ДИПЛОМНА РОБОТА
БАКАЛАВРА

ДРБ. ІП - 40.00.00.000 ПЗ

Група ІП-18-2

Тичинський Владислав

2022

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

УДК 004.4

ДИПЛОМНА РОБОТА БАКАЛАВРА

Тема: «Розробка алгоритмічної бізнес-логіки текстових реконструкцій та візуалізацій»

(назва згідно з наказом ректора)

Спеціальність – 121 "Інженерія програмного забезпечення"

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДРБ.ІП - 40.00.00.000 ПЗ

(позначення)

Студент

ІП-18-2 /Тичинський В.І./
(шифр групи) (підпис) (дата) (розшифровка підпису)

Керівник роботи

доцент /Михайлюк І.Р./
(посада) (підпис) (дата) (розшифровка підпису)

Консультанти:

з питань нормоконтролю

доц., к.т.н /В.Я. Піх/
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту
Завідувач кафедри

проф., д.т.н. /В.І.Шекета/
(посада) (підпис) (дата) (розшифровка підпису)

2022
(рік)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
проф. **В.І. Шекета**
“ ____ ” _____2022 р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ БАКАЛАВРА СТУДЕНТОВІ

Тичинському Владиславу Ігоровичу

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) «Розробка алгоритмічної бізнес-логіки текстових реконструкцій та візуалізацій»
керівник проекту (роботи) Михайлюк Ірина Романівна, доцент
затверджені наказом закладу вищої освіти від **“ __ ”** __ 2022 р. № _____

2. Строк подання студентом проекту (роботи) **12 червня 2022 р.**

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд існуючих концепцій, рішень та сервісів в даній області
2. Алгоритмічні та математичні рішення для задач даного класу
3. Власне оригінальне алгоритмічне та програмне рішення задач
4. Тестування розробленого продукту

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Основні сили для напрямку агента (рис. 1.1, ст. 27)
2. Блок-схема методології (рис. 2.1, ст. 50)
3. Налаштування кінозалу за замовчуванням (рис. 3.7, ст. 79)
4. Моделювання з використанням висувних сидінь (рис. 3.12, ст. 84)
5. Симуляція, що показує схожість із Uphaar Cinema у випадку КЗ (рис. 3.18)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення теми роботи	01.02.2022	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	01.03.2022	виконано
3	Побудова моделі або алгоритму власного рішення	01.04.2022	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	01.06.2022	виконано
5	Виконання тестування продукту	08.06.2022	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2022	виконано

Студент _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

В дипломній роботі була запропонована модель програмного забезпечення здатна працювати з великою кількістю параметрів, які можна використовувати для моделювання різних сценаріїв і множинних середовищ з великою кількістю агентів. Це може дозволити дослідникам і планувальникам спостерігати за впливом різних ситуацій на етапі проектування та планування без суттєвих затрат.

КЛЮЧОВІ СЛОВА: ПРОГРАМНЕ РІШЕННЯ, ВІЗУАЛІЗАЦІЯ, СЦЕНАРІЇ ПЛАНУВАННЯ, МОДЕЛЮВАННЯ, ПАРАДИГМИ ПРОЕКТУВАННЯ.

ANNOTATION

The thesis proposed a software model capable of working with a large number of parameters that can be used to simulate various scenarios and multiple environments with a large number of agents. This can allow researchers and planners to observe the impact of different situations at the design and planning stage without significant costs.

KEY WORDS: SOFTWARE SOLUTION, VISUALIZATION, PLANNING SCENARIOS, MODELING, DESIGN PARADIGMS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 6
ВСТУП 7
1. ОГЛЯД ІСНУЮЧИХ КОНЦЕПЦІЙ, РІШЕНЬ ТА СЕРВІСІВ В ДАНІЙ ОБЛАСТІ 11
1.1 Основні інциденти предметної області 11
1.2 Пов'язані моделі програмного забезпечення 12
1.3 Список параметрів для порівняння моделі програмного забезпечення 22
1.4 Огляд літератури по темі роботи 23
2 АЛГОРИТМІЧНІ ТА МАТЕМАТИЧНІ РІШЕННЯ ДЛЯ ЗАДАЧ ДАНОГО КЛАСУ 37
2.1 Існуючі моделі програмного забезпечення 37
2.2 Запропонована методика 45
2.3 Доступність параметрів в Unity 61
3 РОЗРОБКА ОРИГІНАЛЬНОГО АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО РІШЕННЯ ВІЗУАЛІЗАЦІЇ ТЕКСТОВИХ РЕКОНСТРУКЦІЙ 64
3.1 Фізичні зміни, які розглядаються для моделювання 64
3.2 Конфігурації апаратного та програмного забезпечення 70
3.3 Експериментальне дослідження та результати імітаційного моделювання 70
ВИСНОВКИ 88
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА 89
БІБЛІОГРАФІЧНА ДОВІДКА

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ТЗ – Технічне завдання

БД – База даних

СУБД - Система управління базами даних

ПЗ - Програмне забезпечення

ВСТУП

Кожен громадський простір має бути обладнаний ефективною та функціональною системою процедур/протоколів у надзвичайних ситуаціях, щоб забезпечити безпеку людей, які можуть потрапити всередині у разі будь-якого непередбачуваного інциденту. Як випливає з цієї фрази, непередбачуваний інцидент — це той інцидент, який важко передбачити з точки зору часу його виникнення, форми або засобів, через які він може статися і, у свою чергу, загрожувати людським життям. Таким чином, не можна недооцінити необхідність створення ретельної та добре перевіреної системи евакуації для кожного громадського місця, такого як школи, кінотеатри, торгові центри тощо. Системи евакуації є основною вимогою, особливо під час таких небезпечних або смертельних надзвичайних ситуацій .

Існують програмні моделі систем евакуації, що моделюють різні сценарії надзвичайних ситуацій з метою зменшення кількості постраждалих. Якщо хтось хоче побудувати безпечне місце, він/вона повинен переконатися, що коридори, ширина дверей, розміщення виходів або розташування місць мають бути побудовані таким чином, щоб ніхто не постраждав або не поранився під час будь-якої екстреної евакуації. Це зменшить витрати, а також час, якщо копія будівлі/місця, розроблена за допомогою комп’ютера, ідеально працює в комп’ютерному моделюванні до початку фактичного будівництва. Цю копію будівлі/місця, розробленого за допомогою комп’ютерного програмного забезпечення, часто називають 3D-середовищем. Крім того, якщо місце вже побудоване, але потребує певних змін у середовищі, його слід протестувати на комп’ютерному моделюваному середовищі, перш ніж вносити ці зміни. Ці зміни у фізичному середовищі, такі як зміна ширини дверей, зміна розташування сидінь або складання нового комплекту стільців, називаються фізичними змінами. Наприклад, якщо будь-який інженер-будівельник хоче збільшити ширину коридорів у певній будівлі на 1 м, ми повинні спробувати реалізувати це спочатку на комп’ютері, щоб перевірити ефективність цієї зміни,

оскільки процес збільшення ширини передпокоїв на реальна будівля може коштувати багато в порівнянні зі змінами в комп'ютерно змодельованому середовищі. Однак більшість доступних програмних моделей працюють із 3D-середовищем, яке попередньо визначено всередині програмного пакета. Жодна з цих моделей програмного забезпечення не забезпечує метод/процес для включення будь-яких можливих фізичних змін у внутрішньому середовищі. Таким чином, у цій дипломній роботі ми розробляємо програмну модель систем евакуації для внутрішнього середовища, яка може включати будь-які можливі фізичні зміни до того, як вони будуть фактично побудовані або побудовані. Наша програмна модель допомагає оцінити фізичні зміни, щоб перевірити їх ефективність у разі надзвичайної ситуації для різних сценаріїв на основі інцидентів, що трапилися в минулому. Крім того, ми можемо впровадити будь-які можливі фізичні зміни в 3D-середовищі всередині приміщень, які можуть відбутися в майбутньому за допомогою Autodesk maya і механізму розробки ігор Unity.

Тим не менш, ніхто не може передбачити інциденти, які можуть статися в будь-який час і в будь-якому місці. Таким чином, сьогодні уряди суворо зобов’язують усі закриті громадські місця (тобто школи, кінозали, аудиторії тощо) проводити тренування для запобігання шкоди у разі пожежі. Крім того, їм необхідно запровадити систему екстреної евакуації, щоб допомогти всім, хто опинився всередині, вибратися або принаймні захистити себе від будь-яких травм. Більшість зареєстрованих інцидентів у всьому світі спричинені пожежами.

Пожежні тренування та інші «імітаційні вправи з евакуації», як правило, не можуть точно підготуватися до небезпечних для життя фактичних сценаріїв евакуації, оскільки люди, як правило, часто ігнорують небезпеку або не мають доступу до ситуації разом . Протягом багатьох років було розроблено багато різних програмних моделей, щоб надати дизайнерам способи прогнозування часу евакуації для будь-якого типу внутрішнього середовища, такого як

багатоповерхові будинки, концертні майданчики тощо. Велика кількість програмних моделей для моделювання пішоходів було розроблено в багато дисциплін, таких як робототехніка, комп'ютерна графіка, динаміка евакуації тощо, які можуть бути використані для вивчення різноманітних ситуацій . Однак використання комп'ютерної графіки для моделювання цього загального явища все ще є важкою роботою. Природний натовп, як правило, є колективним, оскільки «загальну мету» або «взаємне призначення» поділяють багато індивідуумів або тварин . Однак ці підсвідомі рішення важко обчислити. Однак моделювання зазвичай вимагає значних обчислювальних ресурсів. Через такі обмеження, як обчислювальні витрати, деякі ранні моделі програмного забезпечення, як правило, зосереджуються на натовпі як на єдиному динамічному методі, а не на його рухах.

За останні роки продуктивність комп'ютера значно зросла. Сучасні комп'ютери здатні обробляти програми моделювання натовпу та забезпечувати більш досконалу графіку та обчислювальну потужність. Таким чином, продуктивність сучасних комп'ютерів може підтримувати моделювання в реальному часі. Основна ідея симуляції натовпу на основі агентів полягає в управлінні динамікою натовпу, яка створюється автономно. Ця модель вважається однією з найкращих моделей для моделювання натовпу з реалістичною поведінкою, оскільки вона надає методи контролю руху кожного агента окремо. Однак вартість моделювання відносно висока. Зменшення обчислювальних витрат при одночасному підвищенні продуктивності моделювання, таких як масштабованість, гнучкість і реалістичність, залишаються затребуваними в області агентного моделювання натовпу.

Протягом майже двох десятиліть компанії використовують імітаційні моделі/програмне забезпечення для планування евакуації. Дослідники працюють над цим програмним забезпеченням, щоб підвищити їх ефективність, використовуючи останній алгоритм s. Вони також працюють над створенням нового програмного забезпечення з різними умовними параметрами.

Однак існуючі моделі програмного забезпечення мають серйозний недолік роботи лише з набором тривимірних середовищ, доступних всередині цієї моделі програмного забезпечення . Крім того, можливі фізичні зміни, які ще не створені, або ті зміни, які можуть відбутися в майбутньому, не можуть бути інтегровані в ці моделі програмного забезпечення . Таким чином, ми очікуємо скорочення людських жертв на значну кількість шляхом розробки програмної моделі, яка включатиме будь-які можливі фізичні зміни всередині внутрішнього середовища до того, як вони будуть фактично побудовані/збудовані або які можуть статися в майбутньому, щоб імітувати будь-який реальний сценарій евакуації.

Основна мета цього дослідження полягає в тому, щоб створити базову програмну модель систем евакуації для внутрішнього середовища, що включатиме деякі можливі фізичні зміни, перш ніж вони будуть фактично сконструйовані або побудовані з використанням движка розробки гри unity та *Autodesk Maya*. Ця базова модель програмного забезпечення може бути використана для планування та управління надзвичайними ситуаціями, а також для майбутніх досліджень.

1. ОГЛЯД ІСНУЮЧИХ КОНЦЕПЦІЙ, РІШЕНЬ ТА СЕРВІСІВ В ДАНІЙ ОБЛАСТІ

1.1 Основні інциденти предметної області

Більшість зареєстрованих інцидентів у всьому світі в основному пов'язані з пожежами. Ці інциденти здебільшого відбувалися через недотримання політики безпеки під час будівництва цих приміщень. Нижче наведено деякі з прикладів цих прикрих інцидентів:

Щоб зменшити кількість постраждалих, подібних до обговорюваних інцидентів, багато комерційних структур або окремих осіб у всьому світі використовують імітаційні моделі для вивчення та аналізу часу евакуації та ефективності під час надзвичайних ситуацій. Ці програмні моделі надають широкий спектр функцій для моделювання необхідних ефектів під час надзвичайних ситуацій і можуть бути класифіковані на основі методів як: (1) моделі на основі потоку, (2) моделі на основі агентів, (3) моделі на основі діяльності. .

Моделі на основі потоку:

Моделі на основі потоку зосереджені на всьому натовпі як єдиному цілі, а не на його компонентах. Ця модель в першу чергу використовується для оцінки потоку руху великого і щільного натовпу в даному середовищі [51].

Моделі на основі агентів:

Моделювання на основі агентів базується на окремому активному компоненті системи. Активні ідентифікатори, відомі як агенти, розпізнаються за допомогою агентного моделювання, і їх поведінка визначається. Це можуть бути окремі особи, будинки, транспортні засоби, техніка, продукти або будь-що, що має відношення до системи [14]. *Агентні моделі* — це комп'ютерні моделі, які фіксують поведінку осіб/агентів у певному середовищі. Найпопулярнішим прикладом цієї техніки є комп'ютерна гра SIMCity, у якій люди чи інші сутності взаємодіють один з одним та/або своїм середовищем. Ця

дисертація в основному зосереджена на реалізації програмної моделі для евакуації з використанням техніки моделювання на основі агентів, в якій сценарії складаються з різних ідентичностей, що зображують кілька видів поведінки.

Моделі на основі діяльності:

Моделі, засновані на діяльності, засновані на щоденній моделі діяльності людей. Моделювання на основі діяльності передбачає, коли, де, як довго, для та з ким буде виконуватися кожна діяльність [5].

Як ми обговорювали, моделі програмного забезпечення класифікуються на основі методів. Також ці моделі програмного забезпечення можна класифікувати на основі доступності для загального користування.

1.2 Пов'язані моделі програмного забезпечення

У цьому розділі ми обговорюємо різні комерційні та вільно доступні моделі програмного забезпечення для моделювання натовпу та функції/параметри, які ці моделі включають. Ці моделі програмного забезпечення мають як переваги, так і недоліки. Однак основним недоліком усіх цих програмних моделей є те, що вони можуть працювати лише з набором тривимірних середовищ, доступних всередині моделі програмного забезпечення. Крім того, жодна з цих моделей програмного забезпечення не надає методу/процесу для включення будь-яких можливих фізичних змін у внутрішньому середовищі до того, як вони будуть фактично побудовані/побудовані, або які можуть статися в майбутньому.

EXODUS – це вільнодоступна модель програмного забезпечення, яка є загальнодоступною, в якій агенти мають фіксований рівень знайомства з будовою, спритністю та терпінням. Модель імітує аварійну ситуацію великого натовпу з огорожі, а також враховує затримку руху або можливе припинення через екстремальну спеку або вплив токсичних газів. Загальна модель була

розроблена в різних версіях, які відрізняються залежно від кількох різних контекстів, у яких може відбуватися евакуація, включаючи кораблі (maritimeEXODUS), літаки (airEXODUS) та будівлі (buildingEXODUS) [15].

Оскільки EXODUS є переважно агентною моделлю, переміщення індивідів у ній встановлюється фіксованим набором правил руху. Модель в цілому складається з п’яти взаємодіючих підмоделей : рух, поведінка, пасажир (агент), небезпека та токсичність. Наприклад, модель небезпеки генерує значення, які відповідають конкретній конфігурації загрози в змодельованому середовищі. Модель токсичності визначає рівні впливу токсичних речовин, що потім впливає на значення змінних, пов’язаних з поведінкою агента, що, у свою чергу, впливає на обчислення моделі руху. EXODUS є однією з групи моделей, які накопичили вражаючу сукупність факторів, які інформують про більш реалістичні сценарії евакуації. Навіть після стількох складнощів, EXODUS не забезпечує моделювання для осіб з обмеженими фізичними можливостями/повільно рухаються агентів, що створює значну проблему в сучасному світі.

WAYOUT – це безкоштовна програма для загального користування, яка була розроблена для розрахунку транспортного потоку будівлі в аварійних ситуаціях з багатокімнатної і, можливо, багатоповерхової будівлі. Враховується лише об’єднання транспортних потоків. Модель розглядає декілька нелінійних взаємодій потоків з різних приміщень, і заснована на залежності швидкості руху від щільності. Враховуються потоки тільки здорових людей. Психологічні фактори не включені. Програма WAYOUT розглядає до 200 кімнат, з'єднаних у мережу, що об'єднується, до одного виходу. Обмеження на кількість кімнат було введено довільно, щоб обмежити вимоги до пам'яті комп'ютера, і може бути розширене, якщо необхідно, таким чином обмежуючи модель мережею злиття. Це змінює проблему з імовірнісної на детерміновану. WAYOUT розглядає його потоки від дверей до дверей [7].

Програма WAYOUT використовувалася в Австралії протягом кількох років і застосовувалася в практичних інженерних ситуаціях для таких випадків, як багатоповерхові будинки, стадіони та торгові центри, а також в інших, більш простих випадках.

PEDROUTE — це модель програмного забезпечення, яка є у вільному доступі, яка є загальнодоступною, яка використовує метод динамічного присвоєння для імітації руху пасажирів через станцію. Модель також імітує накопичення черги станцій та її затухання, а також ефект блокування черг у сусідніх регіонах. Модель включає різні методи розподілу пасажирів, які можна використовувати для моделювання різних ситуацій поведінкового та фізичного розташування. Методики варіюються від визначених користувачем фіксованих шляхів до динамічних присвоєнь з використанням рівноважних або стохастичних алгоритмів присвоєння. Маршрутні рішення моделі враховують відправлення та пункти призначення пасажира, доступні йому на станції, як правило, дорожні в'їзди та платформи. Користувач визначає ці розташування, як і зв'язки між цими місцями. PEDROUTE враховує швидкість ходьби та затримки для мандрівників під час процесу моделювання запитів пасажирів через станцію та підсумовує місцезнаходження людей протягом визначених користувачем періодів. Потік з одного регіону в інший залежить від кількості людей у цьому регіоні та від того, скільки часу потрібно для переміщення через нього. Модель передбачає, що час є функцією щільності пасажирів у цій зоні [8]. PEDROUTE дозволяє користувачеві подивитися на рівень обслуговування та коефіцієнти обслуговування для кожної зони розташування станції, а також кількість пасажирів, присутніх у кожній зоні протягом певного часу.

Програма моделювання евакуації SIMULEX версії 2.0, яка є у вільному доступі загальнодоступним надбанням, яка допомагає аналізувати переміщення груп. Він надає певний набір параметрів для кожної «людини», щоб швидкість ходьби оцінювалася незалежно від середньої щільності групи в певній місцевості. Модель дозволяє кожній людині вибрати швидкість ходьби. Крім

цього вдосконалення, програма також враховує кілька інших факторів, які включені в створення алгоритмів руху, таких як фізичні рухи та жести, як-от похитування та скручування тіла, близькість інших евакуйованих, форма будівлі, структура та вік, визначені параметри для осіб у віці 12-55 років, які мають соціальне значення, але не ґрунтуються на уявленнях чи інформації про соціальні відносини, культуру чи групову інтеграцію. Натомість програма передбачає наявність раціонального агента, здатного оцінити оптимальний шлях евакуації та здатність агента уникати фізичних перешкод та обігнати інших осіб, які концептуалізуються як перешкоди для пересування [9].

FDS+Evac є вільно доступною загальнодоступною програмною моделлю, яка одночасно імітує процеси пожежі та евакуації. Його також можна використовувати лише для імітації процесу виходу людини без будь-яких впливів вогню, наприклад, протипожежна тренування . FDS+Evac розглядає кожного евакуйованого як окрему сутність або «агента», що має свої особисті властивості та методи втечі. Рух агентів моделюється за допомогою двовимірних площин, які представляють поверхи будівлі. Фундаментальний алгоритм вихідного руху вирішує рівняння руху в безперервному двовимірному просторі та часі для кожного агента, тобто FDS+Evac виконує деяку штучну молекулярну динаміку для агентів [2]. Fire Dynamics Simulator (FDS), базова програма моделювання пожежі, є моделлю розрахункової гідродинаміки (CFD) потоку рідини, керованої пожежею. FDS публікується на мові програмування Fortran 90. Він написаний мовами програмування C/OpenGL разом із використанням супутньої програми Smokeview, яка представляє результати моделювання графічно [2].

SimWalk — це безкоштовне програмне забезпечення для моделювання пішоходів, яке є загальнодоступним, розроблене компанією Savannah Simulations AG, Швейцарія. Це програмне забезпечення можна використовувати для стандартного моделювання евакуації натовпу, але моделювання звичайних ситуацій має обмеження. Можна визначити лише

початкову зону, точку шляху (середню точку) та точку виходу для моделювання в стандартному режимі. SimWalk може моделювати відносно великі натовпи і використовується кількома інженерними та науково-дослідними інститутами. Simwalk надає утиліту SimDraw, яка може імпортувати проекти AutoCAD, створювати та змінювати нові проекти середовища. Програмне забезпечення забезпечує невелику різницю між агентами (затримка руху та швидкість). Пішоходи використовують статичні шляхи, а отже, його можна використовувати для моделювання переміщень всередині будівель. Однак це програмне забезпечення не можна використовувати для такого руху, як Tawaf. Основна модель Сімуолка заснована на моделі соціальних сил. Маршрути руху, навантаження в кожній зоні, лічильники, потік інформації, рівень обслуговування, швидкість і час, затрачений окремим пішоходам на досягнення місця призначення, запис і відтворення, є одними з результатів цього програмного забезпечення [2].

Legion — це програмне забезпечення, яке є у вільному доступі у суспільному надбанні, розроблене Legion International Ltd., Великобританія. Це програмне забезпечення пройшло багато років дослідження руху та поведінки натовпу. Програмне забезпечення може моделювати сотні пішоходів на одному комп'ютері, і його можна використовувати як для сценаріїв евакуації, так і для стандартних досліджень, не пов'язаних з надзвичайними ситуаціями. Він може вказувати вхід, виїзд і маршрути руху. У Legion окремі пішоходи вирішують, виходячи зі своїх цілей і можуть мати різний рівень знань, досвіду та цілей. Вони також мають уявлення про навколишнє середовище.

