

POLITECNICO DI TORINO  
Corso di Laurea Magistrale in Ingegneria Edile



**FIRE SAFETY ENGINEERING E VIRTUAL  
REALITY: APPLICAZIONI PER LO STUDIO  
DELL'ESODO IN EMERGENZA**

**Relatore:** Prof. Roberto Vancetti

**Correlatori:** Ing. Filippo Cosi

Ing. Emiliano Cereda

**Candidato:** Rosanna Caporusso

Anno accademico 2020/2021



Conoscere il comportamento umano in emergenza e studiare la visibilità dei sistemi di segnaletica in presenza di fumo è essenziale per lo studio dell'esodo in emergenza e quindi per garantire la salvaguardia della vita umana.

Per comprendere e prevedere al meglio l'esodo, in una situazione di emergenza, è fondamentale la progettazione di un sistema di esodo efficiente che permetta agli occupanti di prendere la decisione migliore da intraprendere per raggiungere il luogo sicuro. Per questo motivo, in questo lavoro di tesi, particolare importanza è stata data al tema del wayfinding, ovvero, la capacità di comprendere come raggiungere una determinata destinazione.

Questa disciplina attualmente poco conosciuta nel nostro paese permette di studiare le modalità con cui gli occupanti si orientano nello spazio stabilendo il rapporto che si instaura tra individuo e ambiente in cui si muove quest'ultimo. Infatti, in una situazione di emergenza è necessario che la segnaletica, l'illuminazione e la cartellonistica forniscano informazioni chiare in poco tempo in maniera tale che gli occupanti possano raggiungere il luogo sicuro in sicurezza e in breve tempo.

Con gli strumenti della FSE (Fire Safety Engineering) sono state effettuate delle analisi per lo studio della simulazione di incendio, mediante l'utilizzo di software CFD, e della modellazione di esodo, attraverso l'utilizzo di software numerici.

Il lavoro di tesi propone un metodo per l'esportazione della simulazione di incendio e della simulazione di esodo in realtà virtuale e della conseguente visualizzazione di esse. In particolare, viene proposto lo sviluppo di un applicativo che permette di avere il controllo dell'incendio, degli occupanti, dei sistemi di segnaletica e degli scenari di progetto. Il metodo consente di dare un ausilio agli strumenti di ricerca legati al mondo della Fire Safety Engineering e alla virtual reality, in quanto, risulta complessa la replica della distribuzione e della composizione dei fumi reali in ambiente.

## Abstract EN

Knowing human behaviour during emergencies and studying the visibility of signage systems in presence of smoke is essential for the study of emergency exodus and therefore to ensure the protection of human life.

To understand and to predict the exodus in an emergency is essential to design an optimal exodus system that could help the occupants to take the best route to reach a safe zone. For this reason, in this thesis work, a lot of emphasis has been put on the wayfinding, which is the ability to understand how to reach a specified destination.

The wayfinding is a subject that allows us to study the ways in which the occupants orient themselves in a space by establishing a relationship between the individuals and the environment in which they move. In fact, in an emergency it is necessary that the signs and the lighting provide clear information in a short time so that the occupants can reach the safe place in a short time and safely.

The tools of the FSE (Fire Safety Engineering) were used to carry out the study of fire simulation, using CFD software, and exodus modelling, using numerical software.

This thesis work proposes a way to export the simulation of fire and the simulation of exodus in virtual reality environment and the consequent visualization of them. In particular, the development of an application is proposed- This application allows you to control the fire, the occupants, the signage systems and the project scenarios.

The illustrated method is a step forward in the field of research in the area related to Fire Safety Engineering and virtual reality, given that it helps in the replication of the distribution and composition of real smoke in an environment, which is a complex subject.

<b>Abstract IT .....</b>	<b>I</b>
<b>Abstract EN.....</b>	<b>II</b>
<b>Indice delle figure .....</b>	<b>VI</b>
<b>Indice delle tabelle .....</b>	<b>X</b>
<b>Introduzione .....</b>	<b>1</b>
<b>La prevenzione incendi .....</b>	<b>4</b>
1.1.    La prevenzione incendi in Italia .....	4
1.2.    Evoluzione della normativa per i locali di intrattenimento e di pubblico spettacolo	7
1.3.    Dal metodo prescrittivo al metodo prestazionale .....	8
1.4.    L'approccio prestazionale.....	10
1.4.1.    Prima fase – Analisi Preliminare .....	11
1.4.2.    Seconda fase – Analisi Quantitativa.....	13
1.5.    Modellazione di incendio .....	14
1.6.    Modellazione di esodo .....	15
<b>Caratteristiche dell'individuo in una situazione di emergenza</b>	<b>16</b>
2.1.    Caratteristiche del singolo individuo.....	16
2.2.    Caratteristiche collettive .....	18
2.3.    Processo del comportamento umano in emergenza.....	20
2.4.    Il concetto di panico .....	22
2.5.    Incendi in locali di pubblico spettacolo.....	23
<b>Illuminazione di emergenza .....</b>	<b>25</b>
3.1.    La segnaletica di sicurezza .....	28
3.2.    Guida alla progettazione e all'uso della segnaletica in base alla legislazione e agli standard .....	29
3.2.1.    Standard ISO e Detective CEE .....	29
3.2.2.    Legislazione e standard americani.....	30
3.2.3.    Legislazione e standard britannici .....	31
3.2.4.    Legislazione e standard italiani.....	32
3.3.    Tipologie di segnaletica .....	33
3.3.1.    Segnaletica statica .....	33

3.3.2.	Segnaletica attiva.....	35
3.3.3.	Segnaletica dinamica .....	36
3.4.	Sistemi di comunicazione alternativi .....	39
3.5.	Distanza di riconoscimento della segnaletica di sicurezza.....	42
	<b>Interazione tra occupanti e sistemi di segnaletica .....</b>	<b>43</b>
4.1.	Disposizione della segnaletica .....	43
4.2.	La diffusione delle informazioni.....	44
4.2.1.	La visibilità del segnale .....	45
4.3.	L'orientamento in caso di emergenza .....	48
	<b>L'ambiente virtuale come strumento di visualizzazione .....</b>	<b>50</b>
5.1.	Differenza tra realtà virtuale, realtà aumentata e mista .....	50
5.2.	Tecnologie disponibili .....	52
5.3.	Applicazioni .....	54
5.3.1.	Applicazione all'ingegneria della sicurezza antincendio .....	55
5.3.2.	Applicazione VR ad un caso studio .....	55
	<b>Caso studio: La centrale della Lavazza – attività eventi di pubblico spettacolo .....</b>	<b>57</b>
6.1.	Inquadramento generale.....	57
6.1.1.	Lo spazio centrale .....	58
6.1.2.	La terrazza indoor .....	59
6.1.3.	La sala conferenze .....	59
6.2.	Modellazione dell'edificio con metodologia BIM .....	60
6.3.	Modellazioni di Fire Safety Engineering.....	62
6.3.1.	Configurazioni.....	62
6.3.2.	Simulazione di incendio .....	68
6.3.3.	Simulazione di esodo .....	78
	<b>Dai modelli di Fire Safety Engineering alla Virtual Reality .....</b>	<b>89</b>
7.1.	Esportazione della simulazione di incendio .....	89
7.1.2.	Isosuperfici .....	97
7.1.3.	Slices .....	105
	<b>Visualizzazione dei risultati in Virtual Reality .....</b>	<b>111</b>
8.1.	Interfaccia grafica di Unity.....	111
8.2.	Concetti fondamentali.....	114

8.3.	Importazione dei dati .....	116
8.3.1.	Importazione del modello BIM .....	116
8.3.2.	Importazione della simulazione di incendio .....	118
8.3.3.	Importazione della simulazione di esodo .....	122
8.4.	Inserimento della segnaletica .....	125
8.5.	Creazione dell'applicativo.....	128
8.5.1.	Creazione del menù interattivo .....	128
8.5.2.	Movimento .....	134
<b>Conclusioni e sviluppi futuri.....</b>		<b>140</b>
<b>Ringraziamenti.....</b>		<b>143</b>
<b>Appendice .....</b>		<b>144</b>
A.1.	Esportazione isosuperfici.....	144
A.2.	Esportazione Slices .....	147
A.3.	ArrayTest Script per Unity .....	150
A.4.	Vertex Color Shader .....	151
A.5.	Timer .....	152
A.6.	Fly Camera.....	154
A.7.	Door Script .....	158
<b>Bibliografia e sitografia .....</b>		<b>159</b>

## Indice delle figure

Figura 1 - Metodologia utilizzata	1
Figura 2 - Metodologia BIM	3
Figura 3 - Confronto tra ASET e RSET	11
Figura 4 - Fasi dell'incendio	13
Figura 5 - Modelli a zone	14
Figura 6 - Modelli di campo	15
Figura 7 - Esempio di formazione ad arco	19
Figura 8 - Incendio The Station Nightclub	23
Figura 9 - Incendio The Kiss Nightclub	24
Figura 10 - Cartello di emergenza EXIT	30
Figura 11 - Cartello di emergenza conforme alla BS 5499	31
Figura 12 - Segnaletica statica standard	33
Figura 13 - Segnaletica statica fotoluminescente	34
Figura 14 - Segnaletica statica retroilluminata	34
Figura 15 - Esempio di segnaletica attiva	35
Figura 16 - Esempio di Active Dynamic Signage System	37
Figura 17 - Funzionamento dell'Active Dynamic Signage System	38
Figura 18 - Esempio di applicativo che utilizza la posizione dell'utente nell'edificio	39
Figura 19 - Esempio di differenziazione cromatica di una porta	39
Figura 20 - Nastro luminescente a pavimento	40
Figura 21 - Esempio di percorso Loges	41
Figura 22 - Piastre componibili per percorsi Loges	41
Figura 23 - Allarme antincendio	41
Figura 24 - Distanza di riconoscimento per la segnaletica di sicurezza secondo la norma UNI EN 1838	42
Figura 25 - Esempio di sovrabbondanza della segnaletica	44
Figura 26 - Relazione tra visibilità attraverso il fumo e la velocità di camminata	48
Figura 27 - Esempio di uscite sovraffollate e non utilizzate	49
Figura 28 - Classificazione realtà	51
Figura 29 - Oculus Rift	52
Figura 30 - HTC Vive	53
Figura 31 - Cyberith Virtualizer	53
Figura 32 - Il Cave	54
Figura 33 - Inquadramento della Centrale della Lavazza	57
Figura 34 - Progetto dell'Ingegnere Perini per la Società Alta Italia	57
Figura 35 - Lo spazio centrale	58
Figura 36 - La terrazza Indoor	59
Figura 37 - La sala conferenze	59
Figura 38 - Creazione dei dettagli costruttivi mediante il comando Estrusione	60
Figura 39 - Creazione della facciata continua del corpo scale	61
Figura 40 - Rendering interno mediante Autodesk Revit	61
Figura 41 - Rendering esterno mediante Autodesk Revit	61
Figura 42 - Configurazione 1: piano terra	64

Figura 43 - Configurazione 1: primo piano	64
Figura 44 - Configurazione 1: piano secondo	65
Figura 45 - Configurazione 2: piano terra	67
Figura 46 - Configurazione 2: piano primo	67
Figura 47 - Configurazione 2: piano secondo	68
Figura 48 - Risultato finale dell'importazione	69
Figura 49 - Suddivisione dell'ambiente in mesh	70
Figura 50 - Posizionamento del focolare	71
Figura 51 - HRR di due sistemi di stereo	72
Figura 52 - HRR di libri e giornali	72
Figura 53 - HRR di Monitor	72
Figura 54 - HRR di sedie in legno	73
Figura 55 - HRR utilizzato	73
Figura 56 - Modifica della reazione all'interno del software Pyrosim	74
Figura 57 - Visualizzazione del fumo su Pyrosim	75
Figura 58 - Posizionamento del focolare	75
Figura 59 - HRR utilizzato per la seconda configurazione	76
Figura 60 - Modifica della reazione all'interno di Pyrosim	77
Figura 61 - Visualizzazione del fumo su Pyrosim	77
Figura 62 - Metodologia	81
Figura 63 - Interfaccia grafica di MassMotion	81
Figura 64 - Importazione del modello	82
Figura 65 - Conversione automatica degli elementi	82
Figura 66 - Scene Ribbon Bar	83
Figura 67 - Creazione dei "Link"	84
Figura 68 - Creazione delle scale	84
Figura 69 - Modello finale ottenuto per lo scenario 1 (Sala conferenze)	84
Figura 70 - Activities Ribbon Bar	85
Figura 71 - Definizione del profilo degli occupanti	85
Figura 72 - Creazione degli Evacuate	86
Figura 73 - Modello finale per lo scenario 1	87
Figura 74 - Modello finale per lo scenario 2	87
Figura 75 - Modifica dei tempi di pre-evacuazione per il secondo scenario	88
Figura 76 - Modifica dell'intervallo di output su Pyrosim	90
Figura 77 - Coefficiente di estinzione in funzione della lunghezza d'onda	93
Figura 78 - Importazione dei files all'interno di Paraview	94
Figura 79 - Inserimento dei files .q	94
Figura 80 - Visualizzazione delle mesh	95
Figura 81 - Interfaccia di fds2ascii	96
Figura 82 - Visualizzazione dei dati su Paraview	97
Figura 83 - Metodologia	98
Figura 84 - Creazione dell'isosuperficie	98
Figura 85 - Impostazioni per la creazione dell'isosuperficie in Paraview	99
Figura 86 - Visualizzazione della isosuperficie all'istante 0	100
Figura 87 - Visualizzazione della isosuperficie all'istante 200 corrispondete a 50 secondi	100

Figura 88 - Visualizzazione della isosuperficie all'istante 450 corrispondente a 112 secondi	101
Figura 89 – Trasformazione delle azioni in linguaggio Python	101
Figura 90 - Python Shell in Paraview	103
Figura 91 – Interfaccia di Deep Exploration	103
Figura 92 – Metodologia	105
Figura 93 – Creazione delle slices su Praview	105
Figura 94 - Visualizzazione della prima slice di visibilità su Paraview	106
Figura 95 - Visualizzazione della seconda slice di visibilità su Paraview	106
Figura 96 - Proprietà impostate rispettivamente per le slices 1 e 2 di visibilità	107
Figura 97 - Schermata principale del software Blender	108
Figura 98 - Visualizzazione degli oggetti importati su Blender	108
Figura 99 - Visualizzazione della finestra proprietà materiali	109
Figura 100 - Esecuzione dello script e attribuzione dei colori al vertice	110
Figura 101 - Esportazione dei files in formato .fbx	110
Figura 102 - Interfaccia grafica di Unity3D	112
Figura 103 - Esempio della finestra Inspector relativa ad un GameObject	113
Figura 104 - Finestra GameObject	114
Figura 105 - Finestra Component	115
Figura 106 - Metodologia di importazione del modello BIM	116
Figura 107 - Schermata principale del servizio di pubblicazione di BIM Tridify	117
Figura 108 - Plug-in Tridify in Unity3D	117
Figura 109 - Modello dell'edificio importato mediante Tridify	117
Figura 110 - Importazione delle isosuperfici su UNITY3D	118
Figura 111 - Assegnazione dello script ArrayTest	119
Figura 112 - Visualizzazione dell'isosuperficie corrispondente all'istante di tempo 150	120
Figura 113 - Visualizzazione dell'isosuperficie corrispondente all'istante di tempo 400	120
Figura 114 - Visualizzazione della slice di visibilità all'istante 150	121
Figura 115 - Visualizzazione della slice di visibilità all'istante 400	121
Figura 116 - Esempio della tipologia 1 di segnaletica su Unity3D	125
Figura 117 – Esempio della tipologia 2 di segnaletica su Unity	126
Figura 118 - Esempio della tipologia 3 su Unity	126
Figura 119 - Inserimento degli effetti sonori	127
Figura 120 - Esempio di descrizione del sistema di segnaletica	127
Figura 121 – Valori dei livelli di servizio	122
Figura 122 - Visualizzazione dell'Editor della Timeline	123
Figura 123 - Creazione della clip	123
Figura 124 - Risultato finale dell'importazione del file Alembic	124
Figura 125 - Inspector di un Canvas	128
Figura 126 - Creazione del Menù iniziale	129
Figura 127 - Esempio dell'Inspector del pulsante	130
Figura 128 - Creazione del menù principale	131
Figura 129 - Visualizzazione dello script Timer	133
Figura 130 - Visualizzazione del Timer realizzato	133
Figura 131 - Inserimento delle legende per la lettura delle slices	134
Figura 132 - Visualizzazione dell'isosuperficie dello scenario 1	136

Figura 133 - Visualizzazione dell'isosuperficie dello scenario 2 _____	136
Figura 134 - Visualizzazione della slice di visibilità dello scenario 1 _____	136
Figura 135 - Visualizzazione della slice di visibilità dello scenario 2 _____	137
Figura 136 - Visualizzazione degli occupanti nello scenario 1 _____	137
Figura 137 - Visualizzazione degli occupanti dello scenario 2 _____	137
Figura 138 - Differenze tra Monitor e Virtual Reality _____	142

## Indice delle tabelle

Tabella 1 - Sezioni del Codice di Prevenzione Incendi _____	7
Tabella 2 - Differenze tra approccio prescrittivo e prestazionale _____	9
Tabella 3 - Esempio di soglie di prestazione con il metodo di calcolo avanzato _____	12
Tabella 4 - Scenari di incendio di progetto _____	12
Tabella 5 - Grado di autosufficienza dell'individuo _____	21
Tabella 6 - Tipologia dell'illuminazione di emergenza secondo la norma UNI EN 1838 _____	25
Tabella 7 - Valori limite dei parametri fotometrici richiesti dalla norma UNI EN 1838 per le vie di esodo _____	26
Tabella 8 -Valori limite dei parametri fotometrici richiesti dalla norma UNI EN 1838 per l'illuminazione antipanico _____	27
Tabella 9 - Divisione della prima configurazione _____	62
Tabella 10 - Divisione della seconda configurazione _____	65
Tabella 11 - Caratteristiche dell'incendio considerate _____	74
Tabella 12 - Caratteristiche dell'incendio utilizzate _____	76
Tabella 14 - Tabella E.2 del rapporto tecnico ISO/TR 16738:2009 _____	79
Tabella 15 - Tempi di pre-movimento _____	80
Tabella 13 - Profili disponibili su MassMotion _____	85
Tabella 16 - Denominazione delle quantità su Paraview _____	96

# Introduzione

Il lavoro di tesi propone un metodo per l'esportazione della simulazione di incendio e della simulazione di esodo in realtà virtuale e della conseguente visualizzazione di esse. In particolare, viene proposto lo sviluppo di un applicativo per il supporto alla progettazione antincendio ai fini di avere il controllo dell'incendio, degli occupanti, dei sistemi di segnaletica e degli scenari di progetto.

L'obiettivo iniziale, infatti, era l'esportazione dei fumi in Virtual Reality in maniera tale da poter valutare il comportamento umano in caso di emergenza e la visibilità della segnaletica in presenza di fumo. Purtroppo, questo obiettivo non è stato possibile raggiungerlo in quanto al giorno d'oggi non esiste un collegamento diretto tra i software utilizzati per la simulazione di esodo e di incendio e il software utilizzato per l'impiego della Virtual Reality.

Malgrado questo problema, si è cercato un modo per poter visualizzare gli elementi legati alla simulazione di esodo e di incendio all'interno di Unity3D.

La metodologia utilizzata per il raggiungimento di questo obiettivo è riportata nello schema di seguito:

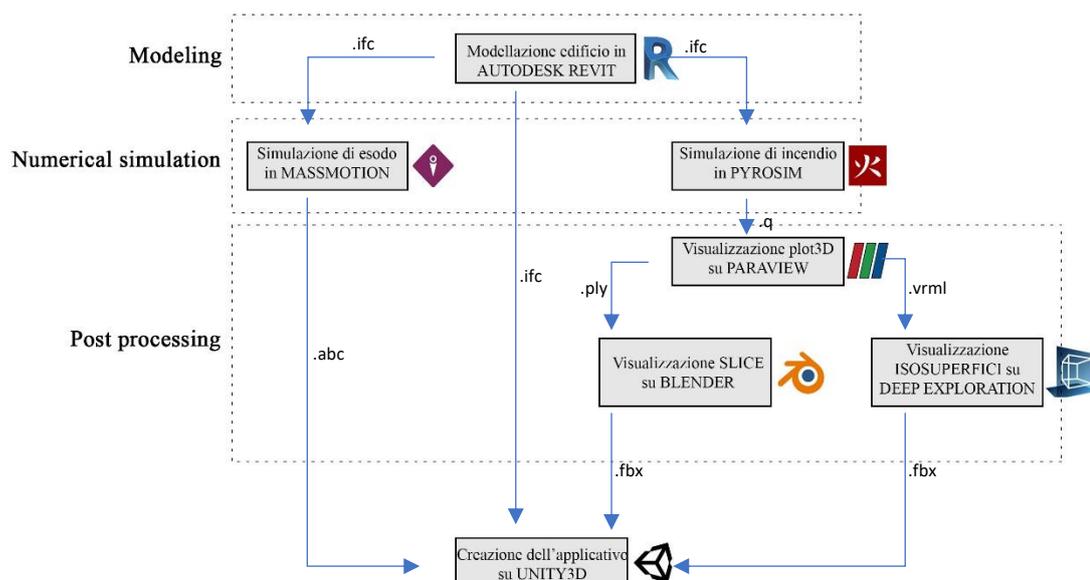


Figura 1 - Metodologia utilizzata

Il primo step effettuato è stato la modellazione dell'edificio su Autodesk Revit, successivamente si è importato il file in formato .ifc su Pyrosim per la simulazione di incendio e su MassMotion per la simulazione di esodo.

Pyrosim non permette l'esportazione diretta dei file 3Dplot che contengono le informazioni sul fumo, per cui, è stato necessario effettuare questo passaggio su altri software. Infatti, i file 3dplot sono stati importati su Paraview.

Per lo step successivo, sono stati scelti due software differenti per l'esportazione delle isosuperfici e delle slices; questa, è stata una scelta legata alla visualizzazione grafica degli elementi. In quanto avremmo potuto utilizzare un unico software (ovvero Blender) ma mediante delle prove effettuate, è risultato che l'esportazione mediante Deep Exploration portava ad una migliore omogeneità del contorno della isosuperficie. Infine, gli elementi esportati sono finalmente stati caricati su Unity3D, dove mediante l'utilizzo di alcuni script è stato possibile animarli.

Per quanto riguarda, invece, la simulazione di esodo effettuata su MassMotion l'esportazione è stata molto più semplice, in quanto questo programma permette l'esportazione del file Alembic.

Nonostante, il processo sembri complesso e lungo, si è cercato mediante degli script di automatizzare l'esportazione dei files, in maniera tale da avere tempi computazionali più brevi.

L'intero progetto è stato elaborato utilizzando la metodologia BIM (Building Information Modelling). Questo approccio punta alla collaborazione tra i progettisti, l'interoperabilità dei software, l'integrazione tra i processi; in questo modo, si permette alle varie discipline di lavorare e scambiarsi dati in modo efficiente e ottimizzato.

Il BIM è infatti un metodo di progettazione collaborativo in quanto consente in ogni momento di interagire in un unico modello, dalla fase di progettazione architettonica, a quella strutturale, impiantistica, energetica fino ad arrivare a quella finale gestionale.



*Figura 2 - Metodologia BIM*  
(<https://www.synchronia.com/en/cos-e-il-bim.html>)

## Capitolo 1

La definizione di prevenzione incendi viene definita dal D.P.R. 29 luglio 1982, n° 577 come “la materia di rilevanza interdisciplinare, nel cui ambito vengono promossi, studiati, predisposti e sperimentati misure, provvedimenti, accorgimenti e modi di azione intesi ad evitare, secondo le norme emanate dagli organi competenti, l'insorgenza di un incendio e a limitarne le conseguenze”.

### La prevenzione incendi

#### 1.1. La prevenzione incendi in Italia

La storia delle procedure di prevenzione incendi ha inizio negli anni '60, in quanto in Italia si ebbe un forte sviluppo industriale che rese necessario un controllo delle attività ritenute pericolose.

La prima legge che istituì i primi vincoli in materia di prevenzione incendi fu la **n° 966 del 1965**, che istituì il certificato di prevenzione incendi. Di seguito, il **D.P.R. n° 577 del 1982** che introdusse l'obbligo di chiedere l'esame del progetto delle attività soggette ai controlli di prevenzione incendi.

Dopo l'incendio del Cinema Statuto di Torino avvenuto la sera del 13 febbraio 1983, in cui persero la vita 64 persone, vennero effettuate perizie che dimostrarono che le cause erano sia dovute alla responsabilità di alcuni individui ma soprattutto a causa del sistema di leggi che era stato redatto e veniva applicato in maniera superficiale.

Per cui, nel **1984** con la **legge del 7 dicembre n° 818**, venne introdotto il nulla osta provvisorio per le attività soggette ai controlli di prevenzione incendi e la figura del professionista abilitato alla prevenzione incendi.

Al giorno d'oggi, questa figura è denominata Professionista antincendio a seguito del **D.M. 5 agosto 2011**.

Nel 1998 con il D.P.R. 37 viene introdotta la Dichiarazione d'Inizio Attività (DIA) e il D.M. del 10 marzo propone i criteri generali di sicurezza antincendio per la gestione delle emergenze nei luoghi di lavoro.

Con l'articolo 15 del Decreto Legislativo n°139 del 2006, "Norme tecniche e procedurali di prevenzione incendi", vengono adottate le norme tecniche di prevenzione incendi, fondate su presupposti tecnico-scientifici. Queste norme tecniche contengono le misure atte a ridurre la probabilità dell'insorgere di un incendio attraverso impianti che agiscono sulle sorgenti e a limitarne le conseguenze.

Nel 2007, con il D.M. 09/05/07 si verifica il passaggio dal metodo prescrittivo al metodo prestazionale; questo passaggio permette al progettista una maggiore libertà in termini di progettazione antincendio.

Il D.P.R. n°151 del 01/08/2011 elenca una serie di attività (per un totale di 80) che devono essere sottoposte ai controlli da parte del Corpo nazionale dei Vigili del Fuoco dividendole in 3 categorie:

- Nella categoria A rientrano quelle attività dotate di regola tecnica di riferimento, chiamata Regola Tecnica Verticale (RTV) e contraddistinte da un livello di complessità limitato;
- Nella Categoria B sono comprese tutte le attività presenti in A, e quindi dotate di regola tecnica, ma caratterizzate da un livello di complessità maggiore. Inoltre, sono comprese tutte le attività sprovviste di una specifica regolamentazione;
- Nella Categoria C sono state inserite tutte quelle attività con un alto livello di complessità indipendentemente dalla presenza o meno della regola tecnica;

Nel 2015 viene pubblicato il Nuovo Codice di Prevenzione Incendi con il D.M. 03/08/2015 con l'obiettivo di semplificare l'attuale corpo normativo della prevenzione incendi attraverso l'introduzione di un unico testo; il Nuovo Codice utilizza un nuovo approccio metodologico più aderente al progresso tecnologico e agli standard internazionali.

Il documento si compone di quattro sezioni:

- Sezione G – Generalità: contiene i principi fondamentali per la progettazione della sicurezza antincendio applicabili a tutte le attività soggette;
- Sezione S – Strategia antincendio: fornisce misure antincendio di prevenzione, protezione e gestionali applicabili a tutte le attività, per comporre la strategia antincendio al fine di ridurre il rischio di incendio;
- Sezione V – Regole Tecniche Verticali: fornisce indicazioni per alcune tipologie d'attività, in aggiunta o in sostituzione di quelle riportate della sezione S - Strategia antincendio;
- Sezione M – Metodi: riporta metodologie di progettazione antincendio innovative ed alternative a quelle riportate nelle prime tre sezioni, volte alla risoluzione di specifiche problematiche tecniche della progettazione antincendio (FSE);

Nel 2019 viene pubblicato il D.M. 18 ottobre 2019, entrato in vigore il 1° novembre 2019, “Approvazione di norme tecniche di prevenzione incendi, ai sensi dell’articolo 15 del decreto legislativo 8 marzo 2006, n° 139” che viene sostituito all’allegato 1 del D.M. 3 agosto 2015.

Lo scopo dell’aggiornamento è quello di migliorare la comprensibilità del testo, articolare le soluzioni conformi e risolvere alcune criticità riscontrate. L’entrata in vigore di questo decreto avviene a seguito dell’entrata in vigore del D.M. 12/04/2019 con il quale si è posto fine al cosiddetto “doppio binario”.

Infatti, dal 20 ottobre 2019, termina il periodo transitorio di applicazione volontaria del Codice per la progettazione delle attività non dotate di specifica regola tecnica.

Il Codice di Prevenzione Incendi [1], al giorno d'oggi è costituito dai seguenti capitoli:

Tabella 1 - Sezioni del Codice di Prevenzione Incendi



## 1.2. Evoluzione della normativa per i locali di intrattenimento e di pubblico spettacolo

Il concetto di locale di pubblico spettacolo raccoglie tutte quelle situazioni con la presenza di spettacoli o di intrattenimento, come ad esempio, manifestazioni sportive e conferenze.

Come detto in precedenza, con il **D.M. del 19 agosto 1996 n°214** si ha l'approvazione della regola tecnica di prevenzione incendi per la progettazione, costruzione ed esercizio dei locali di intrattenimento e di pubblico spettacolo.

Però, con l'entrata in vigore del Codice di Prevenzione Incendi con il D.M. 03/08/2015, si è ritenuto opportuno introdurre una nuova RTV di prevenzione incendi per i locali di intrattenimento e di pubblico spettacolo, da affiancare al D.M. 19/08/1996.

Per questo motivo, nella seduta del 30 settembre 2020 il Comitato Centrale Tecnico Scientifico ha presentato la bozza di Regola Tecnica Verticale (RTV) per le "Attività di intrattenimento e di spettacolo a carattere pubblico".

Si tratta di un'ulteriore Regola Tecnica Verticale che andrà ad integrare il Codice di Prevenzione Incendi.

La bozza di questa regola tecnica verticale riguarda le attività di intrattenimento e di spettacolo, anche a carattere temporaneo, svolte all'aperto o al chiuso. Vengono esclusi dal campo di applicazione: i luoghi all'aperto non delimitati, gli esercizi pubblici in cui vengono utilizzati strumenti musicali in assenza di attività danzanti e le attrazioni di spettacolo viaggiante.

### 1.3. Dal metodo prescrittivo al metodo prestazionale

L'approccio prescrittivo è basato sul concetto che "il rispetto di determinati requisiti nel progetto rappresenti la garanzia del raggiungimento dei livelli minimi di sicurezza".

In poche parole, l'approccio "consiste nel rispetto di regole tecniche, basate su strumenti di calcolo (ad esempio, gli Eurocodici per il calcolo analitico della classe REI delle strutture, le curve standard d'incendio, ecc.), e consistenti in misure preventive e protettive da rispettare come vincoli di progetto e codificate in disposizioni legislative. Il rispetto di tali regole rappresenta, dunque, garanzia di un livello di rischio residuo accettabile".

Questo tipo di approccio ha origine grazie alla prima legge sulla sicurezza sul lavoro, ovvero, il D.P.R. 547 del 27 aprile 1955 ma soprattutto, nella richiesta di misure di prevenzione e protezione antincendio in tutte le attività ritenute pericolose.

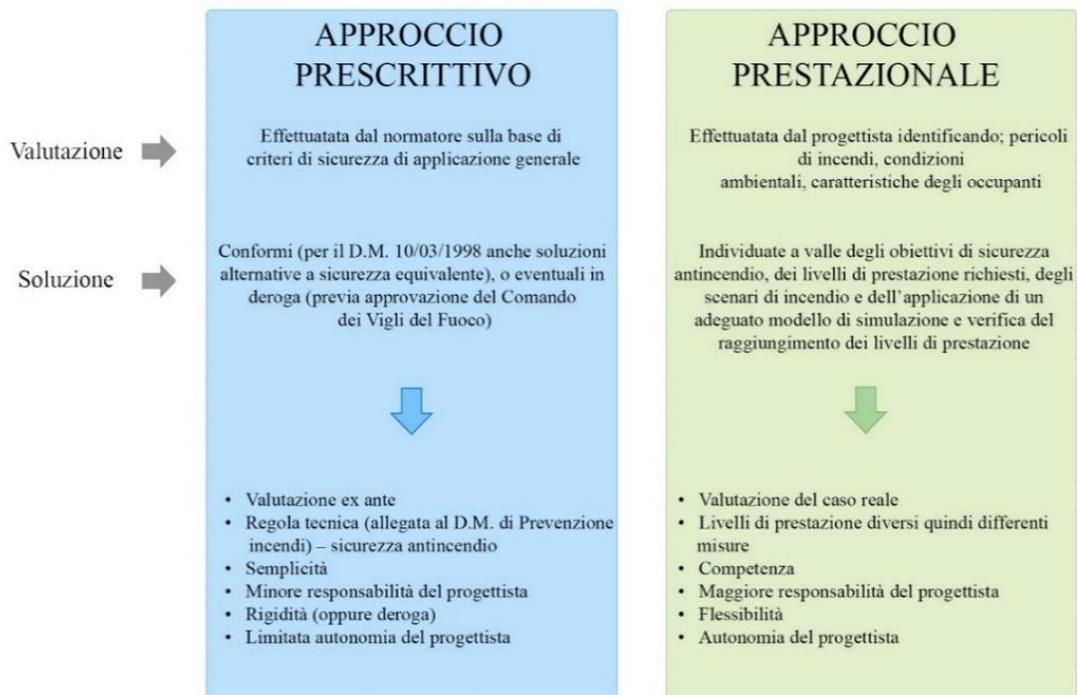
Uno dei vantaggi più importanti del metodo prescrittivo è rappresentato "dalla certezza del raggiungimento del livello minimo di sicurezza antincendio, il che spiega il largo utilizzo, a fronte dell'approccio prestazionale", inoltre, permette un maggior controllo e garantisce uniformità anche dal punto di vista del controllore dei Vigili del Fuoco.

Ma oltre i vantaggi, questo metodo presenta anche degli svantaggi:

- Non prende in considerazione la specificità del contesto, per questo motivo si rischia di non prendere in considerazione soluzioni altrettanto valide;
- Il progettista non è in grado di apportare modifiche se non mediante lo strumento della deroga.

Il metodo prescrittivo è stato l'unico approccio valido nella prevenzione incendi fino alla prima introduzione del metodo prestazionale con il D.M. 09/05/2007 "Direttive per l'attuazione dell'approccio ingegneristico alla sicurezza antincendio" e rafforzato con il Nuovo Codice di Prevenzione Incendi.

Tabella 2 - Differenze tra approccio prescrittivo e prestazionale



Questo approccio permette al progettista e al committente di definire gli obiettivi generali, individuando il livello di prestazione necessario alla singola attività, di cui poi spetta al professionista verificarne e garantirne il raggiungimento.

Mediante l'approccio prestazionale si individuano soluzioni utilizzando modelli matematici per prevedere gli effetti di un determinato evento.

#### 1.4. L'approccio prestazionale

L'approccio prestazionale o approccio di tipo ingegneristico o Fire Safety Engineering (ingegneria della sicurezza antincendio) viene applicato già da tempo in vari paesi dell'Unione Europea e del mondo, in quanto permette una più coerente aderenza delle misure di sicurezza antincendio al rischio specifico dell'attività o della costruzione.

La Fire Safety Engineering può essere volta alla Life Safety ovvero alla salvaguardia della vita delle persone oppure alla Structural Safety che riguarda l'analisi dell'incendio dopo la fase di flashover e lo studio, quindi, del livello di sicurezza strutturale.

L'applicazione di questo approccio comprende la valutazione:

- Dello sviluppo dell'incendio, valutando anche il moto e l'evacuazione dei fumi;
- Dei sistemi di rivelazione, di allarme e di estinzione;
- Delle prestazioni della struttura portante e della compartimentazione sottoposte ad elevata temperatura;

I metodi di tale approccio si trovano nella “sezione M – Metodi” del Codice di Prevenzione Incendi.

Il **capitolo M1** “Metodologia per l'ingegneria della sicurezza antincendio” contiene in l'iter progettuale da seguire per la progettazione prestazionale.

La metodologia descritta in questo capitolo si suddivide in due fasi: analisi preliminare e analisi quantitativa, descritte successivamente.

Il **capitolo M2** “Scenari di incendio per la progettazione prestazionale” contiene indicazioni per l'applicazione del metodo e l'individuazione degli scenari.

Infine, il **capitolo M3** “Salvaguardia della vita con la progettazione prestazionale” affronta lo studio dell'esodo dal punto di vista della Fire Safety Engineering ovvero, viene introdotto il criterio  $ASET > RSET$  dove ASET (Available Safe Escape Time) corrisponde al tempo disponibile per l'esodo e RSET (Required Safe Escape Time) corrisponde al tempo richiesto per l'esodo.

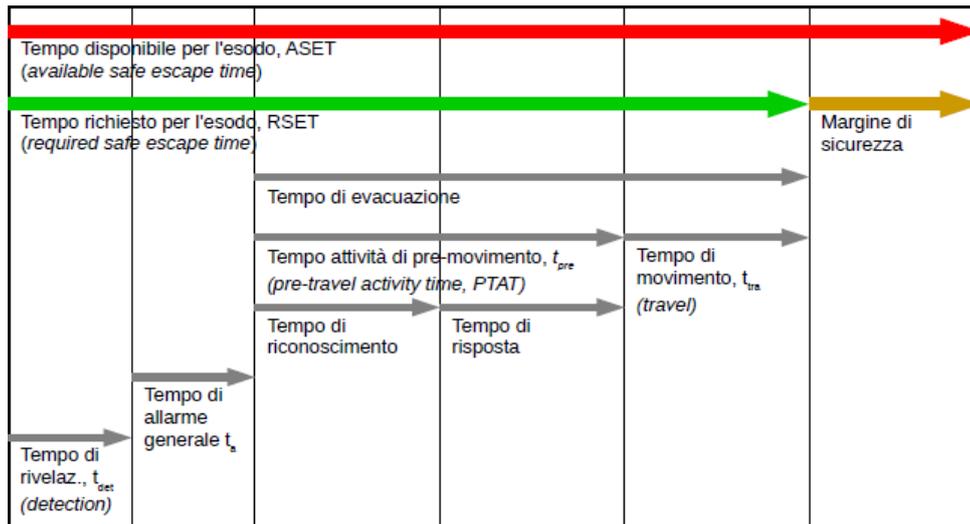


Figura 3 - Confronto tra ASET e RSET

#### 1.4.1. Prima fase – Analisi Preliminare

L'analisi preliminare si compone delle seguenti sotto-fasi:

- Definizione del progetto: in questa sotto-fase viene definito lo scopo del progetto e vengono indicati aspetti come: la destinazione d'uso, i vincoli progettuali, le caratteristiche degli occupanti ecc.
- Identificazione degli obiettivi della sicurezza antincendio: vengono specificati qualitativamente gli obiettivi di sicurezza antincendio come il livello di salvaguardia dell'incolumità degli occupanti e la continuità d'esercizio a seguito di un evento incidentale;
- Definizione delle soglie di prestazione: in questa sotto-fase vengono individuate le soglie di prestazione, di tipo quantitativo e qualitativo, fondamentali per la salvaguardia della vita umana, rispetto ai quali si può svolgere la valutazione oggettiva di sicurezza antincendio;

Tabella 3 - Esempio di soglie di prestazione con il metodo di calcolo avanzato

Modello	Prestazione	Soglia di prestazione
Oscureamento della visibilità da fumo	Visibilità minima valutata ad altezza 1,80 metri dal piano di calpestio	Occupanti: 10 m
		Occupanti in S< 100 mq: 5 m
Gass tossici	FED (Fractional Effective Dose) e FEC (Fractional effective Concentration) per esposizione a gas tossici e gas irritanti ad altezza 1,80 dal piano di calpestio	Soccorritori: 5 m
		Soccorritori in S<100 mq: 2,5 m
Calore	Temperatura massima di esposizione	Occupanti: 0,1
		Soccorritori: nessuna valutazione
	Irraggiamento termico massimo da tutte le sorgenti di esposizione	Occupanti: 60°C
		Soccorritori: 80°C
Irraggiamento termico massimo da tutte le sorgenti di esposizione	Occupanti: 2,5 kW/m <sup>2</sup>	
	Soccorritori: 3 Kw/m <sup>2</sup>	

- Individuazione degli scenari di progetto: si individuano gli scenari, ovvero una schematizzazione degli eventi più gravi che possono verificarsi all'interno dell'attività in esame, in relazione alle caratteristiche del focolare, dell'edificio e degli occupanti.

Tabella 4 - Scenari di incendio di progetto



L'incendio viene caratterizzato dalle curve di incendio fondamentali per la definizione di uno scenario. Queste curve stabiliscono una relazione tra HRR (Heat Release Rate) e il tempo. La curva è costituita da quattro stadi: innesco, pre-flashover, flashover e post-flashover.

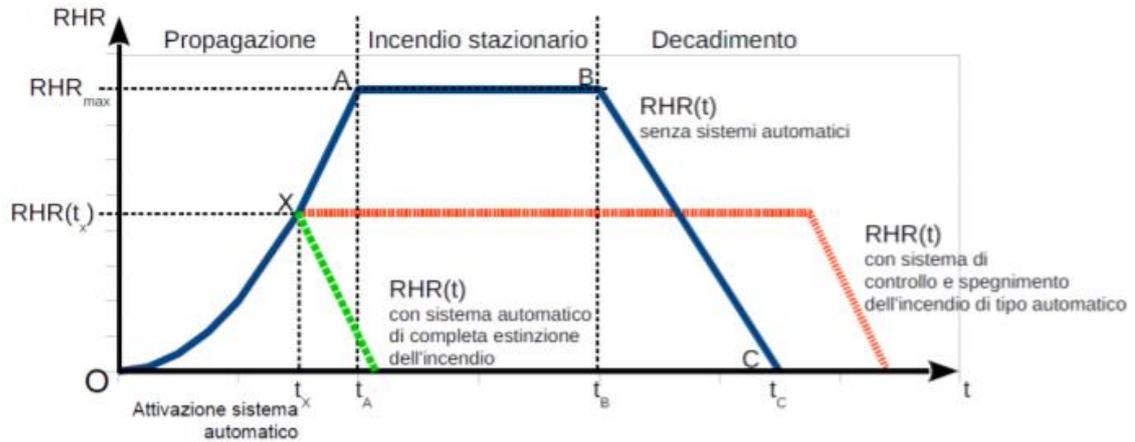


Figura 4 - Fasi dell'incendio

## 1.4.2. Seconda fase – Analisi Quantitativa

L'analisi quantitativa si compone delle seguenti sotto-fasi:

- Elaborazione delle soluzioni progettuali: in questa sotto-fase vengono elaborate una o più soluzioni progettuali, da sottoporre alla successiva verifica degli obiettivi di sicurezza antincendio.
- Valutazione delle soluzioni progettuali: in questa fase, vengono calcolati gli effetti sull'attività degli scenari di incendio di progetto, per ciascuna soluzione progettuale individuata nella fase precedente. Per cui, il professionista antincendio utilizza un modello di calcolo analitico o numerico.
- Selezione delle soluzioni progettuali idonee: in questa sotto-fase, la soluzione progettuale finale viene scelta dal professionista antincendio fra quelle che sono state valutate positivamente.

## 1.5. Modellazione di incendio

Per la valutazione delle soluzioni progettuali il professionista antincendio può scegliere tra i modelli di calcolo disponibili:

- Modelli analitici: sono basati sui principi della chimica e fisica dell'incendio (conservazione di massa dell'energia e della quantità di moto) e a loro volta si dividono modelli a zone e modelli di campo.

Il **modello a zone** permette di studiare la dinamica di un incendio in un ambiente confinato, risolvendo le equazioni di conservazione in due zone omogenee: una zona superiore che presenta i prodotti della combustione e una zona inferiore libera da fumo.

I modelli a zone stimano in funzione del tempo, le temperature medie delle due zone, la concentrazione di ossigeno, la concentrazione di ossido di carbonio, la visibilità e il flusso in entrata e in uscita di aperture verso l'esterno o altri locali. Questo tipo di modello non è idoneo per ambienti stretti e lunghi in quanto non è soddisfatta l'ipotesi di uniformità spaziale delle proprietà termofisiche in ciascuna zona e non è idoneo per sistemi complessi poiché non dotato di modello di combustione e di turbolenza.

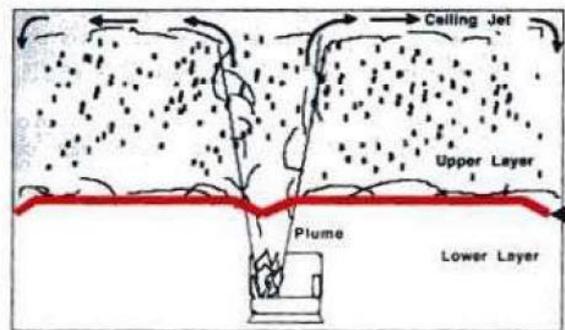


Figura 5 - Modelli a zone

Per quanto riguarda i **modelli di campo** permettono di studiare l'incendio per via numerica, dividendo lo spazio in un numero elevato di celle e risolvendo all'interno di ciascuna cella, le equazioni di conservazione di massa, di energia, ecc. attraverso il metodo degli elementi finiti.

Mediante questo modello, si possono studiare geometrie differenti e di forme complesse grazie alla definizione delle celle tridimensionali.

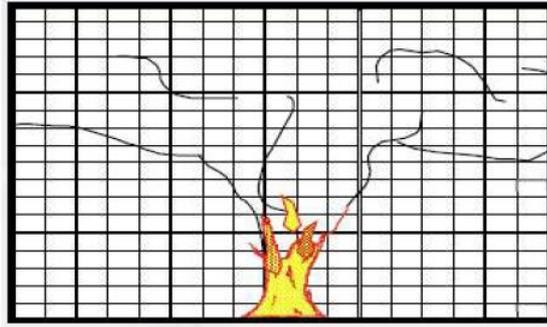


Figura 6 - Modelli di campo

- Modelli numerici:
  - Simulazione di incendio a zone per ambienti confinati
  - Simulazione di incendio di campo
  - Simulazione di esodo
  - Analisi termostrutturale

### 1.6. Modellazione di esodo

Le dinamiche dell'esodo possono essere studiate mediante due tipi di modelli:

- Il modello idraulico: definisce le caratteristiche del moto degli occupanti secondo un approccio fluidodinamico, infatti, è basato sulla risoluzione delle equazioni sulla regolazione dei fluidi di Navier-Stokes. Questo modello non tiene conto degli effetti comportamentali e decisionali delle persone, come ad esempio, la scelta della via da percorrere ad un bivio nel percorso seguito.
- Il modello agent-based: basato sulle interazioni persona-struttura-ambiente, in cui viene risolta un'equazione di moto per ogni occupante e si tiene conto di forze sociali, familiarità, comportamento. In questo modello, è il progettista ad imporre le leggi comportamentali degli occupanti.

Lo schema di movimento dell'occupante è composto da due fasi, in primo luogo riconosce il contesto nel quale è collocato, successivamente esegue le regole che il modello prevede per il contesto individuato.

## Capitolo 2

Gli aspetti da analizzare per lo studio dell'esodo in emergenza sono molteplici, è necessario effettuare uno studio sul **comportamento umano** e uno studio sull'illuminazione e i **sistemi di segnaletica**.

Lo studio del comportamento umano in emergenza può essere effettuato sia dal punto di vista del singolo individuo analizzando le caratteristiche ad esso annesse oppure analizzando le dinamiche di esodo di una collettività intesa come gruppo di occupanti. In questo capitolo vengono analizzate le caratteristiche individuali e collettive che bisogna tenere in considerazione per lo studio del comportamento umano in emergenza

### Caratteristiche dell'individuo in una situazione di emergenza

L'esodo in emergenza è un argomento protagonista di numerosi studi che sono stati condotti sempre con maggiore interesse negli ultimi anni. Questo perché, le misure atte a favorire l'esodo degli occupanti sono sicuramente le più importanti per la salvaguardia della vita umana.

È importante tenere presente, come il tempo di evacuazione sia strettamente dipendente dal comportamento umano sia dal punto di vista del singolo individuo, ma soprattutto dal punto di vista collettivo.

#### 2.1. Caratteristiche del singolo individuo

Per assicurare l'esodo degli occupanti e comprendere al meglio il comportamento umano in emergenza, è necessario tenere in considerazione una serie di fattori che lo influenzano. Gli individui, infatti, di fronte ad una situazione di emergenza, prendono decisioni che variano a seconda di una serie di caratteristiche riassumibili come di seguito:

- Genere: dallo studio di numerose situazioni di emergenza, sono emerse delle differenze relative al comportamento tra uomo e donna. Gli uomini,

generalmente, sono più propensi a cercare di aiutare a spegnere il fuoco, mentre le donne preferiscono raggiungere la famiglia e mettersi in sicurezza;

- Et : secondo alcuni studi effettuati da Kose [2],   possibile valutare il comportamento dei singoli individui in base all'et  utilizzando tre categorie: abilit  sensoriali, decisionali e di azione. Le persone anziane sono pi  esposte al rischio, in quanto, possano avere una scarsa resistenza all'effetto debilitante del fumo e del calore;
- Capacit  fisiche: durante una situazione di emergenza, possono essere presenti persone con disabilit  fisiche e sensoriali, che potrebbero rallentare sia la propria evacuazione sia quella degli altri individui;
- Familiarit : in caso di emergenza, gli individui tendono a muoversi verso luoghi o persone a loro familiari;
- Attaccamento sociale: questo fattore pu  rallentare l'evacuazione degli occupanti a causa del legame affettivo, inoltre questo fattore porta ad una riduzione del pericolo;
- Attaccamento agli oggetti: prima di evacuare l'edificio, alcune persone tendono a recuperare i propri oggetti, anche se ci  compromette o rallenta la loro fuga;
- Cultura: persone aventi culture o religioni differenti possono avere comportamenti diversi in una situazione di emergenza. Infatti, persone provenienti da Nazioni differenti possono avere una percezione diversa del pericolo;

Per cui, di fronte al pericolo ogni individuo agisce in maniera differente. Conoscere il fattore umano in una situazione di emergenza   fondamentale per aiutare a migliorare l'azione di soccorso e a garantire la sicurezza degli individui coinvolti.

