Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

CROWD PRESSURE SIMULATION

ANNA GNOIŃSKA, JULIA IGNACYK, MICHAŁ PIENIĄDZ

IMPLEMENTACJA PRZEJŚCIA TŁUMU PRZEZ WĄSKIE GARDŁO W OPARCIU O MODEL SOCIAL FORCE

Kraków 2020

Spis treści

1. Wprowadzenie

1.1. Opis problemu symulacji

Implementacja przejścia tłumu przez wąskie gardło np. bramki. Symulacja będzie wykonana przy pomocy modelu Social Force – model ciągły. Zmianę w ciśnieniu będą obrazowały odpowiednie zmiany koloru w wizualizacji pieszych poddanych dużym siłom (wyliczanym z Social Force). Do przetestowania jest kilka prostych scenariuszy obejmujących barierki czy kolumnę redukującą siły.

1.2. Wstępna analiza problemu

W dzisiejszych czasach modelowanie ruchu pieszych ma duże znaczenie teoretyczne i praktyczne, staje się nieodłączną częścią wielu dziedzin życia takich jak projektowanie budynków mieszkalnych oraz użyteczności publicznej, organizacji imprez masowych, czy też tworzeniu grafiki komputerowej. Ostatnie eksperymenty i ich wyniki z powodzeniem przełożono na równania matematyczne. Ponadto porównanie symulacji komputerowych dużej liczby pieszych z obserwowaną empirycznie dynamiką tłumów doprowadziły do głębszego zrozumienia, w jaki sposób indywidualne interakcje międzyludzkie mają wpływ na różne zjawiska w skali makroskopowej [12].

Nieliniowe interakcje pieszych prowadzą do różnych złożonych zjawisk przestrzenno-czasowych. Pojawienie się nowego, funkcjonalnego lub złożonego kolektywu zachowań w systemach społecznych przyciągnęło wielu naukowców. Wzorce współpracy oraz koordynacji opartych na interakcjach indywidualnych są wynikiem inteligentnych ludzkich działań. Początkowo stosowane metody oceny opierały się na bezpośredniej obserwacji, fotografiach i filmach poklatkowych. Przez długi czas w trakcie badań skupiano się na opracowaniu najefektywniejszych elementów konstrukcyjnych obiektów dla pieszych, wytycznych planowania. Nie były jednak one zbyt dobrze dostosowane do przewidywań przepływu ruchu pieszych w strefach oraz budynkach w trudnych, i stresujących sytuacjach ewakuacyjnych. Z tego powodu zaproponowano szereg różnych modeli symulacyjnych:

queuing models, transition matrix models, stochastic models, które są częściowo ze sobą powiązane. Bardzo dużą popularnością cieszą się modele mikroskopowe oparte na metodach Dynamiki Molekularnej m.in Social Forces Model stworzony przez Helbinga i Molnara [7].

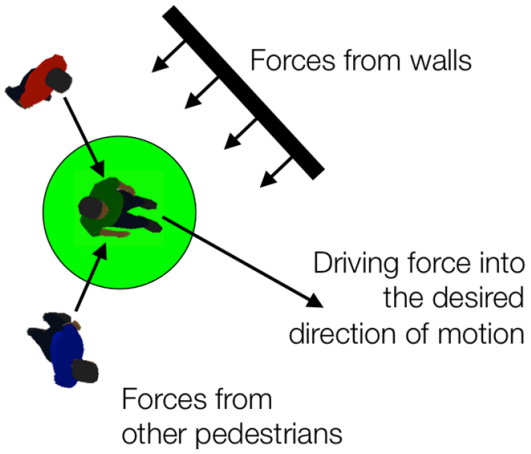
Potencjalnych możliwości rozwiązania problemu przejścia tłumu przez wąskie gardło jest wiele. Szczegółowy opis zagadnienia modelowania ruchu pieszych przedstawiono w kolejnej sekcji wraz z metodami symulowania ruchu wybranego przez nas algorytmu.

2. Przegląd literatury

2.1. Model Social Force

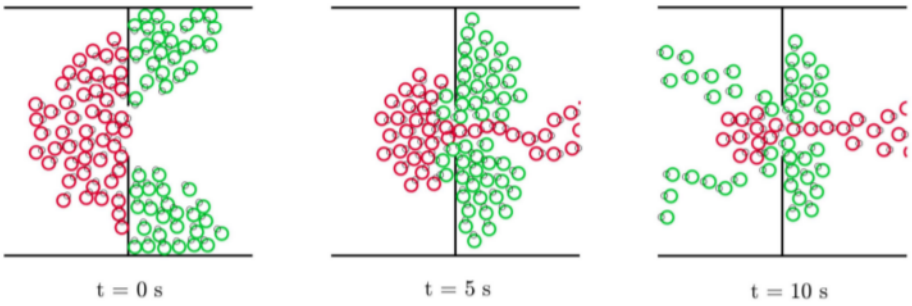
Model Social Force przedstawia koncepcje siły społecznej, która odtwarza większość obserwacji empirycznych w prosty i naturalny sposób. Ma bardzo wiele modyfikacji oraz implementacji. Jednostki są reprezentowane jako punktowe cząstki, na które działają siły wzajemnego odpychania, ale również przyciągania od innych elementów środowiska, gdzie wszystkie składowe podlegają dynamice tłumu. Jest to odwołanie do fizycznej analogii. We wspomnianym modelu zakładamy, że ludzie poruszają się izotropicznie, co nie jest zgodne z rzeczywistością. Ludzkie zachowanie często wydaje się „chaotyczne", nieregularne i nieprzewidywalne. Empiryczne dowody udowadniają, że każdy z nas ma preferowany kierunek ruchu, w którym cały czas podąża. Ruchy boczne można zaobserwować tylko w specyficznych sytuacjach takich jak poszukiwanie drogi czy omijania napotkanych przeszkód [5].

Należy pamiętać, że model Social Force jest jednym z ogromnej klasy modeli ruchu pieszych bazujących na sile. Pierwsze aplikacje SFM skupiały się głównie na symulacji ewakuacji z budynków. Celem takich aplikacji jest jak najszybsze dojście do wyjścia przez pieszych. Model Social Force jest modelem ciągłym. Zakłada, że indywidualne jednostki są podmiotem fizycznych i socjalnych sił. Można rozszerzyć go o różne typy zachowań, takie jak dynamiczny wybór trasy w otwartej przestrzeni, pieszych wolących chodzić w grupach lub mających inne preferencje poruszania się, wyprzedzanie, czekanie, zwalnianie, unikanie kolizji i podążanie za liderem. Takiego typu modelu nie stosujemy jednak do symulacji komputerowych na dużą skalę - złożoność obliczeniowa zwiększa się kwadratowo wraz ze wzrostem liczby jednostek. Zdecydowaną zaletą tej metodologii jest duża dokładność symulacji, precyzja odwzorowania mikroskopowych oddziaływań. Do wad należy zaliczyć małą wydajność obliczeniową oraz trudność w przypadku odwzorowań niektórych sytuacji. Prace nad udoskonalaniem algorytmów wciąż trwają, przeszkodą są wysokie nakłady obliczeniowe, które sprawiają że metoda ma ograniczone możliwości rozwoju[6].



Rysunek 2.1: Zobrazowanie sił wpływających na jednostkę fizyczną [6]

„Social Force Model" zakłada, że przyspieszenie, opóźnienie i zmiany kierunkowe pieszych mogą być przybliżane przez sumę różnych sił, z których każda wychwytuje inny efekt interakcji. Na przykład dostosowanie rzeczywistej prędkości pieszego do pożądanej prędkości i preferowanego kierunku ruchu pieszego można modelować za pomocą prostej „siły napędowej", opisującej stopniowe dostosowanie prędkości w typowym okresie czasu. Co więcej, chęć uniknięcia kolizji i poszanowania pewnego „terytorium" wokół innych znajduje odzwierciedlenie w odpychających siłach interakcji między pieszymi, których siła wykładniczo maleje wraz z odległością. Odpychające interakcje ze ścianami lub ulicami mogą być wychwytywane przez podobne siły[10].



Rysunek 2.2: Przykładowa symulacja przejścia ludzi w dwóch kierunkach [4]

W tym przypadku jest to symulacja drzwi pociągu. Piesi oznaczeni na czerwono próbują wysiąść z pociągu, podczas gdy piesi oznaczeni na zieloną chcą do niego wsiąść.