Ця модель використовує безперервний алгоритм «шляху найменших зусиль» для моделювання рухів. Серед даних, які може надати програмне забезпечення, є локальні рівні щільності, перевантажені зони, час, необхідний для людей до місця призначення, вихідні дані про рух, відео та зображення [23].

MassMotion — це вільнодоступна модель програмного забезпечення для моделювання, яка є загальнодоступною, яка розроблена, щоб дати можливість

фахівцям з проектування та планування швидко тестувати й аналізувати рух людей у багатьох середовищах. Він забезпечує користувачам набір інструментів для створення та модифікації 3D-середовища, визначення операційних сценаріїв, виконання динамічного моделювання та розробки надійного аналізу [4].

MassMotion моделює реальні простори, розбиваючи ці простори на частини та класифікуючи частини відповідно до функцій. Люди в симуляції MassMotion знають обходити перешкоду, оскільки вона позначена як бар'єр. Швидкість руху зменшується при підйомі по поверхні, оскільки ця поверхня позначена як сходи. Спосіб розташування класифікованих об'єктів може мати значний вплив на те, як люди орієнтуються в просторі, впливаючи на їх швидкість, моделі пересування та вибір маршруту. Основними елементами сцени є підлога, ланка, сходи, пандус, ескалатор, шлях, портал і бар'єр [5].

Кожна людина в симуляції MassMotion є автономним агентом. Кожен агент має здатність контролювати своє середовище та реагувати на нього відповідно до унікального набору характеристик і цілей. Агенти створюються та розміщуються на сцені за допомогою подій [5].

Pathfinder — це комерційний симулятор аварійного виходу з вбудованим інтерфейсом користувача та анімованими 3D-результатами. Це дає змогу швидше оцінювати моделі евакуації та створювати більш реалістичну графіку, ніж з іншими симуляторами. Pathfinder підтримує імпорт файлів DXF і DWG у форматі AutoCAD. Інструмент витягування підлоги Pathfinder дозволяє швидко використовувати імпортовану геометрію для визначення пішохідного простору людей для режиму евакуації. Щоб представити геометрію моделі, Pathfinder використовує трикутну тривимірну сітку. Як наслідок, Pathfinder може точно представляти геометричну інформацію та криві [6].

Pathfinder підтримує два режими моделювання. У режимі керування агенти, уникаючи інших пасажирів і бар'єрів, самостійно продовжують до своєї мети. Швидкість потоку дверей не визначена, але є наслідком взаємодії

мешканців один з одним і з кордонами. Агенти використовують поведінку, яка відповідає правилам SFPE, у режимі SFPE, зі швидкістю ходьби залежно від щільності та обмежень потоку до дверей. Результати SFPE забезпечують вирішальну базову лінію для порівняння з іншими результатами, але розрахунки SFPE не заважають кільком особам займати той самий простір [7].

AllSafe — це модель комерційного програмного забезпечення, яка визначає, чи знаходяться мешканці під загрозою, залежно від вхідних даних для будівлі, використання будівлі, мешканців та сценарію пожежі. Вона має глобальну перспективу. Модель приписує поведінкові характеристики групам мешканців. У мешканців є глобальна перспектива будівництва, оскільки вони можуть подорожувати лише до одного виходу. Будівля вводиться в модель через серію вузлів. Були зроблені спроби перевірити його з іншими моделями, такими як Simulex. Модель була розроблена для розрахунку часу евакуації, коли мешканці не знають про пожежу пізніше за сценарієм. Основний розрахунок включає часові затримки перед евакуацією та оцінку часу ходьби мешканців [8].

CRISP — це комерційна модель сценаріїв усіх пожеж з Монте-Карло. Підмоделі CRISP представляють такі фізичні об’єкти, як кімнати, двері, вікна, сигналізації та сповіщувачі, меблеві вироби тощо, шари теплого диму та індивідів. Рандомізовані елементи включають початкові обставини, такі як різні відкриті або закриті вікна та двері, кількість, тип і розташування людей у будівлі, камін і тип палаючого продукту. Основна конструкція CRISP — це двошарова зонна модель потоку диму для різних приміщень у поєднанні з комплексною моделлю поведінки та руху людини. Контролер Монте-Карло контролює всі фізичні об’єкти, завдяки чому кожен з них виконує роботу для кожного кроку часу. Контролер також керує всіма введенням і виведенням, ініціалізацією для кожного запуску, і кожен запуск запускається автоматично. Функції включаються з будь-якого розподілу для отримання випадкових чисел. Ітеративно обчислення виконуються для кожного запуску зі змінними

інтервалами часу, щоб забезпечити ефективність, точність і стабільність програми.

Вважається, що люди приймають різні поведінкові ролі, природно або завдяки навчанню. Залежно від стану навколишнього середовища їх поведінку можна визначити як дії, від яких можна відмовитися і замінити їх свіжими. Раціональні рішення приймаються на основі поточних знань (які можуть бути обмеженими та/або неправильними) [3].

EGRESS — це комерційна модель програмного забезпечення, призначена як простий та ефективний спосіб покращити наші знання про евакуаційні заходи з певної структури. Він був задуманий як загальний інструмент моделювання, який пропонує необхідні засоби для ефективного моделювання різноманітних сценаріїв, від театрів до офісних будівель, залізничних вокзалів і човнів. EGRESS особливо корисний для моделювання евакуації з будівель і споруд, де немає даних про навчання евакуації. Це стосується нових будинків на етапі проектування, або будинків, які були побудовані через характер діяльності або велику кількість залучених осіб. Під час моделювання EGRESS площа підлоги споруди покривається осередками, еквівалентними мінімальній площі, яку б займала особа. Моделювання EGRESS просувається шляхом переміщення індивідуумів (автоматів) по гексагональній сітці від клітинки до клітинки відповідно до зваженого кидка кубика. Цей метод притаманний використанню клітин з розміром, еквівалентним мінімальній області, яку займає індивід [3].

EGRESS був призначений як простий та ефективний засіб для покращення нашого розуміння та вдосконалення механізмів евакуації. Він ідеально підходить для моделювання евакуації з великого офісного комплексу, торгового центру, станції метро, терміналу аеропорту або іншої складної будівлі з великою кількістю людей. Він пропонує корисні дані для проектувальників будівель, власників, мешканців та аварійних служб для моделювання процесу евакуації [3].

SGEM — це комерційна модель, яка бере просторову геометрію будівлі (або набору всередині будівлі) і розділяє обстановку на площинні сітки. Відносна координата представляє кожного умовного мешканця в будівлі, а модель руху індивіда в обстановці моделюється шляхом розв'язування відповідних різницевих рівнянь. Можна брати до уваги вплив і поведінку між людьми. Оскільки просторове планування будівлі представлено системою сітки, модель отримала назву просторово-сіткової моделі евакуації (SGEM) [3].

Люди на пожежній сході з внутрішніх приміщень, дверей, коридорів, дверей, вихід у безпечне місце крок за кроком. Подібно до інших моделей, вони визначають огорожу будівлі в мережу (вузли/дуги) відповідно до можливого напрямку евакуації людей або оптимізації маршруту евакуації. У моделі кожен вузол (зона) може представляти собою кімнату, коридор або частину холу. Вузли з'єднані зі своїми сусідами за допомогою отворів, таких як двері, виходи тощо. Характеристику кожного вузла можна описати, записавши координацію налаштування. Напрямок потоку для вузлів також визначається відносно положень кінцевих точок виходу. Вибір виходу всередині вузла визначається різними факторами навколишнього середовища, такими як відстань до виходу, наявність знаків виходу, візуальна доступність тощо, а також особисті характеристики, такі як знайомство з маршрутами тощо. У кожному вузлі люди описуються як особи, розташовані на сітках, що представляють будівлі (або обстановку). Вважається, що поведінку кожної особистості можна представити її просторовим розташуванням і вектором швидкості руху. Вектор швидкості кожної людини є функцією різних впливів навколишнього середовища всередині будівлі, таких як вплив інших людей, геометрія доріжок, поведінкова реакція тощо. Розташування кожної людини в просторі може бути представлено його/її відносні координати в системі. Записи координат у різні моменти часу описують траєкторії руху людей [3].

SGEM можна використовувати для прогнозування траєкторії руху кожної особи під час евакуації. Основний принцип полягає в тому, що він бере геометрію будівлі та розбиває обстановку на площинні сітки [4].

EXIT89 — це комерційна модель, яка має такі недоліки, як соціальна взаємодія та реакція групи, подібні до SIMULEX. EXIT89 включає індивідуальні розміри тіла (американський, радянський або австрійський) і дозволяє вказати кількість осіб з обмеженими фізичними можливостями, але він не включає тілесні дії та жести [5]. Він також враховує зустрічну реакцію евакуйованих, чий шлях під час виходу заблокований скупченням диму біля виходу. Модель визначає час подорожі як функцію щільності та швидкості в межах побудованої мережі вузлів і дуг. Алгоритм «найкоротшого маршруту» поєднується з індивідуальною перспективою для кожного евакуйованого, щоб відстежувати шлях і прогрес окремих евакуйованих. Однак усі особи, що перебувають у певному вузлі, спочатку проходять той самий шлях, визначений користувачем, або найкоротший відомий шлях до виходу. Крім того, користувач також може встановити відсоток мешканців, яким призначено час затримки. Ці подвійні функції (окремий шлях для всієї групи та затримка) неявно імітують поведінку групи [6].

Суттєвим недоліком залишається те, що кожен окремий евакуйований демонструє певні фізичні характеристики, які впливають на перебіг евакуації; однак ці характеристики позбавлені інтерактивних соціальних характеристик, таких як моніторинг інших, керівництво, колективна оцінка та колективна угода про відповідне реагування [6].

Кожна модель програмного забезпечення, яка обговорюється, має як переваги, так і недоліки симуляції натовпу. Однак усі ці моделі програмного забезпечення мають серйозний недолік роботи лише з набором тривимірних середовищ, доступних всередині цієї моделі програмного забезпечення. Крім того, будь-які можливі фізичні зміни, які ще не створені, або ті зміни, які можуть відбутися в майбутньому, не можуть бути інтегровані в ці моделі

програмного забезпечення . Крім того, зовнішні файли не можна імпортувати в ці моделі програмного забезпечення, щоб додати нове 3D-середовище для моделювання. Ми розглянемо набір параметрів/функцій для ретельного порівняння цих програмних моделей.

1.3 Список параметрів для порівняння моделі програмного забезпечення

Параметри/характеристики, які ми розглядаємо для порівняння, можна розділити на два: основні ознаки та особливі ознаки.

Основними характеристиками моделі є обов'язкові параметри, які можуть визначати роботу моделювання. Нижче наведено основні характеристики, які ми розглядаємо для порівняння моделей програмного забезпечення:

- Доступність для громадськості
- Візуалізація моделі
- Мета
- Переміщення агентів
- Включення даних про пожежу
- Комп'ютерне проектування (CAD)

Особливістю моделі є додаткові параметри, які модель може мати або не мати. Однак ці особливості підвищують рівень сценаріїв або можливостей у конкретній моделі.

Нижче наведено особливості, які ми розглядаємо для порівняння моделей програмного забезпечення:

- Протитечний
- Вихід заблоковано
- Токсичність агентів
- Непрацездатні/повільні агенти
- Вплив умов пожежі на агенти

Ми обговорюємо ці особливості під час порівняння цих моделей програмного забезпечення. Окрім цих програмних моделей для моделювання натовпу, ми також розглядаємо ряд наукових робіт від різних авторів, що працюють у сфері агентного моделювання для моделювання натовпу. Усі ці наукові роботи містять методи оцінки тимчасових результатів та ефективності евакуації .

1.4 Огляд літератури по темі роботи

Декілька авторів запропонували моделі для агентного моделювання натовпу. Helbling. Модель соціальної сили [3] вирішує фундаментальну ідею про те, як застосувати підхід, заснований на агентах, до симуляції натовпу. У цій моделі автори застосовують сили відштовхування та дотичні до кожного агента; тому рульовий агент може уникати перешкод та інших рухомих речовин. Однак основним недоліком цього методу є те, що кожен агент має однакові атрибути без індивідуальної ідентичності; Крім того, агенти переміщуються з однаковою швидкістю, на відміну від справжньої людської натовпу, яка має індивідуальні відмінності щодо мобільності. Крім того, коли метод переміщення Хелблінга застосовується до 3D-моделі, агенти мають тенденцію непередбачувано «трусити» та «вібрувати» із збільшенням щільності та застосуванням правила непріоритету. Загалом, модель Хельбінга представила фундаментальну концепцію дослідження симуляції агента та натовпу (ACS), і його дослідження надихнули нас надзвичайно.

А. Завін та ін. [7] запропонували інтелектуальну автоматизовану систему наведення пожежного виходу з використанням алгоритму пошуку A*, де поряд з наведенням постраждалих людей через найбезпечніший оптимальний шлях, система обчислює найменш скупчений і найкоротший шлях з урахуванням відстані, вузла під загрозою і механізму розподілу натовпу. Він також виявляє, чи якийсь шлях уже скомпрометований вогнем, і динамічно пропонує другий

оптимальний шлях до пожежного виходу. У запропонованій системі частина плану поверху задається і представлена односпрямованим зваженим графіком. Зважений графік представлений у вигляді сітки, де кожна клітинка сітки є квадратом вартості одиниці. У представленні сітки деякі осередки сітки розглядаються як протипожежні виходи, а деякі — стіни, тоді як інші — кімнати/передпокій. Результати експерименту показали, що запропонована система виконана точно та ефективно завдяки вирішенню проблем, пов'язаних із відстанню та безпекою, щоб забезпечити мінімальний термін служби та ресурсів.

Однак існують деякі обмеження щодо запропонованої системи, а не для проектування будь-якого прототипу; було представлено лише програмне моделювання шляхом створення зразка плану поверху будівлі. Деякі реальні фактори не враховуються під час виходу з пожежі, наприклад, система резервного електропостачання, паніка та миттєвий психологічний стан людей, які постраждали від пожежі, що також може вплинути на навколишнє середовище. Таким чином, як майбутнє розширення цієї статті, розроблено прототип на основі запропонованої системи, де враховуються та реалізуються реальні фактори. Для проектування такої системи наведення протипожежного виходу необхідні кілька протипожежних виходів, резервне джерело живлення, світлодіодна система облицювання підлоги для показу оптимальних шляхів тощо . Більше того, для впровадження системи для кожного поверху будівлі достатньо таких датчиків, як датчики диму та температури, датчики руху та інші засоби, підключені до центрального управління.

С. Вонг та ін. [8] запропонував ідею допомоги у швидкій евакуації. Вони представили алгоритм для розрахунку оптимального маршруту для кожної місцевості з ідеєю зменшити затори та максимізувати кількість евакуйованих, які прибувають до виходів за кожен проміжок часу. Їх система враховує розподіл натовпу, розташування виходів і ширину коридору при визначенні оптимальних маршрутів. Він також імітує рух натовпу під час оптимізації

маршруту. Як основа, вони очікують, що сусідні натовпи, які йдуть різними шляхами евакуації, повинні прибути до відповідних виходів майже в один і той же час. Якщо це не так, їхня система оновлює маршрути або шляхи повільних натовпів. Оскільки моделювання натовпу є нелінійним, оптимальний маршрут розраховується ітераційно. Система повторюється, поки не буде досягнуто оптимального стану. Система дозволяє розкласти структуру ситуації та визначає маршрути в ієрархічний спосіб щодо безпосереднього розрахунку оптимальних маршрутів для ситуації. Такий підхід не тільки зменшує обчислювальні витрати, але також дозволяє евакуювати натовпи людей у різних районах з різними пріоритетами. Ця система може працювати з різними структурами для ситуації та положень перешкод, щоб забезпечити оптимальний шлях. Вони використовували комбіновану стратегію, щоб отримати результати для різних ситуацій. Вони використовували різні алгоритми, такі як Djikstra, A* тощо, для різних середовищ, таких як тематичний парк, стадіон тощо. Враховуються різні параметри натовпу, такі як швидкість ходьби, радіус агента тощо. У цій системі, однак, бракує у тривимірній візуалізації та моделюванні людей з обмеженими фізичними можливостями або повільно рухаються.

Важливою мірою є екстрена евакуація з аеропортів, враховуючи триваючу подію стихійних і техногенних катаклізмів, що зачіпають ці місця. Ці інциденти зосередили увагу на потребах людей з обмеженими можливостями, які частіше постраждають у випадках надзвичайної ситуації. M. Manley та ін. [9] запропонував модель на основі агентів, яка може бути використана інженерами при плануванні та проектуванні для оцінки ефективності евакуації гетерогенного населення з аеропортів. Модель групує середовище за характеристиками доступності, включаючи різні умови, які окремо впливають на поведінку людей з обмеженими можливостями під час евакуації. Архітектура моделі складається з чотирьох логічних компонентів : (a) середовище, (b) населення, (c) візуалізація та (d) моделювання. Крім того, на

вибір напрямку агента впливають три основні сили: (a) притягання, (b) відштовхування і (c) тертя (див. рисунок 1.1).

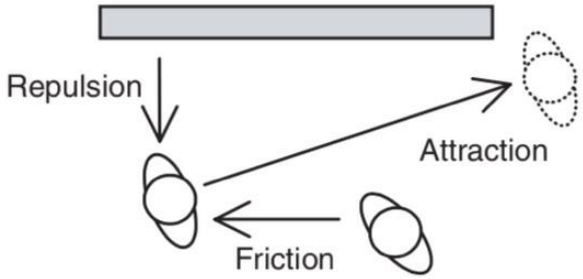


Рисунок 1.1: Основні сили для напрямку агента [39]

Метою імітаційного експерименту є оцінка впливу розміщення бомби на час евакуації осіб з та без інвалідності. Експеримент проводиться з використанням карти міжнародного аеропорту міжгірського західного регіону США. Аеропортовий комплекс являє собою двоповерхову конструкцію, що складається з трьох терміналів і п'яти залів за зразком терміналу пірсу. На основі наявності та розташування бомби визначено чотири сценарії моделювання. Сценарій 1 виконується без бомби для встановлення базової лінії для подальших порівнянь. Сценарії 2, 3 і 4 імітували виявлення бомби в одному з пристроїв системи виявлення вибухівки (EDS), знайдених у терміналах 1, 2 і 3 відповідно. Для кожного сценарію проводиться 50 моделювання.

Результати імітаційного експерименту з густонаселеним аеропортом продемонстрували поведінку, що відповідає очікуванням щодо людей з та без інвалідності. Крім того, було виявлено кілька висновків, які інформують як архітектурну інженерію, так і практику управління надзвичайними ситуаціями, включаючи (a) обмеження конструкції пристані аеропорту під час екстреної евакуації, (b) ідентифікацію осіб, які найбільше піддаються ризику або тих, хто має меншу витривалість, і (c) потенційні вузькі місця на сходах. Вони також не

розглядали кілька пожежних виходів або різні види агентів, таких як діти або люди з надмірною вагою.

Симуляція евакуації в зоні аеропорту потребує додаткових характеристик, крім більшості симуляцій, включаючи характерну поведінку відвідувачів, які вперше прийшли з неповним розумінням місцевості, і сімей, які не обов'язково підкоряються звичкам пішоходів, які часто вважаються. Симуляції евакуації, які не налаштовані для домену ai rport, не інтегрують важливі для нього фактори, що призводить до неточностей при застосуванні до нього. J. Tsai та ін. [7] запропонував ESCAPES - інструмент моделювання для евакуації кількох агентів, який включає чотири основні характеристики: (i) різні види агентів; (ii) емоційні взаємодії; (iii) інформаційні взаємодії; (iv) поведінкові взаємодії. Їхній симулятор відтворює явища, які спостерігалися під час евакуації, і функції, які вони інтегрують, значно впливають на час втечі. Вони використали ESCAPES для моделювання міжнародного терміналу міжнародного аеропорту Лос-Анджелеса (LAX) і отримали високу оцінку органів безпеки.

Вони обговорюють систему моделювання евакуації з кількома агентами, ESCAPES, у двох частинах: окремі типи агентів і взаємодії агентів. ESCAPES включають звичайних мандрівників, представників влади/безпеки та сімей, оскільки задокументовано, що вони мають найбільший вплив на евакуацію з аеропорту. Взаємодія агентів ESCAPES включає три окремих явища: поширення знань, емоційне зараження та соціальне порівняння. Система ESCAPES — це система, що складається з двох частин, що складається з 2D середовища OpenGL на основі проекту OpenSteer з відкритим вихідним кодом та компонента 3D візуалізації з використанням Massive Software. Сценарій розгортається в стандартному аеропорту, який складається з 2 воріт, 3 коридорів і 14 магазинів. Експерименти для сценарію включають наступне: 100 мандрівників, серед яких 10 сімей, 10 співробітників аеропорту. Ці сутності демонструють поширення знань і соціальне порівняння. ESCAPES був інструментом, який містить чотири ключові функції: (i) різні типи агентів; (ii)

емоційні взаємодії; (iii) інформаційні взаємодії; (iv) поведінкові взаємодії. Ці особливості ґрунтуються на соціальній психології та дослідженнях евакуації та пристосовані лише до потреб співробітника служби безпеки аеропорту.

ESCAPES не можна використовувати для кількох середовищ і великого натовпу.

Моделювання евакуації з будівлі дає дизайнерам ефективний спосіб перевірити безпеку будівлі перед будівництвом. У найрізноманітніших галузях (комп'ютерна графіка, робототехніка, динаміка евакуації тощо) розроблено значну кількість моделей. Н. Пеленчано та ін. [8] надають огляд моделей моделювання натовпу та вибраних комерційних програмних засобів для моделювання евакуації з висотних споруд. Вибрані бізнес-інструменти (STEPS і EXODUS) — це моделювання на основі сітки, яке забезпечує ефективне виконання, але вносить артефакти в результати. Вони зосереджуються на описі основних труднощів і обмежень цих інструментів, а також пояснюють важливість інтеграції психологічних і фізіологічних факторів людини в моделі.

Основна модель полягає в імітації загальної поведінки зграювання, яка складається з трьох простих правил, які описують, як окремий боїд маневрує на основі положень і швидкостей його сусідніх товаришів по зграї (див. рисунок 2):

Розділення : рульове управління для запобігання скупчення місцевих товаришів по стаді

Вирівнювання : орієнтація на середній курс місцевих товаришів по стаді • Згуртованість : орієнтація на середню позицію місцевих товаришів по стаді.

