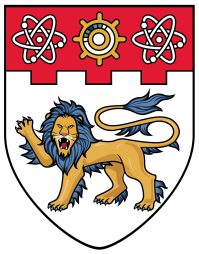

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SOC: Space Optimization for COVID-19

Charissa Irene Utomo and Kieran Ng
Matriculation number: U2040071K and U2040691K
Email: cutomo001@e.ntu.edu.sg and m200084@e.ntu.edu.sg

Nanyang Technological University
School of Physical and Mathematical Sciences

May 2021

Abstract

Given the contagious nature of the COVID-19 virus, it is important to isolate those who tested positive for the virus to minimise further spread of the virus. However, space is a luxury in this country. Hence it is a necessity to optimally house and isolate the patients.

We will create a model that can optimally split a given space into as many subsections as possible in order to adequately isolate patients. We would like to maximise the given space and sustain the functionality of each ward, while also allowing for as many wards as possible. We decided to implement *Python* programming with *PyGame* library to create a 2D diagram which visualise the rooms allocations.

Keywords: optimally, isolate, isolation wards

Acknowledgement

We would like to give our sincere gratitude to SPMS Odyssey Committee for giving us an opportunity to tap our knowledge and allow us to explore and research on *SOC*. Thank you to those who helped make this *SCOPE* project possible.

Table of Contents

Abstract **i**

Acknowledgement **ii**

Nomenclature **iv**

1 Introduction **1**

2 Material and Methods **2**
2.1 Deliverables 2

3 Features **3**
3.1 Frontend 3
3.2 Backend 3
3.3 UI/UX 3
3.4 Others 3

4 Behind the math **4**
4.1 Voronoi Diagram 4
4.2 Fortune Algorithm 4
4.2.1 Data Structures 4
4.2.2 Classes 4
4.3 DCEL 5
4.4 Program Characteristics 6
4.5 Creation of Rooms 6

5 Conclusion **7**

References **8**

Lists of Figures **8**

Nomenclature

$COVID-19$	Coronavirus disease 2019
$DCEL$	Doubly Connected Edge List
$s.t.$	Such That
$SCOPE$	SPMS Concurrent Projects for Everyone
SOC	Space Optimization for COVID-19
$SPMS$	School of Physical and Mathematical Sciences
UI	User Interface
UX	User Experience

Introduction

Given the contagious nature of the Covid-19 virus, it is important to isolate those who tested positive for the virus to minimize further spread of the virus. However, space is a luxury in this country. Hence it is a necessity to optimally house and isolate the patients.

Covid-19 is an unprecedented event and there are many facilities in various countries that are incredibly unprepared to tackle such a pandemic. We hope that this program can help isolation wards better plan their spaces and allow for greater allocation of patients. The program is available at https://github.com/charutomo/SOC.

Keywords: optimally, isolate, isolation wards

Material and Methods

2.1 Deliverables

A model that can optimally split a given space into as many subsections as possible in order to adequately isolate patients. It should maximise the given space and sustain the functionality of each ward, while also allowing for as many wards as possible.

The product is a model of a Top Down 2D paradigm drawn using the PyGame library. The program would output a diagram that splits into small sections of any 2D shape and considers optimizing the most effective space without a tradeoff of the operations of the original generic ward.

Features

3.1 Frontend

3.2 Backend

We implemented an algorithm such that it would generate the room allocation while considering the constraints such as specific equipments in the room, the dimensions, the pathways, number of exits, thickness of the dividers, amenities and facilities (toilet, consultation room, food collection area). We also optimize the time complexity of the program

3.3 UI/UX

3.4 Others

Behind the math

4.1 Voronoi Diagram

We are given a set of vertices V, a sweepline L, a beachline B, a priority queue Q.
Now, for each vertex v in V, we create a new site event using the position v and add it into a priority queue, sorted by ascending values of y.
The sweepline starts at $y = 0$, and moves downwards, passing through each event e in Q. At each event, you pop it from the list.
If the event is a site event, we add a new vertex v to a list of vertices $V*$. A parabola will be created with the focus being the vertex v and the directrix being the sweepline L. The parabola must be added to the beachline B.
B is a binary search tree. The leaf nodes in B represent the arcs of the beachline. The leftmost leaf is the leftmost arc and the rightmost leaf is the rightmost arc. It is stored as a reference to the site that created the arc. Each internal node is a breakpoint. It is stored as an ordered tuple of sites, the left site being the left arc and the right site being the right arc.
Each arc also stores a reference to a circle event in the priority queue Q. Each internal node has a reference to a half edge in the voronoi diagram.

4.2 Fortune Algorithm

Fortune's algorithm is used to generate a Voronoi diagram. It is a sweep line algorithm and has a time complexity of $O(nlogn)$.

4.2.1 Data Structures

We decided to implement a integrated binary search tree and priority queue to reduce the runtime of the algorithm

4.2.2 Classes

- Points:
Circumcircle

- Sweep Line:
A horizontal line that will travel across the plane either horizontally or vertically.