Model Social Force można łączyć z programowaniem agentowym (Agent-based

model) oraz automatami komórkowymi (Cellular Automata). Jednak klasycznego przypadku automatu komórkowego tj. synchronicznego i homogeniczego nie wykorzystamy do tworzenia symulacji systemów złożonych. Dopiero przy zastosowaniu automatów niehomogenicznych przy uwzględnieniu różnych typów komórek (np. przestrzeń ruchu, ściany, przeszkody, drzwi) czy zróżnicowanie funkcji przejścia na siatce automatu, a także asynchronicznych (ustalenie kolejności synchronizacji stanów komórek automatu), możemy stworzyć bardziej skomplikowane i zaawansowane symulacje zachowania pieszych. W praktyce możemy zastosować trzy rodzaje automatów:

– automaty klasyczne (Cellular Automata), w których funkcja przejścia uwzględnia wyłącznie relacje lokalne

– globalne automaty komórkowe (Global Cellular Automata) , w których funkcja przejścia uwzględnia relacje globalne, obejmujące obszar całej siatki

– „rozszerzone" automaty komórkowe (Extended Cellular Automata), w których funkcja przejścia obejmuje relacje lokalne oraz wybrane komórki z całej siatki [12].

Systemy agentowe pozwalają na odwzorowanie bardzo złożonych interakcji. Dla modelu Social Force jednostka ma charakter makroskopowy. Każda osoba jest wirtualnym agentem, który posiada indywidualny zestaw zmiennych. Otwartość modelu i możliwość definiowania nowych zachowań pozwala definiować zestawy zachowań dla różnych scenariuszy, które są związane z pokazaniem rzeczywistego sposobu przemieszania się osób (m.in. omijanie przeszkód, szukaniem najlepszych dróg, zachowanie określonej prędkości).

Definiowanie własnych zestawów reguł pozwala na symulację osób, które mają różne role w tłumie (np. osoba uciekająca z płonącego budynku, strażak, pojedyncza osoba, rodzice z dziećmi). W modelu wykorzystano własną interpretację zagadnienia stref personalnych, według której w zależności od zagęszczenia zmieniają się całe zestawy zachowań. Zdefiniowane strefy mają charakter ekspercki i mogą być dowolnie zmieniane w celu uzyskania pożądanych charakterystyk bazowych związanych z ruchem[8]. Niesamowite jest to, że symulacje komputerowe tego modelu, pomimo swojej prostoty, pasują do wielu obserwowanych empirycznie zjawisk zaskakująco dobrze.

3. Proponowany model

3.1. Cele modelu

Naszym celem jest zbudowanie aplikacji, która pozwoliłabym użytkownikowi przeglądać różne scenariusze modelu Social Force - przejścia tłumu ludzi przez cienkie gardło.

Opracowanie modelu poznawczego zachowania pieszych wymaga rozwiązania dwóch zasadniczych pytań: Jakiego rodzaju informacji używa pieszy ? I w jaki sposób przetwarzane są te informacje w celu dostosowania zachowania podczas chodzenia ?

W odniesieniu do pierwszego pytania badania wykazały, że wzrok jest głównym źródłem informacji wykorzystywanych przez pieszych do kontrolowania ich ruchu. W związku z tym zaczynamy od przedstawienia informacji wizualnej pieszych.

Aby odpowiedzieć na drugie pytanie, proponujemy dwie heurystyki oparte na tych informacjach wizualnych, które określają pożądane kierunki chodzenia α_{des} i pożądane prędkości chodzenia v_{des} pieszych. Na koniec zakładamy, że piesi stale dostosowują swoje obecne zachowanie podczas chodzenia, aby dopasować je do pożądanego zachowania z czasem relaksacji τ.

Zdefiniowanie wejść i wyjść z modelu dla naszego algorytmu:

Wejście

1. Piesi:
   - cel
   - prędkość normalna
   - masa
   - maksymalny kąt widzenia
   - zasięg wzroku
   - czas relaksacji
   - pozycja startowa

2. Przeszkody

   – umiejscowienie przeszkód

Wyjście

1. wizualizacja przebiegu przechodzenia pieszych
2. wizualizacja tworzenia i rozładowywania się tłumu
3. ilość kroków przejścia pieszych do celu
4. czas przejścia pieszych do celu
5. odczyt największej siły nacisku na pieszego w czasie przyjścia

3.2. Ruch pieszych

W naszym modelu każdy pieszy i charakteryzuje się aktualną pozycją $\vec{x}_i$ i prędkością $\vec{v}_i$. Każdy pieszy dodatkowo charakteryzuje się wygodną prędkością chodzenia v_i^0 i punktem docelowym, czyli miejscem do którego chce dotrzeć, takim jak drzwi wyjściowe pokoju lub koniec korytarza. Wreszcie pole widzenia pieszego waha się w lewo i w prawo o ϕ° w stosunku do linii wzroku .

Wcześniejsze badania wykazały, że osoby chodzące mogą oszacować czas zderzenia z przeszkodami otaczającymi za pomocą specjalistycznych mechanizmów neuronowych na poziomie siatkówki i mózgu. Dla wszystkich możliwych kierunków $\alpha[-\phi, \phi]$ obliczamy odległość do pierwszego zderzenia $f(\alpha)$, jeśli pieszy porusza się w kierunku α z prędkością v_i^0 , biorąc pod uwagę prędkości marszu innych pieszych i rozmiary ciała. Jeżeli nie przewiduje się wystąpienia kolizji w danym kierunku to ustawiona na wartość domyślną wartość maksymalna d_{max}, który reprezentuje „odległość w poziomie” pieszego.

– Funkcja oceniająca odległość przeszkody od pieszego, zależnie od kierunku (brak przeszkód równoznaczny ze zwróceniem d_{max}):

$$f(\alpha)$$

– Obliczanie pożądanego kierunku

Pieszy wybiera kierunek α_{des}, który umożliwia najbardziej bezpośrednią drogę do punktu docelowego, biorąc pod uwagę obecność przeszkód. Wybrany kierunek $\alpha_{\mathrm{des}}(t)$ oblicza się poprzez minimalizację odległości$d(\alpha)$ do miejsca docelowego:

$$d_{\max} = H_i$$
$$d(\alpha) = d_{\max}^2 + f(\alpha)^2 - 2 \cdot d_{\max} \cdot f(\alpha) \cdot \cos(\alpha_0 - \alpha)$$
$$\alpha_{des} = \min(d(\alpha))$$

– Pożądana prędkość: Pieszy zachowuje odległość od pierwszej przeszkody w wybranym kierunku chodzenia, co zapewnia czas na zderzenie co najmniej τ. Obliczamy ze wzoru:

$$v_{des}(t) = \min\left(v_{0i}, \frac{d_h}{\tau}\right)$$

Gdzie:

d_h- odległość między pieszym,a pierwszą przeszkodą w pożądanym kierunku α_{des} w czasie t.

v_i^0 - wygodna prędkość chodzenia

– Przyspieszenie naszego pieszego:

$$\frac{dv_i}{d_t} = \frac{(v_{des} - v_i)}{\tau}$$

Gdzie:

$\vec{v_i}$ - prędkość rzeczywista pieszego

τ - czas relaksacji pieszego

NAZWA	ZMIENNA
Pieszy	i
Ściana	W
Masa	m_i[kg]
Promień	$r_i = \frac{m_i}{320}\ [m]$
Pozycja	$x_i\ (wektor)$
Prędkość normalna	$v_{0i}\ [\frac{m}{s}]$
Prędkość	$v_i(wektor)$
Punkt docelowy	$O_i(wektor)$
Max. Kąt widzenia	$\emptyset\ [rad]$
Czas relaksacji	$\tau[s]$
Kierunek	$\alpha\epsilon[-\emptyset, \emptyset]$ [rad]
Kierunek docelowy	α_0[rad]
Zasięg wzroku	$H_i\ [m]$
Pożądany kierunek	α_{des}(wektor)
Pożądana prędkość	$v_{\mathrm{deg}}(wektor)$
Max. Widziana odległość	$d_{\max}[m]$
Odległość między pieszym a przeszkodą w kierunku	$d_h\ [m]$
Czas	t

Rysunek 3.3: Ogólne zależności wykorzystywane w naszej symulacji

3.3. Efekt kolizji pieszych

W przypadkach przeludnienia mogą wystąpić fizyczne interakcje między ciałami, powodując niezamierzone ruchy, których nie określa powyższa kalkulacja. Rzeczywiście, przy ekstremalnych gęstościach konieczne jest rozróżnienie między celowym unikaniem przez pieszych dostosowujących ich ruch zgodnie z postrzeganymi wskazówkami wizualnymi a niezamierzonymi ruchami wynikającymi z sił interakcji spowodowanych zderzeniem z innymi ciałami. Dlatego rozszerzyliśmy powyższy opis, uwzględniając fizyczne siły kontaktowe.