## 2.2. Caratteristiche collettive

Il comportamento del singolo individuo si trasforma in un comportamento collettivo quando la sua attività di esodo è influenzata dai suoi vicini e ne modifica il comportamento.

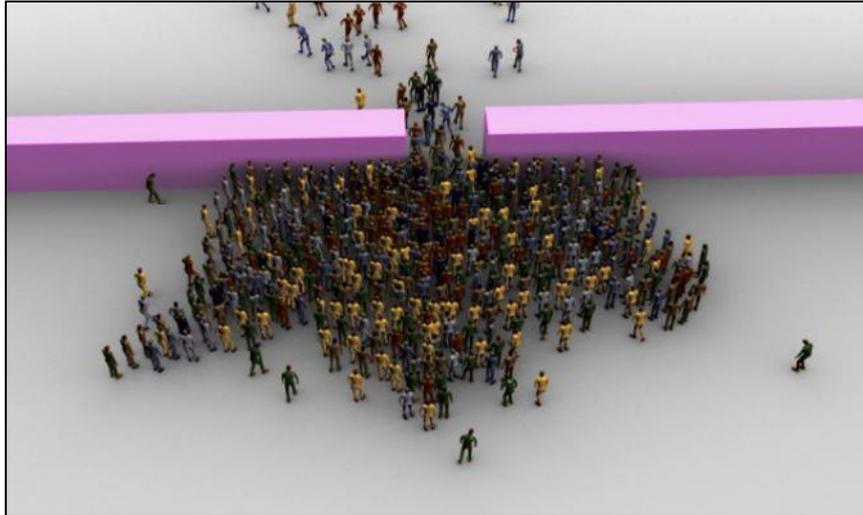
Nel 1970 Bibb Latané e John Darley [3], due psicologi della Columbia University conducono una serie di esperimento con l'ausilio di studenti universitari.

L'esperimento consisteva nel raggruppare gli studenti in una stanza con la porta chiusa, intenti a compilare un questionario e osservare la loro reazione nel momento in cui dopo alcuni minuti al di sotto della porta della stanza veniva pompato del fumo, per simulare una situazione di incendio al di fuori della stanza. I risultati ottenuti sono i seguenti:

- Se gli studenti erano da soli all'interno della stanza, dopo che il fumo iniziava ad espandersi nell'ambiente, il 75% delle persone presenti all'interno della stanza era in allarme dopo pochi secondi e già pronto a fuggire;
- Se nella stanza erano presenti anche dei "complici" che facevano finta di nulla e continuavano a compilare il questionario, solo il 26% degli studenti era già pronto a fuggire pochi secondi dopo la diffusione del fumo;

Per cui, da questo si può affermare che la presenza di altre persone calme e tranquille, può rallentare o modificare la percezione del pericolo in una situazione di emergenza. Per tale motivo, è necessario conoscere le tipologie di comportamento che la folla può assumere.

In una situazione di emergenza, le persone tendono a muoversi con più velocità a causa dello stato di ansia e preoccupazione per la sopravvivenza, questo provoca la creazione di code in corrispondenza delle uscite con la formazione delle strutture ad arco.



*Figura 7 - Esempio di formazione ad arco*

*(<https://www.testo-unico-sicurezza.com/comportamento-umano-in-caso-di-incendio-modelli-di-evacuazione-2017.html>)*

Più gli occupanti si muovono velocemente verso l'uscita, più si rallentano a causa dell'accalcamento che si genera in corrispondenza di esse. Inoltre, la velocità può causare la caduta degli occupanti che diventano dei veri e propri ostacoli.

Ne è un esempio, la **tragedia di Piazza San Carlo** a Torino, adibita con il grande schermo per la partita della Juventus e Real Madrid il 3 giugno 2017. A seguito di un forte rumore improvviso, la folla ha iniziato a correre verso i lati della piazza causando un grande numero di feriti.

Inoltre, un altro comportamento, chiamato comportamento pro-sociale, spesso riscontrato in condizioni di emergenza, è la cooperazione tra gli occupanti. Infatti, alcuni tendono a mostrare capacità di leadership permettendo l'evacuazione degli occupanti in difficoltà.

Uno studio interessante, che mette in luce come i legami sociali contribuiscono sui tassi di mortalità nelle situazioni di emergenza, è quello effettuato da Sime [4]. Egli mediante delle interviste, ha osservato come il 70% degli individui fuggono con una o più persone che nella maggior parte dei casi sono familiari.

Un altro studio è quello effettuato da Proulx nel 1995 [5], afferma che i gruppi di occupanti assumono la velocità del membro più lento, come anziani o bambini, ragione per cui si ha un ritardo nell'evacuazione.

Un ulteriore studio effettuato sul comportamento collettivo in emergenza ha osservato come i gruppi di occupanti convergono verso un luogo comune caratterizzato dall'assenza di fumo o da maggiore visibilità; in questo modo, gli occupanti percepiscono una riduzione dell'ansia e della tensione provocate dalla situazione. Infine, è stato osservato come alcune persone sfuggite all'incendio, spesso scelgono di rientrare all'interno dell'edificio; uno degli studi più importanti condotto da John Bryan nel 1956 [6] sull'incendio ad Arundel Park, evidenziò che un terzo delle persone che si erano messe in salvo erano successivamente rientrate per salvare le persone care o recuperare gli effetti personali.

### 2.3. Processo del comportamento umano in emergenza

È importante avere un'idea del modo in cui le persone tentano di scappare in un processo di evacuazione.

È possibile identificare tre fasi nel processo di evacuazione di un essere umano:

- Consapevolezza del pericolo da stimoli esterni (**processo di osservazione**)
- Validazione e reazione ai segnali di allarme (**processo decisionale**)
- Spostarsi/rifugiarsi verso un luogo sicuro (**punto di spostamento**)

La storia ha dimostrato più volte che le persone hanno un basso livello di consapevolezza in relazione al pericolo. Un allarme non è sempre preso sul serio; questa supervisione degli occupanti è causata dall'enorme quantità di falsi allarmi che le persone hanno subito nel corso della loro vita.

Nel processo decisionale sono importanti una serie di fattori. Come i fattori sociali: infatti, le persone, come detto in precedenza, hanno la tendenza a seguire altre persone che possono portarle in pericolo piuttosto che in sicurezza; Altri fattori sono la disposizione dell'edificio e altri fattori ambientali come l'ubicazione dell'incendio e la presenza e il volume di fumo nella struttura o nello spazio.

Ci sono molti fattori che hanno un'influenza sullo spostamento verso un ambiente sicuro: come la segnaletica, lo stress, la conoscenza dell'edificio, la pressione dei tempi, le caratteristiche dell'edificio, essere soli o in un gruppo ed età e funzionalità della persona. Anche questo tipo di fattori lo hanno un'influenza sull'autosufficienza di una persona in caso di incendio.

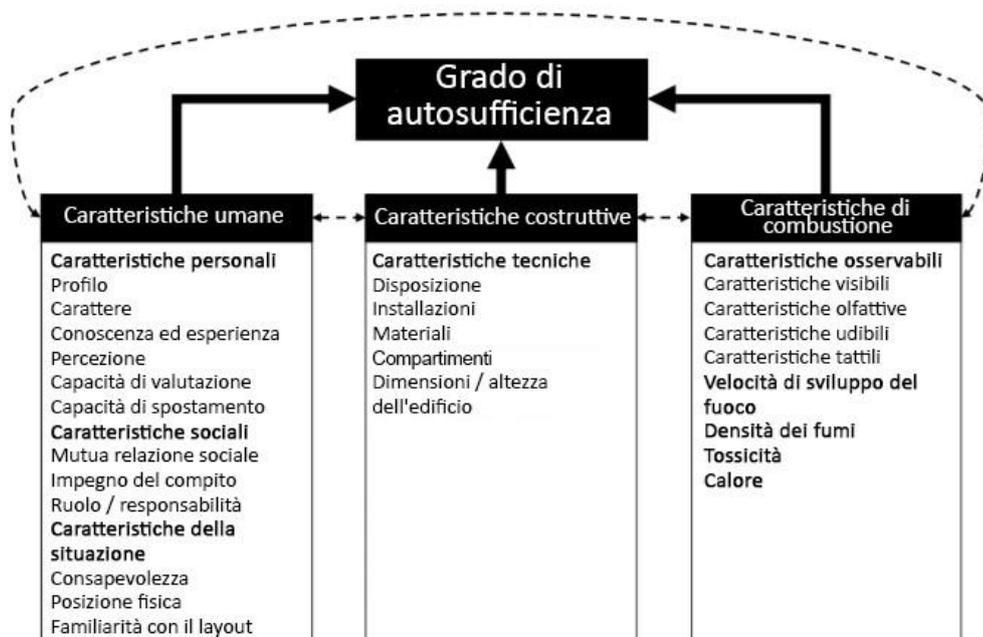
*"L'autosufficienza in caso di incendio è la capacità umana di notare i segnali di pericolo, di interpretarli e la capacità di prendere decisioni per intraprendere azioni finalizzate alla sopravvivenza della situazione di incendio."*

Dallo studio della letteratura è emerso che sono tre le categorie di fattori che determinano il grado di fiducia in sé stessi in caso di incendio:

- Caratteristiche dell'incendio
- Caratteristiche umane
- Caratteristiche costruttive

Questi tre gruppi di fattori sono suddivisi in diverse caratteristiche dettagliate:

Tabella 5 - Grado di autosufficienza dell'individuo



Diverse fasi del processo di evacuazione possono essere applicate per migliorare l'osservazione, la consapevolezza e lo spostamento.

Per cui, il comportamento umano durante l'evacuazione di un edificio in caso di emergenza è il risultato di un processo comportamentale.

### 2.4. Il concetto di panico

Il termine “panico” viene molto spesso associato ai disastri, ma ha subito numerose modifiche nel corso della storia. Inizialmente si pensava, che le persone in situazioni di emergenza perdessero la loro umanità trasformandosi in animali in preda alla paura.

Successivamente alcuni studiosi hanno cercato di dare un significato a questo termine, come Mawson che definisce il panico come: “Paura e/o fuga inadeguata o eccessiva”. Nel 1954 Quantarelli [7] definisce il panico come “una reazione acuta di paura segnata dalla perdita di autocontrollo che è seguita da un comportamento non sociale e non razionale”. Secondo questo studioso, il panico è un fenomeno di breve durata che di conseguenza si traduce in calpestio e schiacciamento della folla in quanto le persone tendono a perdere temporaneamente la capacità di controllare le proprie azioni ed emozioni.

Sime [4] nel suo studio sul concetto di panico sottolinea come uno dei problemi legati all'utilizzo di tale termine sia la prospettiva differente tra osservatore esterno e persona a cui viene attribuito il comportamento di panico. Infatti, tale comportamento viene definito, generalmente, da un osservatore esterno all'accaduto che ha una prospettiva totalmente differente dalla persona che vive l'accaduto stesso.

Infatti, numerosi sono i casi, in cui il concetto di panico è stato valutato in maniera inappropriato per descrivere e spiegare il comportamento umano in caso di emergenza, come ad esempio nell'incendio del Beverly Hills Supper Club avvenuto nel 1977. In questo caso, i giornali descrissero la causa della grande perdita di vite umane come “persone intrappolate nel panico”, in realtà, le interviste con i sopravvissuti all'incendio hanno rilevato che il personale e gli occupanti hanno reagito in modi tutt'altro che presi dal panico, rispettando i propri ruoli e prendendosi cura dei propri clienti, rendendo più facile l'evacuazione della discoteca.

Inoltre, la mancanza di panico, contrariamente a quanto si crede, è quella che contribuisce alle conseguenze legate agli incendi. Neil Townsend, funzionario della divisione del London Fire Rescue Service, sottolinea che quando le persone muoiono nel fuoco, non è a causa del panico, ma è più probabile che sia la mancanza di panico.

Pertanto, è fondamentale conoscere quali sono le diverse caratteristiche e fattori, come età, sesso, impatto culturale, per poter comprendere il comportamento umano in emergenza.

### 2.5. Incendi in locali di pubblico spettacolo

Numerosi sono gli incendi avvenuti nei locali di pubblico spettacolo, come ad esempio, l'incendio del **nightclub The Station** avvenuto il 20 febbraio 2003 a Rhode Island durante un concerto di un gruppo musicale statunitense. I sopravvissuti furono all'incirca 230 persone, che successivamente hanno affermato di aver avuto difficoltà nel trovare le vie d'uscita a causa della presenza di fumo, inoltre, il personale ha aiutato gli occupanti ad evacuare l'edificio e ha diffuso la notizia dell'incendio. Questo è uno dei casi in cui gli occupanti hanno cercato di scappare dall'ingresso principale in quanto era familiare; questo ha causato il rallentamento dell'evacuazione dell'edificio.

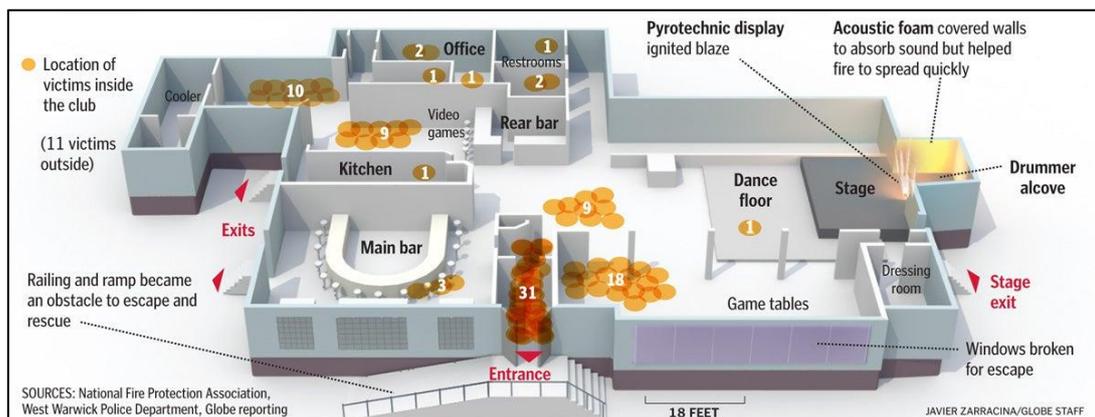


Figura 8 - Incendio The Station Nightclub

(<http://c.o0bg.com/2013/02/17/bgcom-fire/F1bKsdLNcY9zywFmt34tMI/story.html>)

L'incendio della discoteca Kiss [8] avvenuta in una discoteca di santa Maria in Brasile il 27 gennaio 2013. All'interno dell'edificio erano presenti all'incirca 2000 persone che non appena le fiamme si propagarono, presi dal panico, si diressero in massa verso l'uscita di sicurezza più vicina. Questo causò l'accalcamento e la morte di circa 230 persone a causa di asfissia.

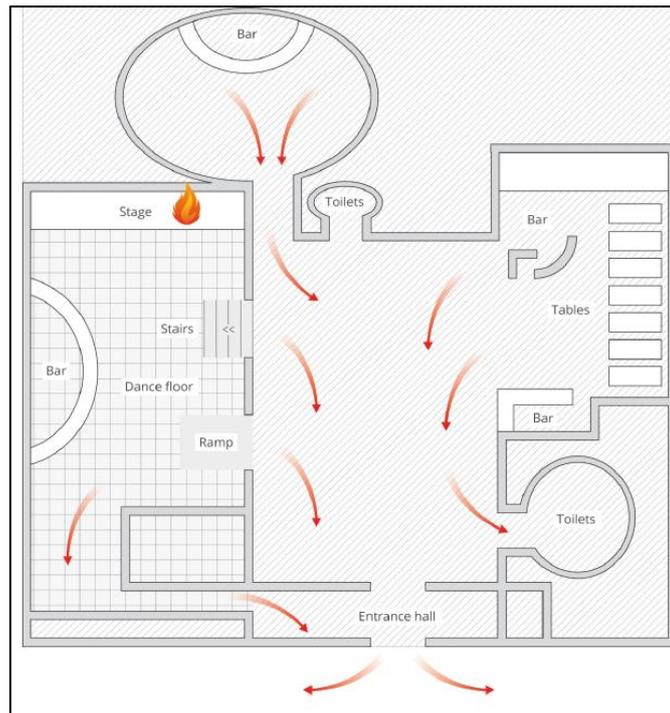


Figura 9 - Incendio The Kiss Nightclub

L'edificio presentava una sola uscita di sicurezza e davanti ad essa c'erano delle barriere metalliche progettate per far entrare le persone in modo lento e controllato, che in questo caso, hanno ulteriormente rallentato l'uscita degli occupanti. Inoltre, l'edificio non presentava finestre ma era presente un sistema di allarme antincendio, un sistema di controllo del fumo e del calore e l'illuminazione di emergenza. Nonostante la segnaletica e l'illuminazione di emergenza, alcuni testimoni hanno raccontato che durante l'evacuazione, le persone hanno scambiato le porte che conducevano a vicoli ciechi per uscite di emergenza. Infatti, molte delle persone morte nell'incendio furono trovate in prossimità dei bagni.

L'**incendio al Ghost Ship** nel quartiere Fruitvale di Oakland in California avvenuto il 2 dicembre del 2016, a causa di alcuni danni al sistema elettrico. All'interno dell'edificio erano presenti all'incirca 100 persone, di cui 36 hanno perso la vita a causa dell'inalazione di fumo. [9]

## Capitolo 3

Come detto in precedenza, gli aspetti da analizzare per l'esodo in emergenza sono il comportamento umano, l'illuminazione e i sistemi di segnaletica.

Per comprendere e prevedere al meglio il comportamento umano in emergenza, è fondamentale la progettazione di un sistema di esodo semplice ed efficiente che permetta agli occupanti di prendere la decisione migliore da intraprendere per raggiungere il luogo sicuro.

In questo capitolo, viene analizzata l'illuminazione di emergenza e la segnaletica di sicurezza con le rispettive normative.

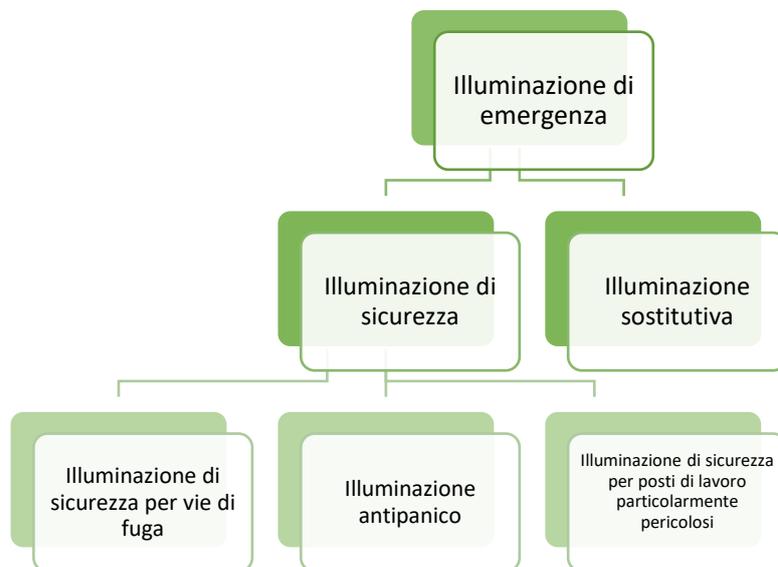
### Illuminazione di emergenza

Al giorno d'oggi, la segnaletica luminosa è ovunque. Unita a un impianto d'illuminazione indipendente dalla rete, essa contribuisce in misura fondamentale alla sicurezza.

In caso di pericolo o di mancanza della corrente, è necessario garantire che si possa abbandonare l'edificio in sicurezza e che siano presenti misure di salvataggio.

La normativa di riferimento è la UNI EN 1838 "Applicazioni illuminotecniche – Illuminazione d'emergenza" che prevede la seguente suddivisione:

Tabella 6 - Tipologia dell'illuminazione di emergenza secondo la norma UNI EN 1838



L'**illuminazione di sicurezza** [11] permette di fornire un livello di sicurezza adeguato quando si presentano situazioni di mancanza dell'illuminazione ordinaria, in modo tale da evitare che avvengano incidenti. Questa tipologia di illuminazione non può essere utilizzata per svolgere attività ordinarie, ma il suo unico scopo è quello di permettere la mobilità delle persone in totale sicurezza.

Il livello di illuminamento richiesto dal Codice di Prevenzione Incendi nel paragrafo S.4.5.10 "Illuminazione di sicurezza" è  $\geq 1$  lx lungo la linea centrale delle vie d'esodo. A sua volta, l'illuminazione di sicurezza è suddivisa in:

- **Illuminazione delle vie di esodo:** questa tipologia di illuminazione ha lo scopo di consentire alle persone presenti all'interno dell'edificio di identificare le vie di fuga. Le vie di esodo devono essere illuminate e anche segnalate; infatti, la segnaletica di sicurezza si inserisce nel contesto dell'illuminazione di esodo. È utile notare che, i livelli di illuminamento previsti per l'esodo devono essere forniti solamente da apparecchi di illuminazione, non tenendo conto dell'eventuale luce emessa dalla segnaletica retroilluminata. Infatti, la segnaletica retroilluminata contribuisce sull'illuminamento, ma non sostituisce gli apparecchi illuminanti.

Tabella 7 - Valori limite dei parametri fotometrici richiesti dalla norma UNI EN 1838 per le vie di esodo

Illuminamento						
$E_{\text{minimo}} = 1 \text{ lx}$ (illuminamento orizzontale minimo sul pavimento)						
Uniformità						
$E_{\text{massimo}} : E_{\text{minimo}} \leq 40 : 1 \text{ lx}$ $d < 4\text{h}$ installazione						
Limitazione dell'abbagliamento						
h/m	< 2,5	$2,5 \leq h < 3$	$3 \leq h < 3,5$	$3,5 \leq h < 4$	$4 \leq h < 4,5$	$\geq 4,5$
$L_{\text{max}}/\text{cd}$	500	900	1600	2500	3500	5000
I valori di questa tabella all'interno della zona da $60^\circ$ a $90^\circ$ rispetto alla verticale non devono essere superati in tutte le angolature azimutali.						
Resa del colore						
$R_a \geq 40$						
Autonomia nominale nelle vie di fuga						
UNI EN 1838 art. 4.2.5 e art. 4.3.5						
1 ora						
Velocità di accensione						
UNI EN 1838 art. 4.2.6 e art. 4.3.6						
Entro 5 secondi al 50 %, entro 60 secondi al 100 % dell'illuminamento previsto						

Nel momento in cui un'uscita non è riconoscibile immediatamente oppure se possono sorgere dubbi sulla sua posizione, è necessario indicarne la posizione mediante una sequenza di pittogrammi che permettono all'occupante di essere guidato fino all'uscita.

- **Illuminazione antipanico:** questa tipologia di segnaletica fornisce l'illuminazione necessaria per raggiungere le vie di fuga riducendo la possibilità che tra gli occupanti si scateni il panico;

Tabella 8 - Valori limite dei parametri fotometrici richiesti dalla norma UNI EN 1838 per l'illuminazione antipanico

Illuminamento UNI EN 1838						
E (orizzontale sul pavimento) $\geq 0,5$ lx h installazione > 2m (non sono considerate le zone marginali di larghezza 0,5 m)						
Uniformità						
$E_{\text{massimo}} : E_{\text{minimo}} \leq 40 : 1$ lx						
Limitazione dell'abbagliamento						
h/m	< 2,5	$2,5 \leq h < 3$	$3 \leq h < 3,5$	$3,5 \leq h < 4$	$4 \leq h < 4,5$	$\geq 4,5$
$L_{\text{max}}/\text{cd}$	500	900	1600	2500	3500	5000
I valori di questa tabella all'interno della zona da 60° a 90° rispetto alla verticale non devono essere superati in tutte le angolature azimutali.						
Resa del colore						
$R_a \geq 40$						
Autonomia nominale UNI EN 1838 art. 4.2.5 e 4.3.5						
1 ora fatte salve disposizioni legislative particolari.						
Velocità di accensione UNI EN 1838 art. 4.2.6 e art. 4.3.6						
Entro 5 secondi al 50 %, entro 60 secondi al 100 % dell'illuminamento previsto						

Le normative a cui fare riferimento per l'illuminazione antipanico sono le UNI EN 1838 e la UNI EN 50172 (CEI 31-11).

- **Illuminazione delle aree ad alto rischio:** questa tipologia di illuminazione ha l'obiettivo di garantire la sicurezza delle persone che lavorano in situazioni pericolose;

L'**illuminazione di riserva** serve per poter continuare le stesse attività che si stavano svolgendo durante il funzionamento dell'illuminazione ordinaria. Per questo motivo, è necessario che il livello di illuminamento dell'illuminazione di riserva sia almeno pari a quello dell'illuminazione ordinaria, in quanto, in caso contrario, non sarebbe possibile continuare a svolgere la stessa attività. Solo in un caso è consentito avere un

livello di illuminazione di riserva inferiore a quello dell'illuminazione normale: se viene utilizzata solo per terminare e chiudere l'attività in corso e non per continuarla.

### 3.1. La segnaletica di sicurezza

La segnaletica di sicurezza antincendio è un elemento essenziale per la protezione antincendio perché aiuta a garantire la salvaguardia delle persone che occupano un edificio. Infatti, viene utilizzata per indicare le attrezzature di pronto soccorso, le vie di fuga, i punti di chiamata di allarme antincendio, le uscite di emergenza ecc.

La necessità di trasmettere informazioni in materia di sicurezza con un sistema che, per quanto possibile, non faccia ricorso all'utilizzo di testi e sia facilmente comprensibile, ha spinto alla normalizzazione di segni grafici.

Ma nella loro routine quotidiana, le persone tendono a non essere consapevoli della loro presenza o a ricordare le loro posizioni. Queste conclusioni sono state stabilite da Mc Clintock, Shields, Reinhardt-Rutland e Leslie nei loro esperimenti riguardanti l'irrelevanza appresa della segnaletica delle uscite di emergenza [12].

Infatti, uno studio condotto dall'Università di Greenwich [13] (Fire Safety Engineering Group, FSEG) ha evidenziato che la segnaletica delle vie di esodo spesso viene ignorata: il 38% delle persone “guarda” il cartello rettangolare di colore verde e bianco con il pittogramma dell'omino verde che corre o con le frecce direzionali, ma non lo percepisce davvero.

Ciò è stato spiegato come un fenomeno cognitivo di **irrelevanza acquisita** per esposizione, cioè il cervello tende ad ignorare quei segnali mai utilizzati; dunque, solo una percezione efficace può assicurare l'acquisizione delle informazioni.

Nel loro caso in particolare, Mc Clintock [12] ha proposto l'uso di luci stroboscopiche blu lampeggianti come metodo per superare l'irrelevanza appresa durante un'evacuazione.

Successivamente, Nilsson, Frantzich e Saunders [14] hanno eseguito esperimenti utilizzando luci lampeggianti associate alle uscite di emergenza per verificare se potrebbe esserci un colore diverso dal blu scelto da Mc Clintock per l'orientamento in caso di emergenza.

I risultati hanno mostrato che le luci lampeggianti aumentano l'uso e la visibilità delle uscite di emergenza e che il verde è il colore adatto per questi dispositivi. Infatti, negli

esperimenti, il verde è stato valutato come il colore con più associazione positiva in caso di emergenza. Secondo Wickens and Hollands, le persone associano il verde alla sicurezza e il rosso al pericolo.

Quando si fa riferimento ai colori da una prospettiva culturale, si può notare che, in alcuni casi, lo stesso colore potrebbe avere significati diversi per le persone a seconda della regione del mondo in cui si trovano. Nel suo studio sulla metafora del "rosso" nella cultura cinese, Wong [15] esalta il grande significato e l'importanza di questo colore, che è profondamente radicato nella storia e nelle tradizioni della Cina. Per la popolazione cinese, infatti, il rosso è legato alla vita, alla felicità, al bene, alla fortuna e gioia. D'altra parte, le culture occidentali considerano il rosso principalmente come sinonimo di pericolo o precauzione e associare il verde alla sicurezza e al permesso.

#### 3.2. Guida alla progettazione e all'uso della segnaletica in base alla legislazione e agli standard

L'ampio riconoscimento dell'importanza della segnaletica di sicurezza negli edifici e negli spazi di lavoro ha visto lo sviluppo e la pubblicazione di una serie di normative e standard nazionali e internazionali dalla fine degli anni '70.

L'obiettivo di questi ultimi è garantire la trasmissione delle informazioni fornite dalla segnaletica di sicurezza, in maniera tale che siano facilmente comprensibili, inoltre, è rendere obbligatorio l'utilizzo della segnaletica di sicurezza in tutti quei luoghi in cui è identificato un rischio.

##### 3.2.1. Standard ISO e Detective CEE

Con la continua crescita dei viaggi internazionali, del commercio e della mobilità della manodopera, c'è stata una grande necessità di un metodo di comunicazione universale per trasmettere informazioni sulla sicurezza.

Alla luce di questa esigenza, la International Standards Organization (ISO) ha pubblicato il primo standard internazionale per la segnaletica di salute e sicurezza, ISO 3864: 1984 (colori di sicurezza e segnaletica di sicurezza), nel 1984, nel tentativo di creare un linguaggio di segnaletica universale. La ISO 3864 è stata rivista tecnicamente più tardi nel 2002 e ancora nel 2011.

Una serie di altri standard più specifici sono stati pubblicati dalla ISO, tra cui ISO 6309: 1987 “Protezione antincendio-Segnaletica di sicurezza” (ritirata ufficialmente senza sostituzione nel 2011), ISO 7010: 2003 “Simboli grafici - Colori di sicurezza e segnaletica di sicurezza - Segnaletica di sicurezza utilizzata nei luoghi di lavoro e nelle aree pubbliche” revisionata nel 2011 e successivamente nel 2019.

Infine, la ISO 16069: 2004 “Simboli grafici - Segnaletica di sicurezza - Sistemi di guida alla sicurezza” la cui ultima revisione risale al 2017.

Questi standard internazionali sono stati ampiamente adottati e convertiti in standard nazionali da molti paesi.

#### 3.2.2. Legislazione e standard americani

Negli Stati Uniti, secondo il manuale NFPA 1 Fire Code Handbook le porte di uscita “devono essere contrassegnate da un cartello approvato che è facilmente visibile da qualsiasi direzione di uscita di accesso”. Il rosso è l'obbligatorio colore per legge in molti stati degli Stati Uniti ed è considerato il colore tradizionale per l'emergenza. Tuttavia, il codice NFPA non prevede un colore specifico per i segnali di uscita perché a seconda della situazione, è possibile utilizzare il rosso o il verde.

In casi specifici, è persino possibile utilizzare un colore diverso da questi due a seconda della visibilità fornita. I pittogrammi sono consentiti con l'approvazione dell'autorità competente, altrimenti deve essere utilizzato il segnale con la scritta EXIT.



*Figura 10 - Cartello di emergenza EXIT*

*(<https://www.qrfs.com/blog/64-emergency-exit-signs-regulations-and-types/>)*

Per quanto riguarda il Canada, ha deciso di passare all'International Fire Code (IFC) dal 2015 sostituendo il segnale con la scritta EXIT in rosso con l'omino bianco che corre su sfondo verde, per nuove costruzioni e grandi ristrutturazioni.

La principale differenza sulla segnaletica di uscita rispetto alla maggior parte dei paesi del mondo è che gli Stati Uniti e il Canada utilizzano ancora il segnale rosso dell'uscita di emergenza con la legenda EXIT e/o SORTIE (nel caso della parte francofona del Canada) invece di utilizzare il segnale di uscita verde con il pittogramma dell'uomo che corre.

#### 3.2.3. Legislazione e standard britannici

Prima dell'introduzione degli standard internazionali e della direttiva europea, i segnali di uscita utilizzati nel Regno Unito erano principalmente segnali di testo e le frecce direzionali venivano aggiunte ai segnali di testo solo dove era ritenuto necessario.

In risposta alla Direttiva 92/58 / CEE, il Dirigente per la salute e la sicurezza del Regno Unito ha prodotto un'implementazione legale, il Regolamento per la salute e la sicurezza (segnaletica e segnali di sicurezza) del 1996 (SI n. 341), per regolamentare la progettazione e l'uso dei segnali di sicurezza.

I regolamenti suggerivano un nuovo design, colore, forma della segnaletica di sicurezza e hanno cessato l'uso di segnali di uscita di solo testo entro la fine del 1998. Nel frattempo, la British Standards Institution (BSI) ha pubblicato la serie di standard BS 5499, che sono per lo più allineati con ISO 3864: 1984 e ISO 6309: 1987.



*Figura 11 - Cartello di emergenza conforme alla BS 5499  
(<https://www.firesafe.org.uk/fire-exit-signs/>)*

Secondo il Regno Unito, infatti, i segnali di sicurezza delle vie di fuga dovrebbero essere normalmente costituiti da tre elementi: il simbolo grafico riconosciuto a livello

internazionale per le uscite di emergenza, un testo supplementare (Exit o Fire exit) e una freccia direzionale.

Sebbene questi cartelli conformi alla BS 5499 fossero diversi nei dettagli da quelli specificati nella direttiva 92/58, non sono stati modificati, in quanto seguono lo stesso schema di base e quindi sono stati considerati conformi anche alla direttiva.

Tuttavia, questi standard ISO e la direttiva 92/58 / CEE suggerivano diversi modelli di simboli grafici per i segnali di uscita e i segnali di percorso di uscita.

Di conseguenza, ci sono attualmente due tipi di segnali di uscita utilizzati nel Regno Unito. Sebbene entrambi i tipi siano accettati dagli attuali regolamenti in materia di salute e sicurezza del 1996, i segnali di uscita conformi agli standard BS sono approvati dagli standard ISO, ottenendo così potenzialmente una migliore comprensibilità.

#### 3.2.4. Legislazione e standard italiani

Il 2 febbraio 2017 è entrata in vigore la norma **UNI EN ISO 7010:2017** (sostituita successivamente con la UNI EN ISO 7010:2020, in cui sono stati aggiunti nuovi cartelli/pittogrammi), che contiene una raccolta di simboli riconosciuti universalmente e prescrive i segnali di sicurezza da utilizzare nella protezione dal fuoco, per l'informazione sui pericoli alla salute e nelle vie di fuga in caso di emergenza.

La forma e il colore di ogni segnale di sicurezza sono conformi alla norma ISO 3864-1 e la progettazione dei segni grafici è conforme alla norma ISO 3864-3.

La norma specifica che le dimensioni dei segnali di sicurezza possono essere modificate a seconda delle esigenze di applicazione.

Il Codice di Prevenzione Incendi nel paragrafo S.4.5.9 "Segnaletica d'esodo ed orientamento" rimanda alla normativa UNI EN ISO 7010 per la corretta identificazione direzionale della segnaletica.

### 3.3. Tipologie di segnaletica

La segnaletica fa parte del sistema di emergenza che si prende cura della necessità di identificare un percorso sicuro per l'esodo degli occupanti della struttura al momento dell'emergenza, una volta che il sistema di allarme antincendio è stato attivato e gli allarmi acustici e/o visivi avvisano gli occupanti della presenza di un'emergenza in corso. È possibile effettuare una divisione per le tipologie di segnaletica: statica, attiva e dinamica.

#### 3.3.1. Segnaletica statica

La segnaletica statica consiste principalmente in pittogrammi collocati in determinate posizioni: vicino ad un'uscita di sicurezza, lungo le vie di esodo ecc.

I requisiti per questo tipo di segnaletica vengono descritti nella norma UNI EN ISO 7010 descritta precedentemente.



Figura 12 - Segnaletica statica standard  
(<https://meccanicatecnica.altervista.org/la-protezione-passiva-dallincendio/>)

Questa segnaletica, nella maggior parte dei casi viene posizionata in alto sul muro o vicino al soffitto. Viene definita statica in quanto, è fissata in modo permanente indifferentemente dalla presenza o meno di un'emergenza.

Una tecnica nota per la segnaletica statica è la **fotoluminescenza**, che utilizza la luminescenza del fosforo e permette di illuminarsi nell'oscurità. Le sorgenti luminose presenti nello spazio vengono utilizzate per energizzare lo stato di fosforescente del materiale. Per questo motivo, è necessario che nelle zone in cui si trova la segnaletica fotoluminescente ci sia un livello di luce sufficientemente intenso da consentire di ricaricarsi e immagazzinare le particelle luminose.

I componenti che utilizzano il principio della fotoluminescenza forniscono un colore giallo/verde luce che dura dai 30 ai 60 minuti.

Il vantaggio di questo sistema è che non utilizza corrente elettrica, perciò anche in caso di interruzione della stessa, continuerebbero a funzionare. Inoltre, sono di facile installazione ed economiche.

La ricerca ha dimostrato che la fotoluminescenza ha un effetto positivo sulla ricerca del percorso di una persona in un edificio. Il lato negativo è che la fotoluminescenza non è così brillante come l'illuminazione comune.



*Figura 13 - Segnaletica statica fotoluminescente*  
(<https://www.fotoluminescente.it/70-cartelli-fotoluminescenti-uscita-di-emergenza>)

L'altra modalità di utilizzo è la **segnaletica retroilluminata**. Questa tipologia presenta numerosi vantaggi: permette di abbandonare l'edificio in totale sicurezza quando viene a mancare la corrente generale e assicura orientamento e condizioni visive sufficienti nelle vie di esodo.



*Figura 14 - Segnaletica statica retroilluminata*  
(<https://www.architutti.it/lo-spazio-calm/>)

## 3.3.2. Segnaletica attiva

La segnaletica attiva in questo momento non è richiesta dalla legge europea. Gli sviluppi in questo campo riguardano iniziative private per migliorare il processo di evacuazione in emergenza.

La caratteristica principale della segnaletica attiva è che si attiva al momento dell'emergenza, con lo scopo di avvisare gli occupanti, cioè per attirare l'attenzione al momento giusto e in modo chiaro segnalare la via di fuga. Quando si considera la segnaletica attiva, viene fatta una distinzione tra il suono e la luce.



Figura 15 - Esempio di segnaletica attiva

La ricerca mostra che le persone prendono un allarme più seriamente se dopo il primo allarme un **messaggio vocale** viene trasmesso mediante il sistema di altoparlanti. Il messaggio trasmesso ingrandisce la gravità della situazione.

Per quanto riguarda la luce si accende solo in una situazione di emergenza quando viene attivata da un sistema di emergenza nell'edificio. La ricerca mostra che una maggioranza significativa (da 80 a 90%) degli occupanti riconosce che le luci lampeggianti li hanno aiutati a fare una rapida decisione quando si cerca di evacuare. La segnaletica attiva migliora l'efficacia dei segnali di uscita, perché il percorso si nota molto più facilmente.

Inoltre, se la segnaletica attiva è combinata con la segnaletica a basso livello (ad esempio segnaletica a pavimento), si potrebbe avere un risultato migliore ma al momento non ci sono ricerche disponibili che combinando la segnaletica attiva all'illuminazione a basso livello. l'aspettativa è quella del totale il risultato positivo migliorerà ulteriormente.

### 3.3.3. Segnaletica dinamica

Un sistema statico o attivo non considera la posizione della calamità, come un incendio, all'interno dell'edificio e una persona in fuga può essere condotta al pericolo. Inoltre, può accadere che le vie di fuga siano congestionate provocando ritardi nell'evacuazione.

Un sistema di segnaletica dinamica cerca di garantire che ogni occupante siano in grado di fuggire dalla propria posizione nell'edificio.

Diversi sono i fattori che hanno un'influenza sulla determinazione di una via di fuga ottimale: le caratteristiche dell'incidente, le condizioni dell'edificio e le condizioni e l'ubicazione degli occupanti. Ad esempio, la via di fuga standard può essere bloccata dal fuoco. Per ottenere un percorso di evacuazione ottimale è possibile implementare un piano di evacuazione dinamico. Parecchi metodi sono disponibili per realizzare un'evacuazione intelligente.

La differenza tra i sistemi è costituita principalmente dai sensori complementari utilizzati (complementari ai sistemi standard di rilevazione del fumo e dell'incendio) per individuare l'incendio e il modo in cui le informazioni vengono trasferite.

Al giorno d'oggi sono molteplici le applicazioni: il primo sistema è costituito da due fasi: installazione ed evacuazione. L'installazione consiste nel posizionare e configurare i sensori.

In primo luogo, utilizza sensori di fumo per dirigere il percorso di evacuazione attraverso un pannello di controllo. Questi sensori sono posizionati ad ogni bivio di corridoi e sopra ogni uscita.

La configurazione del sistema avviene automaticamente in base al movimento degli occupanti dell'edificio. In questo sistema, tutti gli occupanti devono indossare un tag RFID (Radio Frequency Identification) che comunica con i sensori. In questo modo i sensori sono in grado di misurare la distanza dai sensori circostanti.

Per un'evacuazione i sensori utilizzano un algoritmo di apprendimento "gradiente-discesa" per calcolare il più vicino uscita, considerando l'ubicazione degli occupanti. Il metodo più avanzato di questo è la possibilità di guidare le persone in una direzione diversa quando un'uscita è congestionata. Può anche inviare un file mappa virtuale per

i soccorritori, in modo che possano vedere la posizione di eventuali problemi; tuttavia, non è facile mappare un edificio a più piani.

L'altra soluzione è l'utilizzo di telecamere, microfoni e sensori fisici. Le telecamere sono utilizzate per rilevare l'incendio e presentano vantaggi rispetto ai rilevatori di fumo in quanto sono più veloci.

Se il sistema non è certo della presenza di un incendio, ha due possibilità: inviare una persona a riconoscere l'incendio o se non c'è persona attiva nei dintorni, può inviare un robot, che può utilizzare la termografia misurazioni per confermare un incendio.

L'**Active Dynamic Signage System (ADSS)** [13] è costituito da due nuovi concetti di design della segnaletica, un insieme di luci lampeggianti e correnti per evidenziare l'esistenza del cartello della via di fuga e una croce rossa lampeggiante per indicare una via d'uscita che non è più percorribile.

La serie di luci lampeggianti e correnti viene introdotta nella segnaletica standard "uomo che corre verde" per migliorare l'affidabilità della segnaletica (ovvero aumentare la rilevabilità della segnaletica) pur mantenendo la massima conformità alle normative sulla segnaletica esistente. Il sistema di segnaletica statica convenzionale viene quindi trasformato in un sistema di segnaletica dinamica (DSS), mentre le dimensioni e il formato della segnaletica e delle informazioni veicolate rimangono invariate.

La natura dinamica dei segnali (cioè il ciclo lampeggiante) si attiva solo durante una situazione di emergenza, quando l'allarme è attivato, superando il problema della irrilevanza acquisita della segnaletica.



*Figura 16 - Esempio di Active Dynamic Signage System*

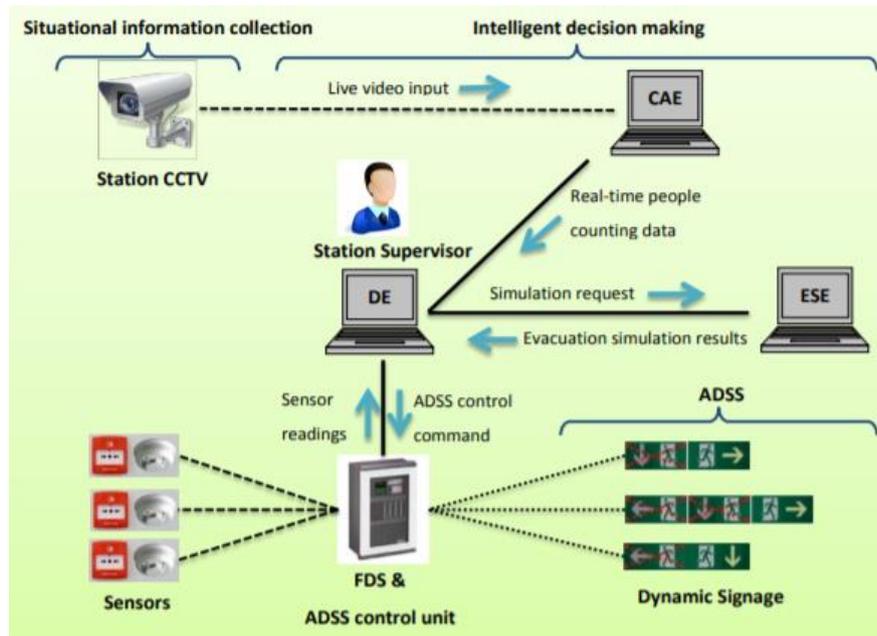


Figura 17 - Funzionamento dell'Active Dynamic Signage System

Numerose sono gli esperimenti effettuati con questa tipologia di segnaletica, i risultati delle prove sperimentali dimostrano che il 77% delle persone percepisce il segnale e il 100% continua a seguirlo. Inoltre, i risultati dimostrano che la segnaletica dinamica riduce il tempo impiegato dalle persone nella scelta della direzione di evacuazione.

Un'ulteriore tecnica è quella che utilizza gli smartphone esistenti per determinare la posizione degli utenti. Il percorso di evacuazione ottimale viene mostrato sul dispositivo dell'utente in caso di emergenza. Il sistema calcola il percorso di evacuazione più veloce per ciascuna persona.

È importante che il sistema sappia la posizione esatta dell'utente, supponendo che il sistema riconosce la mappa dei dintorni è possibile fare una foto, ad esempio, del numero della stanza per inserire queste informazioni nel sistema. Utilizzando un contapassi combinato con il normale percorso di camminata dell'utente, la posizione dell'utente è determinata esattamente. Questa informazione è trasferita al server del sistema, che calcola il percorso di evacuazione più veloce e sicuro.

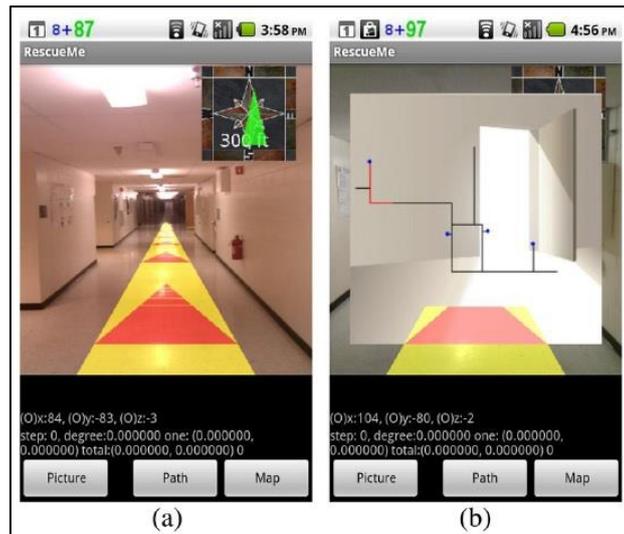


Figura 18 - Esempio di applicativo che utilizza la posizione dell'utente nell'edificio

### 3.4. Sistemi di comunicazione alternativi

I sistemi di segnaletica e di illuminazione possono essere accompagnati da altri sistemi di comunicazione alternativa:

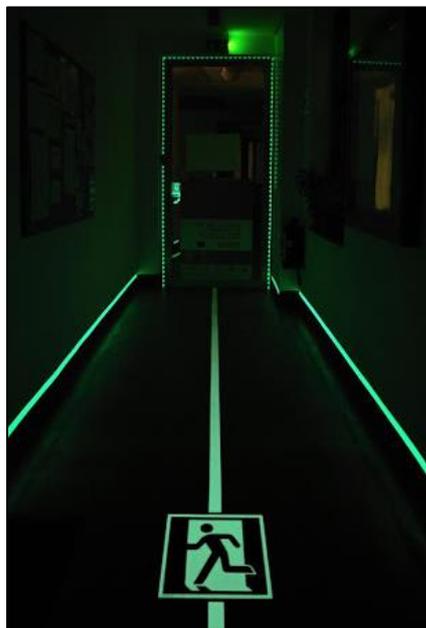
- **Differenziazione cromatica:** la creazione di interni uniformi e con poca differenziazione fisica può compromettere l'uso degli indizi per raggiungere un'uscita, per questo motivo sarebbe utile creare una differenziazione cromatica delle pareti in maniera tale da agevolare il riconoscimento delle uscite di sicurezza;



(a) condizioni comuni      (b) colore su sfondo      (c) nessun colore su sfondo

Figura 19 - Esempio di differenziazione cromatica di una porta

- Realizzazione di pavimentazioni e superfici verticali in sono presenti **riferimenti tattili**;
- **Cartelli fotoluminescenti** antiscivolo applicabili a **pavimento o parete**;



*Figura 20 - Nastro luminescente a pavimento  
(<http://antincendio-italia.it/segnali-luminescenti-per-le-vie-di-esodo/>)*

- **Suono direzionale:** vi sono interessanti studi che dimostrano come un sistema sonoro direzionali potrebbe rappresentare una valida risorsa per individuare percorsi da impiegare durante un'emergenza.

Anche l'NFPA ha considerato la risorsa di un suono direzionale proponendolo nell'ambito delle indicazioni sulla gestione dell'evacuazione quando sono coinvolte persone con disabilità.

