Dokumentacja projektu programu CupCarbon

Temat: inteligentne światła

Wojciech Tomczyk 12K2

Cyprian Wojtas 12K2

1. Wstęp

CupCarbon jest programem służącym do symulacji czujników Smart City i Internetu Rzeczy SCI-WSN, programowany w Javie. Przykładowymi celami są projektowanie, wizualizacja, debugowanie i weryfikacja rozproszonych algorytmów monitorowania lub gromadzenia danych środowiskowych, jak i tworzenie scenariuszy środowiskowych, np. Pożary, telefony komórkowe, czy gaz. Interfejs służący do projektowania wykorzystuje platformę OpenStreetMap (OSM), ergonomiczny i łatwy w użyciu, umożliwia rozmieszczanie czujników na mapie. Poza umożliwianiem rozmieszczania elementów w przestrzeni dwuwymiarowej zawiera ona także skrypt nazwany SenScript, konieczny do programowania poszczególnych czujników. Symulacja CupCarbon opiera się na warstwie aplikacji węzłów, nie symuluje wszystkich warstw protokołów, gdyż ilość informacji konieczna do takiej symulacji byłaby za duża, dlatego też CupCarbon traktowany jest jako uzupełnienie istniejących symulatorów. Podsumowując program ten pozwala na utworzenie symulacji w określonych przez użytkownika warunkach, np. miejskim, który reaguje na zaplanowane, bądź losowo wygenerowane zdarzenia.

2. Opis sprzętu

Wykorzystane moduły programu CupCarbon:

- Moduł modelu miasta - służy on jako graficzne zaprezentowanie rozmieszczenia czujników, jak są połączone i jaką drogą prowadzony jest sygnał, w tym projekcie spełnia rolę obszaru miasta z siecią inteligentnych świateł.
- Moduł sieci – pozwala na zaprojektowanie czujników, które potem są rozmieszczane na mapie, i symulują określone zdarzenia.
- Moduł communication script – stosowany jest jako interpreter kodu stosowanego w czujnikach, czyli języka SenScript. W zależności od zdarzeń w projekcie instrukcje w kodzie są wykonywane podczas symulacji.
- Moduł symulacji - moduł symulujący zdarzenia, w tym projekcie odpowiada za ruch urządzenia mobilnego, który z kolei wchodzi w interakcję z innymi czujnikami.

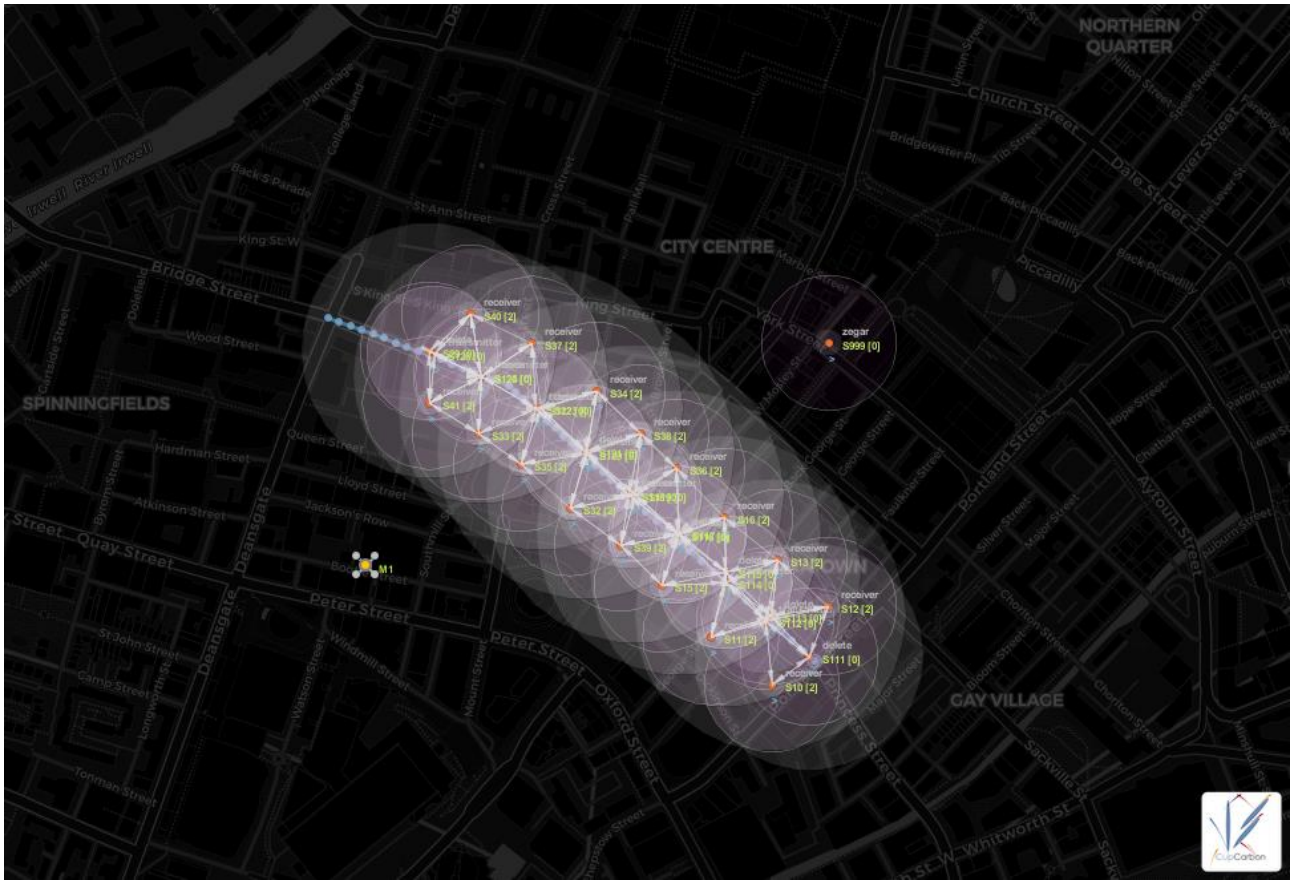
Zastosowane w projekcie elementy to ścieżka określona markerami, urządzenie mobilne(mobile) zaprogramowane do podążania po ścieżce i sensory.