– Jeżeli piesi się nie dotykają:

$$g(x) = 0$$

– Jeśli się dotykają:

$$g(x) = x$$

– Siły kontaktu między pieszymi:

$$f_{ij} = k \cdot g\left(r_i + r_j - d_{ij}\right) \cdot n_{ij}$$

Gdzie:
n_{ij} - znormalizowany wektor od pieszego j do pieszego i
d_{ij} - odległość między środkami mas pieszych

– Siły kontaktu między pieszym a przeszkodą (ścianą):

$$f_{iW} = k \cdot g\left(r_i + d_{iW}\right) \cdot n_{iW}$$

Gdzie:
W - ściana
d_{iW} - odległość od ściany
n_{iW} - kierunek prostopadły do ściany

– Ostateczne przyspieszenie:

$$\frac{d_{vi}}{d_t} i = \frac{(v_{des} - v_i)}{\tau} + \sum_{j, j \neq i} \frac{f_{ij}}{m_i} + \sum_W \frac{f_{iW}}{m_i} + \varepsilon$$

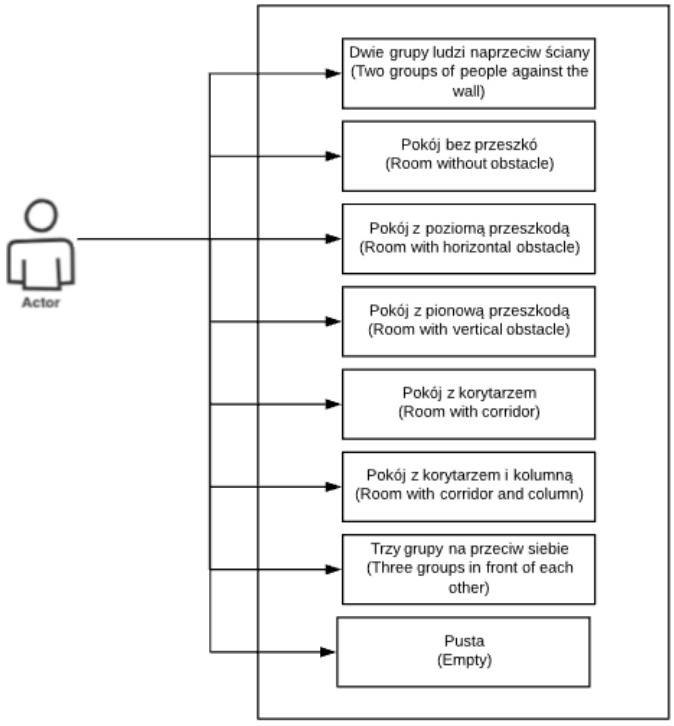
NAZWA	ZMIENNA
Odległość pomiędzy środkami mas pieszych	$d_{ij}[m]$
Odległość pomiędzy środkiem masy pieszego a ścianą	d_{iw}[m]
Znormalizowany wektor pomiędzy pieszymi	$n_{ij}\ (wektor)$
Kierunek prostopadły do ściany	$n_{iw}(wektor)$
Współczynnik skalujący	k
Siła fluktuacji	$(random)$

Rysunek 3.4: Zmienne i zależności w przypadku kolizji

3.4. Stosowane podejście w nawiązaniu do opisu literaturowego

Wykorzystany przez nas model jest zmodyfikowaną wersją modelu przedstawionego w artykule"How simple rules determine pedestrian behavior and crowd disasters" autorstwa: Mehdi Moussaïd, Dirk Helbinga i Guy Theraulaza.

3.5. Diagram przypadków użycia



Rysunek 3.5: Możliwe scenariusze symulacji

4. Symulacja zjawiska - implementacja

4.1. Wybór technologii i narzędzi

Podczas wykonywania projektu istotny był wybór technologii, środowiska, repozytorium. Po dogłębnej analizie zdecydowaliśmy się na konkretne rozwiązania.

4.1.1. Wybór języka - JAVA

– Jest bardzo wiele przydatnych bibliotek i gotowych przykładowych implementacji aplikacji

– Bardzo popularny język, co niesie za sobą łatwość rozwiązywania błędów i problemów.

– Daje możliwość tworzenia aplikacji z interfejsem graficznym

– Zawiera dużą ilość potrzebnych gotowych bibliotek do stworzenia aplikacji.

JavaFx - otrzymujemy możliwość definiowania widoku aplikacji w języku XML (dużo łatwiejsza edycja, szybkie tworzenie kodu), wspiera rozdzielenie ról w zespole. Narzędzie Scene Builder pozwala na eksperymentowanie, bez obaw, że coś się zepsuje. W prosty sposób można modyfikować wygląd kontrolek, co jest dodatkową wygodą. JavaFX to również dużo lepsze wsparcie dla odtwarzania mediów, bez problemu odtworzymy w niej pliki MP3.

4.1.2. Środowisko INTELLIJ

– Mocno stawia na produktywność i pracę bez użycia myszy, dużo skrótów klawiaturowych.

– Automatyczne podpowiedzi, które oferuje to środowisko naprawdę potrafią zaskoczyć w bardzo pozytywnym stopniu.

– Większość funkcjonalności wbudowana w platformę. Nie ma potrzeby instalowania dodatkowych pluginów

4.1.3. Serwis hostingowy - BitBucket

– Pozwala na lepsze zrozumienie zmian repozytorium dzięki podglądom kodu wyświetlającym diffy zunifikowane lub side by side

– Może pokazać wyniki kompilacji z naszego systemu CI. Pojedyncza pozytywna lub negatywna ikona powie jaka jest kondycja kodu.

– Przyjazny i przejrzysty design.

– Pozwala ściągać wnioski i recenzje kodu.

– Darmowy dla zespołu do 5 osób.

4.1.4. SourceTree, czyli dodatkowa pomoc

– SurceTree to świetny darmowy program, z poziomu którego możemy połączyć się z własnym kontem na BitBucket lub GitHub. Samodzielnie śledzi wszelkie zmiany w plikach. Po zakończeniu pracy jednym przyciskiem możemy wysłać wszelkie modyfikacje do chmury wraz z komentarzami.

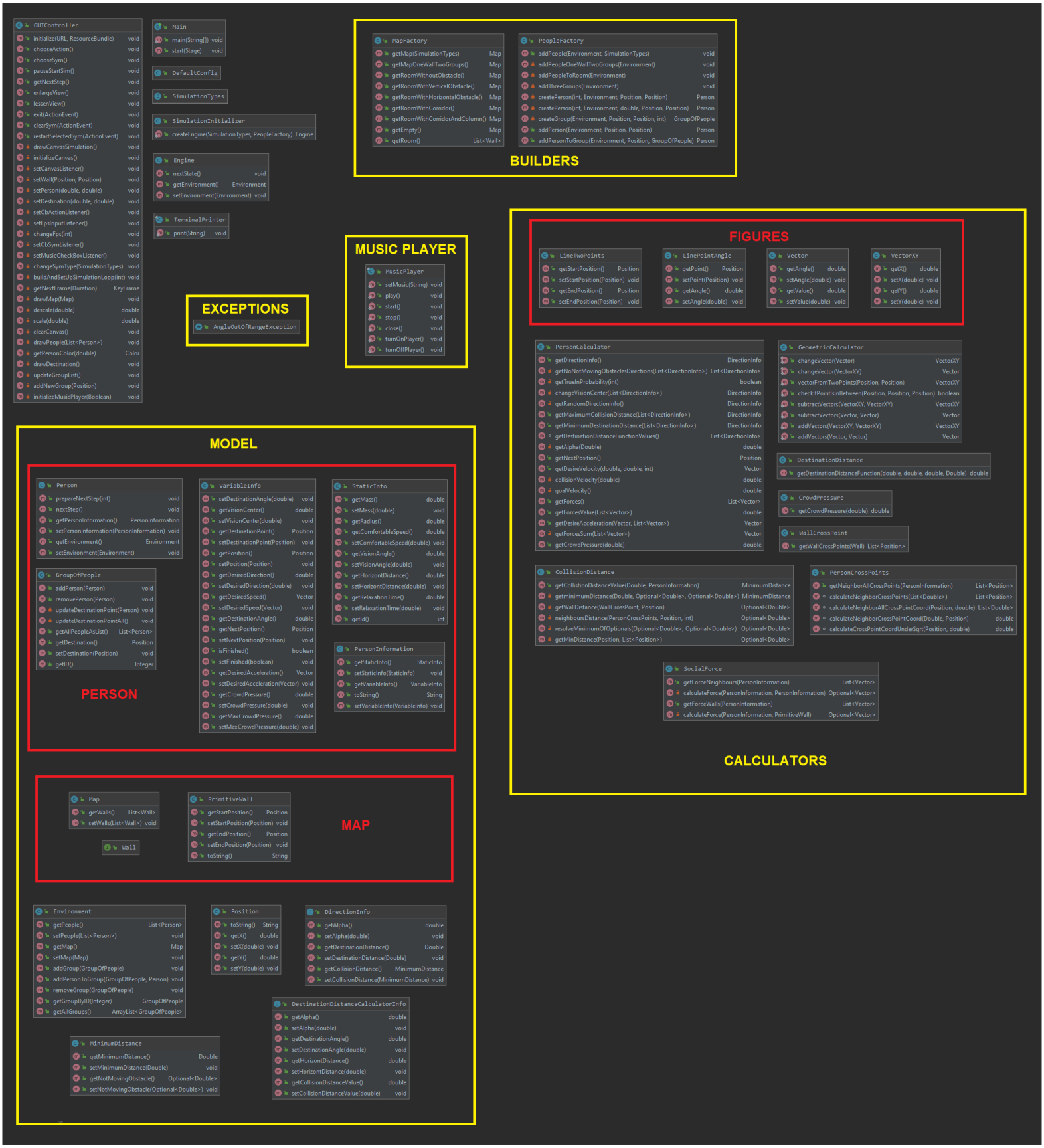
– Największą zaletą SourceTree jest jego graficzny interfejs, który każdemu użytkownikowi niezależnie od stopnia zaawansowania pozwoli w prosty i przystępny sposób odnaleźć się w świecie GIT.

