελιά του πλέγματος ένας collider με το `isTrigger` επιλεγμένο, και μέσω αυτού ανιχνεύεται ποια αντικείμενα περιέχει το κάθε κελί. Αποθηκεύεται προσωρινά στην μνήμη η πληροφορία αυτή, χρησιμοποιώντας δομές δεδομένων όπως οι λίστες και τα λεξικά.

Εν κατακλείδι, με την εκκίνηση της σκηνής ο αλγόριθμος ανακτά δεδομένα από τα datasets και το πλέγμα, και αποθηκεύει σε δομές δεδομένων πληροφορίες όπως:

- Ποια είναι η τρέχουσα θερμοκρασία ενός κελιού
- Ποια είναι η θερμοκρασία ανάφλεξης ενός κελιού
- Με ποια κελιά έρχεται σε επαφή το κάθε έπιπλο.
- Ποια κελιά είναι σε επαφή με τη φωτιά
- Ποια είναι τα γειτονικά κελιά κάθε κελιού

5.7.6 Εκκίνηση σκηνής και εξάπλωση της φωτιάς

Με την εκκίνησης της προσομοίωσης ο αλγόριθμος αναθέτει στα

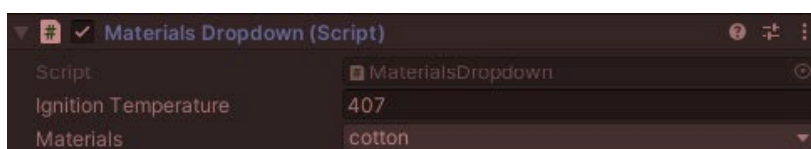
αντικείμενα, που έχουν επιλεγθεί ως εύφλεκτα, την ετικέτα flammable.

```
gameObject.tag = "Flammable";
```

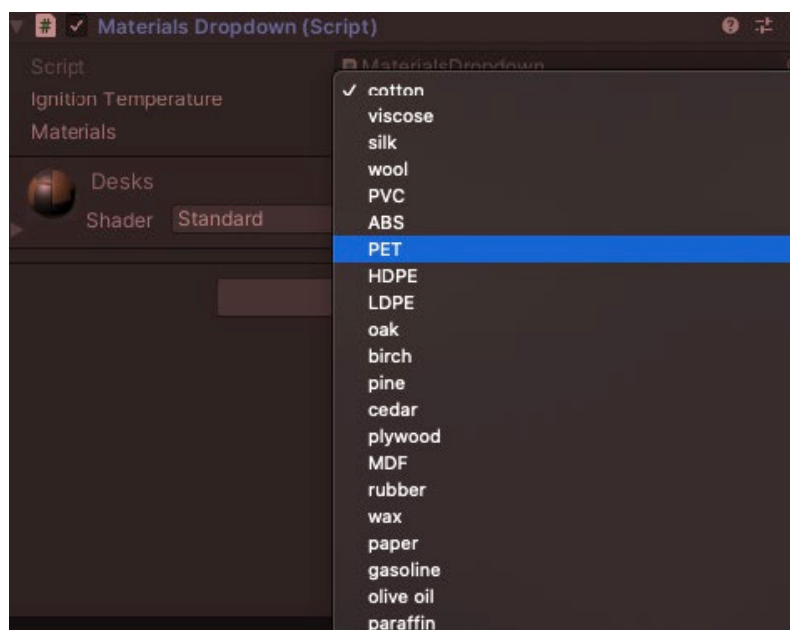
Επίσης, τους αναθέτει ένα script το οποίο μας επιτρέπει να επιλέξουμε τον τύπο (εικόνα 5-13) του φλεγόμενου υλικού.

```
gameObject.AddComponent<FlammableObjects>();
```

Αυτός ο τύπος, αποτελείται από κάποιες ιδιότητες, αλλά εμείς, σε αυτό το σημείο, θα χρησιμοποιήσουμε μία εξ' αυτών —τη θερμοκρασία ανάφλεξης. Αφού έχει επιλεγθεί η κατηγορία, αυτό το αντικείμενο έχει μια συγκεκριμένη θερμοκρασία ανάφλεξης (εικόνα 16), η οποία χρησιμοποιείται στην διαδικασία της εξάπλωσης.



Εικόνα 5-13: Επιλογή τύπου cotton



Εικόνα 5-14: Επιλογές τύπου υλικού

Βάσει σεναρίου, η φωτιά πρέπει να εξαπλωθεί, πρώτα, σε ένα έπιπλο, και έπειτα στα υπόλοιπα, βάσει των παραμέτρων που έχουν τεθεί. Με την εκκίνηση της σκηνης, ο αλγόριθμος, αρχικά, θέτει όλα τα κελιά του πρώτου επίπλου προς ανάφλεξη, και αρχίζει και δημιουργεί κλώνους φωτιάς σε αυτά. Αυτό πραγματοποιείται ελέγχοντας, πρώτα, το λεξικό με τα κελιά του αντικειμένου και

έπειτα προσθέτει τα κελιά αυτά, σε μια λίστα η οποία τρέχει επαναληπτικά δημιουργώντας κλώνους σε κάθε κελί που προστίθεται.

```
if (flammableProps.IsOnFire) {  
  
    var fireList = dictCtrl.cellsOnFireList;  
  
    if (gameObject == firstObject) {  
        foreach (var cell in flammableProps.ObjectCells) {  
            fireList.Add(cell);  
        }  
    }  
    else {  
        foreach (var fireCell in fireList.ToList()) {  
            if (flammableProps.objectCells.Contains(fireCell)) {  
                foreach (var cell in GenerateAdjacentClones(fireCell).ToList()) {  
                    if (flammableProps.objectCells.Contains(cell)) {  
                        fireList.Add(cell);  
                    }  
                }  
            }  
        }  
    }  
}
```

Αλγόριθμος 5-15: Εξάπλωση της φωτιάς σε διπλανά κελιά

Πιο συγκεκριμένα (αλγόριθμος 5-12), ανατίθεται η λίστα `cellsOnFireList` — η οποία είναι υπεύθυνη για τη δημιουργία κλώνων στην μεταβλητή `fireList` (χάριν αναγνωσιμότητας του κώδικα). Σε περίπτωση που η φωτιά έρχεται σε επαφή με το πρώτο έπιπλο (εκκίνηση σκηνής), ο αλγόριθμος εκτελεί μια επανάληψη, και προσθέτει όλα τα κελιά αυτού του επίπλου στην `fireList`, δηλαδή τα θέτει προς ανάφλεξη. Αν το έπιπλο που έρχεται σε επαφή με τη φωτιά δεν είναι το πρώτο, τότε ο αλγόριθμος εκτελεί μια επανάληψη στη λίστα `fireList` —δηλαδή, τα κελιά που φλέγονται— και για το καθένα από αυτά, εκτελεί την μέθοδο `GenerateAdjacentClones`, η οποία παίρνει ως όρισμα ένα κελί και επιστρέφει μια λίστα από γειτονικά, εύφλεκτα κελιά, τα οποία δεν υπάρχουν στην `fireList`. Τέλος αν ένα κελί πληροί όλες τις συνθήκες που προαναφέρθηκαν, προστίθεται στην λίστα `FireList`, και εν προκειμένω δημιουργείται ένας κλώνος φωτιάς σε αυτό. Στον αλγόριθμο 5-10, γεμίζουμε τη λίστα `cellsOnFireList`, με τα κελιά που αναφέρονται παραπάνω, και εξασφαλίζουμε πως σε κάθε κελί θα υπάρχει ένας, μόνο, κλώνος. Για να το πετύχουμε αυτό, εκτελείται ένα διπλός έλεγχος. Με το μέγεθος της λίστας, αλλά και μέσω ενός μετρητή ο οποίος αυξάνεται κατά ένα κάθε φορά που εισάγεται ένα στοιχείο. Επίσης, κάθε κλώνος παίρνει και ένα μοναδικό όνομα, σχετικά με τη