La tecnologia del suono direzionale offre una soluzione per migliorare l'evacuazione degli occupanti in un edificio. Per aiutare gli occupanti a trovare la strada per l'uscita, l'area di rifugio o altri mezzi di uscita, le sirene direzionali integrate con il sistema di allarme antincendio potrebbero fornire informazioni aggiuntive che possono aiutare gli occupanti nella localizzazione delle vie di fuga. Le sirene direzionali utilizzano un sistema acustico a banda larga caratteristica che si distingue dai suoni udibili di campane, clacson o altoparlanti vocali. Il suono direzionale viene utilizzato in contemporanea con

il sistema di allarme standard, infatti, non sostituiscono i dispositivi richiesti dai codici;

- **Percorsi Loges:** sono dei sistemi modulari per agevolare il movimento dei disabili all'interno di una struttura.



Figura 21 - Esempio di percorso Loges

Le piastre componibili sono costituite da gomma sintetica e presentano dei rilievi a seconda dei codici necessari alle differenti segnalazioni;

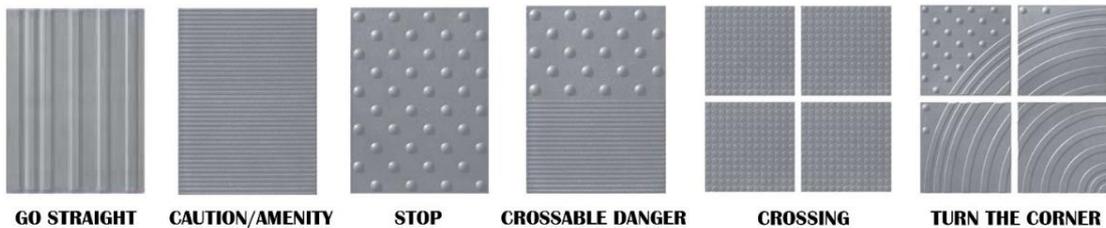


Figura 22 - Piastre componibili per percorsi Loges

([https://projectstone.it/news/?SISTEMA\\_LOGES\\_-\\_Guida\\_Tattile\\_per\\_Disabili\\_Visivi](https://projectstone.it/news/?SISTEMA_LOGES_-_Guida_Tattile_per_Disabili_Visivi))

- **Sistemi lampeggianti:** le persone con problemi di udito non possono sentire allarmi e annunci vocali che avvertono del pericolo e della necessità di evacuare l'edifici.



Figura 23 - Allarme antincendio

### 3.5. Distanza di riconoscimento della segnaletica di sicurezza

La normativa UNI EN 1838 distingue tra i segnali illuminati internamente (retroilluminati) che sono distinguibili a distanze maggiori, e i segnali illuminati esternamente, fornendo la seguente formula per determinare la massima distanza di visibilità “d”:

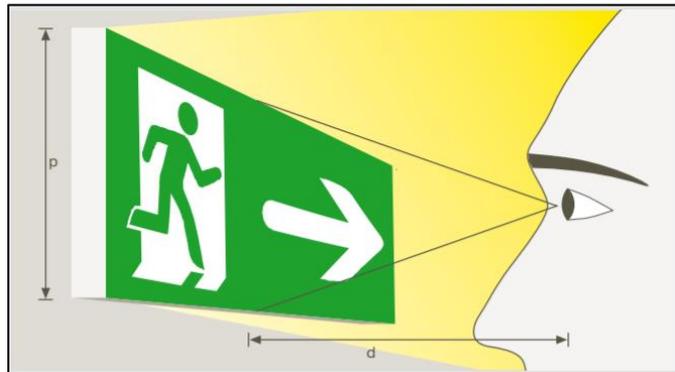


Figura 24 - Distanza di riconoscimento per la segnaletica di sicurezza secondo la norma UNI EN 1838

$$d = s \cdot p$$

Dove:

- p corrisponde all'altezza del pittogramma espresso in metri
- s è una costante (pari a 100 nel momento in cui il pittogramma è illuminato esternamente e pari a 200 quando il pittogramma è illuminato internamente)

Il Decreto Legislativo 493/96, al contrario della normativa, non prende in considerazione i segnali retroilluminati e fornisce una formula valida fino a distanze di circa 50 metri. La formula per determinare la massima distanza di riconoscibilità del cartello è la seguente:

$$L < \sqrt{A \cdot 2000}$$

Dove:

- A è la superficie del cartello espressa in metri quadri

Il Decreto Legislativo è più restrittivo della norma UNI EN 1838.

## Capitolo 4

Un uso efficace dei sistemi di segnaletica per aiutare l'orientamento degli occupanti dipende da due aspetti chiave: la qualità delle informazioni trasmesse dalla segnaletica e la diffusione delle informazioni da parte degli occupanti.

### Interazione tra occupanti e sistemi di segnaletica

#### 4.1. Disposizione della segnaletica

Il primo aspetto riguardante un utilizzo efficace dei sistemi di segnaletica (ovvero la qualità delle informazioni veicolate dalla segnaletica) si riferisce principalmente alla disposizione dei segnali, inclusi design e posizione. Il design dovrebbe garantire che le informazioni trasmesse sotto forma di testo e il simbolo grafico raggiungano un buon livello di comprensibilità tra gli utenti dell'edificio;

Quando si considera la leggibilità dei segni, si dovrebbe anche tener conto delle situazioni anormali come quella che potrebbe essere la presenza di fumo e mancanza di corrente elettrica nell'edificio.

La posizione della segnaletica dovrebbe garantire che i segnali siano posti in luoghi in cui non è possibile la vista diretta di un'uscita e possano sussistere dubbi la riguardo la sua posizione. L'indicazione direzionale dei segnali dovrebbe essere coerente con l'obiettivo primario di via di fuga, in modo che gli occupanti possano sempre raggiungere un luogo sicuro o un'uscita definitiva se seguono la segnaletica.

Quando si considera il numero di segni necessari, il progetto dovrebbe evitare due casi: troppi e troppo pochi. Il primo può causare un sovraccarico di informazioni, mettendo in difficoltà l'occupante nello scegliere la direzione da intraprendere in caso di emergenza.



*Figura 25 - Esempio di sovrabbondanza della segnaletica*

#### 4.2. La diffusione delle informazioni

Il secondo aspetto riguardante un uso efficace dei sistemi di segnaletica riguarda l'assorbimento da parte degli occupanti delle informazioni trasmesse dai segnali. Nonostante l'ampia guida sulla disposizione dei segnali, ci sono poche informazioni nella legislazione sulla sicurezza e negli standard riguardo le modalità della messa in pratica dei sistemi di segnaletica efficaci.

Pertanto, è spesso assunto dai designer, ingegneri e funzionari edili che se la segnaletica, che soddisfa i criteri di progettazione e di installazione, è presente in un edificio, gli occupanti saranno in grado di percepire e interpretare i segni, e rispettare le informazioni trasmesse.

Però, disastri precedenti, di cui abbiamo parlato precedentemente, hanno dimostrato che i sistemi di segnaletica sono sottoutilizzati.

Questo problema di sottoutilizzo della segnaletica di emergenza durante le evacuazioni ha sollevato preoccupazioni circa l'ambiguità nell'efficacia dei sistemi di segnaletica e l'esigenza di migliorare la progettazione dei sistemi di segnaletica come ausilio per il wayfinding.

Per raggiungere questo obiettivo, è necessaria una buona comprensione dell'interazione tra occupanti e segnaletica.

L'interazione tra occupanti e segnaletica può essere suddivisa in tre fasi: percepire il

segnale, interpretando le informazioni veicolate dal segno e il successivo processo decisionale. Innanzitutto, il segno deve essere visivamente percepibile dagli occupanti. Purché un segnale sia percepibile, devono essere soddisfatte due condizioni: l'occupante deve essere situato entro una certa distanza dal segnale e il segnale deve essere nel campo visivo dell'occupante.

Se le condizioni permettono di vedere il segnale, l'occupante interpreterà e agirà grazie alle informazioni fornite dalla segnaletica.

### 4.2.1. La visibilità del segnale

Dato che il fumo può essere presente durante un'evacuazione, è anche importante comprendere l'impatto del fumo sull'interazione dell'evacuazione con la segnaletica.

L'impatto del fumo sulla visibilità della segnaletica è stato ampiamente studiato [16], [17] attraverso esperimenti condotti per esaminare la soglia di oscuramento e la leggibilità del segnale in varie condizioni di prova.

I risultati ottenuti da questi esperimenti mostrano che la capacità del partecipante di discernere i segnali di uscita si deteriora con l'aumento delle concentrazioni di fumo e del livello di irritazione.

Questo avviene perché i flussi luminosi diretti dai segnali di uscita vengono diffusi dalle particelle di fumo, riducendo l'intensità. Inoltre, i flussi luminosi ambientali, diffusi dalle particelle di fumo, si sovrappongono ai flussi dei segnali di uscita, riducendo il contrasto tra la segnaletica e lo sfondo, rendendo così difficile la loro distinzione.

Oltre a questi studi sperimentali, Zhang e Rubini [17], [18] hanno analizzato il trasporto della luce visibile attraverso il fumo e hanno proposto un modello teorico per valutare la visibilità del segno nel fumo.

L'impatto del fumo sulla capacità di trovare un'uscita durante un'evacuazione è stato studiato da Paulsen [19]. I risultati mostrano che il tradizionale segnale di uscita non è riuscito a soddisfare le aspettative di efficacia nel fumo.

Esistono due approcci per stimare l'influenza del fumo sulla visibilità.

Il primo approccio, spiegato qui di seguito, utilizza la densità ottica, invece, il secondo approccio verrà spiegato nei capitoli seguenti.

Secondo McGrattan [20] la quantità più utile per la valutazione della visibilità in uno spazio è il coefficiente di estinzione della luce  $K$ .

L'intensità della luce che passa attraverso un mezzo ad una distanza  $L$  è attenuata secondo la seguente legge:

$$\frac{I}{I_0} = e^{-KL}$$

Dove:

- $I_0$  è l'intensità della luce iniziale
- $I$  è l'intensità dopo
- $L$  è la distanza

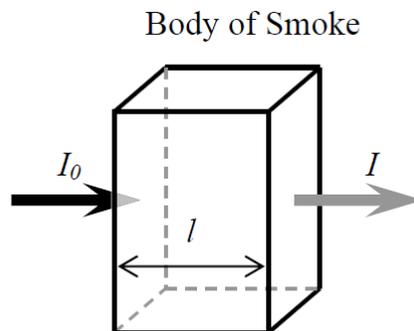
Questa legge è chiamata legge di Beer-Lambert, definisce l'assorbimento della luce di un materiale a seconda delle sue proprietà.

La densità ottica si ottiene misurando l'attenuazione di un fascio di luce che passa attraverso il fumo, utilizzando questa legge:

$$D = \log_{10} \frac{I_0}{I}$$

La densità ottica per unità di distanza è calcolata come segue:

$$D_0 = \frac{D}{L} = \frac{1}{L} \cdot \log_{10} \left( \frac{I_0}{I} \right)$$



Il coefficiente di estinzione della luce  $K$  viene calcolato come segue:

$$K = \frac{1}{L} \cdot \ln \frac{I_0}{I} = 2,3 \cdot D_0$$

Poiché la densità ottica è anche correlata alla misurazione della densità del fumo, questo approccio è in grado di stimare la diminuzione dell'intensità della luce dopo che questa ha attraversato una certa distanza attraverso il fumo.

Questo approccio affronta la prima interferenza, ovvero l'influenza delle concentrazioni fumo e la distanza che la luce attraversa, ma non tiene conto della seconda forma di interferenza, ovvero la segnaletica.

Sulla base di questo sono stati effettuati degli esperimenti: un corridoio lungo 20 metri è stato riempito di fumo corrispondente alla fase iniziale di un incendio. Un fumo bianco irritante veniva prodotto bruciando presepi in legno e un fumo meno irritante è stato prodotto bruciando cherosene. I soggetti coinvolti nell'esperimento avevano il compito di camminare lungo il corridoio da un'estremità all'altra e registrare i luoghi in cui avevano visto un cartello e leggere le parole scritte.

La visibilità del fumo irritante diminuisce bruscamente quando la densità del fumo supera un certo livello. Nel fumo denso irritante, i soggetti non potevano tenere gli occhi aperti per molto tempo, inoltre, le lacrime non permettevano la vista delle parole sulla segnaletica.

L'irritazione al fumo riduce inoltre la visibilità delle persone stesse causando il panico e diminuendo la velocità di evacuazione.

Dal grafico sottostante è possibile notare come la velocità dell'occupante sia strettamente correlata alla densità ottica:

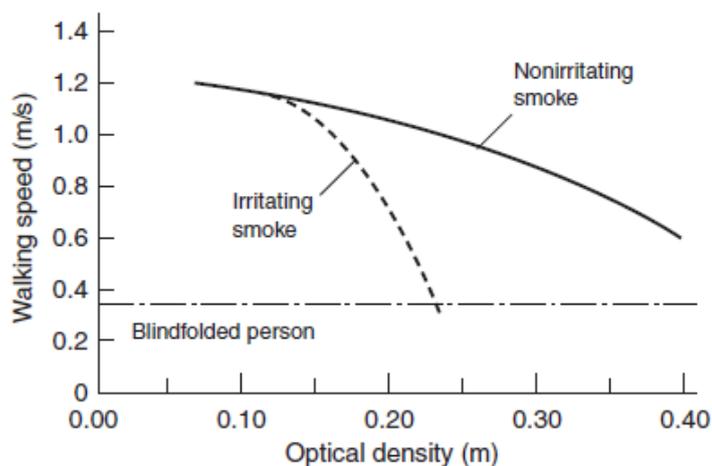


Figura 26 - Relazione tra visibilità attraverso il fumo e la velocità di camminata

### 4.3. L'orientamento in caso di emergenza

In una situazione di emergenza, l'orientamento dell'individuo può essere reso difficile dall'inadeguatezza della segnaletica presente.

Sulla base di questo, occorre effettuare una distinzione tra i termini "orientamento" e "wayfinding" [10], il primo viene utilizzato per identificare la determinazione della posizione e della direzione dal punto di vista geografico o geometrico, quindi, è il processo con il quale l'occupante è in grado di dire dove si trova. Invece, il wayfinding è ciò che permette di comprendere alle persone com'è possibile spostarsi da un luogo all'altro., in relazione ai meccanismi di percezione sensoriale dell'ambiente e alle condizioni qualitative dell'ambiente che le circonda.

Il termine wayfinding è stato introdotto dall'urbanista Kevin Lynch nel 1960 nel testo "The Image of the city".

Il sistema di segnaletica cerca di affrontare il tema del wayfinding da due prospettive differenti: in primo luogo, gli edifici complessi potrebbero non avere un layout semplice ed intuitivo, dall'altro lato le informazioni disponibili potrebbero non essere sufficienti per consentire agli occupanti di effettuare una rapida decisione quando è necessario un'evacuazione. Per esempio, l'edificio potrebbe non avere un accesso diretto alle uscite in alcuni punti oppure ci potrebbero essere scelte multiple di percorsi per raggiungere le uscite. In entrambi i casi, però, sono necessarie delle informazioni aggiuntive per aiutare gli occupanti a prendere delle decisioni.

In secondo luogo, spesso gli occupanti tendono a lasciare l'edificio utilizzando lo stesso percorso che hanno utilizzato per raggiungere quella posizione. Questo comportamento può ridurre l'efficienza dei sistemi di esodo in quanto alcune uscite potrebbero non essere utilizzate e altre invece, potrebbero essere sovraffollate.

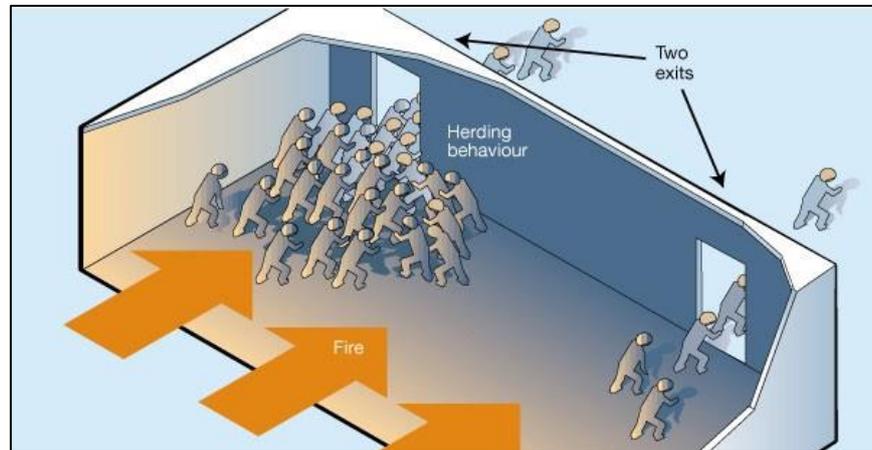


Figura 27 - Esempio di uscite sovraffollate e non utilizzate

Per questo motivo, è necessario avere un sistema di segnaletica capace di fornire informazioni di supporto agli occupanti e incoraggiarli ad utilizzare tutti i mezzi di fuga disponibili.

## Capitolo 5

In questo capitolo, vengono analizzati i campi di applicazione della Virtual Reality e le strumentazioni.

### L'ambiente virtuale come strumento di visualizzazione

Il termine “realtà virtuale” (VR, Virtual Reality), fu coniato nel 1989 da Jaron Lanier, un informatico statunitense. La realtà virtuale (VR) è diventata un approccio popolare per studiare il comportamento umano durante le emergenze.

Gli ambienti virtuali forniscono il massimo controllo sperimentale, sono facili da replicare e consentono uno studio sicuro del comportamento degli occupanti in scenari che altrimenti sarebbero troppo pericolosi. Infatti, sono state effettuate simulazioni sempre più realistiche e i progressi tecnologici offrono nuove opportunità per questo metodo relativamente giovane.

#### 5.1. Differenza tra realtà virtuale, realtà aumentata e mista

Realtà virtuale, realtà aumentata e realtà mista. Per quanto possano essere simili questi termini, è importante sottolineare che esistono una serie di differenze per ognuno, con diversi scenari applicativi.

Con il termine **realtà virtuale** (VR) si fa riferimento ad un ambiente creato mediante il computer, nel quale l'utente può essere immerso parzialmente o totalmente. Per cui, la realtà virtuale è una simulazione della realtà effettiva.

La realtà virtuale si divide principalmente in:

- Realtà virtuale **immersiva** in cui l'utente ha la percezione di essere interamente immerso nell'ambiente tridimensionale; questo è possibile mediante l'utilizzo di alcuni strumenti di cui parleremo successivamente come il visore VR.
- Realtà virtuale **non immersiva** o parziale: in questo caso, l'utente non utilizza visori VR ma l'interazione avviene mediante dei joystick o mediante la tastiera di un computer. Dunque, lo schermo funge da finestra per il mondo tridimensionale.

Con il termine **realtà aumentata** (AR) si indica una tecnologia capace di sovrapporre il mondo digitale al mondo reale, arricchendo l'esperienza sensoriale mediante strumenti tecnologici. Questo avviene mediante l'utilizzo di dispositivi mobili o di computer dotato di webcam oppure mediante l'utilizzo di particolari occhiali che proiettano direttamente sulla retina. Lo scopo è aggiungere contenuti multimediali, in tempo reale, alla realtà esistente.

Con la terminologia **realtà mista** (MR), come suggerisce il nome stesso, identifichiamo una combinazione unica tra la Realtà Virtuale e la Realtà Aumentata. Mediante questa tecnologia è possibile osservare il mondo reale, ottenendo maggiori informazioni mediante l'utilizzo della realtà aumentata ma anche di interagire con oggetti ricreati mediante la realtà virtuale.

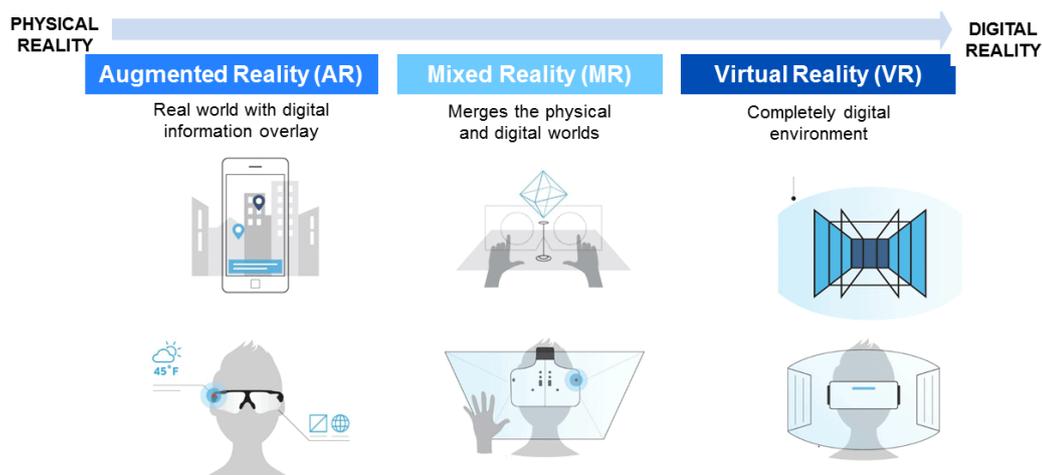


Figura 28 - Classificazione realtà

(<https://medium.com/@shivsoni377/the-difference-between-augmented-reality-virtual-reality-and-mixed-reality-a028bdd81f9d>)

## 5.2. Tecnologie disponibili

Per poter usufruire dell'esperienza di completa immersione all'interno di una realtà virtuale sono disponibili diverse apparecchiature. Al giorno d'oggi esistono differenti tipologie di tecnologie per fruire della realtà virtuale, partendo dallo smartphone o dal computer, ma il sistema che permette una realtà più immersiva è il visore, che permette una visione stereoscopica a 360 gradi inserendolo sulla testa.

L'**Oculus** è un dispositivo molto semplice, esso contiene:

- HMD Oculus con cuffie integrate
- Sensore Oculus
- Oculus Controller
- Controller Xbox One Wireless
- Cavi di collegamento dell'HMD

Il visore è costituito da lenti AMOLED con una risoluzione 1200x1080 ciascuna, che permettono di creare delle esperienze in VR molto fluide ed immersive a discapito però di grandi necessità computazionali.



Figura 29 - Oculus Rift

(<https://www.library.ucsf.edu/news/oculus-rift-s-improved-virtual-reality/>)

L'**HTC Vive** è un dispositivo della Valve Corporation, a differenza degli altri dispositivi, esso permette grazie ad una tecnologia chiamata “room scale” di muoversi liberamente nello spazio 3D. Inoltre, permette di avere un fedele tracciamento delle mani dell'utente per mezzo di due controller ergonomici venduti insieme al visore. In questo dispositivo, il visore è costituito da due lenti di Fresnel AMOLED con una risoluzione di 1080x1200 per lente.

---

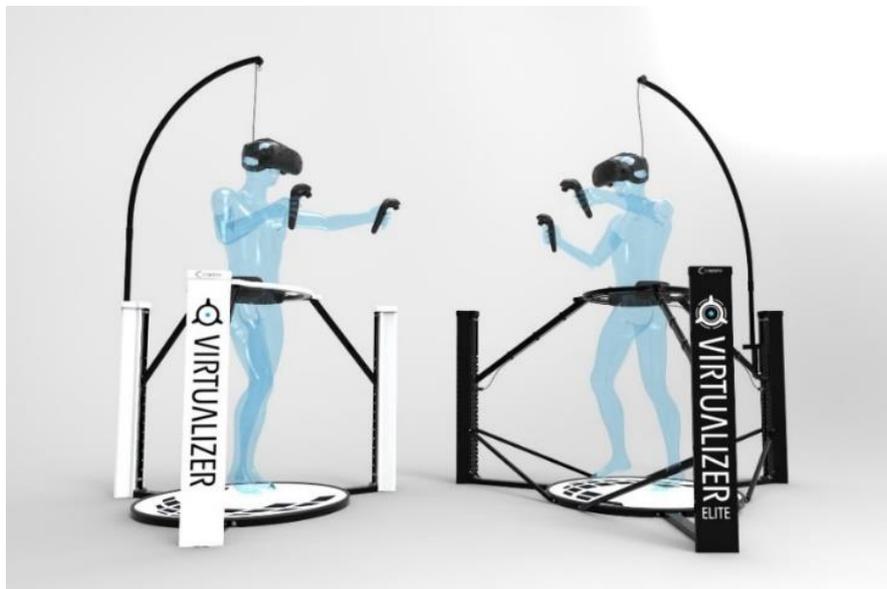


*Figura 30 - HTC Vive*

*(<https://www.techrepublic.com/article/htc-vive-business-edition-targets-enterprise-with-1200-vr-kit/>)*

Il **Cyberith Virtualizer** è un'idea originaria di Tuncay Cakmak e Holger Hager, fisici e ricercatori dell'Università di Vienna.

Questa nuova tecnologia garantisce il raggiungimento di una totale esperienza immersiva, grazie all'integrazione di complessi sensori che permettono di rilevare l'esatta posizione del corpo quando è accovacciato e quando si salta.



*Figura 31 - Cyberith Virtualizer*

*(<https://www.cyberith.com/virtualizer-and-elite/>)*

Un'ulteriore tecnologia utilizzata è il **CAVE** costituito da una stanza a forma di cubo e dai proiettori video diretti verso tre o sei facce del cubo. L'ambiente virtuale viene ricreato proiettando le immagini sulle pareti e sul soffitto della stanza. Il vantaggio di questa tecnologia consiste nel fatto che l'utilizzatore indossa solamente un paio di cuffie audio e si limita a camminare all'interno della stanza interagendo con gli oggetti virtuali che vengono proiettati mediante dei controller che tiene nelle mani. È possibile indossare comunque occhiali LCD per ottenere una vista stereoscopica e una maggiore percezione di profondità. Inoltre, questa tecnologia permette a più utenti di condividere la stessa esperienza di Virtual Reality contemporaneamente.



*Figura 32 - Il Cave*

### 5.3. Applicazioni

Al giorno d'oggi sono molteplici i campi di applicazione della Virtual Reality come ad esempio l'architettura, la medicina, l'arte e lo sport, in pratica in qualsiasi campo, dove fare qualcosa nella realtà sarebbe pericoloso, troppo costoso o poco pratico:

- Settore medico – sanitario: aumenta la precisione durante le operazioni chirurgiche e riduce gli errori medici;
- Settore turistico: permette di visitare luoghi e strutture, fornendo informazioni storiche e strutturali dei monumenti;

- **Ambito militare:** viene utilizzato generalmente per l'addestramento;

Infatti, uno dei campi principali della realtà virtuale è l'addestramento. Per quanto riguarda la sicurezza antincendio, combattere gli incendi è pericoloso e di conseguenza un buon allenamento e una buona formazione sono molto importanti. Per far fronte a questo e migliorare l'allenamento dei Vigili del Fuoco, numerosi sono gli applicativi sviluppati in questo campo.

### 5.3.1. Applicazione all'ingegneria della sicurezza antincendio

Nell'ambito della sicurezza antincendio, la realtà virtuale è stata applicata in molti modi. Per esempio:

- Nella formazione del pubblico in generale nelle situazioni di evacuazione e soccorso nelle gallerie stradali e negli edifici universitari;
- Nel migliorare le capacità di sicurezza antincendio dei bambini e dei loro comportamenti di evacuazione in edifici residenziali;
- Nella formazione dei vigili del fuoco in generale
- Per una selezione ottimale del percorso di salvataggio;
- Nella stima del comportamento di persone in pericolo e per capire il comportamento umano nel fuoco

Attualmente i software utilizzati per la modellazione dell'esodo e dell'incendio non permettono un'immersione totale dell'utente all'interno del modello. Infatti, non risulta possibile un'esplorazione dell'edificio a 360° ma è possibile solamente creare un percorso guidato oppure immedesimarsi in uno degli occupanti della simulazione stessa.

### 5.3.2. Applicazione VR ad un caso studio

Dopo un'analisi accurata sulle criticità al giorno d'oggi presenti sulle tecnologie disponibili nel settore della prevenzione incendi si è pensato di proporre un metodo per la visualizzazione dei risultati di FDS nella realtà virtuale ovvero l'esportazione indiretta. L'esportazione è indiretta in quanto i software utilizzati per le simulazioni di esodo e di incendio non permettono l'esportazione dei risultati in formati di files leggibili dal software finale, ma è necessario passare attraverso altri software intermedi.

Nei capitoli seguenti viene analizzato il caso studio dal punto di vista della modellazione BIM e della modellazione FDS. Successivamente viene spiegato dettagliatamente il processo di esportazione ed infine, con i risultati ottenuti da quest'ultimo passaggio, si è creato un applicativo per la visualizzazione in virtual reality.

Questo applicativo ha il fine di supportare e assistere il professionista antincendio durante la fase di verifica delle misure di sicurezza, come ad esempio la scelta dei sistemi di segnaletica e la loro posizione.

L'applicativo utilizza principalmente la realtà virtuale non immersiva, adoperando lo schermo come mezzo tra realtà e modello tridimensionale. Ma è facilmente adattabile per l'utilizzo della realtà virtuale immersiva, utilizzando le tecnologie viste precedentemente.

## Capitolo 6

In questo capitolo viene descritto il caso preso in esame in questo lavoro di tesi. Il caso studio riguarda La Centrale della Lavazza che si inserisce nel contesto urbano del quartiere Aurora a Torino.

### Caso studio: La centrale della Lavazza – attività eventi di pubblico spettacolo

#### 6.1. Inquadramento generale



Figura 33 - Inquadramento della Centrale della Lavazza

L'edificio è una ex centrale elettrica realizzata nel 1897 dall'ingegnere torinese Ermenegildo Perini per la Società Anonima Elettrica Alta Italia.

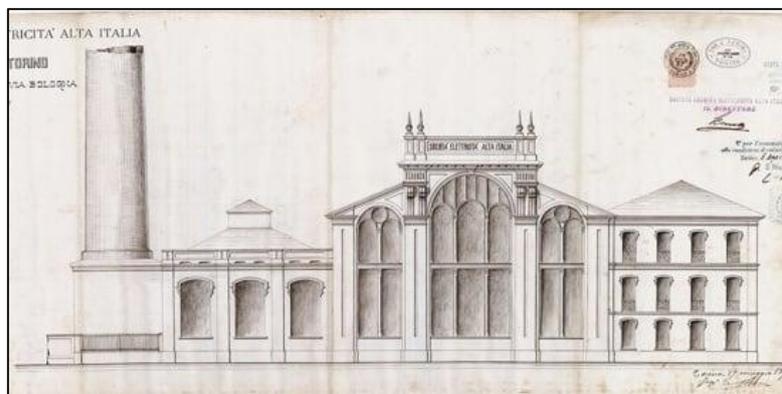


Figura 34 - Progetto dell'Ingegnere Perini per la Società Alta Italia  
(<https://www.lavazza.it/it/museo-lavazza/scopri-nuvola/la-centrale.html>)

Al giorno d'oggi, questo edificio prende una nuova forma grazie alla riqualificazione e allo sviluppo del quartiere Aurora in cui si colloca dell'architetto Cino Zucchi. L'edificio è costituito da diversi spazi per un totale di 4500 mq ed è in grado di ospitare eventi culturali, di intrattenimento e conferenze.

#### 6.1.1. Lo spazio centrale



*Figura 35 - Lo spazio centrale*

*(<https://www.lavazza.it/it/museo-lavazza/scopri-nuvola/la-centrale.html>)*

Lo spazio centrale è costituito da elementi dell'edificio originale che sono stati valorizzati e ristrutturati come le ampie vetrate liberty che caratterizzano la facciata nord-ovest dell'edificio e il carroponete. Questi elementi creano un legame tra passato e presente. Lo spazio è flessibile e si estende per circa 1400 mq, inoltre, è uno spazio flessibile in cui palchi e attrezzature possono essere disposti sia in modo longitudinale che trasversale.

### 6.1.2. La terrazza indoor

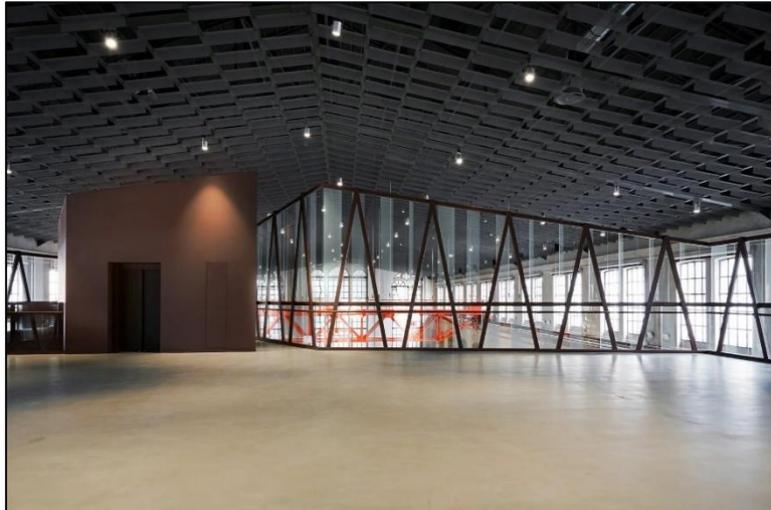


Figura 36 - La terrazza Indoor (<https://www.lavazza.it/it/museo-lavazza/scopri-nuvola/la-centrale.html>)

La terrazza posta al secondo piano dell'edificio, affaccia sul salone centrale e permette una vista panoramica. Può ospitare all'incirca 150 persone e si estende per 380 mq. Utile a diversi usi, può accogliere eventi e incontri, essere allestita come lounge per gli ospiti o come sala stampa, per coffee break, laboratori, sessioni di yoga e molto altro.

### 6.1.3. La sala conferenze



Figura 37 - La sala conferenze (<https://www.lavazza.it/it/museo-lavazza/scopri-nuvola/la-centrale.html>)

Al primo piano della Centrale, la sala conferenze: un'area modulabile di 210 mq complessivi, che può essere utilizzata per intero, ospitando 200 persone, oppure suddivisa in due spazi (da 80 e 120 posti a sedere).

È l'ideale per incontri, convegni o workshop e dotata di regia audio-video residente.

## 6.2. Modellazione dell'edificio con metodologia BIM

Ai fini dell'obiettivo di questa tesi, ovvero avere a disposizione un modello tridimensionale per poter studiare il moto dei fumi e l'evacuazione degli occupanti, è stato necessario modellare l'edificio dal punto di vista architettonico.

Il software utilizzato per la modellazione architettonica è Revit, un programma di Autodesk basato sul Building Information Modelling (BIM)

di cui abbiamo parlato precedentemente.

L'edificio è stato modellato prestando notevole attenzione ai dettagli, ai materiali e alle dimensioni. È stato necessario modellare l'arredo e gli ingombri, in quanto, sono fondamentali per l'evacuazione degli occupanti quindi per la modellazione dell'esodo. Anche il copro centrale contenente il vano scale per raggiungere le sale conferenze poste al piano primo e la terrazza indoor posta al secondo piano, è stato modellato con particolare attenzione. Inoltre, sono stati modellati i sistemi di segnaletica ed eventuali estintori, idranti e naspi.

Di seguito vengono riportate delle immagini delle fasi significative della modellazione BIM:

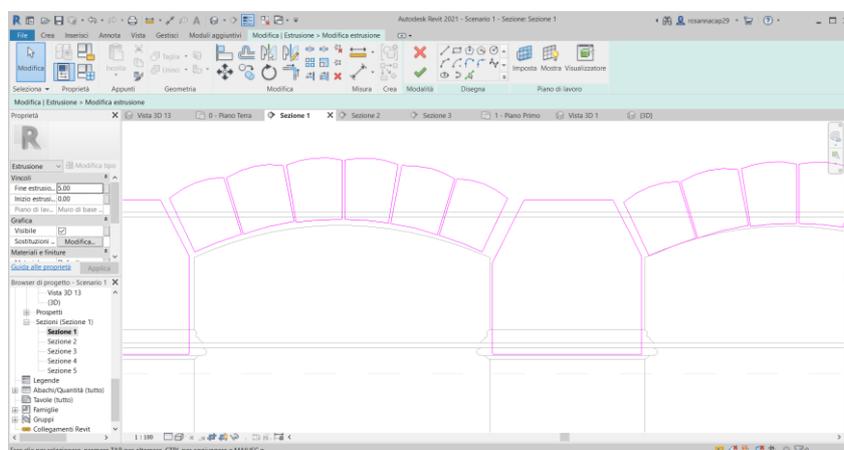


Figura 38 - Creazione dei dettagli costruttivi mediante il comando Estrusione

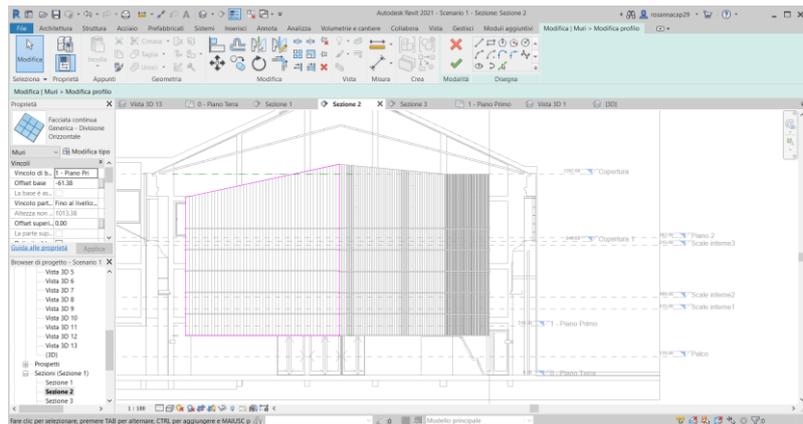


Figura 39 - Creazione della facciata continua del corpo scale

Infine, di seguito vengono riportati alcuni render effettuati al termine della modellazione architettonica.



Figura 40 - Rendering interno mediante Autodesk Revit



Figura 41 - Rendering esterno mediante Autodesk Revit

### 6.3. Modellazioni di Fire Safety Engineering

Successivamente, modellato l'edificio mediante Autodesk Revit, si è passati alla modellazione dell'incendio e alla modellazione dell'esodo.

#### 6.3.1. Configurazioni

Come abbiamo visto nei paragrafi precedenti, La Centrale presenta differenti spazi, questo permette di gestire numerose configurazioni di allestimento.

Per ciascuna configurazione, è necessario effettuare una differente simulazione di esodo e un'attenta simulazione di incendio, in quanto le vie di fuga si differenziano in ognuna di essa.

Nel caso qui presente, sono state scelte solo due configurazioni. La scelta è ricaduta sulle due configurazioni che presentano una destinazione d'uso completamente diversa tra loro, in maniera tale da analizzare diversi tipi di scenari.

La **prima configurazione**, con destinazione d'uso **sala conferenze**, prevede:

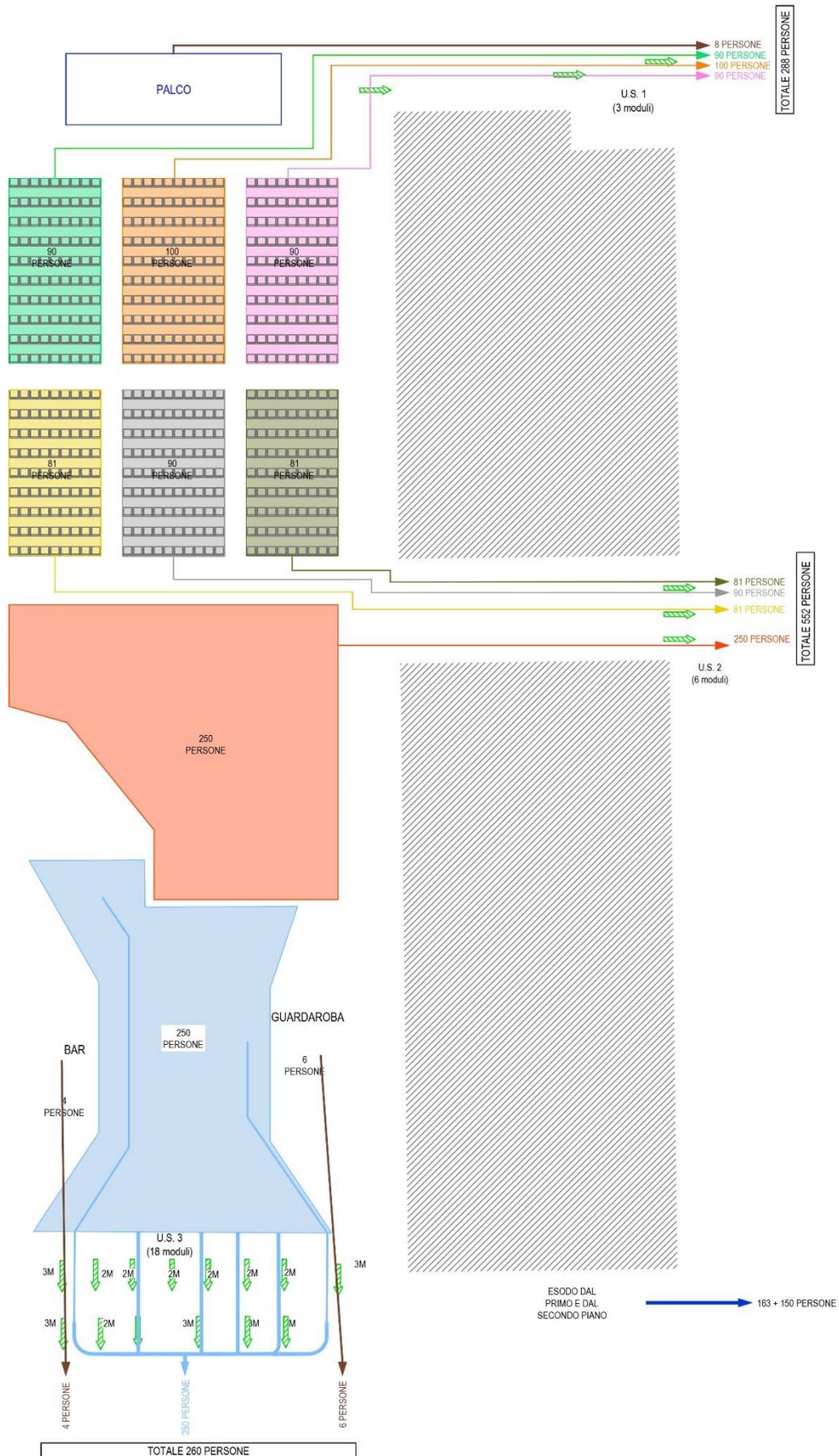
*Tabella 9 - Divisione della prima configurazione*

CONFIGURAZIONE 1: Sala conferenze	
Piano terra	532 posti in piedi suddivisi in 6 settori
	500 posti in piedi
	10 addetti bar + guardargoba
Piano primo	163 posti a sedere
Piano secondo	150 posti in piedi

Il piano terra ospita un auditorium e un palco predisposto per una conferenza, come il secondo piano ed infine, la terrazza indoor posta al secondo piano, è adibita per un'esposizione.

Il piano terra presenta 27 moduli d'uscita, il primo piano presenta 7 moduli d'uscita ed infine, il secondo piano ne presenta 4.

## 6. Caso studio: La centrale della Lavazza



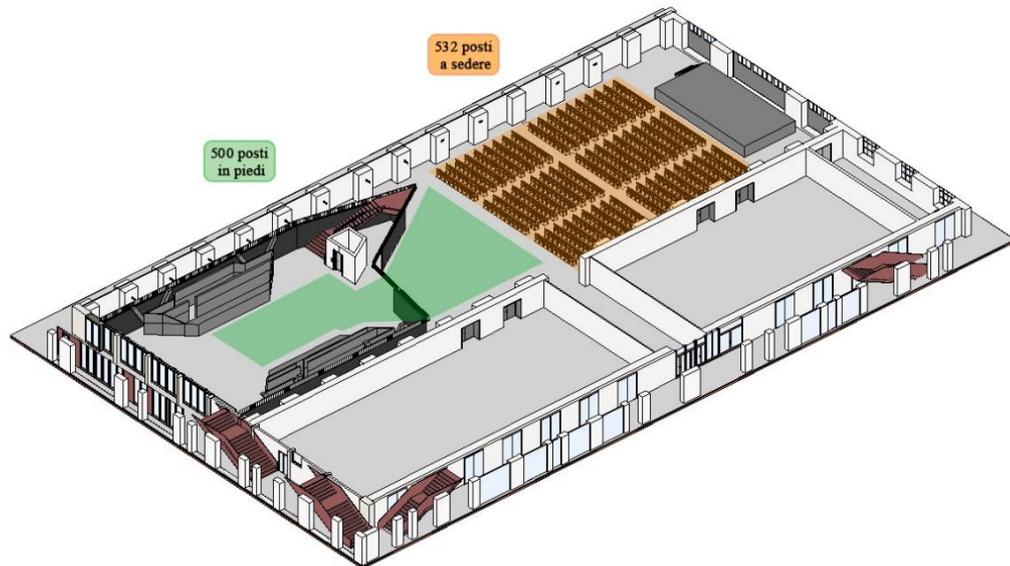


Figura 42 - Configurazione 1: piano terra

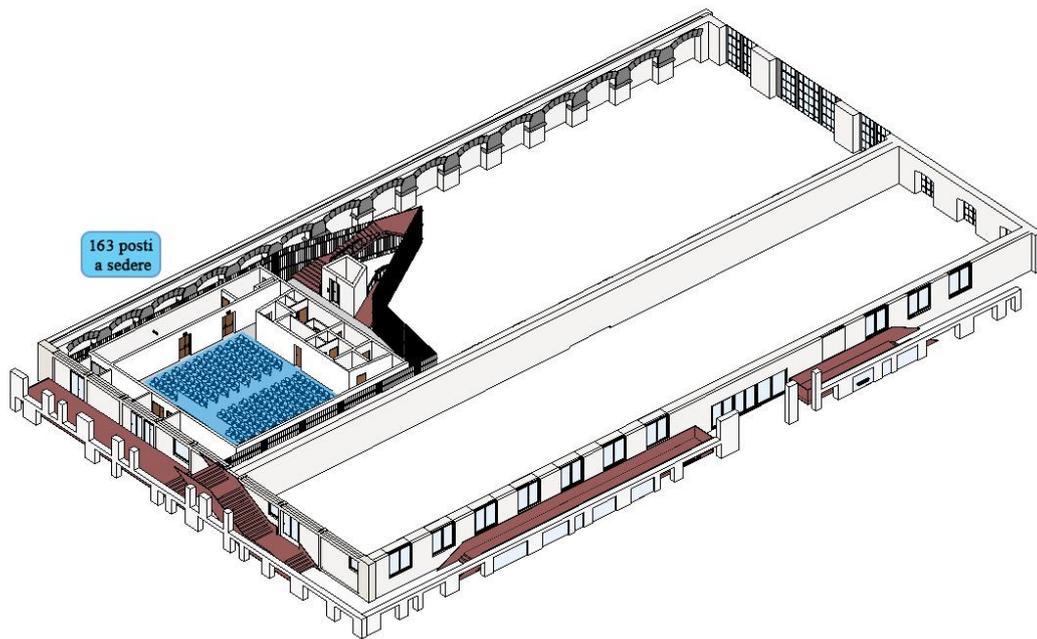


Figura 43 - Configurazione 1: primo piano

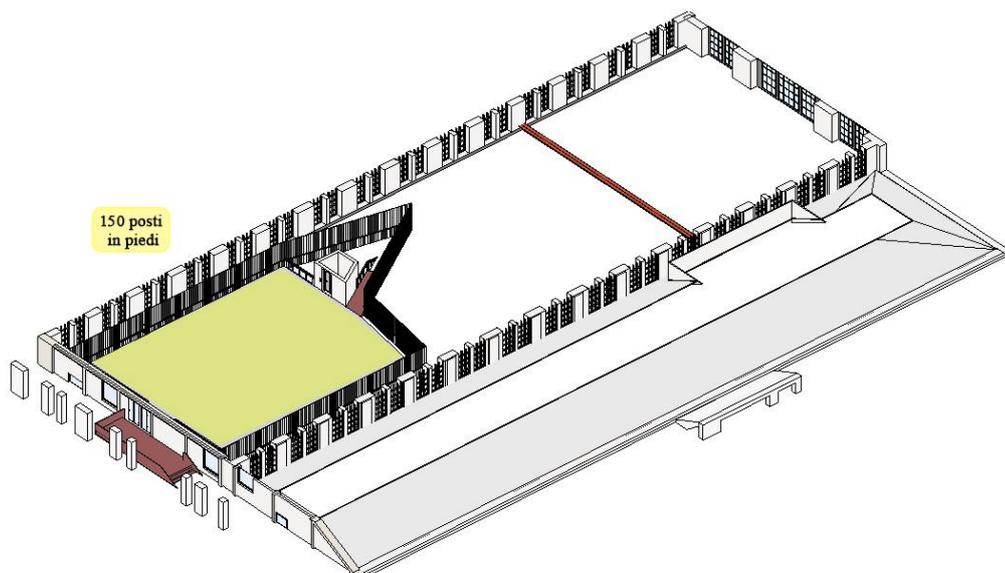


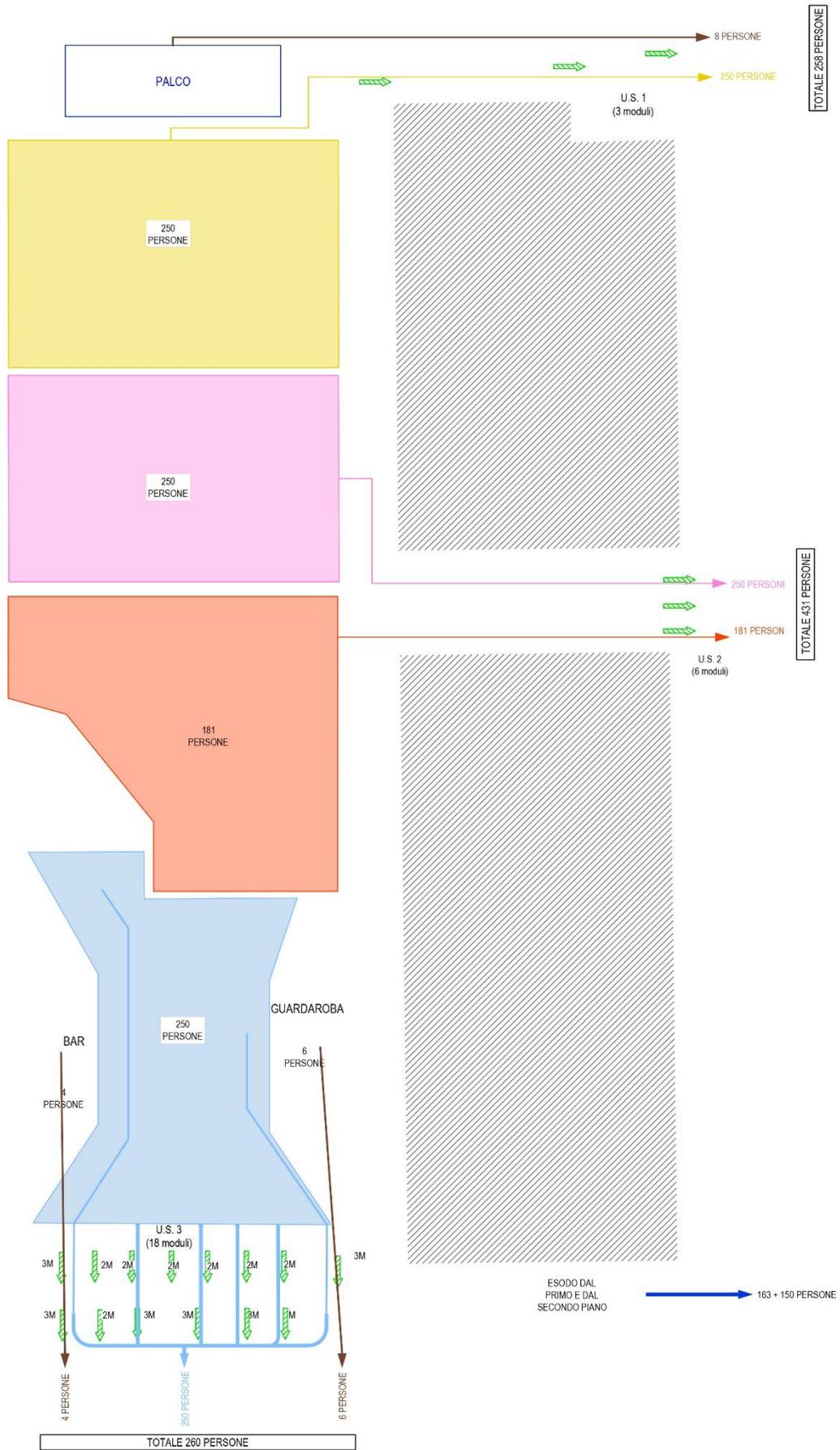
Figura 44 - Configurazione 1: piano secondo

La **seconda configurazione**, con destinazione d'uso **sala da ballo**, prevede:

Tabella 10 - Divisione della seconda configurazione

CONFIGURAZIONE 2: Sala da ballo	
Piano terra	931 posti in piedi
	10 addetti bar + guardargoba
Piano primo	150 posti in piedi
Piano secondo	150 posti in piedi

Il piano terra presenta 27 moduli d'uscita, il primo piano presenta 7 moduli d'uscita ed infine, il secondo piano ne presenta 4.