– System kontroli wersji pozwala na sprawne zarządzanie kodem źródłowym projektów. Nie trzeba się już martwić o to, czy ktoś wcześniej dokonał jakichkolwiek modyfikacji w kodzie bez naszej wiedzy.

4.1.5. Testy jednostkowe

Ze względu na wybór Javy jako języka, z którym pracujemy, testy jednostkowe zostały przeprowadzone przy użyciu JUnit.

4.2. Diagramy klas UML



Rysunek 4.2: Diagram klas UML

5. Wyniki symulacji

5.1. Statystyki

W fazie testowania aplikacji wykonaliśmy ponad 100 opcji różnych symulacji. Zmienialiśmy parametry pieszych, budowaliśmy rożne mapy, dodawaliśmy lub usuwaliśmy przeszkody. Podczas testów poprawiliśmy algorytm, aby wyświetlał jak najdokładniejsze wyniki. Uważamy, że aplikacja działa dobrze, ale dokładność wyliczania siły maksymalnej działającej na pieszego oraz szukanie najkrótszej drogi dla pieszego można poprawić.

5.2. Zastosowane procedury kalibracji i walidacji

W fazie rozwoju naszego projektu testowaliśmy różne ustawienia konfiguracji. Najbardziej znacząca dla poprawnego zobrazowania problemu były ustawienia pieszego - przede wszystkim masa, zakres widzenia i czas reakcji pieszego. Na ruch ludzi wpływały również przeszkody oraz inni piesi. W czasie procedur walidacji skupiliśmy się na metodzie jakościowej - wizualnej.

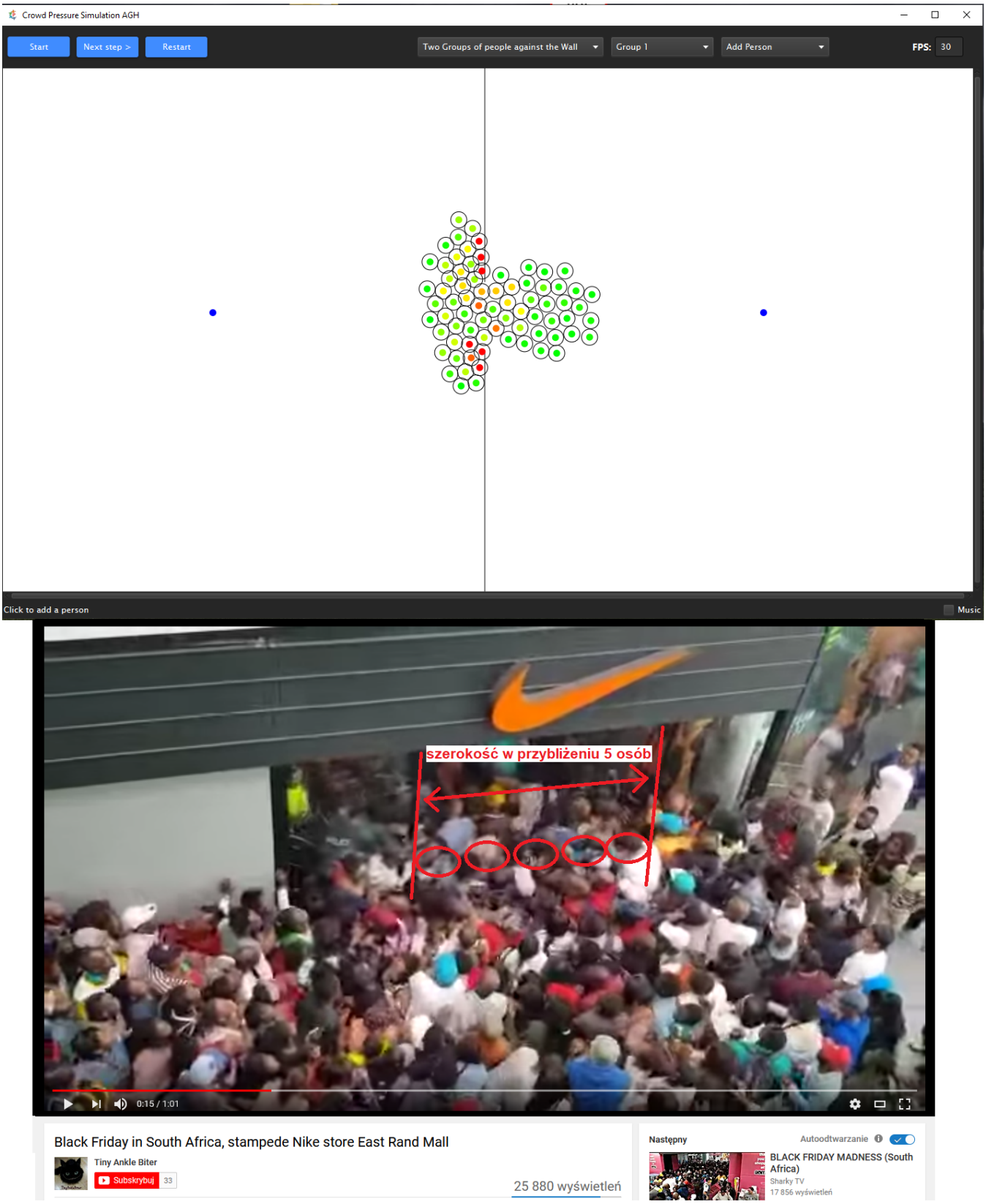
Najpierw przetestowaliśmy model w kontekście prostych sytuacji interakcji, w których dwóch pieszych unikało się (pusta mapa z dwoma pieszymi bez przeszkód). W serii eksperymentów śledziliśmy ruch pieszego mijając innego poruszającego się w przeciwnym kierunku. Model przewiduje indywidualne trajektorie unikania, które bardzo dobrze zgadzają się z trajektoriami obserwowanymi eksperymentalnie przedstawionymi w książce “Experimental study of the behavioural mechanisms underlying self-organization in human crowd” przez M. Moussaïda.

Następnie zbadaliśmy prognozy modelu w kontekście zbiorowym. W przypadku ruchu dwukierunkowego na ulicy (dwie grupy idące naprzeciw siebie), zakładającego losowe początkowe pozycje pieszych, widzimy, że kierunki przepływu rozdzielają się spontanicznie po krótkim czasie, jak zaobserwowano empirycznie. Ta zbiorowa organizacja odzwierciedla dobrze znane zjawisko formowania linii, które jest charakterystyczną właściwością dynamiki tłumu. Naszą sytuację porównaliśmy z danymi opisanymi w książce Roberta A. Meyersa “Encyclopedia of

Complexity and Systems Science ” w rozdziale *Dynamika ewakuacji: wyniki empiryczne, modelowanie i zastosowania*.

Zbadaliśmy również wpływ zagęszczenia pieszych na przepływy jednokierunkowe. Przewidywana przez model relacja prędkości do gęstości dobrze zgadza się z danymi empirycznymi opisanymi w dziele S.J. Oldera "Movement of Pedestrians on Footways in Shopping Streets ” rozdział *Traffic Eng Control*. Ponadto, gdy gęstość przekracza wartości krytyczne, nasz model pokazuje przejścia od płynnych przepływów do fal stop-and-go i „turbulencji tłumu”, jak zaobserwowano przed katastrofami tłumu opisanymi w artykule “Dynamics of crowd disasters: An empirical study Phys Rev” D. Helbinga, A. Johanssona oraz HZ Al-Abideena.