σειρά που δημιουργείται.

```
foreach (var cell in dictionaryController.cellsOnFireList) {
    if (cloneCount < dictionaryController.cellsOnFireList.Count && !tempList.Contains(cell.name)) {
        tempList.Add(cell.name);
        cloneCount++;
        var clone = Instantiate(fireClone, cell.transform.position, rotation);
        clone.name = "fireClone-" + cloneCount;
        cloneList.Add(clone);
    }
}
```

Αλγόριθμος 5-16: Δημιουργία κλώνων

Αφού η φωτιά εξαπλωθεί στα κελιά του πρώτου αντικειμένου που φλέγεται, αρχίζει ο έλεγχος για τα γειτονικά κελιά (αλγόριθμος 5-12). Ο αλγόριθμος ελέγχει ξανά τις δομές δεδομένων για τις απαιτούμενες πληροφορίες και αρχίζει να συγκρίνει την τρέχουσα τιμή της θερμοκρασίας του φλεγόμενου κελιού, με την θερμοκρασία ανάφλεξης του γειτονικού κελιού, εφόσον περιέχει κάποιο εύφλεκτο αντικείμενο (αλγόριθμος 12).

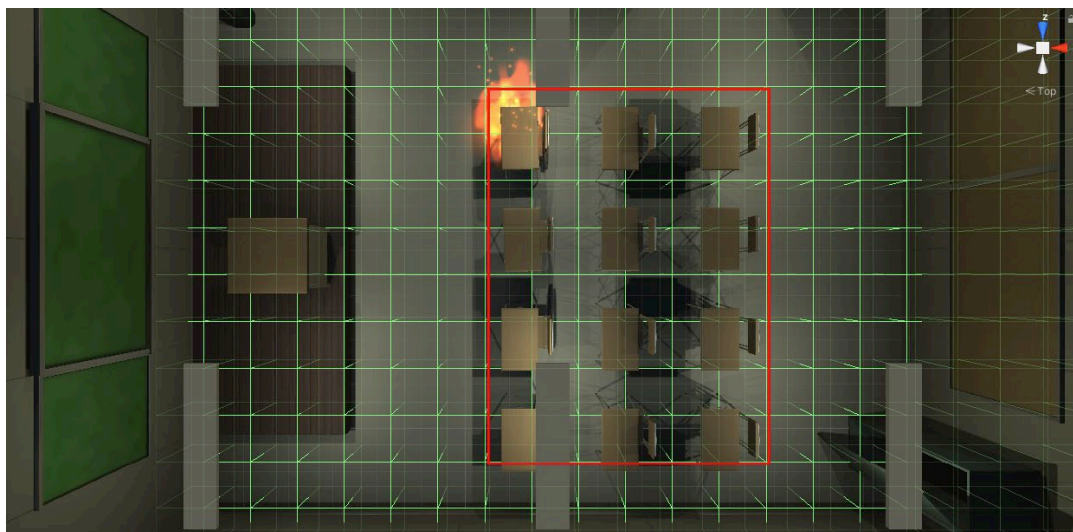
```
if (currentTemp[currentCell.name] > ignitTemp[cellAdjacentName]) {
    onFireList.Add(dictCtrl.positionToObjectMap[currentCellAdjacent]);
}
```

Αλγόριθμος 5-17: Έλεγχος θερμοκρασίας

Ο αλγόριθμος 5-14, βρίσκεται στο script CollisionController, το οποίο είναι σε κάθε κελί του πλέγματος (grid). Έτσι ανιχνεύονται όλες οι συγκρούσεις και αποθηκεύονται οι πληροφορίες, για τα αντικείμενα που συγκρούονται, σε λεξικά (dictionaries). Στο τέλος ελέγχονται τα κελιά που φλέγονται, ένα προς ένα. Ελέγχεται η τρέχουσα θερμοκρασία τους, καθώς και η θερμοκρασία ανάφλεξης των γειτονικών κελιών. Αν η τρέχουσα θερμοκρασία του κελιού που ελέγχεται, είναι μεγαλύτερη από αυτή του γειτονικού κελιού, τότε αυτό το κελί προστίθεται στην λίστα των φλεγόμενων αντικειμένων, και κατά συνέπεια, ένας κλώνος δημιουργείται.

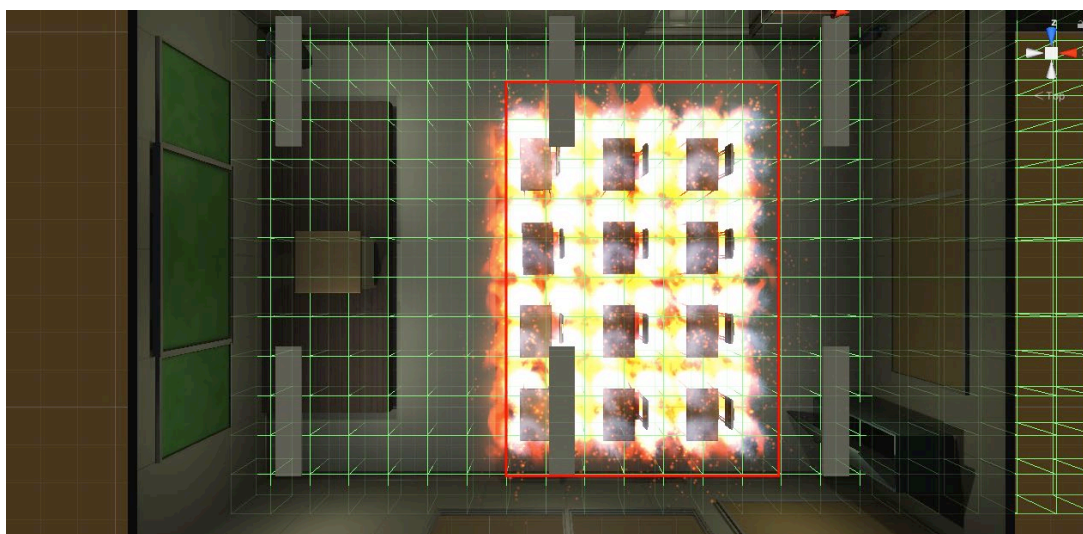
Όπως φαίνεται και στην εικόνα 5-15, η φωτιά ξεκινά από ένα θρανίο και αναμένεται να εξαπλωθεί στα σημεία όπου υπάρχει άμεση γειτνίαση με εύφλεκτα

αντικείμενα —δηλαδή, τα σημεία στο κόκκινο πλαίσιο.