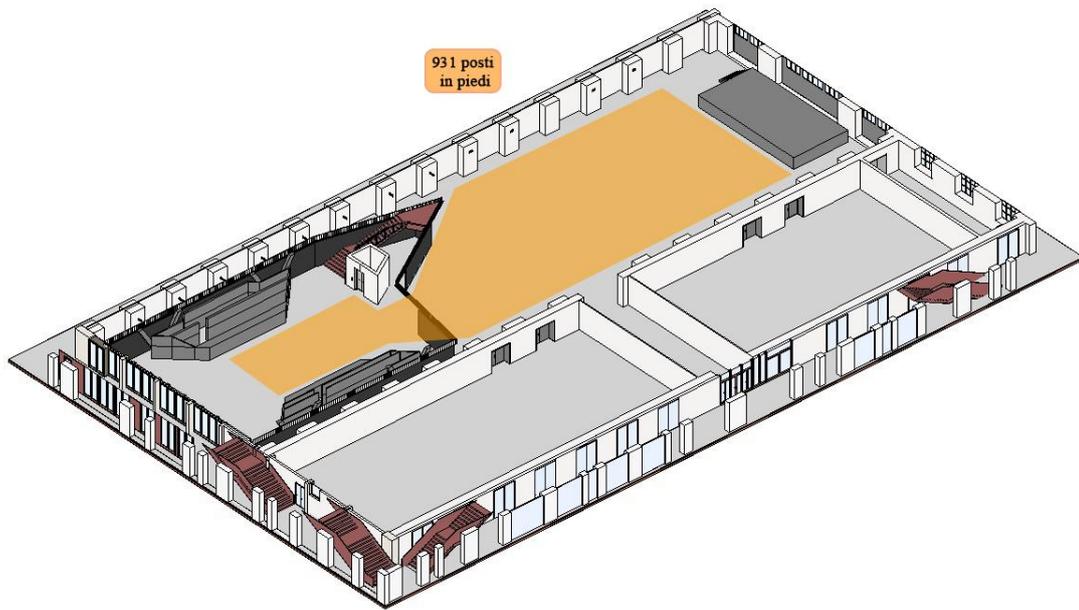


Figura 45 - Configurazione 2: piano terra

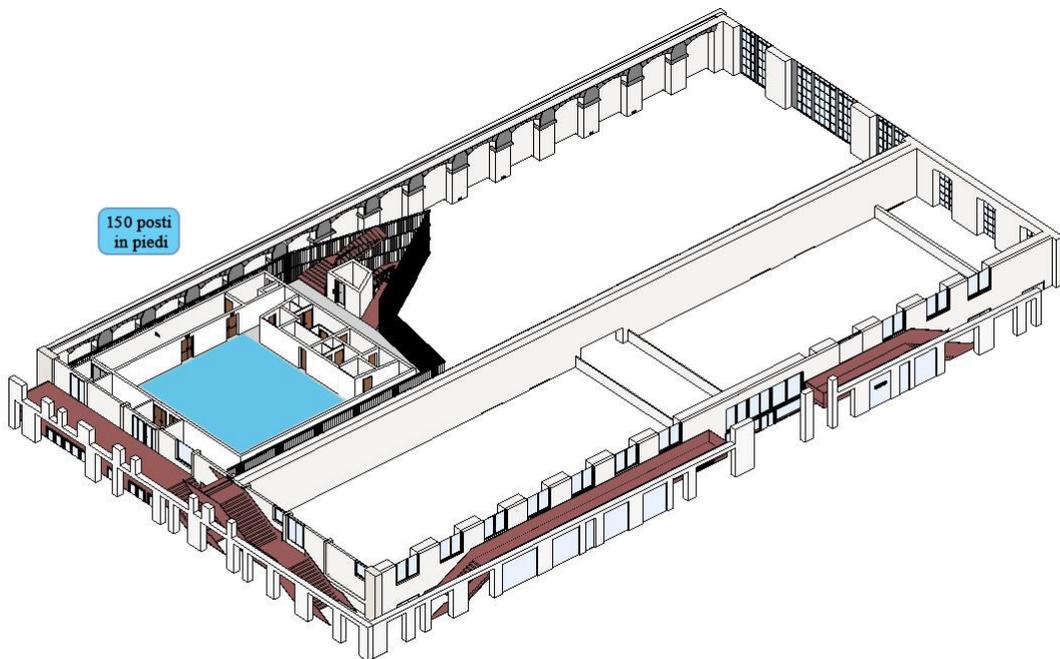


Figura 46 - Configurazione 2: piano primo

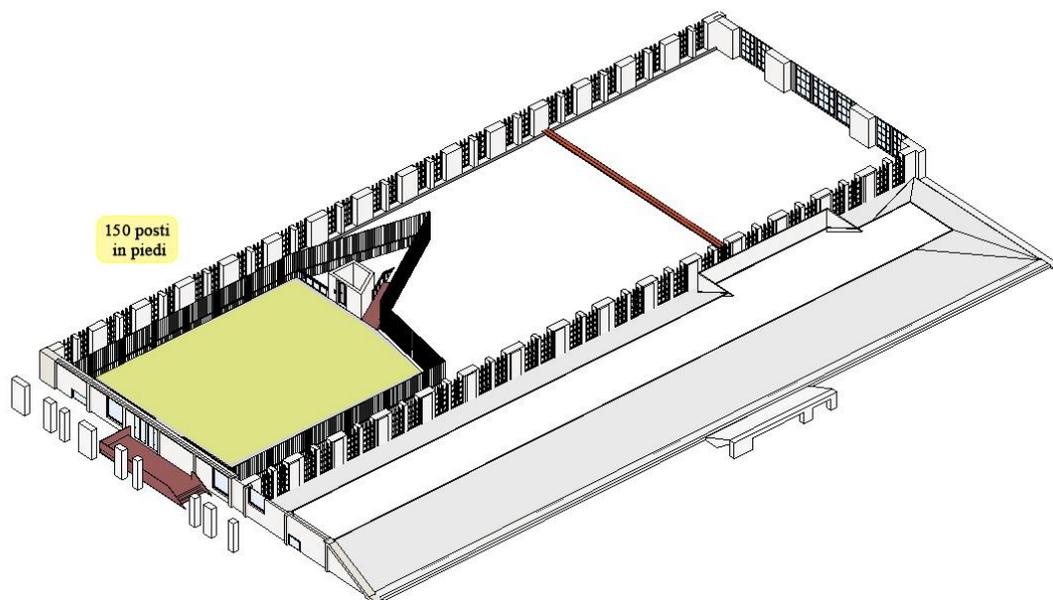


Figura 47 - Configurazione 2: piano secondo

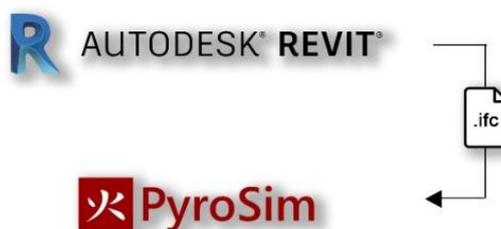
### 6.3.2. Simulazione di incendio

Per la modellazione dell'incendio è stato utilizzato il software Pyrosim, della Thunderhead Engineering Consultants.

Questo software è un'interfaccia grafica per i modelli Fire Dynamics Simulator (FDS) e permette la simulazione degli effetti dell'incendio quali fumo, monossido di carbonio, temperatura e altre sostanze che si propagano a partire da un focolare.

Il progetto architettonico della Centrale della Lavazza è stato realizzato mediante il software Autodesk Revit, questo ha permesso di ottenere un modello di partenza ricco di informazioni e tridimensionale.

Nel dicembre 2018 è stata rilasciata la versione che permette di importare modelli in formato IFC e riconoscere direttamente i materiali e le ostruzioni.



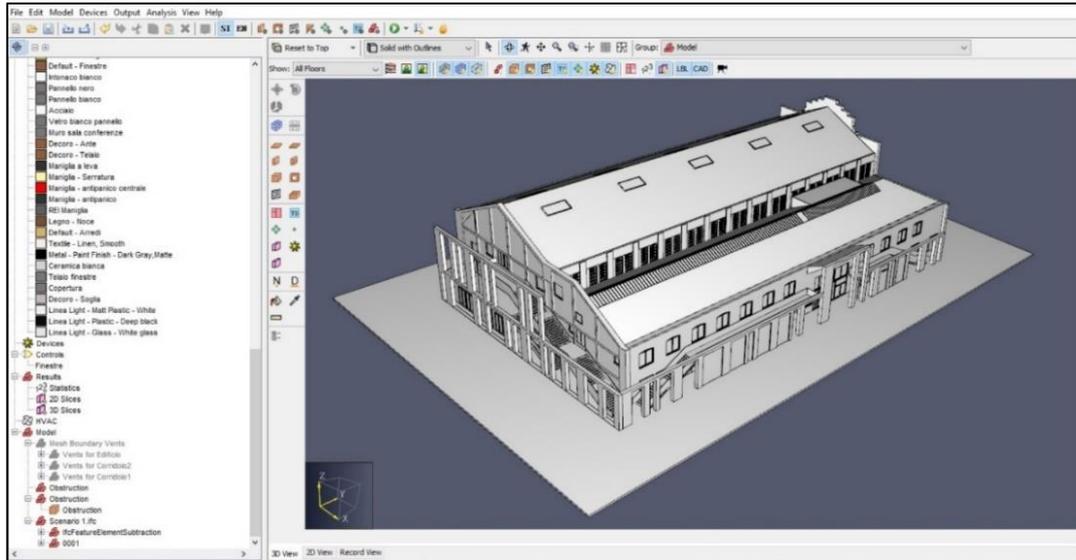


Figura 48 - Risultato finale dell'importazione

### Calibrazione della mesh

Il dominio di campo, secondo le equazioni della fluidodinamica, deve essere suddiviso in elementi chiamati “celle” che dipendono dalla risoluzione dei risultati che si intende ottenere. L’insieme delle celle viene chiamato “mesh”.

L’accuratezza della discretizzazione del dominio viene calcolata mediante il diametro  $D^*$ , ovvero il diametro caratteristiche dell’incendio, secondo la seguente formula:

$$D^* = \left( \frac{\dot{Q}}{\rho_{\infty} \cdot c_p \cdot T_{\infty} \cdot \sqrt{g}} \right)^{\frac{2}{5}} \quad [m]$$

In cui:

- $\dot{Q}$  corrisponde al rilascio termico espresso in kW;
- $\rho$  corrisponde alla densità dell’aria espressa in  $\text{kgm}^3$ ;
- $T$  è la temperatura espressa in K;
- $c_p$  è il calore specifico in  $\text{kJkg}^{-1}\text{K}^{-1}$ ;
- $g$  è l’accelerazione di gravità espressa in  $\text{ms}^{-2}$ ;

Nonostante questo permetta di ottenere risultati molto accurati, è anche vero che i tempi computazionali aumentano significativamente. Per questo motivo, una volta importato il modello, l’ambiente è stato suddiviso in celle, al cui interno vengono

risolte le equazioni di conservazione di massa, energia e quantità di moto, attraverso il metodo degli elementi finiti.

Per entrambi gli scenari di seguito analizzati, è stato scelto di suddividere l'ambiente in tre mesh, le cui celle, hanno dimensioni differenti:

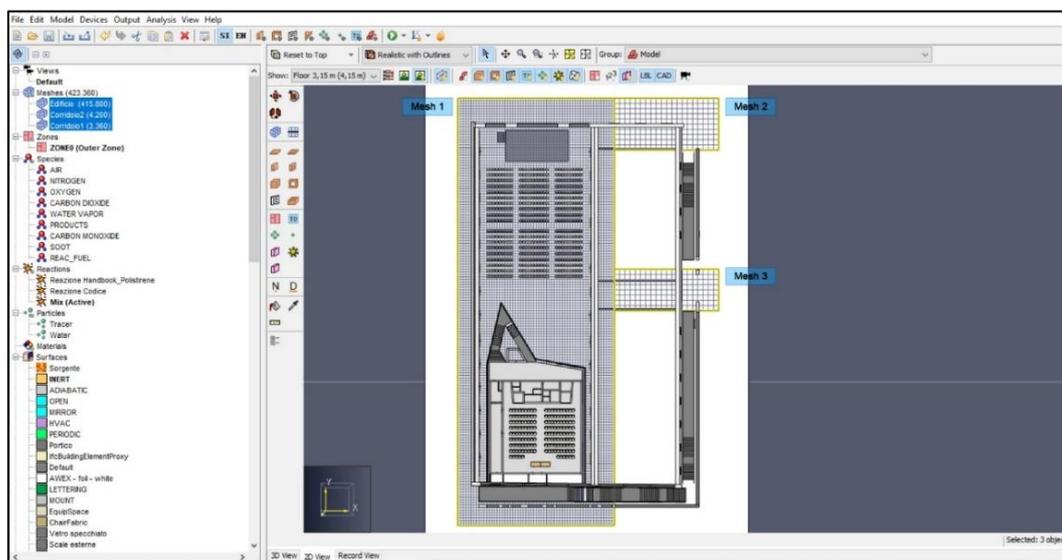


Figura 49 - Suddivisione dell'ambiente in mesh

Le mesh 2 e 3 presentano dimensioni delle celle pari a 1,0x1,0x1,0 metri, invece, la mesh 1 essendo l'ambiente principale dello scenario presenta dimensioni delle celle pari a 0,5x0,5x0,5 metri.

È utile ricordare che, la dimensione delle celle è uno dei fattori che contribuisce sostanzialmente all'aumento del tempo computazionale di simulazione.

#### 6.3.2.1. Scenario 1 – Sala conferenze

Ai fini della descrizione delle configurazioni individuate, occorre selezionare gli scenari di incendio più idonei, descrivendoli mediante alcuni fattori principali.

#### **Individuazione del focolare**

Per il primo scenario, la sala conferenze, si è scelto di analizzare il focolare in prossimità del palco dovuto ad un corto circuito della linea elettrica.

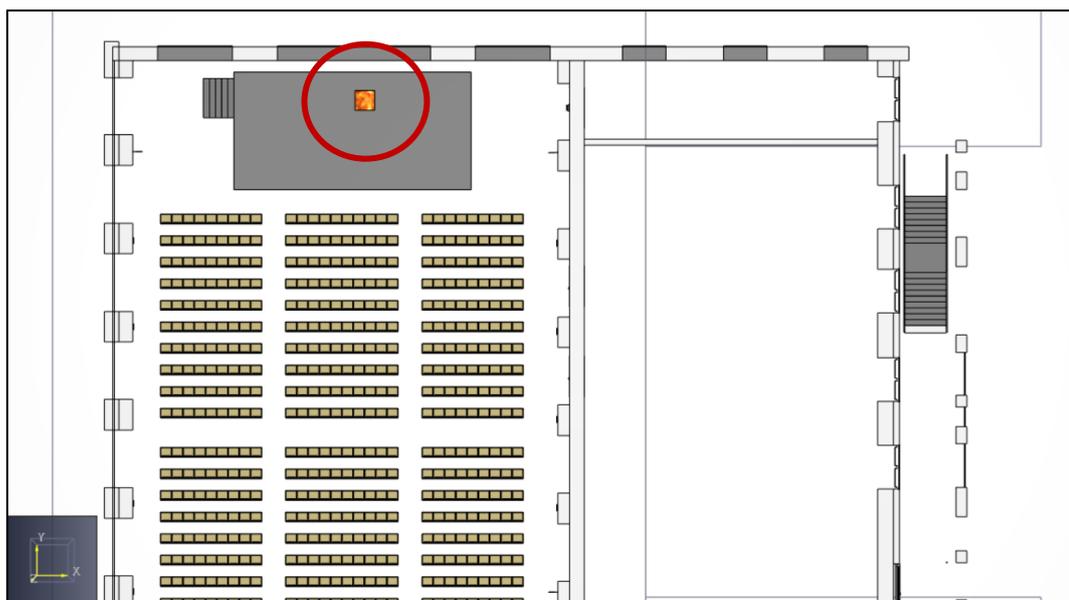


Figura 50 - Posizionamento del focolare

### Caratterizzazione del focolare

L'elemento chiave necessario per descrivere ciascuno degli scenari è rappresentato dal tipo di focolare e l'evoluzione dell'incendio nel tempo. Per questo motivo si sono individuati alcuni degli elementi e materiali con maggiore probabilità di presenza all'interno degli ambienti studiati.

Per cui, a partire dal manuale “SFPE Handbook of Fire Protection Engineering” [21] e della letteratura [22] sono state individuate le curve HRR relative agli elementi che partecipano alla combustione.

Gli elementi presi in esame sono:

- Stereo
- Giornali
- Monitors
- Sedie in legno

Di seguito si riportano le curve considerate per ciascun elemento:

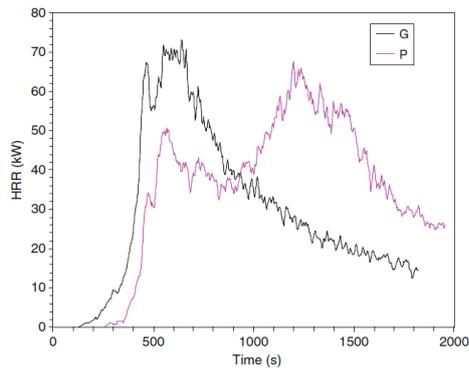


Figura 51 - HRR di due sistemi di stereo

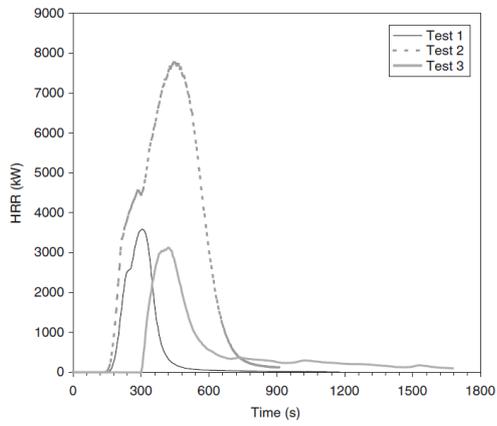


Figura 52 - HRR di libri e giornali

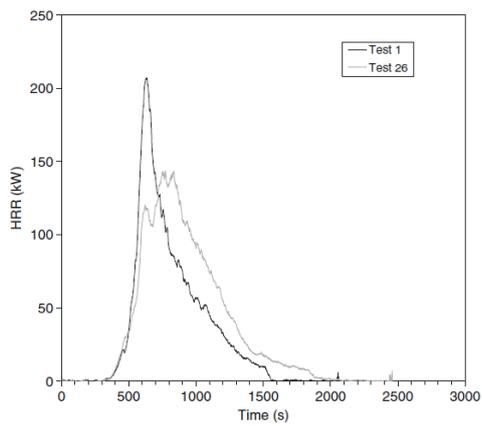


Figura 53 - HRR di Monitor

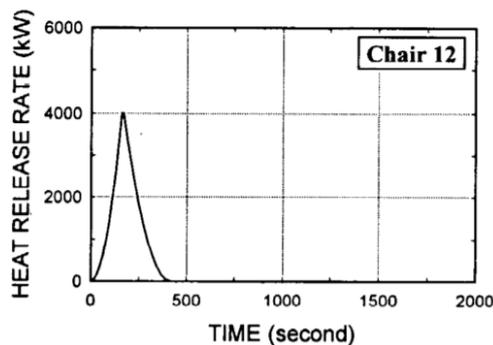


Figura 54 - HRR di sedie in legno

Per ogni curva in esame, è stata individuata quella più critica e sono state combinate in un unico grafico, come è possibile vedere nelle immagini successive:

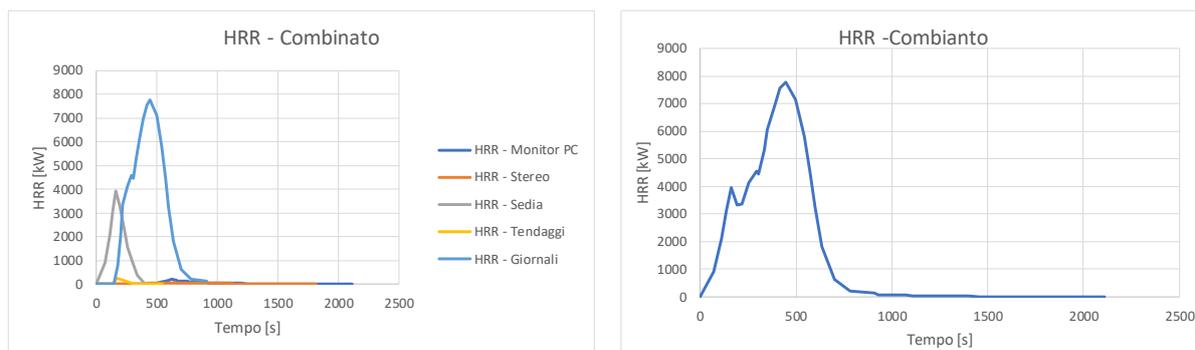


Figura 55 - HRR utilizzato

Una volta individuata correttamente la curva relativa all'incendio, i dati relativi all'andamento della curva nel tempo sono stati inseriti all'interno di Pyrosim. Successivamente, è stata definita la reazione caratteristica dell'incendio i cui dati sono stati estrapolati dal manuale "SFPE Handbook of Fire Protection Engineering" [21].

Tabella 11 - Caratteristiche dell'incendio considerate

Elemento	Combustibile	Reazione	$\Delta H_c$ (kJ/g)	$Y_{co}$ (g/g)	$Y_s$ (g/g)
(Giornali e tende)	Cellulose	$C_6H_{10}O_5$	16,09		
(Palco)	Wood	$CH_{1.7}O_{0.83}$	19,5	0,005	0,015
(Sedie)	Polipropilene	$C_3H_6$	43,31	0,024	0,058
(Monitor e Stereo)	Polistirene	$C_8H_8$	39,85	0,065	0,21
		$C_6H_{10}O_5$	29,6875	0,0235	0,07075

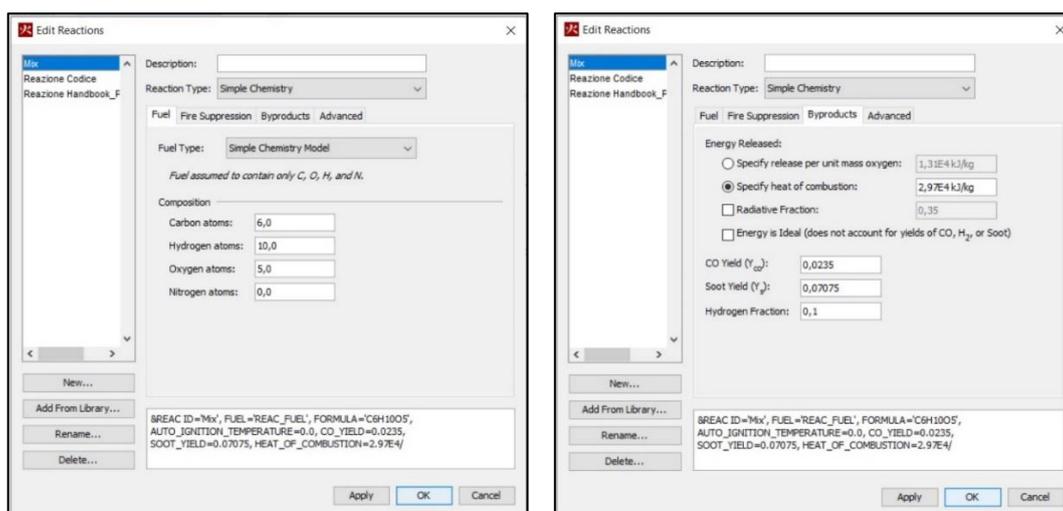


Figura 56 - Modifica della reazione all'interno del software Pyrosim

Ai fini dell'obiettivo di questo lavoro di tesi, ovvero l'esportazione e la conseguente visualizzazione dei fumi in Virtual Reality, non è necessario inserire devices, isosuperfici o slices. Inoltre, è utile tenere presente che, nonostante siano presenti degli evacuatori di fumo e di calore che si attivano a circa 250 secondi, negli scenari analizzati non sono stati considerati in quanto il tempo di simulazione è di 160 secondi.

La simulazione ha prodotto i seguenti risultati:

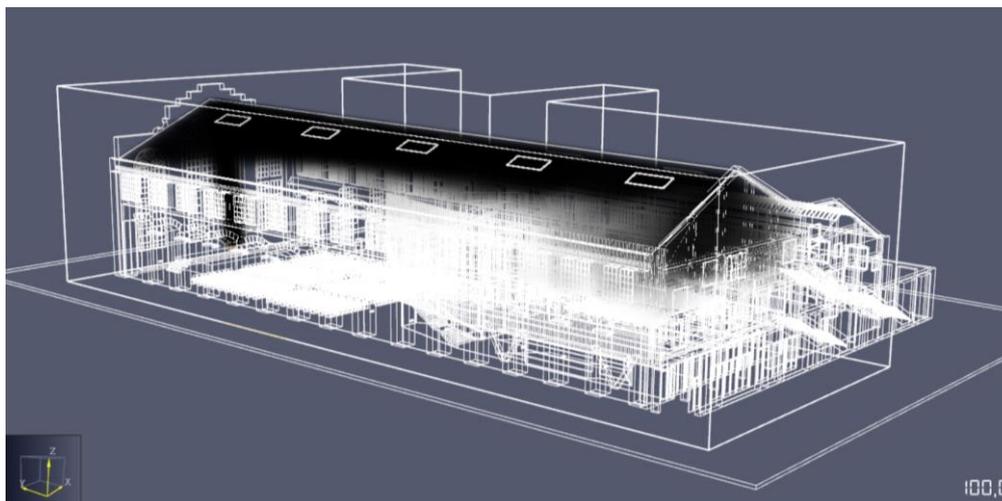


Figura 57 - Visualizzazione del fumo su Pyrosim

#### 6.3.2.2. Scenario 2: SALA DA BALLO

##### **Individuazione del focolare**

Per quanto riguarda il secondo scenario, ovvero la sala da ballo, è stato considerato come focolaio di progetto quello relativo al guardaroba, che risulta peggiorativo rispetto a quello della caffetteria.

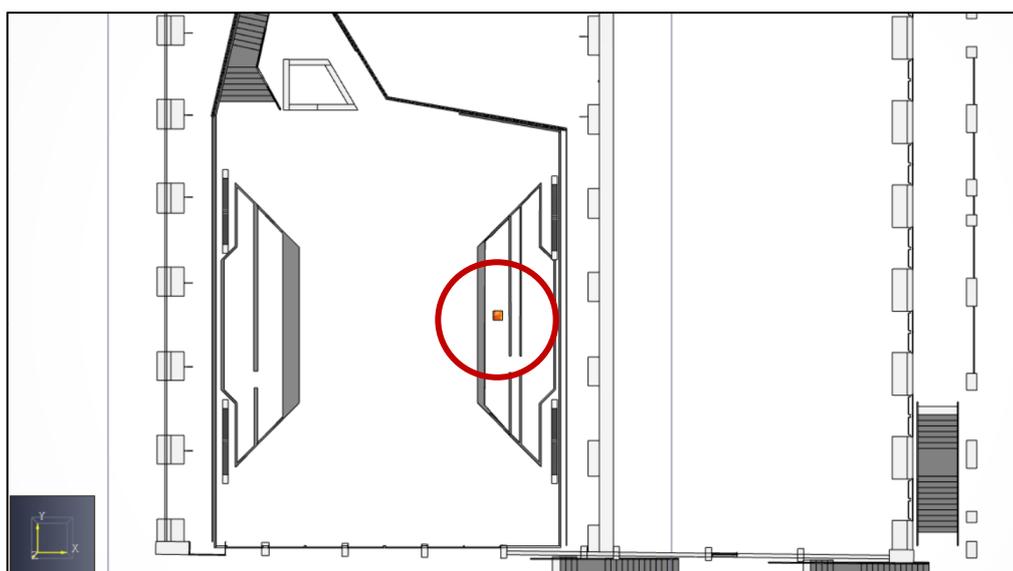


Figura 58 - Posizionamento del focolare

### Caratterizzazione del focolare

Per la valutazione della potenza termica si è considerato l'incendio di 250 cappotti presenti all'interno del guardaroba, aventi una potenza termica di 40 kW ognuno.

Per cui la potenza termica massima risulta essere 10000 kW ovvero 10 MW.

La curva scelta in questo caso è la seguente:

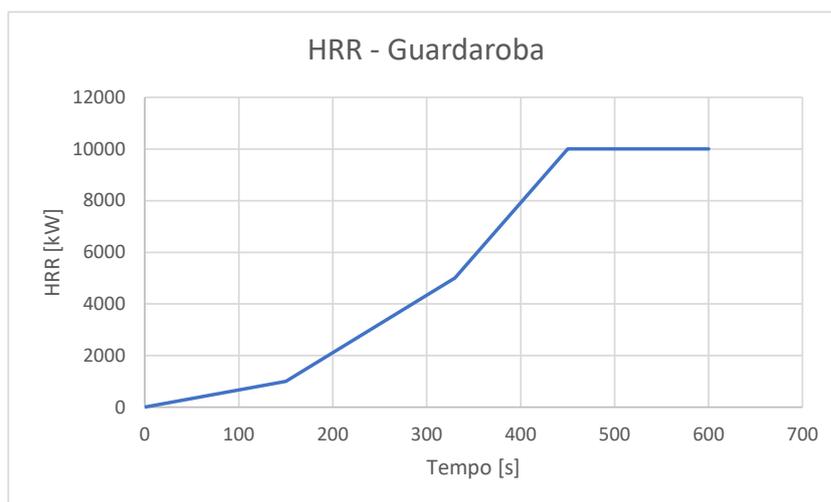


Figura 59 - HRR utilizzato per la seconda configurazione

La curva presenta un andamento fast, infatti, in 150 secondi la potenza termica arriva a 1000 kW.

Successivamente, per quanto riguarda le caratteristiche dell'incendio sono stati utilizzati i dati relativi al focolare predefinito del Codice di Prevenzione Incendi [1]:

Tabella 12 - Caratteristiche dell'incendio utilizzate

Reazione	$\Delta h_c$ (kJ/g)	$Y_{co}$ (g/g)	$Y_{soot}$ (g/g)
CH <sub>2</sub> O <sub>0,5</sub>	20	0,1	0,07

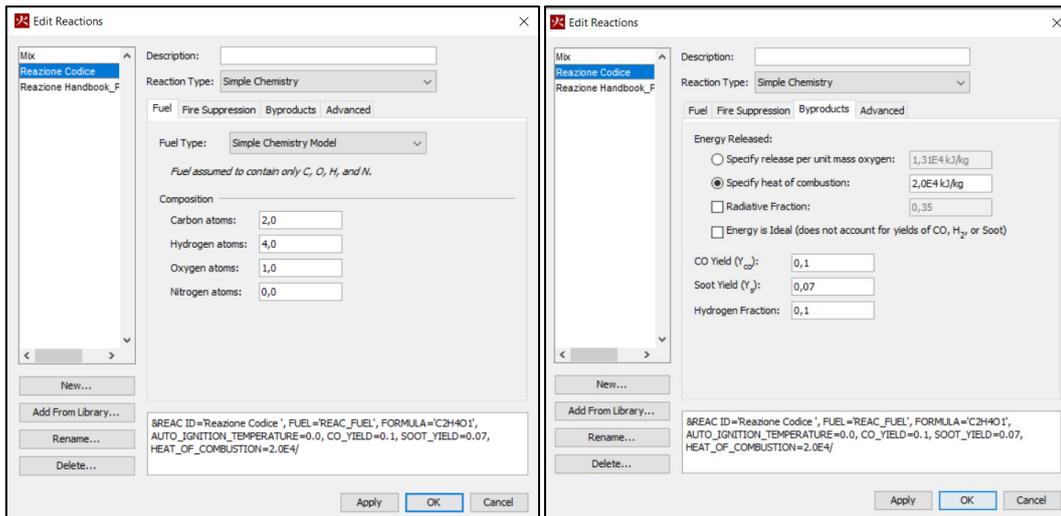


Figura 60 - Modifica della reazione all'interno di Pyrosim

La simulazione ha prodotto i seguenti risultati:

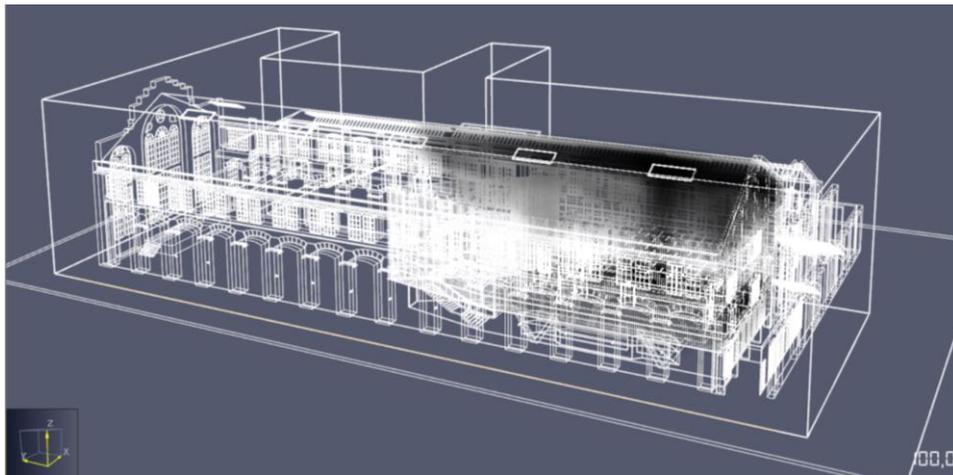


Figura 61 - Visualizzazione del fumo su Pyrosim

---

### 6.3.3. Simulazione di esodo

Nel paragrafo 1.6 abbiamo visto che esistono due tipologie di modellazione dell'esodo: il modello idraulico e il modello agent-based. In questo caso studio, l'esodo è stato valutato mediante il modello ABM (Agent-Based Model) secondo cui ogni agente valuta individualmente la propria situazione e prende decisioni sulla base di una serie di regole.

Come detto in precedenza, il tempo disponibile per evacuare l'edificio ASET deve essere maggiore del tempo richiesto dagli occupanti per esodare RSET.

RSET secondo il rapporto tecnico ISO/TR 16738:2009 è pari a:

$$RSET = \Delta t_{det} + \Delta t_a + \Delta t_{pre} + \Delta t_{trav}$$

Dove:

- $\Delta t_{det}$ : corrisponde al tempo di rilevazione ovvero il tempo necessario affinché si attivi il sistema di rilevazione. Nel caso in esame, è pari a 30 secondi ovvero pari al tempo di attivazione dei rilevatori di fumo;
- $\Delta t_a$ : corrisponde al tempo di allarme generale ovvero il tempo che sussiste tra il momento in cui viene rilevato l'incendio e quello in cui gli occupanti vengono informati dell'allarme. E' nullo quando l'allarme è attivato automaticamente dal sistema di rilevazione, al contrario, se non viene attivato automaticamente è pari al ritardo necessario affinché venga attivato manualmente dal un addetto. Nel caso in esame è nullo in quanto la struttura è fornita di un impianto di allarme collegato all'impianto di rilevazione;
- $\Delta t_{pre}$ : rappresenta il lasso di tempo nel quale gli occupanti già si muovono per compiere diverse azioni. Viene stimato in accordo con quanto previsto nella tabella E.2 del rapporto tecnico ISO/TR 16738:2009;

Tabella 13 - Tabella E.2 del rapporto tecnico ISO/TR 16738:2009

Scenario category and modifier levels <sup>a</sup>	First occupants <sup>f</sup> pre (1st percentile)	Occupant distribution <sup>f</sup> pre (99th percentile)
<b>A: Awake and familiar</b>		
M1 B1 – B2 A1 – A2 <sup>a</sup>	0,5	1,5
M2 B1 – B2 A1 – A2	1	3
M3 B1 – B2 A1 – A3	> 15 <sup>b</sup>	> 30 <sup>b</sup>
For B3, add 0,5 for way-finding.	—	—
M1 normally requires a voice alarm/PA if unfamiliar visitors likely to be present.	—	—
<b>B: Awake and unfamiliar</b>		
M1 B1 A1 – A2	0,5	2,5
M2 B1 A1 – A2	1,0	4,0
M3 B1 A1 – A3	> 15 <sup>b</sup>	> 30 <sup>b</sup>
For B2, add 0,5 for way-finding.	—	—
For B3, add 1,0 for way-finding.	—	—
M1 normally requires a voice alarm/PA.	—	—
<b>Ci: Sleeping and familiar (e.g. dwellings, individual occupancy)</b>		
M2 B1 A1	5 <sup>b</sup>	10 <sup>b</sup>
M3 B1 A3	10 <sup>b</sup>	> 40 <sup>b</sup>
For other units in a block, assume 1 h.	—	—
<b>Cii: Managed occupancy (e.g. serviced apartments, hall of residence)</b>		
M1 B2 A1 – A2	10 <sup>b</sup>	30 <sup>b</sup>
M2 B2 A1 – A2	15 <sup>b</sup>	40 <sup>b</sup>
M3 B2 A1 – A3	> 20	> 40
<b>Ciii: Sleeping and unfamiliar (e.g. hotel, boarding house)</b>		
M1 B2 A1 – A2	15 <sup>b</sup>	30 <sup>b</sup>
M2 B2 A1 – A2	20 <sup>b</sup>	40 <sup>b</sup>
M3 B2 A1 – A3	> 20 <sup>b</sup>	> 40 <sup>b</sup>
For B3, add 1,0 for way-finding.	—	—
M1 normally requires a voice alarm/PA.	—	—

Nel nostro caso, consideriamo gli ospiti dell'edificio come occupanti svegli ma non familiari e gli addetti come occupanti svegli e familiari.

- *Attra*: rappresenta il tempo necessario a tutti gli occupanti presenti all'interno dell'edificio di raggiungere un luogo sicuro.

E' possibile attingere a questi dati anche dalla letteratura [23] [24], in un studio sono stati effettuati 40 esperimenti di evacuazione per sei diverse destinazioni d'uso senza preavviso per quantificare il tempo di pre-movimento durante l'evacuazione di un edificio. Le destinazioni d'uso considerate sono state: uffici, cinema, centri commerciali, ristoranti, università e sale da ballo.

Ciascun edificio era dotato di un differente sistema di allarme di evacuazione.

Per quanto riguarda gli esperimenti realizzati nelle sale da ballo è stata effettuata una distinzione tra staff attivo e staff inattivo; gli esperimenti hanno portato a tempi di pre-movimento molto più lunghi quando lo staff non è presente.

Tabella 14 - Tempi di pre-movimento

**Table 6  
Compilation of Observed Total Pre-movement Times Divided by  
Occupancy**

Occupancy	Number of observations	Distribution (s)	Mean (min-max) (s)	SD (s)
Office	45	Loglogistic ( $\gamma = 0; \beta = 52.5; \alpha = 3.0$ )	64.4 (12–201)	45.6
Cinema (voice alarms)	891	Loglogistic ( $\gamma = 0; \beta = 40.5; \alpha = 5.8$ )	44.0 (17–138)	18.0
Cinema (siren)	89	Lognormal ( $\mu = 25.0; \sigma = 6.2$ )	30.0 (14–179)	28.1
Cinema (alarm bell)	880	Loglogistic ( $\gamma = 0; \beta = 29.5; \alpha = 5.0$ )	32.5 (11–224)	17.2
Department store	229	Lognormal ( $\mu = 35.9; \sigma = 18.3$ )	35.9 (5–111)	17.7
Restaurant/Café	27	Weibull ( $\alpha = 3.8; \beta = 58.1$ )	52.5 (20–86)	15.7
School	72	Gamma ( $\alpha = 3.0; \beta = 24.5$ )	74.9 (13–170)	42.3
Night club (active staff)	62	Weibull ( $\alpha = 2.7; \beta = 52.5$ )	46.6 (11–87)	18.7
Night club (passive staff)	84	Loglogistic ( $\gamma = 0; \beta = 50.6; \alpha = 2.3$ )	65.4 (5–417)	64.0

Per il caso in esame, si è deciso di utilizzare i dati previsti dal rapporto tecnico, in quanto in questo studio, gli esperimenti effettuati nei nightclub sono solamente quattro e non sono ritenuti sufficienti.

In conclusione, il tempo di pre – evacuazione considerato è pari a:

- $\Delta t_{det} = 0$  secondi;
- $\Delta t_a = 30$  secondi;
- $\Delta t_{pre} = 60 - 240$  secondi.

I software utilizzati per le simulazioni di esodo sono molteplici e hanno il compito di calcolare RSET ovvero la somma dei diversi addenti.

Il software scelto è MassMotion, sviluppato da Oasys; questo software nasce per simulare flussi ordinari di persone in ambienti chiusi e/o aperti. La scelta è ricaduta su MassMotion in quanto ai fini degli obiettivi di questo lavoro di tesi, ovvero la creazione di un metodo per l'esportazione e conseguente visualizzazione dei modelli di FDS in Virtual Reality, nessun altro software permetteva l'esportazione degli occupanti.



Figura 62 - Metodologia

Infatti, il software è costituito da un potente post-processore che consente una produzione semplice e veloce di video ed immagini, sfruttando il formato Alembic (.abc).

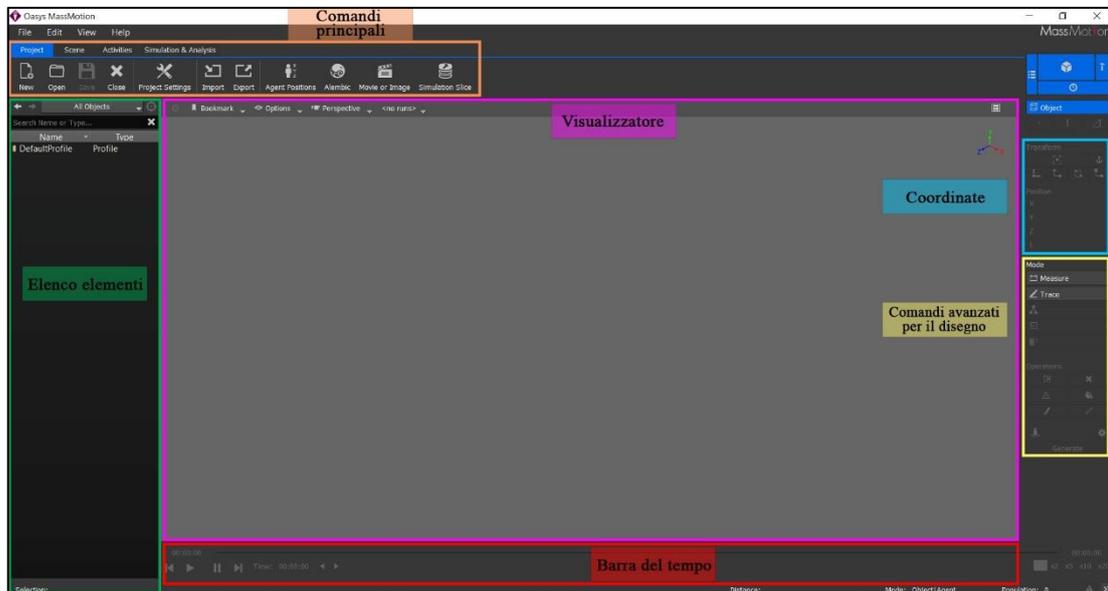


Figura 63 - Interfaccia grafica di MassMotion

La prima fase da effettuare è l'importazione del modello dell'edificio da Autodesk Revit mediante il formato .ifc.

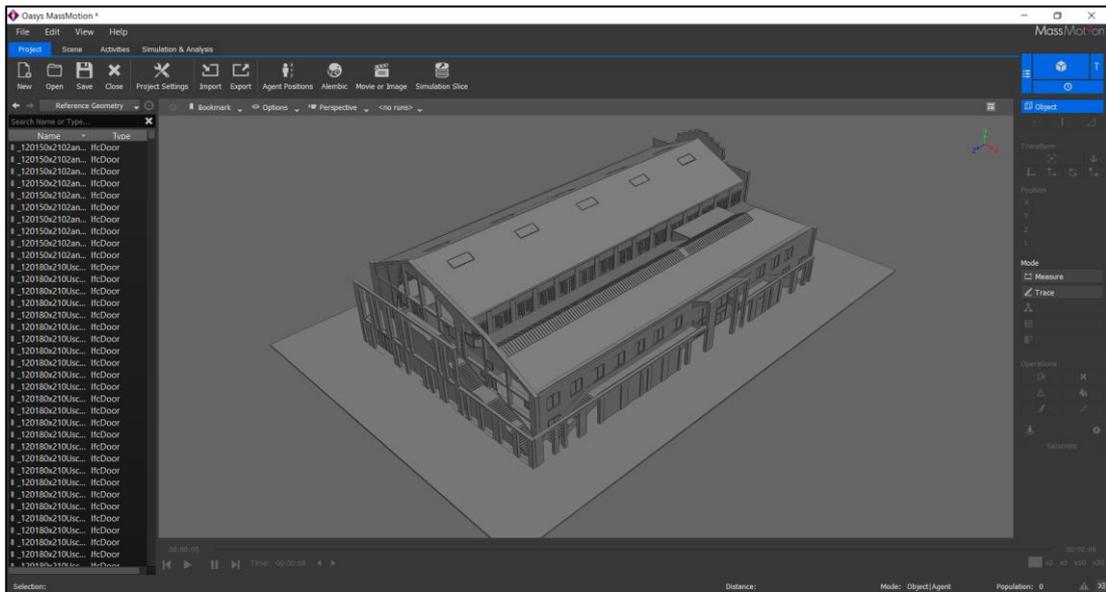


Figura 64 - Importazione del modello

Il formato .ifc permette al software di riconoscere automaticamente pavimenti, scale, rampe, porte ecc.

