

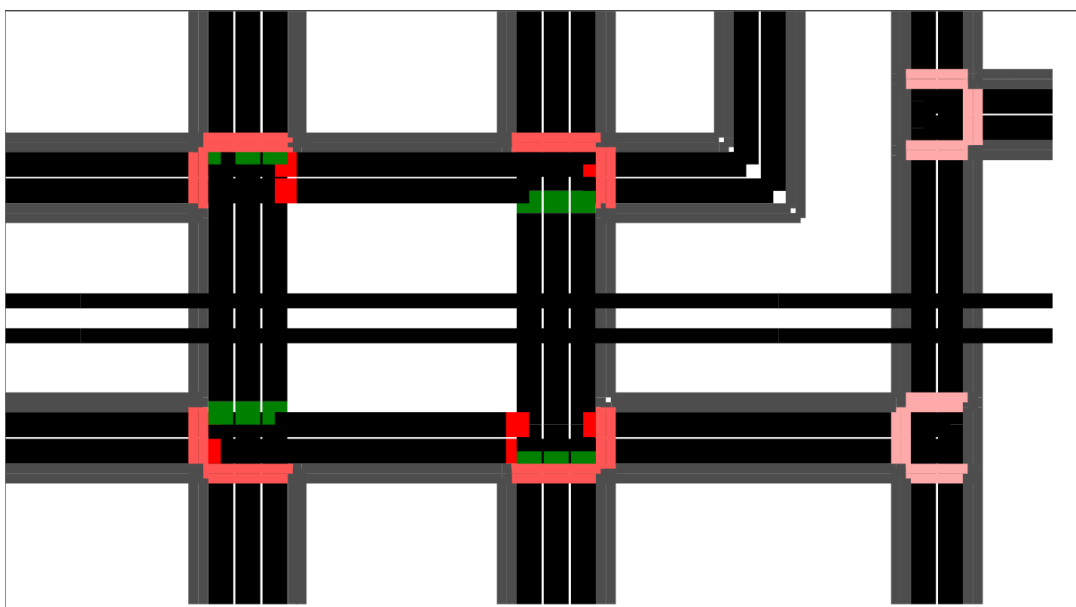
Symulacja ruchu drogowego

Jakub Wojtalewicz 188636, Piotr Wesołowski 188923

Napisany przez nas projekt w języku C# to zaawansowana symulacja ruchu drogowego, która obejmuje różnorodne elementy, takie jak tramwaje, samochody i piesi. Jego celem jest realistyczne odwzorowanie zasad ruchu drogowego i praca nad wdrażaniem systemów autonomicznych które zwiększają przepustowość i bezpieczeństwo ruchu.

1. Stworzenie mapy miasta ze skrzyżowaniami jezdni, chodników i torowisk

Korzystając z programu do tworzenia grafiki wektorowej Inkscape stworzyliśmy model naszego miasta, łącząc odpowiednie wektory odcinków dróg, dbając o ich ortogonalność oraz dokładność. Następnie stworzoną grafikę wyeksportowaliśmy do pliku w formacie .svg który pozwolił nam późniejsze pracowanie nad nim w naszym programie. Tak prezentuje się końcowa wersja tego pliku:

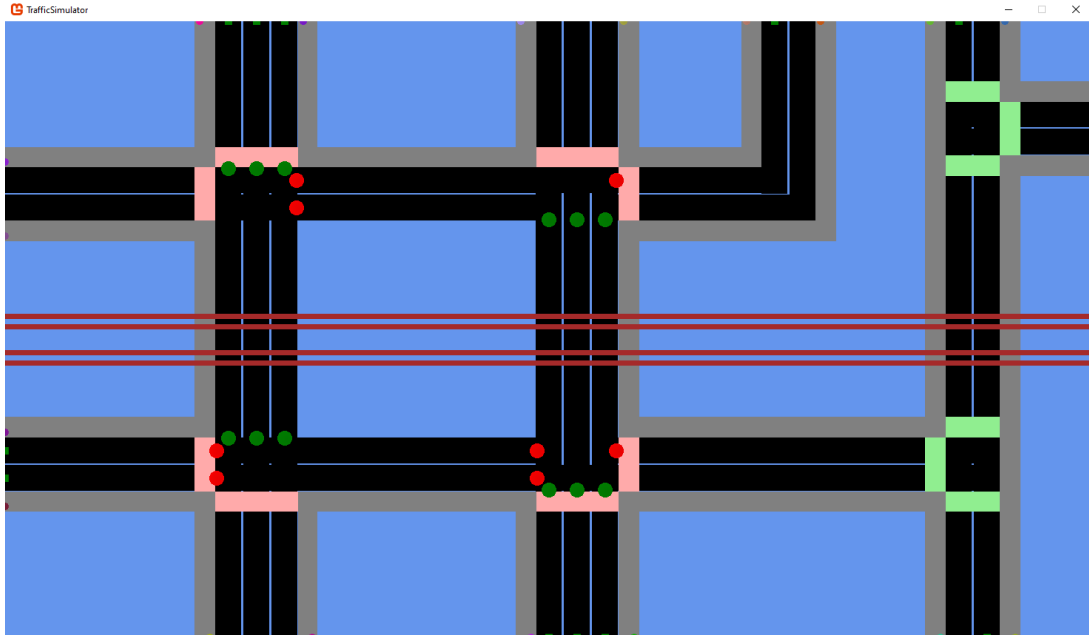


Następnie korzystając z dobrodziejstw klasy XmlReader zaczytywaliśmy do programu położenia wszelkich odcinków a także sygnalizacji świetlnej wykorzystując kolejne linie kodu w formacie svg:

```
id="path1176" />
<path
  style="fill:none;stroke:#000000;stroke-width:5;stroke-linecap:butt;stroke-linejoin:miter;st
  d="M 48.565,82.817788 V 88.159"
  id="path1176" />
```

korzystając z wartości atrybutów odpowiednich elementów.

Tak przygotowane fragmenty naszego na razie pustego miasta byliśmy w stanie wyświetlać w interfejsie graficznym Monogame, dostępnym w języku C#.



Każdy odcinek istniejący w naszej symulacji jasno precyzuje jedyny możliwy kierunek poruszania się po nim obiektów.

2. Dodanie uczestników ruchu do naszej symulacji

a) Samochody

W naszej symulacji porusza się kilkanaście samochodów, których to celem od momentu pojawienia się na naszej mapie drogowej jest przedostanie się do dowolnie wylosowanego zjazdu z mapy. Wykorzystaliśmy do tego zadania algorytm który oblicza najkrótszą możliwą trasę pomiędzy dwoma dowolnymi punktami na mapie. Te skomplikowane obliczenia wykonaliśmy raz, a ich rezultaty zapisaliśmy do pliku, z którego to odczytujemy potrzebne informacje w celu uniknięcia powtarzania wykonywania tych obliczeń.

Samochody podobnie jak pozostali „aktorzy” naszej symulacji muszą stosować się do obowiązujących zasad ruchu drogowego. min:

- Poprawnie respektują sygnały z sygnalizatora świetlnego
- Poza skrzyżowaniami kierowanymi światłami, ustępują pierwszeństwa pieszym przechodzącym przez przejścia drogowe.
- Ustępują pierwszeństwa tramwajowi przejeżdżającemu przez jezdnię, co więcej znajdując się przed przejazdem tramwajowym, decydują się na przejechanie przez nie, tylko mając pewność, że jest miejsce, aby zatrzymać się za nimi. (W razie korku żaden samochód nie będzie stał na torach)
- Respektują zasady prawej ręki poza skrzyżowaniami kierowanymi, tj. ustępują pierwszeństwa samochodom nadjeżdżającym z prawej strony.
- Używają kierunkowskazów, aby wyraźnie kierunkować pozostałym uczestnikom ruchu drogowego ich zamiary dalszego ruchu na najbliższym skrzyżowaniu.

b) Piesi

Po chodnikach naszego miasta porusza się 100 pieszych, którzy to podobnie jak samochody poruszają się do wylosowanego na początku miejsca na brzegu miasta. Pomimo faworyzowania ich przez prawo drogowe muszą oni jednak stosować się do pewnych reguł, a oto one:

- Konieczność respektowania przejść drogowych kierowanych światłami
- Oraz podobnie jak samochody ustąpić pierwszeństwo nadjeżdżającemu tramwajowi w momencie przejścia przez tory na pasach

c) Tramwaje

W naszym małym mieście występują 2 tramwaje. Zgodnie z Polskimi zasadami ruchu drogowego, tramwaje mają pierwszeństwo ponad innymi uczestnikami ruchu drogowego. Każdy tramwaj porusza się zgodnie z wcześniej ustalonym kursem, zatrzymując się na 2 przystankach drogowych na kilka sekund po czym ruszają w dalszą trasę.

3. Serwer zarządzający naszą symulacją

Z uwagi na wytyczne naszego projektu, aby każdy uczestnik ruchu był osobnym procesem, wszyscy „aktorzy” komunikują się ze sobą za pomocą rozwiązania sieciowego TCP/IP, co umożliwia wymianę informacji o swoim aktualnym położeniu oraz zapytań dotyczących możliwości przemieszczenia się w określonym czasie.

```
private TcpListener mainServer = new TcpListener(IPAddress.Any, 13131);
private TcpListener carServer = new TcpListener(IPAddress.Any, 15000);
private TcpListener pedestrianServer = new TcpListener(IPAddress.Any, 16000);
private TcpListener tServer = new TcpListener(IPAddress.Any, 17000);
private List<TcpClient> carClients = new List<TcpClient>();
private List<TcpClient> pedClients = new List<TcpClient>();
private List<TcpClient> tramClients = new List<TcpClient>();
```