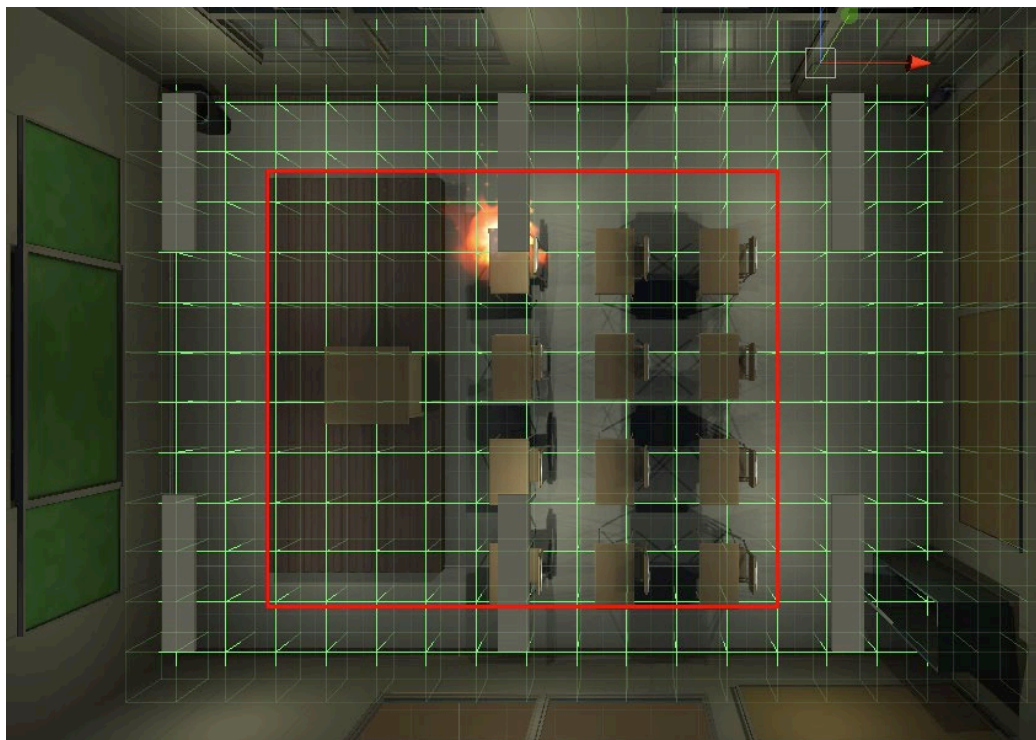


Εικόνα 5-15: Πλαίσιο στο οποίο αναμένεται να εξαπλωθεί η φωτιά

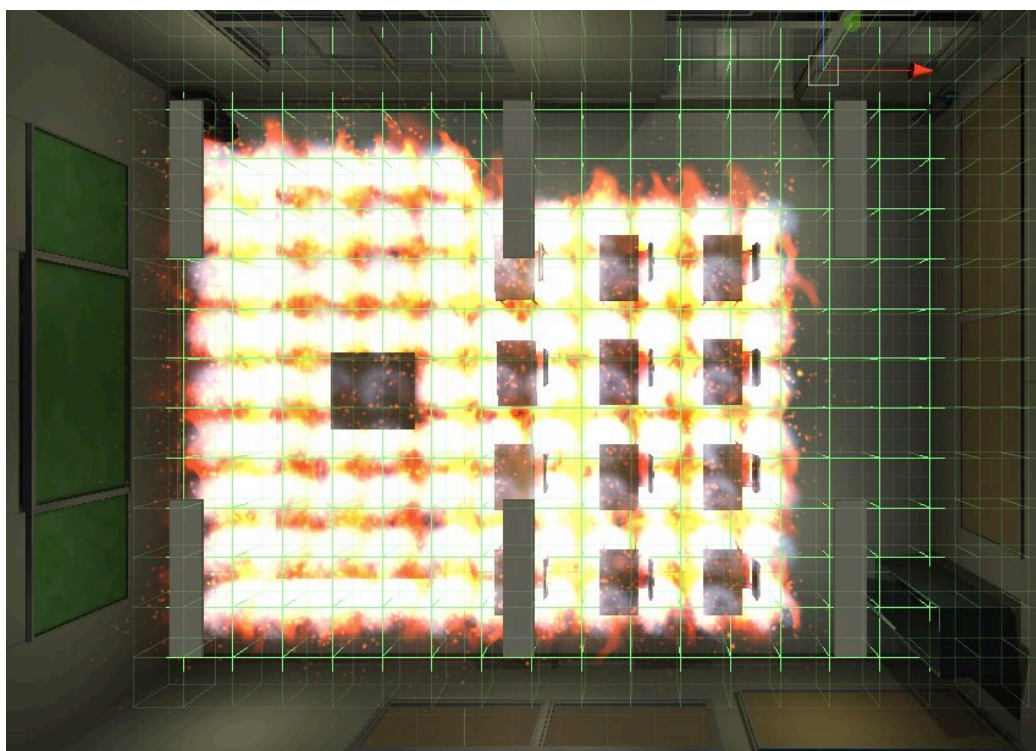
Όπως φαίνεται και στην εικόνα 5-16, η φωτιά αναπτύχθηκε και εξαπλώθηκε στα γειτονικά εύφλεκτα αντικείμενα. Αν μετακινούσαμε το βάθρο και την έδρα του καθηγητή θα είχαμε ανάλογα αποτελέσματα (εικόνα 5-18)



Εικόνα 5-16: Τελική εξάπλωση της φωτιάς στα αντικείμενα



Εικόνα 5-17: Πλαίσιο στο οποίο αναμένεται να εξαπλωθεί η φωτιά

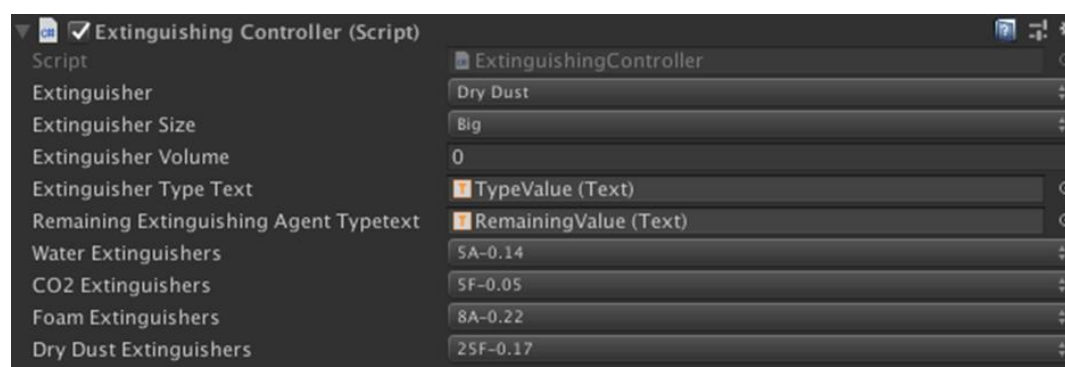


Εικόνα 5-18: Εξάπλωση της φωτιάς στα αναμενόμενα αντικείμενα

5.8 Πυρόσβεση

5.8.1 Παράμετροι Πυρόσβεσης

Κατά τη διάρκεια μελέτης του φαινομένου, ήταν απαραίτητος ο προσδιορισμός συγκεκριμένων πτυχών πυρόσβεσης και η ενσωμάτωσή τους στο σύστημά μας για να επιτύχουμε ένα ακόμα πιο ρεαλιστικό αποτέλεσμα. Συγκεκριμένα, προσδιορίστηκαν οι τύποι κατηγοριών φωτιάς και πυροσβεστήρων. Με άλλα λόγια, ο χρήστης πρέπει να επιλέξει τον τύπο, το μέγεθος και τον τύπο του πυροσβεστήρα (Εικόνα 5-19). Ανάλογα με τον τύπο της φωτιάς, κρίνεται αν ένας πυροσβεστήρας είναι κατάλληλος για την κατάσβεση της φωτιάς—δηλαδή, δεν αντιστοιχούν όλες οι κατηγορίες πυροσβεστήρων σε όλους τους τύπους φωτιάς.