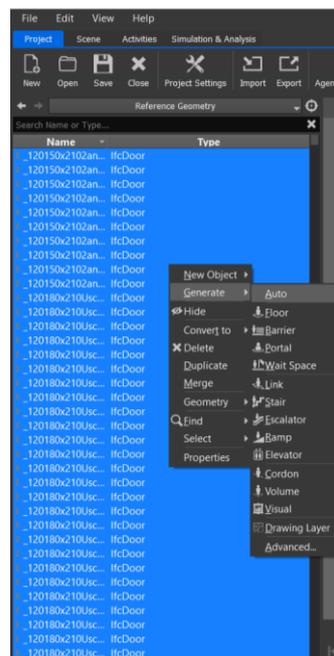


Figura 65 - Conversione automatica degli elementi

Nonostante questa funzione velocizzi notevolmente il lavoro, è necessario risolvere alcuni errori e convertire manualmente gli elementi che non sono stati riconosciuti automaticamente dal software.

I concetti principali di MassMotion sono essenzialmente tre:

- Scene: gli spazi del mondo reale vengono modellati e suddivisi in parti componenti in base alla loro funzione come: pavimento, collegamento, scala, rampa, scala mobile, percorso, ascensore, portale e barriera.

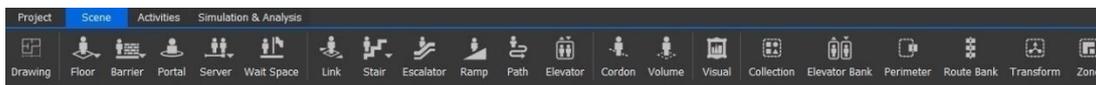
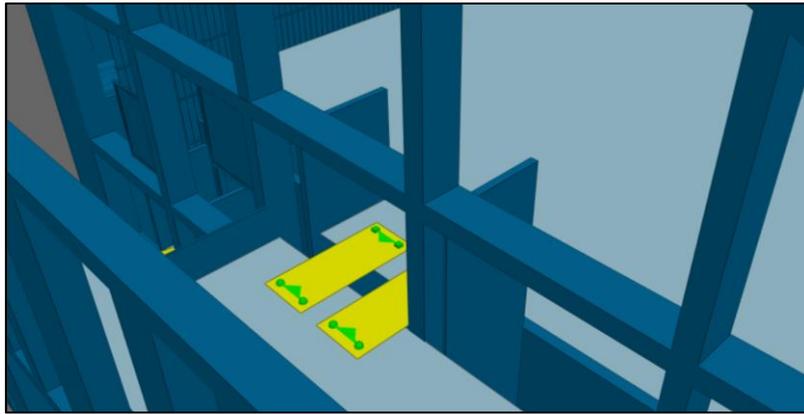


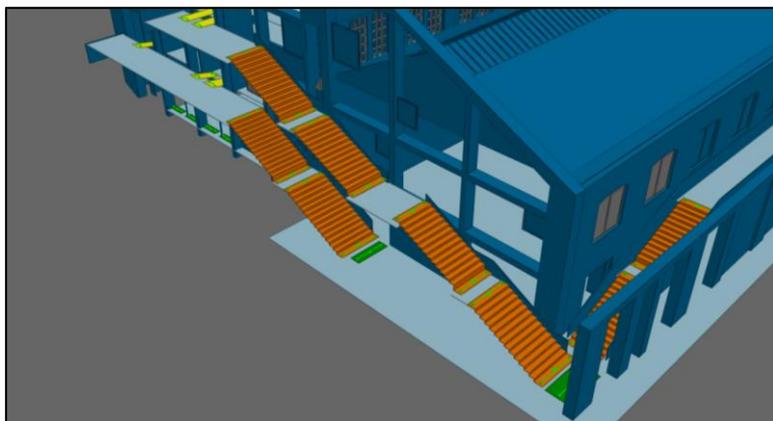
Figura 66 - Scene Ribbon Bar

Le geometrie di base sono le seguenti:

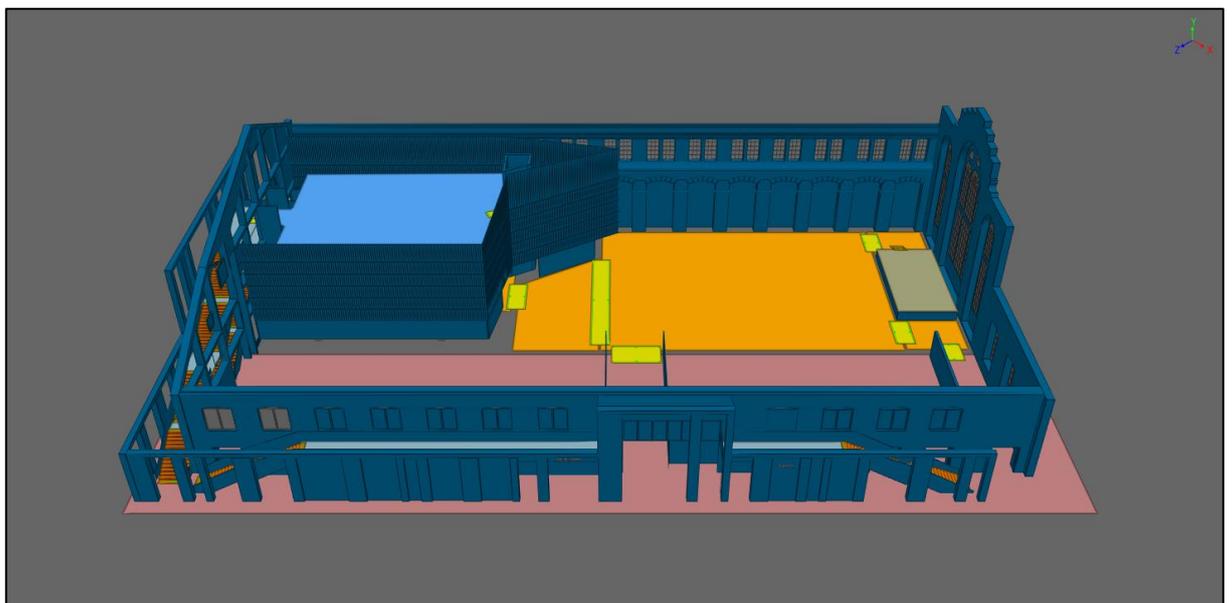
- *Floor*: i pavimenti sono l'oggetto della scena più fondamentale. Rappresentano gli spazi che definiscono le aree del progetto. Ogni pavimento definisce una zona calpestabile separata e distinta, con movimento pedonale vincolato al perimetro del solaio;
- *Barrier*: le barriere rappresentano muri, colonne, tavoli, panchine e qualsiasi altra cosa che possa limitare il movimento su un pavimento;
- *Portals*: i portali svolgono due funzioni principali ovvero contrassegnano le aree in cui le persone possono accedere alla simulazione e rappresentano anche le destinazioni a cui le persone possono essere inviate;
- *Stair*: elemento di collegamento verticale (scale);
- *Escalator*: elemento di collegamento verticale (scala mobile);
- *Ramp*: elemento di collegamento verticale (rampa);
- *Elevator*: elemento di collegamento verticale (ascensore);
- *Server*: viene utilizzato per modellare code e comportamenti complessi;
- *Visual*: sono tutti quegli elementi che non hanno una vera e propria funzione all'interno della simulazione, ma hanno il solo obiettivo di migliorare la visualizzazione grafica;
- *Link*: sono dei collegamenti orizzontali che permettono di connettere due piani appartenenti ad uno stesso livello;



*Figura 67 - Creazione dei "Link"*



*Figura 68 - Creazione delle scale*



*Figura 69 - Modello finale ottenuto per lo scenario 1 (Sala conferenze)*

- **Activities:** è possibile creare gli agenti, definire le loro caratteristiche e il loro comportamento.

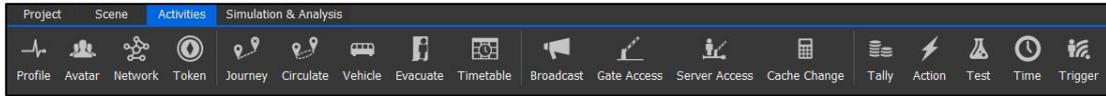


Figura 70 - Activities Ribbon Bar

- **Profile:** questo comando permette di caratterizzare gli agenti, ovvero gli occupanti. È possibile selezionare dei profili preimpostati dal software, derivanti dai maggiori studi sull'argomento oppure si possono impostare manualmente. È possibile stabilire la velocità orizzontale e verticale dell'occupante e associare un determinato avatar.

I profili disponibili su MassMotion sono i seguenti:

Tabella 15 - Profili disponibili su MassMotion

Description	Distribution	Velocità dell'occupante			
		Minimo	Massimo	Media	Deviazione standard
Fruin communter	Normal	0.65	2.05	1.35	0.25
PD 7974	Constant	-	-	1.19	-
Non - PRM	Normal	1.1	1.9	1.53	0.13
Wheelchair	Constant	-	-	0.58	-
Mobility Impaired	Constant	-	-	0.8	-
Small Luggage	Normal	1.1	1.9	1.53	0.13
Large Luggage	Normal	0.65	2.05	1.53	0.23
Heavy Luggage	Normal	0.9	1.8	1.32	0.15
Adult with child	Normal	1.0	1.9	1.37	0.15

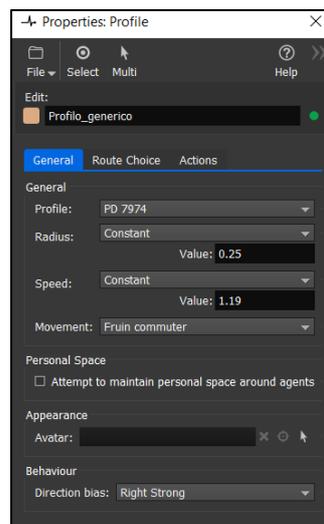


Figura 71 - Definizione del profilo degli occupanti

- *Avatar*: permette di modificare la rappresentazione fisica di un agente;
- *Network*: questo comando permette di definire gli oggetti di rete disponibili per l'agente, ad esempio, potrebbero essere simulati agenti con mobilità ridotta utilizzando una rete che esclude tutte le scale;
- *Token*: vengono utilizzati per identificare determinati agenti principalmente attraverso l'uso di test;
- *Journey*: questo comando viene utilizzato per creare agenti su un determinato portale di origini e li conduce ad un altro portale di uscita;
- *Evacuate*: viene utilizzato per simulare le evacuazioni. Permette di indicare il numero degli occupanti e di gestirne l'esodo, inoltre, permette di indicare l'intervallo di tempo che gli occupanti devono attendere prima di muoversi verso i portali di destinazione.

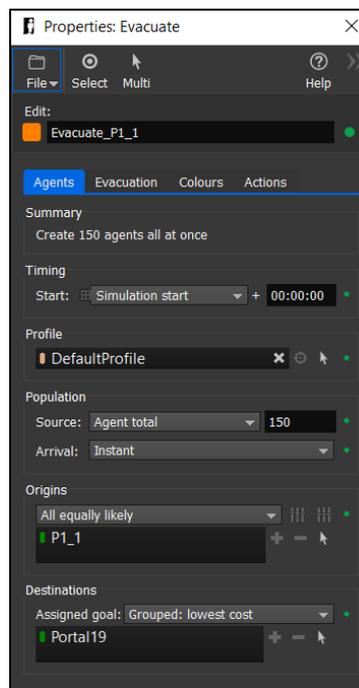


Figura 72 - Creazione degli Evacuate

Mediante il comando *Origins* vengono indicati i portali in cui verranno creati gli occupanti, invece, con il comando *Destinations* vengono indicati i portali che gli occupanti devono raggiungere.

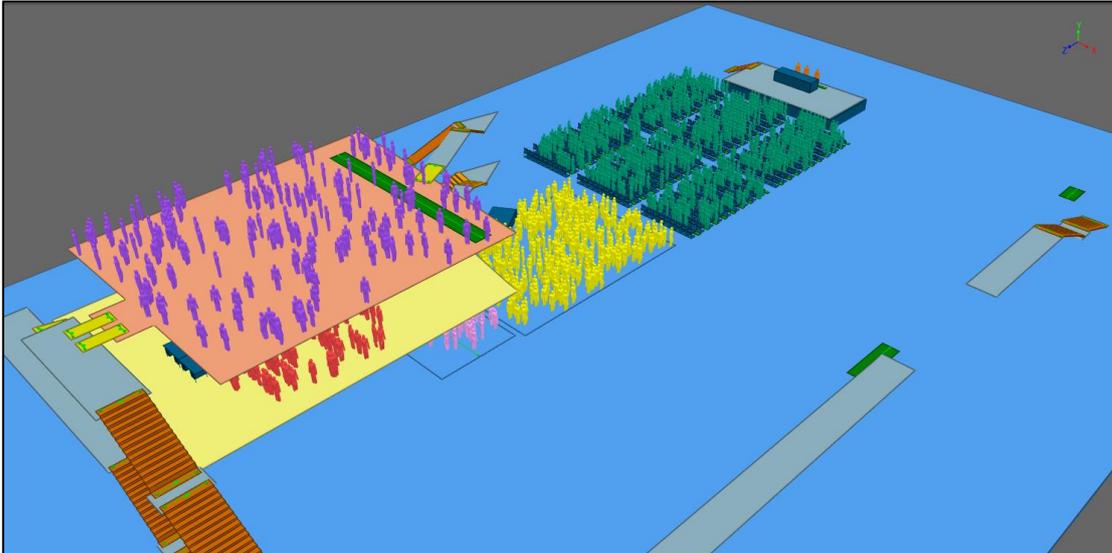


Figura 73 - Modello finale per lo scenario 1

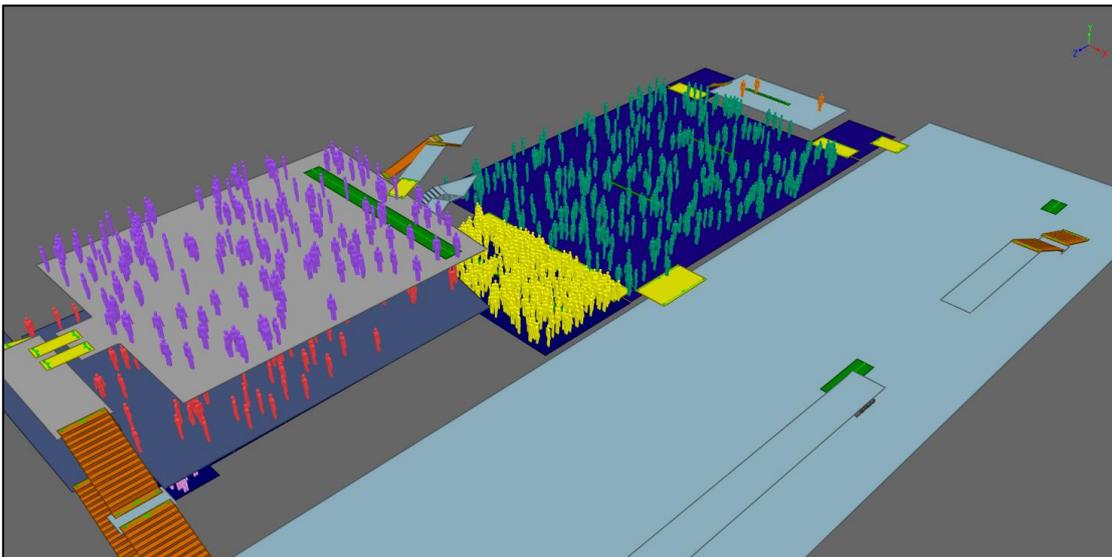


Figura 74 - Modello finale per lo scenario 2

Ai fini dell'obiettivo di questo lavoro di tesi, non è stato necessario prendere in considerazione i tempi di pre-movimento, in quanto, ciò che principalmente ci interessa è l'animazione degli occupanti. Infatti, per le simulazioni di incendio, abbiamo preso in considerazione curve HRR molto alte, che non ci permettono di rispettare l'equazione  $ASET > RSET$ .

Ai soli fini dimostrativi, per il **secondo scenario**, ovvero la sala da ballo, sono stati considerati.

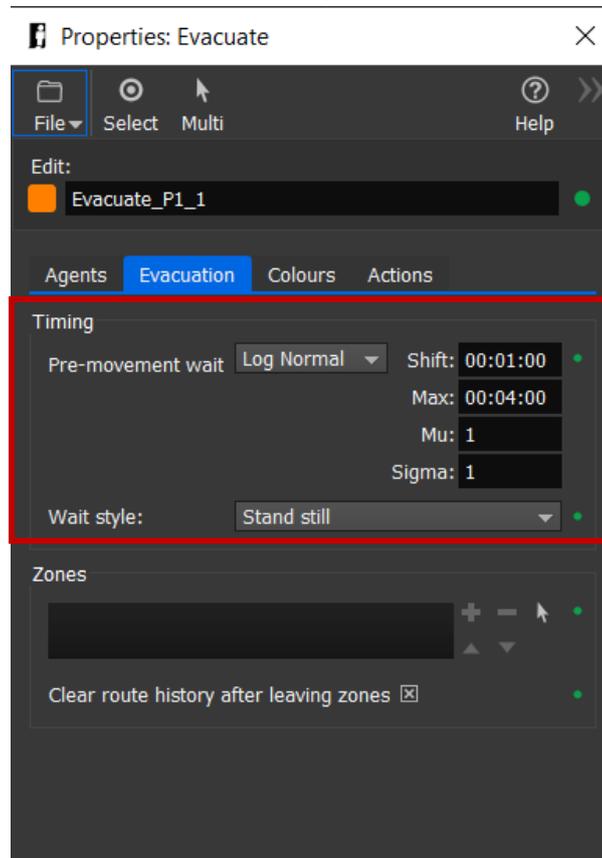


Figura 75 - Modifica dei tempi di pre-evacuazione per il secondo scenario

## Capitolo 7

# Dai modelli di Fire Safety Engineering alla Virtual Reality

L'uso della Virtual Reality come strumento di ricerca nella tecnologia antincendio e nelle simulazioni di esodo si sta sempre più affermando nel mondo.

Tuttavia, è ancora necessaria la visualizzazione fisica del fuoco e del fumo, ciò include l'interazione tra la sorgente luminosa e il fumo in un ambiente 3D che può essere applicato alla realtà virtuale interattiva.

Nel campo dell'ingegneria, la dinamica dei fluidi computazionali (CFD) è ampiamente utilizzata per prevedere con precisione i comportamenti dei fenomeni fluidi. I dati risultanti, tuttavia, non sono applicati ai sistemi di realtà virtuale in tempo reale.

Come detto precedentemente, l'obiettivo di questo lavoro di tesi è **sviluppare un metodo per l'esportazione** dei modelli di Fire Safety Engineering, simulazione di incendio e simulazione di esodo, in Virtual Reality e la loro conseguente visualizzazione in essa. Il metodo affronta numerosi steps, questo perché, non esiste un collegamento diretto tra i software. Infatti, è stato necessario ricorrere a software intermedi che hanno permesso l'esportazione finale. La scelta dei software è stata effettuata sulla base della visualizzazione grafica dei dati, difatti, è stato necessario eseguire molteplici prove per raggiungere dei risultati ottimali.

Di seguito, vengono descritte dettagliatamente tutte le fasi svolte.

### 7.1. Esportazione della simulazione di incendio

Nonostante nel mondo si stia sempre più affermando il mondo della Virtual Reality, non è ancora possibile una semplice visualizzazione dei fumi all'interno di essa.

La prima fase per lo sviluppo del metodo ha previsto l'esportazione dei dati dal software utilizzato per la simulazione di incendio.

Innanzitutto, all'interno di Pyrosim (il software utilizzato per la modellazione dell'incendio) è stato necessario impostare come valori di output dei 3Dplot pari 0,25 secondi per poter migliorare la visualizzazione dei dati in virtual reality.

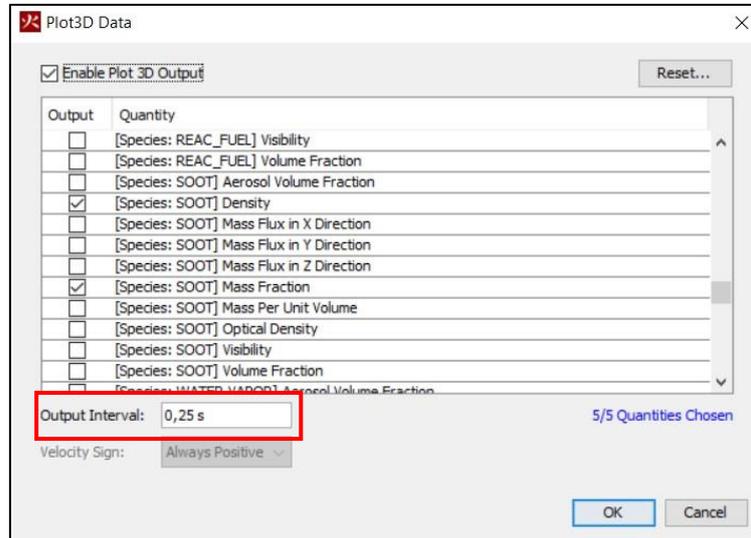


Figura 76 - Modifica dell'intervallo di output su Pyrosim

Si sono esportate le seguenti quantità:

- Soot density
- Soot mass fraction
- Density
- Temperature
- Visibility

La scelta è ricaduta su queste quantità in quanto la visibilità in FDS viene calcolata nel seguente modo:

$$S = \frac{C}{K_m \cdot \rho \cdot Y_{soot}} \quad [m] \quad (1)$$

Dove:

- S: corrisponde alla visibilità
- C: è una costante che dipende dalla segnaletica (valori predefiniti di FDS sono C=8 per un segnale che emette luce e C=3 per un segnale che riflette luce, nel nostro caso è stato utilizzato C=3);

- $K_m$ : corrisponde al coefficiente di estinzione di massa pari a:

$$K_m = \frac{3}{2\rho m} \int_{d_{min}}^{d_{max}} \frac{1}{d} \frac{\Delta m}{\Delta d} Q_{ext} \left( \frac{d}{\lambda}; n_\tau \right) \delta d \quad \left[ \frac{1}{m} \right]$$

Con:

- $\Delta m/\Delta d$ : rappresenta la distribuzione dimensionale della massa;
- $Q_{ext}$ : rappresenta l'efficienza di estinzione della singola particella, ovvero è una funzione del rapporto tra il diametro delle particelle e la lunghezza d'onda della luce ( $d/\lambda$ ) e l'indice di rifrazione complesso della particella ( $n_\tau$ );

In FDS è pari a 8700 m<sup>2</sup>/Kg.

- $Y_{soot}$ : corrisponde alla frazione di massa locale di fuliggine;
- $\rho$  è la massa per unità di volume della miscela fluida pari a:

$$\rho_{CO,9H0,1} = \rho \cdot Y_{soot}$$

Dove  $\rho_{CO,9H0,1}$  rappresenta la massa per unità di volume della fuliggine.

L'equazione (1) descritta precedentemente, deriva da alcuni studi effettuati da Mulholland nel 2002 [25] riguardanti la stima della visibilità attraverso il fumo mediante la seguente equazione:

$$S = \frac{1}{K} \ln \left( \frac{B_{EO}}{\delta_c k L} \right) = \frac{C}{K} \quad [m]$$

Dove:

- $B_{EO}$  rappresenta la luminosità del segnale, misurato in cd/m<sup>2</sup>
- $\delta_c$  è la soglia di contrasto (0,01-0,05)
- $k=9s/K$  (0,4-1)
- $L=1/\pi$  (1 m/m<sup>2</sup>)
- $C$  è una costante adimensionale basata sulla natura dell'oggetto osservato attraverso i media partecipanti.

Per cui è possibile affermare che tre parametri controllano la produzione e la visibilità del fumo:

- Il primo è SOOT\_YIELD, definita come la frazione della massa di combustibile che viene convertita in fuliggine se si utilizza l'approccio chimico semplice. Viene definito dalla seguente relazione:

$$soot\ yield = \frac{mass\ soot}{mass\ fuel} \left[ \frac{Kg}{Kg} \right]$$

- Il secondo parametro, MASS\_EXTINCTION\_COEFFICIENT, è il  $K_m$  e il suo valore predefinito è 8700 m<sup>2</sup> / kg.
- Il terzo parametro, VISIBILITY\_FACTOR è la costante C. È pari a 3 per impostazione predefinita su FDS.

È utile notare che, FDS non può segnalare una visibilità pari ad infinito, ma piuttosto riporta una visibilità massima pari a 30 metri.

Come già detto, il valore del **coefficiente di estinzione di massa** assunto di default in FDS è pari a 8700 m<sup>2</sup>/kg.

Questo valore è consigliato per il fumo prodotto da incendi fiammeggianti secondo studi effettuati da Mukholland e Croarkin [25] effettuati su 29 combustibili differenti. Per il fumo generato dalla combustione senza fiamma, ovvero la pirolisi, il valore varia da 4000 a 5000 m<sup>2</sup>/kg, con il valore minore dovuto al basso assorbimento luminoso del fumo stesso. Tutti gli esperimenti riportati nel documento di Mulholland e Croarking sono stati eseguiti utilizzando una lunghezza d'onda pari a 366 nm, corrispondete al rosso nello spettro visibile.

Widmann nel 2003 ha analizzato i dati per gli incendi fiammeggianti (stechiometrici e iperventilati) e ha trovato la correlazione tra lunghezza d'onda  $\lambda_{nm}$  (nm) e  $K_m$ :

$$K_m = \alpha \cdot \lambda^{-\beta} = 4,8081 \cdot \lambda^{-1,0088}$$

Questa equazione viene rappresentata nel seguente grafico:

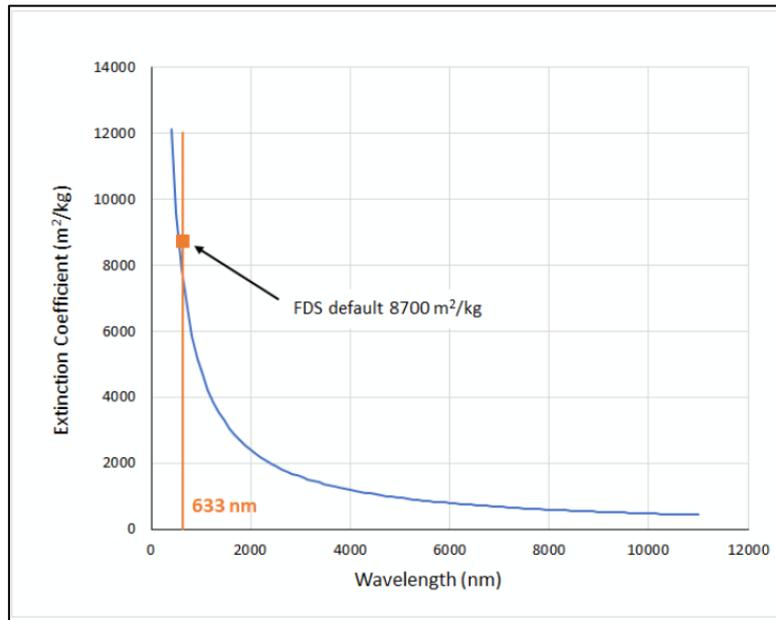


Figura 77 - Coefficiente di estinzione in funzione della lunghezza d'onda  
(<https://support.thunderheadeng.com/tutorials/pyrosim/smoke-visibility-obscuration/>)

Lo spettro della luce visibile si estende da circa 400 nm (viola) a 700 nm (rosso). Valutando la correlazione tra lunghezza d'onda e il coefficiente specifico di estinzione, alla lunghezza d'onda di 633 nm (la lunghezza d'onda che si ottiene con il valore di default di FDS) corrisponde un coefficiente di estinzione pari a 7626 m<sup>2</sup>/Kg, che è inferiore rispetto al valore di default utilizzato da FDS, pari a 8700 m<sup>2</sup>/Kg ma vicino al limite di incertezza inferiore consigliato.

La seconda fase dell'esportazione ha previsto l'importazione dei file 3Dplot all'interno del software Paraview, un'applicazione open source per l'analisi e la visualizzazione grafica dei dati multipiattaforma.

Il software utilizzato per la simulazione di incendio, Pyrosim, al termine della simulazione genera automaticamente dei file in base agli output che abbiamo stabilito inizialmente.

Uno di questi è il file XYZ che contiene informazioni sulla struttura della mesh, e che appunto, dobbiamo aprire all'interno di Paraview. In questo caso, avendo creato tre mesh, avremo tre files XYZ.

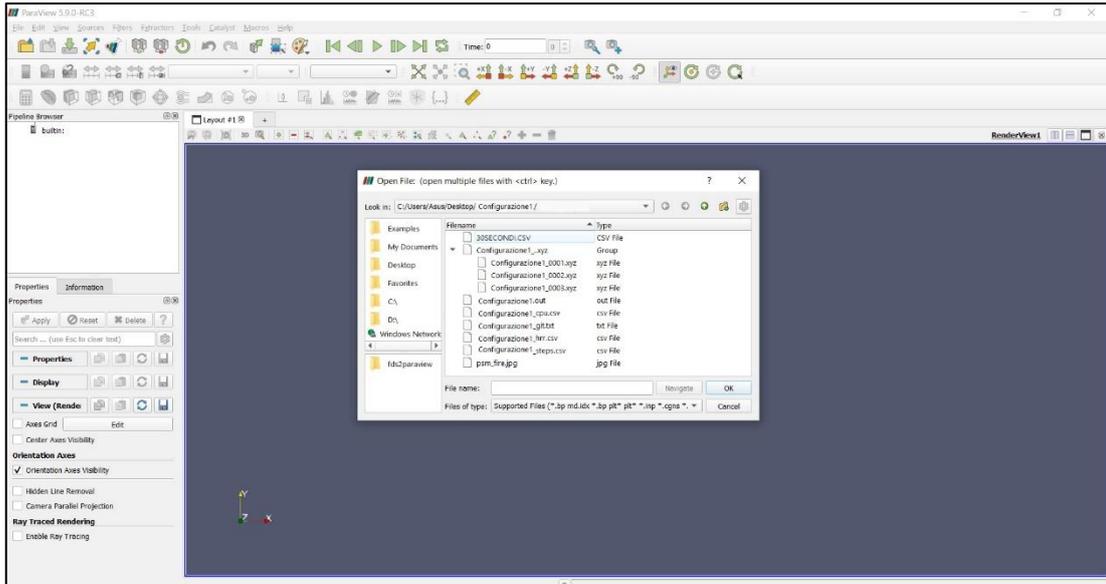


Figura 78 – Importazione dei files all'interno di Paraview

Successivamente, selezionando ciascun file è necessario associare ad essi, i file contenenti le informazioni al variare del tempo.

Per cui, andando nel browser delle proprietà, vengono selezionati file .q per ciascun file in formato .xyz.

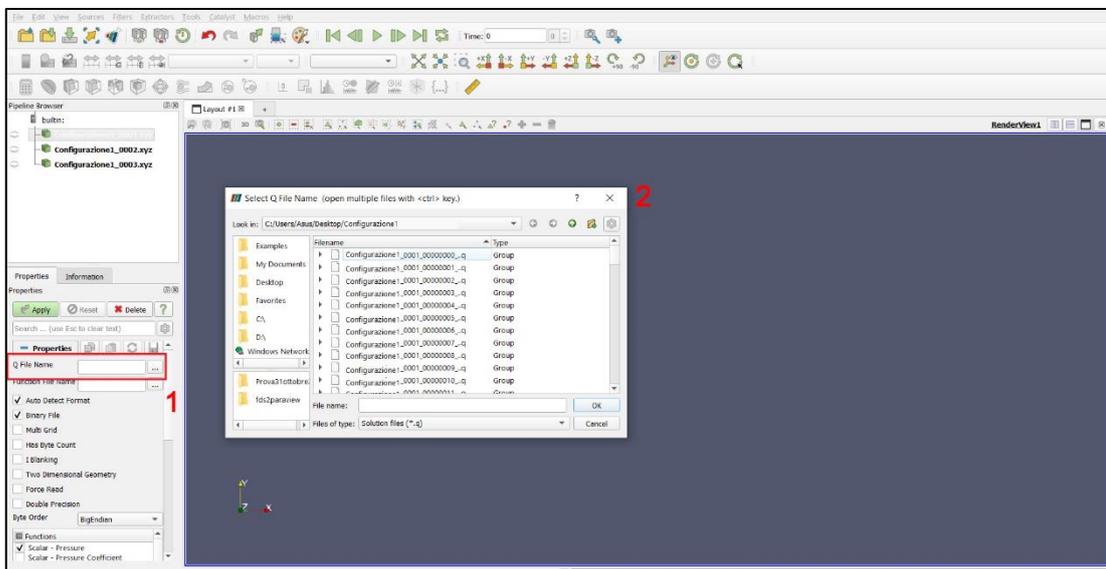


Figura 79 – Inserimento dei files .q

Una volta effettuati questi passaggi, questo sarà il risultato, ovvero saranno visibili solamente i contorni delle mesh create sul software precedente:

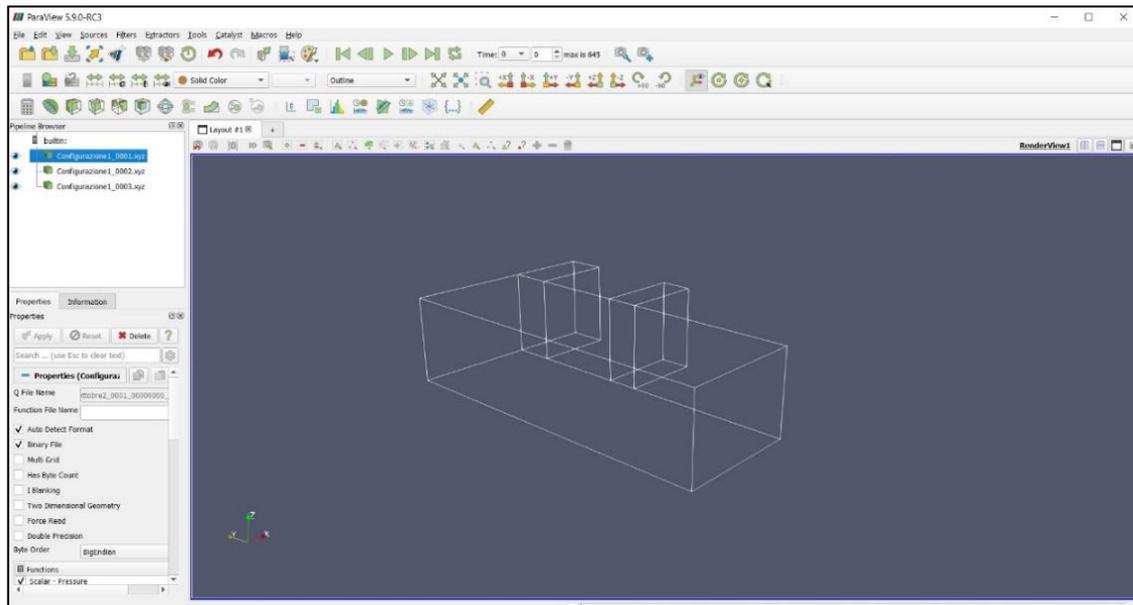


Figura 80 – Visualizzazione delle mesh

Sono presenti un po' di perplessità in merito a come il software Paraview legga i plot3D dati dai file in formato .q, per questo motivo è stata effettuata un'ulteriore verifica, andando ad aprire i file in formato .q con fds2ascii, un'applicazione presente nella cartella di installazione di Pyrosim che permette la lettura dei file FDS come slices (.sf), boundary (.bf) e Plot3D (.q).

Per poter essere utilizzare fds2ascii, è necessario inserirlo nella cartella dove sono presenti i files .q generati automaticamente da Pyrosim.

Per cui, una volta digitato il nome del file “madre”, bisogna scegliere quali file è necessario aprire, slices, boundary o Plot3D, successivamente si individua il dominio ed infine compariranno i file scelti da noi nel primo passaggio.

Nell'immagine sottostante è possibile tutti i passaggi:

```

C:\Users\Asun\Desktop\Pyrosim2\ctobrev\configurazioni\M52ascii.exe
Enter Job ID string (CHID):
Configurazione1
What type of file to parse?
FL2D file? Enter 1
SLCF file? Enter 2
IBDF file? Enter 3

Enter sampling factor for Data?
(1 for all data, 2 for every other point, etc.)
1

Domain selection:
? - domain size is limited
? - domain size is not limited
? - domain size is not limited and 2 levels are offset
Yes, no or all - slice files are selected based on type and location.
The y, n, z prefix are defined as before.

1 Configurazione1_0001_00000000_34.0, MESH 1, Time: 0.
2 Configurazione1_0002_00000000_34.0, MESH 2, Time: 0.
3 Configurazione1_0003_00000000_34.0, MESH 3, Time: 0.
4 Configurazione1_0004_00000000_51.0, MESH 1, Time: 1.
5 Configurazione1_0007_00000000_51.0, MESH 2, Time: 1.
6 Configurazione1_0001_00000000_51.0, MESH 3, Time: 1.
7 Configurazione1_0001_00000000_86.0, MESH 1, Time: 1.
8 Configurazione1_0002_00000000_86.0, MESH 2, Time: 1.
9 Configurazione1_0003_00000000_86.0, MESH 3, Time: 1.
10 Configurazione1_0001_00000001_01.0, MESH 1, Time: 1.
11 Configurazione1_0002_00000001_01.0, MESH 2, Time: 1.
12 Configurazione1_0003_00000001_01.0, MESH 3, Time: 1.
13 Configurazione1_0001_00000001_38.0, MESH 1, Time: 1.
14 Configurazione1_0002_00000001_38.0, MESH 2, Time: 1.
15 Configurazione1_0003_00000001_38.0, MESH 3, Time: 1.
16 Configurazione1_0001_00000001_55.0, MESH 1, Time: 2.
17 Configurazione1_0002_00000001_55.0, MESH 2, Time: 2.
18 Configurazione1_0003_00000001_55.0, MESH 3, Time: 2.
19 Configurazione1_0001_00000001_89.0, MESH 1, Time: 2.
20 Configurazione1_0002_00000001_89.0, MESH 2, Time: 2.
21 Configurazione1_0003_00000001_89.0, MESH 3, Time: 2.
22 Configurazione1_0001_00000002_41.0, MESH 1, Time: 2.
23 Configurazione1_0002_00000002_41.0, MESH 2, Time: 2.
24 Configurazione1_0003_00000002_41.0, MESH 3, Time: 2.
25 Configurazione1_0001_00000002_76.0, MESH 1, Time: 3.
26 Configurazione1_0002_00000002_76.0, MESH 2, Time: 3.
27 Configurazione1_0003_00000002_76.0, MESH 3, Time: 3.
28 Configurazione1_0001_00000002_76.0, MESH 1, Time: 3.
29 Configurazione1_0002_00000002_58.0, MESH 2, Time: 3.
30 Configurazione1_0003_00000002_58.0, MESH 3, Time: 3.
31 Configurazione1_0002_00000002_76.0, MESH 2, Time: 3.
32 Configurazione1_0002_00000001_18.0, MESH 2, Time: 3.
33 Configurazione1_0002_00000001_27.0, MESH 2, Time: 3.
34 Configurazione1_0003_00000001_18.0, MESH 3, Time: 3.
35 Configurazione1_0003_00000001_27.0, MESH 3, Time: 3.

```

Figura 81 – Interfaccia di fds2ascii

Una volta scelto il file, viene generato un singolo file con il nome CHID\_fds2ascii.csv. Dall'apertura di quest'ultimo file, è stato possibile confrontare le variabili con quelle del software Paraview; in questo caso, infatti, le variabili da noi scelte con output in Pyrosim vengono denominate come segue:

Tabella 16 - Denominazione delle quantità su Paraview

PYROSIM	PARAVIEW	UNITA' DI MISURA
Soot Density	Density	kg/m3
Mass Fraction	Momentum x	kg/kg
Density	Momentum y	kg/m3
Temperature	Momentum z	°C
Visibility	Stagnation Energy	m

Infatti, dall'immagine seguente è possibile vedere come Paraview rinomina le variabili:

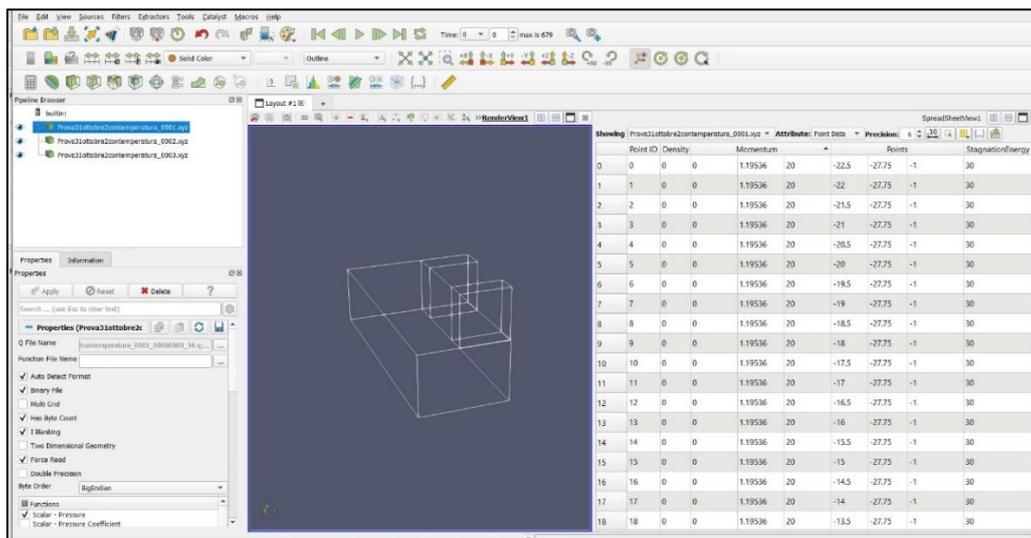


Figura 82 – Visualizzazione dei dati su Paraview

Per la fase successiva, in base all'elemento da esportare relativo alla simulazione di incendio, isosuperfici o slices, la metodologia utilizzata è differente. Nei paragrafi successivi, analizziamo dettagliatamente le due metodologie utilizzate.

### 7.1.2. Isosuperfici

L'isosuperficie è una superficie tridimensionale che rappresenta punti di valore costante, come per esempio, pressione, temperatura, velocità ecc. all'interno di un volume di spazio. La metodologia utilizzata per l'esportazione delle isosuperfici viene descritta successivamente e riassunta nell'immagine di seguito:

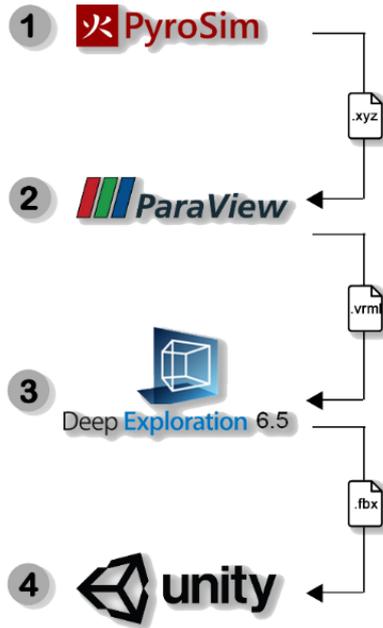


Figura 83 - Metodologia

Una volta capito come funziona il software, è possibile creare le isosuperfici cliccando con il tasto destro sul file in cui ci interessa crearla, selezionando *Add Filter* → *Alphabetical* → *Contour*, come mostrato di seguito:

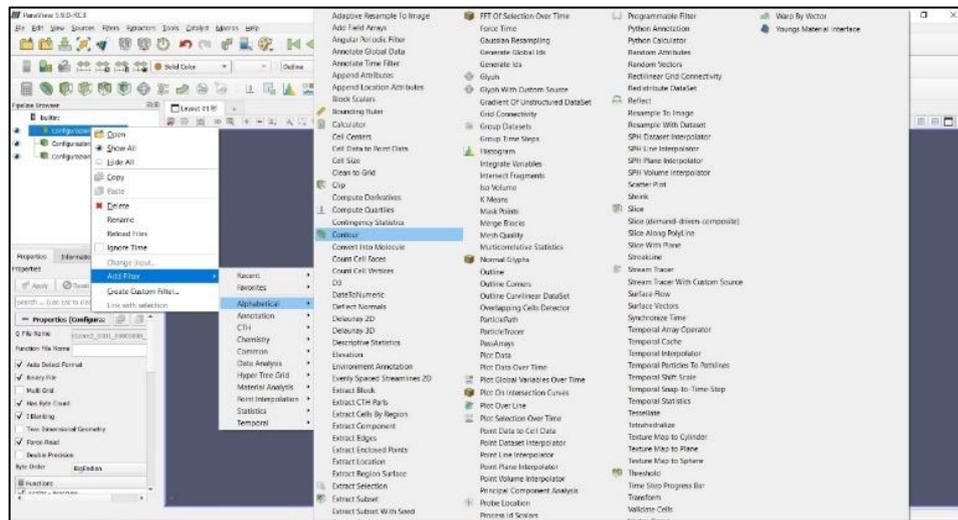


Figura 84 – Creazione dell'isosuperficie

Ai fini degli obiettivi di questa tesi, è stato ritenuto opportuno creare una isosuperficie della Soot Density quindi della Density in Paraview del valore pari a  $3,5 \cdot 10^{-5}$  corrispondete ad una visibilità di 10 metri.

Infatti, utilizzando la formula (1) si ottiene:

$$Y_{soot} \cdot rho = \frac{C}{S \cdot Km}$$

Quindi:

$$rho_{CO,9H0,1} = \frac{C}{S \cdot Km} = \frac{3}{10 \cdot 8700} = 0,00003448 \frac{kg}{m^3}$$

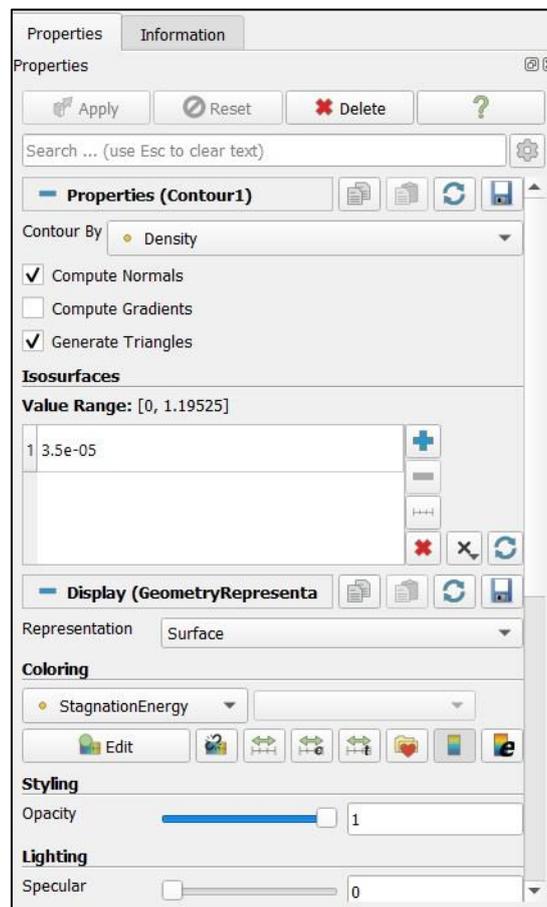


Figura 85 - Impostazioni per la creazione dell'isosuperficie in Paraview

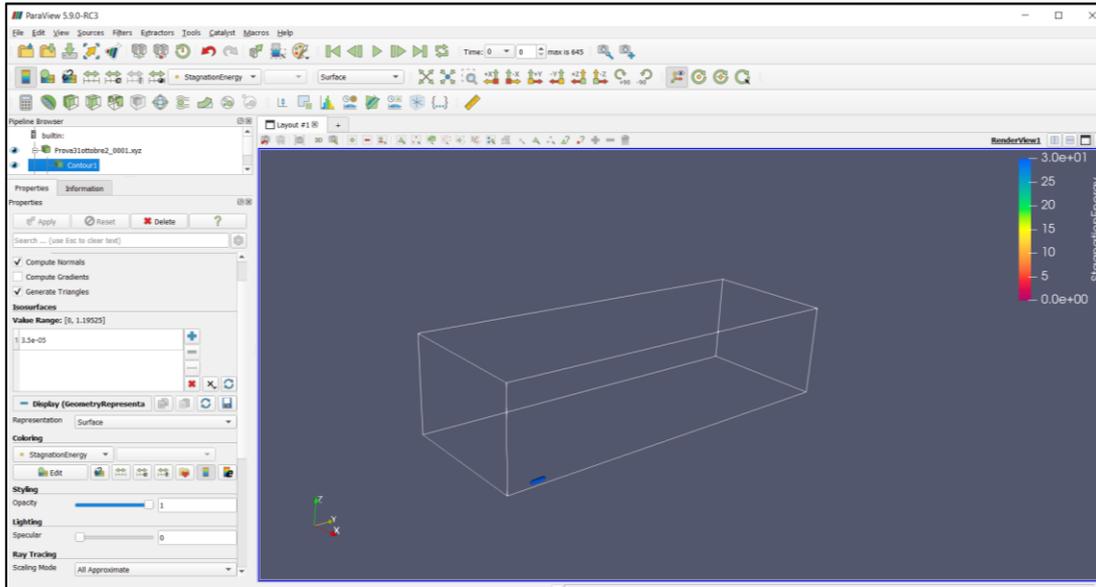


Figura 86 - Visualizzazione della isosuperficie all'istante 0

Gli istanti di tempo disponibili sono 645 perché come detto in precedenza, l'intervallo di output impostato su Pyrosim è pari a 0,25 secondi. La simulazione di incendio è stata interrotta a 161 secondi in quanto i tempi computazionali erano molto lunghi.