-Beach Line:
A piecewise defined function made up of multiple parabolas. It follows the sweep line. The breakpoints of the beach line lie on Voronoi edges of the final diagram. We shall define the beach line to be a mutable list of points, sorted from left to right. (Parabola – from Wolfram MathWorld, no date)

- Parabola:

A parabola is the set of all points in the plane equidistant from a given line L, the conic section directrix, and a given point F, the focus, not on the line. (Parabola – from Wolfram MathWorld, no date)

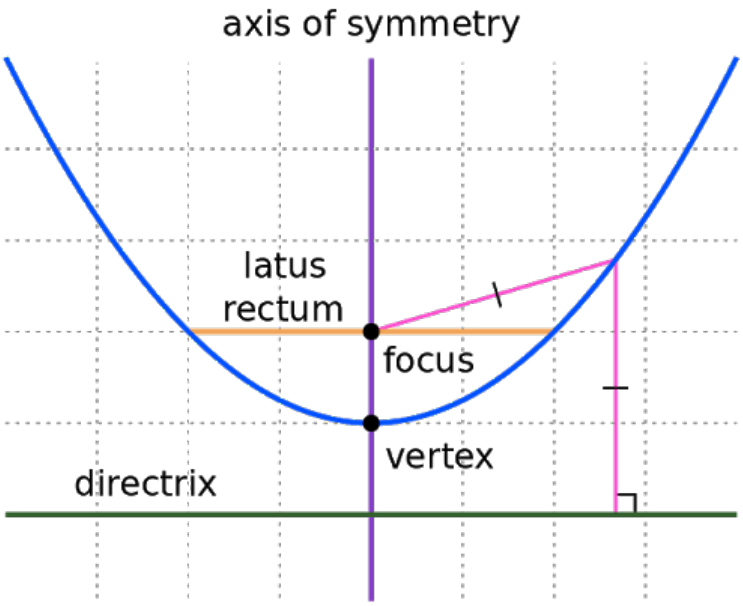

Figure 4-1: Visual representation of parabola

- Events:
There are two main events that the program will have to handle.

The first event will be called a Point Event. This happens when the sweep line passes through a point. When this happens, a new parabola will be created and it will be added to the beach line. To accommodate a new parabola, the nearest parabola to the point will be split into two.

> Suppose B is the set containing the parabolas making up the beach line.
> Suppose parabola J is the closest parabola to the point.
> Then we split J into $J1$ and $J2$, s.t. $B = \{\dots J,\dots\} \Longrightarrow \{\dots J1, J2,\dots\}$.
> Then we add a new parabola Q into the beach line. $\{\dots J1, J2,\dots\} \Longrightarrow \{\dots J1, Q, J2,\dots\}$.

The second event will be called a Vertex Event. This happens when the length of a parabola becomes 0. The parabola will be deleted and a new vertex will be created at that location. This vertex will be used for drawing the edges of the Voronoi diagram.

4.3 DCEL

DCEL, also known as Doubly connected edge list

4.4 Program Characteristics

The program characteristics are as follows:

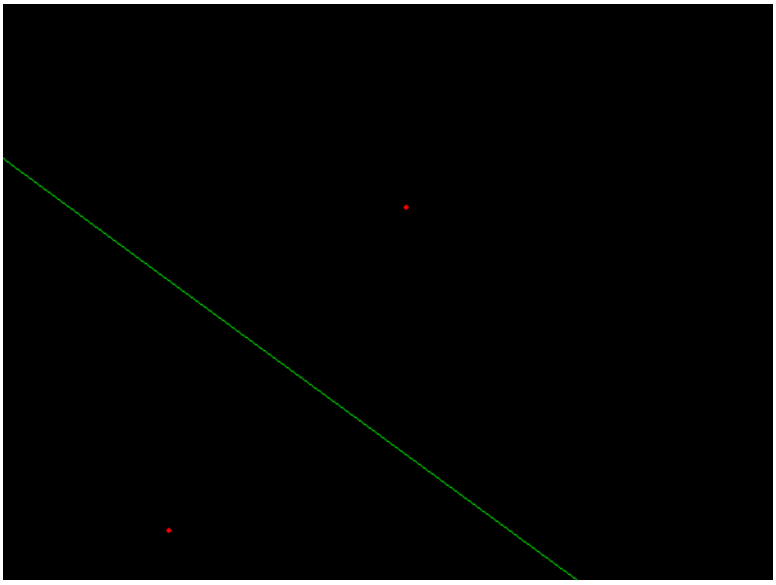

Figure 4-2: A sample of 2 point allocation of voronoi diagram

---

4.5 Creation of Rooms

Using the application of graph theory, we have decided to create a room that has the optimal number of wards using the following methods.

So given a doubly connected edge list (doubly linked list of edges), $G = (V, E, F, W)$. We are trying to find a subset of $G, H = (V, E*, F, W)$, such that there are a minimal number of edges, and H is still connected.

Each edge e in E has a w in W, where w is the weight of the edge. In our program, we can use w as the total area that the edge covers. Each w can be a list of 2 elements, $w[0]$ =length of e, $w[1]$ = width of e. Then the area is easily calculated.

Conclusion

Overall, it is an enriching experience for us to discover and learn more about optimization. Dealing with minimal spaces in the COVID-19 situation by maximising the resources of community and minimising space constraints using optimization. Other applications of our algorithm can be further applied to any given room that needs division of spaces such as HDB house allocations, creating floor plan and many more.

We hope that our program will be useful for better allocation of spaces in the pandemic. This concludes our project (readily available at https://github.com/charutomo/SOC) and feel free to contact us via email for further enquires. Thank you for your upmost support.

Lists of Figures

4-1 Visual representation of parabola 5
4-2 A sample of 2 point allocation of voronoi diagram 6