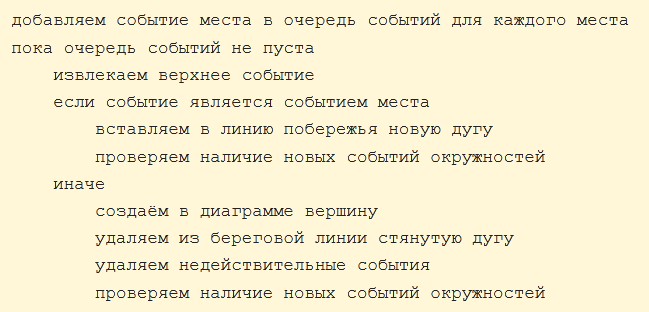
Звіт з лабораторної роботи №9  
на тему «Діаграма Вороного – метод Форчуна»  
з дисципліни «Комп’ютерна графіка»  
студента 3-го курсу Факультету комп’ютерних наук та кібернетики   
групи ІПС-32  
Поліщук Єгора Даниловича

**Постановка задачі.**  
В просторі E2 задана множина точок S, потужності N. Необхідно для кожної точки pi з множини S визначити локус точок (x,y) на площині, для яких відстань до pi менша ніж до будь-якої іншої точки множини S.  
Таке формулювання задачі називається задачею областей близькості.   
Її розв’язок описує діаграма Вороного.  
  
**Розв’язання.**Означимо поняття.Якщо задано дві точки pi та pj, то множина точок більш близьких до pi, ніж до pj є півплощиною, що визначається серединним перпендикуляром відрізка pipj. Позначимо таку площину H(pi, pj). Множину точок, більш близьких до точки pi ніж до будь-якої з деякої скінченної множини OtherPoints, позначатимемо Vi. Вона утворена перетином N-1 півплощин Hij(pi, OtherPoints[j]). Ця множина опуклим многокутником, який має не більш ніж N-1 сторін.  
Така область Vi називається многокутником Вороного, що відповідає точці рi.   
Для множини S з N точок, отримані N областей Вороного утворять діаграму Вороного.  
Точки pi  - sites, Vi – cells.  
  
Один з способів побудови діаграми Вороного – застосування алгоритму Форчуна.  
Алгоритм Форчуна базується за замітаючій прямій (sweeping line).

Отже, нехай на площині задано множину точок {pi} i = 1…N.  
Для визначеності вважаємо, що замітаюча пряма (ЗП) рухається згори вниз.  
Тому сортуємо точки за зменшенням їх ординати.  
  
Важливо зауважити, що в кожен момент часу можено розглядати тільки точки, що знаходяться вище ЗП. Також треба розуміти що границя діаграми, що «межує» з ЗП, змінюється, тому необхідно зберігати її, для цього використовується берегова лінія (beach line). Вона зберігає точки site (для яких формування клітинки ще не завершене), арки парабол (шматки) побудованих для поточних точок таким чином, що site є фокусом параболи, а ЗП – директрисою, та точки перетину арок.  
  
Для ЗП визначено два види подій (event): “point event” та “circle event”. Події зберігаємо у пріоритетній черзі подій Q.   
Кожна подія асоційована з деякою точкою.  
Point event – це фактично site з деякою додатковою інформацією.  
Cirle event – фактично точка перетину двох арок, утворена при їх розширенні таким чином, що арка, яка була між ними видаляється з берегової лінії. Утворена точка стає вершиною клітинки Ворого, з’єднується з відповідною їй попередньою вершиною та утворює ребро.  
ЗП рухається по подіях, тобто на кожному кроці вона знаходиться або на site або на точці перетину арок, яка відповідає вершині клітинки Вороного.  
Важливо зауважити, що арка вставляється в берегову лінію тільки при події точки (point event) – тобто коли ЗП виявляє нову точку, утворюється нова парабола з фокусом у даній точці, спочатку ця парабола – фактично промінь, що виходить з поточної точки.  
Видалення арки з берегової точки може проводитися тільки при події кола (circle event).  
  
Отримані ребра, що утворюють діаграму Вороного, необхідно зберігати у зручній структурі. Такою виявляються DCEL (doubly linked egde list) – реберний список з подвійними зв’язками. При формуванні ребра шляхом описаним виже, необхідно додавати його до списку.

Отже, повертаючись до алгоритму, після відсортовані точки (sites) кладемо у вигляді відповідних подій у чергу подій Q.  
Надалі поки черга не пуста:  
1) дістаємо подію з Q;  
2) якщо це подія точки – оброблюємо як подію точки (описаним вище шляхом);  
3) якщо це подія кола – оброблюємо як подію кола.  
Після роботи циклу, необхідно коректно завершити ребра.

Беручи до уваги усе вище сказане, основну частину алгоритму можна записати так:



**Складність.**Складність сортування O(N\*logN), вставка точки до берегової лінії (beach line) – O(logN),  
загальна складність виконання алгоритму – O(N\*logN).

**Джерела.**

1. <https://habr.com/ru/post/309252/>
2. <https://habr.com/ru/post/430628/>
3. <https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A4%D0%BE%D1%80%D1%87%D1%83%D0%BD%D0%B0>
4. <http://www.ams.org/publicoutreach/feature-column/fcarc-voronoi>
5. <https://en.wikipedia.org/wiki/Fortune%27s_algorithm>