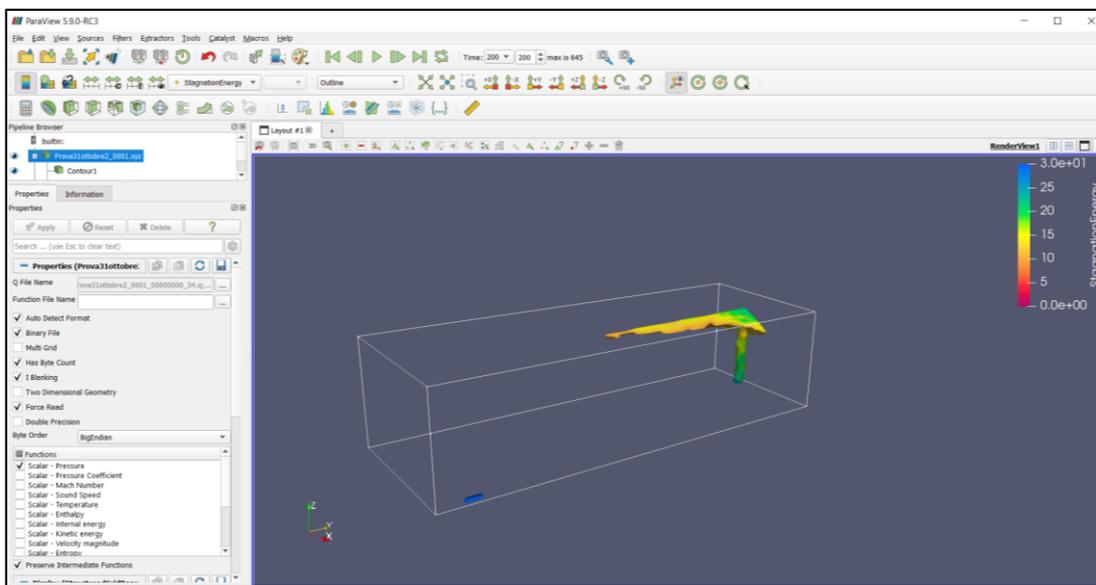


Figura 87 – Visualizzazione della isosuperficie all'istante 200 corrispondete a 50 secondi

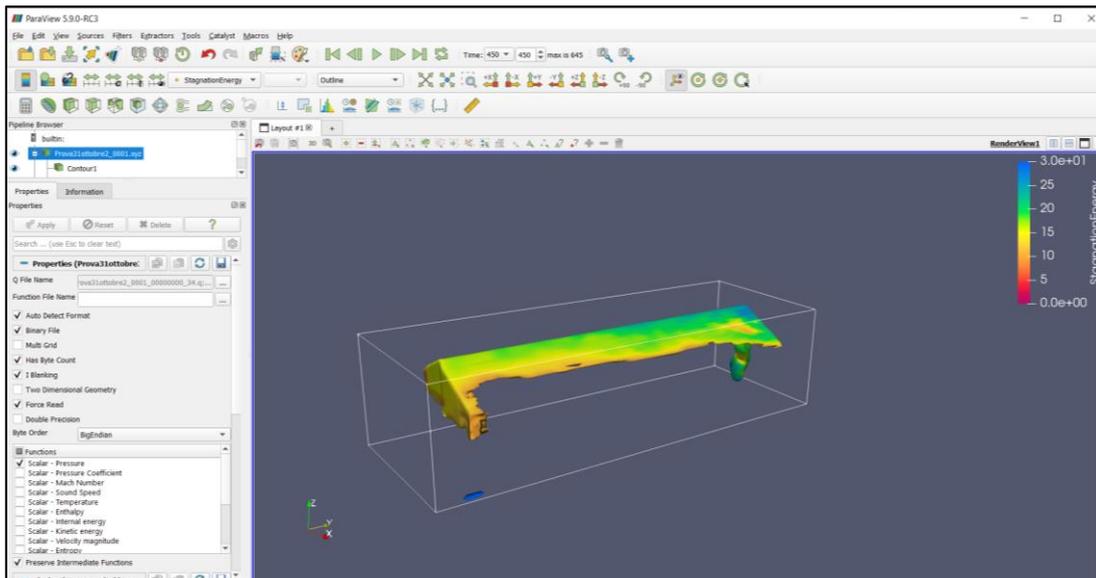


Figura 88 - Visualizzazione della isosuperficie all'istante 450 corrispondente a 112 secondi

A causa dei numerosi file da esportare, è stato necessario automatizzare il processo mediante il software Python.

Il software Paraview permette di trasformare le azioni effettuate sul programma in un codice in linguaggio Python, mediante il comando *Start Trace*:

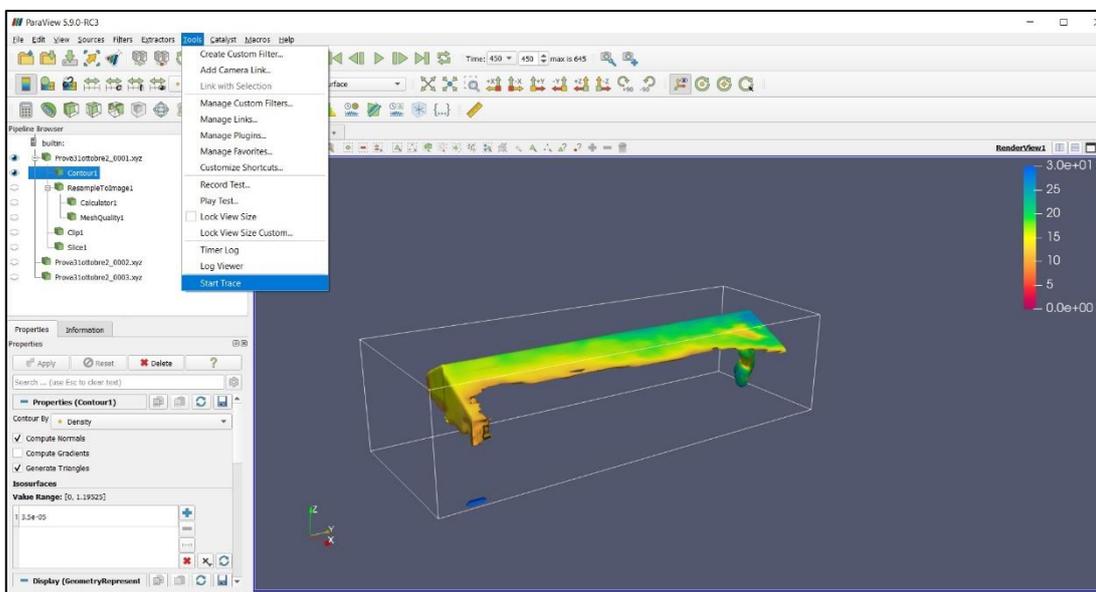


Figura 89 – Trasformazione delle azioni in linguaggio Python

Allo script (paragrafo A.1) risultante da questa operazione è stato necessario aggiungere un *ciclo for* che ha permesso al software di eseguire l'esportazione di ciascun istante di tempo ed inoltre, di rinominare il file esportato con il relativo istante di tempo associato.

Qui di seguito è possibile visionare lo stralcio di script per la creazione del ciclo for:

```

1. for x in numpy.arange(1.0, 651.0, 1.0):
2.     animationScene1.AnimationTime = x
3.
4.     # get the time-keeper
5.     timeKeeper1 = GetTimeKeeper()
6.
7.     # save data
8.     timeIstant = int(x)
9.
10.    SaveData(f'C:/Users/Asus/Desktop/fds2paraview/Esportazione PARAVIEW/ply files/{timeIstant}.ply',

```

*Stralcio di script utilizzato per l'esportazione*

Il ciclo for è costituito dalla funzione arange:



I parametri della funzione `arange` sono tre:

- Start: il punto iniziale dell'interazione;
- Stop: il punto finale dell'interazione;
- Step: il passo dell'avanzamento quindi di quanto incrementa la variabile `x` ad ogni interazione.

Una volta modificato lo script, è possibile avviarlo aprendo la Python Shell mediante *View* → *Python Shell*. Una volta aperta questa finestra cliccando su *Run Script* bisogna selezionare lo script appena creato.



Figura 90 - Python Shell in Paraview

Paraview permette l'esportazione di numerosi formati, nel caso in esame, è stato scelto di esportare il formato .vrml.

Sfortunatamente Unity3D non prevede la lettura di nessun formato disponibile per l'esportazione in Paraview, per cui è necessario ricorrere ad un ulteriore programma.

Il software utilizzato, per quest'ultima fase di esportazione è Deep Exploration.

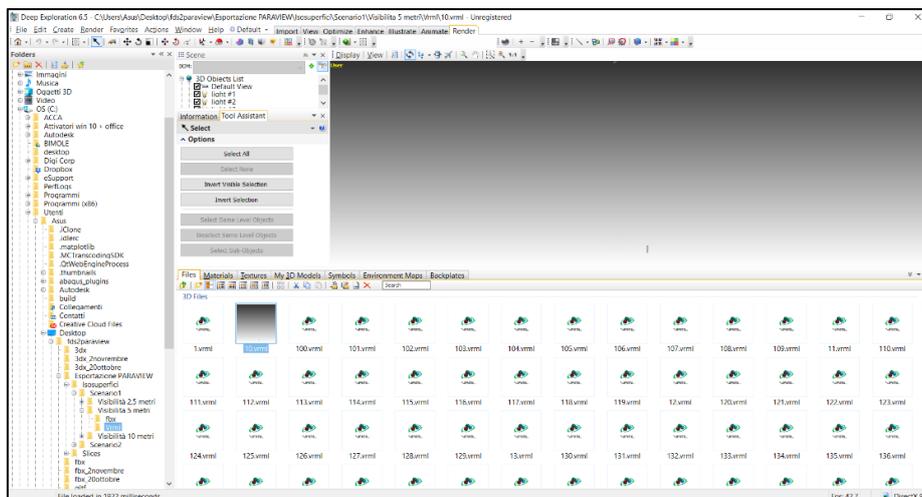


Figura 91 – Interfaccia di Deep Exploration

Selezionando il file e cliccando il tasto destro del mouse, bisogna selezionare *Batch Operations* → *Convert 3D files*.

La versione gratuita del software non permette l'esportazione di tutti i files contemporaneamente ma ogni 25 files, per cui è stato necessario ripetere l'operazione diverse volte.

Il formato scelto per l'esportazione è .fbx,

Le isosuperfici che si è deciso di esportare, in questo caso studio, sono le seguenti:

- **Soot Density pari a  $1,4e^{-4}$  kg/m<sup>3</sup>** corrispondente ad una visibilità di **2,5 metri**;
- **Soot Density pari a  $6,9e^{-5}$  kg/m<sup>3</sup>** corrispondente ad una visibilità di **5,0 metri**;
- **Soot Density pari a  $3,5e^{-5}$  kg/m<sup>3</sup>** corrispondente ad una visibilità di **10 metri**.

Per ciascuna isosuperficie corrispondente ad un specifico intervallo di tempo, è stato necessario crearla su Paraview, esportarla mediante lo script precedentemente descritto in formato .vml, importarla su Deep Exploration ed infine esportarla in formato .fbx.

## 7.1.3. Slices

Le slices 2D sono dei piani di taglio che misurano i dati della fase gassosa (ad esempio, pressione, temperatura, velocità) su un asse allineato aereo.

La metodologia utilizzata per l'esportazione delle slices viene descritta successivamente e riassunta nell'immagine seguente:

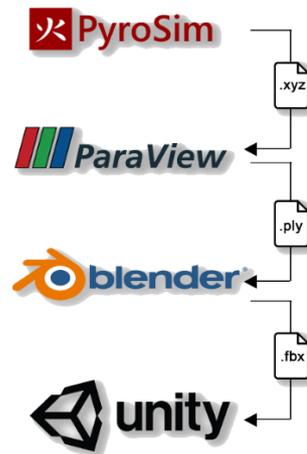


Figura 92 – Metodologia

Per quanto riguarda le slices, è possibile crearle su Paraview cliccando con il tasto destro sul file in cui ci interessa crearla, selezionando

*Add Filter* → *Alphabetical* → *Slice*, come mostrato di seguito:

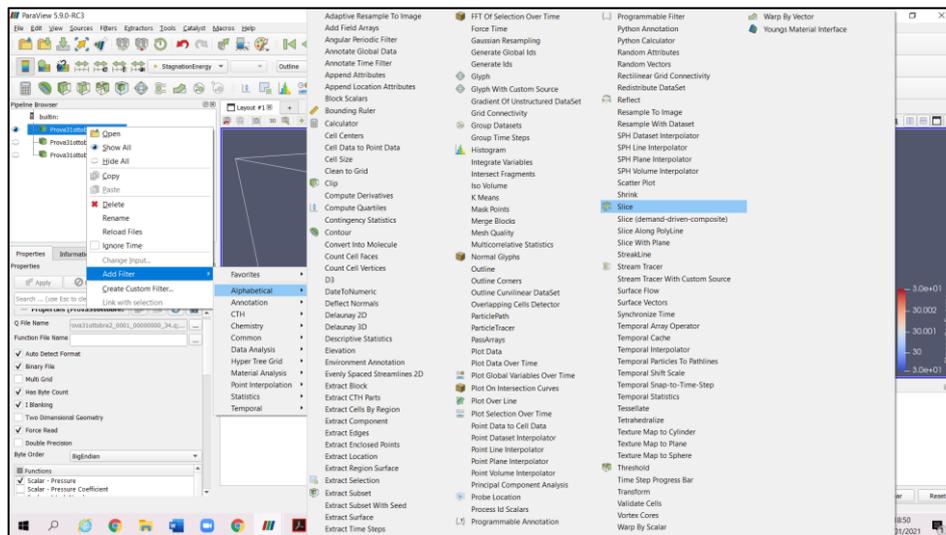


Figura 93 – Creazione delle slices su Paraview

In base alla posizione del focolare, sono state scelte due slices significative.

Nelle immagini seguenti vengono mostrate le slices scelte all'istante di tempo 400 corrispondente a 100 secondi.

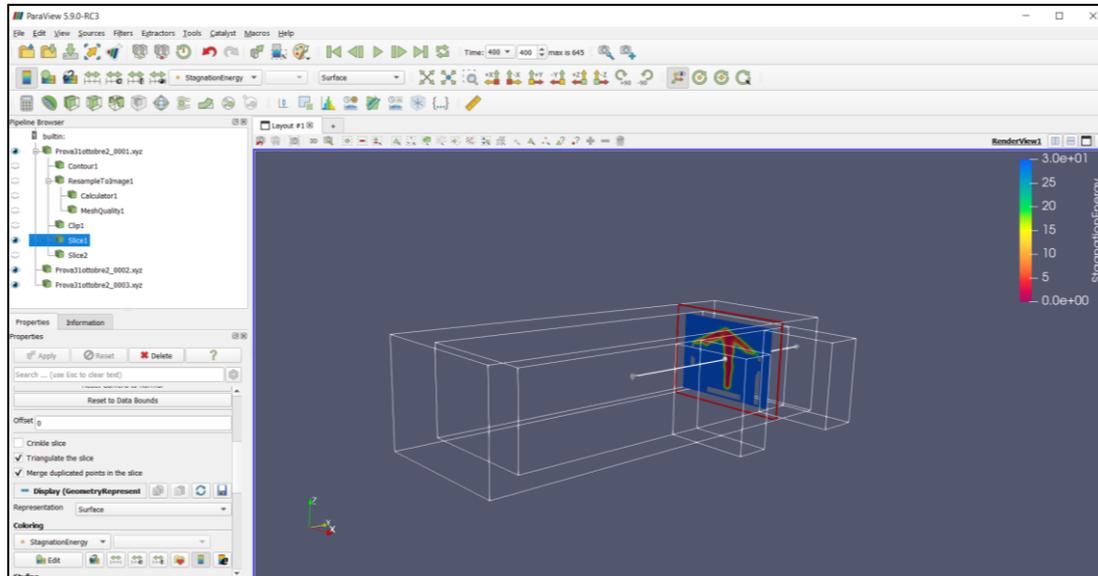


Figura 94 - Visualizzazione della prima slice di visibilità su Paraview

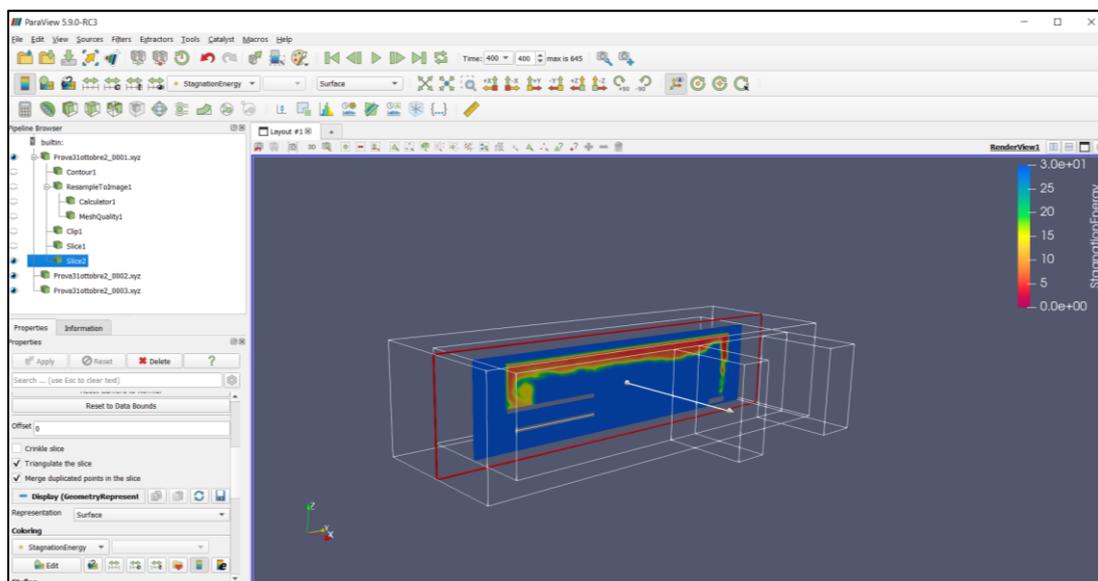


Figura 95 - Visualizzazione della seconda slice di visibilità su Paraview

Inoltre, le quantità che si è deciso di analizzare sono essenzialmente due:

- Visibilità
- Temperatura

Per quanto riguarda le proprietà impostate per le due slices sono le seguenti:

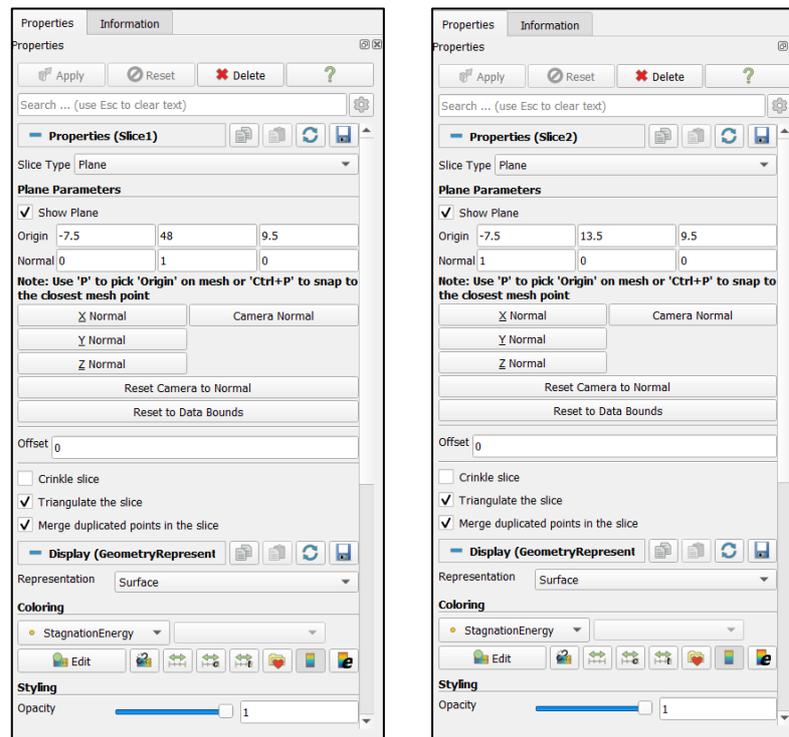


Figura 96 - Proprietà impostate rispettivamente per le slices 1 e 2 di visibilità

Come per le isosuperfici, anche per le slices è stato creato uno script per automatizzare il processo di esportazione (paragrafo A.2), conforme a quello descritto nel paragrafo precedente.

Per l'esportazione delle slices, dopo numerose prove, è stato scelto il formato .ply in quanto permette di esportare anche i dati dei colori assegnati al vertice di ogni mesh. Anche in questo caso, come per le isosuperfici, è stato necessario ricorrere ad un ulteriore programma, in quanto Paraview non permette l'esportazione di nessun formato utilizzabile in Unity.

Il programma utilizzato in questo caso è Blender, un software libero e multiplatforma di modellazione, rigging, animazione, montaggio video, composizione e rendering di immagini tridimensionali e bidimensionali. Dispone inoltre di funzionalità per mappature UV, simulazioni di fluidi, di rivestimenti, di particelle, altre simulazioni non lineari e creazione di applicazioni/giochi 3D.

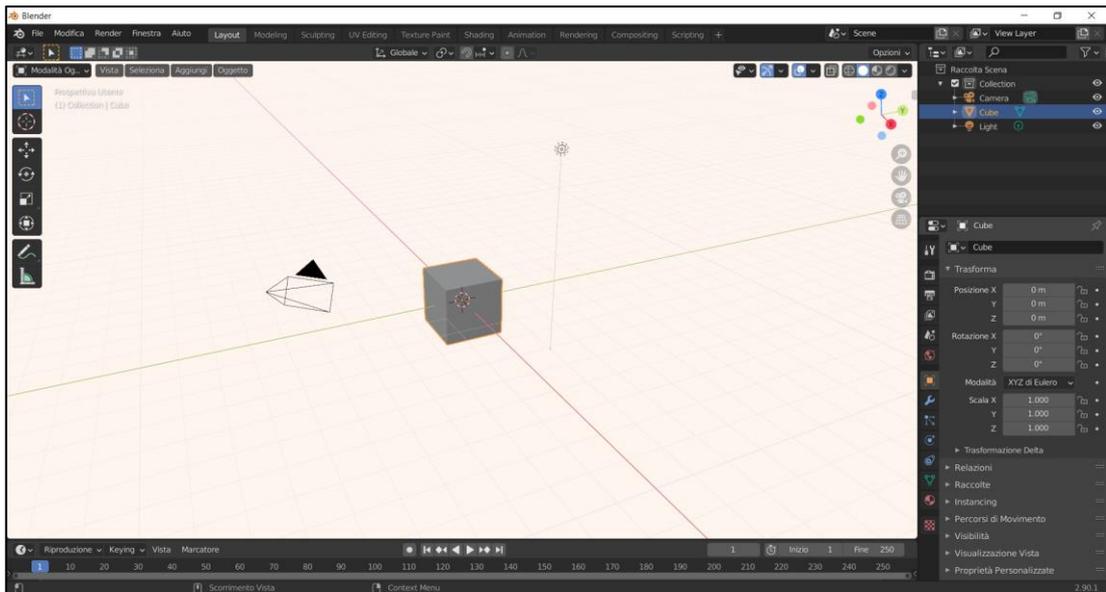


Figura 97 - Schermata principale del software Blender

Per prima cosa, nel menu *File* a tendina selezioniamo *Importa*, clicchiamo su Stanford (.ply) e selezioniamo tutti i file che vogliamo importare all'interno di Blender.

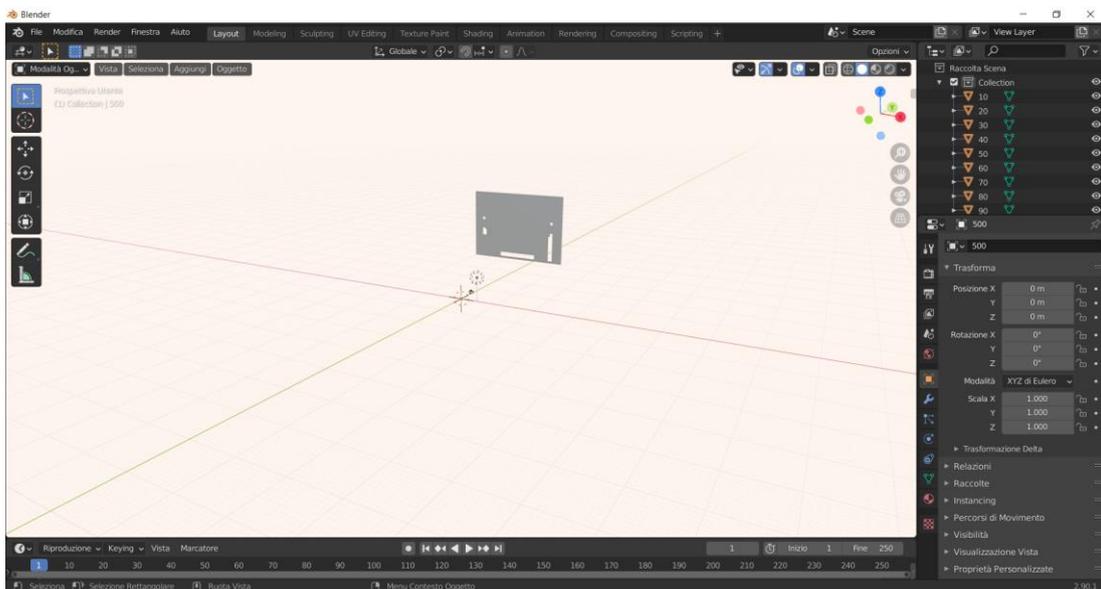


Figura 98 - Visualizzazione degli oggetti importati su Blender

Com'è possibile notare dall'immagine sovrastante, l'oggetto importato non presenta colori. In realtà, i colori sono presenti ma non sono visibili questo avviene perché sono memorizzati ai vertici della mesh.

Infatti, selezionando l'oggetto, nel menù delle proprietà compare la voce "Col" com'è possibile vedere nell'immagine seguente:

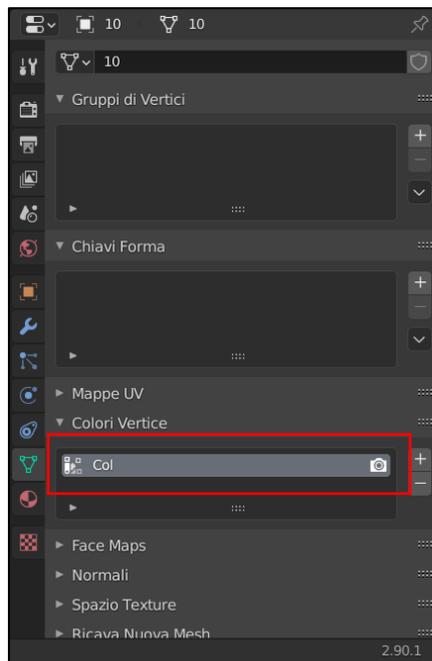


Figura 99 - Visualizzazione della finestra proprietà materiali

Per ottenerli è necessario creare uno shader materiale che utilizzi il colore di input, per fare ciò è stato creato il seguente script:

```

1. import bpy
2.
3. scene = bpy.context.scene
4.
5. for idx ,o in enumerate(scene.objects):
6.     if o.type == 'MESH':
7.         mat = bpy.data.materials.new("My Material_"+str(idx)) #to make your
           personal index into name (if you want)
8.         mat.use_nodes = True
9.         nodes = mat.node_tree.nodes
10.        node_attribute = nodes.new(type="ShaderNodeAttribute")
11.        node_attribute.location = (-1405.21, -1784.65)
12.        node_attribute.attribute_name = "Col"
13.
14.        o.data.materials.append(mat)

```

Script utilizzato per creare uno shader materiale

Selezionando tutti gli oggetti importati e cliccando su *Esegui Script* nella finestra *Scripting*, si attribuisce il colore al vertice ad ogni elemento.

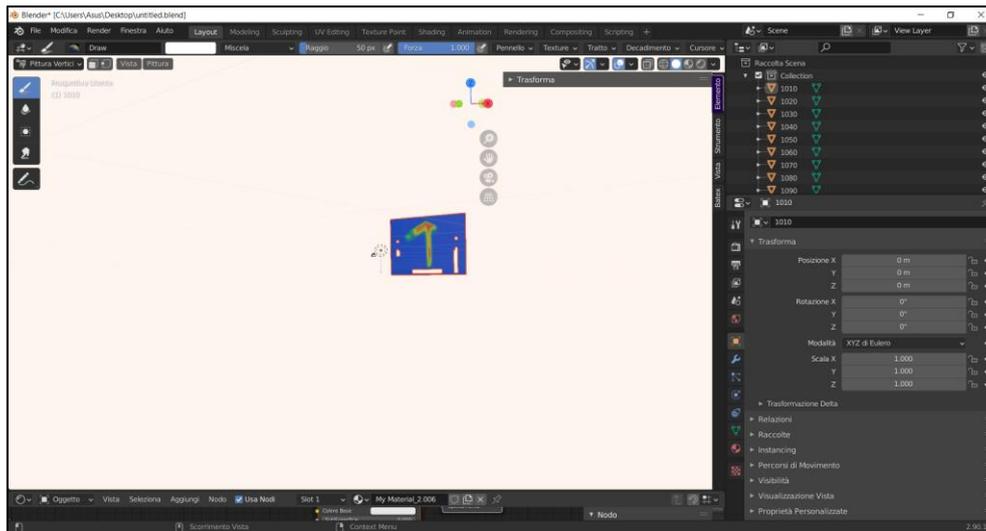


Figura 100 - Esecuzione dello script e attribuzione dei colori al vertice

Anche in questo caso, la grande quantità di file da esportare non ha permesso di eseguire l'esportazione manualmente, per cui è stato necessario installare un plug-in chiamato Batch FBX Export che ha permesso di automatizzare il processo di esportazione.

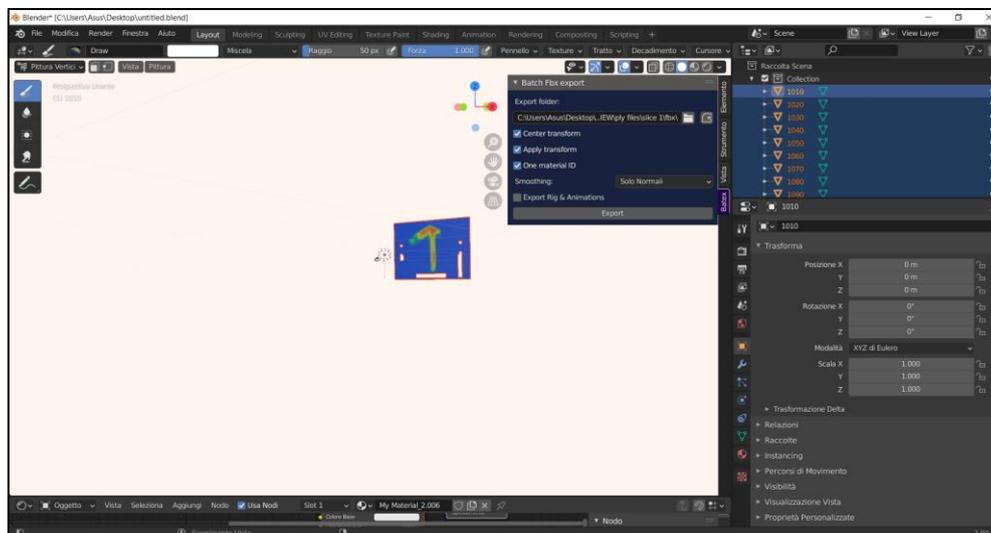


Figura 101 - Esportazione dei files in formato .fbx

Al termine di questo capitolo, si è conclusa la fase di esportazione, per cui, si hanno a disposizione i files in grado di essere importati e gestiti all'interno della piattaforma utilizzata per realizzare l'applicativo.

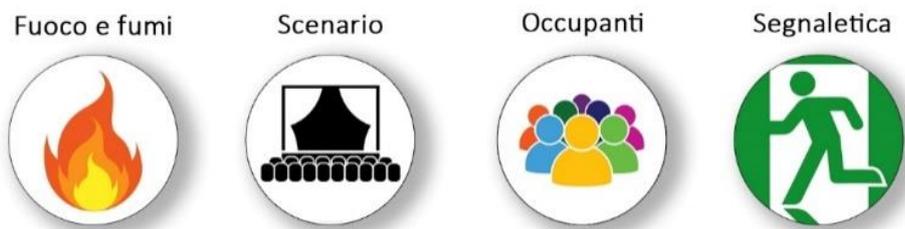
## Capitolo 8

### Visualizzazione dei risultati in Virtual Reality

L'utilizzo della realtà virtuale come strumento di ricerca nella tecnologia antincendio e nell'evacuazione si sta sempre più affermando nel mondo.

Una volta ottenuti i file dall'esportazione, è necessario importarli sul software utilizzato per la visualizzazione in Virtual Reality. Successivamente, verranno descritte le fasi per la creazione dell'applicativo che sarà da supporto ai Vigili del Fuoco e che permetterà il controllo contemporaneo di:

- Fumi e fuoco
- Scenario
- Occupanti
- Sistemi di segnaletica



#### 8.1. Interfaccia grafica di Unity

L'ambiente di sviluppo di Unity è costituito da un motore grafico, un motore fisico molto potente e un live game preview. Mediante questo software è possibile realizzare videogames 3D o altri contenuti interattivi.

Unity viene definito “multiplatforma” in quanto il suo motore permette di “scrivere il giorno” una sola volta e realizzarlo o trasformarlo per ambienti o circuiti differenti come PC, Play Station, iOS ecc.

L'interfaccia di Unity è composta da sei pannelli base, ciascuno nato per svolgere delle funzioni ben precise nella progettazione del videogioco.

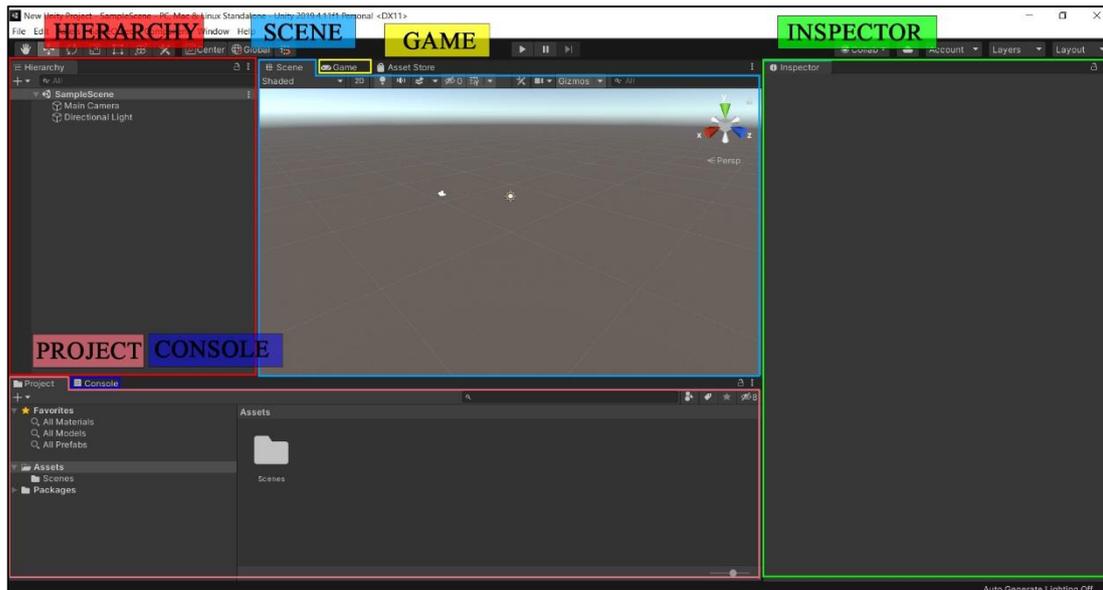


Figura 102 - Interfaccia grafica di Unity3D

- **Hierarchy:** essa è la finestra che contiene l'elenco dei GameObject, ovvero gli oggetti, presenti nella scena corrente.  
 Quando un oggetto viene aggiunto o viene rimosso dalla scena, essi appaiono o scompaiono rispettivamente dalla finestra Hierarchy.  
 Unity utilizza un concetto chiamato Parenting che consiste nella creazione di una relazione di parentela fra due o più oggetti. Infatti, basta trascinare il nome di un oggetto dentro un altro per creare la parentela.  
 L'oggetto esterno verrà chiamato padre mentre quello all'interno sarà il figlio.  
 In questo modo, quando sposteremo l'oggetto padre, verrà seguito dall'oggetto figlio.  
 Inoltre, mediante il comando *Create* è possibile inserire nuovi oggetti all'interno della scena.
- **Scene:** questa finestra contiene gli ambienti, come la camera o gli elementi tridimensionali, i menu del gioco e le luci.  
 Quando viene creato un nuovo Unity Project, la finestra della scena è vuota ad eccezione della Camera (chiamata Main Camera) e della luce (chiamata Directional Light).  
 All'interno di una scena è possibile importare elementi creati con altri software o inserire oggetti direttamente da Unity.

- **Game:** questa finestra mostra la scena che viene inquadrata dalla Main Camera, ovvero la telecamera principale. L'obiettivo di questa sezione è quello di fornire in tempo reale il risultato di ciò che si sta modellando.
- **Inspector:** in questa finestra vengono raccolte tutte le impostazioni e i parametri relativi all'oggetto selezionato, come ad esempio, gli script ad esso collegato e tutte le caratteristiche che l'oggetto possiede.

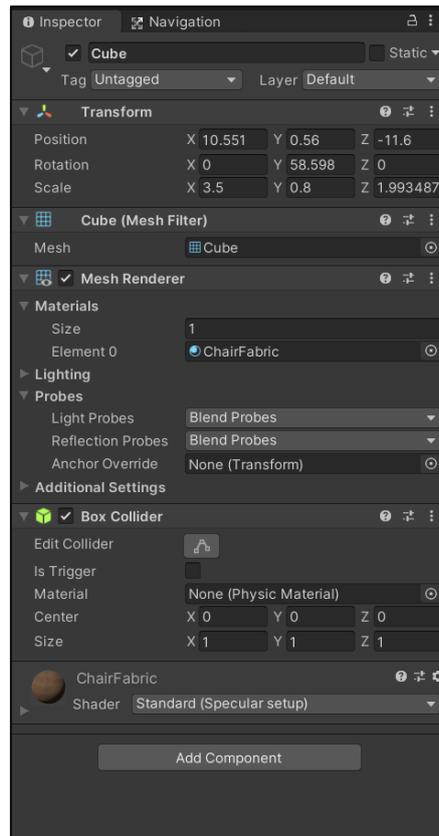


Figura 103 - Esempio della finestra Inspector relativa ad un GameObject

- **Project:** in questo pannello si trovano gli assets, ovvero tutti gli elementi che useremo nel nostro gioco come texture, modelli 3D, materiali, scene, script, effetti sonori ecc.

Alla sinistra del pannello è presente un menù di navigazione dal quale possiamo accedere facilmente alle directory dei files.

- **Console:** la finestra Console mostra gli errori, gli avvisi e altri messaggi generati da Unity.

## 8.2. Concetti fondamentali

Nel paragrafo precedente è stato analizzato l'aspetto grafico ed interattivo di Unity, chiamato Unity Editor; in questo paragrafo, invece, verrà analizzato il motore di sviluppo chiamato Unity Engine, ma soprattutto le funzioni principali utilizzate per la creazione dell'applicativo.

In Unity, prevalgono tre concetti fondamentali:

- **GameObject:** sono gli oggetti fondamentali in Unity, rappresentano personaggi, oggetti di scena e scenari.

Ogni GameObject ha di default, un *Transform* che definisce la posizione, la rotazione e la grandezza dell'oggetto secondo un sistema di coordinate xyz.

GameObject	Component	Window	Help
Create Empty		Ctrl+Shift+N	
Create Empty Child		Alt+Shift+N	
3D Object			>
2D Object			>
Effects			>
Light			>
Audio			>
Video			>
UI			>
Camera			
Center On Children			
Make Parent			
Clear Parent			
Set as first sibling		Ctrl+=	
Set as last sibling		Ctrl+-	
Move To View		Ctrl+Alt+F	
Align With View		Ctrl+Shift+F	
Align View to Selected			
Toggle Active State		Alt+Shift+A	

Figura 104 - Finestra GameObject

- **Component:** sono i componenti aggiunti di ogni *GameObject*, infatti, il *GameObject* è un contenitore per i diversi componenti.

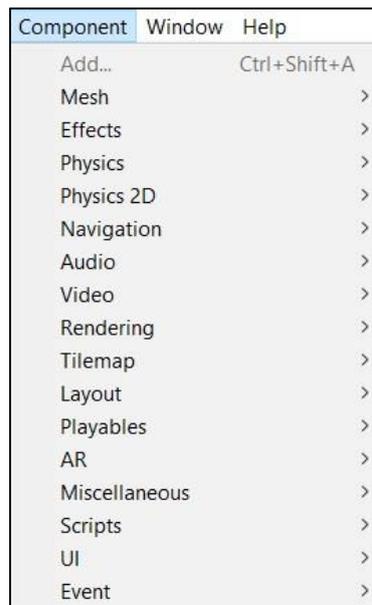


Figura 105 - Finestra Component

I componenti più utilizzati e più importanti sono:

- Collider: permette la collisione e l'interazione tra gli oggetti presenti nella scena; esistono numerose classi di Collider come *BoxCollider* che prevede uno di forma scatolare, *SphereCollider* che prevede uno di forma sferica, *MeshCollider*, invece, prevede la creazione di un collider in base alla mesh applicata all'oggetto ed infine, *CapsuleCollider*, ne prevede uno di forma di capsula;
  - Script: è un componente che permette di attribuire ad un oggetto un determinato script;
  - AudioSource: permette di riprodurre suoni 3D e suoni 2D, questi ultimi sono costanti, ovvero, in qualsiasi luogo si trovi il giocatore nella scena, il volume di riproduzione sarà sempre lo stesso. Invece, i suoni 3D simulano i comportamenti del suono nella vita reale, infatti, più ci si avvicina alla fonte sonora più il suono sarà forte e viceversa;
  - Rigidbody: permette di attribuire caratteristiche come la gravità, l'accelerazione e l'attrito, è un componente personalizzabile, in quanto, permette di definire la velocità, il centro di massa, la massa e la modalità di collisione.
- Prefab: sono dei GameObject prefabbricati.

### 8.3. Importazione dei dati

#### 8.3.1. Importazione del modello BIM

All'interno di Unity3D è stato importato il modello dell'edificio esportato in formato .IFC dal software Autodesk Revit.

L'importazione in Unity3D può avvenire in diversi modi in funzione degli obiettivi finali, ma soprattutto della complessità dell'edificio. I metodi più comuni per l'importazione sono l'utilizzo di software intermedi come 3dsMax o SimLabSoft tramite cui è possibile conservare ed attribuire un materiale a ciascun elemento tridimensionale.

Nel caso in esame, dopo numerose prove effettuate, è stato deciso di utilizzare il plug-in di Unity, chiamato Tridify BIM Tools.

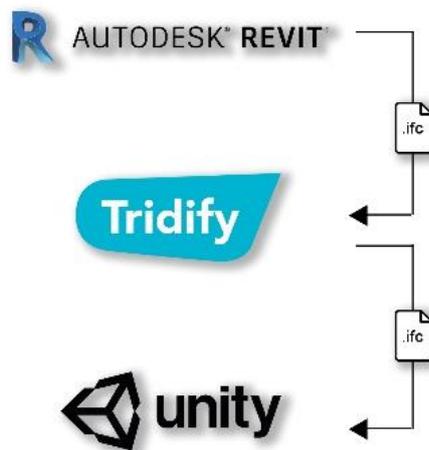


Figura 106 - Metodologia di importazione del modello BIM

Per poter utilizzare questo plug-in è necessario:

- Aprire un account del servizio di pubblicazione BIM Tridify
- Caricare l'IFC interessato; in questo modo, il file viene elaborato

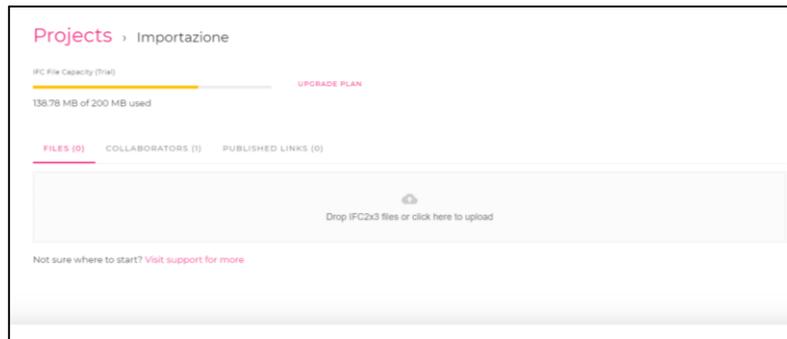


Figura 107 - Schermata principale del servizio di pubblicazione di BIM Tridify

- Scaricare mediante l'Assets Store il plug-in di Tridify;
- Mediante il plug-in installato sarà possibile caricare il file precedentemente elaborato.

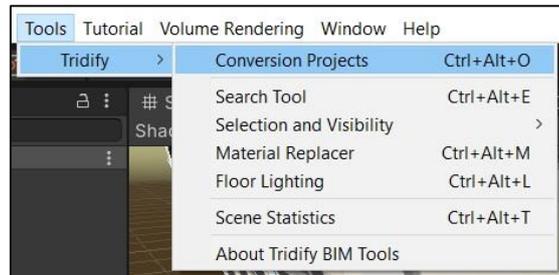


Figura 108 - Plug-in Tridify in Unity3D

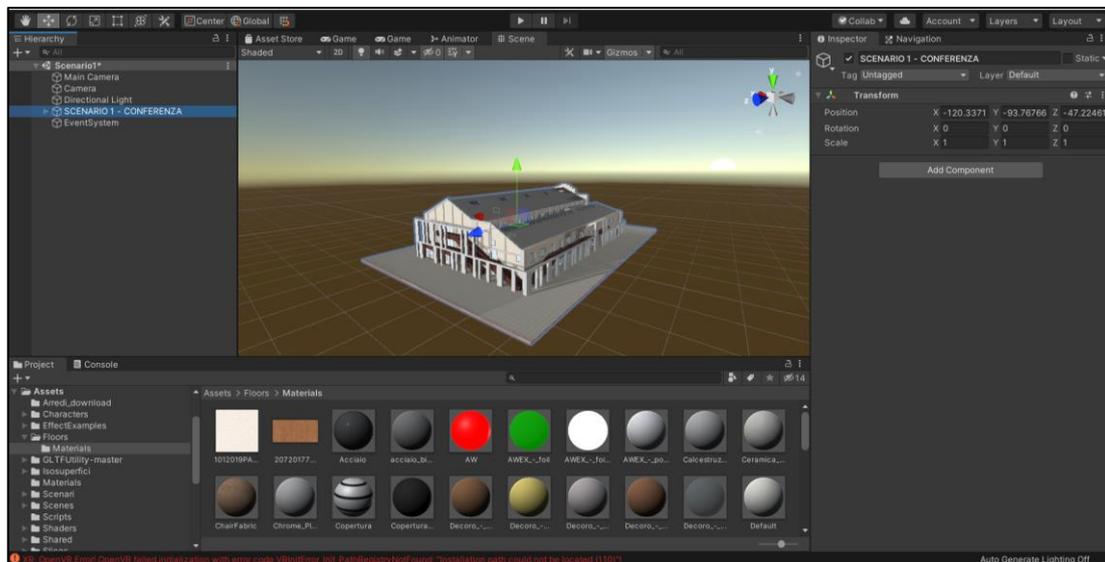


Figura 109 - Modello dell'edificio importato mediante Tridify

Com'è possibile notare dall'immagine sovrastante, il modello è stato correttamente importato con i rispettivi materiali. È stato necessario modificare o aggiungere direttamente in Unity alcuni di essi, in quanto il plug-in non li ha riconosciuti.

### 8.3.2. Importazione della simulazione di incendio

Nel capitolo precedente è stato spiegato il processo di esportazione da FDS di isosuperfici e slices. In questo paragrafo, vedremo come importare su Unity3D i files precedentemente generati.

Per quanto riguarda le isosuperfici, precedentemente abbiamo generato files in formato .fbx; cliccando con il tasto destro all'interno del pannello project *Import New* → *Asset*, selezioniamo tutti i files che vogliamo importare.

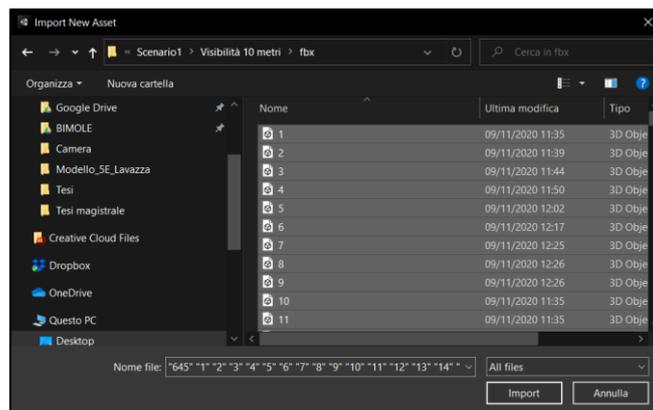


Figura 110 - Importazione delle isosuperfici su UNITY3D

Una volta aggiunte all'interno degli Assets, basterà trascinarle nel pannello Hierarchy per visualizzarle all'interno della scena.

È utile ricordare che ciascun file è una isosuperficie corrispondente ad un determinato istante di tempo con intervallo di 0,25 secondi. Il formato FBX non crea un'animazione, per cui è necessario creare una Coroutine, affinché le isosuperfici generino una sequenza.

Le Coroutine sono uno strumento molto utile nella programmazione in Unity, esse vengono invocate attraverso il comando `StartCoroutine(Coroutine())`. Lavorano come normali funzioni fino all'istruzione del comando `Yield Return X`, dove X corrisponde ai vari tipi di istruzioni.

Nel nostro caso, è necessario creare una Coroutine che permetta ad una determinata isosuperficie corrispondente ad un ben preciso istante di essere invocata per un certo intervallo di tempo e di invocare l'isosuperficie corrispondente all'istante di tempo successivo subito dopo, facendo sparire la precedente.

Per fare ciò, è necessario creare uno script.

Inoltre, per evitare di applicare lo script tante volte quante sono le isosuperfici importate, è stato creato un array ovvero una matrice i cui elementi sono proprio i files delle isosuperfici. Infatti, gli array consentono di memorizzare più oggetti in una singola variabile.

Ciò è stato possibile realizzarlo creando un GameObject vuoto, che fungerà da “padre” e inserire all'interno i files “figli” corrispondenti ai file importati, ed infine, associare al GameObject vuoto lo script chiamato *ArrayTest*, mostrato nel paragrafo A.3.

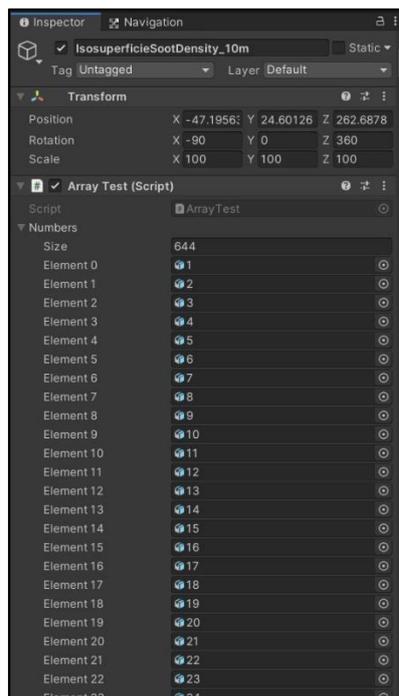


Figura 111 - Assegnazione dello script *ArrayTest*

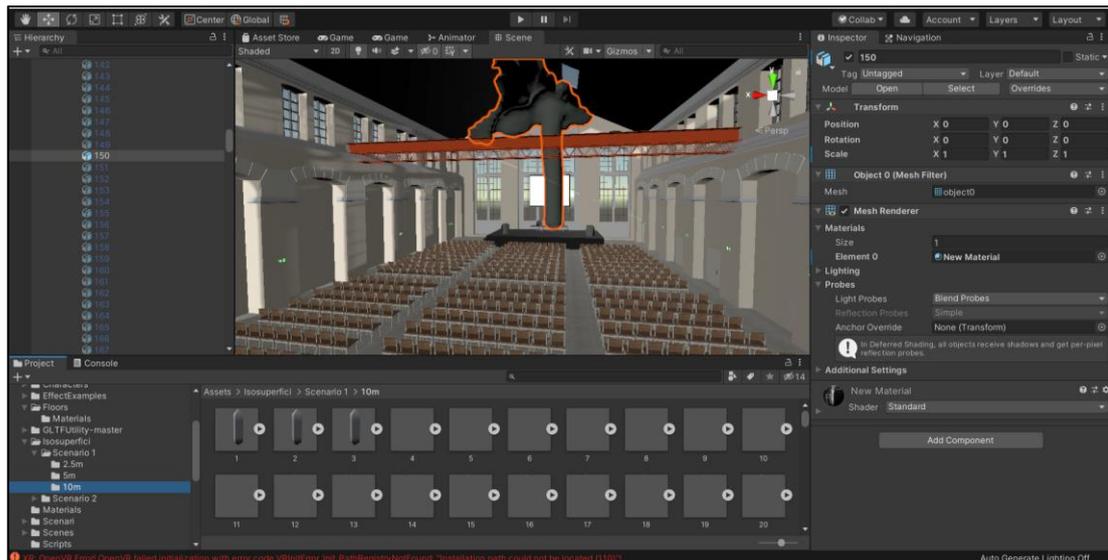


Figura 112 - Visualizzazione dell'isosuperficie corrispondente all'istante di tempo 150

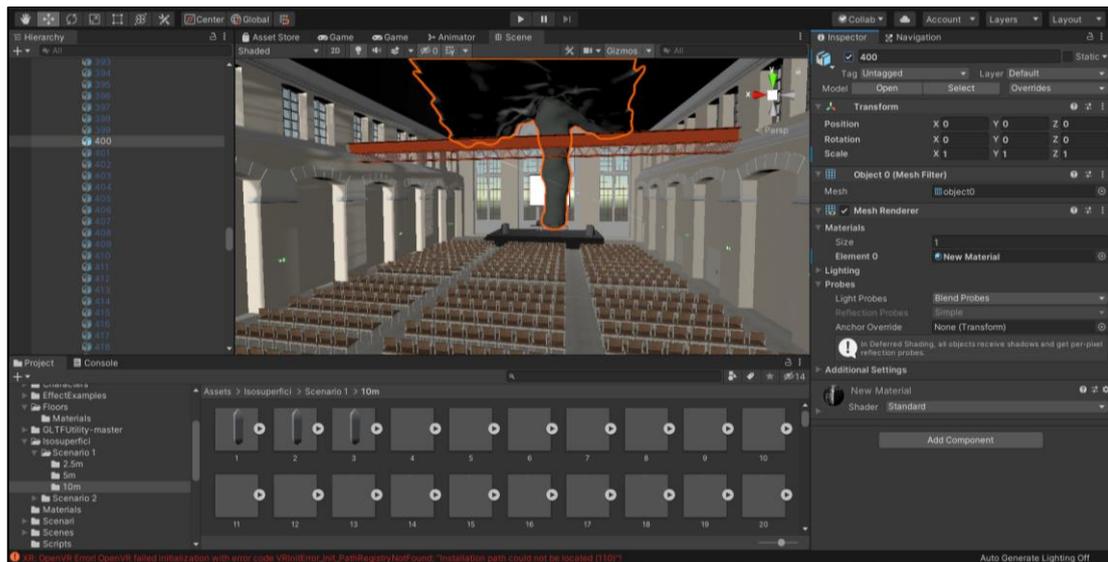


Figura 113 - Visualizzazione dell'isosuperficie corrispondente all'istante di tempo 400

Questo procedimento è stato effettuato per le isosuperfici corrispondenti a 10 metri, 5 metri ed infine 2,5 metri di visibilità.

Per quanto riguarda l'importazione delle slices, il procedimento di esportazione è il medesimo descritto per le isosuperfici.

Il passaggio in più che viene effettuato per le slices è la creazione di un nuovo shader per poter rendere visibili i colori al vertice, assegnati, come descritto precedentemente, mediante il software Blender.

Gli shader sono dei modelli matematici che indicano alla scheda grafica come rendere a schermo un oggetto 3D.

Descrivono, come l'oggetto riceve la luce o se la riceve, che colore ha, se proietta ombre, se ha una tralucenza, se crea rifrazioni, se è lucido, opaco o trasparente ecc.

Gli shader si dividono in diverse tipologie, uno dei più utilizzati è il Vertex Shader, utilizzato in questo caso, che appunto, opera a livello di vertici: leggono le caratteristiche della geometria e partendo dai vertici, si ricavano quello che c'è in mezzo.

Lo script utilizzato in questo lavoro di tesi è mostrato al paragrafo A.4.



Figura 114 - Visualizzazione della slice di visibilità all'istante 150



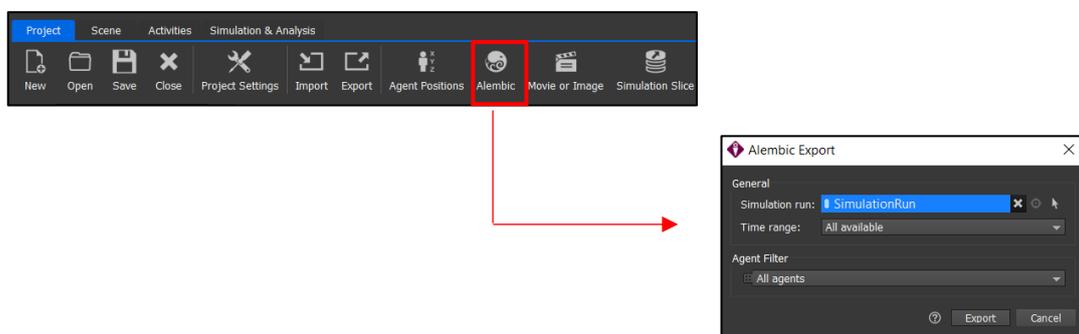
Figura 115 - Visualizzazione della slice di visibilità all'istante 400

## 8.3.3. Importazione della simulazione di esodo

Per poter visualizzare le simulazioni di esodo e i relativi occupanti su Unity3D è necessario esportare il file da MassMotion.

Una volta costruito il modello all'interno del software MassMotion (come visto nel paragrafo 7.5.), nel menù "Simulation & Analysis" si trovano i comandi fondamentali per avviare la simulazione e per esportare i dati.

A simulazione avvenuta, nel menù "Project" selezionando "Alembic" è possibile esportare l'animazione degli occupanti in formato .abc.



Possono essere esportati tutti gli occupanti della simulazione oppure si possono esportare alcuni di essi in funzione dell' "Evacuate" che li ha generati. Nel caso in esame, si è scelto di esportare tutti gli occupanti in un unico file.

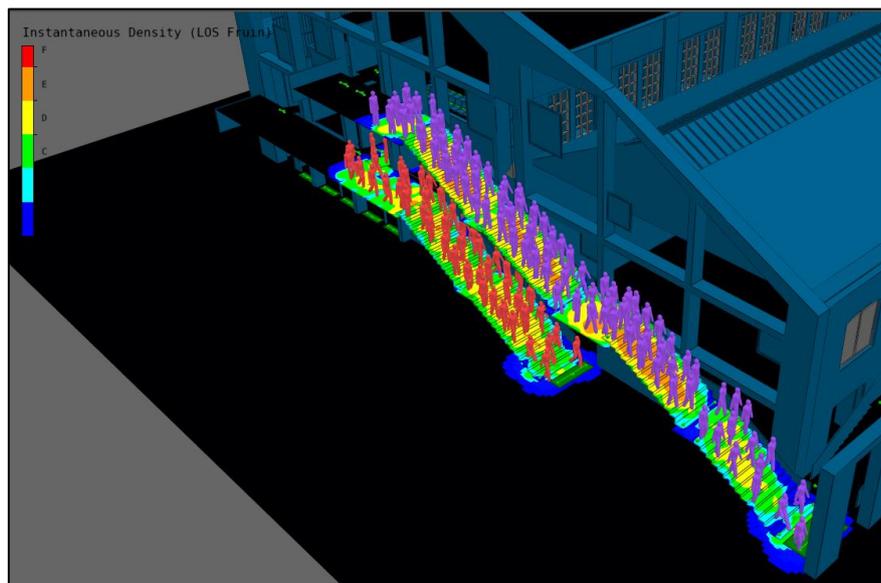


Figura 116 – Valori dei livelli di servizio

L'importazione in Unity del file in formato .abc riguardante l'animazione degli occupanti avviene mediante un plug-in di Unity chiamato **Alembic for Unity**. Una volta caricato il file Alembic all'interno del pannello project, il file deve essere trascinato all'interno del pannello.

Per poter associare l'animazione è necessario aprire l'editor Timeline, mediante i comandi *Windows* → *Sequencing* → *Timeline* e selezionare *Create* per creare una componente Director e una risorsa della sequenza temporale.

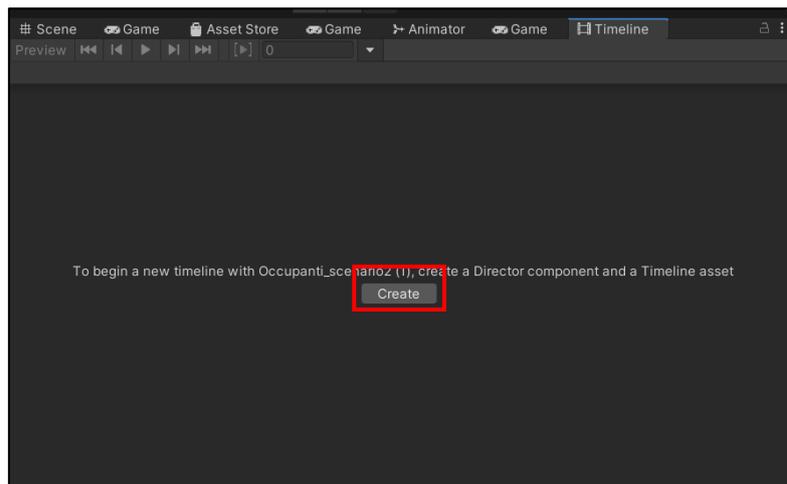


Figura 117 - Visualizzazione dell'Editor della Timeline

Ciò che Unity mostrerà sarà l'Editor della linea temporale con una traccia di animazione predefinita che elimineremo. Successivamente, trasciniamo il file Alembic all'interno dell'Editor della Timeline e selezioniamo *Add Clip With Alembic*.

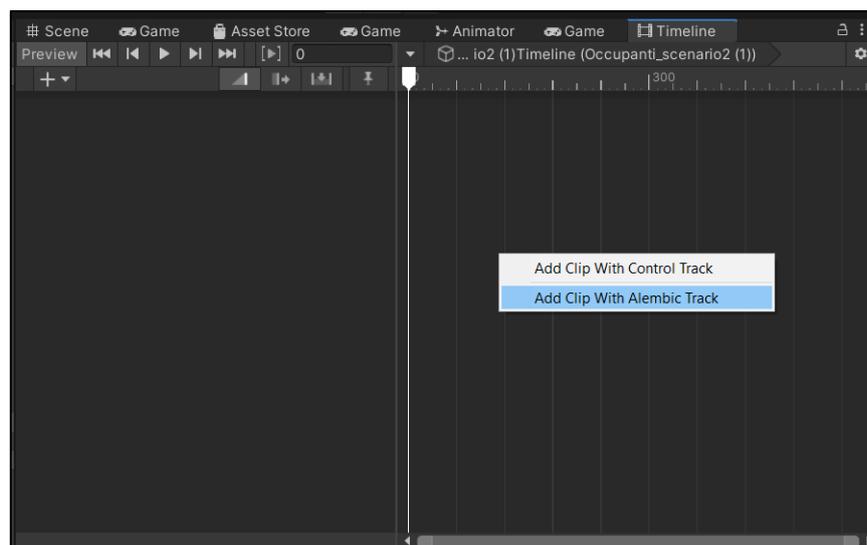


Figura 118 - Creazione della clip

In questo modo verrà associata l'animazione al file Alembic importato.

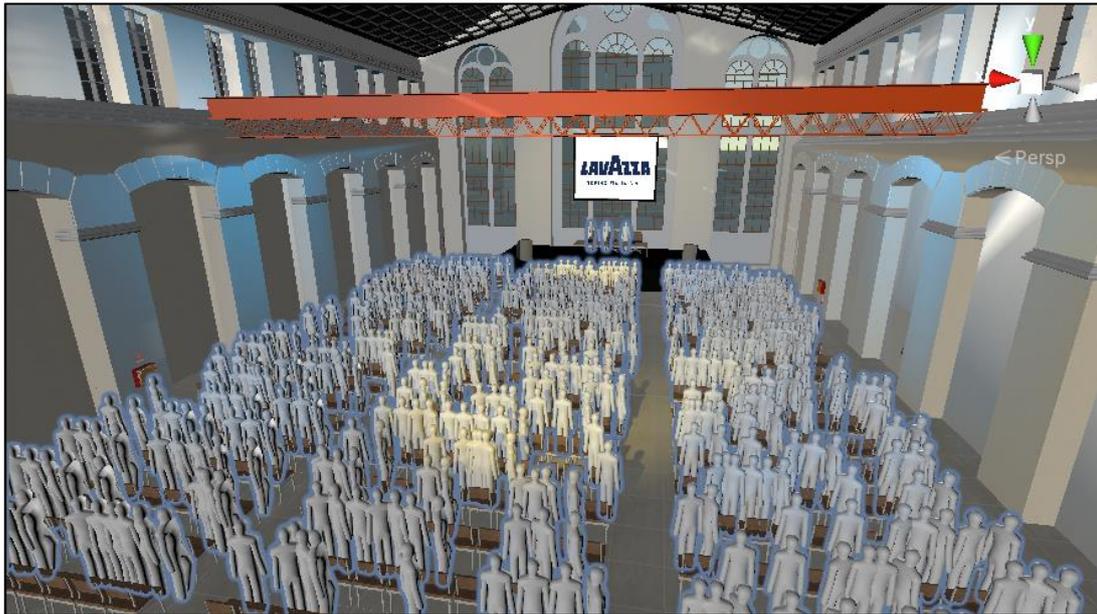
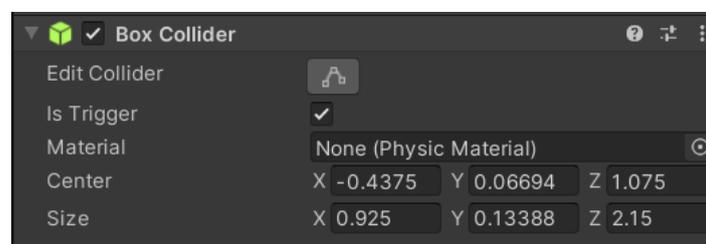


Figura 119 - Risultato finale dell'importazione del file Alembic

Successivamente è stata assegnata un'animazione alle porte in maniera tale da simulare l'apertura al passare degli occupanti. Per fare ciò è stato necessario assegnare a ciascuna anta della porta un *Box Collider* e selezionare *Is Trigger*.



Dopodiché è stata assegnata l'animazione di apertura della porta ed è stato assegnato lo script (paragrafo A.7) che permette alla porta di aprirsi nel momento in cui collide con gli occupanti.

Terminato questo step, abbiamo tutti gli elementi necessari per la costruzione dell'applicativo. L'obiettivo di questa fase è la creazione di un menù interattivo a schermo che permetta all'utente di scegliere cosa visualizzare all'interno della scena.

## 8.4. Inserimento della segnaletica

Il passo successivo è stato la modellazione della segnaletica.

È stato scelto di studiare tre tipologie di segnaletica:

1. **Tipologia 1:** cartellonistica standard retroilluminata conforme alla normativa, posta all'altezza di 2,20 metri dal pavimento associata ad un sistema di allarme antincendio;

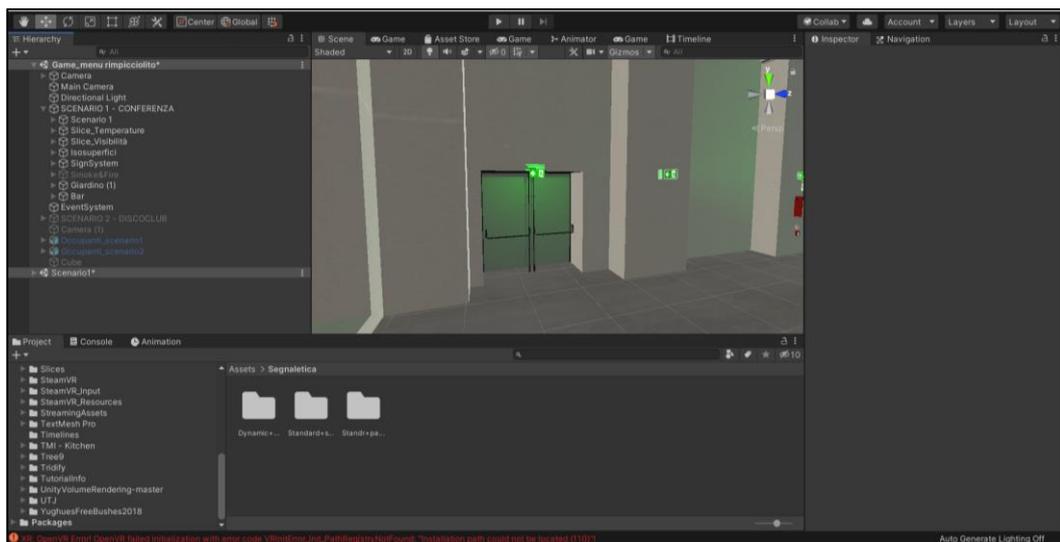


Figura 120 - Esempio della tipologia 1 di segnaletica su Unity3D

2. **Tipologia 2:** cartellonistica standard retroilluminata conforme alla normativa, posta all'altezza di 2,20 metri dal pavimento, associata un messaggio vocale e ad un sistema di guida a pavimento con rivestimento fotoluminescente;

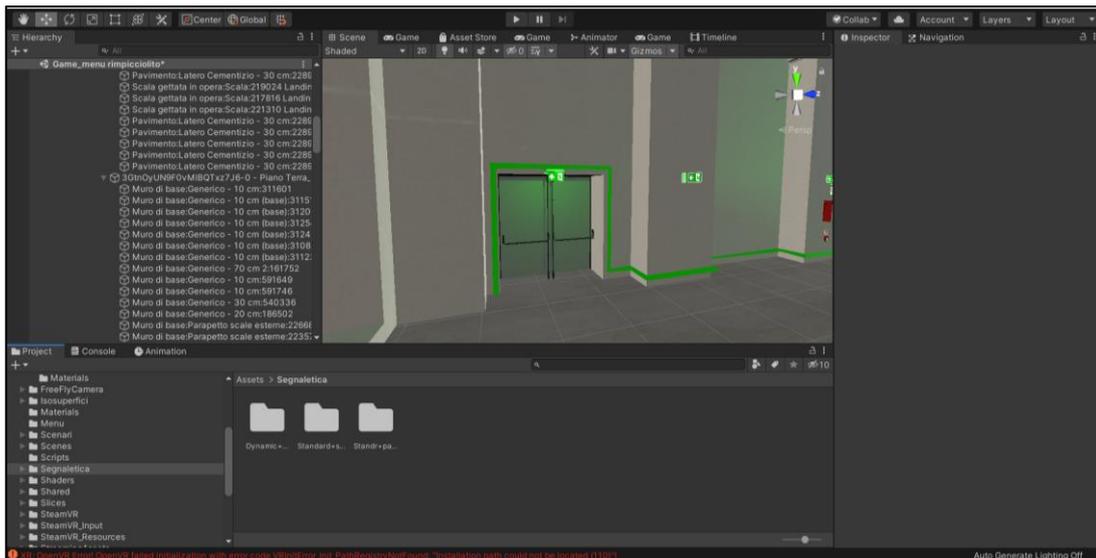


Figura 121 – Esempio della tipologia 2 di segnaletica su Unity

### 3. Tipologia 3: segnaletica dinamica associata un messaggio vocale.

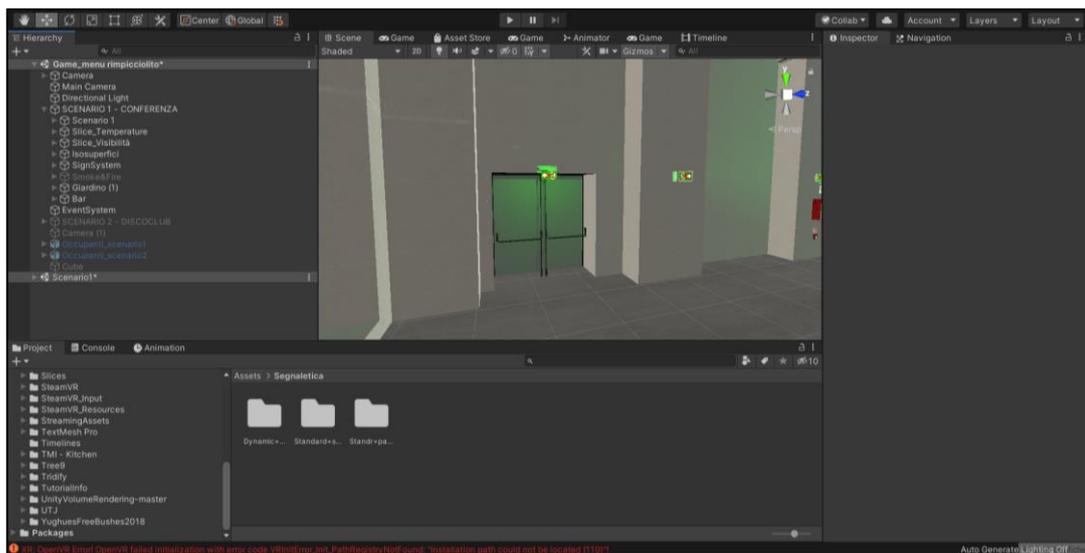


Figura 122 - Esempio della tipologia 3 su Unity

Per quanto riguarda l'inserimento degli effetti sonori, che come detto in precedenza, riproducono l'allarme antincendio e il messaggio vocale per avvisare gli occupanti dell'esodo:

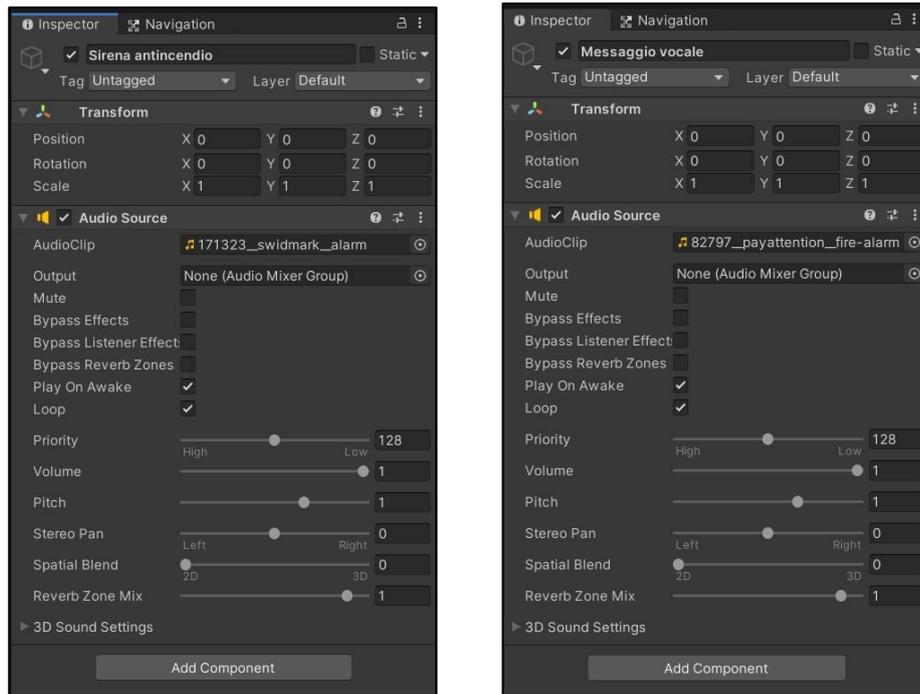


Figura 123 - Inserimento degli effetti sonori

Inoltre, si è pensato di creare una finestra in maniera tale da descrivere il sistema di segnaletica che si è scelto di visualizzare.

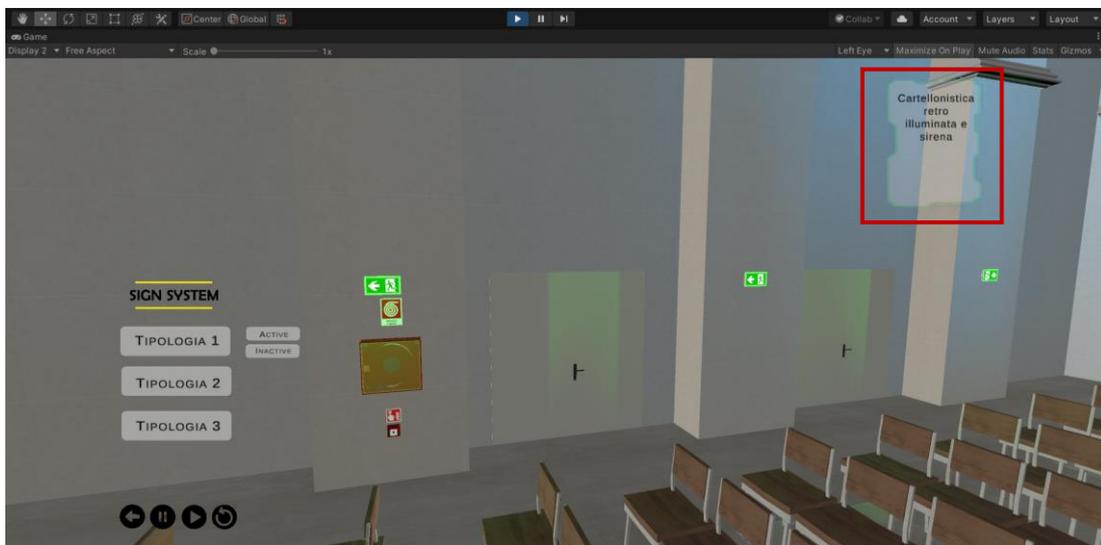


Figura 124 - Esempio di descrizione del sistema di segnaletica

## 8.5. Creazione dell'applicativo

### 8.5.1. Creazione del menù interattivo

Per la creazione del menù è necessario inserire un “Canvas”, un elemento base per le interfacce utente; ogni elemento della UI deve trovarsi all’interno di esso o non potrà essere renderizzato.

Il Canvas viene rappresentato sullo schermo mediante un rettangolo bianco.

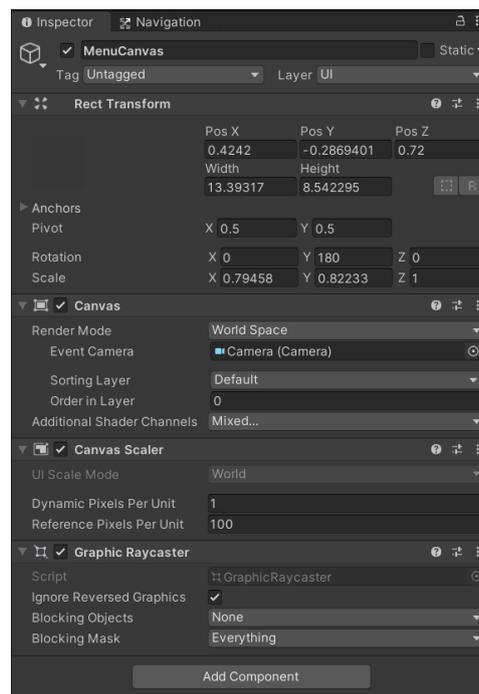


Figura 125 - Inspector di un Canvas

Il Rect Transform di un Canvas ha la stessa funzione di un Transform, visto precedentemente per i GameObject, infatti, permette di gestirne la posizione, la dimensione e la rotazione. È leggermente più complesso in quanto esistono un’infinità di schermi differenti, con dimensioni e risoluzioni diverse.

Nel menù a tendina Render Mode possiamo scegliere tre tipologie di visualizzazione:

- Screen Space - Overlay: questa tipologia viene visualizzata di default quando si crea un Canvas, permette a quest’ultima di cambiare automaticamente le dimensioni per riadattarsi quando lo schermo viene ridimensionato o cambia la risoluzione.

Usando tipologia di sistema di visualizzazione non è necessario spostare o ruotare l'oggetto in un punto specifico della scena perché esso sarà posizionato sempre al centro dello schermo;

- Screen Space – Camera: questo sistema di visualizzazione necessita di una telecamera, infatti, si comporta come se la Canvas fosse un oggetto 3D posto ad una certa distanza dalla telecamera specificata;
- World Space: con questo sistema di visualizzazione la Canvas si comporta come un qualsiasi altro oggetto 3D nella scena. Questa modalità viene utilizzata nel momento in cui le interfacce utente devono essere parte integrante del mondo di gioco.

Per la creazione dell'applicativo obiettivo di questo lavoro di tesi, è stato ritenuto opportuno scegliere la modalità *World Space*.

Successivamente, si è passati alla creazione del menu principale, quindi alla creazione di due bottoni che permettono di scegliere quale scenario visualizzare: sala conferenze o discoteca.

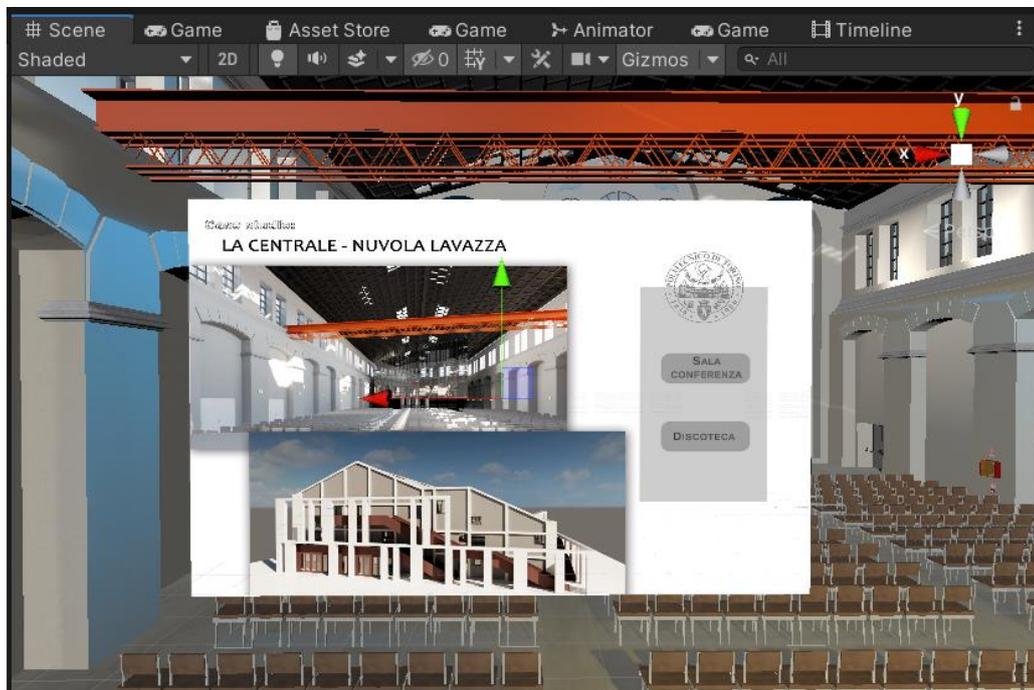


Figura 126 - Creazione del Menù iniziale

I pulsanti vengono creati mediante il comando *GameObject* → *UI* → *Button*, in questo modo viene creato un pulsante che permette di attivare un evento.

Affinché il pulsante compia questo lavoro, è necessario collegare la funzione al pulsante. Quindi, clicchiamo sul pulsante e nel suo Inspector troviamo la voce *OnClick ()*, clicchiamo sul pulsante + e trasciniamo l'oggetto a cui vogliamo collegare la funzione del pulsante. Lo stop finale è impostare la funzione *GameObject.SetActive* affinché al click si attivi l'oggetto selezionato.

Inoltre, il pulsante può essere personalizzato a nostro piacimento. Infatti, mediante alcuni comandi è possibile assegnare un colore per ogni azione svolta da noi sul pulsante stesso:

- Normal Color: è il colore che il pulsante assume quando non viene selezionato
- Highlighted Color: è il colore che il pulsante assume quando ci passiamo sopra con il mouse;
- Pressed Color: è il colore che assume quando lo premiamo;
- Selected Color: è il colore che assume quando è selezionato.

Al pulsante è possibile anche assegnare un'immagine.

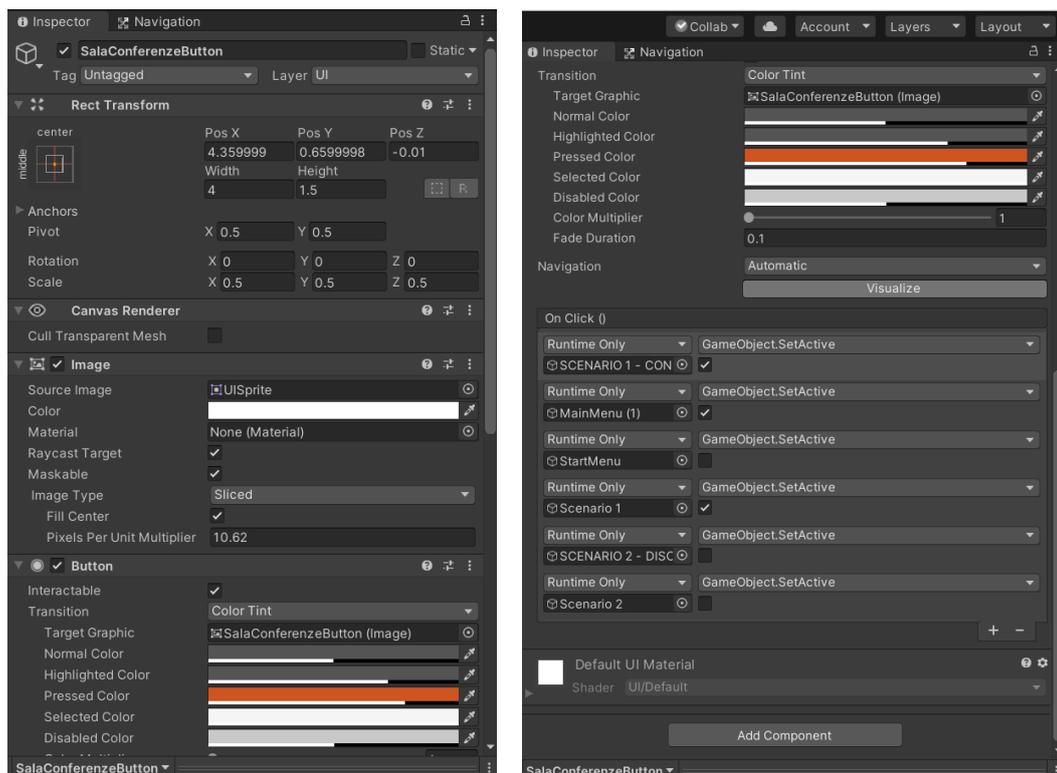


Figura 127 - Esempio dell'Inspector del pulsante

Successivamente, si è passati alla creazione dei menù che permettono la visualizzazione degli elementi importati precedentemente: isosuperfici, slices, occupanti e segnaletica.

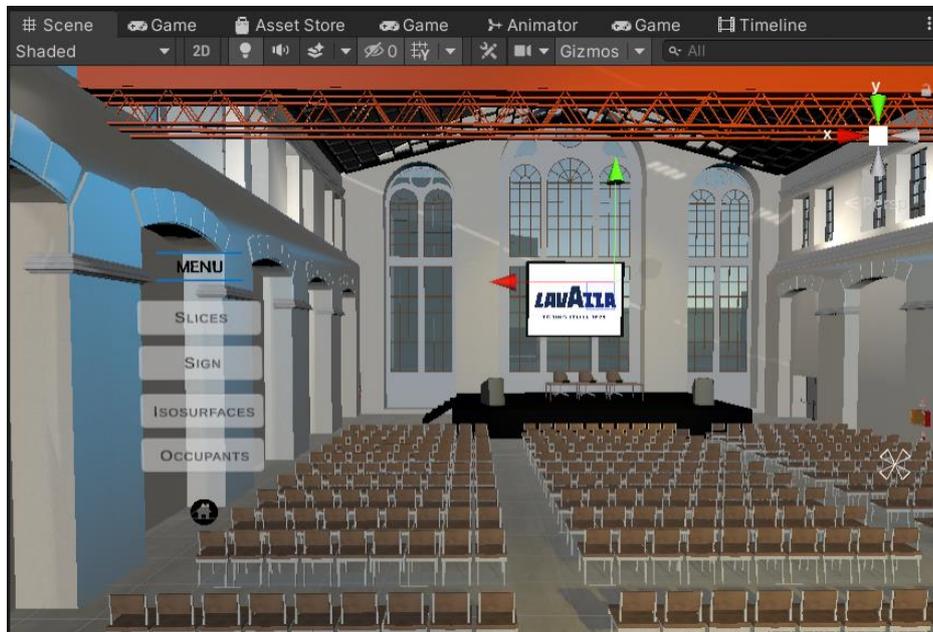
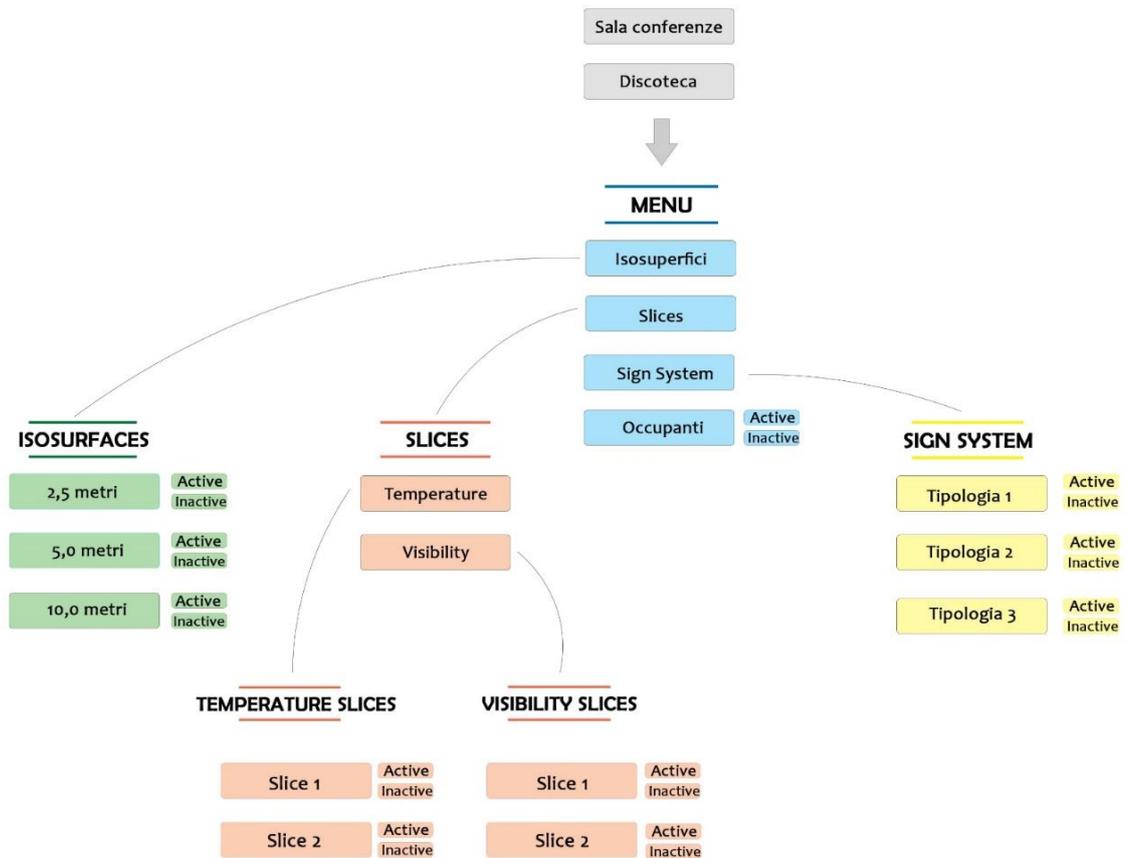


Figura 128 - Creazione del menù principale

Oltre alla creazione dei pulsanti che permettono la visualizzazione degli elementi, sono stati creati dei pulsanti che permettono una migliore flessibilità all'interno del menù.

- Il pulsante Home permette di ritornare al menù iniziale; 
- Il pulsante Back permette di ritornare al menù precedente; 
- Il pulsante Pause permette di mettere in pausa l'animazione; 
- Il pulsante Play permette di far ripartire l'animazione messa in pausa; 
- Il pulsante Reset permette di far ricominciare dall'inizio l'animazione. 

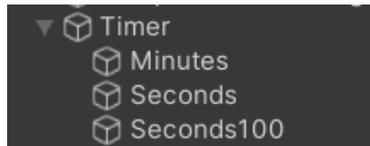
Di seguito viene mostrato il menù completo realizzato mediante il software:



Nello schema sopra riportato viene mostrato il flusso attraverso il quale l'utente finale può scegliere cosa visualizzare: la schermata iniziale è costituita da un menu a tendina che permette di scegliere quale scenario visualizzare. Una volta scelto lo scenario, il menu principale mostra gli elementi che possono essere visualizzati per ogni elemento si aprirà un ulteriore menù permette all'utilizzatore di scegliere esattamente cosa visualizzare. I pulsanti "Active" e "Inactive" permettono di visualizzare più elementi contemporaneamente.

Inoltre, è stato inserito un timer con lo scopo di far capire all'utilizzatore dell'applicativo il tempo della simulazione.

Per fare ciò, è stata creata un'altra Canvas nel quale è stato inserito un GameObject vuoto chiamato "Timer", all'interno di quest'ultimo sono stati creati tre elementi testo mediante il comando *GameObject* → *UI* → *Text*, questi elementi rappresentano i minuti, i secondi e i decimi di secondi.



All'elemento vuoto chiamato "Timer" è stato associato lo script Timer riportato nel paragrafo A.5; una volta associato lo script è necessario collegare gli elementi testo precedentemente creati:

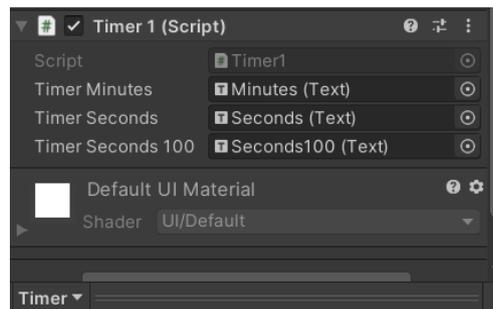


Figura 129 - Visualizzazione dello script Timer

Lo script permette di mettere in pausa, resettare e far partire nuovamente il timer.

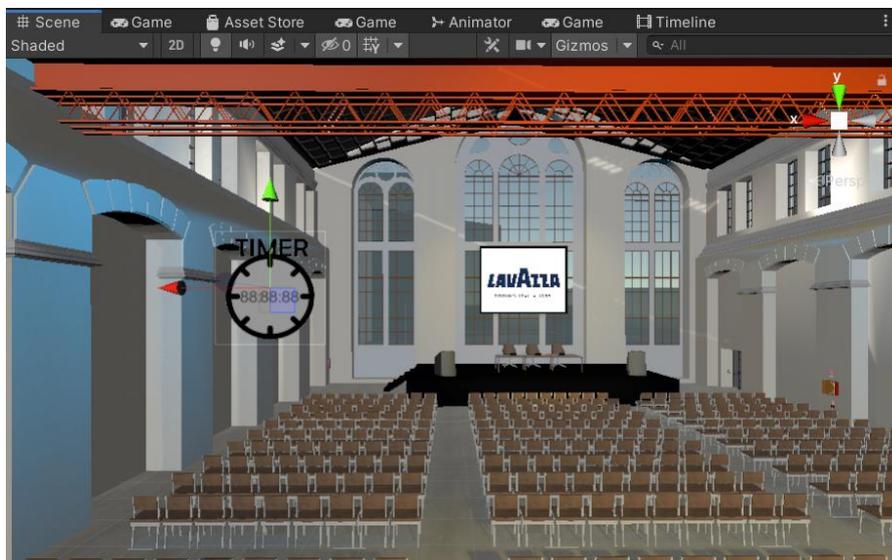


Figura 130 - Visualizzazione del Timer realizzato

Inoltre, all'interno della Canvas creata per il timer, sono state inserite le legende per le slices di visibilità e temperatura per permettere una maggiore comprensibilità dei valori dei colori delle stesse.

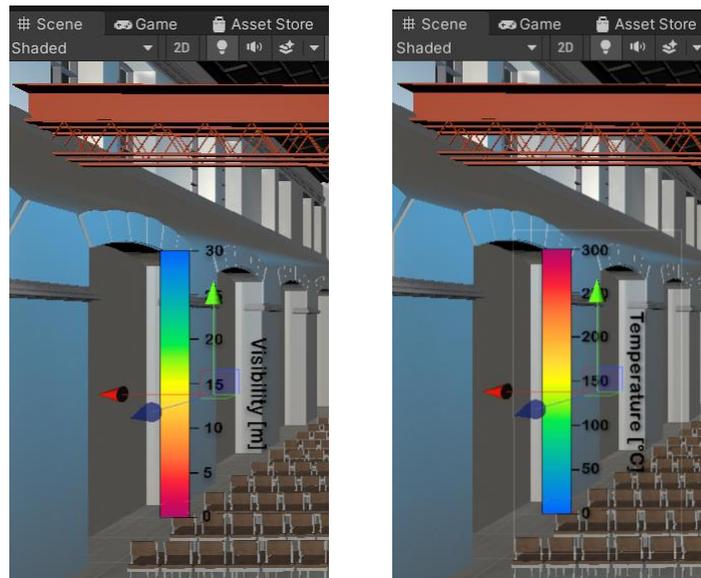
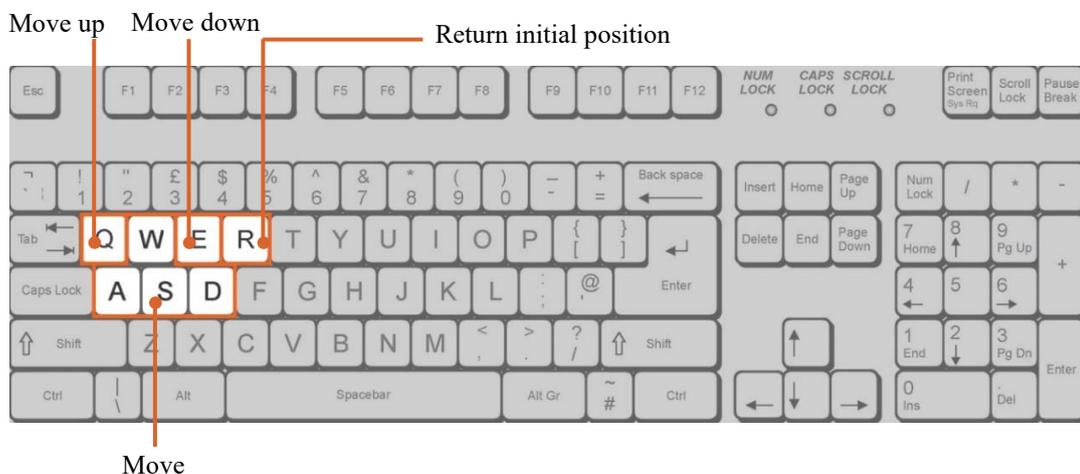


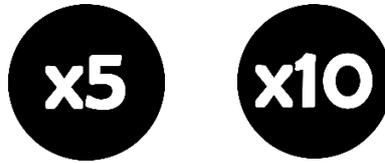
Figura 131 - Inserimento delle legende per la lettura delle slices

### 8.5.2. Movimento

L'ultimo step per la creazione dell'applicativo è stato l'inserimento di una Camera, chiamata "Fly Camera" per permettere all'utilizzatore di muoversi liberamente all'interno dell'edificio. Innanzitutto, le Canvas che presentano i menu, il timer e le legende sono state inserite all'interno di una Camera e successivamente è stato associato a quest'ultima lo script del paragrafo A.6. Lo script inserito permette di muoversi mediante l'utilizzo della tastiera, infatti, con i pulsanti W, A, S e D ci si muove rispettivamente in avanti, a sinistra, indietro e a destra, con i pulsanti Q e E ci si può spostare in su e in giù. L'utilizzo del mouse, invece, permette la rotazione a 360°.