Naszą symulację zweryfikowaliśmy również porównując ją z rzeczywistym przepływem osób przez przeszkodę - na podstawie filmów [1] [2]. W aplikacji, którą stworzyliśmy w ciągu 3 sekund przez wąskie gardło (o szerokości dającej możliwość jednoczesnego przedostania się 5 osób) przeszło około 30/35 osób z odświeżaniem 30 fps. Do porównania w przedstawionym filmie w takim samym czasie (z wykorzystaniem zwolnionego tempa odtwarzania 0.25) udało się nam policzyć około 30/40 osób, które przeszły przez drzwi.[1] Należy wziąć pod uwagę, iż film jest słabej jakości, a ludzie przemieszczają się bardzo szybko i chaotycznie. Jak widać piesi różnie zachowują się w zależności od rzeczywistej sytuacji - są przypadki, kiedy mimo ścisku poruszają się dosyć sprawnie, a czasem wręcz odwrotnie. Ma na to wpływ wiele czynników tj. panująca atmosfera, sytuacja stresowa, agresja pojedynczych osób, których nie uwzględniamy. Inaczej przedstawi się sytuacja ewakuacji budynku, w którym jest pożar, a inaczej wyścig po promocje w sklepie. Nasza symulacja w sytuacjach “uśrednionych ” odpowiada rzeczywistości. Wybór filmu był uzależniony bliskim odwzorowaniem sytuacji z symulacji, która bardzo dobrze odzwierciedla badany przez nas model.



Rysunek 5.2: Porównanie sytuacja rzeczywista, a nasza symulacja

Po kalibracji oraz walidacji modelu wartości atrybutów konfiguracyjnych symulację przedstawiają się w następujący sposób:

Atrybuty klasy *Configuration* pieszy:

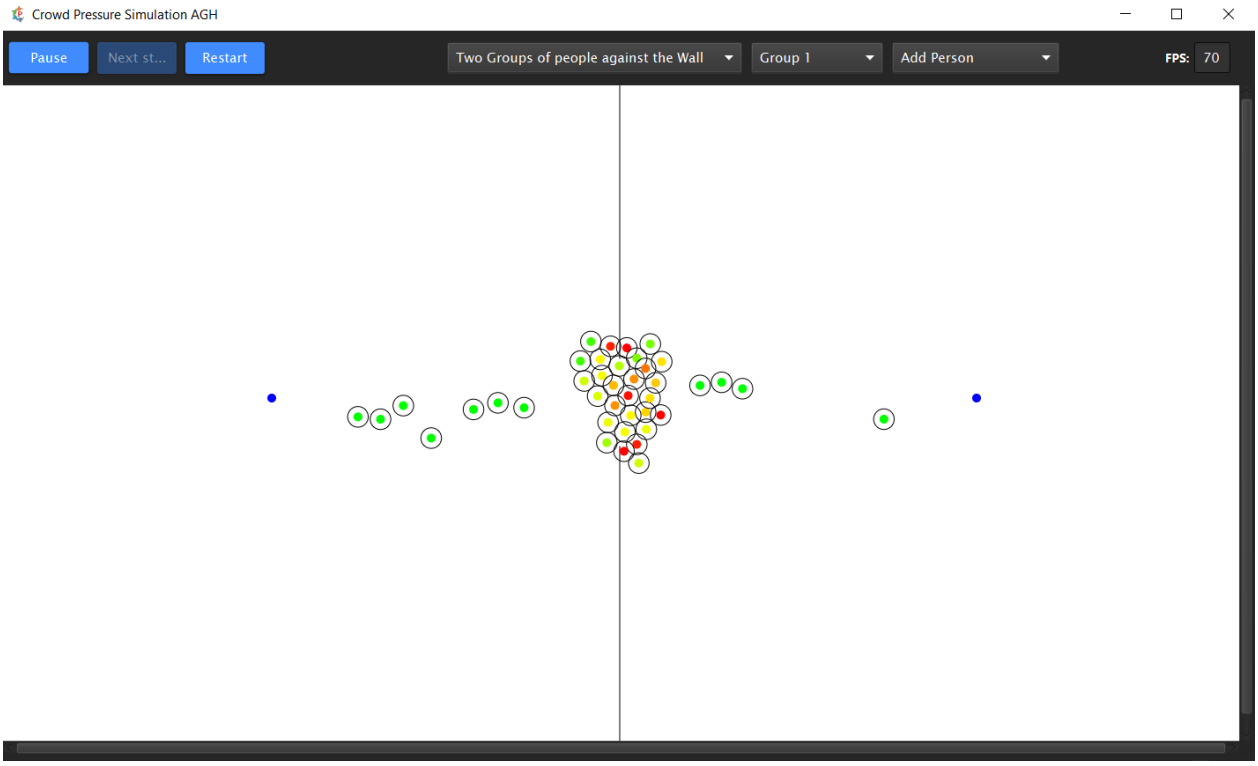
- double $DEFAULT\_PERSON\_COMFORTABLE\_SPEED$ = 0.14
- double $DEFAULT\_PERSON\_VISION\_ANGLE$ = 0.38
- double $DEFAULT\_PERSON\_HORIZON\_DISTANCE$ = 40
- double $DEFAULT\_PERSON\_RELAXATION\_TIME$ = 1
- double $DEFAULT\_PERSON\_MASS$ = 360.0
- double $MASS\_RADIUS\_RATIO$ =320.

Atrybuty klasy *Configuration* ogólne:

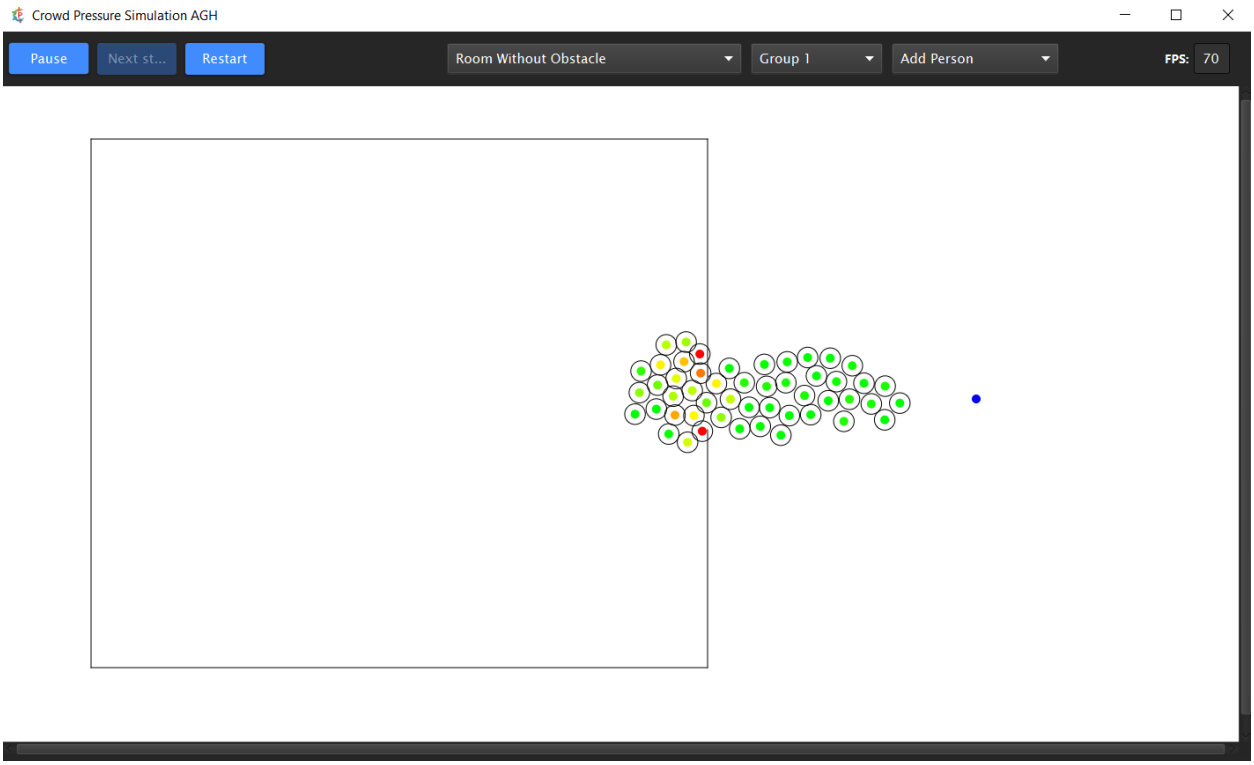
- $Function < Destination, Distance, CalculatorInfo, Double >$ $DESTINATION\_DISTANCE\_FUNCTION$ = DestinationDistance.originalDDFunction - funkcja określona w klasie DestinationDistance
- boolean $FORCES$ = true
- double $MAX\_DISTANCE\_TO\_GOAL$ = 1.8
- double $K\_PARAMETER$ = $3 * 10^3$
- double $PRECISION\_OF\_CALCULATIONS$ = 0.001
- int $PERCENT\_PROBABILITY\_OF\_CHANGING\_VISION\_CENTER$ = 5
- boolean $CHANGE\_VISION\_CENTER\_RANDOM$ = true
- int $PERCENT\_PROBABILITY\_OF\_COMPUTE\_NOT\_FOR\_WALLS$ = 5
- double $DESTINATION\_ANGLE\_RANGE$ = 0.2
- double $WALL\_DISTANCE\_TO\_CHANGE\_DIRECTION$ = 4.0