3. Wymagania projektu

Projekt ma na celu przedstawić system świateł, które w zależności od pory dnia, jak i odległości od urządzeń mobilnych, jest uruchamiany. Działanie jest zaprezentowane na określonym odcinku drogi, zaś dla zasymulowania działania czasu każdy krok urządzenia odpowiada minięciu jednej godziny. Aby program działał poprawnie czujniki odpowiedzialne za odczytywanie pozycji urządzenia mobilnego mają wysyłać informacje do odbiorników, które spełniają rolę świateł. Ponadto cały system ma działać wyłącznie w określonym przedziale czasu, który z poziomu kodu jest łatwy do zmiany.

4. Projekt sieci

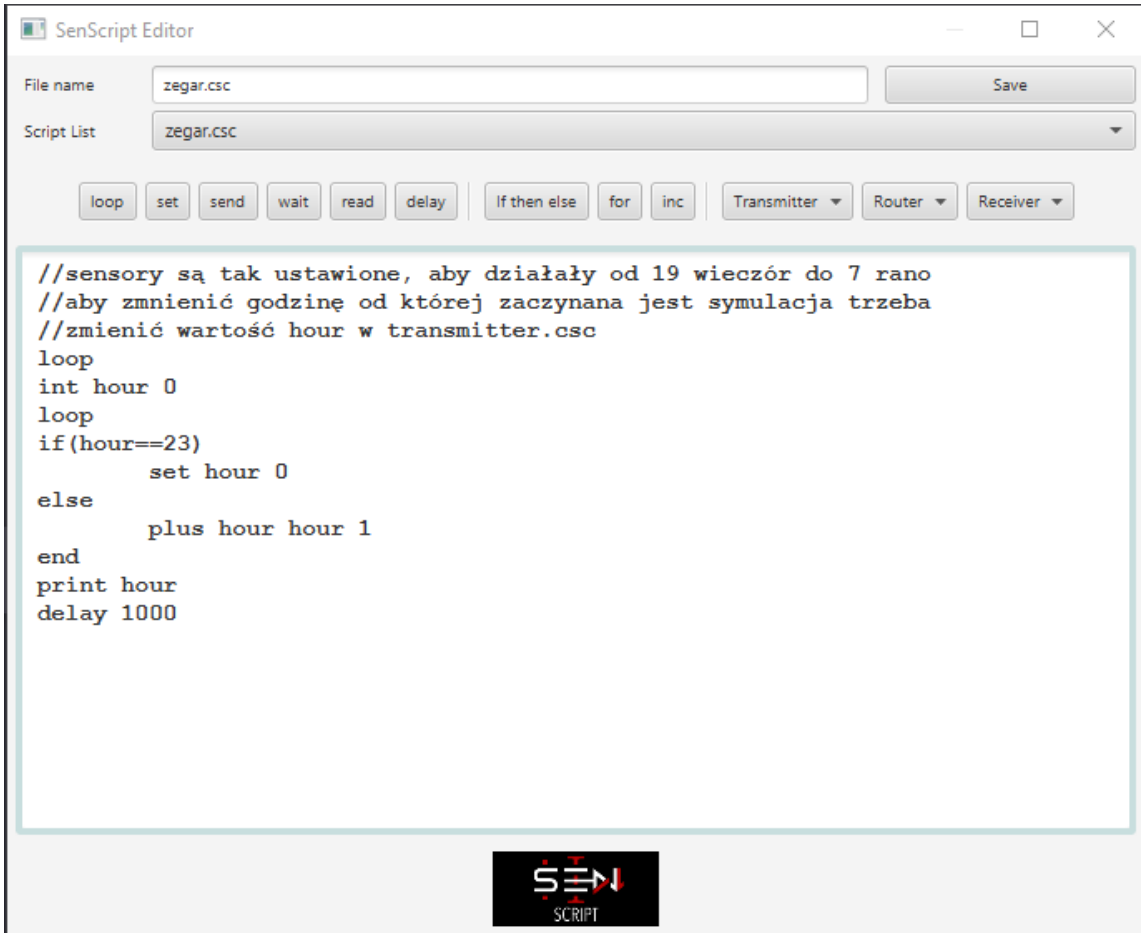
W całym projekcie występują cztery typy czujników, z czego jeden z nich służy jako informacja na temat czasu. Czujniki te to transmitter, receiver, delete, oraz zegar.