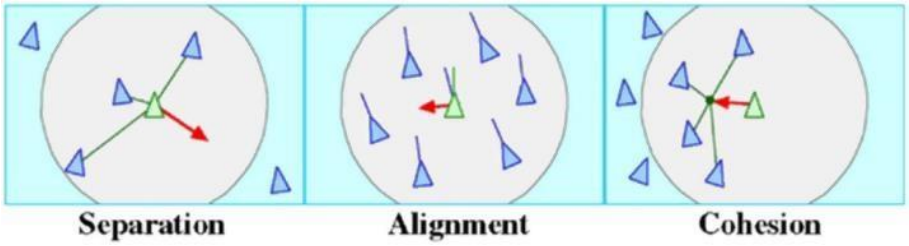


Рисунок 1.2 - Загальна поведінка згону [8]

Для дослідження обидві моделі програмного забезпечення (STEPS і EXODUS) були повністю протестовані та проаналізовані, щоб визначити їхні досягнення, а також їхні обмеження. Моделям, заснованим на клітинних автоматах, бракує реалістичності, оскільки вони обмежені конфігураціями в шаховому порядку, оскільки агенти можуть займати лише дискретні комірки в сітці. Агенти рухаються на основі деякого потенційного поля, розрахованого на сітці. Тому передбачається, що агенти рухаються в статичній середовищі, де шляхи до виходів не змінюються з часом. Агенти переміщуються з поточної комірки до бажаної комірки призначення залежно від зайнятості простору.

Стаття завершує огляд основ, які слід реалізувати для імітації руху людини. У цьому моделюванні швидкість потоку, щільність і швидкість людини не є заздалегідь визначеними значеннями, однак вони динамічно змінюються для обмеженого розміру сітки. Одним з основних недоліків є те, що система не врахувала фактори, що призводять до розподілу натовпу, що значно скорочує час втечі.

Оцінка евакуаційної спроможності станції метро в надзвичайних ситуаціях, таких як пожежі, а також покращення аварійного захисту системи залізничного транзиту є невід'ємною частиною управління транспортною експлуатацією залізничного транспорту. На основі аналізу впливу різних факторів на поведінку пасажирів при пожежній евакуації на станції метро було проведено дослідження X. Zhang et al. [4] моделює споруди станції метро, включаючи ворота, сходи, проходи, входи та виходи, а також моделює процес поширення вогню та сценарії евакуації пасажирів за моделлю стільникових автоматів. Наведено приклад, щоб проілюструвати вплив різних факторів на здатність евакуації та оцінити раціональність розташування станції.

Дослідження моделює 2D візуалізацію процесу евакуації пасажирів на станції метро, аналізує основні вузькі місця. Це також стимулює процес поширення вогню та сценарії евакуації пасажирів за моделлю стільникових автоматів. Нарешті, вони пропонують відповідні заходи щодо покращення. а) У

моделі використовується концепція значення пожежі, яка представляє ступінь небезпеки під час процесу моделювання евакуації вогню та в моделі вибору шляху пасажирів, b) Моделювання виявило, що модель клітинних автоматів має хорошу практичність для моделювання процесу евакуації. підземної будівлі двошарової конструкції. За допомогою комп'ютерного моделювання візуалізується весь процес евакуації евакуйованих, а також доступний розподіл евакуйованих у різний час, щоб можна було виявити вузькі місця на станції метро, які не сприяють евакуації. Результат моделювання може стати основою для реконструкції станційного обладнання та планування станції.

В останні кілька років агентне моделювання використовується для моделювання реальних бізнес-проблем. У цій роботі E. Bonabeau та ін. [4] надають основи ПРО та переваги його перед іншими методами моделювання. Переваги ПРО можна сформулювати у трьох наведених пунктах: (i) ПРО фіксує виникаючі явища, які є результатом взаємодії окремих сутностей; (ii) АВМ описує систему з природним описом, який робить модель ближчою до реальності; і (iii) ПРО є універсальним, оскільки до нього легко додати більше агентів. Однак очевидно, що іншими перевагами є здатність АВМ впоратися з новими подіями.

Використовуючи реальні додатки, розглядаються чотири його сфери застосування:

Моделювання потоку : здається, що ПРО ідеально підходить для надання цінного уявлення про механізми та передумови паніки та заклинювання через порушення координації. АВМ фіксує це явище природним чином.

Організаційне моделювання : можливо змоделювати в певному контексті або на певному рівні опису колективну поведінку організації або її частини. Принаймні, процес моделювання створює цінні якісні ідеї. Однак у деяких випадках також можуть бути отримані напівкількісні уявлення. Прекрасним прикладом цього є агентна модель операційного ризику.

Моделювання ринку : динаміка фондового ринку є результатом поведінки багатьох взаємодіючих агентів, що призводить до нових явищ, які найкраще зрозуміти за допомогою підходу знизу вгору-ABM. Щоб оцінити вплив зменшення розміру галочки, NASDAQ використала модель на основі агентів, яка моделює вплив регуляторних змін на фінансовий ринок за різних обставин. Модель дозволяє регулюючим органам перевіряти та прогнозувати результати різних стратегій, спостерігати за поведінкою агентів у відповідь на зміни та відстежувати прогрес, надаючи попередження про ненавмисні наслідки нововиконаних нормативних актів швидше, ніж у режимі реального часу, і без ризику ранніх тестів у реальному часі. ринок.

Симуляція дифузії : ABM застосовується до випадків, коли на людей впливає їхній соціальний контекст; тобто те, що роблять інші навколо них.

Вони також обговорили, як ПРО може принести значну користь при застосуванні до людських систем. Найкраще використовувати ПРО у таких непередбачених випадках:

Коли взаємодії агентів є складними, нелінійними, переривчастими або дискретними (наприклад, коли інші агенти можуть різко, навіть непостійно, змінити поведінку агента).

Коли простір необхідний, а позиції агентів не фіксовані. Приклад: тематичний парк, супермаркет, трафік, пожежна драбина.

Коли населення неоднорідне, де кожна людина (потенційно) різна.

Коли топологія взаємодії неоднорідна і складна. Приклад: коли взаємодії однорідні та змішуються по всьому світу, немає потреби в агентному моделюванні, але соціальні мережі рідко бувають однорідними, групованими, що призводить до відхилень від нормальної поведінки.

Коли агенти демонструють складну поведінку, включаючи навчання та адаптацію. Приклад: NASDAQ, провайдери.

ПРО не завжди ефективна на 100 відсотків і має певні недоліки. Модель загального призначення не може працювати, тому модель має бути побудована

на правильному рівні опису, з правильною кількістю деталей, щоб служити своїй меті. Іншою проблемою є сама характеристика систем, які моделюються з ПРО в соціальних науках: вони найчастіше вимагають людських агентів з потенційно нерозумною поведінкою, суб'єктивними рішеннями та складною психологією, тобто м'якими факторами, які важко кількісно оцінити, відкалібрувати та іноді виправданий.

Залишається остання практична проблема в ПРО, яку не слід ігнорувати. За визначенням, АВМ розглядає систему на рівні її складових одиниць, а не на загальному рівні. Хоча обчислювальна потужність продовжує зростати з вражаючою швидкістю, високі обчислювальні вимоги АВМ залишаються проблемою при моделюванні великих систем.

Система на основі агентів використовує обчислювальну модель автономного взаємодіючого агента в середовищі для оцінки поведінки групи, що розвивається. Ця стаття Н. Вагнара та ін. [4] представляють прототип комп'ютерної системи моделювання та підтримки прийняття рішень, яка використовує агентне моделювання для імітації евакуації натовпу під час пожежі та дозволяє тестувати різні сценарії катастроф. Система має на меті забезпечити можливість тестування кількох сценаріїв і підтримку прийняття рішень для фази планування та готовності управління надзвичайними ситуаціями щодо пожежі на концертних майданчиках. Система призначена для менеджерів з надзвичайних ситуацій, поліції та будь-яких адміністраторів, яким доручено планувати ліквідацію наслідків пожежі на концертних майданчиках. Користувачі системи можуть отримати користь від оцінки наслідків потенційних заходів безпеки.

Прототип є унікальним у сучасній літературі, оскільки він спеціально призначений для імітації налаштувань концертного майданчика, такого як стадіон або аудиторія, і надзвичайно налаштовується, щоб дозволити користувачеві визначати концертні майданчики з будь-яким розташуванням місць, маршрутів, етапів. , виходи та особи.

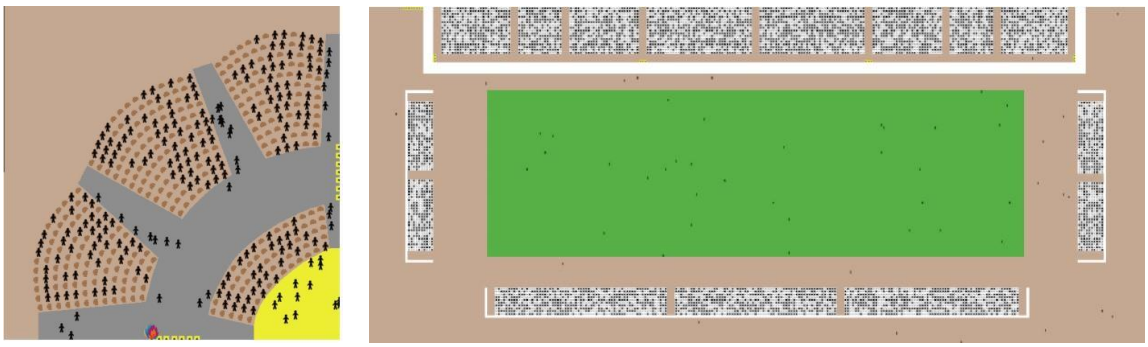


Рисунок 1.3 - Облаштування місця на концерті та стадіоні [4]

Система ABS, описана в цій статті, є системою підтвердження концепції. Як і в будь-якому прототипі proofof-concept, у систему можна внести кілька вдосконалень, і необхідно вирішити кілька питань, щоб перевести систему в життєздатну комерційну систему. Систему можна було б удосконалити, включивши більш складну модель процесу прийняття рішень особою в кожного агента. Ще одне покращення, яке можна було б зробити, - це динаміка вогню. Більш точні фізичні моделі пожежі можуть включати фактори, пов'язані з погодою, такі як температура, вологість, атмосферний тиск і вітер, а також топографічні фактори та фактори, пов'язані з паливом. Відповідні топографічні фактори включають розмір, форму та контур навколишнього середовища, тоді як паливні фактори включають тип матеріалів, присутніх у навколишньому середовищі.

Наразі система обмежує моделювання багаторівневих майданчиків, і завжди важко імітувати агентів для переміщення між рівнями місця проведення. Таким чином, додавання цієї можливості моделювання є ще одним важливим удосконаленням, яке можна зробити і дозволить моделювати більш складне середовище концертного майданчика. Крім того, хоча поточна система дозволяє специфікувати високо налаштовані середовища, це досягається за допомогою вхідного файлу, для створення якого потрібен мінімальний рівень технічних знань. Інтуїтивно зрозумілий графічний інтерфейс користувача має бути створений для специфікації середовища, що робить цю діяльність більш зручною для користувачів.

Інше дослідження Н. Пелачано та ін. [4] розглядає анімацію евакуації натовпу зі складних будівель, які можуть не розуміти зв'язку конструкції або випадково виявити заблоковані маршрути. Це вимагає врахування моделювання поведінки натовпу за двох обставин: агенти повідомляють про розуміння маршруту, а агенти виконують різні ролі, такі як кваліфікований персонал, лідери та прихильники. Булави, без централізованого контролера, є розподіленою мультиагентною системою. Кожен агент має свою поведінку на основі природних змінних особистості, що представляють реальні психологічні фактори. Рух агентів розраховується на двох рівнях. Високий рівень відповідає процесу пошуку послідовності кімнати, тоді як низький рівень — паралелі локального руху всередині кімнати. враховуються вхідні характеристики атмосфери, схожої на лабіринт, а саме розміри, кількість виходів, кількість небезпек і параметри, необхідні для моделювання, такі як кількість агентів, відсоток навчених агентів і відсоток лідерів. Щоразу, коли два або більше агентів зустрічаються в кімнаті, вони обмінюються двома частинами даних: розташуванням деяких ризиків, які блокують маршрути, і частинами будівлі, які інші агенти ретельно дослідили і не знайшли доступного виходу.

Ця оцінка показала значне покращення швидкості евакуації при використанні міжагентського зв’язку. Вони також спостерігали за поведінкою групування, яка виникає, коли натовп містить велику частку залежних агентів. Найкращі показники евакуації забезпечуються лише порівняно меншою часткою підготовлених керівників. Ці результати можна візуалізувати в режимі реального часу за допомогою 2D або 3D-переглядача.

Області, де є місце для вдосконалення, включаючи додавання індивідуалізму в модель Хельбінга, полягає в тому, що агенти будуть мати різні локальні рухи залежно від їхньої ролі. Високий рівень пошуку шляху має бути змінений, тому що люди мають менше шансів зайти в повну кімнату, коли є інші можливі шляхи.

Середовище будівлі та проведення евакуації є вирішальними факторами для ефективності евакуації з будівлі. Переходячи до використання моделей на основі агентів, багато сучасних досліджень розглядали різнорідну поведінку при евакуації, що виникає через індивідуальне сприйняття будівельних наборів, але деякі дослідження інтегрували свідомість евакуйованих щодо передбачуваного зміни просторової доступності через активовану пожежну безпеку. споруди під час надзвичайних ситуацій[4]. Метою цього дослідження було вивчити специфічний вплив такої просторової зміни на ефективність евакуації. L. Tan та співавт. [4] представили модель евакуації з будівлі на основі агентів, в якій враховуються знання евакуйованого, включаючи як просторове розуміння стабільного середовища під час типового сценарію, так і знання подій передбачуваної просторової зміни для цілей пожежогасіння. Крім того, при розгляді об’єктів протипожежної безпеки створюється семантичне уявлення середовища будівлі, щоб відобразити змінну структуру зв’язку. Використовуючи запропоновану модель, протягом трьох конкретних пожежних ситуацій було проведено серію моделювання евакуації для груп евакуйованих з різним рівнем знань.

Для моделювання індивідуального вибору маршруту під впливом потенційно зміненої структури зв'язку враховуються знання евакуйованих про просторове середовище, в тому числі знання стаціонарних просторових особливостей та обізнаність про спрацювані засоби пожежної безпеки. Кожен агент обирає свій шлях евакуації на основі передбачуваної просторової доступності. Коли виникає затор або маршрут блокується, агент оновлює свої знання та коригує маршрут евакуації, якщо є інший альтернативний маршрут. Щоб імітувати локальне переміщення евакуйованих у кожному внутрішньому просторі, середовище будівлі геометрично представлено у вигляді сітки, що складається з невеликих осередків. У запропонованій моделі середовище будівлі представлено семантичною моделлю на макрорівні, де взаємозв’язок між внутрішніми просторами, такими як кімнати та коридори, оновлюється

відповідно до стану об’єктів пожежної безпеки. На мікрорівні прийнято представлення на основі сітки для полегшення моделювання руху евакуйованих у внутрішньому просторі.

Важливо відзначити, що запропонована модель в основному зосереджена на впливі передбачуваної зміни просторової доступності і не включає всі фактори, які мають або можуть вплинути на просторову доступність середовища будівлі, такі як пожежний дим, токсичні гази, тощо

У цьому дослідженні вони могли врахувати реакцію евакуйованих при зіткненні з димом на додаток до їхньої здатності уникати маршруту, заблокованого спрацюваними засобами пожежної безпеки. Інше питання стосується розміру комірки представлення на основі сітки. У цьому дослідженні вони використовували 40*40 осередків з урахуванням загальної площі, яку займає пішохід, і обчислювальної ефективності. Розмір комірки може вплинути на результати моделювання. Необхідно дослідити чутливість результатів моделювання до розміру комірки. Крім того, при моделюванні руху евакуйованих на мікрорівні були внесені спрощення, а деякі фактори не були повністю враховані, наприклад, уявлення про тіло людини та ефект достатньої динамічної ширини виходу. Враховуючи ці факти, представлена тут модель є прототипом. Необхідно докласти більше зусиль, щоб перевірити модель і зробити її більш застосовною в майбутньому.

Підсумовуючи, існує велика кількість дослідницьких модулів на тему агентного моделювання натовпу, які завжди намагалися покращити існуючі програмні моделі та методи. Тим не менш, жодна з цих моделей програмного забезпечення не надає методу/процесу для включення будь-яких можливих фізичних змін у внутрішньому середовищі до того, як вони будуть фактично побудовані/побудовані або які можуть статися в майбутньому. Крім того, зовнішні файли не можна імпортувати в ці моделі програмного забезпечення, щоб додати нове 3D-середовище для моделювання.

2 АЛГОРИТМІЧНІ ТА МАТЕМАТИЧНІ РІШЕННЯ ДЛЯ ЗАДАЧ ДАНОГО КЛАСУ

2.1 Існуючі моделі програмного забезпечення

Ми описали 14 програмних моделей, які ми розглядаємо для порівняння. Кожна модель програмного забезпечення має переваги та недоліки. Раніше ми лише детально пояснювали всі ці моделі програмного забезпечення. Однак, щоб пояснити модель програмного забезпечення, яку ми розробляємо, нам також потрібно порівняти існуючі моделі програмного забезпечення. Ці моделі програмного забезпечення порівнюються на основі деяких параметрів/функцій. Потім ми впроваджуємо ці відсутні функції з кожної моделі програмного забезпечення в нашу модель програмного забезпечення.

На початку цього розділу ми порівнюємо 14 програмних моделей для моделювання аварійної евакуації, а пізніше обговорюємо наш метод розробки програмної моделі.

14 моделей програмного забезпечення можна розділити на 2 типи за методом їх доступності: доступні для громадськості та ті, які доступні на основі консультацій. У таблиці 1 і 2 моделі програмного забезпечення розділені жирною чорною лінією відповідно до способу їх доступності. Моделі, які більше не використовуються, не розглядаються для порівняння.

Основними характеристиками моделі є обов'язкові параметри, які можуть визначати роботу моделювання. Нижче наведено основні характеристики , які ми враховуємо для порівняння моделей програмного забезпечення:

- Доступність для громадськості
- Візуалізація моделі
- Мета
- Переміщення агентів
- Включення даних про пожежу
- Комп'ютерне проектування (CAD)

Кожна основна функція, описана тут, представляє стовпець у таблиці 1. ми порівнюємо 14 моделей програмного забезпечення на основі основних функцій.

Особливими функціями моделі програмного забезпечення є додаткові параметри, які модель може мати або не мати. Однак ці функції підвищують рівень сценаріїв або можливостей у конкретній моделі програмного забезпечення.

Нижче наведено особливості, які ми розглядаємо для порівняння моделей програмного забезпечення:

- Вихід заблоковано
- Токсичність агентів
- Непрацездатні/повільні агенти
- Вплив умов пожежі на агенти

Доступність

Хоча моделі програмного забезпечення, включені в це дослідження, з'явилися в рецензованих журналах, існують різні варіанти того, як ці моделі робляться загальнодоступними. Щоб диференціювати доступність програмних моделей, ми розділимо їх на дві групи. У першій групі моделі програмного забезпечення в деяких випадках доступні безкоштовно. Друга група включає моделі, які будуть використовуватися на основі консультацій розробником. Консалтингова фірма використовує моделі цієї групи, і користувачеві надаються лише результати. У таблиці 1 моделі програмного забезпечення, які є загальнодоступними безкоштовно, позначені цифрою «1», моделі, які доступні лише через платну компанію-розробник, позначені «2».

Візуалізація

Візуалізація визначає, чи дозволяє модель програмного забезпечення користувачеві візуалізувати вихід евакуації конструкції. Візуалізація евакуації дозволяє користувачеві побачити розташування вузьких місць і точок скупчення в кімнаті. Багато моделей забезпечують принаймні двовимірну візуалізацію, а останнім часом опубліковані версії або співпрацюють з іншими

віртуальними програмами для представлення 3-вимірних результатів. Вибір візуалізації часто залежить від аудиторії. У таблиці 1 моделі програмного забезпечення, які забезпечують двовимірну візуалізацію, позначені «2D», моделі програмного забезпечення, які забезпечують 3-вимірну візуалізацію, позначені як «3D», а моделі, які не мають жодних можливостей візуалізації, позначені «N». .

Мета

Призначення визначає використання моделі програмного забезпечення, оскільки воно стосується конкретних типів будівель, визначених розробниками моделі. Деякі моделі програмного забезпечення підкреслюють певний тип будівлі, а інші можна використовувати для багатьох типів конструкцій. Основна мета використання цього як функції полягає в тому, щоб дізнатися, чи призначена модель для моделювання проекту будівлі, обраного користувачем. Моделі програмного забезпечення, описані в таблиці 1, включають моделі, які можуть імітувати будь-яку будівлю, позначені цифрою «1», моделі для житлових будинків позначені «2», моделі для державних транспортних станцій позначені «3», а моделі які імітують лише 1-маршрут/виїзд з будівлі, позначені «4».

Рух Агентів

Переміщення агентів стосується того, як моделі програмного забезпечення переміщують мешканців по всій будівлі. Для більшості моделей користувач або програма моделювання зазвичай призначають особливу безперешкодну (низьку щільність) швидкість для пасажирів. Більш значні відмінності в моделях виникають, коли мешканці підходять ближче, що призводить до черг і заторів всередині будівлі. Нижче наведено різні методи, за допомогою яких моделі відображають рух пасажирів і обмежений потік по всій конструкції:

Кореляція щільності (D): модель призначає швидкість і потік залежно від щільності простору для людей або населення.

Вибір користувача (UC): користувач призначає значення швидкості, потоку та щільності для певних полів будівництва.

Відстань між людьми (ID): «бульбашка» на 360 ° оточує кожну людину, що вимагає мінімальної відстані від інших мешканців, бар'єрів та елементів конструкції (стін, кутів, поручнів тощо).

Порожня наступної комірки сітки (E): мешканець не мігрує в комірку сітки, яку вже займає інший мешканець. Тому мешканець чекає, поки наступна камера спорожніє. Якщо на подібну комірку чекає більше одного мешканця, тоді модель виправляє суперечки, які можуть виникнути, вирішуючи, хто з мешканців рухається першим.

Умовний (C): переміщення по всій будівлі залежить від умов навколишнього середовища, конструкції, інших евакуйованих та/або стану пожежі з умовними моделями. Не так багато уваги приділяється перевантаженню всередині приміщення лише для цього позначення.

Клітинні автомати (CA): у цій моделі мешканці переміщуються через імітований кидок зваженої кубика з кімнати клітини/сітки в іншу камеру. При цьому всі агенти поводяться як єдине ціле, наприклад, якщо один агент помирає, інші гинуть через ті ж ефекти, як пожежа.