5.3. Poszczególne scenariusze symulacji

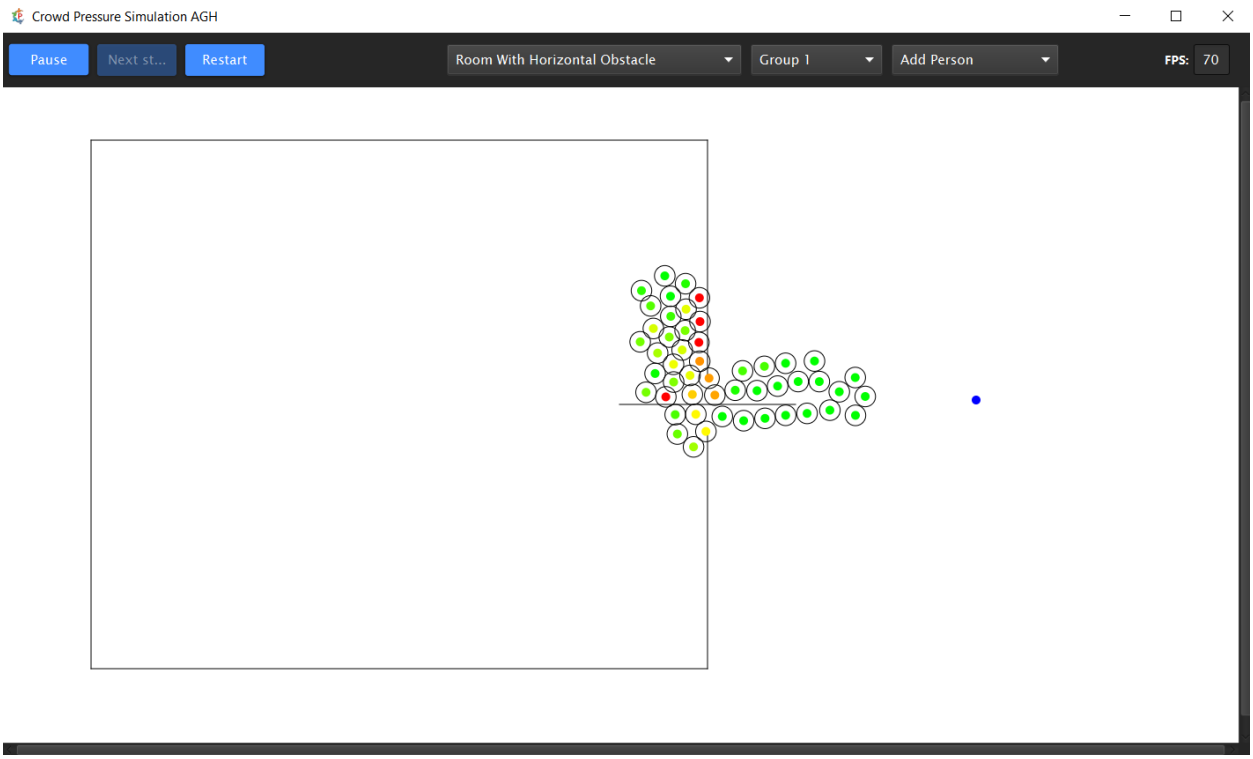
Przykłady symulacji, które zawarliśmy w naszej aplikacji przedstawiają różne sytuacje, z którymi spotykamy się w życiu codziennym. Nasza aplikacja daje możliwość wybrania siedmiu gotowych map, które opisują najczęstsze sytuacje, z którymi mamy do czynienia. Dodatkowo dodaliśmy $EMPTY\_MAP$, aby każdy użytkownik mógł sam zasymulować interesującą go sytuację. Istnieje możliwość określenia $Destination$ - miejsca docelowego, do którego zmierzają piesi, wielu grup pieszych ($Add\ Group$, $Add\ Another\ Group$), decydując o ich ilości osób w każdej z nich ($Add\ Person$). W dodatku mamy możliwość budowania pomieszczeń, przeszkód - opcja $Add\ Wall$.