Ilustracja 4.1: Widok całego projektu.

4.1. Czujnik zegar

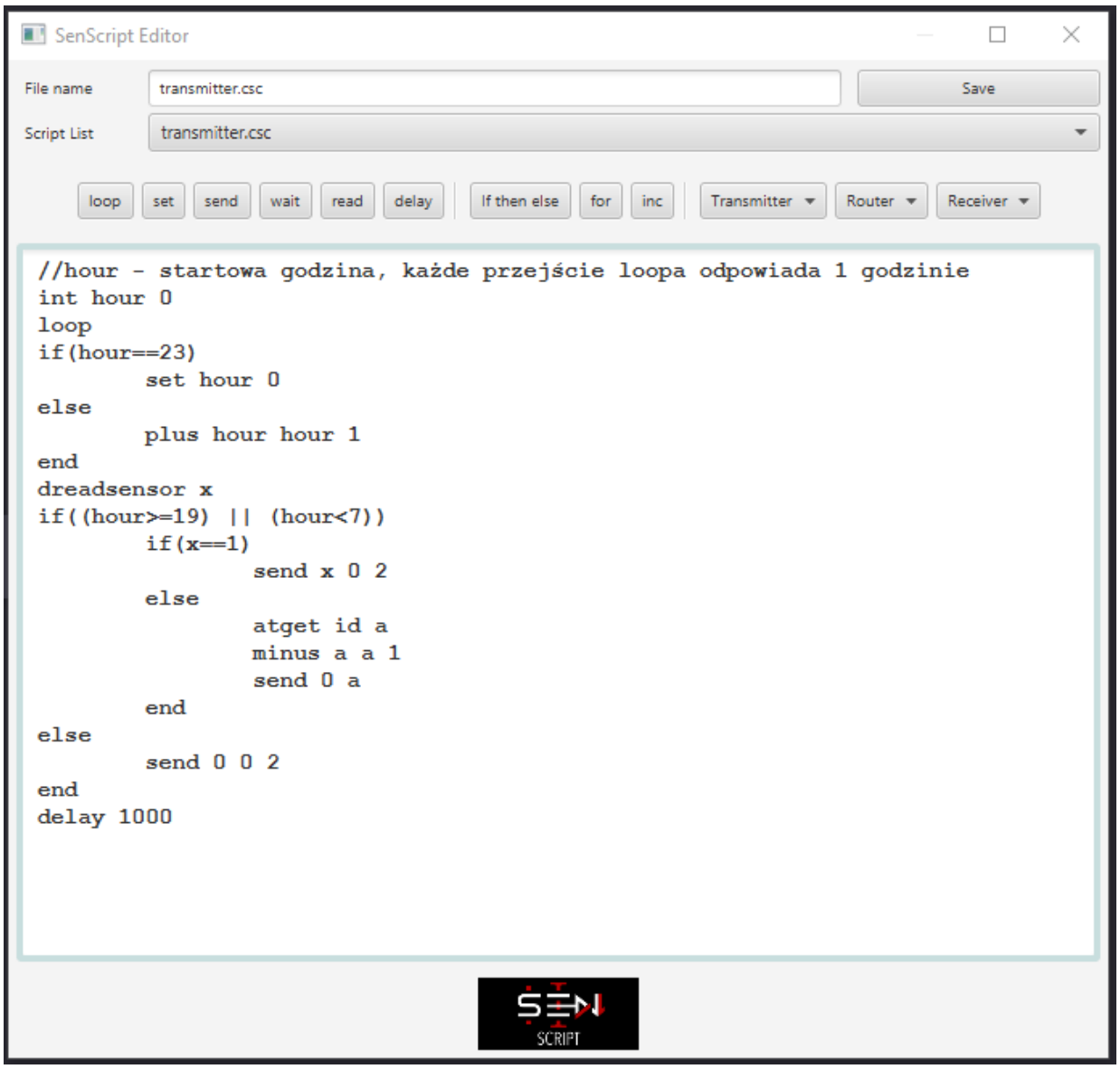
Celem czujnika zegar jest graficzne pokazanie aktualnej godziny, jaka jest w symulacji. W momencie w którym wartość hour doszłaby do 24, jest ona zmieniana na zero, co odpowiada rozpoczęciu kolejnego dnia.