У таблиці 1 різні методи переміщення агента позначені таким чином: кореляція щільності як «D», вибір користувача як «UC», відстань між людьми як «ID», порожнеча наступної клітинки сітки як «E», умовна як « C», а клітинні автомати як «CA».

Дані про пожежу

Дані про пожежу описують, чи дозволяє модель інтегрувати вплив пожежі в моделюванні евакуації. Модель може інтегрувати інформацію про пожежу такими способами: імпортувати дані про пожежу/результати з іншої моделі, дозволяючи користувачеві вводити дані про пожежу під час евакуації в певний час, або модель може мати свою модель пожежі, яка працює одночасно з моделлю евакуації. Якщо модель не може інтегрувати дані про пожежу, вона

запускає всі симуляції в режимі без пожежі. Метою моделей евакуації, які включають таку інформацію, є оцінка безпеки мешканців, які переміщуються через погіршені обставини, і побачити, як їх поведінку можна змінити. У таблиці 1 моделі програмного забезпечення, які можуть імпортувати дані про пожежу, позначені як «Y1», моделі програмного забезпечення, які мають пожежний елемент, що працює одночасно з евакуацією, позначені як «Y2», а модель, яка запускає всі симуляції в непожежному режимі. режим позначено як «N».

Креслення автоматизованого проектування (САПР).

Функція CAD визначає, чи дозволяє модель користувачеві імпортувати документи в модель із програми автоматизованого проектування (CAD) або інших файлів, що містять макет конструкції. У багатьох випадках імпорт файлів CAD економить час і є більш точним. Якщо користувач може покладатися на точні креслення САПР замість того, щоб вручну закладати будівлю, буде менше місця для помилки введення будівлі. Однак користувачеві потрібно бути більш обережним щодо правильного підключення всіх окремих компонентів . Якщо модель має функцію імпорту моделі CAD, для цих конкретних моделей використовується мітка (Y) у таблиці 1 . З іншого боку, коли модель не має такої потужності, використовується мітка (N).

Нижче ми згадали кожну етикетку, яка використовується в таблиці 2.1.

Доступність:

- 1: безкоштовно
- 2: платна візуалізація:
- 2D : 2-D візуалізація
- 3D : 3-D візуалізація
- N : Без візуалізації Мета:
- 1 : Змоделюйте будь-який тип будівлі
- 2 : спеціалізується лише на резиденціях
- 3 : Спеціалізується на станціях громадського транспорту

• 4 : імітувати лише будівлю з 1 маршрутом/виїздом

D : Щільність

UC : вибір користувача

ID : Відстань між людьми

E : Порожня наступна клітинка сітки

C : Умовно

CA: дані про пожежу стільникових автоматів:

N : Модель не може включати дані про пожежу

Y1 : модель може імпортувати дані про пожежу

Y2 : Модель має власну САПР елемента одночасного вогню:

N : не дозволяє імпортувати файли CAD

Y : дозволяє імпортувати файли CAD

Таблиця 2.1: Порівняння основних ознак

PathFinder	2	2D	1	д	Н	Ю
БЕЗПЕЧНО	2	2D	4	E	Y1	Н
CRISP	2	2D, 3D	1	E,D	Y2	Ю
EGRESS 2002	2	2D	1	D(CA)	Y2	Н
SGEM	2	2D	4	д	Н	Ю
ВИХІД89	2	Н	3	д	Y1	Н

Таблиця 2.1 включена таким чином, щоб описати більш конкретні можливості кожної моделі. Ця таблиця включена для користувачів, які хочуть змоделювати конкретні ситуації евакуації та/або для користувачів, щоб зрозуміти варіації у складності моделі програмного забезпечення. Можна помітити, що з підвищенням рівня витонченості зростає кількість відмінних характеристик , що моделюються моделлю.

Під час евакуації, коли евакуйовані мешканці будівлі рухаються в одному напрямку, рятувальникам або іншим особам може знадобитися рухатися в протилежному напрямку. Це може призвести до зменшення ширини для

мешканців конструкції. Зворотний потік визначає ті моделі, які можуть імітувати зустрічний потік. Зустріч – це паралельне переміщення людей у різних напрямках, яке можна використовувати, якщо бажана імітація включає в себе пожежників або інших осіб, які мають виконувати обов'язки під час евакуації. Якщо в кінозалі є персонал, який надає допомогу агентам, до допоміжного персоналу застосовується механізм.

Для багатьох цілей виїзні ворота можна заблокувати. Блок виходу визначає моделі, які дозволяють користувачеві блокувати виходи, які будуть використовуватися імітованими мешканцями. Користувача програмної моделі можуть зацікавити сценарії, де можна використовувати не всі шляхи. Блок виходу визначає ті моделі, які дозволяють користувачеві блокувати вихід із використання імітованими мешканцями в сценарії які мають таку здатність або дозволяють користувачам вручну заблокувати використання виїзних воріт до або навіть під час ситуації, або модель блокує вихід із використання через певні обставини (наприклад, падіння сміття з даху може заблокувати вихід) . Користувачі повинні знати, як мешканці бачать заблокований вихід і чи можна заблокувати вихід у певний час чи лише на початку моделювання.

Під час пожежі в будівлі цілком імовірно, що умови навколишнього середовища, викликані пожежею, можуть вплинути на мешканців, що знаходяться поблизу. Деякі моделі програмного забезпечення, які включають обставини пожежі, мають можливість змінювати поведінку пасажирів залежно від цих обставин пожежі, включаючи такі змодельовані ситуації: пасажири вибирають інший вихід через велику пожежу та можуть постраждати від пожежі, пасажири повертаються назад і відвернувшись від вогню. Умови пожежі визначають, чи здатна модель евакуації змінити поведінку мешканців через обставини пожежі.

При горінні матеріалу утворюються отруйні гази (наркотичні та дратівливі гази), аерозолі для куріння та тепло. Пожежі можуть поширювати дим і теплі гази по всій будівлі, представляючи мешканців токсичними газами, які можуть

викликати симптоми невеликого головного болю, що потенційно призведе до зниження швидкості руху, інвалідності (тобто, коли мешканці перестають рухатися без допомоги) або навіть смерті. Токсичність визначає ті моделі, які імітують інвалідність або смерть пасажирів через токсичні продукти від пожежі, що містяться в димі.

Для всіх цих функцій модель програмного забезпечення позначається як має таку можливість або не має такої можливості. Якщо модель має таку можливість, «Y» для «так» поміщається в таблицю 2. Якщо ні, «N» для «ні» поміщається в таблицю 2.

Таблиця 2.2 - Порівняння особливостей

програмне забезпечення Модель	Протитечний	Блок виходу	Токсичність	Фізично кинутий виклик/ Повільний рух	Умови пожежі
ВИХІД	Н	Н	Н	Н	Н
ПЕДРУТ	Н	Н	Н	Ю	Н
Simulex	Ю	Ю	Н	Н	Н
FDS+Evac	Ю	Ю	Ю	Н	Н
SimWalk	Ю	Н	Н	Ю	Н
Будівля ВИХОД	Ю	Ю	Ю	Н	Ю
легіон	Н	Ю	Ю	Ю	Ю
MassMotion	Ю	Ю	Н	Ю	Ю
PathFinder	Н	Н	Н	Н	Н
БЕЗПЕЧНО	Н	Н	Н	Н	Ю
CRISP	Ю	Ю	Н	Ю	Н
EGRESS 2002	Ю	Ю	Ю	Ю	Ю
SGEM	Ю	Ю	Н	Ю	Н
ВИХІД89	Ю	Ю	Ю	Ю	Ю

Слід зазначити, що ці моделі можуть працювати лише з набором параметрів, які попередньо визначені всередині програмного пакета. Ці моделі програмного забезпечення не можуть імпортувати файли для додавання нового 3D-середовища до свого програмного пакета. Тому, якщо в майбутньому з'являться будь-які можливі фізичні зміни та нова функція/параметр у технології, вони не можуть бути реалізовані в програмній моделі. Однак наш підхід може працювати з будь-якою функцією/параметром або будь-якими можливими фізичними змінами, які можна легко імпортувати та реалізувати з незначною конфігураційною обробкою, яку ми обговорюємо. Ми також пропонуємо набір фізичних змін та різні інші функції, щоб наша модель працювала в будь-якому сценарії, щоб подолати недоліки цих моделей програмного забезпечення.

2.2 Запропонована методика

Наша дипломна робота включає розробку базової програмної моделі систем евакуації для внутрішнього середовища, яка може включати будь-які можливі фізичні зміни до того, як вони будуть фактично побудовані/будовані. Наша програмна модель допомагає оцінити фізичні зміни, щоб перевірити їх ефективність у разі надзвичайної ситуації для різних сценаріїв на основі інцидентів, що трапилися в минулому. Крім того, ми можемо реалізувати будь-які можливі фізичні зміни в 3D внутрішньому середовищі, які можуть статися в майбутньому, використовуючи метод моделювання натовпу на основі агентів. Однак комплексний робочий процес може допомогти краще зрозуміти ідею нашої дипломної роботи.

Рисунок 2.1 ілюструє повний робочий процес нашої моделі програмного забезпечення. Кожне поле в цьому робочому процесі дає широке розуміння розвитку нашої моделі програмного забезпечення, а також впровадження деяких можливих фізичних змін, які ми пропонуємо.

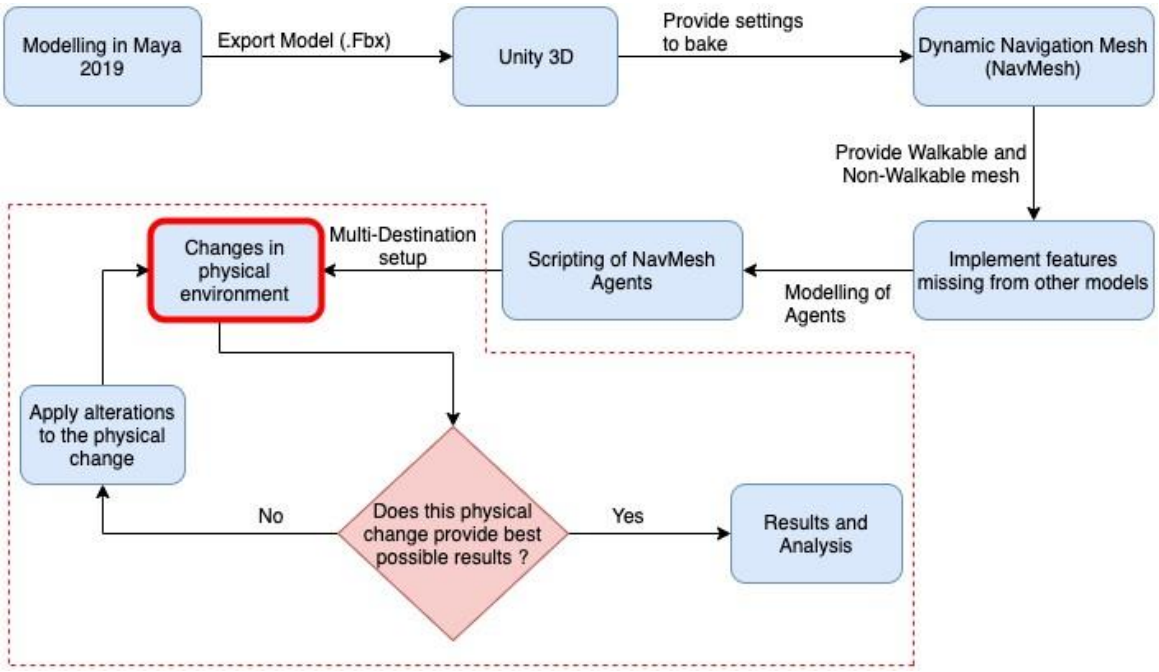


Рисунок 2.1 - Блок-схема методології

Блок-схема на малюнку зображує різні методи та програмне забезпечення для розробки всієї моделі програмного забезпечення евакуації. Також виділена частина зображує нашу головну ідею включення будь-яких можливих фізичних змін, які можуть допомогти покращити терміни евакуації та ефективність. Ми детально обговоримо кожен компонент блок-схеми в наступних підрозділах. Крім того, щоб зрозуміти, як працює наша модель програмного забезпечення для моделювання, ми отримуємо уявлення з узагальненого псевдокоду для моделювання будь-яких фізичних змін, які можуть відбутися в майбутньому на рисунку 2.2.

Запуск Maya 2014 приніс кілька потужних інструментів моделювання та набір інструментів моделювання, який починає вирівнювати ігрове поле, з деякими чудовими інструментами перетопології та швидшим робочим процесом. Відомо , що Maya з величезною бібліотекою інструментів для анімації є набагато більш потужною програмою, коли справа доходить до анімації. Запатентована мова сценаріїв Maya MEL надзвичайно налаштована, і це одна з причин, чому Maya може бути багато дисків у кіноіндустрії, тому що

студії люблять мати можливість створювати власний набір характеристик, які можуть допомогти їм у виконанні своєї роботи набагато легше.

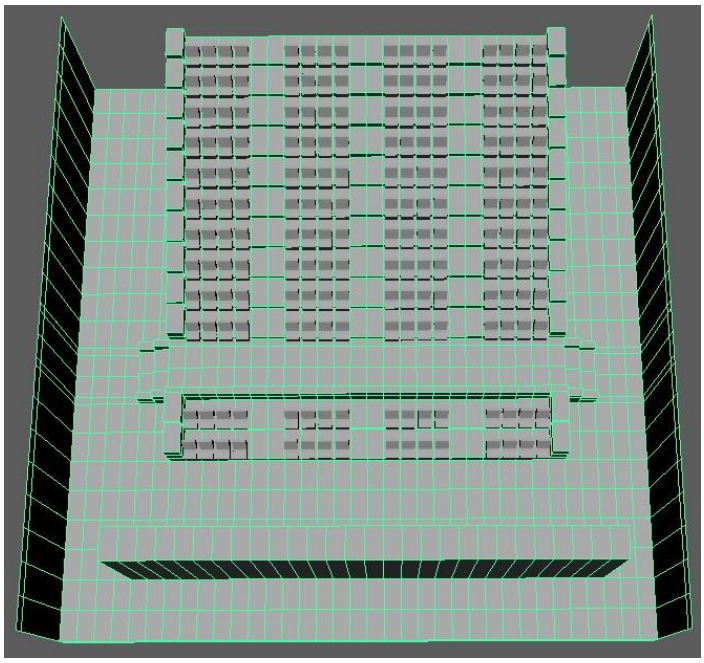


Рисунок 2.3 - Модель кінозалу в Autodesk Maya

• Експорт файлів з Maya в Unity 3D

Однією з інших причин, чому ми вибираємо Maya, є те, що вона надає широкий спектр опцій під час експорту нашої 3D-моделі в будь-яке інше програмне забезпечення. Оскільки ми збиралися використовувати Unity 3D після моделювання, ми експортували наш файл у форматі .Fbx для кращої стабільності 3D-моделі. Файл експортовано без жодних збоїв і з неушкодженими всіма дрібними деталями. Крім того, цей процес експорту файлу .Fbx виступає в якості вхідних даних для нашої моделі програмного забезпечення, описаної псевдокодом на малюнку.

Моделювання в Unity

Наступним компонентом після експорту файлу з Autodesk Maya 2019 є Unity 3D. Решта компоненти розроблені та реалізовані в Unity 3D після успішного імпортування файлу .Fbx.

Як працює Unity?

Unity розроблений Unity Technologies і є кросплатформним ігровим движком. Движок можна використовувати для створення симуляцій, тривимірних, двовимірних, віртуальної реальності та ігор розширеної реальності. Він також використовується в секторах, які не є відеоіграми, таких як кіно, автомобільна промисловість, інженерія, архітектура та будівництво. Unity дозволяє користувачам створювати як 2D, так і 3D ігри та досвід, а двигун надає основний API сценаріїв C# для редактора Unity у вигляді плагінів та ігор, а також функції перетягування. Навігаційна система в Unity дозволяє нам дуже легко надавати потужний пошук шляху персонажам/агентам в іграх або симуляторах. Система забезпечує інтелектуальну поверхню, яка називається навігаційною сіткою (Navmesh), для пересування агентів.

Поверхня, по якій переміщуються агенти , зберігається NavMesh у вигляді опуклих багатокутників . Опуклі багатокутники корисні, оскільки ми розуміємо, що в багатокутнику між будь-якими двома точками немає перешкод. По відношенню до меж полігонів ми зберігаємо дані про сусідні полігони. Це дає нам можливість обміркувати весь пішохідний регіон.

• Агент Navmesh

Агент Navmesh — це компонент, який бере деякі точки шляху та здійснює патрулювання агента AI на карті. Тому, щоб перемістити агента, нам потрібно вказати агенту navmesh перемістити агента до місця призначення. Щоб досягти цього, ми можемо вказати агенту розпочати обчислення простого шляху, використовуючи агент NavMesh і властивість призначення з точкою, яку ми хочемо перемістити агентом AI. Після завершення обчислення агент AI починає рухатися по шляху, поки не досягне місця призначення.

• Анімація та жорстке тіло (колайдер)

Після розгортання моделей Maya в Unity3D ми маємо намір розгорнути агентів (персонажів). Для переміщення агента в симуляції ми використовували такі методи, визначені unity:

Анімаційні рухи о Використання Rigidbody

Rigidbody – це компонент, який використовується для контролю положення об’єкта за допомогою фізичного моделювання. Додавання компонента Rigidbody до об’єкта передає його рух під контроль фізичного двигуна Unity. Функція гравітації не потрібна; оскільки об’єкти Rigidbody тягнуться вниз під дією сили тяжіння, яку забезпечує фізичний двигун Unity. Щоб перемістити об’єкт Rigidbody в реальному просторі, розробнику необхідно застосувати властивість add force до цього об’єкта. Силу можна застосовувати з будь-якого місця, наприклад вгору, вниз, ліворуч і праворуч. Сила прикладається безперервно вздовж напрямку вектора. Тверді тіла дозволяють агентам в імітованій моделі діяти під контролем фізики. Rigidbody може отримувати сили та крутний момент, щоб змусити агентів рухатися реалістично. Будь-який агент повинен містити Rigidbody, щоб на нього впливала гравітація, діяти під додатковими силами за допомогою сценаріїв або взаємодіяти з іншими агентами в натовпі через фізичний механізм NVIDIA PhysX [5].

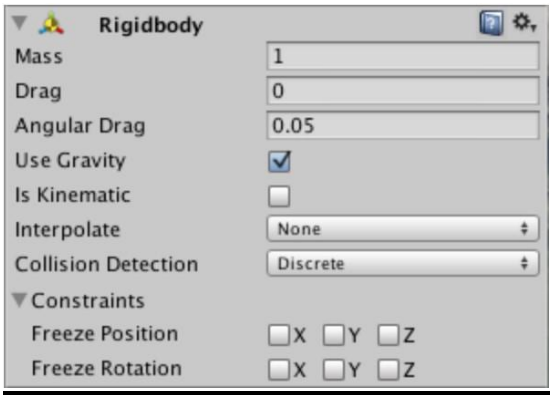


Рисунок 2.4 - Значення компонентів твердого тіла[5]

Тверді тіла повинні бути явно додані до наших агентів, перш ніж на них вплине фізичний двигун. Ми можемо додати Rigidbody до наших вибраних агентів у меню Компоненти>Фізика-> . Тепер наш агент готовий до фізики; він впаде під силу тяжіння і може отримувати сили за допомогою сценаріїв, але нам може знадобитися додати коллайдер або Joint, щоб змусити його вести себе саме так, як ми хочемо.

Коллайдери — це ще один тип елементів, які додаються разом із Rigidbody, щоб дозволити відбуватися зіткнення. Коли два Rigidbody врізаються один в одного, фізичний механізм не визначає зіткнення, якщо Коллайдер не підключений до обох агентів. Тверді тіла без колайдерів будуть просто проходити одне крізь одне під час фізичного моделювання [5].

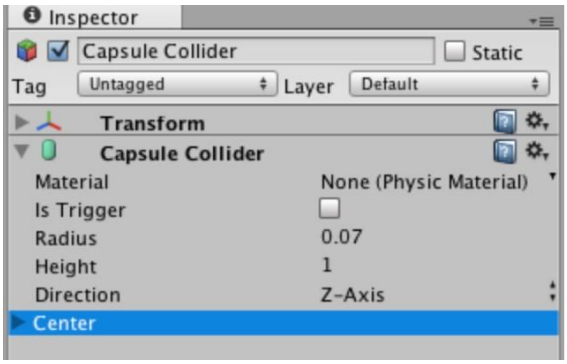


Рисунок 2.5 - Значення капсульного колайдера

Algorithm 1: Emergency Evacuation Software Model

Input : .fbx file from Maya

Output: simulated model

define *dynamicNavMesh*; using unity 3D
if *number*(*agents*) > 1 **then**
    define NavMesh(Average size of the agents, Average radius of the agents);
    create walkable blue region && create non-walkable red region;
    return *NavMesh*
else
    Error
end
define variable *thresholdCasualty*;
define variable *thresholdEvacuationtime* ;
define list of *physicalChanges*;
while *(numberCasualty* ≤ *thresholdCasualty) &&*
(evacuationtime ≤ *thresholdEvacuationtime)* **do**
    apply alteration to physical changes;
    define new physical changes ;
    add new physical changes to the list;
end
upload results and analysis
return *results*

Рисунок 2.6 - Псевдокод для моделювання фізичних змін

Таким чином, для моделювання будь-якої фізичної зміни вхідним має бути файл .fbx, який містить будь-яке 3D середовище в приміщенні. Крім того, на початку ми повинні визначити параметри, які створюють навігаційну сітку, поріг постраждалих і час евакуації для досягнення очікуваних результатів від нашої програмної моделі. У наступному підрозділі ми пояснюємо кожен компонент робочого процесу та псевдокод.

Моделювання в Autodesk Maya

Першим компонентом, який потрібно розпочати, є моделювання в Autodesk Maya 2019

Maya — це програма, яка використовується для створення 3D-об'єктів для використання в кіно, телебаченні, розробці ігор та архітектурі, які відображаються в режимі реального часу. Maya надає багато характеристик, таких як атрибути тому, обробка шляху до файлу, поверхня ефектів фарбування, відповідність кліпу, вбудована довідка, підтримка URI, прив'язка тощо.

Як створити 3D-моделі за допомогою основних інструментів Maya?

Основний дисплей Maya складається з різних вкладок для моделювання різних типів сцен. Ці вкладки включають криві/поверхні, полігони, такелаж, анімацію тощо. Ми використовували багатокутники та криві, щоб створити моделі для кінозалу з нуля. Багатокутники складаються з геометрії на основі вершин, ребер і граней, які можна використовувати для створення тривимірних моделей у Maya.

У Unity 3D доступні різні типи коллайдерів, наприклад Box Collider, Sphere Collider, Capsule Collider, Mesh Collider, Wheel Collider і Terrain Collider. Існують різні специфічні значення, пов'язані з кожним з них, як показано на малюнку 8 для капсульного колайдера [5].

Анімаційні рухи

Щоб використовувати більше однієї анімації, розробнику необхідно використовувати компонент Animator. Аніматор насправді є кінцевою

машиною. У компоненті Animator анімації потрібно перетягувати та використовувати переходи для переходу з одного стану в інший. Для переходу з одного стану в інший використовуються умови переходу. В умовах переходу нам потрібно зробити деякі параметри та додати їх до умови переходу. Можна встановити різні типи параметрів для використання таких умов переходу, як Float, Boolean, Integer і тригер. Однак ми не використовували Rigidbody у нашій симуляції для переміщення гравця, оскільки ми використовували рух анімації для переміщення гравця по реальному простору. Тригер і bool насправді такі ж, як коли ми встановлюємо тригер, його значення змінюються з false на true, а коли завдання виконано, його значення повертається до false. Існує інший тип аніматора для керування анімацією, який був представлений в Unity5 під назвою Blend Tree. Для виконання складних анімацій використовується Blend Tree, яке складається з кількох невеликих станів.

Для деяких ситуацій, в основному створюючи певні специфічні ефекти, необхідно перемикати керування агентом між анімацією та фізикою. Для цього можна позначити жорсткі тіла є кінематичним . Хоча Rigidbody позначено як кінематичний, на нього не впливають зіткнення, сили чи будь-яка інша частина фізичної системи. Це означає, що ми повинні керувати агентом, безпосередньо маніпулюючи компонентом Transform. Кінематичні тверді тіла впливатимуть на інших агентів, але фізика не вплине на них самих. Наприклад, суглоби, які прикріплені до кінематичних агентів, будуть обмежувати будь-які інші тверді тіла, приєднані до них, а кінематичні тверді тіла впливатимуть на інші тверді тіла через зіткнення [5].

Щоб пояснити фізику, візьмемо приклад Землі. У нашому повсякденному житті, коли ми підкидаємо щось у повітря, ця річ має тенденцію повертатися на землю через гравітацію. Unity3D надає деякі корисні компоненти для досягнення подібної поведінки. Щоб досягти гравітації, Unity3D допомагає нам, надаючи компонент Rigidbody, який можна використовувати для виявлення гравітації, а також розробник може змінювати значення гравітації.

Однак його значення за замовчуванням встановлено на рівні 9,8 м/с. Залишається постійним питанням, як ми можемо знати, зіткнеться щось із поверхнею чи ні? Для виявлення зіткнення Unity3D надає компонент, який називається коллайдером. Існують різні види коллайдерів. Якщо об'єктом моделювання є сфера або капсула, то капсульний коллайдер краще, а якщо сітка занадто складна, то кращий вибір.

Щоб мати переконливу фізичну поведінку, агент в імітаційній моделі повинен правильно прискорюватися і на нього впливають гравітація, зіткнення та інші сили. Інтегровані фізичні механізми Unity надають нам компоненти, які керують фізичним моделюванням. За допомогою лише кількох налаштувань параметрів ми можемо створити агентів, які ведуть себе пасивно реалістично (тобто вони будуть переміщені зіткненнями та падіннями, але не почнуть рухатися самі по собі). Насправді в Unity є два різних фізичних механізму: один для фізики 2D, а інший для 3D фізики. Основні концепції двох движків схожі (за винятком додаткового виміру в 3D), але вони реалізовані за допомогою різних компонентів. Наприклад, існує компонент Rigidbody для 3D фізики та подібний Rigidbody 2D компонент для 2D фізики [5].

Unity — один з найкращих ігрових движків для розробки ігор та симуляторів. Основними перевагами, які пояснюють, чому Unity є потужним, є низька крива навчання для початківців, активна екосистема розробників активів і плагінів, швидка швидкість розробки та його кросплатформна інтеграція, яка підтримує 25 платформ. Unity Prefabs, які є попередньо налаштованими ігровими об'єктами, забезпечують корисні та гнучкі робочі процеси, які дозволяють вам функціонувати впевнено, не турбуючись про помилки, що забирають багато часу. Він також забезпечує написання сценаріїв на C#, який є однією з найпопулярніших і легко зрозумілих мов для початку.

Динамічна навігаційна сітка

Планування маршруту та навігація відіграють значну роль у імітованих віртуальних середовищах та комп'ютерних іграх. Однак, розробляючи

надзвичайну ситуацію, нам потрібне динамічне середовище, в якому в будь-який момент часу можуть впасти або виникнути перешкоди. Щоб це розібратися, нам потрібна динамічна навігаційна сітка, по якій агенти можуть рухатися і знаходити шлях до найближчого виходу. Крім того, цей процес визначення динамічної навігаційної сітки описаний на кроці 1 псевдокоду на малюнку .

• Як unity забезпечує прохідну та непрохідну сітку?

Є дві проблеми, які потрібно вирішити, якщо ми хочемо розумно перемістити персонажів у грі. Ці проблеми: по-перше, як виявити ціль, а по-друге, як туди рухатися. Ці дві проблеми тісно пов'язані, але за своєю природою досить різні. Проблема виявлення цілі є більш глобальною та статичною, оскільки вона розглядає всю сцену. Проте рухатися до мети є більш локальним і яскравим. Кожен агент дивиться лише на напрямок руху та як уникнути зіткнень з іншими рухомими агентами.

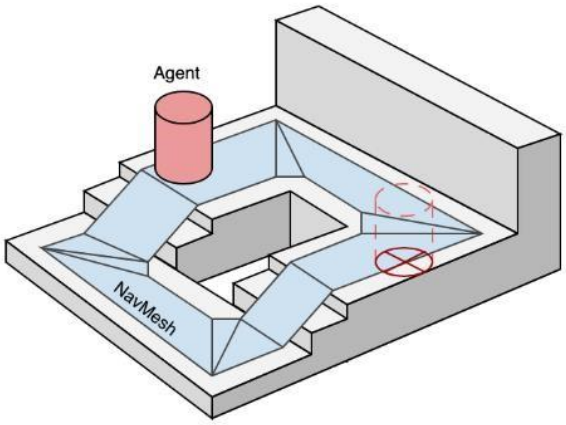


Рисунок 2.7 - Пішохідні зони [5]

Навігаційна система створюється автоматично на основі геометрії сцени, щоб представляти зони, де можна прогулятися (де агент може стояти і рухатися). Потім місця з’єднуються з верхньою поверхнею, яка називається навігаційною сіткою (Скороче NavMesh). Визначаючи навігаційну сітку, щоб забезпечити доступну та непрохідну область, ми повинні надати параметри за

замовчуванням, такі як радіус агента, висота агента, швидкість ходьби тощо. Цей процес надання параметрів за замовчуванням описаний у кроках 2-10 псевдокоду.

Функції реалізації, відсутні в інших моделях

Ми взяли до уваги різні особливості для порівняння існуючих моделей програмного забезпечення в Таблицях 1 і Таблицях 2. Далі ми обговорюємо всі функції, яких не вистачає в деяких із цих моделей програмного забезпечення, і детально реалізуємо їх у нашій моделі програмного забезпечення.

Розподіл натовпу: під час напружених ситуацій, таких як екстрена евакуація, кожен натовп використовує поведінку пасти овець. Тому ми впроваджуємо розподіл натовпу, щоб зменшити жертви та покращити час евакуації. Для реалізації цієї функції ми використали функцію cast cast. В одному кадрі агент кидає промінь, щоб перевірити, скільки інших агентів йде до цих дверей і чи є шлях до цих дверей чи ні. Типовий запис функції raycast для кожного виходу встановлюється на значення, що дорівнює 10% населення для цього сценарію. Якщо більше 10% людей йдуть до єдиного виходу, наступна людина йде іншим найближчим виходом.

Пожежа: як ми обговорювали функцію даних, ми можемо реалізувати компонент пожежі, імпортуючи його або впровадивши в нашу програмну модель. Ми вирішили реалізувати його в unity 3D одночасно, щоб отримати більш точні результати. У цьому сценарії ми також можемо перевірити, скільки людей загинуло від пожежі (Пожежа також впливає на поведінку агента)

Блок виходу: ми використовували випадкові бетонні блоки для падіння зверху або будь-який інший ефект, щоб блокувати ворота в певних ситуаціях. Таким чином, змушуючи агентів зробити інший вихід, який не заблокований.

Токсичність: Токсичність дає нам огляд сценарію під час удушення агентів через дим або будь-який біологічно розкладний невидимий газ. У цьому випадку ми можемо встановити час, коли агенти починають задихатися тощо.

Якщо вони все ще залишаються всередині після часу початку задушення, це призводить до смерті від задишки.

фізичними труднощами/повільно рухається : це явище було одним із найважливіших явищ, які необхідно застосувати під час евакуації, оскільки над ним працює найменша кількість програмних моделей. До того ж ми використовували інвалідний візок, в якому ми забезпечили доступні для інвалідних візків двері, і пріоритет надається інвалідним візкам порівняно зі звичайними біговими агентами.

Написання сценарію Navmesh Agent

Система завдань Unity C# дозволяє користувачам писати багатопотоковий код, який добре взаємодіє з рештою Unity і спрощує написання правильного коду. Багатопотокове написання коду може забезпечити переваги високої продуктивності. Вони включають значне збільшення частоти кадрів. Пошук агента

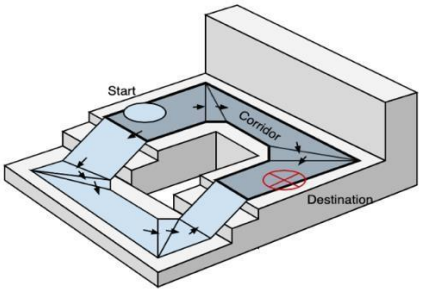


Рисунок 2.8 - Пошук шляху агента

Спочатку нам потрібно нанести на карту місця початку та призначення з найближчими полігонами, щоб знайти маршрут між двома місцями сцени. Потім ми починаємо дивитися з початкової точки, відвідуючи всіх сусідів, поки не досягнемо багатокутника цілі. Відстеження відвіданих полігонів дає змогу виявити послідовність полігонів, яка веде до початкового розташування (Рисунок 7). Алгоритмом, який використовує unity для пошуку маршруту є A*.

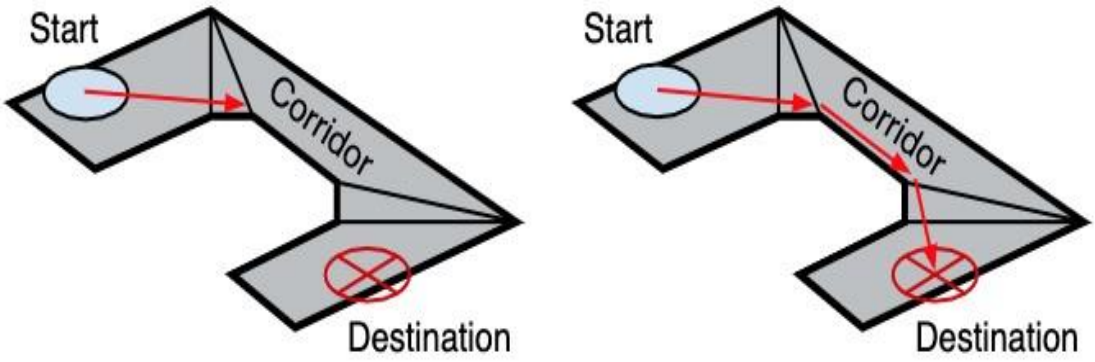


Рисунок 2.9 - Відстеження пункту призначення

Послідовність полігонів, яка описує маршрут від початку до полігону цілі, називається коридором. Надаючи вказівки до наступного видимого кута коридору, агент досягає цілі. Якщо у вас є простий збіг, коли на сцені рухається лише один агент, можна знайти всі кути коридору одним махом і анімувати особистість, щоб вона рухалася уздовж сегментів, які з'єднують кути. Маючи справу з кількома агентами, що рухаються одночасно, уникаючи один одного, вони змушені відхилятися від початкового маршруту. Спробувати виправити такі відхилення, використовуючи маршрут, що складається з сегментів, стає складно, а також може бути схильний до скоєння поширених помилок.

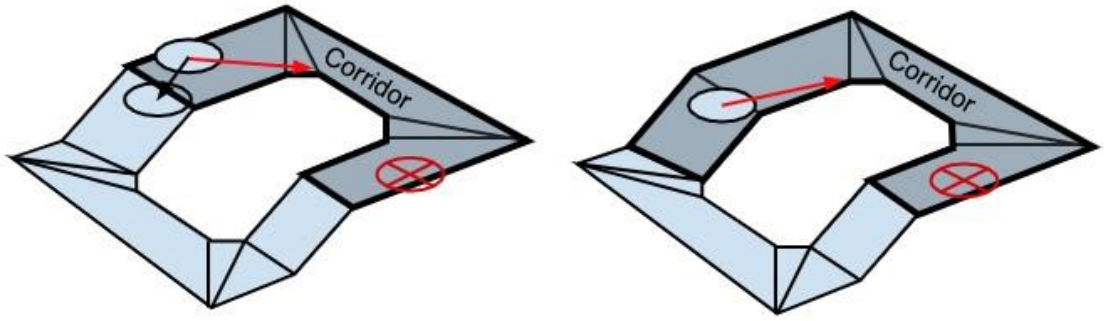


Рисунок 2.10 - Багатонаправлене трасування до місця призначення

Оскільки рух агента в кожному кадрі є досить низьким, ми можемо використовувати багатокутне підключення, щоб вирішити коридор, якщо нам потрібно трохи обійти. Потім ми виявляємо наступний помітний кут, яким потрібно швидко керувати.

• Уникнення перешкоди

Логіка рульового керування вимагає наступного кутового положення та необхідного напрямку та швидкості

(або швидкість), необхідна для досягнення цілі, заснована на цих цифрах. Використання необхідної швидкості для переміщення агента може призвести до зіткнення з іншими агентами. Уникнення перешкод вибирає нову швидкість, яка балансує між рухом у потрібному напрямку та запобіганням майбутніх зіткнень з іншими агентами та краями навігаційної сітки. Unity використовує бар'єри для прогнозування та припинення зіткнень із зворотною швидкістю (RVO).

• Переміщення агента до цілі

Остаточна швидкість розраховується після кермування та об’їзду перешкод. Агенти моделюються в Unity з використанням простої динамічної моделі, яка також вимагає прискорення, щоб забезпечити більш природний і плавний рух. На цьому етапі ви можете надати систему анімації швидкості від імітованого агента, щоб перемістити персонажа за допомогою кореневого руху. Навігаційна система також може подбати про це. Після переміщення агента за допомогою будь-якої техніки позиція змодельованого агента змінюється і обмежується NavMesh. Для надійної навігації цей останній крихітний крок є важливим. Функції реалізації, відсутні в інших моделях

• Знищення об'єкта

Public static void Destroy(Object obj, float t = 0.0F);

Функція Destroy() використовується для знищення будь-якого об’єкта. Якщо obj є компонентом, елемент видаляється та знищується об’єктом GameObject. Якщо obj є GameObject, він знищує всі компоненти та дочірні

елементи GameObject. Фактичне знищення об’єктів завжди відкладається до завершення поточного циклу оновлення. Destroy є спадковим від базового класу UnityEngine. Об'єкт.

• Інші функції Unity, які ми використовували:

o RAYCAST

Передача променів — це процес спуску невидимого променя з точки на визначеному курсі, щоб визначити, чи є якісь зіткнення на його шляху. Простежуючи кулю або лазер від початку до кінця, ми дізнаємося, як він поводиться. У середовищі моделювання ми можемо фізично спостерігати і маніпулювати ним. Це має бути корисним для пошуку шляху до найближчих воріт або для перевірки, чи є перешкода попереду чи ні, а також у випадку висувних крісел, про які ми будемо говорити в наступних модифікаціях. Unity3D raycasting ділиться на дві окремі частини, 2D і 3D. Обидва є об'єктами фізики. o Сопрограми

Корутини гарантують, що будь-яка дія, яка має місце у функції, має відбуватися в межах одного оновлення кадру; виклик функції не може використовуватися з часом, щоб містити процедурну анімацію або послідовність випадків. Розглянемо процес поступового зменшення значення альфа (прозорості) об’єкта, поки він не стане повністю невидимим як екземпляр. За замовчуванням, після його повернення, спрограму перезапускають у кадрі, хоча також можна ввести обмеження часу.

• Налаштування кількох місць призначення

Після впровадження всіх методів пошуку агентів і моделювання агентів для пошуку найближчого виходу наступним компонентом блок-схеми є налаштування кількох місць призначення. Таким чином, щоб показати реалістичний сценарій надзвичайної ситуації, нам потрібно кілька виходів для кінозалу або будь-якої іншої моделі. Якщо будь-який вихід заблокований, агенти йдуть до найближчого наступного виходу.

Таким чином, ми обговорили наш підхід до розробки моделі програмного забезпечення, яка може включати функції, які відсутні в інших 14 моделях програмного забезпечення. Щоб показати, що наша модель програмного забезпечення може включати ці функції, ми порівнюємо моделей з нашим підходом до моделі програмного забезпечення. Тепер ми можемо порівняти, які моделі програмного забезпечення наблизилися до нашого підходу, а чого їм не вистачає з точки зору функцій. Подібно до попереднього порівняння (Таблиці 1 і 2), ми додаємо окремий рядок в кінці таблиці 3 і таблиці 4, щоб порівняти інші 14 моделей програмного забезпечення з нашим підходом.

Наш підхід працює краще, ніж доступні моделі програмного забезпечення. Модель, яка найбільше наблизилася до нашого підходу і використовується для різних проектів, — це BuildingEXODUS (Стовпець із «зеленим»), проте в ній також відсутні деякі функції, такі як агенти з обмеженими фізичними можливостями/повільно рухомі агенти, які вона не може надати, і вона може лише імпортувати дані про пожежу, але не можна реалізувати запуск пожежного елемента одночасно з евакуацією. Слід зазначити, що ці моделі можуть працювати лише з набором параметрів, які попередньо визначені всередині програмного пакета. Ці моделі програмного забезпечення не можуть імпортувати файли для додавання нового 3D-середовища до свого програмного пакета. Тому, якщо в майбутньому з’являться будь-які можливі фізичні зміни та нова функція/параметр у технології, вони не можуть бути реалізовані в програмній моделі. Однак наш підхід може працювати з будь-якою функцією/параметром або будь-якими можливими фізичними змінами, які можна легко імпортувати та реалізувати з незначною конфігурацією. Таким чином ми долаємо всі недоліки інших моделей програмного забезпечення. Крім того, порівняння нашої програмної моделі з попередніми моделями симуляції натовпу на основі агентів дало нам ідеї щодо покращення часу моделювання та ефективності, використовуючи деякі можливі фізичні зміни. Реалізація цих

фізичних змін показує, що наша модель може працювати в будь-якому сценарії. Ми детально обговоримо всі ці фізичні зміни.

Зміни у фізичному середовищі

Після того, як усі основні компоненти робочого процесу розроблені, ми тепер можемо імпортувати зовнішній файл, розроблений у програмному забезпеченні САПР (формат файлу .fbx) або розробити фізичну зміну всередині нашої моделі, щоб перевірити, чи фізична зміна створює різницю в часі та ефективності евакуації. . Це допомагає інженерам, які беруть участь у плануванні та проектуванні, створювати прототипи фізичних змін за межами нашої моделі програмного забезпечення та імпортувати їх та перевіряти, як усі ці фізичні зміни працюють. Після впровадження будь-якої фізичної зміни або фізична зміна не дає належних очікуваних результатів, або дає відповідний результат.

Якщо це не дає належного результату, то при змінах фізичних змін можна отримати задовільні результати. Цей процес застосування змін до існуючих фізичних змін до тих пір, поки модель програмного забезпечення не надасть відповідні очікувані результати, описано в кроках 12-18 псевдокоду на малюнку.

Результати та аналіз

Цей компонент робочого процесу забезпечує результати та порівняння з фактичною подією. Ми детально обговорили після моделювання різних фізичних змін у нашій моделі кінозалу щодо їх ефективності в разі надзвичайної ситуації.

2.3 Доступність параметрів в Unity

Щоб працювати з параметрами, які користувач може перевірити, щоб увімкнути та вимкнути їх, ми розробили панель з логічним значенням з нуля. Коли нам потрібно, ми можемо додати нову функцію, до якої можна легко отримати доступ із життєво важливими значеннями, які можна відповідно

змінити, щоб забезпечити найкращі результати для моделювання. Крім того, ми можемо додати список фізичних змін у цю булеву панель, яка описана на кроці 11 псевдокоду на рисунку.

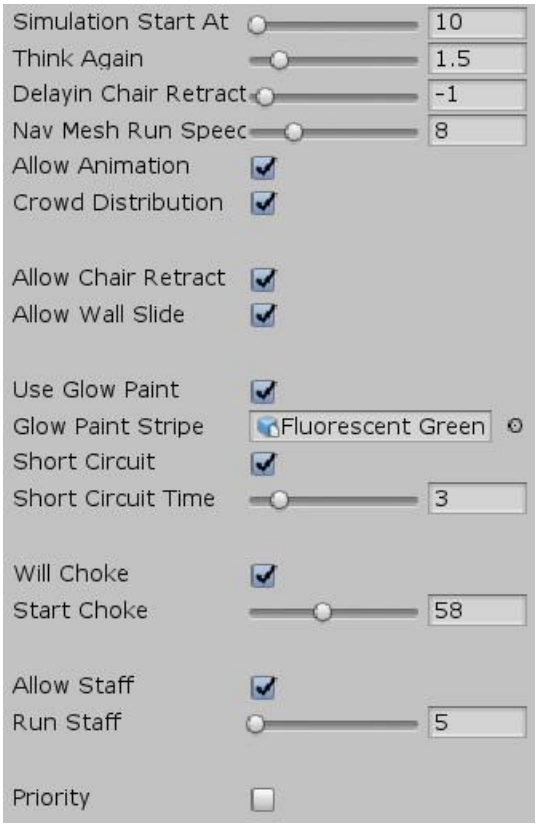


Рисунок 2.11 - Панель доступності параметрів у unity

Нижче ми обговорюємо кожен параметр/зміну, який ми включили в панель Boolean в єдності, щоб наше моделювання проходило безперебійно. Деякі з цих параметрів дійсні лише для певних фізичних змін, наприклад, дозволити втягування стільця, яке може працювати лише з висувними стільцями. Аналогічно, використання параметрів фарби світіння працює лише з фізичною зміною фарби, що світиться в темряві. Деякі параметри є обов'язковими для всіх типів сценаріїв, таких як час початку симуляції, швидкість роботи navmesh тощо.