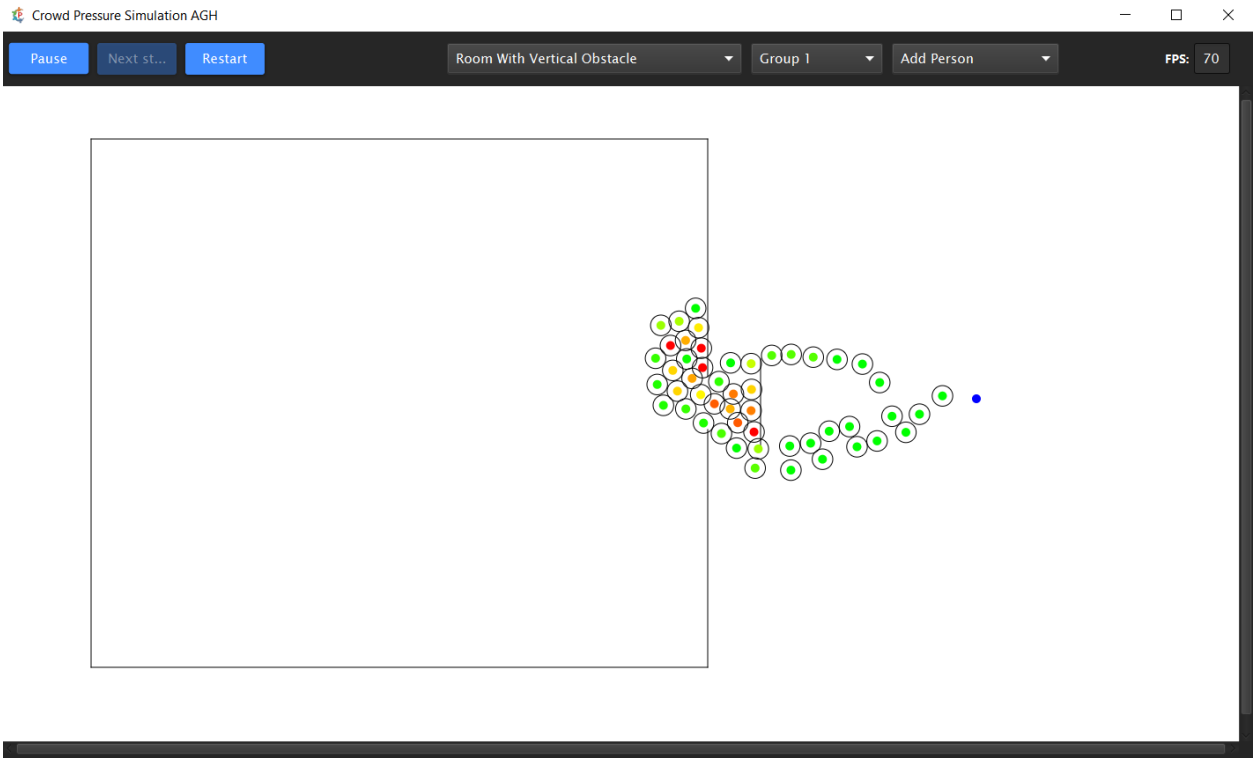
Rysunek 5.3: SYM_ONE_WALL_TWO_GROUPS



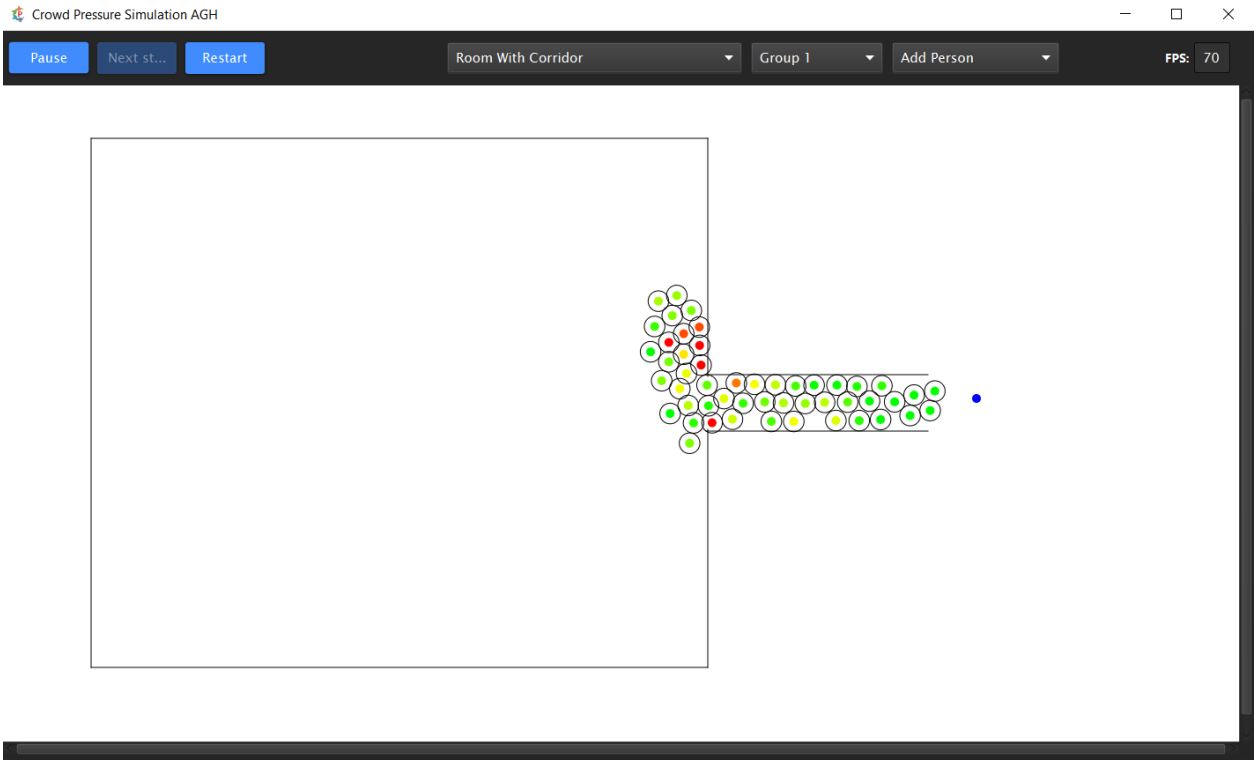
Rysunek 5.3: SYM_ROOM_WITHOUT_OBSTACLE



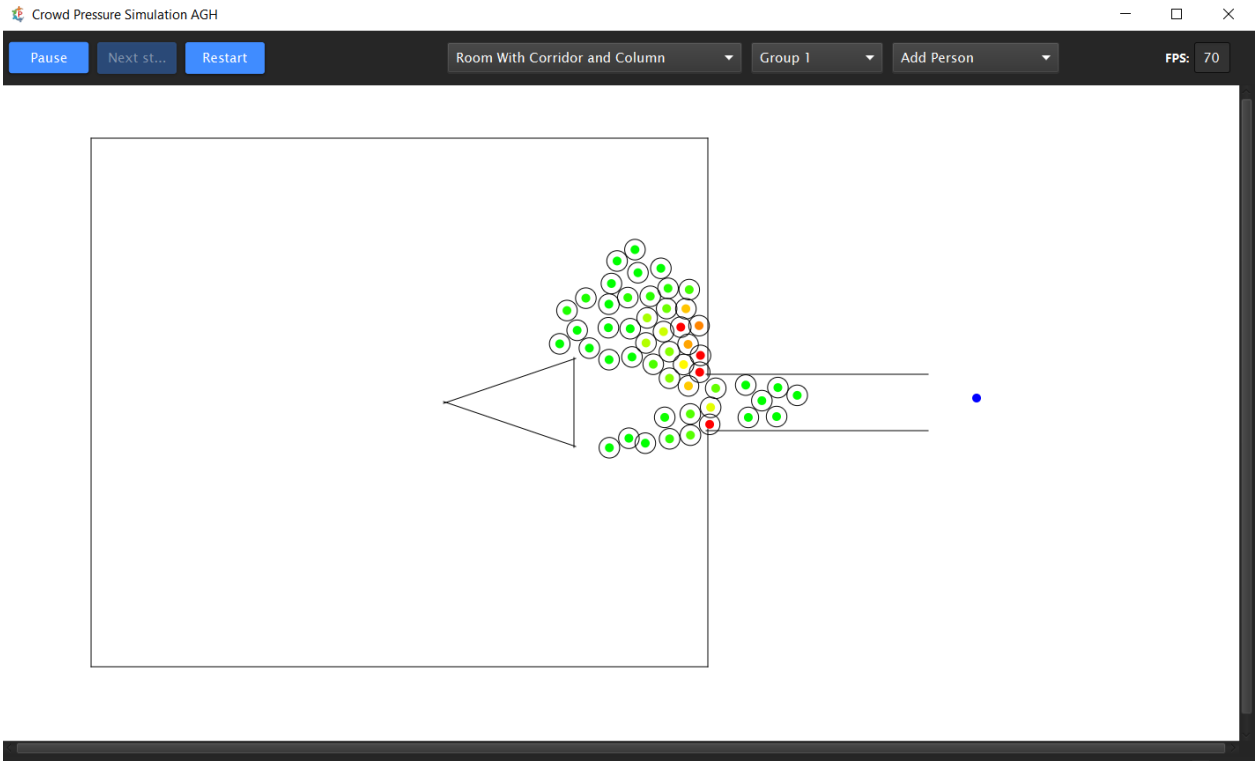
Rysunek 5.3: SYM_ROOM_WITH_HORIZONTAL_OBSTACLE



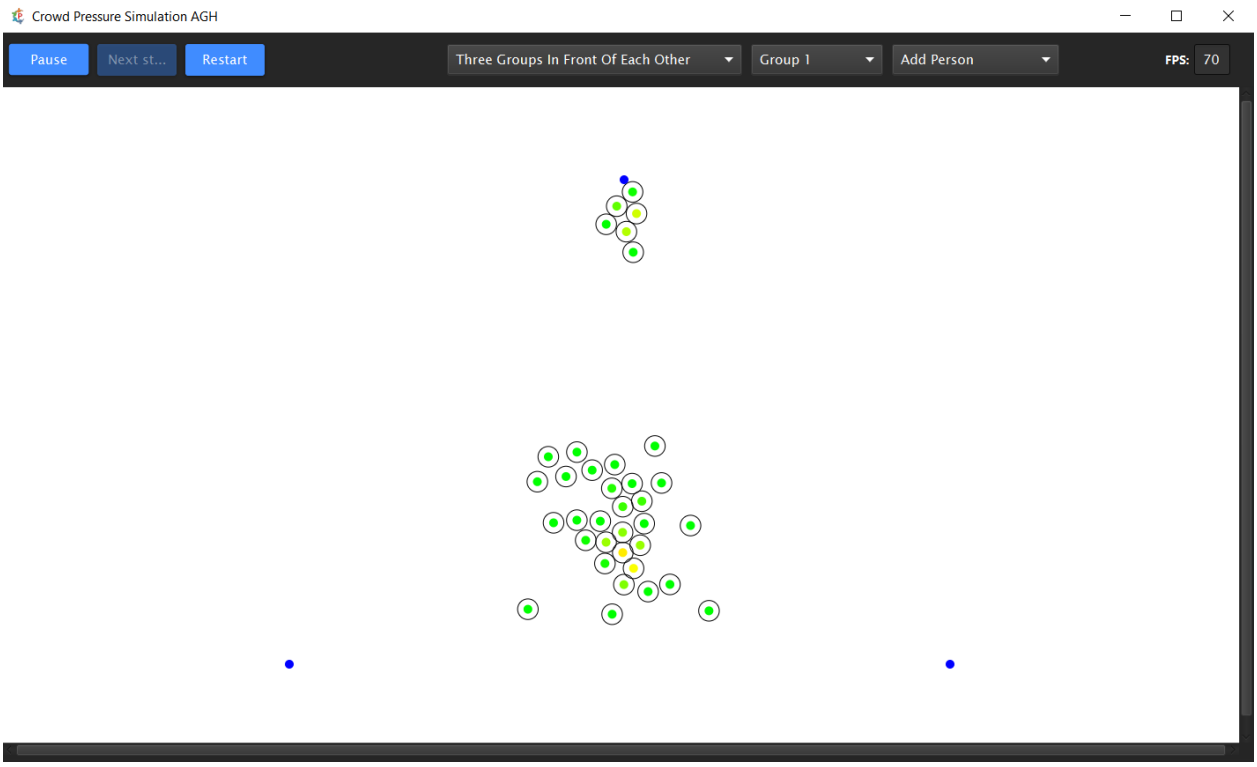
Rysunek 5.3: SYM_ROOM_WITH_VERTICAL_OBSTACLE