Ilustracja 4.2: Kod pliku zegar.csc

4.2. Czujnik transmitter

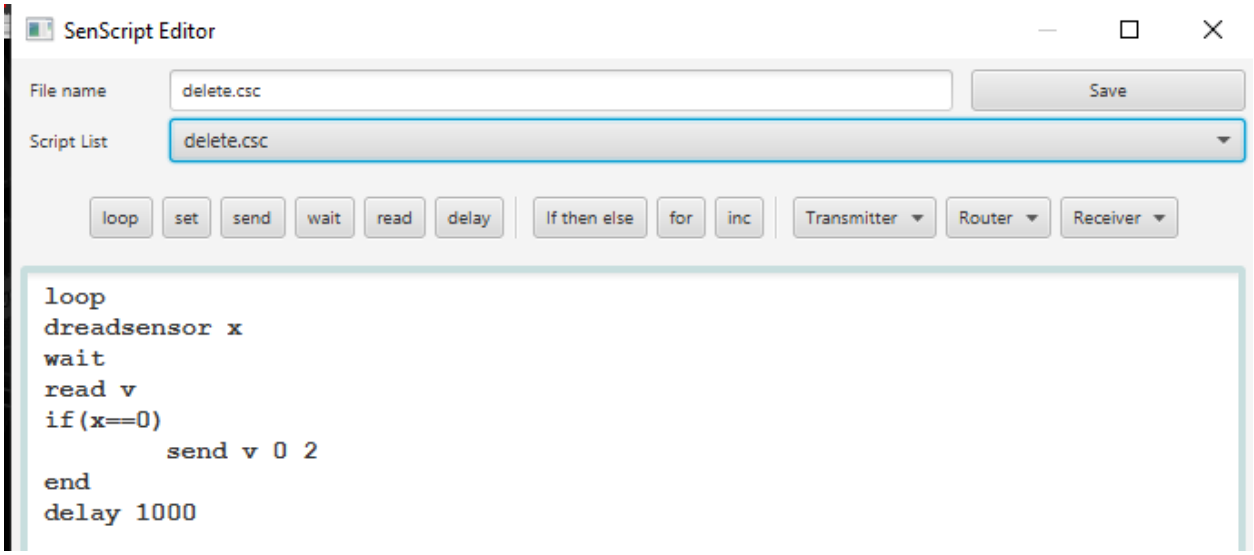
Główny czujnik, którego zadaniem jest sprawdzanie, czy urządzenie mobilne jest w zasięgu i wysyłanie informacji do poszczególnych innych czujników. Zarówno jak zegar, transmitter sprawdza jaka jest godzina, jeżeli wartość mieści się w określonym przedziale, to następnie wysyłana jest wiadomość, czy w określonym zasięgu znajduje się jakieś urządzenie, czy nie.



Ilustracja 4.3: Kod pliku transmitter.csc

4.3. Czujnik delete

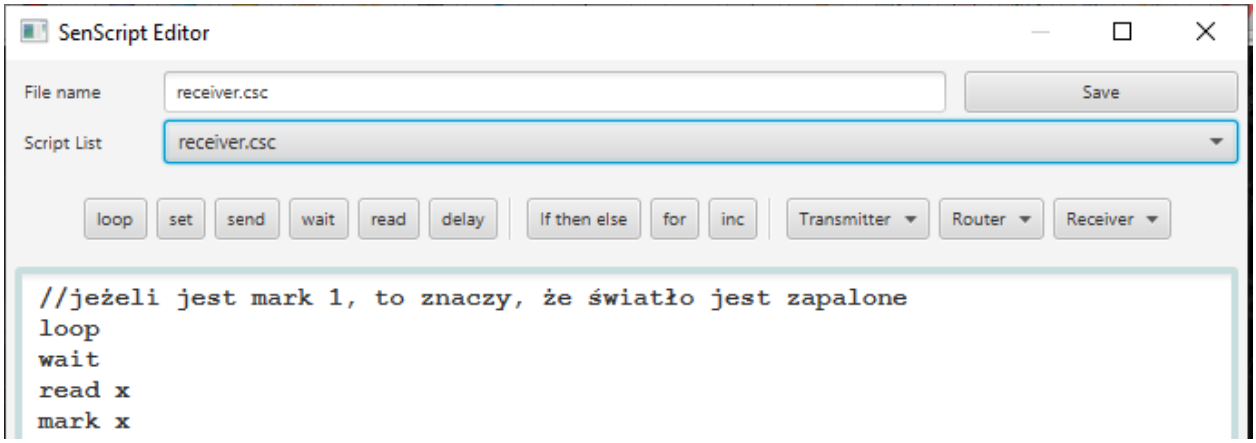
Czujnik o znacznie większym zasięgu, niż transmitter. Jego zadaniem jest wyłączanie świateł, które znajdują się poza zasięgiem użytkownika. Informacja o tym, czy należy poszczególne światło wyłączyć jest dostarczana przez transmitter, co odczytywane jest jako wartość v, a następnie wysyłane do czujników odpowiadających za światła.