Початок моделювання о: цей параметр надає повзунок часу для часу початку моделювання. Ми можемо вибрати від 5 секунд до 1200 секунд/10 хвилин.

Подумайте ще раз: цей параметр працює з NavMesh AI. Таким чином, щоб досягти найближчого виходу, агент повинен думати на кожному кадрі, оскільки NavMesh динамічно змінюється. Значення визначає кількість кадрів, після яких AI починає обчислення, щоб знайти найближчий вихід.

Затримка у відступі крісла: це час, який займає крісло після того, як жоден агент не наближається до крісла.

NavMesh Run Speed: він забезпечує швидкість за замовчуванням для агента, проте ми можемо забезпечити відмінні швидкості для різних типів агентів, наприклад, чоловіків (швидкість від 4 до 8), інвалідів (швидкість від 3 до 7), агентів із зайвою вагою (від 1 -4), діти (5-9) тощо.

Дозволити анімацію: цей параметр вмикає послідовність анімації для бігу, руху, падіння, спрямування тощо.

Розподіл натовпу: цей параметр включає функцію розподілу натовпу для симуляції

Дозволити відтягування стільця та дозволити ковзання стіни: після перевірки цих функцій він включає фізичні зміни висувних сидінь і рухомих стін у симуляції.

Ми реалізували сценарій диму, за яким дим піднімається через 20 секунд після симуляції. Дим може призвести до удушення агентів; Використовуючи цей параметр, ми можемо або не можемо дозволити внутрішнім агентам задихнутися. Ми також повинні призначити значення за допомогою повзунка часу, після якого агенти починають задихатися всередині симуляції.

Після детального огляду нашого підходу ми впроваджуємо набір фізичних змін, щоб перевірити роботу нашої моделі програмного забезпечення.

3 РОЗРОБКА ОРИГІНАЛЬНОГО АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО РІШЕННЯ ВІЗУАЛІЗАЦІЇ ТЕКСТОВИХ РЕКОНСТРУКЦІЙ

3.1 Фізичні зміни, які розглядаються для моделювання

Нижче наведено деякі фізичні зміни, які недоступні для комерційного використання, а деякі з них доступні, але не використовуються в повній мірі. У розділі 3.1 ми пояснюємо, як діють різні фізичні зміни. Усі роз'яснені фізичні зміни відповідають ходу нашої методології. Спочатку проектування моделі кінозалу виконується в Autodesk Maya 2018, потім файл .Fbx моделі, включаючи фізичні зміни, імпортується до Unity 3D для подальших удосконалень. На наступному кроці створюється динамічна навігаційна сітка, по якій переміщуються всі мешканці/агенти. Пізніше реалізація деяких специфічних функцій, таких як дим, вогонь тощо, виконується на моделі кінозалу, щоб забезпечити реалістичний сценарій евакуації. Потім виконується сценарій для агентів, щоб направити їх до найближчих дверей. Вся ця установка є багатоцільовою установкою. Зрештою, ми моделюємо ці фізичні зміни окремо, а також разом, щоб забезпечити час та ефективність евакуації для конкретного типу сценарію. Результати моделювання детально пояснюються в розділі . Ці результати забезпечують ефективність цих фізичних змін у разі надзвичайної ситуації.

У всьому світі баскетбольні стадіони зазвичай мають висувні сидіння, щоб зменшити простір стадіонів. Хоча місця на стадіонах можна зняти, лише якщо пасажири звільнили весь ряд або зону відпочинку. Висувні сидіння на стадіонах не можуть працювати, навіть якщо на будь-якому з сидінь сидить один пасажир, оскільки вся частина сидінь механічно з'єднана між собою, щоб одночасно вбиратися. Проте висувні сидіння ніколи не розглядалися для розстановки місць у кінозалі. Тому для цієї дипломної роботи ми включили ці

висувні сидіння в нашу модель кінозалу з окремим набором механізму, в якому не обов'язково звільняти весь ряд або частину сидінь, щоб зменшити простір. У цьому методі, коли людина встає і більше не знаходиться поблизу стільця, він автоматично втягується в підлогу, що дає іншій людині вільно переміщатися по цій ділянці під час надзвичайної ситуації. Завдяки цьому, як тільки люди в першому ряду звільняються, це дає додатковий простір для людей у задньому ряду, щоб перейти до найближчого місця виходу. Ми використовували функцію raycast для цього сценарію, в якому промінь відкидається від стільця під кутом 45 градусів, щоб побачити, чи є деякі перед кріслом, якщо це не так, він автоматично втягується. Таким чином, ми розробили нашу модель кінозалу з нуля за допомогою Autodesk maya, щоб імітувати всі фізичні зміни. Спочатку була розроблена модель, а потім ми імпортували файл .fbx, що містить кожну фізичну зміну, в даному випадку висувні сидіння. Отже, якщо в майбутньому відбудуться будь-які можливі фізичні зміни, наша модель програмного забезпечення може легко включити цю фізичну зміну, щоб перевірити її ефективність.



Рисунок 3.1 - Висувні сидіння

Через це стався інцидент з пожежею Karamay Fire, коли партійним чиновникам було надано перевагу перед школярами, щоб ми могли віддавати перевагу переміщенню конкретних агентів (дітей та агентів з повільним рухом) над звичайними агентами. Як тільки інвалідний візок досягає дверей (спеціальні двері, доступні для інвалідних візків), звичайні агенти відходять

убік і поступаються місцем особам з обмеженими фізичними можливостями/повільно рухаються або дітям.

Іноді проблеми виникають, коли більшість людей використовує один і той же вихід. Таким чином, за допомогою додаткових аварійних виходів можна зменшити тисняву. Аналогічно, для додаткових аварійних виходів ми імпортуємо файл .fbx із внесеними змінами та моделюємо його на моделі програмного забезпечення, щоб перевірити ефективність, перш ніж він буде створено.

Найбільша причина деяких людських тисняв полягає в тому, що ворота не відкривалися. У таких ситуаціях можуть бути корисні ручки, які відкриваються просто натисненням. Аналогічно, для дверей із тривожними планками ми імпортуємо файл .fbx, який вносить зміни та імітує їх на програмній моделі, щоб перевірити ефективність, перш ніж вона буде створена.

У кожному кінотеатрі, як тільки починається фільм, співробітники розбігаються, залишаючи зал без нагляду, а в деяких випадках нікого немає, щоб допомогти людям під час надзвичайної ситуації. У таких випадках персонал може замовників і направити їх до найближчого виходу. Аналогічно, для додаткового персоналу ми імпортуємо файл .fbx із внесеними змінами та моделюємо його на програмній моделі, щоб перевірити ефективність, перш ніж вона буде створена.



Рисунок 3.2 - Допоміжний персонал

Під час пожежі іноді вимикається світло, що може призвести до затемнення в залі. Люди не бачать найближчого виходу. Однак світиться в темряві фарба на підлозі може дозволити людям знайти шлях до виходу. Це зменшує хаос і підвищує ефективність системи. Світяться в темряві смуги фарби також можна побачити на проході літака, який направляє пасажирів до найближчого виходу. Аналогічно, для фарби, яка світиться в темряві, ми імпортуємо файл .fbx, який включає зміни та імітує їх у моделі програмного забезпечення, щоб перевірити ефективність перед фактичною збіркою.

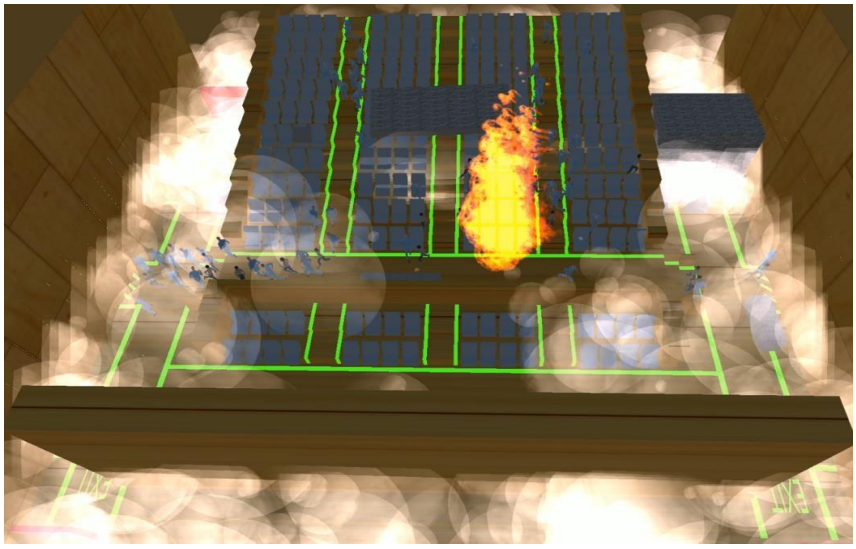


Рисунок 3.3: Світиться в темряві фарба [48]

Під час аварійної евакуації більшість коридорів і виходів створює вузьке місце, що призводить до жертв через тисняву. Усі зали кінотеатрів світу мають максимум 4-6 виходів шириною 2 м, що недостатньо для натовпу в 300-400 людей, що мчить до виходів одночасно. Також виходи в кінозалі розміщені на дальніх кінцях, що збільшує час евакуації. Тому для цієї дипломної роботи ми включили набір рухомих стінок, завдяки яким мешканцям не доведеться поспішати знайти вихід і рухатися до нього. У цьому процесі бічні стінки кінозалів зсуваються з середини, створюючи прохід для виходу з кінозалу, таким чином зменшуючи зусилля та час для мешканців, щоб дістатися до виходів навіть у віддалених кінцях. Для того, щоб у цей час зменшити кількість

людей, можна зробити конструкцію з рухомими боковинами, щоб люди могли вільно пересуватися. Аналогічно, для рухомих стін ми імпортуємо файл .fbx, який містить зміни, і моделюємо його на програмній моделі, щоб перевірити ефективність, перш ніж вона буде створена.

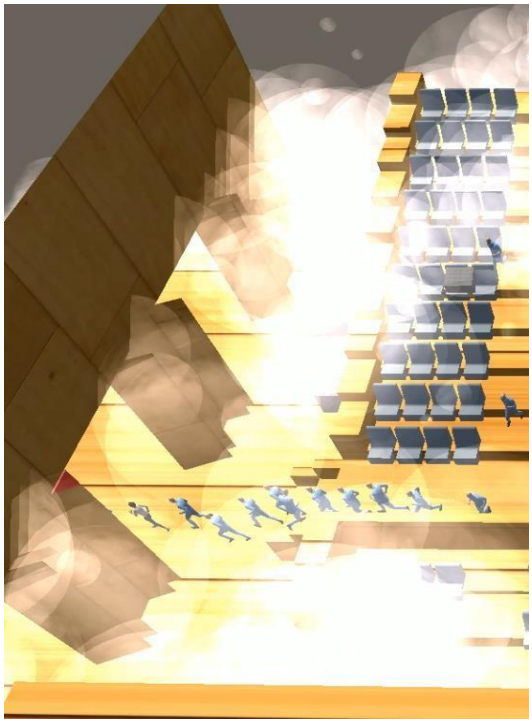


Рисунок 3.4: Рухомі стінки [49]

Після спроби реалізувати ці фізичні зміни з меншим натовпом ми спробували створити модель для розміщення 300-500 людей одночасно в будівлі, яка зображує

Ми надали середовищі той самий набір правил і протоколів, які ми надали в моделі кінозалу, як-от пожежа, розподіл натовпу, інвалідні візки тощо. Оскільки натовп збільшує обчислювальну потужність, а також збільшується потужність обробки.

Нижче наведено зображення, що зображують організацію моделі центру. До них відноситься розстановка стільців по всій арені, розміщення дверей, сцени і т.д.

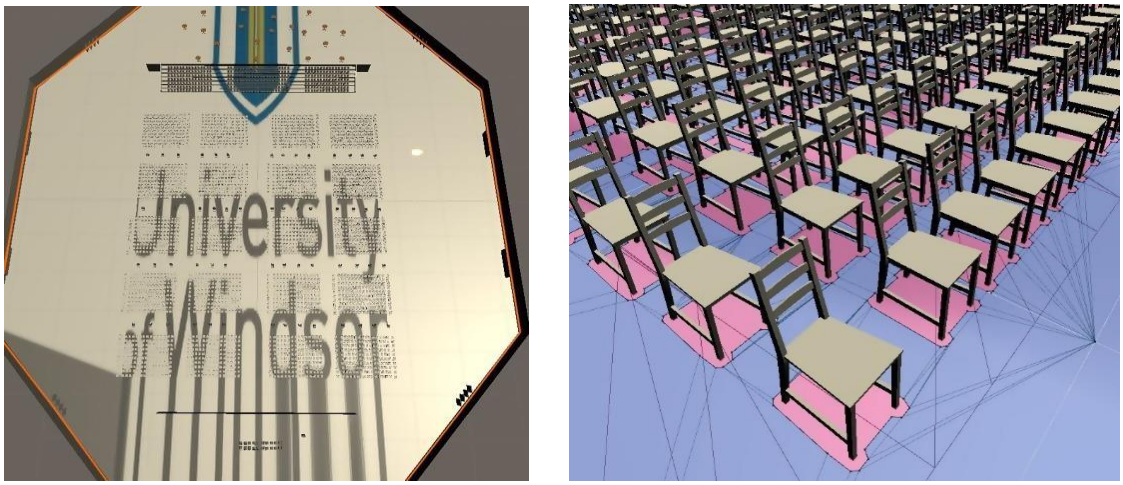


Рисунок 3.5: Арена та сітка між кріслами, що проходять

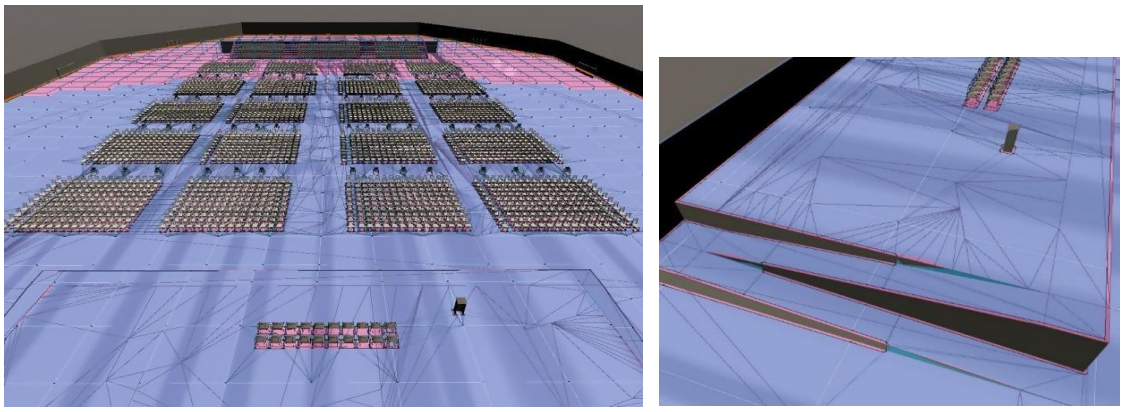


Рисунок 3.6: Меш з можливістю проходження та без проходження в повному масштабі

Арена-центр зазначити, що інші моделі програмного забезпечення не можуть імпортувати жодного файлу для додавання нового 3D-середовища до свого програмного пакету. Тому, якщо в майбутньому з’являться будь-які можливі фізичні зміни та нова функція/параметр у технології, вони не можуть бути реалізовані в програмній моделі. Однак наш підхід може працювати з будь-якими можливими фізичними змінами до того, як вони будуть фактично сконструйовані/побудовані, або які можуть статися в майбутньому, які можна легко імпортувати та реалізувати з незначною конфігурацією. Пропонуючи набір фізичних змін та різні інші функції, щоб наша модель працювала в будь-

якому сценарії, щоб подолати недоліки інших моделей програмного забезпечення.

3.2 Конфігурації апаратного та програмного забезпечення

Ми розробили та протестували нашу модель програмного забезпечення на комп'ютері з ОС Windows. Він містить 3,3 ГГц процесор Intel® Core i7TM з 32 ГБ пам'яті DDR3 і 4 ГБ NVIDIA GeForce GTX 960 відеокарта. Основна мета нашої програмної моделі — забезпечити реалістичне моделювання натовпу для аварійної ситуації з роздільною здатністю дисплея 2560 X 1440 пікселів.

Ми розробили нашу модель програмного забезпечення, використовуючи певне програмне забезпечення, яке може працювати на кількох платформах:

- Unity 3.1.4f1 Personal (64-розрядна версія)
- AutoDesk Maya 2019 (освітня версія)

3.3 Експериментальне дослідження та результати імітаційного моделювання

У цьому розділі ми обговорюємо експерименти, які ми проводили над моделлю нашого кінозалу, використовуючи набір сценаріїв, що включають певні фізичні зміни та порівнюючи їх із реальною подією. Пізніше ми фіксуємо результати та робимо короткий висновок, чи були фізичні зміни ефективними чи ні. Цей розділ розміщено в наступному порядку:

Спочатку обговорюємо облаштування загального кінозалу та результати у разі відсутності надзвичайної ситуації.

Ми моделюємо сценарій, схожий на реальний інцидент, який стався, для порівняння результатів без урахування короткого замикання. Ми розглядаємо це як сценарій 1 для моделювання.

Ми розглядаємо всі результати сценарію 1 як результати за замовчуванням для порівняння.

Ми моделюємо роботу висувних сидінь і пов’язані з цим результати, щоб показати його ефективність у разі надзвичайної ситуації.

Ми моделюємо роботу рухомих стін і пов’язані з нею результати, щоб показати ефективність у разі надзвичайної ситуації.

Ми моделюємо комбіновану стратегію, в якій моделювання зображує роботу висувних сидінь і рухомих стін разом, а також результати, пов’язані з цим, щоб показати ефективність у разі надзвичайної ситуації.

Якщо порівняти з фактичною подією, то в той день було чорне середовище. Щоб відтворити навколишнє середовище, ми моделюємо сценарій з електричним коротким замиканням для порівняння наших результатів. Ми розглядаємо це як сценарій 2 для моделювання, а також результати цього сценарію вважаються за замовчуванням для подальшого експериментування інших фізичних змін.

Ми моделюємо роботу смуг фарби, що світяться в темряві, та результати, пов’язані з цим, щоб показати його ефективність у разі надзвичайної ситуації.

Ми моделюємо роботу додаткового персоналу та пов’язані з нею результати, щоб показати ефективність у разі надзвичайної ситуації.

Ми моделюємо комбіновану стратегію, в якій моделювання зображує роботу світлових смужок у темряві та додаткового персоналу разом, а також результати, пов’язані з цим, щоб показати ефективність у разі надзвичайної ситуації.

Щоб отримати результати, ми розробили модель нашого кінозалу з нуля за допомогою Autodesk Maya 2019 і Unity 3D. Ця модель включає в себе місця для 192 агентів, з чотирма вихідними дверима. Ми створили 101 агента для цієї розстановки, включаючи п’ять інвалідних візків для людей з обмеженими фізичними можливостями в перших рядах.

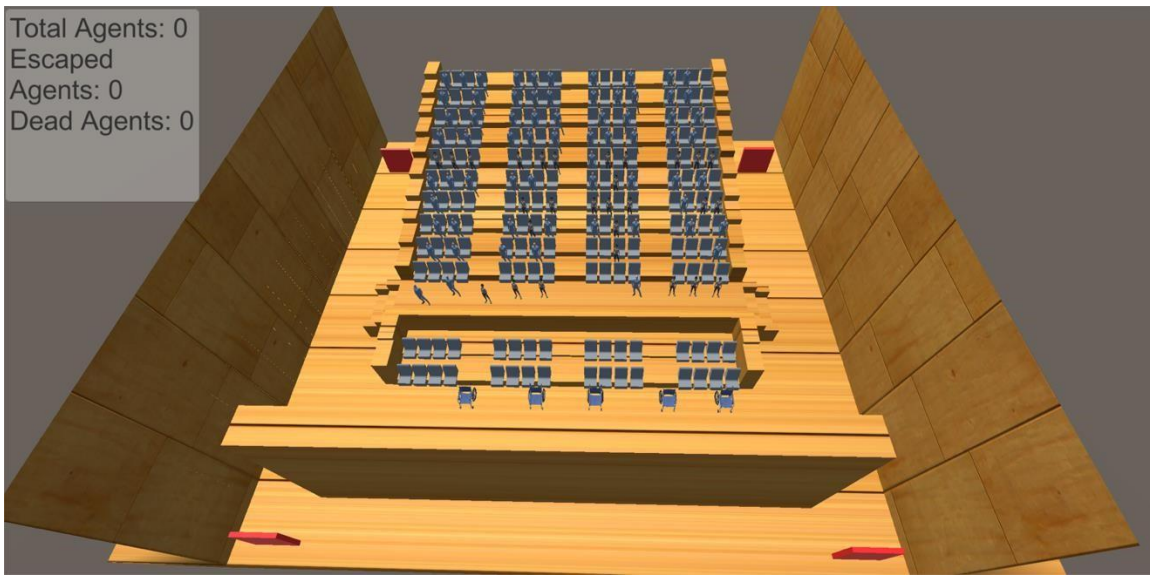


Рисунок 3.7: Налаштування кінозалу за замовчуванням

Ми записали час евакуації за відсутності будь-якої аварійної ситуації, наприклад пожежі, задимлення, заблокованого виходу тощо. Мітка результатів (верхній лівий кут) містить різноманітний набір параметрів, таких як загальна кількість агентів, які розглядаються для моделювання, кількість агентів, які можуть втеча з довкілля без будь-якої шкоди, кількість мертвих агентів наприкінці симуляції та час у секундах, поки останній агент може втекти або померти відповідно до конкретного сценарію. Також є 3 параметри, які вказують на причину смерті певної кількості агентів.

Смерть від пожежі: цей параметр враховує смерть агента, який сидів біля вогню або біг біля вогню в середовищі моделювання. Цей параметр імітує загибель агентів, викликану горінням.

Смерть від даху (падіння сміття): цей параметр враховує смерть агента, викликану падінням даху/сміття безпосередньо на його голову. Так як блоки з даху могли впасти в будь-якому місці всередині середовища.