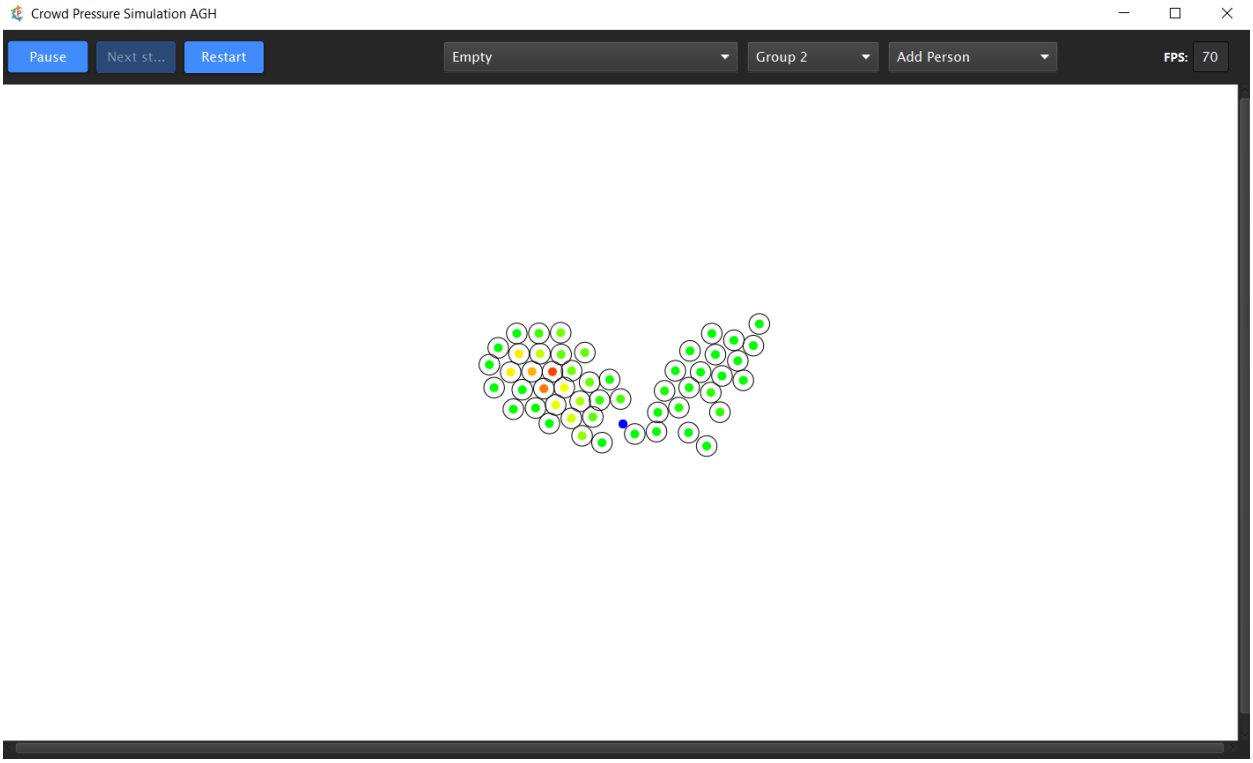
Rysunek 5.3: SYM_ROOM_WITH_CORRIDOR



Rysunek 5.3: SYM_ROOM_WITH_CORRIDOR_AND_COLUMN



Rysunek 5.3: SYM_THREE_GROUPS



Rysunek 5.3: SYM_EMPTY

6. Wnioski

6.1. Najważniejsze wnioski

Stworzone przykłady symulacji przejścia tłumu przedstawiają różne sytuacje z którymi możemy spotkać się w życiu codziennym. $SYM\_ONE\_WALL\_TWO\_GROUPS$ przedstawia przejście dwóch grup ludzi przez wąskie przejście. Jedna grupa z prawej strony gardła stara się przejść na lewą, a druga na odwrót. Grupy klinują się na dłuższy czas w przejściu.

$SYM\_ROOM\_WITHOUT\_OBSTACLE$ - przedstawia pomieszczenie, z którego wychodzą piesi przez wąskie przejście. W tym przypadku nie umieściliśmy przeszkód. Grupy nawzajem zatrzymują się ocierając się o siebie. Mamy również sytuacje związane z poziomą i pionową przeszkodą (w rzeczywistości może to być bramka). W symulacji $SYM\_ROOM\_WITH\_VERTICAL\_OBSTACLE$ widać, że "pierwsi", którzy przeszli przez wąskie gardło są zatrzymywani przez równoległy do pokoju obiekt. Na tym korzystają następni, którzy poruszają się za nimi i łatwiej omijają przeszkodę. W sytuacji $SYM\_ROOM\_WITH\_CORRIDOR\_AND\_COLUMN$ występuje kolumna, która redukuję siłę. Dzięki temu grupa pieszych nie korkuje się przy wąskim przejściu.

W wykonanym przez nas projekcie można zauważyć jak ruch ludzi, przeszkody oraz otoczenie wpływają na zachowanie jednostki. W każdej z sytuacji widać duży nacisk pieszych na siebie w okolicach wyjścia - co spowalnia przejście grupy do celu. Piesi wzajemnie wywierają na siebie siły, które wzmacniają lub stopują ich ruch, co obrazują zmieniające się kolory na każdym z tłumu. W pewnych przypadkach wzajemne oddziaływanie różnych czynników powoduje całkowite zablokowanie ruchu.

6.2. Usytuowanie modelu i symulacji na tle istniejących rozwiązań

Nasza aplikacja nie wprowadza żadnych rewolucyjnych rozwiązań w porównaniu z tym, co znajduje się na rynku. Jednak jej zdecydowanym plusem jest prostota oraz intuicyjność w obsłudze, możliwość budowania własnych scenariuszy, a także estetyczne walory graficzne.

6.3. Wyzwania i trudności problemu oraz sukcesy projektu

Największe wyzwania i osiągnięcia, jakie napotkaliśmy, polegało na stworzeniu niezawodnego algorytmu pozwalającego zasymulować ruch ludzi przechodzących przez wąskie gardło. Jak widać w części 3 proponowany model jest skomplikowany i składa się z wielu części. Jednym z kroków jakie musieliśmy pokonać, który zakończył się sukcesem była walidacja i kalibracja naszej symulacji. Wyznaczenie najlepszego rozwiązania obliczeń symulacji w naszym algorytmie oraz przełożenie zjawiska rzeczywistego - ruchu ludzi na wzory i obliczenia matematyczne było dla nas czymś nowym, z którym na początku nie wiedzieliśmy jak się zmierzyć. Obliczenia zawarte w kodzie oraz ich poprawność wymagały wielkiego zaangażowania. Po tygodniach zmagań z kodem i algorytmem udało się go zooptymalizować.

Opracowanie wszystkich rozwiązań na postawione przez nas założenia zajęło sporo czasu. Kolejną rzeczą było wdrożenie tych rozwiązań, co zajęło jeszcze więcej czasu. Bardzo się cieszymy, że opracowaliśmy działające rozwiązanie, które sprawuje się całkiem niezawodnie i jest dla nas imponujące.

Bibliografia

[1] Black friday in south africa, stampede nike store east rand mall - https://www.youtube.com/watch?v=xtpomoyyxi4.

[2] Black friday madness (south africa) - https://www.youtube.com/watch?v=d6jdpqyy30c.

[3] M. Chraibi, A. Schadschneider, and A. Seyfried. *Modeling, Simulation and Visual Analysis of Crowds*. Springer-Verlag, New York, 2013.

[4] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo. *Walking ahead: The headed social force model*. PLOS ONE, 2017.

[5] U. N. Hassan, Z. Zainuddin, and I. M. Abu-Sulyman. A modified social force model for crowd dynamics. 2017.

[6] D. Helbing. Social forces: Revealing the causes of success or disaster, 2014.

[7] D. Helbing and A. Johansson. Pedestrian, crowd and evacuation dynamics. *SpringerLink*, 2011.

[8] S. P. Hoogendoorn, P. H. L. Bovy, and W. Daamen. *Pedestrian Evacuation Dynamic*. 2002.

[9] M. Kapałka. *Simulation of crowd behavior in dynamic environment. Symulacja w Badaniach i Rozwoju*. 2015.

[10] M. Moussaïd, D. Helbing, and G. Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *PNAS*, 2011.

[11] C. Ningbo, W. Wei, Q. Zhaowei, Z. Liying, and B. Qiaowen. *Simulation of pedestrian crossing behaviors at unmarked roadways based on social force model.Discrete Dynamics in Nature and Society*. 2017.

[12] J. Wąs. *Modelowanie dynamiki tłumu. Pomiary Automatyka Robotyka*. Kraków, 2011.