Ilustracja 4.4: Kod pliku delete.csc

4.4. Czujnik receiver

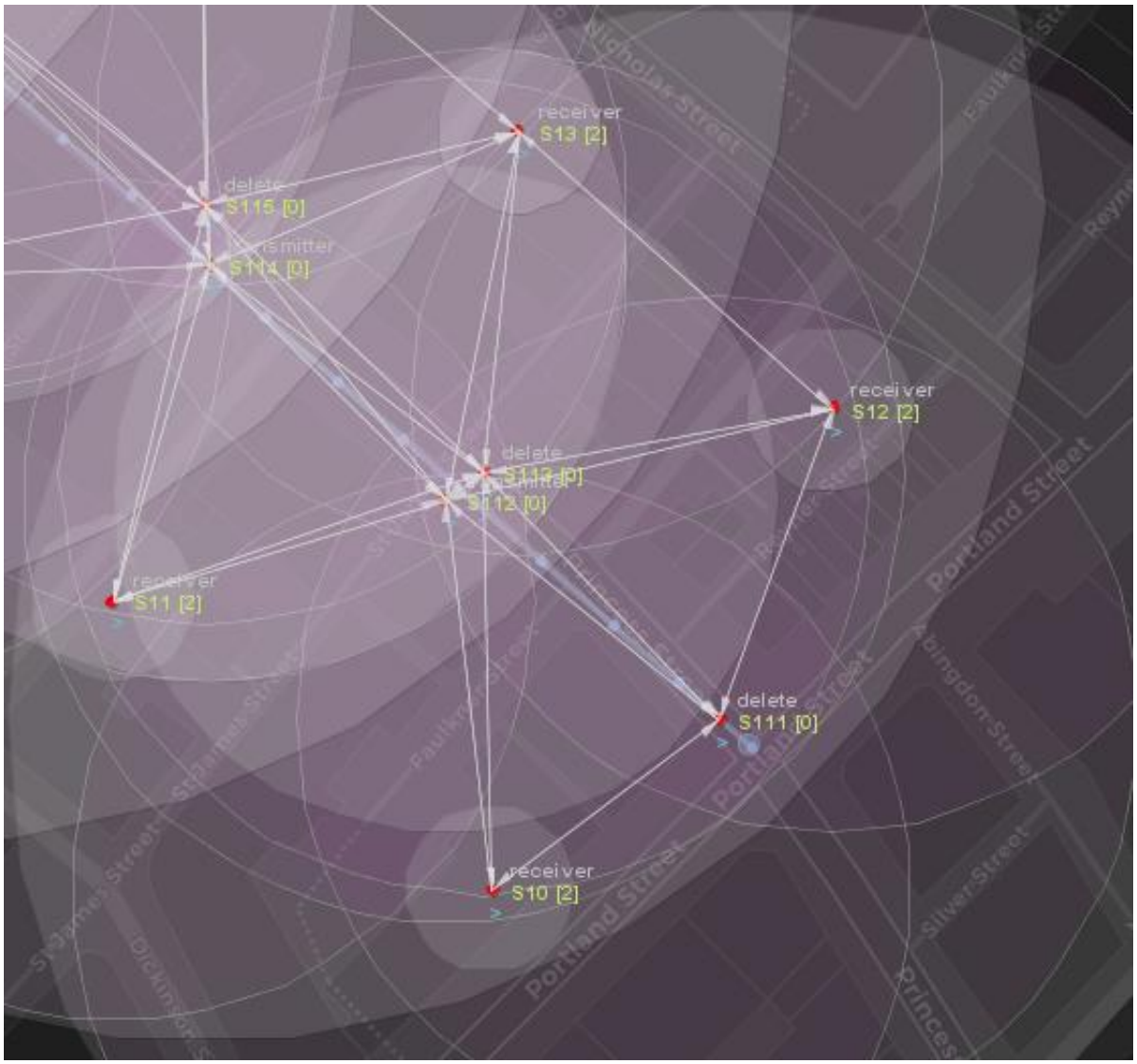
Czujnik światła, w zależności od informacji otrzymanej od transmittera, bądź delete zmieniana jest wartość mark, która odpowiada za to czy lampa jest zaświecona, czy nie.



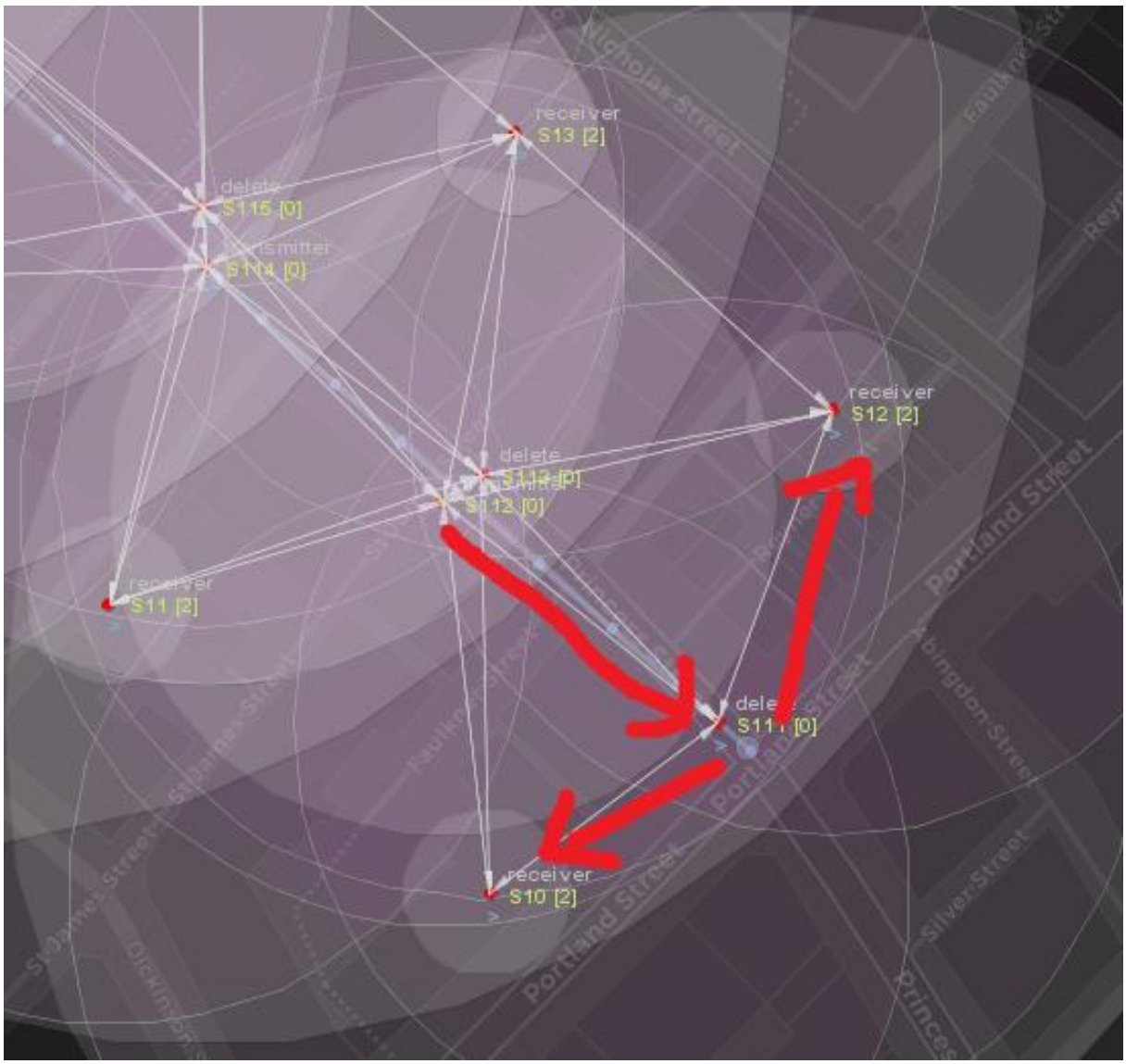
Ilustracja 4.5: Kod pliku receiver.csc

4.5. Szczegółowe omówienie zasady działania

Aby cały system mógł odpowiednio działać, określone czujniki mają przypisane identyfikatory i klasy. Wszystkie receivery mają przypisaną klasę 2, dzięki czemu możliwe jest wysyłanie informacji dotyczących wyłącznie świateł. Zapobiega to też zapętlaniu się informacji, gdzie na przykład dwa transmittery w nieskończoność wysyłają do siebie te same dane. Dodatkowo czujniki delete i transmitter mają tak ułożone identyfikatory, że idąc po kolei numerami występują one na przemian. Pozwala to czujnikom transmitter wysyłać informacje do czujników delete, które mają wartość id o jeden mniejszą. Zapobiega to nie tylko zapętlaniu, ale także wysyłaniu informacji do czujników delete znajdujących się tuż obok transmittera, bądź przed nim.