Εικόνα 5-19: Παράμετροι πυρόσβεσης

Η ταξινόμηση της φωτιάς ορίζεται βάσει του ευρωπαϊκού προτύπου κατηγοριοποίησης φωτιάς EN2 [55, 56] και τις κατηγορίες A, B, Γ, Δ, όπου κάθε κατηγορία αντιστοιχεί στην πηγή κάθε φωτιάς. Για παράδειγμα εάν η φωτιά αναπτύσσεται σε ένα υλικό που είναι κατασκευασμένο από ξύλο, τότε η κατηγορία φωτιάς είναι A (βλ. Ενότητα 5.8.2 για περισσότερες πληροφορίες).

Το άλλο βασικό κριτήριο κατηγοριοποίησης της πυρόσβεσης, είναι η ικανότητα κατάσβεσης του πυροσβεστήρα. Είναι προφανές ότι η ικανότητα κατάσβεσης δεν είναι ένα μέτρο που μπορεί εύκολα, και αντικειμενικά, να επαληθευτεί, επειδή η κατασβεστική ικανότητα του πυροσβεστήρα είναι αποτέλεσμα δοκιμαστικών μετρήσεων.

Η ικανότητα κατάσβεσης καθορίζεται από τον κατασκευαστή που είναι υποχρεωμένος να το επαληθεύσει με δοκιμές που πραγματοποιούνται σε πιστοποιημένα εργαστήρια δοκιμών [55]. Συγκεκριμένα, η ικανότητα κατάσβεσης του πυροσβεστήρα αναφέρεται στο μήκος της τυπικής φωτιάς που κατασβέστηκε κατά τη διάρκεια των δοκιμών και τον τύπο της πυρκαγιάς που κατευθύνεται. Για

παράδειγμα, ο υποτύπος πυροσβεστήρα 21Α, αναφέρεται σε πυροσβεστήρα που δοκιμάστηκε σε πυρκαγιά τύπου Α (κοινά υλικά) του οποίου η πυρκαγιά ήταν 2.1 m (μήκος). Αυτός ο πυροσβεστήρας έχει ικανότητα πυρόσβεσης 0.59 —δηλαδή ανά δευτερόλεπτο, ο πυροσβεστήρας μπορεί να σβήσει 0.59 m³ τυπικού όγκου φωτιάς.

5.8.2 Κατηγορίες φωτιάς

Ανάλογα με τον τύπο καυσίμου/ εύφλεκτου υλικού, υπάρχουν τέσσερις κύριοι τύποι πυρκαγιάς, όπως φαίνεται στον Πίνακα 5-1. Ανάλογα με τον τύπο καυσίμου, αναγράφονται και τα κατάλληλα πυροσβεστικά υλικά —που είναι ικανά να σβήσουν τον εκάστοτε τύπο φωτιάς.

Κατηγορίες Φωτιάς	Κάυσιμο υλικό	Κατάλληλος τύπος Πυροσβεστήρα/ Κατασβεστικό υλικό
A	Κοινά καύσιμα υλικά (ξύλο, χαρτί, ύφασμα, διάφορα πλαστικά κ.λπ.)	Νερό, αφρός, ξηρή σκόνη
B/F	Εύφλεκτα υγρά, υγρά καύσιμα (πετρέλαιο, άλλα εύφλεκτα υγρά κ.λπ.)	Διοξείδιο του άνθρακα (CO ₂), ξηρή σκόνη, αφρός, νερό
C	Εύφλεκτα υγρά, υγρά καύσιμα (πετρέλαιο, άλλα εύφλεκτα υγρά κ.λπ.)	CO ₂ , Ξηρή σκόνη
D	Εύφλεκτα υγρά, υγρά καύσιμα (πετρέλαιο, άλλα εύφλεκτα υγρά κ.λπ.)	Ξηρή σκόνη

Πίνακας 5-1: Κατηγορίες φωτιάς

Με βάση τις κατηγορίες πυροσβεστήρων που αναλύονται παρακάτω στην Ενότητα 5.8.3 μπορούν να επισημανθούν τα εξής:

- Οι πυροσβεστήρες νερού δεν συνιστώνται για χρήση σε πυρκαγιές που περιλαμβάνουν ηλεκτρικό εξοπλισμό, πυρκαγιές σε κουζίνα, εύφλεκτα αέρια και υγρά.
- Πυροσβεστήρες αφρού, αντίστοιχα, δε συνιστώνται σε πυρκαγιές κουζίνας, πυρκαγιές με ηλεκτρικό εξοπλισμό, ή εύφλεκτα μέταλλα.
- Οι πυροσβεστήρες ξηράς σκόνης δεν συνιστώνται για χρήση σε πυρκαγιές που περιλαμβάνουν μαγειρικό λάδι, πυρκαγιές με ηλεκτρικό εξοπλισμό άνω των 1000V, ή σε κλειστούς χώρους όπως γραφεία ή σπίτια.

5.8.3 Κατηγορίες πυροσβεστήρων

Ανάλογα με το περιεχόμενό τους, δηλαδή το κατασβεστικό υλικό που περιέχουν, οι πυροσβεστήρες κατηγοριοποιούνται σε πυροσβεστήρες νερού, πυροσβεστήρες ξηράς σκόνης, πυροσβεστήρες διοξειδίου του άνθρακα (CO₂), πυροσβεστήρες μηχανικού αφρού κ.λπ.

Στο συγκεκριμένο μοντέλο λαμβάνουμε υπόψη τις ακόλουθες κατηγορίες πυροσβεστήρων:

Πυροσβεστήρας νερού

Η ικανότητα πυρόσβεσης βασίζεται στην ικανότητά του να απορροφά σημαντικές ποσότητες θερμότητας από τη φωτιά, οπότε εξατμίζεται. Ταυτόχρονα, μονώνει θερμικά ολόκληρη την περιοχή της εστίας. Το νερό που εκτοξεύεται πρέπει να έρθει σε επαφή με αντικείμενα στις περιοχές καύσης.

• Πυροσβεστήρας διοξειδίου του άνθρακα (CO₂)

Η κατάσβεσή του οφείλεται στην αραίωση του αέρα καύσης και του οξυγόνου. Χρησιμοποιείται επιτυχώς ως αδρανές, μη ηλεκτρικά αγώγιμο και «καθαρό» πυροσβεστικό υλικό επειδή δεν αφήνει υπολείμματα μετά τη χρήση. Είναι κατάλληλο για περιοχές που περιέχουν υγρά ή αέρια καύσιμα, περιοχές ηλεκτρικών συσκευών και ηλεκτρικών εγκαταστάσεων, συνηθισμένα καύσιμα (χαρτί, ξύλο κ.λπ.), στερεά καύσιμα κλπ. Η χρήση του πυροσβεστήρα διοξειδίου του άνθρακα (CO₂) επιτρέπεται σε υπολογιστές και τηλεπικοινωνίες για να αποφευχθεί η άμεση πυροδότηση σε ηλεκτρονικό εξοπλισμό που ενδεχομένως να είναι σε λειτουργία.

• Πυροσβεστήρας ξηράς σκόνης

Η ικανότητά κατάσβεσης τους βασίζεται στην ικανότητα επέμβασης στην αλυσίδα της φωτιάς. Οι πιο κοινές ξηρές χημικές σκόνες έχουν όξινο ανθρακικό κάλιο ή νάτριο ως βασικό υλικό τους. Θεωρητικά μπορούν να σβήσουν οποιαδήποτε φωτιά. Είναι κατάλληλοι για κατάσβεση σε περιβάλλον υψηλής τάσης, αν και έχουν το μειονέκτημα ότι τα υπολείμματα σκόνης μπορούν να προκαλέσουν μεγάλη ζημιά στις εγκαταστάσεις και ύστερα από την χρήση τους εκπέμπουν επικίνδυνους ρύπους. Γενικά, όπου δεν υπάρχει κίνδυνος υπολειπόμενης βλάβης, οι

πυροσβεστήρες ξηράς σκόνης είναι μια εξαιρετική λύση.

● Πυροσβεστήρας αφρού

Κατάλληλο για χρήση σε συμπαγείς εύφλεκτες πυρκαγιές κατηγορίας Α και εύφλεκτα υγρά πυρκαγιάς κατηγορίας Β. Οι πυροσβεστήρες αφρού δημιουργούν εφέ ψύξης, απαλύνουν τα καύσιμα υλικά, και αποτρέπουν τυχόν εκρήξεις κατά τη διάρκεια φωτιάς. Είναι πιο κατάλληλα για την κατάσβεση υγρών πυρκαγιών με εύφλεκτο υγρό όπως η βενζίνη ή το diesel και είναι πιο ευέλικτοι από τους υπόλοιπους πυροσβεστήρες, επειδή μπορούν επίσης να χρησιμοποιηθούν σε στερεά όπως ξύλο και χαρτί.

5.8.4 Χρόνος εκκένωσης Πυροσβεστήρων

Μέγεθος Πυροσβεστήρα

Επειδή η προσομοίωση του συγκεκριμένου έργου αναφέρεται σε ένα δοκιμαστικό δωμάτιο, δεν μπορούμε να υπερβούμε τα 12 κιλά πυροσβεστήρα (υπάρχουν και μεγαλύτεροι σε μέγεθος πυροσβεστήρες αλλά χρησιμοποιούνται σε βιομηχανίες ή παρόμοιες εγκαταστάσεις). Στις ακόλουθες τιμές εννοούνται τα κιλά του υλικού πυρόσβεσης (νερό, αφρός κ.λπ.) του πυροσβεστήρα και όχι το συνολικό βάρος του πυροσβεστήρα (Πίνακας 2). Μονάδα μέτρησης Kg, εκτός από τον πυροσβεστήρα νερού που είναι Lt. Επομένως στο project ορίζουμε ως επιλογές

πυροσβεστήρα τα μεγέθη 6kg και 9kg με τους αντίστοιχους χρόνους εκκένωσης.

Κατηγορία Φωτιάς	Μέγεθος Πυροσβεστήρα, (κατασβεστικού υλικού, Kg)	Συνολικός Χρόνος εκκένωσης πυροσβεστήρα (sec)	Απώλεια υλικού/ δευτερόλεπτο
A	3	-	-
B (Medium)	6	30	0.2 kg/sec
C (Big)	9	40	0.22 kg/sec
D	12	-	-

Πίνακας 5-2: Κατηγορίες φωτιάς

(Ο χρόνος εκκένωσης ορίζεται ως ο χρόνος (δευτερόλεπτα) που χρειάζεται ο πυροσβεστήρας για να αδειάσει εντελώς το κατασβεστικό υλικό που περιέχει. Είναι γραμμένο στην ετικέτα που φέρει από τον κατασκευαστή. (Τυπικές τιμές EN3) [57].

Η κύρια πτυχή που λαμβάνεται υπόψη είναι η ικανότητα πυρόσβεσης του πυροσβεστήρα. Με αυτό εννοούμε την απόδοση ενός πυροσβεστήρα κατά την κατάσβεση όπου αναγράφεται στην ετικέτα του πυροσβεστήρα και καθορίζεται από τον κατασκευαστή. Με βάση τις κατηγορίες της Πυροσβεστικής Ικανότητας (πόση επιφάνεια του αντικειμένου καύσης μπορεί να σβήσει ο πυροσβεστήρας ανά δευτερόλεπτο) καθορίζονται τυπικές τιμές (EN3) για το ποσοστό φόρτισης του πυροσβεστήρα.

5.8.5 Χωρητικότητα κατάσβεσης Πυροσβεστήρων

Στην συγκεκριμένη ενότητα αναφέρουμε λεπτομερώς την ικανότητα πυρόσβεσης των πυροσβεστήρων ανά κατηγορία πυρκαγιάς.

· **Κατηγορία φωτιάς A (Κοινά καύσιμα υλικά : ξύλο, χαρτί, ύφασμα, πλαστικό, κλπ.)** Τυπικό μήκος πυρκαγιάς L (Μονάδα μέτρησης m), Όγκος τυπικής

πυρκαγιάς A (μονάδα μέτρησης m³).

Κατηγορία πυροσβεστήρα	Όγκος τυπικής φωτιάς (m ³ /sec)
5A	0.14 (Πυροσβεστήρας νερού)
8A	0.22 (Πυροσβεστήρας αφρού)
13A	0.36 (Πυροσβεστήρας νερού)
21A	0.59 (Πυροσβεστήρας ξηράς σκόνης)
27A	0.77 (Πυροσβεστήρας ξηράς σκόνης)
34A	0.95 (Πυροσβεστήρας Αφρού)

Σημείωση:

- Ο πυροσβεστήρας με ένδειξη **21A** έχει τη δυνατότητα να σβήσει 0,59 m³ τυπικού όγκου πυρκαγιάς ανά δευτερόλεπτο σε ένα αντικείμενο καύσης .
- Η ένδειξη **21A** αναφέρεται σε πυροσβεστήρα που δοκιμάστηκε σε πυρκαγιά τύπου A (κοινά υλικά) του οποίου η πυρκαγιά ήταν 2,1 m (μήκος)

Κατηγορία φωτιάς B (Εύφλεκτα υγρά, υγρά καύσιμα (πετρέλαιο, άλλα εύφλεκτα υγρά κ.λπ.)

Κατηγορία πυροσβεστήρα	Όγκος τυπικής φωτιάς (m ³ /sec)
21B	0.66 (Πυροσβεστήρας αφρού)
34B	1.07 (Πυροσβεστήρας ξηράς σκόνης)
55B	1.73 (CO ₂)
70B	2.20 (Πυροσβεστήρας Αφρού)
89B	2.80 (CO ₂)
113B	3.55 (Πυροσβεστήρας ξηράς σκόνης)

Κατηγορία φωτιάς F (Μαγειρικά σκεύη, Λάδια μαγειρέματος)

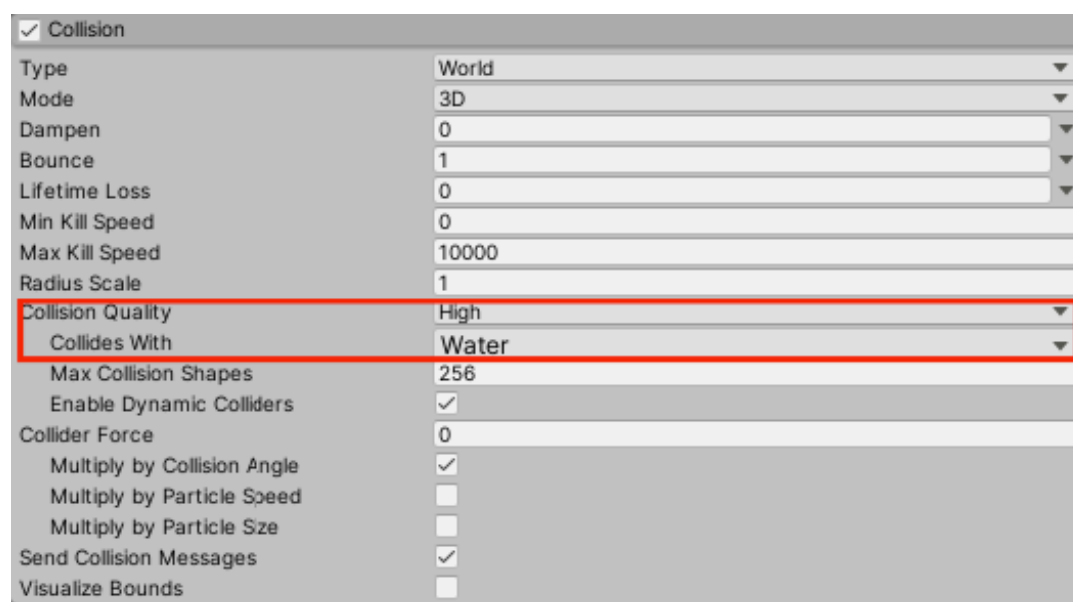
Κατηγορία πυροσβεστήρα	Όγκος τυπικής φωτιάς (m ³ /sec)
5F	0.05 (CO ₂)
25F	0.17 (Πυροσβεστήρας ξηράς σκόνης)