Inoltre, sono stati creati due pulsanti che permettono di accelerare l'animazione di isosuperfici, slices ed occupanti.



Modificando lo script utilizzato per la creazione delle Coroutine (paragrafo A.3) è stato possibile dare una velocità maggiore agli istanti di tempo che si susseguono, questo è stato fatto andando a modificare la riga 32:

- Per una velocità **x5**: `yield return new WaitForSeconds(0.05f);`
- Per una velocità **x10**: `yield return new WaitForSeconds(0.025f);`

Qui di seguito si riportano immagini significative dell'applicativo:

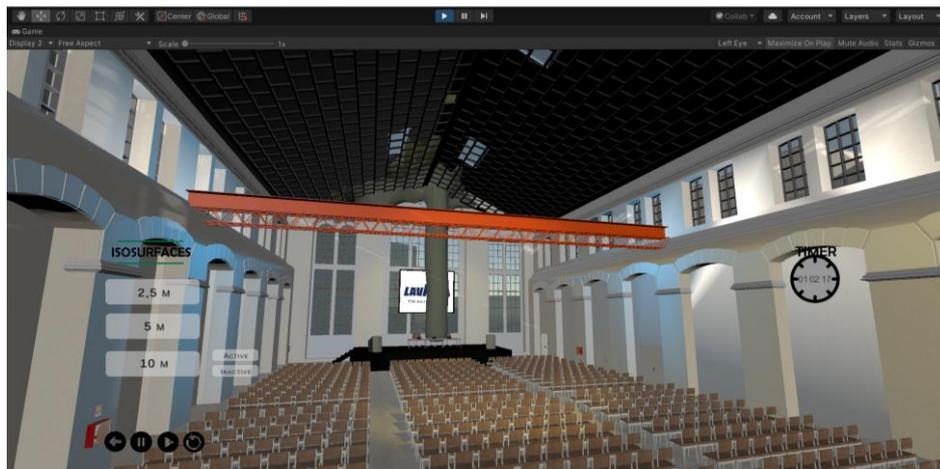


Figura 132 - Visualizzazione dell'isosuperficie dello scenario 1



Figura 133 - Visualizzazione dell'isosuperficie dello scenario 2

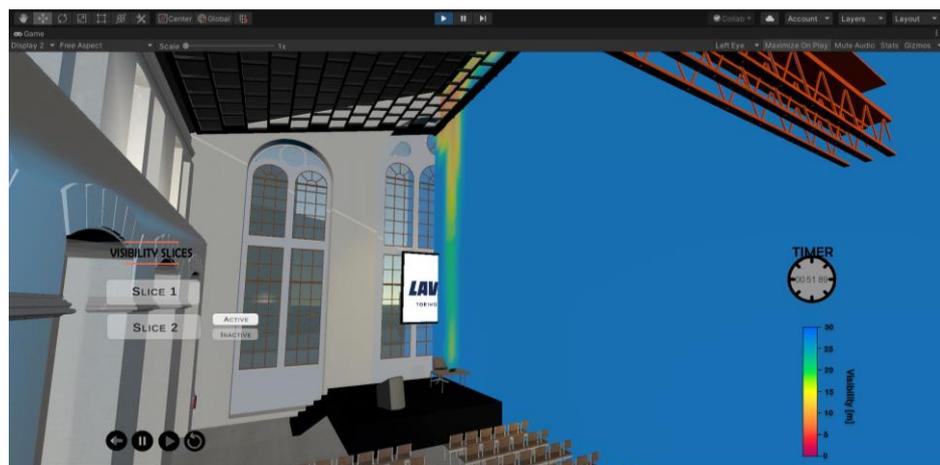


Figura 134 - Visualizzazione della slice di visibilità dello scenario 1



Figura 135 - Visualizzazione della slice di visibilità dello scenario 2

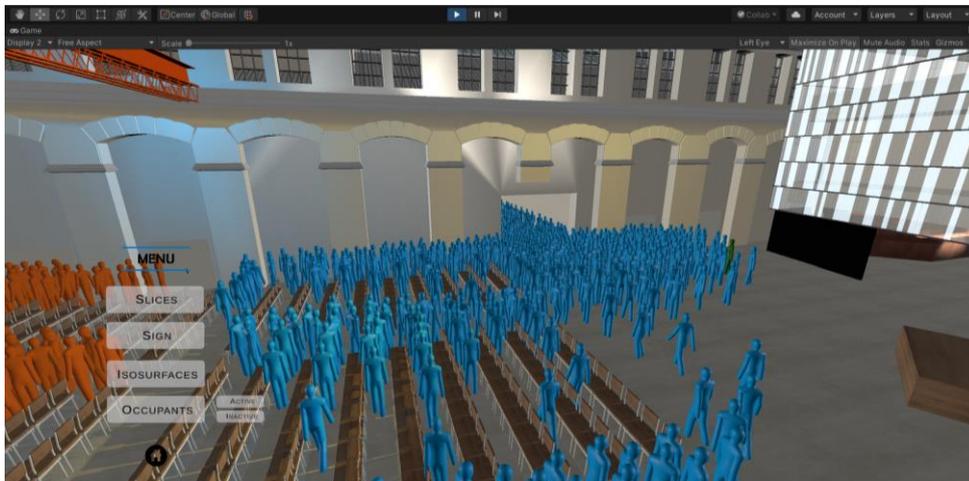


Figura 136 - Visualizzazione degli occupanti nello scenario 1

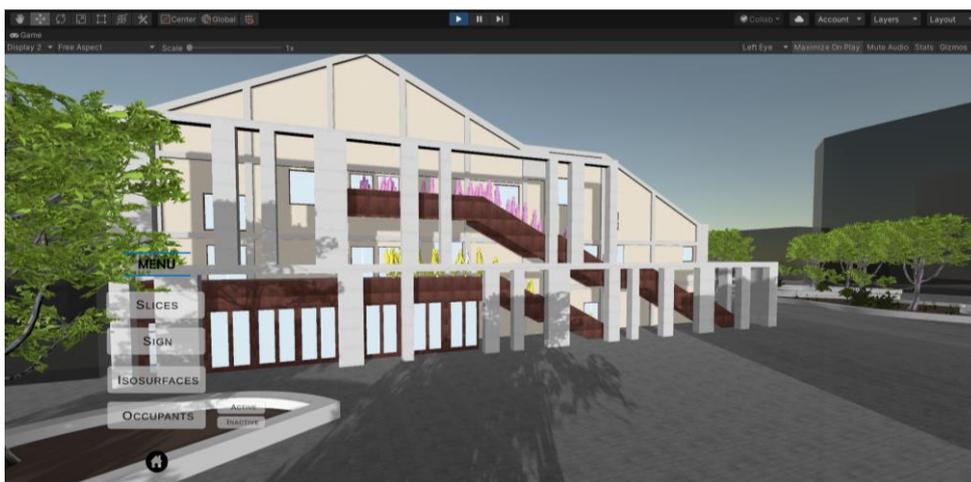


Figura 137 - Visualizzazione degli occupanti dello scenario 2

Di seguito si riportano i QR code relativi ad alcune scene significative dell'applicazione realizzata.



Visualizzazione dell'isosuperficie dello scenario 1 sull'applicativo realizzato – Sala conferenze



Visualizzazione dell'isosuperficie e della segnaletica dello scenario 2 sull'applicativo realizzato – Sala da ballo



Visualizzazione della simulazione di esodo dello scenario 2 sull'applicativo realizzato – Discoteca



Visualizzazione delle slices di visibilità dello scenario 2 sull'applicativo realizzato –  
Discoteca



Avvio dell'applicativo

## Capitolo 9

### Conclusioni e sviluppi futuri

Il lavoro di tesi si è proposto l'obiettivo di portare il mondo della Fire Safety Engineering all'interno della realtà virtuale, cercando di visualizzare gli elementi inerenti la simulazione di esodo e di incendio. Come detto precedentemente, non è stato possibile effettuare uno studio sulla visibilità della segnaletica e del comportamento umano in realtà virtuale, in quanto, l'esportazione del fumo è ancora troppo complessa. Infatti, bisognerebbe realizzare un Volume Rendering del fumo stesso.

La realizzazione di questo lavoro è stata permessa grazie all'utilizzo della metodologia BIM e dell'interoperabilità dei software. L'unico svantaggio che si è riscontrato è l'esportazione dei materiali, infatti, alcuni di essi esportati con Autodesk Revit non vengono riconosciuti ed è stato necessario crearli su Unity e riassegnarli manualmente. Si è visto come la segnaletica è una componente fondamentale del wayfinding dell'edificio e della progettazione della sicurezza. Le informazioni trasmesse dai segnali dovrebbero identificare chiaramente la direzione e l'ubicazione delle vie di fuga da qualsiasi punto dell'edificio verso un luogo sicuro o un'uscita finale. Spesso si presume che se il sistema di segnaletica è conforme, sarà efficace nel trasmettere le informazioni agli occupanti, che a loro volta interpreteranno e utilizzeranno correttamente tali indicazioni.

In realtà, abbiamo visto come l'interazione tra gli occupanti e la segnaletica in una situazione di emergenza sia un processo complesso influenzato da una serie di fattori fisici, psicologici e cognitivi. Nonostante l'importanza dei sistemi di segnaletica, nessun software di simulazione di esodo permette la loro inclusione in quanto non ci sono approcci che consentono all'utente di assegnare l'efficacia del sistema di segnaletica. Per questo motivo, nasce l'esigenza di valutare un sistema di segnaletica in prima persona, grazie alla realtà virtuale o alla realtà aumentata.

Nonostante al giorno d'oggi, gli strumenti della Fire Safety Engineering e della realtà virtuale siano molto avanzati, ci sono ancora molte difficoltà nell'esportazione dei files e nella visualizzazione del fumo. Infatti, questo lavoro di tesi affronta numerosi steps

per arrivare all'obiettivo finale, ovvero, la visualizzazione delle isosuperfici e delle slices. Inoltre, sono pochi i software di simulazione di esodo che permettono l'esportazione dell'animazione degli occupanti, infatti, come abbiamo visto, è stato utilizzato il software MassMotion, che è uno dei pochi programmi che permette di effettuare questo passaggio. Inoltre, un aspetto da non sottovalutare è la limitazione legata alla predisposizione degli utenti all'utilizzo della tecnologia avanzata. Secondo alcune ricerche e esperimenti effettuati su questa tematica, l'utente nel momento in cui viene immerso nel mondo virtuale è soggetto ad una condizione di stress psicologica non trascurabile, questo non è dovuto a caratteristiche come l'età o il genere dell'individuo, ma al fatto che al giorno d'oggi, nonostante il mondo virtuale si stia sempre di più sviluppando, non è ancora molto conosciuto e utilizzato.

Utilizzando la teoria dell'**Affordance** è possibile valutare il miglior utilizzo di questo applicativo: a schermo oppure mediante l'utilizzo di un visore.

La **teoria dell'Affordance**, sviluppata da Gibson nel 1977, è quella teoria secondo cui un oggetto è percepito in relazione a ciò che offre all'individuo. Un'Affordance è, quindi, ciò che l'oggetto offre all'individuo in relazione al suo obiettivo. Questa teoria è stata spesso utilizzata nell'ambito dell'Ingegneria della Sicurezza Antincendio, soprattutto per la valutazione dei sistemi di segnaletica [26], per la progettazione delle uscite di emergenza e per la valutazione del comportamento umano in emergenza.

Al fine di consentire un'analisi migliore, è utile suddividere le Affordance in diverse categorie: una possibile divisione è stata proposta da Hartson che suggerisce di dividere le offerte nelle seguenti categorie:

- Affidabilità sensoriale: percepire o vedere
- Affidabilità cognitiva: comprensione
- Affidabilità fisica: fare o usare fisicamente
- Affidabilità funzionale: realizzazione dell'obiettivo dell'individuo

È stato affermato che la Teoria delle Affordance può essere uno strumento utile per identificare potenziali difetti di progettazione dei sistemi di evacuazione nelle prime fasi del processo di progettazione. Esplorando sistematicamente le offerte sensoriali, cognitive, fisiche e funzionali fornite da un sistema di evacuazione, dovrebbe essere possibile identificare i conflitti e la progettazione non ottimale. Quindi, la teoria può essere utilizzata per analizzare una serie di possibili progetti di sistema al fine di

escludere il sistema meno appropriato. Tuttavia, questo tipo di analisi richiede un'ampia comprensione dei diversi tipi di offerte in relazione al sistema in esame.

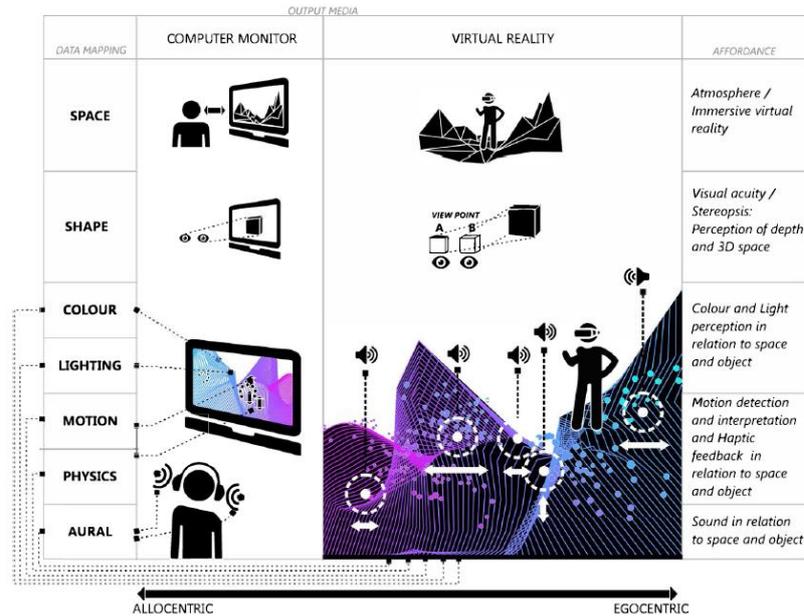


Figura 138 - Differenze tra Monitor e Virtual Reality

Quindi, utilizzando la teoria dell'Affordance è possibile inquadrare tutte le differenze tra realtà virtuale immersiva e visuale sul monitor di un computer [27]. Sarebbe utile in futuro, effettuare dei questionari utilizzando questa teoria, in maniera tale che dei partecipanti possano esprimere la loro valutazione in merito all'utilizzo della Virtual Reality e ai suoi possibili vantaggi rispetto all'utilizzo comune del monitor. Un esperimento simile è stato condotto presso l'università di Lund [26] in cui 96 partecipanti hanno fornito raccomandazioni sulla progettazione di luci lampeggianti nei pressi delle uscite di sicurezza nei tunnel stradali. In questo esperimento sono state studiate una serie di variabili:

- Colore delle luci
- Velocità di lampeggiamento
- Tipologia di sorgente luminosa
- Numero e la disposizione delle luci

Per cui, è stato chiesto ai partecipanti di classificare diversi tipi di progetti utilizzando un questionario basato sulla teoria dell'Affordance. Utilizzando una metodologia simile, sarebbe dunque, possibile effettuare lo stesso esperimento per comparare l'utilizzo dell'applicativo in realtà virtuale immersiva e non immersiva.

## Ringraziamenti

Giunta al termine di questo lavoro vorrei ringraziare tutte le persone che mi sono state vicine e che hanno permesso la realizzazione di questa tesi.

I miei più sentiti ringraziamenti vanno al Professore Roberto Vancetti, per avermi sostenuta e supportata durante il periodo di tesi e tirocinio. Inoltre, ringrazio l'Ingegnere Emiliano Cereda per avermi seguito durante lo svolgimento del lavoro.

Infine, ringrazio lo studio Ai Engineering, ma soprattutto l'Ingegnere Filippo Così per avermi fornito il materiale necessario alla stesura di questa tesi.

Vorrei ringraziare anche il Dottor Ruggiero Lovreglio, ricercatore presso la Massey University per il tempo prezioso dedicatomi.

# Appendice

## A.1. Esportazione isosuperfici

```

1. # trace generated using paraview version 5.9.0-RC3
2.
3. ##### import the simple module from the paraview
4. from paraview.simple import *
5. import numpy
6.
7. ##### disable automatic camera reset on 'Show'
8. paraview.simple._DisableFirstRenderCameraReset()
9.
10. # find source
11. prova31ottobre2_0001xyz = FindSource('Prova31ottobre2_0001.xyz')
12.
13. # create a new 'Contour'
14. contour1 = Contour(registrationName='Contour1', Input=prova31ottobre2_0001x
    yz)
15. contour1.ContourBy = ['POINTS', 'Density']
16. contour1.Isosurfaces = [0.5976265668869019]
17. contour1.PointMergeMethod = 'Uniform Binning'
18.
19. # find source
20. prova31ottobre2_0002xyz = FindSource('Prova31ottobre2_0002.xyz')
21.
22. # find source
23. prova31ottobre2_0003xyz = FindSource('Prova31ottobre2_0003.xyz')
24.
25. # Properties modified on contour1
26. contour1.Isosurfaces = [3.5e-05]
27.
28. # get active view
29. renderView1 = GetActiveViewOrCreate('RenderView')
30.
31. # show data in view
32. contour1Display = Show(contour1, renderView1, 'GeometryRepresentation')
33.
34. # get color transfer function/color map for 'Density'
35. densityLUT = GetColorTransferFunction('Density')
36.
37. # trace defaults for the display properties.
38. contour1Display.Representation = 'Surface'
39. contour1Display.ColorArrayName = ['POINTS', 'Density']
40. contour1Display.LookupTable = densityLUT
41. contour1Display.SelectTCoordArray = 'None'
42. contour1Display.SelectNormalArray = 'Normals'
43. contour1Display.SelectTangentArray = 'None'
44. contour1Display.OSPRayScaleArray = 'Density'
45. contour1Display.OSPRayScaleFunction = 'PiecewiseFunction'
46. contour1Display.SelectOrientationVectors = 'Momentum'
47. contour1Display.ScaleFactor = 0.3499969482421875
48. contour1Display.SelectScaleArray = 'Density'
49. contour1Display.GlyphType = 'Arrow'
50. contour1Display.GlyphTableIndexArray = 'Density'
51. contour1Display.GaussianRadius = 0.017499847412109377
52. contour1Display.SetScaleArray = ['POINTS', 'Density']
53. contour1Display.ScaleTransferFunction = 'PiecewiseFunction'
54. contour1Display.OpacityArray = ['POINTS', 'Density']
55. contour1Display.OpacityTransferFunction = 'PiecewiseFunction'
56. contour1Display.DataAxesGrid = 'GridAxesRepresentation'

```

```
57. contour1Display.PolarAxes = 'PolarAxesRepresentation'
58.
59. # init the 'PiecewiseFunction' selected for 'ScaleTransferFunction'
60. contour1Display.ScaleTransferFunction.Points = [3.5000000934815034e-
05, 0.0, 0.5, 0.0, 3.500745151541196e-05, 1.0, 0.5, 0.0]
61.
62. # init the 'PiecewiseFunction' selected for 'OpacityTransferFunction'
63. contour1Display.OpacityTransferFunction.Points = [3.5000000934815034e-
05, 0.0, 0.5, 0.0, 3.500745151541196e-05, 1.0, 0.5, 0.0]
64.
65. # show color bar/color legend
66. contour1Display.SetScalarBarVisibility(renderView1, True)
67.
68. # update the view to ensure updated data information
69. renderView1.Update()
70.
71. # get color transfer function/color map for 'StagnationEnergy'
72. stagnationEnergyLUT = GetColorTransferFunction('StagnationEnergy')
73.
74. # Rescale transfer function
75. stagnationEnergyLUT.RescaleTransferFunction(0.0, 30.0)
76.
77. # get opacity transfer function/opacity map for 'StagnationEnergy'
78. stagnationEnergyPWF = GetOpacityTransferFunction('StagnationEnergy')
79.
80. # Rescale transfer function
81. stagnationEnergyPWF.RescaleTransferFunction(0.0, 30.0)
82.
83. # get opacity transfer function/opacity map for 'Density'
84. densityPWF = GetOpacityTransferFunction('Density')
85.
86. # set scalar coloring
87. ColorBy(contour1Display, ('POINTS', 'StagnationEnergy'))
88.
89. # Hide the scalar bar for this color map if no visible data is colored by i
t.
90. HideScalarBarIfNotNeeded(densityLUT, renderView1)
91.
92. # rescale color and/or opacity maps used to include current data range
93. contour1Display.RescaleTransferFunctionToDataRange(True, False)
94.
95. # show color bar/color legend
96. contour1Display.SetScalarBarVisibility(renderView1, True)
97.
98. # hide data in view
99. Hide(prova31ottobre2_0001xyz, renderView1)
100.
101.     # get animation scene
102.     animationScene1 = GetAnimationScene()
103.
104.     # Properties modified on animationScene1
105.
106.     #arange è una funzione che ad ogni iterazione mi crea un numero deci
male (salvato nella variabile x) che parte dal primo parametro (quindi prim
a iterazione x = 1.0).
107.     #il secondo parametro (651) indica il limite massimo (escluso) delle
iterazioni del ciclo for, quindi il loop si ripete fino a quando x = 650.0
108.     #il terzo parametro indica di quanto la variabile x incrementa ad og
ni iterazione (x = 1.0 poi 2.0 e così via)
109.     for x in numpy.arange(1.0, 5.0, 1.0):
110.         #ora animationTime prende il valore di x ad ogni iterazione quindi
si automodifica per ogni istante di tempo
111.         animationScene1.AnimationTime = x
```

```

112.
113.     # get the time-keeper
114.     timeKeeper1 = GetTimeKeeper()
115.
116.     # export view
117.     #x è un numero decimale, dobbiamo salvarlo nel file come numero in
   tero quindi ho fatto la conversione (ESEMPIO: x = 2.0 -> timeInstant = 2)
118.     timeInstant = int(x)
119.     #ho parametrizzato la stringa in maniera tale che ad ogni iterazio
    ne cambi in base all'istante di tempo
120.     ExportView(f'C:/Users/Asus/Desktop/fds2paraview/vrml_8novembre/{ti
    meInstant}.vrml', view=renderView1)
121.
122.
123.     #=====
124.     # addendum: following script captures some of the application
125.     # state to faithfully reproduce the visualization during playback
126.     #=====
127.
128.     # get layout
129.     layout1 = GetLayout()
130.
131.     #-----
132.     # saving layout sizes for layouts
133.
134.     # layout/tab size in pixels
135.     layout1.SetSize(1483, 505)
136.
137.     #-----
138.     # saving camera placements for views
139.
140.     # current camera placement for renderView1
141.     renderView1.CameraPosition = [51.21461015291642, -
    54.462948482340266, 28.213366977303743]
142.     renderView1.CameraFocalPoint = [21.3120578457211, -
    8.369003480247585, 10.334386741260063]
143.     renderView1.CameraViewUp = [-
    0.187904734433346, 0.2468447363382908, 0.9506626567398204]
144.     renderView1.CameraParallelScale = 1.837092600537982
145.
146.     #-----
147.     # uncomment the following to render all views
148.     # RenderAllViews()
149.     # alternatively, if you want to write images, you can use SaveScreen
    shot(...).

```

## A.2. Esportazione Slices

```
1. # trace generated using paraview version 5.9.0-RC3
2.
3. ##### import the simple module from the paraview
4. from paraview.simple import *
5. import numpy
6.
7. ##### disable automatic camera reset on 'Show'
8. paraview.simple._DisableFirstRenderCameraReset()
9.
10. # find source
11. prova31ottobre2_0001xyz = FindSource('Prova31ottobre2_0001.xyz')
12.
13. # create a new 'Slice'
14. slice1 = Slice(registrationName='Slice1', Input=prova31ottobre2_0001xyz)
15. slice1.SliceType = 'Plane'
16. slice1.HyperTreeGridSlicer = 'Plane'
17. slice1.SliceOffsetValues = [0.0]
18.
19. # init the 'Plane' selected for 'SliceType'
20. slice1.SliceType.Origin = [-7.5, 13.5, 9.5]
21.
22. # init the 'Plane' selected for 'HyperTreeGridSlicer'
23. slice1.HyperTreeGridSlicer.Origin = [-7.5, 13.5, 9.5]
24.
25. # find source
26. prova31ottobre2_0002xyz = FindSource('Prova31ottobre2_0002.xyz')
27.
28. # find source
29. prova31ottobre2_0003xyz = FindSource('Prova31ottobre2_0003.xyz')
30.
31. # get active view
32. renderView1 = GetActiveViewOrCreate('RenderView')
33.
34. # show data in view
35. slice1Display = Show(slice1, renderView1, 'GeometryRepresentation')
36.
37. # get color transfer function/color map for 'Density'
38. densityLUT = GetColorTransferFunction('Density')
39.
40. # trace defaults for the display properties.
41. slice1Display.Representation = 'Surface'
42. slice1Display.ColorArrayName = ['POINTS', 'Density']
43. slice1Display.LookupTable = densityLUT
44. slice1Display.SelectTCoordArray = 'None'
45. slice1Display.SelectNormalArray = 'None'
46. slice1Display.SelectTangentArray = 'None'
47. slice1Display.OSPRayScaleArray = 'Density'
48. slice1Display.OSPRayScaleFunction = 'PiecewiseFunction'
49. slice1Display.SelectOrientationVectors = 'Momentum'
50. slice1Display.ScaleFactor = 8.25
51. slice1Display.SelectScaleArray = 'Density'
52. slice1Display.GlyphType = 'Arrow'
53. slice1Display.GlyphTableIndexArray = 'Density'
54. slice1Display.GaussianRadius = 0.41250000000000003
55. slice1Display.SetScaleArray = ['POINTS', 'Density']
56. slice1Display.ScaleTransferFunction = 'PiecewiseFunction'
57. slice1Display.OpacityArray = ['POINTS', 'Density']
58. slice1Display.OpacityTransferFunction = 'PiecewiseFunction'
59. slice1Display.DataAxesGrid = 'GridAxesRepresentation'
60. slice1Display.PolarAxes = 'PolarAxesRepresentation'
61.
```

```
62. # init the 'PiecewiseFunction' selected for 'ScaleTransferFunction'
63. slice1Display.ScaleTransferFunction.Points = [0.0, 0.0, 0.5, 0.0, 1.1952531
337738037, 1.0, 0.5, 0.0]
64.
65. # init the 'PiecewiseFunction' selected for 'OpacityTransferFunction'
66. slice1Display.OpacityTransferFunction.Points = [0.0, 0.0, 0.5, 0.0, 1.19525
31337738037, 1.0, 0.5, 0.0]
67.
68. # show color bar/color legend
69. slice1Display.SetScalarBarVisibility(renderView1, True)
70.
71. # update the view to ensure updated data information
72. renderView1.Update()
73.
74. # get opacity transfer function/opacity map for 'Density'
75. densityPWF = GetOpacityTransferFunction('Density')
76.
77. # set scalar coloring
78. ColorBy(slice1Display, ('POINTS', 'StagnationEnergy'))
79.
80. # Hide the scalar bar for this color map if no visible data is colored by i
t.
81. HideScalarBarIfNotNeeded(densityLUT, renderView1)
82.
83. # rescale color and/or opacity maps used to include current data range
84. slice1Display.RescaleTransferFunctionToDataRange(True, False)
85.
86. # show color bar/color legend
87. slice1Display.SetScalarBarVisibility(renderView1, True)
88.
89. # get color transfer function/color map for 'StagnationEnergy'
90. stagnationEnergyLUT = GetColorTransferFunction('StagnationEnergy')
91.
92. # get opacity transfer function/opacity map for 'StagnationEnergy'
93. stagnationEnergyPWF = GetOpacityTransferFunction('StagnationEnergy')
94.
95. # hide data in view
96. Hide(prova31ottobre2_0001xyz, renderView1)
97.
98. # hide data in view
99. Hide(prova31ottobre2_0002xyz, renderView1)
100.
101.     # hide data in view
102.     Hide(prova31ottobre2_0003xyz, renderView1)
103.
104.     # get animation scene
105.     animationScene1 = GetAnimationScene()
106.
107.     # Properties modified on animationScene1
108.
109.     #arange è una funzione che ad ogni iterazione mi crea un numero deci
male (salvato nella variabile x) che parte dal primo parametro (quindi prim
a iterazione x = 1.0).
110.     #il secondo parametro (651) indica il limite massimo (escluso) delle
iterazioni del ciclo for, quindi il loop si ripete fino a quando x = 650.0
111.     #il terzo parametro indica di quanto la variabile x incrementa ad og
ni iterazione (x = 1.0 poi 2.0 e così via)
112.     for x in numpy.arange(1.0, 21.0, 1.0):
113.         #ora animationTime prende il valore di x ad ogni iterazione quindi
si automodifica per ogni istante di tempo
114.         animationScene1.AnimationTime = x
115.
116.     # get the time-keeper
```

```

117.         timeKeeper1 = GetTimeKeeper()
118.
119.
120.         # save data
121.         #x è un numero decimale, dobbiamo salvarlo nel file come numero in
            tero quindi ho fatto la conversione (ESEMPIO: x = 2.0 -> timeIstant = 2)
122.         timeIstant = int(x)
123.         #ho parametrizzato la stringa in maniera tale che ad ogni iterazio
            ne cambi in base all'istante di tempo
124.         SaveData('C:/Users/Asus/Desktop/fds2paraview/Esportazione PARAVIE
            W/ply files/{timeIstant}.ply', proxy=slice1, ChooseArraysToWrite=0,
125.         PointDataArrays=['Density', 'IBlack', 'Momentum', 'Pressure',
            'StagnationEnergy'],
126.         CellDataArrays=['vtkGhostType'],
127.         FieldDataArrays=['Properties'],
128.         EnableColoring=1,
129.         ColorArrayName=['POINTS', 'StagnationEnergy'],
130.         LookupTable=stagnationEnergyLUT)
131.
132.         #=====
133.         # addendum: following script captures some of the application
134.         # state to faithfully reproduce the visualization during playback
135.         #=====
136.
137.         # get layout
138.         layout1 = GetLayout()
139.
140.         #-----
141.         # saving layout sizes for layouts
142.
143.         # layout/tab size in pixels
144.         layout1.SetSize(1436, 755)
145.
146.         #-----
147.         # saving camera placements for views
148.
149.         # current camera placement for renderView1
150.         renderView1.CameraPosition = [259.29620698968864, -
            103.80952188921118, 113.05374330736055]
151.         renderView1.CameraFocalPoint = [-
            7.472917453031861, 2.225933254074433, -1.0410232426023978]
152.         renderView1.CameraViewUp = [-
            0.3437718862766036, 0.13504563469955866, 0.9292919706714459]
153.         renderView1.CameraParallelScale = 45.13105914999115
154.
155.         #-----
156.         # uncomment the following to render all views
157.         # RenderAllViews()
158.         # alternatively, if you want to write images, you can use SaveScreen
            shot(...).

```

### A.3. ArrayTest Script per Unity

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class ArrayTest : MonoBehaviour
6. {
7.
8.     public GameObject[] numbers;
9.
10.
11.     // Start is called before the first frame update
12.     void Start()
13.     {
14.
15.     }
16.
17.     bool attractorMode = true, isRunning;
18.     int currentIndex;
19.
20.     // Update is called once per frame
21.     void Update()
22.     {
23.         // only invoke if the coroutine isn't already running
24.         if (attractorMode && !isRunning) StartCoroutine(Cycle());
25.     }
26.
27.     IEnumerator Cycle()
28.     {
29.         // these lines execute by the end of the frame in which Cycle() was
           invoked
30.         isRunning = true;
31.         numbers[currentIndex].SetActive(true);
32.         yield return new WaitForSeconds(0.25f);
33.
34.         // execution will not return to the next statement until 0.25 seconds
           later
35.         numbers[currentIndex].SetActive(true);
36.         Destroy(numbers[currentIndex], 0.05f);
37.
38.         // increment the index and ensure it loops around at the end
39.         currentIndex++;
40.         if (currentIndex >= numbers.Length) currentIndex = 0;
41.         isRunning = false;
42.
43.
44.
45.     }
46. }
```

## A.4. Vertex Color Shader

```
1. Shader "Custom/fbx_prova"
2. {
3.     Properties{
4.         _Color("Main Color", Color) = (1, 1, 1, 1)
5.         _SpecColor("Spec Color", Color) = (1, 1, 1, 1)
6.         _Emission("Emmision Color", Color) = (0, 0, 0, 0)
7.         _Shininess("Shininess", Range(0.01, 1)) = 0.7
8.         _MainTex("Base (RGB)", 2D) = "white" {}
9.     }
10.
11.     SubShader{
12.         Pass{
13.             Material{
14.                 Shininess[_Shininess]
15.                 Specular[_SpecColor]
16.                 Emission[_Emission]
17.             }
18.             ColorMaterial AmbientAndDiffuse
19.             Lighting On
20.             SeparateSpecular On
21.             SetTexture[_MainTex] {
22.                 Combine texture * primary, texture * primary
23.             }
24.             SetTexture[_MainTex] {
25.                 constantColor[_Color]
26.                 Combine previous * constant DOUBLE, previous
27.                 s * constant
28.             }
29.         }
30.
31.         Fallback "VertexLit", 1
32.     }
```

## A.5. Timer

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class Timer1 : MonoBehaviour
7. {
8.
9.     public Text timerMinutes;
10.    public Text timerSeconds;
11.    public Text timerSeconds100;
12.
13.    private float startTime;
14.    private float stopTime;
15.    private float timerTime;
16.    private bool isRunning = false;
17.
18.    // Use this for initialization
19.    void Start()
20.    {
21.        TimerReset(); }
22.
23.    public void TimerStart()
24.    {
25.        if (!isRunning)
26.        {
27.            print("START");
28.            isRunning = true;
29.            startTime = Time.time; }
30.    }
31.
32.    public void TimerStop()
33.    {
34.        if (isRunning)
35.        {
36.            print("STOP");
37.            isRunning = false;
38.            stopTime = timerTime; }
39.    }
40.
41.    public void TimerReset()
42.    {
43.        print("RESET");
44.        stopTime = 0;
45.        isRunning = false;
46.        timerMinutes.text = timerSeconds.text = timerSeconds100.text = "00"
47.    };
48.
49.    // Update is called once per frame
50.    void Update()
51.    {
52.        timerTime = stopTime + (Time.time - startTime);
53.        int minutesInt = (int)timerTime / 60;
54.        int secondsInt = (int)timerTime % 60;
55.        int seconds100Int = (int)(Mathf.Floor((timerTime - (secondsInt + minutesInt * 60)) * 100));
56.
57.        if (isRunning)
58.        {
```

---

```
59.         timerMinutes.text = (minutesInt < 10) ? "0" + minutesInt : minu
           tesInt.ToString();
60.         timerSeconds.text = (secondsInt < 10) ? "0" + secondsInt : seco
           ndsInt.ToString();
61.         timerSeconds100.text = (seconds100Int < 10) ? "0" + seconds100I
           nt : seconds100Int.ToString();
62.     }
63. }
64. }
```

## A.6. Fly Camera [12]

```
1. using UnityEngine;
2.
3. [RequireComponent(typeof(Camera))]
4. public class FreeFlyCamera : MonoBehaviour
5. {
6.     #region UI
7.
8.     [Space]
9.
10.    [SerializeField]
11.    [Tooltip("The script is currently active")]
12.    private bool _active = true;
13.
14.    [Space]
15.
16.    [SerializeField]
17.    [Tooltip("Camera rotation by mouse movement is active")]
18.    private bool _enableRotation = true;
19.
20.    [SerializeField]
21.    [Tooltip("Sensitivity of mouse rotation")]
22.    private float _mouseSense = 1.8f;
23.
24.    [Space]
25.
26.    [SerializeField]
27.    [Tooltip("Camera zooming in/out by 'Mouse Scroll Wheel' is active")]
28.    private bool _enableTranslation = true;
29.
30.    [SerializeField]
31.    [Tooltip("Velocity of camera zooming in/out")]
32.    private float _translationSpeed = 55f;
33.
34.    [Space]
35.
36.    [SerializeField]
37.    [Tooltip("Camera movement by 'W','A','S','D','Q','E' keys is active")]
38.    private bool _enableMovement = true;
39.
40.    [SerializeField]
41.    [Tooltip("Camera movement speed")]
42.    private float _movementSpeed = 10f;
43.
44.    [SerializeField]
45.    [Tooltip("Speed of the quick camera movement when holding the 'Left Shift' key")]
46.    private float _boostedSpeed = 50f;
47.
48.    [SerializeField]
49.    [Tooltip("Boost speed")]
50.    private KeyCode _boostSpeed = KeyCode.LeftShift;
51.
52.    [SerializeField]
53.    [Tooltip("Move up")]
54.    private KeyCode _moveUp = KeyCode.E;
55.
56.    [SerializeField]
57.    [Tooltip("Move down")]
58.    private KeyCode _moveDown = KeyCode.Q;
59.
```

```
60.     [Space]
61.
62.     [SerializeField]
63.     [Tooltip("Acceleration at camera movement is active")]
64.     private bool _enableSpeedAcceleration = true;
65.
66.     [SerializeField]
67.     [Tooltip("Rate which is applied during camera movement")]
68.     private float _speedAccelerationFactor = 1.5f;
69.
70.     [Space]
71.
72.     [SerializeField]
73.     [Tooltip("This keypress will move the camera to initialization position
74.     ")]
75.     private KeyCode _initPositonButton = KeyCode.R;
76.
77. #endregion UI
78.     private CursorLockMode _wantedMode;
79.
80.     private float _currentIncrease = 1;
81.     private float _currentIncreaseMem = 0;
82.
83.     private Vector3 _initPosition;
84.     private Vector3 _initRotation;
85.
86. #if UNITY_EDITOR
87.     private void OnValidate()
88.     {
89.         if (_boostedSpeed < _movementSpeed)
90.             _boostedSpeed = _movementSpeed;
91.     }
92. #endif
93.
94.
95.     private void Start()
96.     {
97.         _initPosition = transform.position;
98.         _initRotation = transform.eulerAngles;
99.     }
100.
101.     private void OnEnable()
102.     {
103.         if (_active)
104.             _wantedMode = CursorLockMode.Locked;
105.     }
106.
107.     // Apply requested cursor state
108.     private void SetCursorState()
109.     {
110.         if (Input.GetKeyDown(KeyCode.Escape))
111.         {
112.             Cursor.lockState = _wantedMode = CursorLockMode.None;
113.         }
114.
115.         if (Input.GetMouseButtonDown(0))
116.         {
117.             _wantedMode = CursorLockMode.Locked; }
118.
119.         // Apply cursor state
120.         Cursor.lockState = _wantedMode;
121.         // Hide cursor when locking
122.         Cursor.visible = (CursorLockMode.Locked != _wantedMode);
```

```
123.     }
124.
125.     private void CalculateCurrentIncrease(bool moving)
126.     {
127.         _currentIncrease = Time.deltaTime;
128.
129.         if (!_enableSpeedAcceleration || _enableSpeedAcceleration &&
!moving)
130.         {
131.             _currentIncreaseMem = 0;
132.             return; }
133.
134.         _currentIncreaseMem += Time.deltaTime * (_speedAccelerationF
actor - 1);
135.         _currentIncrease = Time.deltaTime + Mathf.Pow(_currentIncrea
seMem, 3) * Time.deltaTime;
136.     }
137.
138.     private void Update()
139.     {
140.         if (!_active)
141.             return;
142.
143.         SetCursorState();
144.
145.         if (Cursor.visible)
146.             return;
147.
148.         // Translation
149.         if (_enableTranslation)
150.         {
151.             transform.Translate(Vector3.forward * Input.mouseScrollID
elta.y * Time.deltaTime * _translationSpeed); }
152.
153.         // Movement
154.         if (_enableMovement)
155.         {
156.             Vector3 deltaPosition = Vector3.zero;
157.             float currentSpeed = _movementSpeed;
158.
159.             if (Input.GetKey(_boostSpeed))
160.                 currentSpeed = _boostedSpeed;
161.
162.             if (Input.GetKey(KeyCode.W))
163.                 deltaPosition += transform.forward;
164.
165.             if (Input.GetKey(KeyCode.S))
166.                 deltaPosition -= transform.forward;
167.
168.             if (Input.GetKey(KeyCode.A))
169.                 deltaPosition -= transform.right;
170.
171.             if (Input.GetKey(KeyCode.D))
172.                 deltaPosition += transform.right;
173.
174.             if (Input.GetKey(_moveUp))
175.                 deltaPosition += transform.up;
176.
177.             if (Input.GetKey(_moveDown))
178.                 deltaPosition -= transform.up;
179.
180.             // Calc acceleration
181.             CalculateCurrentIncrease(deltaPosition != Vector3.zero);
```

```
182.
183.         transform.position += deltaPosition * currentSpeed * _cu
    rrentIncrease; }
184.
185.         // Rotation
186.         if (_enableRotation)
187.         {
188.             // Pitch
189.             transform.rotation *= Quaternion.AngleAxis(
190.                 -Input.GetAxis("Mouse Y") * _mouseSense,
191.                 Vector3.right
192.             );
193.
194.             // Paw
195.             transform.rotation = Quaternion.Euler(
196.                 transform.eulerAngles.x,
197.                 transform.eulerAngles.y + Input.GetAxis("Mouse X") *
    _mouseSense,
198.                 transform.eulerAngles.z
199.             );
200.         }
201.
202.         // Return to init position
203.         if (Input.GetKeyDown(_initPositonButton))
204.         {
205.             transform.position = _initPosition;
206.             transform.eulerAngles = _initRotation; }
207.         }
208.     }
```

## A.7. Door Script

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. public class DoorScript : MonoBehaviour
5. {
6.     Animator anim;
7.     // Use this for initialization
8.     void Start()
9.     {
10.        anim = GetComponent<Animator>();
11.    }
12.    // Update is called once per frame
13.    void Update()
14.    {
15.    }
16.    void OnTriggerEnter(Collider other)
17.    {
18.        anim.SetTrigger("OpenDoor");
19.    }
20.    void OnTriggerExit(Collider other)
21.    {
22.        anim.enabled = true;
23.    }
24.    void pauseAnimationEvent()
25.    {
26.        anim.enabled = false;
27.    }
28. }
```

---

## Bibliografia e sitografia

- [1] “Norme tecniche di prevenzione incendi,” 2020.
- [2] S. Kose, “Emergence of an Aged Populace,” pp. 1–5.
- [3] L. Bibb and D. John, “Group inhibition of bystander intervention in emergencies,” *J. Pers. Soc. Psychol.*, vol. 10, no. 3, pp. 215–221, 1968.
- [4] J. D. Sime, “Escape Behaviour in Fires : ‘Panic’ or Affiliation?,” pp. 1–321, 1984.
- [5] T. J. Shields and G. Proulx, “The science of human behaviour: Past research endeavours, current developments and fashioning a research agenda,” *Fire Saf. Sci.*, no. April, pp. 95–114, 2000, doi: 10.3801/iafss.fss.6-95.
- [6] J. L. Bryan, *A Study of the Survivors Reports on the Panic in the Fire at the Arundel Park in Brooklyn, Maryland, 1/29/56*. 1957.
- [7] E.L. Quarantelli, “The sociology of panic,” *Think. Reason.*, 2001.
- [8] J. Strick, “Development of Safety Measures for Nightclubs,” 2014, [Online]. Available: <http://lup.lub.lu.se/student-papers/record/4522713/file/4522718.pdf>.
- [9] NFPA, “NFPA Case Study: Nightclub Fires,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1243, pp. 89–103, 1997.
- [10] S. Zanut, T. Villani, and I. Montanari, “Il codice e l’evacuazione in caso di incendio: agevolare l’esodo in emergenza,” *Antincendio*, no. novembre, pp. 74–89, 2017.
- [11] G. Ceresini, “Guida all’illuminazione di sicurezza,” 2004.
- [12] J. McClintock, T., Shields, T., Reinhardt-Rutland, A. & Leslie, “A behavioural solution to the learned irrelevance of emergency exit signage,” 2001.
- [13] E. R. Galea, H. Xie, D. Cooney, and L. Filippidis, “Active dynamic signage system: A full-scale evacuation trial,” *Intersci. Commun. Ltd*, no. 1996, pp. 303–314, 2015, [Online]. Available: <http://gala.gre.ac.uk/id/eprint/14348/>.
- [14] D. Nilsson, H. Frantzich, and W. Saunders, “Coloured flashing lights to mark emergency exits - Experiences from evacuation experiments,” *Fire Saf. Sci.*, no. December 2015, pp. 569–579, 2005, doi: 10.3801/IAFSS.FSS.8-569.
- [15] L. T. Wong and K. C. Lo, “Experimental study on visibility of exit signs in buildings,” *Build. Environ.*, vol. 42, no. 4, pp. 1836–1842, 2007, doi: 10.1016/j.buildenv.2006.02.011.
- [16] T. Jin, “Visibility through fire smoke,” vol. Journal of, 1978.
- [17] R. Q., Zhang, P., “Simulation of preceived visibility in smoke laden environment,” 2009.
- [18] R. Q., Zhang, P., “Quantitative validation of simulated visibility in smoke laden environment,” 2010.

- 
- [19] T. Paulsen, “The effect of escape route information on mobility and way finding under smoke logged conditions,” 1994.
- [20] K. McGrattan, R. McDermott, S. Hostikka, and J. Floyd, “NIST Special Publication 1019-5 Fire Dynamics Simulator ( Version 5 ) User ’ s Guide,” *Nist Spec. Publ.*, no. Version 5, 2010.
- [21] R. A. Klein, *SFPE handbook of fire protection engineering (1995)*, vol. 29, no. 1. 1997.
- [22] H. J. Kim and D. G. Lilley, “Heat release rates of burning items in fires,” *J. Propuls. Power*, vol. 18, no. 4, pp. 866–870, 2002, doi: 10.2514/2.6011.
- [23] M. Forsberg, J. Kjellström, H. Frantzich, A. Mossberg, and D. Nilsson, “The Variation of Pre-movement Time in Building Evacuation,” *Fire Technol.*, vol. 55, no. 6, pp. 2491–2513, 2019, doi: 10.1007/s10694-019-00881-1.
- [24] R. Lovreglio, E. Kuligowski, S. Gwynne, and K. Boyce, “Corrigendum to ‘A pre-evacuation database for use in egress simulations’ (Fire Safety J. 105, 2019, (107–128), (S0379-7112(18)30221-2),(10.1016/j.firesaf.2018.12.009)),” *Fire Saf. J.*, vol. 108, no. August, p. 102829, 2019, doi: 10.1016/j.firesaf.2019.102829.
- [25] K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, C. Weinschenk, and K. Overhold, “Fire Dynamics Simulator (FDS) User ’ s Guide,” *NIST Spec. Publ.*, vol. 6, 2020.
- [26] E. Ronchi *et al.*, “A Virtual Reality Experiment on Flashing Lights at Emergency Exit Portals for Road Tunnel Evacuation,” *Fire Technol.*, vol. 52, no. 3, pp. 623–647, 2016, doi: 10.1007/s10694-015-0462-5.
- [27] J. Moloney, B. Spehar, A. Globa, and R. Wang, “The affordance of virtual reality to enable the sensory representation of multi-dimensional data for immersive analytics: from experience to insight,” *J. Big Data*, vol. 5, no. 1, pp. 1–19, 2018, doi: 10.1186/s40537-018-0158-z.
- [28] <https://www.ctif.org/news/panic-and-human-behavior-fire-emergency-situations>
- [29] <https://www.internet4things.it/industry-4-0/realta-virtuale-aumentata-e-mista-differenze-e-applicazioni/>
- [30] <https://www.ingenio-web.it/6687-il-comportamento-umano-in-caso-di-incendio-i-modelli-di-evacuazione>
- [31] <https://forum.unity.com/>
- [32] <https://assetstore.unity.com/packages/tools/camera/free-fly-camera-140739>
- [33] <https://biblus.acca.it/rtv-attivita-di-intrattenimento-e-di-spettacolo-a-carattere-pubblico/>
-