Смерть від удушення : цей параметр враховує смерть агентів через надмірне вдихання диму або будь-якого іншого токсичного матеріалу. Це явище призводить до задухи і агент гине протягом кількох секунд у середовищі моделювання.

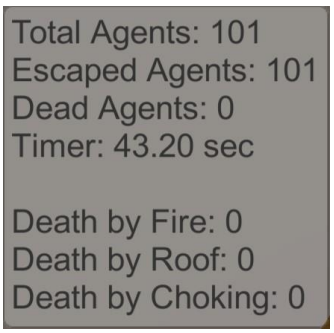


Рисунок 3.8 - Результат часу евакуації за замовчуванням

У нашому першому експерименті без аварійної ситуації всім 101 агенту знадобилося 43,20 секунди, щоб вийти з кінозалу. Беручи це до уваги, ми додатково перевіряємо результати, збільшуючи складність моделювання.

Щоб надати результати для моделювання, його слід порівняти з інцидентом, який стався в минулому. У цьому інциденті 59 осіб, які залишилися в пастці всередині, загинули, насамперед через задуху, задуху, пожежу, заблоковані виходи, чорну тьму, навколишнє середовище тощо. Щоб забезпечити подібну перспективу, ми впровадили різні функції та параметри, які залишаються незмінними під час застосування наших запропонована методологія.

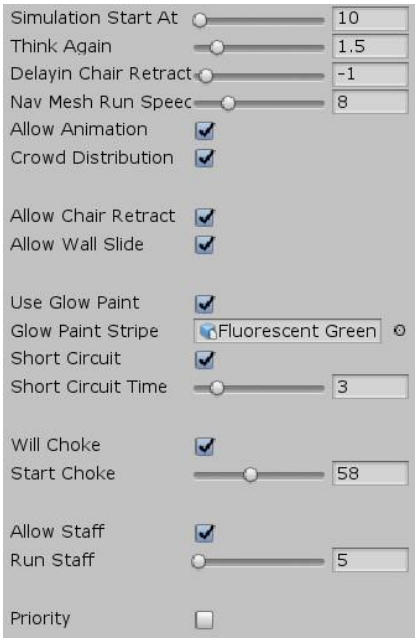


Рисунок 3.9: Панель доступності параметрів у unity

Аналогічно, у випадку затримки втягування стільця, час за замовчуванням вважається -1, оскільки це дозволяє порожнім стільцям втягнутися до початку моделювання. Таким чином, зображуючи реалістичний сценарій.

Інші параметри та функції, такі як імітація починається з, дозволяє відтягувати крісло, дозволяти ковзати стіну, використовувати фарбу, що тліється, коротке замикання, час короткого замикання, захлинається, запускається задушення, дозволяє персоналу, запускати персонал і Пріоритет змінюється залежно від сценарію.

На рисунку 3.10 зображено схожість із кінотеатром. Навколо був дим, на певних етапах вогонь, деякі виходи заблоковані, дах падає на людей, скрізь уламки.

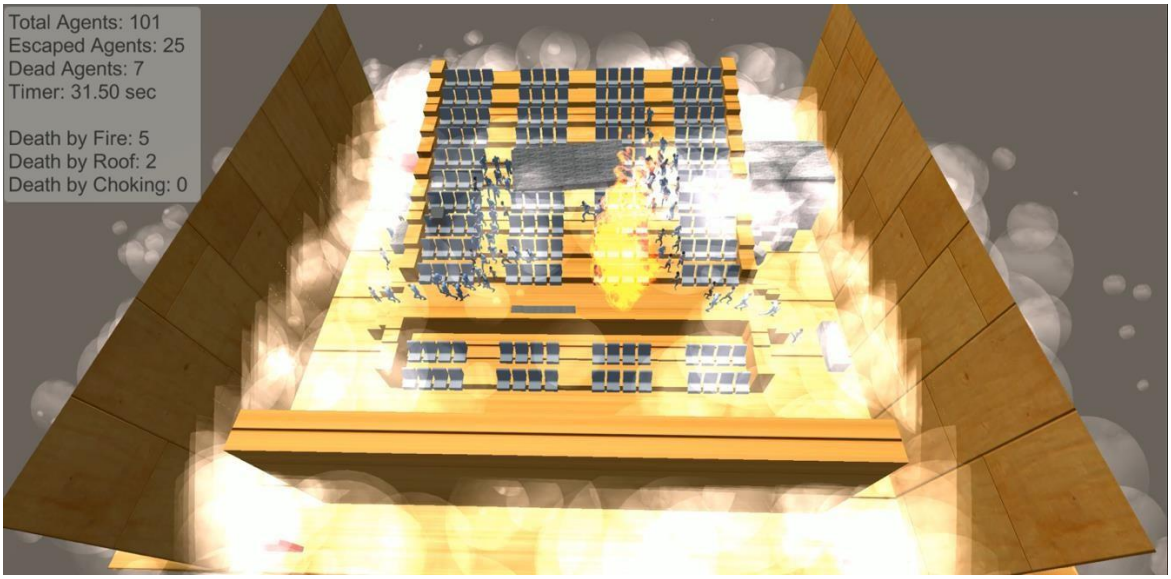


Рисунок 3.10: Симуляція, що показує схожість із Uphaar Cinema

Як повідомлялося раніше, за цей день із зазначених причин загинуло 59 осіб. Ми розглянули подібний відсоток живих агентів і мертвих агентів у порівнянні зі сценарієм у кінотеатрі. Після встановлення всіх точних параметрів наше моделювання програмної моделі дало подібні результати.

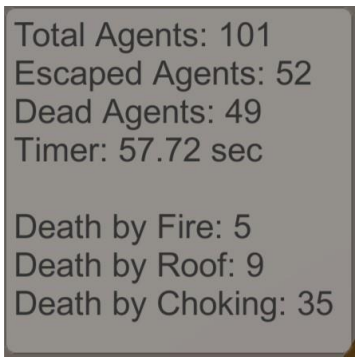


Рисунок 3.11: Результати для сценарію за замовчуванням без короткого замикання (сценарій 1)

Загальна кількість використаних нами агентів становить 101, з яких 52 втекли протягом 57,72 сек. Загинуло 49 агентів, які можна відсортувати за типом смерті.

Як показано на агентів загинули, перебуваючи поблизу вогню. Також 9 агентів були розчавлені з бетонних блоків, які випали зі стелі, а 35 агентів, охоплених димом, не вийшли вчасно і захлинулися.

Тепер ми розглядаємо це як сценарій за замовчуванням для подальшого порівняння з нашою запропонованою методологією, включаючи певні фізичні зміни, які можуть вплинути на терміни та ефективність евакуації. Таким чином, час для порівняння з іншими результатами вважається 57,72 секунди, за які загинуло 49 агентів, що допомагає нам знайти ефективність запропонованої нами програмної моделі. Ми розглядаємо сценарій без короткого замикання як «сценарій 1».

Тепер ми розглянемо кожну фізичну зміну чи функцію окремо, що було обговорено. Спочатку ми коротко пояснимо, як ця зміна чи функція працює. Потім ми надаємо терміни та ефективність евакуації в тому ж форматі, щоб зменшити плутанину. Наприкінці кожного результату ми надаємо висновок для нього.

Висувні сидіння:

Перша фізична зміна, яку ми розглядаємо, це висувні сидіння.

Він працює на основі основної ідеї: якщо поблизу сидіння нікого немає, крісло автоматично втягнеться в підлогу, забезпечуючи таким чином додатковий простір для пересування агентів у кінозалі.

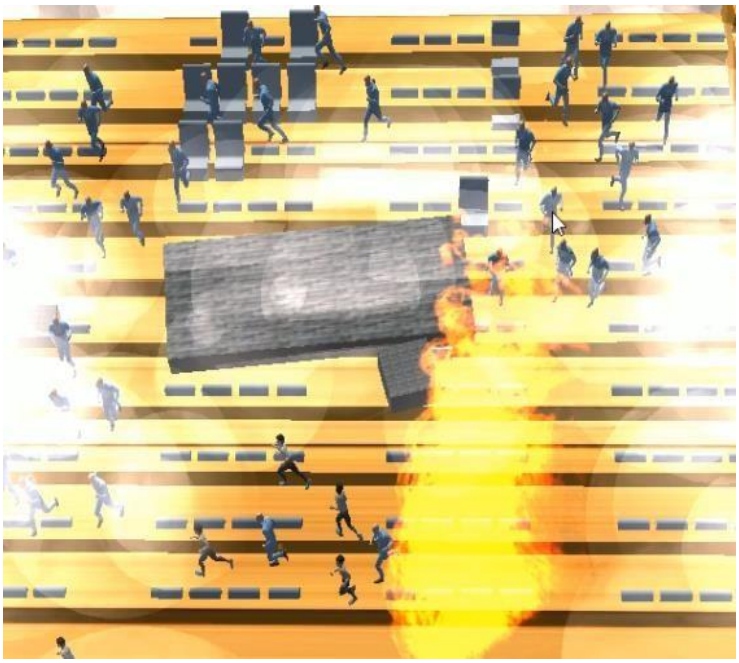
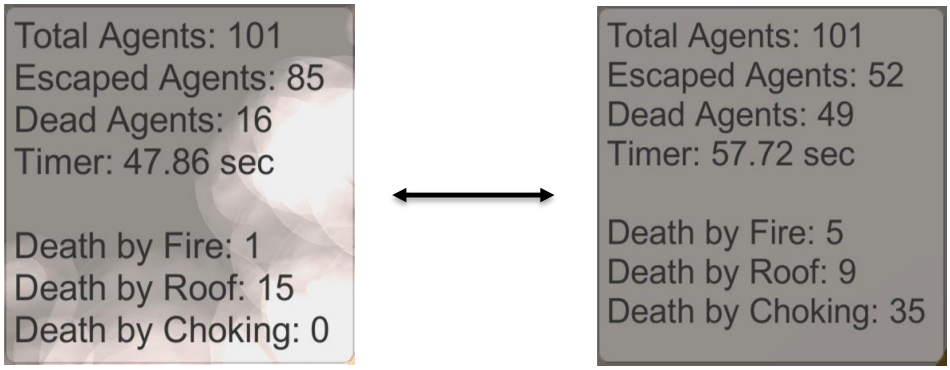


Рисунок 3.12: Моделювання з використанням висувних сидінь

Як зображено на малюнку 3.12, агенти рухаються до найближчого виходу, і більшість крісел прибрано на підлогу. Однак деякі місця все ще підняті, тому що агенти все ще рухаються навколо них, і навіть без місць вони можуть рухатися по діагоналі, що скорочує час на евакуацію.



(а) (б)

Рисунок 3.13: Порівняння (а) результатів моделювання з використанням висувних сидінь з (б) результатами сценарію 1

У цій симуляції лише 16 агентів загинули, а решті 85 агентів знадобилося 47,86 секунди, щоб безпечно вибратися, але деякі з агентів загинули через пожежу та падіння даху/уламків. Що ще важливіше, агенти вийшли до того, як їх задимлють, в основному завдяки тому, що висувні сидіння економлять час.

Порівнюючи результати на малюнку, ми можемо зробити висновок, що час евакуації скоротився на 17,08%. Аналогічно ефективність системи зросла на 67%, оскільки врятовано 33 життя порівняно зі сценарієм 1 (сценарій без короткого замикання).

Переміщення стін:

Наступна фізична зміна, яку ми запровадили, — рухомі стіни після зняття прапорців з висувних сидінь. При цьому бічні стінки кінозалу зсуваються з середини, створюючи прохід для виходу з кінозалу, таким чином зменшуючи зусилля та час для агентів, щоб дістатися до виходів навіть у віддалених кінцях.

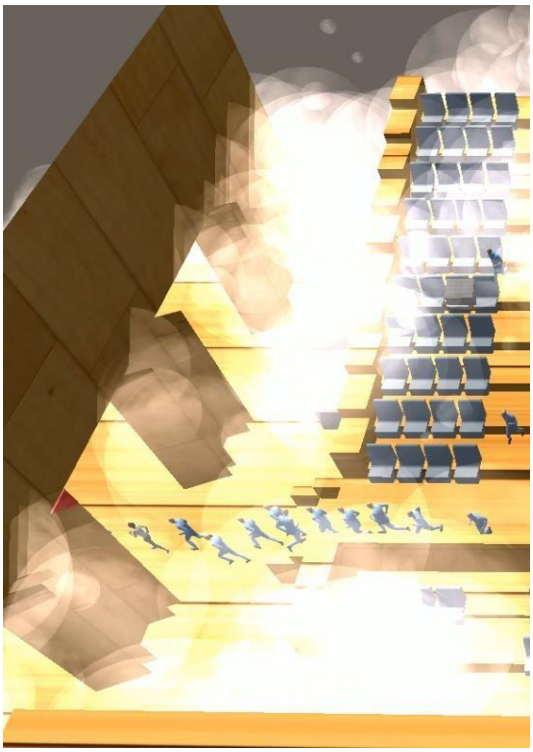
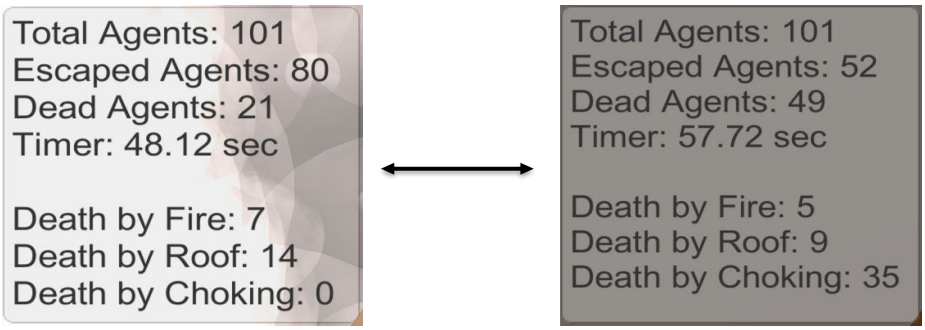


Рисунок 3.14: Моделювання за допомогою рухомих стін

Як показано на малюнку, агенти рухаються до проходу, створеного ковзними стінками.

Таким чином, вони можуть швидко вийти з кінозалу, а не шукати найближчий вихід.



(а) (б)

Рисунок 3.15: Порівняння (а) результатів моделювання з використанням рухомих стін з (б) результатами сценарію 1

Моделювання показує, що в цьому випадку загинув лише 21 агент, тоді як 80 агентів змогли вибратися лише за 48,12 секунди. Загибель агентів сталася насамперед через пожежу та падіння даху. Розсувні стінки відігравали життєво важливу роль у скороченні часу та зусиль, тому задуха призвела до 0 смертей, оскільки агенти вийшли до початку задушення.

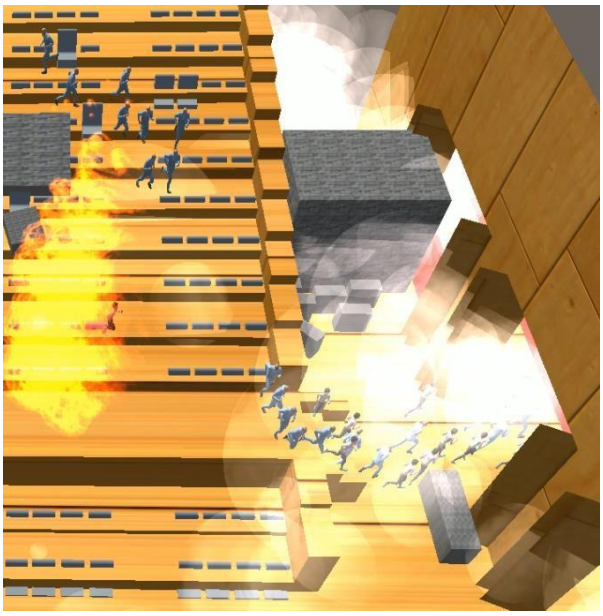
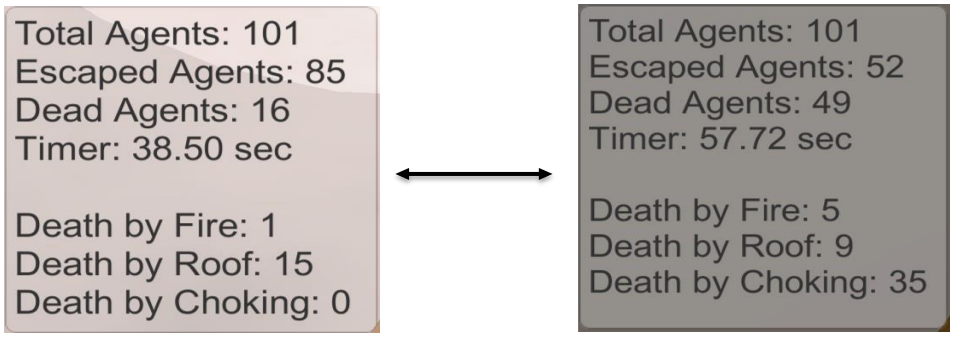


Рисунок 3.16: Моделювання з використанням як висувних сидінь, так і рухомих стін

Як показано на малюнку, висувні сидіння та розсувні стінки працюють одночасно. Цей випадок дає нам більш помітні результати порівняно з раніше обговореним, використовуючи лише одну зміну в даний момент часу.



(а) (б)

Рисунок 3.17: Порівняння (а) результатів моделювання з використанням як висувних сидінь, так і рухомих стін з (b) результатами сценарію 1

Загальний результат показує, що лише 16 агентів загинули, і 85 агентам знадобилося 38,50 секунди, щоб безпечно вибратися, але деякі з агентів загинули через пожежу та падіння даху. Порівнюючи результати на малюнку 30, ми можемо зробити висновок, що, використовуючи обидві фізичні зміни разом, ми можемо значно скоротити час евакуації, який виявився на 33,23 % менше. Ефективність системи залишилася рівною в порівнянні зі сценарієм, коли ми використовували лише висувні сидіння, що становило 67%. Результати моделювання з використанням висувних сидінь і рухомих стін у порівнянні зі сценарієм без електричного короткого замикання наведено в таблиці 3.1.

Таблиця 3.1: Результати часу та ефективності сценарію без електричного короткого замикання

Фізичні зміни	Час евакуації (зменшений) %	Ефективність системи (Підвищена) %
Висувні сидіння	17.08	67,00
Рухомі стіни	16.63	57.14
Висувні сидіння + Рухомі стіни	33.23	67,00

У таблиці 3.1 ми надали підсумок скорочення часу евакуації та підвищення ефективності системи для фізичних змін, таких як висувні сидіння та рухомі стінки. Нам також вдалося записати результати для нашої стратегії змішування та матчу для цих двох фізичних змін.

Як згадувалося раніше, того дня в кінотеатрі панувала непроглядна темна атмосфера. Щоб відтворити навколишнє середовище, ми запровадили новий сценарій моделі кінозалу, в якому світло згасало через коротке замикання, що, у свою чергу, ускладнювало агентам пошук найближчого виходу.

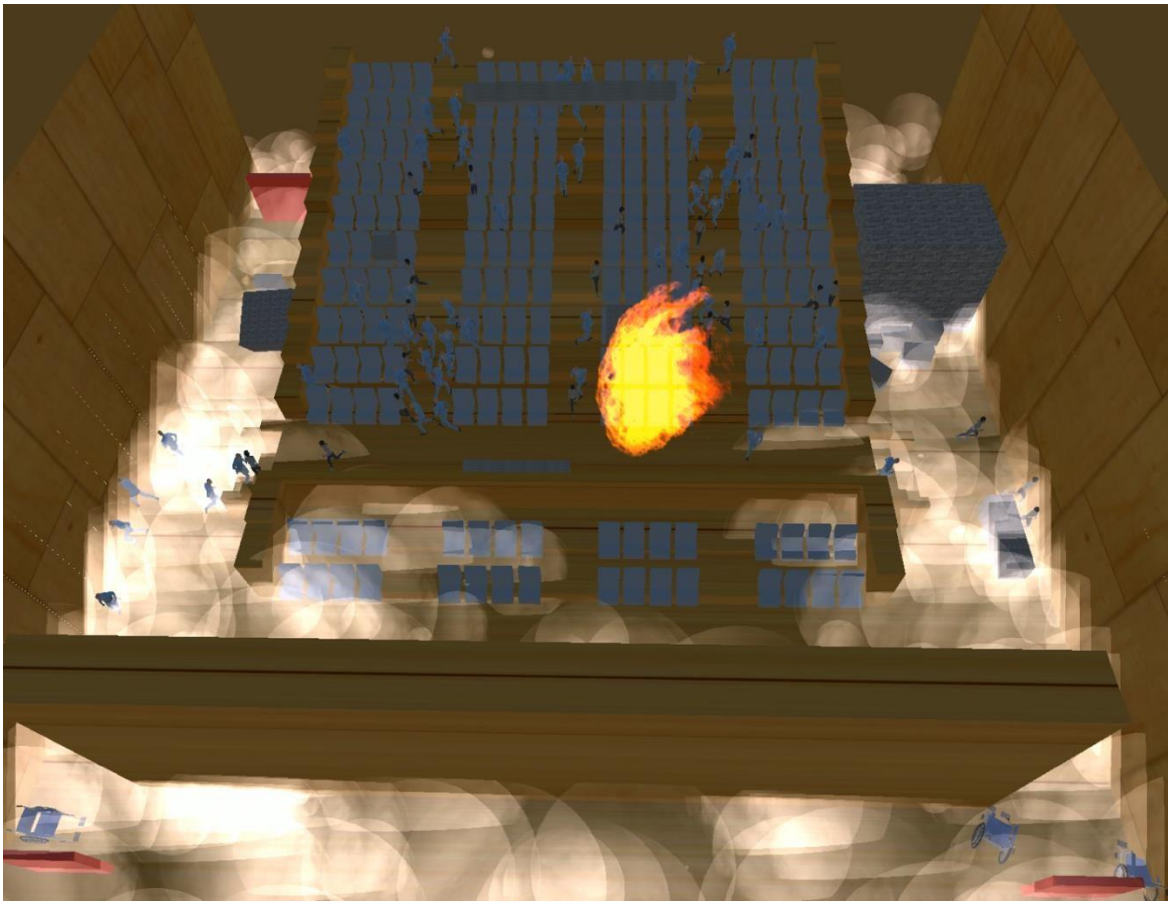


Рисунок 3.18: Симуляція, що показує схожість із Uphaar Cinema у випадку короткого замикання

Ми встановлюємо час короткого замикання в 3 секунди, щоб коротке замикання відбувалося через 3 секунди, коли починається моделювання. Як

показано на малюнку, темне середовище на додаток до попереднього сценарію з пожежею, димом, заблокованими виходами тощо.