5.8.6 Σύστημα πόντων

Για ένα ολοκληρωμένο σύστημα πυρασφάλειας, θεωρήθηκε απαραίτητο να αναπτυχθεί και λειτουργία πυρόσβεσης. Το σύστημα πυρόσβεσης είναι το βασικό εκπαιδευτικό κομμάτι της προσομοίωσης, καθώς, σε αυτό το σημείο, ο εκπαιδευόμενος καλείται να επιλέξει τον κατάλληλο πυροσβεστήρα για να σβήσει το υλικό που φλέγεται και βάσει της επιλογής αυτής, αυξάνονται, ή μειώνονται οι πόντοι του.

Κάθε κλώνος φωτιάς, καπνού, όπως επίσης και τα σωματίδια του κατασβεστικού υλικού αποτελούνται από Particles. Χρησιμοποιώντας την μέθοδο της Unity, OnParticleCollision, η οποία ανιχνεύει όλες τις συγκρούσεις μεταξύ των σωματιδίων, μπορούμε να εντοπίσουμε πότε το κατασβεστικό υλικό έρχεται σε επαφή με τη φωτιά. Στην παραμετροποίηση του Particle system, για το κάθε αντικείμενο, θέσαμε τη φωτιά να μπορεί να συγκρουστεί (collide) μόνο με κατασβεστικό υλικό (water —εικόνα 5-20), ενώ στο κατασβεστικό υλικό, θέσαμε να συγκρούεται με οτιδήποτε στη σκηνή (everything). Όπως περιεγράφηκε και

παραπάνω, κάθε υλικό έχει και τις δικές του ιδιότητες, τα οποία αντιστοιχούνται με μια κατηγορία φωτιάς.



Εικόνα 5-20: Ρύθμιση του *particle collision*

Στον αλγόριθμό μας έχουμε τις κατηγορίες A, B, F. Κατά την διάρκεια της πρώτης συλλογής των δεδομένων, για κάθε κελί, καταχωρείται και η κατηγορία της φωτιάς, σχετικά με το έπιπλο που περιέχει. Επομένως, ξέρουμε, για παράδειγμα, αν στο κελί 1 υπάρχει το έπιπλο με την κατηγορία φωτιάς A, τότε αυτό το έπιπλο μπορεί να κατασβεστεί με όλους τους διαθέσιμους πυροσβεστήρες, εκτός του πυροσβεστήρα διοξειδίου του άνθρακα (CO₂). Αν επιλεγεί αυτός ο πυροσβεστήρας, για το συγκεκριμένο έπιπλο —όσο χρησιμοποιείται—ο εκπαιδευόμενος χάνει πόντους.

5.8.7 Αλγόριθμοι πυρόσβεσης

Η επιλογή πυροσβεστήρα από τον χρήστη και η αλλαγή στο σύστημα φαίνεται στον αλγόριθμο 13. Χρησιμοποιείται μια switch case για έλεγχο της επιλογής πυροσβεστήρα και γίνεται σύγκριση με την κατηγορία κλάσης. Για παράδειγμα —όπως φαίνεται και στον αλγόριθμο 13— αν σε ένα αντικείμενο υπάρχει η κατηγορία φωτιάς A, τότε αυτή μπορεί να κατασβεστεί με όλους τους διαθέσιμους πυροσβεστήρες, εκτός του διοξειδίου του άνθρακα (CO₂). Σε περίπτωση που επιλεγεί ο πυροσβεστήρας διοξειδίου, το score μειώνεται κατά δέκα πόντους. Παράλληλα, αν για παράδειγμα, επιλεγεί ο πυροσβεστήρας νερού 5A (0.14), τότε η μεταβλητή *extinguishingFactor* γίνεται 0.14 (Αλγόριθμος 11). Αυτό

σημαίνει, πως για κάθε δευτερόλεπτο πρέπει να αφαιρείται από το συνολικό μέγεθος της φωτιάς 0.14. Το μέγεθος της φωτιάς καθορίζεται από το μέγεθος των Particles, από τα οποία αποτελείται η φωτιά, και καθορίζεται από την ιδιότητα startSize. Επομένως, ο αλγόριθμος πρέπει να μειώνει την τιμή της startSize κατά 0.14 ανά δευτερόλεπτο. Για να επιτευχθεί αυτό, καλείται η μέθοδος ReduceFireParticles, η οποία είναι υπεύθυνη για την εκτέλεση της μεθόδου της απομείωσης της φωτιάς (ReduceParticlesStartSize), ανά δευτερόλεπτο (Αλγόριθμος 12).

```
if (_dictCtrl.fireCloneClasses.ContainsKey(obj.name) &&
    _dictCtrl.fireCloneClasses[obj.name].Contains("A") &&
    extinguisher != Extinguisher.CO2) {
    switch (extinguisher) {
        case Extinguisher.Water:
            WaterSelected();
            break;
        case Extinguisher.Foam:
            FoamSelected();
            break;
        case Extinguisher.DryDust:
            DryDustSelected();
            break;
        case Extinguisher.CO2:
            // TODO: Show Alert
            break;
        default:
            throw new ArgumentOutOfRangeException();
    }
}

CallReduceParticlesSizeWithDelay();
}
```

Αλγόριθμος 5-18: Έλεγχος επιλογής πυροσβεστήρα για την κατηγορία φωτιάς

```
private void WaterSelected() {
    switch (waterExtinguishersIdx) {
        case 0:
            extinguishingFactor = 0.14f;
            break;
        case 1:
            extinguishingFactor = 0.36f;
            break;
    }
}
```

Αλγόριθμος 5-19: Μέθοδος WaterSelected

```
private void CallReduceParticlesSizeWithDelay() {
    fireTimer += Time.deltaTime;
    if (!(fireTimer > waitTime)) return;
    ReduceParticlesStartSize();
    fireTimer -= waitTime;
    Time.timeScale = scrollBar;
}
```

Αλγόριθμος 5-20: Μέθοδος CallReduceParticlesSizeWithDelay

```
private void ReduceParticlesStartSize() {  
    tempFireClone.startSize -= extinguishingFactor;  
}
```

Αλγόριθμος 5-21: Μέθοδος CallReduceParticlesSizeWithDelay

6 Συμπέρασμα/ Μελλοντική εργασία

Η προσομοίωση που αναπτύχθηκε, πρόκειται να δοκιμαστεί και να αξιολογηθεί από καθηγητές και μαθητές σχολείων, ώστε να μάθουμε από την εμπειρία χρήσης των εκπαιδευόμενων σημεία που θα μπορούσαν να βελτιωθούν. Κάποια σημεία που έχουν εντοπιστεί, από μέρους μας, προς βελτίωση είναι η μοντελοποίηση της φωτιάς, ώστε να είναι πιο ρεαλιστική, μειώνοντας το μέγεθος των κελιών του πλέγματος. Ωστόσο, σε αυτό το σημείο πρέπει να βρεθεί η χρυσή τομή μεταξύ ακρίβειας, και επίδοσης, καθώς όσο μεγαλώνει ο αριθμός των κελιών—τα οποία είναι τρισδιάστατα αντικείμενα—τόσο οι απαιτήσεις της προσομοίωσης μεγαλώνουν.