Główny serwer symulacji pełni rolę arbitra, który analizuje te zapytania, uwzględniając zasady ruchu drogowego oraz bezpieczeństwo uczestników. To serwer zna dokładne położenie każdego uczestnika w danym momencie i w spornych przypadkach decyduje kto znajduje się w danym miejscu w danym czasie.

```
if (cars[index].IsMoveAllowed(TrafficLightsZones, roadStructure, trams, pedestrians))
{
    cars[index].setPosition(xPos, yPos);
    byte[] response = Encoding.ASCII.GetBytes("YE");
    stream.Write(response, 0, response.Length);
}
else
{
    byte[] response = Encoding.ASCII.GetBytes("NO");
    stream.Write(response, 0, response.Length);
}
```

Dzięki skorzystaniu z tego rozwiązania sieciowego mamy pewność odbioru naszych pakietów informacji, więc nie ma ryzyka, że jakiś pakiet się zagubi po drodze. Każdy uczestnik po obliczeniu swojej nowej pozycji na podstawie aktualnej prędkości i kierunku poruszania się, pyta się serwera czy ruch do danego punktu jest możliwy. Jeśli odpowiedź jest twierdząca, to faktycznie następuje ta zmiana, która wyświetlana jest w oknie naszej aplikacji w interfejsie Mono Game. Jeśli jednak serwer nie wyrazi na to zgody (np. w celu uniknięcia kolizji) uczestnik nie może wykonać przemieszczenia, jeśli więc miał w tym momencie pewną prędkość, zaczyna ją wytracać zgodnie z równaniami ruchu jednostajnie opóźnionego a wartością tegoż opóźnienia zdefiniowaną dla każdego z jednostek osobno). Na skutek tego istnieje szansa na wystąpienie kolizji z innym uczestnikiem ruchu, który objawia się stworzeniem zatoru, na drodze który

należałoby rozwiązać dodając do naszej symulacji dodatkowe jednostki które w pewien sposób dbałyby o płynne naprawianie skutków karamboli. Jako że w większości przypadków sprawdzamy, czy ruch jest możliwy odpowiednio wcześniej przed skrzyżowaniem, a prawdopodobieństwo wystąpienia powyższego problemu – znikome, nie zaimplementowaliśmy specjalnego rozwiązania w tej wersji programu.

```
byte[] newPos = new byte[2 * sizeof(int)+3];
Buffer.BlockCopy(Encoding.ASCII.GetBytes("POS"), 0, newPos, 0, 3);
Buffer.BlockCopy(BitConverter.GetBytes(position.X), 0, newPos, 3, sizeof(int));
Buffer.BlockCopy(BitConverter.GetBytes(position.Y), 0, newPos, 3+sizeof(int), sizeof(int));
dataServer.Send(newPos, newPos.Length);
IPEndPoint endPoint = new IPEndPoint(IPAddress.Any, 0);
byte[] ans = connectionServer.Receive(ref endPoint);
string reply = Encoding.ASCII.GetString(ans, 0, ans.Length);
if (reply == "NO")
{
    position.X = prevPosX; position.Y = prevPosY;
}
```

4. Dodatkowe informacje

- Cała symulacja jest oczywiście niezależna od prędkości maszyny, na której jest uruchomiona, wynika to bowiem z liczenia czasu pomiędzy kolejnymi obiegami pętli.
- Zdecydowana większość parametrów takich jak prędkości, rozmiary, przyspieszenia, opóźnienia, czasy oczekiwania itp. są sparametryzowane tzn. zmiana jednego bądź kilku z nich nie spowoduje błędnego wyświetlania symulacji czy też problemów z jej ogólnym działaniem.
- Mapa i rozkład naszej symulacji, również pozwala na mniejsze lub większe modyfikacje, większości przypadków wystarczy zmienić kierunki dróg w grafice wektorowej a dostosowanie kodu do tych zmian nie powinno być dużym wyzwaniem.
- Większa część kodu jest pisana z zachowaniem dobrych praktyk programowania, unikając powtórzeń i wykorzystując ogólne funkcje. Pozwala to na skalowalność projektu i ułatwia wspólną pracę nad jego rozbudową.

Projekt ten jest efektem pracy i zaangażowania, mającego na celu stworzenie realistycznej symulacji ruchu drogowego, która może znaleźć zastosowanie w analizie i doskonaleniu infrastruktury drogowej oraz bezpieczeństwa na drogach.

Jesteśmy przekonani, że nasza symulacja ruchu drogowego będzie interesująca i wartościowa dla osób trzecich, które są zainteresowane badaniem ruchu drogowego, optymalizacją infrastruktury lub bezpieczeństwem drogowym. A oto jeszcze raz, fragment prezentujący ogólny wygląd naszego programu:

