

Politechnika Wrocławska

Sprawozdanie

do zadania rekrutacyjnego nr. 1 do działu *software*

Wykonał

Ivan Danylenko

Wrocław 2021

1. Cel zadania

Stworzenie środowiska wodnego o nieregularnym kształcie dna, pozwalającego przetestować algorytm wyszukiwania źródła dźwięku, znajdującego się w nim.

2. Część teoretyczna

Żeby dźwięk totarł od swojego źródła do pewnego punktu w zbiorniku wodnym, musi on pokonać pewną odległość. Jeżeli na jego drodze pojawia się przeszkoda, prosta przemieszczenia zamienia się na krzywą, długość drogi się zwiększa. Im dłuższa jest ta krzywa, tym słabsze natężenie dźwięku będzie zarejestrowane w obserwowanym punkcie.

W rozwiązaniu przyjęto uproszczony model rozchodzenia się fal, zgodnie z którym natężenie dźwięku w pewnym punkcie definiuje się jako natężenie dźwięku generowanego przez źródło, podzielone przez kwadrat długości najkrótszej¹ łamanej linii, którą musi poruszać się dźwięk, by dotrzeć do punktu:

$$I_P = \frac{I_{ZR}}{d^2} \quad (1)$$

I_P – wartość natężenia dźwięku w punkcie oddalonym od źródła dźwięku o d

I_{ZR} – natężenie dźwięku generowanego przez źródło

Reasumując: natężenie dźwięku zmniejsza się zgodnie z prawem odwrotnych kwadratów.

Krzywa linia zastępuje się łamaną w celu zmniejszenia złożoności obliczeniowej algorytmu, który ma na celu zdefiniowanie natężenia dźwięku w każdym punkcie zbiornika wodnego.

3. Plan wykonania pracy

Wykonanie zadania zostało podzielone na mniejsze podzadania, w szczególności:

1. Stworzenie zamkniętego systemu wodnego
 - 1.1. Stworzenie zbioru strukturalnych jednostek z których składa się system
 - 1.2. Stworzenie mechanizmu relacji pomiędzy sąsiednimi punktami systemu
 - 1.3. Nadanie kształtu dna
2. Dodanie dźwięku do środowiska
 - 2.1. Dodanie źródła dźwięku do środowiska
 - 2.2. Tworzenie logiki rozchodzenia się fal dźwiękowych
 - 2.3. Definiowanie natężenia dźwięku w każdym punkcie środowiska, będącym wodą
3. Wykonanie wyszukiwania źródła dźwięku
4. Wizualizacja

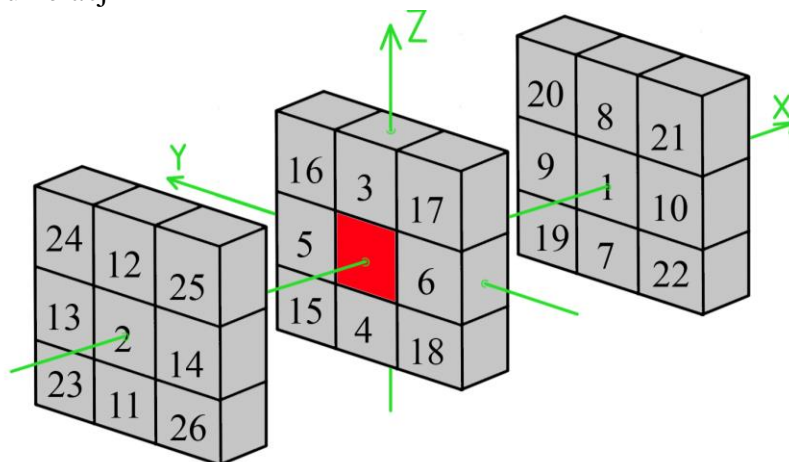
¹ Najkrótszej możliwej w ramach działania algorytmu

4. Wykonanie pracy, tworzenie algorytmów

Podzadanie 1.1. Jako strukturalną jednostkę przyjęto metr sześcienny. Zbiór takich jednostek tworzy system zamknięty, w którym zachodzi sprawdzenie działania wszystkich algorytmów. Owy zbiór ma postać listy trójwymiarowej, składającej się ze strukturalnych jednostek.

Podzadanie 1.2. Relacje pomiędzy sąsiednimi strukturalnymi jednostkami wyglądają w ten sposób, że każda jednostka strukturalna ma referencje do jednostek otaczających ją.

Rysunek 1. Ilustracja sposobu zmieszczenia sąsiednich jednostek strukturalnych oraz ich numeracji



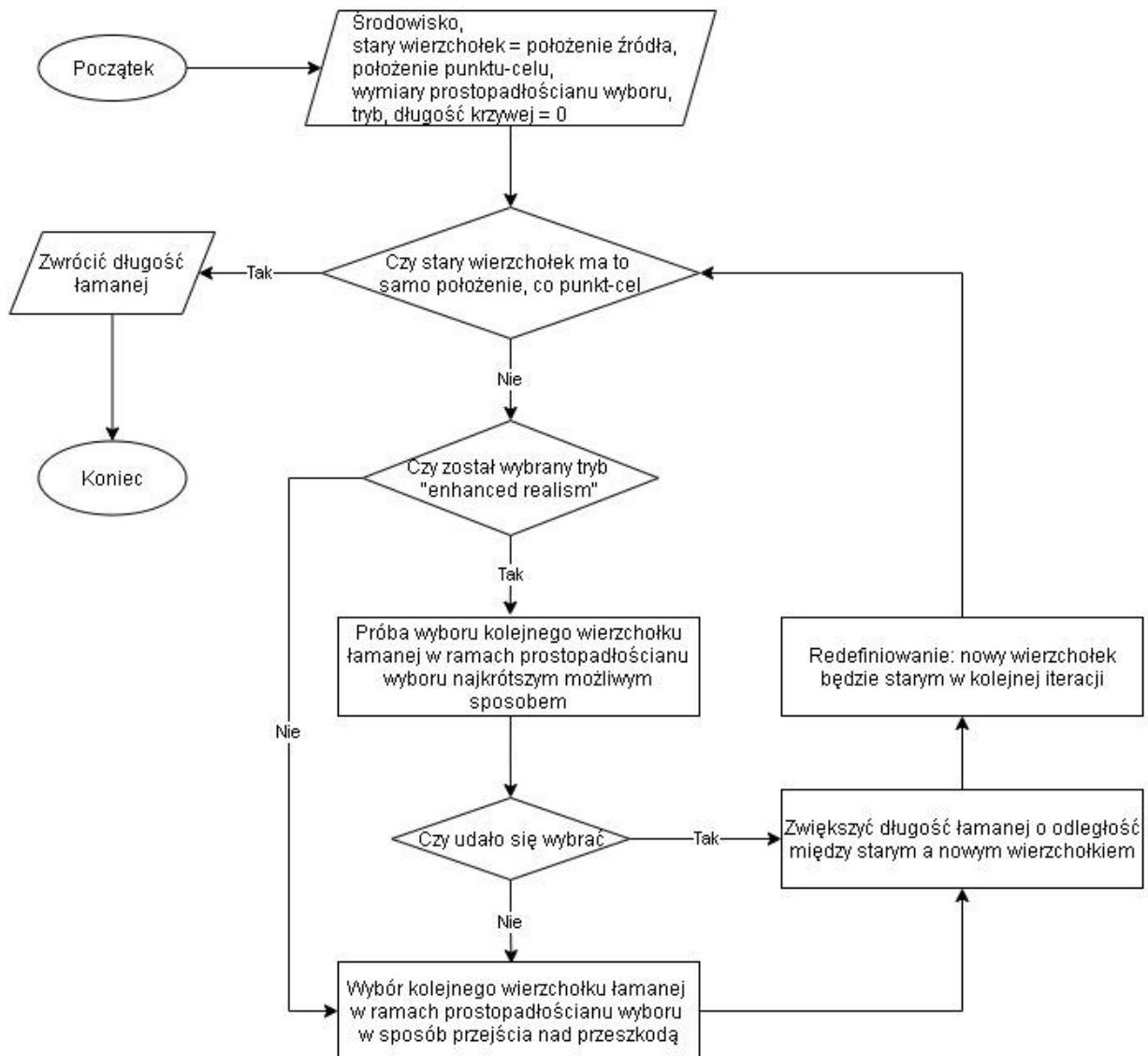
Podzadanie 1.3. Nadanie kształtu dna dokonuje się poprzez wczytywanie mapy wysokości (heightmap): dla każdego piksela mapy wyznacza się wartość jego alpha-kanalu i w zbiorniku wodnym na odpowiednich współrzędnych powstaje „słup ziemi” o wysokości równej wyznaczonej wartości alpha-kanalu. Dzieje się to na skutek zmiany parametru *is_water* odpowiednich jednostek strukturalnych – obiektów *CubicMetre()*. Jeśli parametr *is_water* nie jest logiczną wartością *True*, obiekt zaczyna stanowić barierę.

Podzadanie 2.1. Dodanie źródła dźwięku dokonuje się przez zamianę jednej jednostki strukturalnej z parametrem *is_water* o wartości *True* innym obiektem *CubicMetre()* z parametrem *is_water* o wartości *False* i liczbowo zdefiniowanym parametrem *sound_intensity*.

Podzadanie 2.2. Ogólne rozumienie mechanizmu rozchodzenia się fal zostało opisane w części teoretycznej niniejszego sprawozdania.

Podzadanie 2.3. Do zdefiniowania natężenia dźwięku w pewnym punkcie zbiornika trzeba wyznaczyć długość łamanej, o której mowa w teoretycznej części niniejszego sprawozdania, jak widać ze wzoru nr. 1. Służy do tego algorytm, którego ogólny wygląd jest przedstawiony na schemacie blokowym niżej.

Schemat 1. Algorytm wyszukiwania najkrótszej łamanej od punktu A do punktu B



Szczególnej uwagi potrzebuje nazwa „prostopadłościan wyboru” – jest to abstraktny obiekt, który pozwala w swoich ramach dokonywać wyboru nowego wierzchołka, bez obaw, że nowy odcinek przejdzie całkowicie przez przeszkodę nie uginając się. Oznacza to, że wymiary prostopadłościanu wyboru w płaszczyźnie XY całkowicie zależą od najmniejszych wymiarów przeszkód w dwóch kierunkach na jakiejkolwiek warstwie zbiornika wodnego. Użycie prostopadłościanu wyboru pozwala na bardziej dokładne definiowanie długości najkrótszej łamanej, ale też ma swoje minusy w postaci „ścinania” kawałków przeszkód, co wpływa na ostateczną długość krzywej, i im większe wymiary ma prostopadłościan wyboru, tym większy wpływ to ma na wynik końcowy algorytmu. Warto jednak zaznaczyć, że większość takich odchyłeń od wartości rzeczywistej niwelują same siebie i w końcu ma się mniej więcej jednolodne rozchodzenie się dźwiękowych fal, a rzeczywiste wodne środowiska też nie są idealne, także można to nazwać zmienną losową.

Szczegółowe działanie algorytmu jest opisane w pliku `classes.py` w funkcji o nazwie `shortest_curve()`.

Podzadanie 3. Łódź podwodna jest w stanie rozpoznawać najgłośniejsze miejsce w ramach jednego metru wokół siebie do każdej strony. Wynika z tego, że algorytm wyszukiwania źródła dźwięku polega na wybieraniu „najgłośniejszego” sąsiada strukturalnej jednostki w której znajduje się łódź, przemieszczaniu się do niego i powtórzeniu tych działań do momentu, kiedy wartość natężenia w strukturalnej jednostce, w której znajduje się łódź, będzie równa wartości natężenia w najgłośniejszym sąsiedzie: to będzie oznaczać, że łódź podwodna znajduje się obok źródła dźwięku.

Podzadanie 4. Do wizualizacji użyto silnika *Ursina Engine*. Szczegóły wizualizacji można znaleźć w komentarzach dokumentacyjnych oraz jednolinijkowych do plików *visualisation.py* oraz *functions_for_visualisation.py*.

5. Wnioski

Ze względu na to jaki sposób rozwiązania został wybrany, użycie sonarów łodzi podwodnej nie miało sensu, gdyż łódź nie jest w stanie uderzyć się w przeszkodę. Sens miałyby stworzenie takiego warunku, żeby łódź podwodna nie mogła podchodzić do przeszkód bliżej niż na zadaną odległość, ale w ramach tego zadania to nie znalazło zastosowania.

Sposób realizacji środowiska wodnego pozwoli tworzyć symulacje różnych zjawisk, zachodzących w wodzie, gdyż można przyjąć mniejszą skalę i założyć, że jednostka strukturalna to nie metr sześcienny, a centymetr, mikrometr itd. Wszystko zależy od charakterystyk i możliwości komputera na którym dokonuje się symulacji, ponieważ wraz ze zwiększeniem ilości strukturalnych jednostek rośnie złożoność obliczeniowa. W przypadku tego rozwiązania, pozwala ono na wygenerowanie nawet Rowu Mariańskiego, jednak niepodobna tego zaprezentować w niniejszym rozwiązaniu ze względu na charakterystyki komputera wykonawcy.

Minusem zaprezentowanego rozwiązania jest niemożliwość generowania dna środowiska z jaskiniami podwodnymi, ponieważ w owym rozwiązaniu wodne środowisko generuje swój teren z pobieranej mapy wysokości, jednak nietrudno rozszerzyć funkcjonalność rozwiązania poprzez stworzenie algorytmu, generującego bardziej skomplikowany kształt dna. Aktualny algorytm rozchodzenia się fal dźwiękowych zachowa swoją funkcjonalność przy takich zmianach.

Przyjęty mechanizm rozchodzenia się fal dźwiękowych jest bardzo uproszczony i nie uwzględnia odbicia fal od powierzchni, interferencji, dyfrakcji, tłumienia fal i wielu innych zjawisk. Także niedoskonałym jest algorytm definiujący wartość natężenia dźwięku dla każdej jednostki strukturalnej, ponieważ jeżeli podczas poszukiwania kolejnego wierzchołka łamanej napotka on wysoką płaską przeszkodę, nie będzie on w stanie zidentyfikować, czy krótsza droga znajduje się z boku tej ściany i stworzy kolejny wierzchołek nad ową przeszkodę. Niewątpliwie, można by było stworzyć algorytm który jest jak najbardziej zbliżony do rzeczywistości, jednak miałyby to swoje konsekwencje w postaci zbyt dużej złożoności obliczeniowej. Mniej istotne minusy tego algorytmu zostały opisane w dokumentacji do funkcji wykorzystującej go.

Na koniec, demonstrację działania rozwiązania można obejrzeć w nagraniach, dołączonych do niniejszego sprawozdania, a wszystkie pliki z nim powiązane można znaleźć na stronie <https://github.com/Kowd-PauUh/Water-environment-simulation-PL>

6. Materiały dydaktyczne

- [1] https://en.wikipedia.org/wiki/Sound_intensity
- [2] <https://fis.wikireading.ru/4026>
- [3] https://www.ursinaengine.org/cheat_sheet.html