Якщо уважно помітити, агенти в цій сцені рухаються випадковим чином, оскільки вони не знають напрямків до найближчого виходу, що схоже на ситуацію в кінотеатрі. Приголомшливо, лише дуже невелика кількість агентів змогли знайти вихід із кінозалу, і більшість із них померли всередині.



Рисунок 3.19: Результати для сценарію за замовчуванням з електричним коротким замиканням (сценарій 2)

Результат надає інформацію, що 95 агентів загинули і лише 6 агентів змогли безпечно вибратися. Агенти загинули внаслідок пожежі та падіння даху. Проте дим погіршив ситуацію, 76 агентів загинули через задишку через те, що вони не змогли знайти напрямки до найближчого виходу.

Тепер ми розглядаємо цей сценарій з електричним коротким замиканням як сценарій 2 і вносимо деякі фізичні зміни для отримання кращих результатів у цьому типі ситуації. Ми врахували, що затримка в 3 секунди на час, коли починає задихатися, забезпечує набагато більш реалістичний сценарій у темному оточенні.

Світиться в темряві фарба:

Спочатку розглянемо фарбу, що світиться в темряві. Основна ідея цієї техніки полягає в тому, що вона освітлює шлях до найближчого виходу щоразу,

коли відбувається відключення електроенергії. Цю фізичну зміну можна спостерігати на островах всередині літака. Аналогічно, намальовані смуги на підлозі кінозалу світяться, як показано нижче.

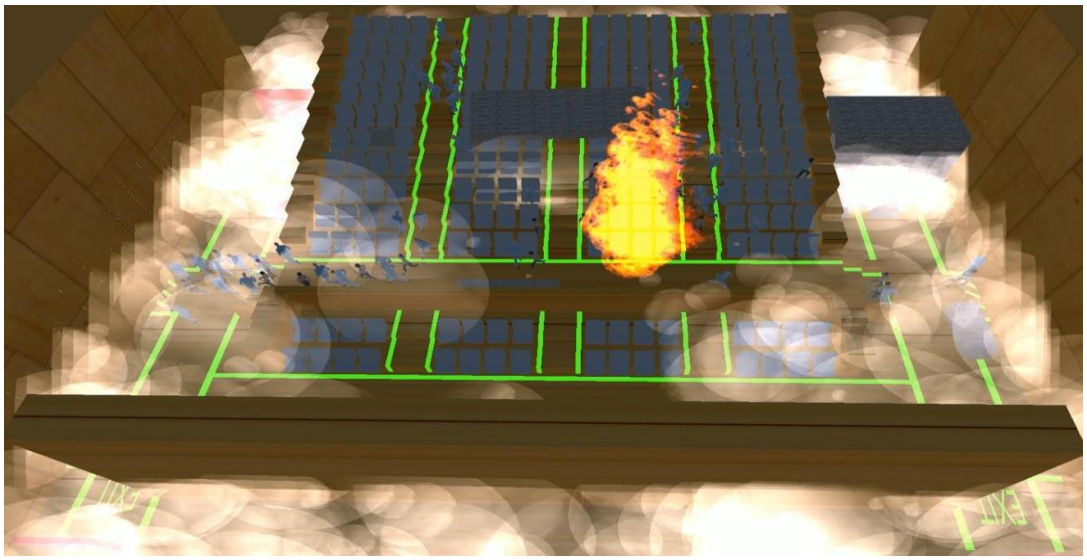
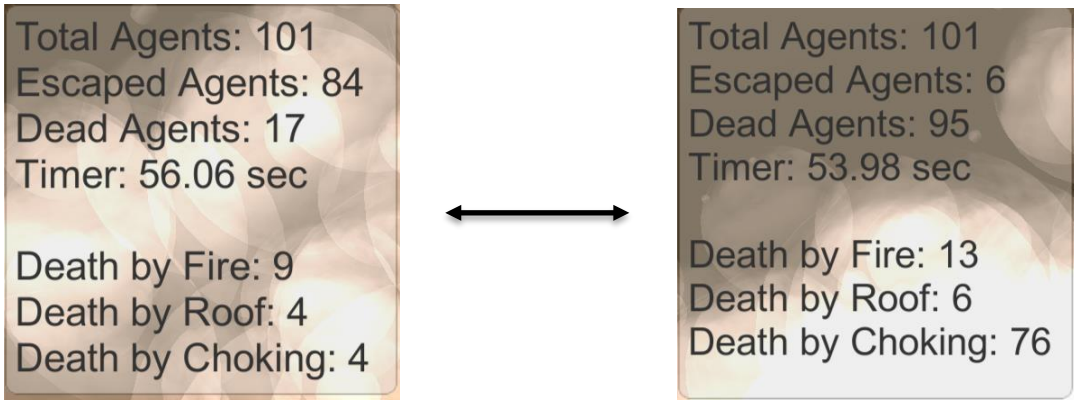


Рисунок 3.20: Моделювання за допомогою смуг фарбування, що світяться в темряві

Як показано на малюнку , сценарій залишається тим же, але намальовані смуги світяться на підлозі. Якщо ми уважно спостерігаємо, смужки спрямовують агентів до найближчих виходів, таким чином зменшуючи їхні зусилля знайти вихід у темному оточенні.



(а) (б)

Рисунок 3.21: Порівняння (а) результатів моделювання з використанням світяться в темряві смуг з (b) результати сценарію 2

Результати цього моделювання показують, що загинуло лише 17 агентів, а 84 агентам знадобилося 56,02 секунди, щоб безпечно вийти. Проте деякі агенти загинули внаслідок пожежі та падіння даху. Фарба, що світиться в темряві, виявилася важливою для безпечного виходу агентів, що призвело лише до 4 смертей через удушення. Агенти можуть вийти, а не випадковим чином рухатися всередині, щоб знайти виходи.

Порівнюючи результати на малюнку, ми можемо зробити висновок, що час евакуації збільшився на 0,04%, оскільки потрібен час, щоб пройти шлях до виходу. Однак ефективність системи збільшується на 82,10%, оскільки врятовано 78 життів порівняно зі сценарієм 2 (сценарій з коротким замиканням)

Ми використали двох агентів, щоб переміститися до певних місць і направити агентів до найближчого виходу та покинути кінозал після того, як усі вийдуть. Таким чином, щоб забезпечити реалістичний сценарій, ця зміна також потребує затримки в часі, оскільки агенти повинні вислухати персонал, а потім бігти до найближчого виходу.



Рисунок 3.22: Моделювання з використанням додаткового персоналу

Як показано на малюнку, персонал спрямовує агентів до найближчого виходу за допомогою анімованих жестів руками.



(а) (б)

Рисунок 3.23: Порівняння (а) результатів моделювання з використанням додаткового персоналу з (б) результатами сценарію 2

Результати показують, що загинуло лише 38 агентів, а 63 агентам знадобилося 59,84 секунди, щоб безпечно вийти. Деякі з агентів загинули внаслідок пожежі та падіння даху. У порівнянні з фарбою, що світиться в темряві, додаткова зміна персоналу не настільки ефективна, хоча вона зменшує кількість жертв у порівнянні з початковим сценарієм. У цьому сценарії 30 агентів загинули від задухи. Ці результати приводять нас до висновку, що час евакуації збільшився на 0,10%, оскільки персоналу знадобився час, щоб зрозуміти. Хоча ефективність системи зросла на 60%, оскільки врятовано 57 життів порівняно зі сценарієм 2 (сценарій з коротким замиканням).

Після впровадження цих двох фізичних змін ми знову спробували комбінувати стратегію, оскільки це дало нам набагато кращі результати. Для цього сценарію ми використали комбінацію фарби, що світиться в темряві, і додаткового персоналу.

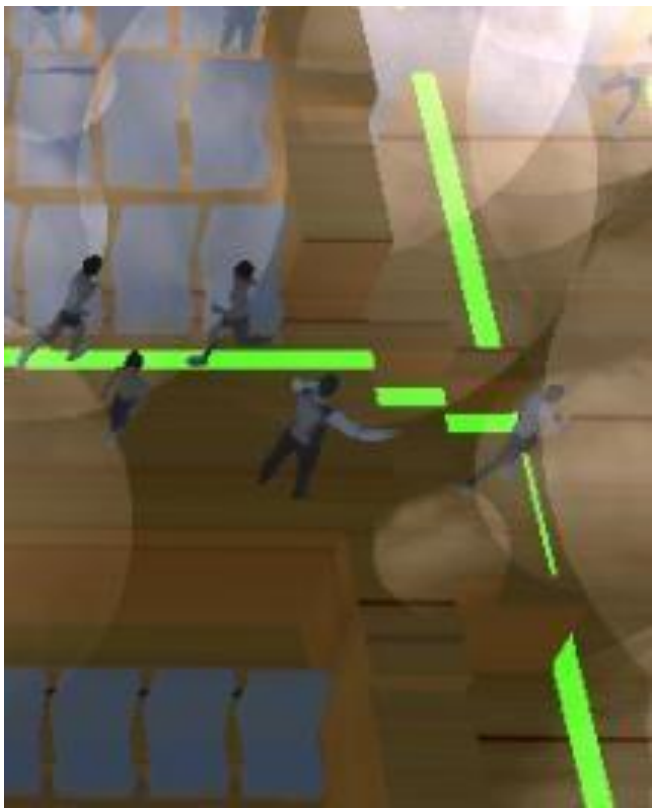


Рисунок 3.24: Моделювання з використанням як світяться в темряві смуг фарби, так і додаткового персоналу

Як показано на малюнку, є співробітник, який спрямовує їх до виходу, і вони також користуються допомогою світильника.



(а) (б)

Рисунок 3.25 Порівняння (а) результатів моделювання з використанням як світяться в темряві смуг фарби, так і додаткового персоналу з (b) результатами сценарію 2

Загальний результат показує, що загинули лише 34 агенти, а 67 агентам знадобилося 59,66 секунди, щоб безпечно вийти. Однак деякі агенти загинули внаслідок пожежі та падіння даху. Порівнюючи результати на малюнку, ми можемо зробити висновок, що, використовуючи обидві фізичні зміни, ми можемо підвищити ефективність системи, яка виявилася на 64% більше. Час евакуації менше на 0,06%.

Результати моделювання з використанням світяться в темряві смуг і додаткового персоналу в порівнянні зі сценарієм із електричним коротким замиканням обговорюються в таблиці 3.2.

Таблиця 3.2 - Результати часу та ефективності сценарію з коротким замиканням

Фізичні зміни	Час евакуації (збільшено) %	Ефективність системи (Підвищена) %
Світиться в темряві фарба	0,04	82.10
Додатковий персонал	0,10	60,00
Світиться в темряві фарба + Додатковий персонал	0,06	64,00

У таблиці ми наводимо підсумок збільшення часу евакуації та підвищення ефективності системи для фізичних змін, таких як світяться в темряві смуги фарби та додатковий персонал. Нам також вдалося записати результати для нашої стратегії змішування та відповідності для цих двох фізичних змін.

Крім усіх згаданих функцій, ми також впровадили систему пріоритетів для людей з обмеженими фізичними можливостями та дітей. У симуляції, коли звичайний агент зустрічається з дитиною або з обмеженими фізичними можливостями, він відходить убік і дає їм дорогу.

Наша модель програмного забезпечення також має параметр розподілу натовпу за замовчуванням. Він забезпечує розподілене розташування натовпу,

що рухається, якщо будь-який вихід занадто переповнений для агента, він знаходить шлях до наступного найближчого виходу.

Крім того, обидві ці функції допомагають скоротити час евакуації та підвищити ефективність системи. Однак зміни в результатах через розподіл натовпу та системи пріоритетів не були помітні, щоб показати їх окремо в таблицях. Ми також можемо показати токсичність навколишнього середовища, не показуючи диму, і все одно працювати з функцією удушення. Цей сценарій може спрацювати у випадку біохімічної небезпеки, коли гази не видно, а агенти все ще впливають у великій кількості.

Ми також змоделювали велику арену на основі церемонії скликання Віндзорського університету – закритої церемонії в спортивному залі, в якій містилися 300–400 гостей та студентів. Рисунок є загальним зображенням сценарію для такого великого натовпу.

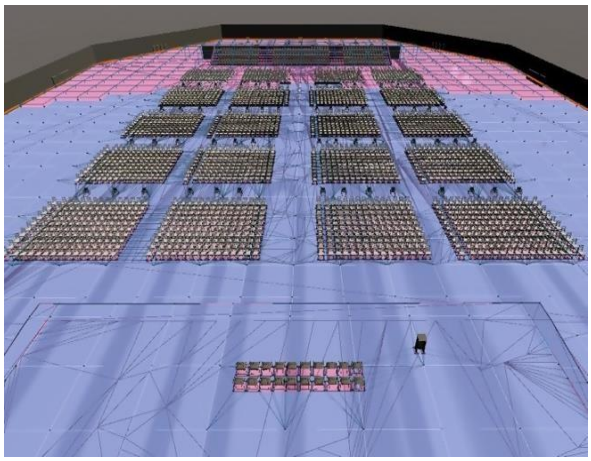


Рисунок 3.25: Прохідна та непрохідна сітка в повному масштабі Арена

Усі ці характеристики або фізичні зміни, які ми описали, дали нам багатообіцяючі результати, яких не вистачає в моделях, описаних раніше. За допомогою відповідного обчислювального середовища можна моделювати великі натовпи та спостерігати за впливом фізичних змін і навколишнього середовища.

ВИСНОВКИ

В дипломній роботі ми працювали над реалізацією моделювання передової системи аварійної евакуації. Наша реалізація базується на обмеженнях існуючих моделей, які використовуються в усьому світі. Таким чином, ми розробили модель кінозалу, щоб забезпечити подібний реалістичний ефект за допомогою Autodesk Maya 2019 та Unity 3D. Ми застосували різні функції та параметри за замовчуванням, які відбувалися протягом цього часу. Після цього ми моделювали різні запропоновані методології або фізичні зміни всередині моделі кінозалу.

Програмна модель здатна інтегрувати нові функції, а також можливі фізичні зміни в 3D внутрішньому середовищі шляхом імпорту зовнішніх файлів у форматі .fbx. Програмна модель також може моделювати різний рівень деталей і щільність натовпу, навіть більш масові натовпи.

Запропонована модель програмного забезпечення здатна працювати з великою кількістю параметрів, які можна використовувати для моделювання різних сценаріїв і множинних середовищ з великою кількістю агентів. Це може дозволити дослідникам/планувальникам спостерігати за впливом різних ситуацій на етапі проектування/планування без вкладення часу та грошей, а також ефект цих фізичних змін для збереження людського капіталу.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Al-Hanbali N., Sadoun B., Saleh M., (2006). GIS Data Model for Hospital Mapping. Proceedings of the Summer Computer Simulation Conference. Calgary, Canada, pp. 376-379.
2. Arslan, E., (2009). Nuclear, Biological and Chemical (NBC) Attack Simulation Base on Expert System. Proceedings of the Summer Computer Simulation Conference. Istanbul, Turkey; pp. 391 – 394.
3. Banks, J. (1998). Principles of Simulation, Handbook of Simulation, Wiley Interscience, New York.
4. Baysan S., Çınar D., Çelebi D., Dursun P., Özdemir D., (2009). Modeling People Flow: A Real Life Case Study in ITU Science Center. Proceedings of the Summer Computer Simulation Conference. Istanbul, Turkey, pp. 246- 253.
5. Bruzzone A., Briano E., Massei M., (2006). Simulating Transportation over a Wide Area During a Regional Crisis. Proceedings of the Summer Computer Simulation Conference. Calgary, Canada, pp. 380-384.
6. Bruzzone A., Chuna G., Elfrey P., Tremori A., (2009). Simulation for Education in Resource Management in Homeland Security. Proceedings of the Summer Computer Simulation Conference. Istanbul, Turkey, pp. 231-238.
7. Bruzzone A., Mosca R., Bocca E., Massei M., (2008). Critical issues in hla integration: piovra cgf vs. wargaming system. Proceedings of the Summer Computer Simulation Conference. Edinburgh, Scotland, pp. 213-218.
8. Chang M.-S. and Hsueh C.-F., (2007). Developing Geographic Information System for Flood Emergency Logistics Planning. Proceedings of the Summer Computer Simulation Conference. San Diego, California, USA, pp. 652-659.
9. Didderen, E., Van Wijngaarden, M., Kobes, M., (2009). Emergency Team Training in Virtual Reality. An Evaluation of the Design Process and of the Performances of NIFV-ADMSTM Dixit V.V. and Radwan E., (2008). Optimal scheduling of evacuation orders for cities. Proceedings of the Summer Computer

Simulation Conference. Edinburgh, Scotland, pp. 194-200. in Training Sessions. Proceedings of the SimTecT Confernce, Adelaide, Australia. ISBN 0 9775257 6 7

10. Dudenhoeffer D., Miller C. and Manic M., (2007). Interdependency Modeling and Emergency Response. Proceedings of the Summer Computer Simulation Conference. San Diego, California, USA, pp. 1230-1237.

11. Ekizoglu B., (2009). Emergency evacuation simulation in Istanbul Technical University. Proceedings of the Summer Computer Simulation Conference. Istanbul, Turkey, pp. 370-376.

12. Filippoupolitis A. and Gelenbe E., (2009). A Decision Support System for Disaster Management in Buildings. Proceedings of the Summer Computer Simulation Conference. Istanbul, Turkey, pp. 141- 147.

13. Filippoupolitis A., Gelenbe E., Gianni D., Hey L., Loukas G., Timotheou S., (2008). Distributed Agent-based Building Evacuation Simulator. Proceedings of the Summer Computer Simulation Conference. Edinburgh, Scotland, pp. 46- 53.

14. Guimarans D., Ramos J. J., Piera M. A., Guasch A., (2006). Proceedings of the Summer Computer Simulation Conference. Calgary, Canada, pp. 110- 115.

15. Han, A. F. 1990. “TEVACS: Decision Support System for Evacuation Planning in Taiwan.” Journal of Transportation Engineering, 116, no. 6: 821-830.

16. Hiramatsu, T. 1983. “Development of a Network Flow Simulator NESSY-IV for Analyses of Mass Evacuation in Case of SCS M&S Magazine – 2010 / n2 (April) Longo Page 7 of 8 Emergency.” Journal of Information Processing, 6, no. 1: 1-9.

17. Hobeika, A. G., and B. Jamei. 1985. “MASSVAC: A Model for Calculating Evacuation Times under Natural Disasters.” In Proceedings of The Conference on Computer Simulation in Emergency Planning, 5, no.1: 23-28.

18. Hobeika, A. G., S. Kim, and R. E. Beckwith. 1994. “A Decision Support System for Developing Evacuation Plans Around Nuclear Power Stations.” Interfaces, 245: 22-35.

19. Jain S., McLean C. R., Lee Y. T., (2007). Towards Standards For Integrated Gaming And Simulation For Incident Management. Proceedings of the Summer Computer Simulation Conference. San Diego, California, USA, pp. 1213-1222.

20. Joshi H., Seker R., Bayrak C., Ramaswamy S., Connelly J., (2007). Ontology for Disaster Mitigation and Planning. Proceedings of the Summer Computer Simulation Conference. San Diego, California, USA.

21. Kanala S., Lephiew R., Ramaswamy S., Seker R., (2008). An Ontology-driven Incident Management System for Scenario Analysis of Disasters. Proceedings of the Summer Computer Simulation Conference. Edinburgh, Scotland, pp. 201-204.

22. Kaup D.J., Clarke T.L., Malone L., Olosn R., (2006). Crowd Dynamics Simulation Research. Proceedings of the Summer Computer Simulation Conference. Calgary, Canada, pp. 365-370.

23. Kelly M., Phillips M. The Application of Live, Virtual and Constructive Simulation to Training for Operations Other Than War. Simulation Industry Association of Australia, available on line at: www.siaa.asn.au/get/2410078921.pdf

24. Kobes M., Oberijé N., Groenewegen K., Morsche T., (2009). Serious Gaming for Behavioural Assessment and Research in Case of Emergency. An Evaluation of Experiments in Virtual Reality. Proceedings of SimTecT, Adelaide, Australia, 15 - 18 June.

25. Lake J.E. and Sanders R.L., (2007). Force On Force Simulation That Provides Facility Stability Analysis. Proceedings of the Summer Computer Simulation Conference. San Diego, California, USA.

26. Lieberman, E. B., and B. J. Andrews. 1980. “TRAFLO: A New Tool To Evaluate Transportation System Management Strategies.” Transportation Research Record, 772: 9-15.

27. Longo F. and Oren T., (2008). Supply Chain Vulnerability and Resilience: A state of the Art Overview. Proceedings of the European Modeling & Simulation Symposium. Campora S.Giovanni (CS), Italy, pp. 527-533.

28. Mclean, M. A., M. Moeller, A. Desrosiers, and T. Urbanik. 1983. “CLEAR: A Model for Calculation of Evacuation Time Estimates in Emergency Planning Zones.” In Proceedings of the Conference on Computer Simulation in Emergency Planning, 112: 58-63.

29. Oleson R. and Kaup D.J., (2006). Implementation of a Crowd Based Social Potential Model. Proceedings of the Summer Computer Simulation Conference. Calgary, Canada, pp. 371- 375.

30. Oren T., Longo F., (2008). Emergence, Anticipation And Multisimulation: Bases For Conflict Simulation. In: Proceedings of the European Modeling & Simulation Symposium. Campora S.Giovanni (CS), Italy, 17-19 September, Vol. I, p. 546 – 555, ISBN: 978-88-903724-0-7.

31. Perumalla K.S. and Beckerman M., (2007). An Analysis Approach to Large-Scale Vehicular Network Simulations. Proceedings of the Summer Computer Simulation Conference. San Diego, California, USA, pp. 1223-1229.

32. Pham H.T.L., Pittman J.L., Court M.C., (2008). A Review of Simulation Modeling Methodologies for Large-Scale Evacuations. Proceedings of the Summer Computer Simulation Conference. Edinburgh, Scotland, pp. 219-226.

33. Pidd, M., F. N. Silva, and R. W. Eglese. 1996. “A Simulation Model for Emergency Evacuation.” European Journal of Operational Research, 90: 413-419.

34. Rathi, A. K. 1994. “A Microcomputer Based Traffic Evacuation Modeling System For Emergency Planning Applications.” Contract No. DE-AC05-84OR21400. SCS M&S Magazine – 2010 / n2 (April) Longo Page 8 of 8 U.S. Department Of Energy, Washington, DC.

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи бакалавра: "Розробка алгоритмічної бізнес-логіки текстових реконструкцій та візуалізацій"

Обсяг пояснювальної записки: 92 аркуші

Дата закінчення дипломної роботи 13 червня 2021р.

Підпис студента _____