Επιπρόσθετα, μπορούν να αναπτυχθούν περισσότερες σκηνές οι οποίες να προσομοιώνουν περισσότερους χώρους, καθώς ο αλγόριθμος μπορεί να προσαρμοστεί σε κάθε σκηνή, σχετικά εύκολα. Επομένως, απομένει μόνο το σχεδιαστικό κομμάτι. Το σύστημα πόντων θα μπορούσε να προσομοιώνει περισσότερο σε αυτό των εμπορικών παιχνιδιών, όπου ο χρήστης μπορεί να βλέπει τα scores των προηγούμενων χρηστών (high scores), ωθώντας τους συμμετέχοντες στην προσπάθεια για καλύτερες επιδόσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Brooks, J., Frederick. (1999). What's Real About Virtual Reality? *IEEE Computer Graphics and Applications*, 19, 16–27. <https://doi.org/10.1109/38.799723>
2. Extended reality. (2022). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από https://en.wikipedia.org/w/index.php?title=Extended_reality&oldid=1072372214
3. LaValle M. Steven. (2020). Virtual Reality [ηλεκτρονικό βιβλίο]. Ανακτήθηκε 22 Φεβρουαρίου 2022 από <http://vr.cs.uiuc.edu/web.html>
4. Neuf.TV. (2021, August 10). Buyer's guide to the best VR headsets. Ανακτήθηκε 22 Φεβρουαρίου 2022 από <https://www.neuf.tv/en/buyer%27s-guide-to-the-best-vr-htc-vive-focus-3-oculus-quest-2-and-more-headsets>
5. M. A., L., & B. A., L. (n.d.). Understanding Plato's Allegory of the Cave. ThoughtCo. Ανακτήθηκε 22 Φεβρουαρίου 2022 από <https://www.thoughtco.com/the-allegory-of-the-cave-120330>
6. The Sociable. Will You Dare to Leave Plato's Cave? Leading Minds Weigh-in on Simulation Theories. (2017, August 14). Ανακτήθηκε 22 Φεβρουαρίου 2022 από <https://sociable.co/technology/plato-cave-simulation-theory/>
7. Evil demon. (2021). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από https://en.wikipedia.org/w/index.php?title=Evil_demon&oldid=1045959525
8. Brain in a vat. (2021). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από https://en.wikipedia.org/w/index.php?title=Brain_in_a_vat&oldid=1045392062
9. Vox. Roberts, D. (2016, June 3). Elon Musk thinks we all live in a video game. So what if we do? Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.vox.com/2016/6/3/11837888/simulation-problem>
10. VPL Research. (2021). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από https://en.wikipedia.org/w/index.php?title=VPL_Research&oldid=1020049427
11. Sorene, P. (2014, November 24). Jaron Lanier's EyePhone: Head And Glove Virtual Reality In The 1980s. Flashbak. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://flashbak.com/jaron-laniers-eyephone-head-and-glove-virtual-reality-in-the-1980s-26180/>
12. CNET. The best VR games. (n.d.). CNET. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.cnet.com/pictures/best-vr-games/>
13. The Verge. Robertson, A. (2014, November 20). Watch Paul McCartney perform in virtual reality with a new Android app. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.theverge.com/2014/11/20/7253617/watch-paul-mccartney-perform-in-virtual-reality-with-a-new-android-app>
14. MIT - Docubase. Clouds Over Sidra. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://docubase.mit.edu/project/clouds-over-sidra/>
15. The Verge —The UN wants to see how far VR empathy will go (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.theverge.com/2016/9/19/12933874/unvr-clouds-over-sidra-film-app-launch>
16. Virtual Reality (VR) in Automotive Industry. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://tecknotrove.com/virtual-reality-vr-in-automotive-industry/>
17. IEC E-Tech. Price, A., & May 2019, 15. (2019, May 15). Virtual reality makes real life even better. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://etech.iec.ch/issue/2019-03/virtual-reality-makes-real-life-even-better>
18. Augmented reality. (2021). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από https://en.wikipedia.org/w/index.php?title=Augmented_reality&oldid=1062792894
19. Coca-Cola HBC. (n.d.). TeamViewer. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.teamviewer.com/en/success-stories/coca-cola-hbc/>
20. Bol, M. & . (2020, September 10). Mobile AR Users Approach 600 Million. AR Insider. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://arinsider.co/2020/09/10/mobile-ar-users-approach-600-million/>
21. Warehousing and Logistics Lead the AR Transformation Adoption Race. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://upskill.io/resources/blog/warehousing-logistics-ar-transformation/>
22. TeamViewer Support . How Augmented Reality Can Benefit Your Company. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://community.teamviewer.com/English/discussion/35870/how-augmented-reality-can-benefit-your-company>
23. Virtual fitting room: The new trend - AR and VR Development Blog. (2018, March 5). Augmented and Virtual Reality Company. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://ardev.es/en/virtual-fitting-room-augmented-reality/>
24. How to Use Augmented Reality for Indoor Navigation in 2021. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://mobidev.biz/blog/augmented-reality-indoor-navigation-app-developement-arkit>
25. Joshi, N. (n.d.). 8 Future Mixed Reality Applications To Watch Out For. Forbes. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.forbes.com/sites/cognitiveworld/2019/11/03/8-future-mixed-reality-applications-to-watch-out-for/>
26. Pinheads Interactive. 7 Best VR Educational Apps For Learning In. (n.d) Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.pinheadsinteractive.com/best-vr-educational-apps-2020/>
27. Moreno, A., Posada, J., Segura, Á., Arbelaz, A., & García-Alonso, A. (2014). Interactive fire spread simulations with extinguishment support for virtual reality training tools. *Fire Safety Journal*, 64, 48–60. <https://doi.org/10.1016/j.firesaf.2014.01.005>

28. Zhao, S. (2010). GisFFE—an integrated software system for the dynamic simulation of fires following an earthquake based on GIS. *Fire Safety Journal - FIRE SAFETY J*, 45, 83–97. <https://doi.org/10.1016/j.firesaf.2009.11.001>
29. Fathima S J, S., & Aroma, J. (2019). Simulation of fire safety training environment using immersive virtual reality. *International Journal of Recent Technology and Engineering*, 7, 347–350.
30. DeChamplain, A., Rosendale, E., McCabe, I., Stephan, M., Cole, V., & Kapralos, B. (2012). Blaze: A serious game for improving household fire safety awareness. 2012 IEEE International Games Innovation Conference, 1–4. <https://doi.org/10.1109/IGIC.2012.6329839>
31. Bettina Schlager and Anton Fuhrmann. 2017, in press. Building a Virtual Reality Fire Training with Unity and HTC Vive. In *CEISG 2017: The 21st Central European Seminar on Computer Graphics*
32. Unity (game engine). (2022). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από [https://en.wikipedia.org/w/index.php?title=Unity_\(game_engine\)&oldid=1073786854](https://en.wikipedia.org/w/index.php?title=Unity_(game_engine)&oldid=1073786854)
33. Hocking, J. (2015). Unity in action: Multiplatform game development in C#. Manning.
34. Wallace, M. (n.d.). It Turns Out That “Cuphead” Is Really Difficult. *Forbes*. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.forbes.com/sites/mitchwallace/2017/09/29/it-turns-out-that-cuphead-is-really-difficult/>
35. Pillars of Eternity—Definitive Edition | Download and Buy Today—Epic Games Store. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://www.epicgames.com/store/en-US/p/pillars-of-eternity>
36. Ori and the Blind Forest on Steam. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από https://store.steampowered.com/app/261570/Ori_and_the_Blind_Forest/
37. BillWagner. (n.d.). A Tour of C#—C# Guide. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
38. Technologies, U. (n.d.). Unity - Manual: The Hierarchy window. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://docs.unity3d.com/Manual/Hierarchy.html>
39. Technologies, U. (n.d.). Unity - Manual: Colliders. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://docs.unity3d.com/Manual/CollidersOverview.html>
40. Getting Started With Unity: Colliders & UnityScript. (n.d.). Code Envato Tuts+. Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <https://code.tutsplus.com/tutorials/getting-started-with-unity-colliders-unityscript--active-8367>
41. Working with Unity Cameras. (n.d.). Unity Learn. Ανακτήθηκε 22 Φεβρουαρίου, 2022 <https://learn.unity.com/tutorial/working-with-unity-cameras-questions>
42. Technologies, U. (n.d.). Unity - Manual: Cameras. Ανακτήθηκε 22 Φεβρουαρίου, 2022 <https://docs.unity3d.com/Manual/CamerasOverview.html>
43. Technologies, U. (n.d.). Unity - Manual: Types of light. Ανακτήθηκε 22 Φεβρουαρίου, 2022 <https://docs.unity3d.com/Manual/Lighting.html>
44. Technologies, U. (n.d.). Unity - Manual: Lighting. Ανακτήθηκε 22 Φεβρουαρίου, 2022 <https://docs.unity3d.com/Manual/LightingOverview.html>
45. Pastor, E., Zarate, L., Planas, E., & Arnaldos, J. (2003). Mathematical models and calculation systems for the study of wildland fire behaviour. *Progress in Energy and Combustion Science*, 29, 139–153. [https://doi.org/10.1016/S0360-1285\(03\)00017-0](https://doi.org/10.1016/S0360-1285(03)00017-0)
46. Tate, D. L., Sibert, L., & King, T. (1997). Virtual environments for shipboard firefighting training. In *IEEE VR Conf* (pp. 68, 215). <https://doi.org/10.1109/VR AIS.1997.583045>
47. Sidh – a Game Based Firefighter Training Simulation | IEEE Conference Publication | IEEE Xplore. (n.d.). Retrieved March 6, 2022, from <https://ieeexplore.ieee.org/document/4272085>
48. Chittaro, L., & Ranon, R. (2009). Serious Games for Training Occupants of a Building in Personal Fire Safety Skills. 2009 Conference in Games and Virtual Worlds for Serious Applications, 76–83. <https://doi.org/10.1109/VS-GAMES.2009.8>
49. Using immersive game-based virtual reality to teach fire-safety skills to children | SpringerLink. (n.d.). Retrieved March 6, 2022, from <https://link.springer.com/article/10.1007/s10055-009-0113-6>
50. Bukowski, R., & Séquin, C. (1997). Interactive simulation of fire in virtual building environments. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 35–44. <https://doi.org/10.1145/258734.258757>
51. St. Julien, T. U., & Shaw, C. D. (2003). Firefighter command training virtual environment. *Proceedings of the 2003 Conference on Diversity in Computing*, 30–33. <https://doi.org/10.1145/948542.948549>
52. Quintiere, J. G. (1998). Book Review: Principles of Fire Behavior. *Journal of Fire Protection Engineering*, 9(1), 58–60. <https://doi.org/10.1177/104239159800900104>
53. McGrattan, K. B., McDermott, R. J., Weinschenk, C. G., & Forney, G. P. (2013). Fire Dynamics Simulator, Technical Reference Guide, Sixth Edition. <https://www.nist.gov/publications/fire-dynamics-simulator-technical-reference-guide-sixth-edition>
54. McGrattan, K. B., McDermott, R. J., Weinschenk, C. G., & Forney, G. P. (2013). Fire Dynamics Simulator, Technical Reference Guide, Sixth Edition. <https://www.nist.gov/publications/fire-dynamics-simulator-technical-reference-guide-sixth-edition>
55. Fire Extinguishers Handbook. LIBRARY. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από, <https://pyroprostasia.org/index.php/en/library/standards/itemlist/category/7-library.html>
56. "European Fire Standards and National Legislation (Fire Categories.n.d), Ανακτήθηκε 22 Φεβρουαρίου, 2022 από <http://highperformanceinsulation.eu>.
57. Πυροσβεστικά Είδη—Ιωάννης Νικ. Μαυρουδής—Πειραιάς. (n.d.). Ανακτήθηκε 22 Φεβρουαρίου, 2022 από, from <https://pyrellas.gr/>

58. What is Virtual Reality? VR Definition and Examples | Marxent. (n.d.). Retrieved March 6, 2022, from <https://www.marxentlabs.com/what-is-virtual-reality/>
59. Brooks, F. P. (1999). What's real about virtual reality? IEEE Computer Graphics and Applications, 19(6), 16–27. <https://doi.org/10.1109/38.799723>
60. Kamińska, D., Sapiński, T., Wiak, S., Tikk, T., Haamer, R. E., Avots, E., Helmi, A., Ozcinar, C., & Anbarjafari, G. (2019). Virtual Reality and Its Applications in Education: Survey. Information, 10(10), 318. <https://doi.org/10.3390/info10100318>
61. Tzani, M., Besharat, J., Charalampous, V., & Stylios, C. (2020). Building a Virtual Reality Fire Environment based on Fire Dynamic Simulator. 2020 International Conference on Information Technologies (InfoTech), 1–5. <https://doi.org/10.1109/InfoTech49733.2020.9211048>
62. Yuen, K., Choi, S., & Yang, X. (2013). A Full-immersive CAVE-based VR Simulation System of Forklift Truck Operations for Safety Training. Computer-Aided Design and Applications, 7, 235–245. <https://doi.org/10.3722/cadaps.2010.235-245>
63. Cha, M., Han, S., Lee, J., & Choi, B. (2012). A virtual reality based fire training simulator integrated with fire dynamics data. Fire Safety Journal, 50, 12–24. <https://doi.org/10.1016/j.firesaf.2012.01.004>
64. Using immersive game-based virtual reality to teach fire-safety skills to children | SpringerLink. (n.d.). Retrieved March 6, 2022, from <https://link.springer.com/article/10.1007/s10055-009-0113-6>
65. Xu, Z., Lu, X. Z., Guan, H., Chen, C., & Ren, A. Z. (2014). A virtual reality based fire training simulator with smoke hazard assessment capacity. Advances in Engineering Software, 68, 1–8. <https://doi.org/10.1016/j.advengsoft.2013.10.004>
66. Smith, S., & Ericson, E. (2009). Using immersive game-based virtual reality to teach fire-safety skills to children. Virtual Reality, 13, 87–99. <https://doi.org/10.1007/s10055-009-0113-6>
67. Charalampous, V., Besharat, J., & Stylios, C. (2020). Designing and Developing a VR Environment for Indoor Fire Simulation. 24th Pan-Hellenic Conference on Informatics, 237–240. <https://doi.org/10.1145/3437120.3437315>
68. J. Besharat, M. Tznani, V. Charalampous, G. Papantzikos and C. Stylios. Develop a Virtual Reality simulation Environment for training purposes, 11th IEEE International Conference on Information, Intelligence, Systems and Applications (IISA 2021), Chania, Greece, 12-14 July (2021).