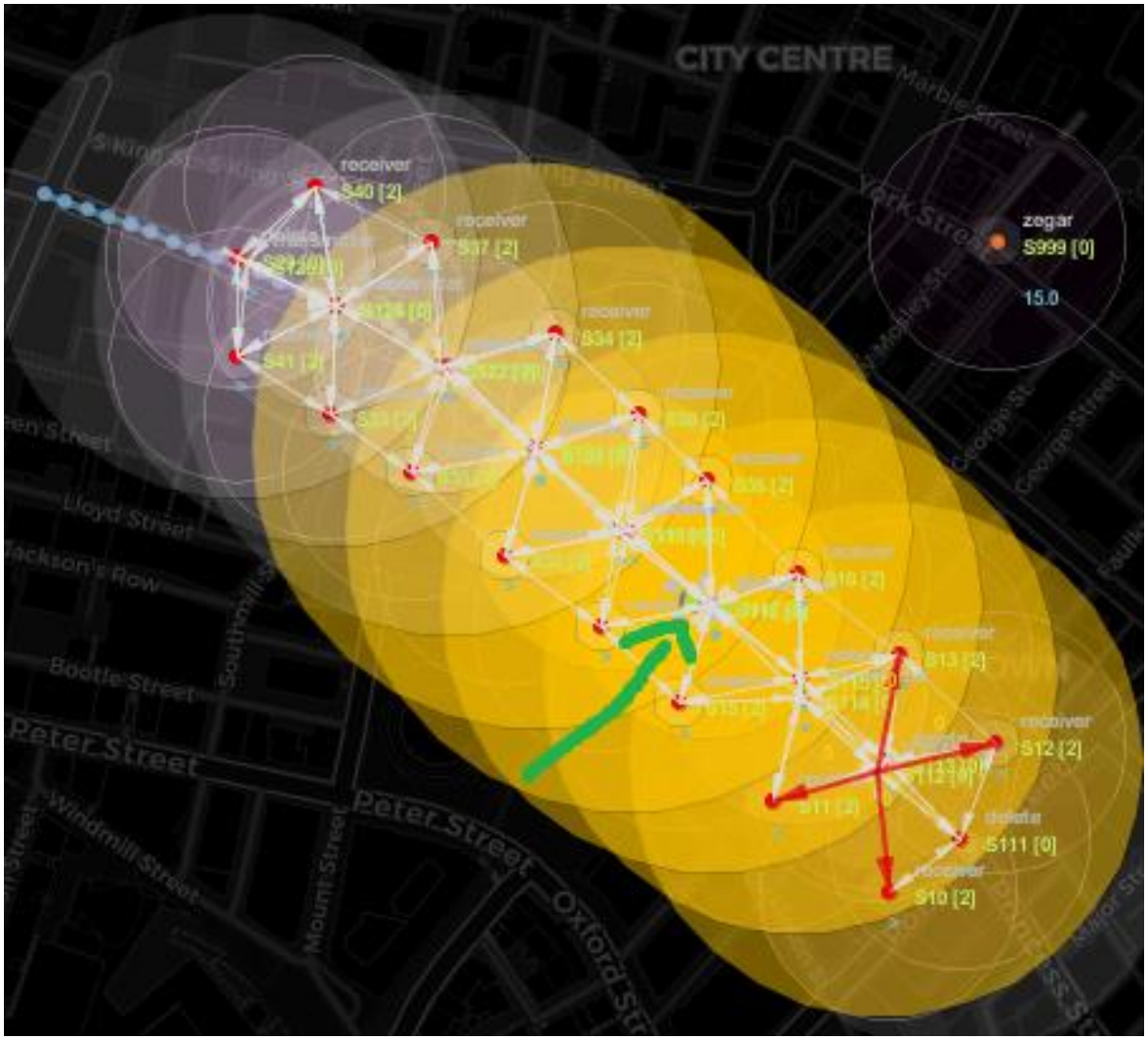


Ilustracja 4.6: Fragment projektu



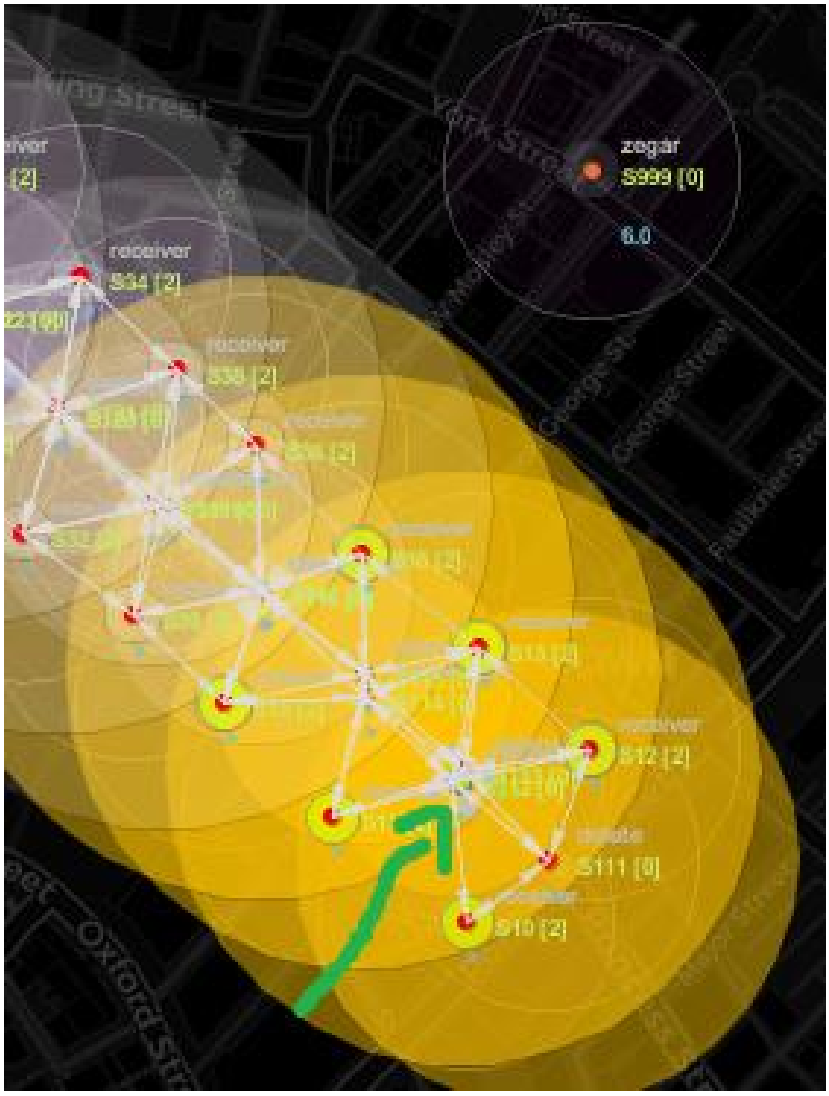
Ilustracja 4.7: Ścieżka, jaką podąża sygnał od transmittera, przez delete, do receivera

Przykład działania systemu, pomimo tego, że urządzenie znajduje się w zasięgu czujników, to światła nie są aktywne, ze względu na godzinę.



Ilustracja 4.8: Strzałką zaznaczona została pozycja urządzenia.

Dodatkowo, aby cały system mógł działać kluczowe jest poprawne ustawienie wszystkich czujników. Transmittery zaprogramowane są, aby wysyłały informację do konkretnych czujników delete, bazując na identyfikatorach. Także oba z tych typów czujników są połączone do określonej liczby świateł, co zezwala na stopniowym uruchamianiu/wyłączaniu świateł podczas przemieszczania się urządzenia mobilnego.



Ilustracja 4.9: W tym przypadku zaś światła są uruchomione, gdyż oba warunki są spełnione,urządzenie jest w zasięgu i czas jest poprawny.

5. Podsumowanie

CupCarbon umożliwia na symulację pracy czujników, które odpowiednio zaprogramowane i umiejscowione mogą nie tylko odwzorować zdarzenia z życia codziennego, ale też być wykorzystane do ułatwienia go, coś co było celem projektu inteligentnych świateł. Program sam na pewno daje użytkownikowi sporo możliwości podczas projektowania, jednakże ilość informacji, czy też przykładów na rozwiązanie konkretnych problemów w internecie nie była duża, co utrudniło pracę. Jednym z większych problemów była niejasność dotycząca kolejności otrzymywania informacji z różnych źródeł w danym kroku, co wymusiło zastosowanie dodatkowych czujników zabezpieczających, aby symulacja odbywała się poprawnie. Mimo tych trudności projekt ten pozwolił nam nauczyć się, jak utworzyć prosty system przesyłu danych w środowisku miejskim, oraz podstawy projektowania sieci czujników.