CREATING A 2D VISIBILITY SHADOW EFFECT ON UNITY PLATFORM WITH ROBUST C# CODE

Prepared by: Chicheng Zheng

Supervisor: Prof. Clark Verbrugge

Preparation for COMP396 Undergraduate Research Project

McGill University

April 2019

# Abstraction

Guards are very common in game, therefore it is important to implement the visibility of the guard in a map with obstacles. This project aims to create a 2D visibility shadow effort and apply it on real scenario.

目录

**[1. Introduction（motivation summary whole work）](#_Toc6884_WPSOffice_Level1)** **[3](#_Toc6884_WPSOffice_Level1)**

[1.1 Methodology](#_Toc29634_WPSOffice_Level2) [3](#_Toc29634_WPSOffice_Level2)

[1.2 Scope of the report](#_Toc8337_WPSOffice_Level2) [3](#_Toc8337_WPSOffice_Level2)

**[2. Background and Related Work](#_Toc29634_WPSOffice_Level1)** **[3](#_Toc29634_WPSOffice_Level1)**

[2.1 Terminology Definition](#_Toc23282_WPSOffice_Level2) [3](#_Toc23282_WPSOffice_Level2)

**[3. Implementation](#_Toc8337_WPSOffice_Level1)** **[5](#_Toc8337_WPSOffice_Level1)**

[3.1 Generate Obstacle and boundary](#_Toc3860_WPSOffice_Level2) [5](#_Toc3860_WPSOffice_Level2)

[3.2 Generate Ray Cast](#_Toc6090_WPSOffice_Level2) [5](#_Toc6090_WPSOffice_Level2)

[3.3 Generate Mesh](#_Toc24966_WPSOffice_Level2) [5](#_Toc24966_WPSOffice_Level2)

**[4. Test Case](#_Toc23282_WPSOffice_Level1)** **[5](#_Toc23282_WPSOffice_Level1)**

[4.1 Convex Polygon Obstacle](#_Toc7974_WPSOffice_Level2) [5](#_Toc7974_WPSOffice_Level2)

[4.2 Concave Polygon](#_Toc636_WPSOffice_Level2) [7](#_Toc636_WPSOffice_Level2)

[4.3 Partially View](#_Toc32200_WPSOffice_Level2) [8](#_Toc32200_WPSOffice_Level2)

[4.4 Visibility With Range Limitation](#_Toc28202_WPSOffice_Level2) [8](#_Toc28202_WPSOffice_Level2)

**[5. Conclusion(impact)](#_Toc3860_WPSOffice_Level1)** **[9](#_Toc3860_WPSOffice_Level1)**

**[6. Reference](#_Toc6090_WPSOffice_Level1)** **[9](#_Toc6090_WPSOffice_Level1)**

[a) define visiblity star shape polygon citation](#_Toc16296_WPSOffice_Level2) [9](#_Toc16296_WPSOffice_Level2)

[b) Art gallary book Ason’ algorithm](#_Toc18157_WPSOffice_Level2) [9](#_Toc18157_WPSOffice_Level2)

# Introduction（motivation summary whole work）

## 1.1 Methodology

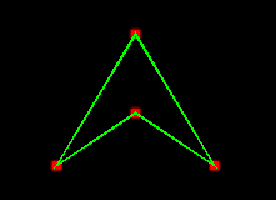
## 1.2 Scope of the report

# Background and Related Work

## 2.1 Terminology Definition

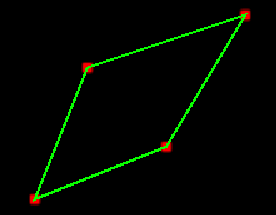
### 2.1.1 Concave Polygon

A concave polygon exist a line cross one of the vertex, and not all of the rest vertex are on one side of the line.



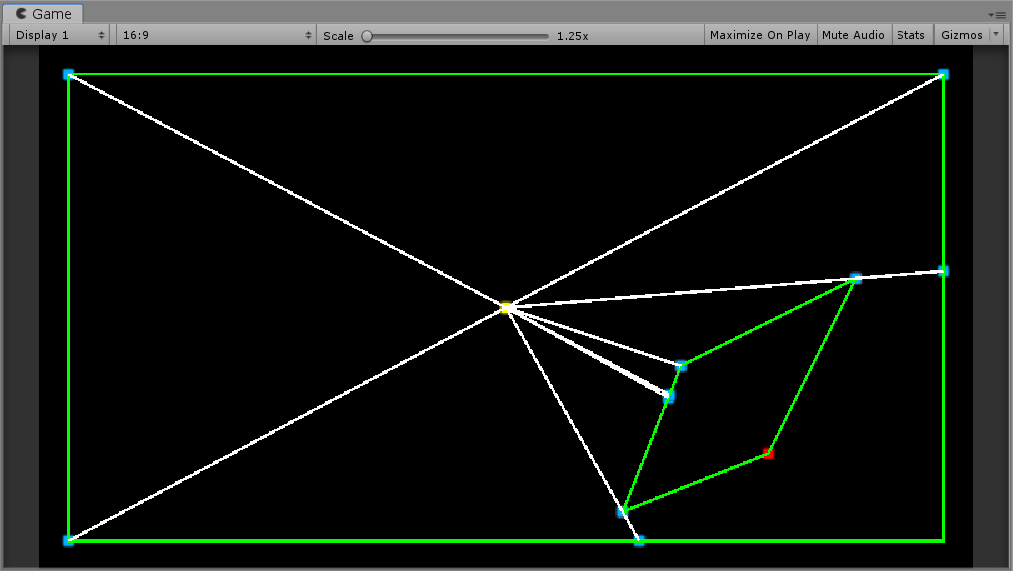
### 2.1.2 Convex Polygon

A convex polygon is the opposite of the concave, for every line cross the vertex, all of the rest vertex are on one side of the line.



### 2.1.3 Critical Point

All the hit points generated by the line cast from the view point, the visibility polygon is surrounded by the critical points. For example, all the blue points are critical points.



### 2.1.4 Sight Range

In the real world, it impossible for a guard to see infinity far away. Therefore, he has a range of visibility. Sight Range is the max value it could see from the view point.

### 2.1.5 Sight Angle

In the real world, it impossible for a guard to see all the things around it. Therefore the sight angle define the max range of perspective.

### 2.1.6 View Point

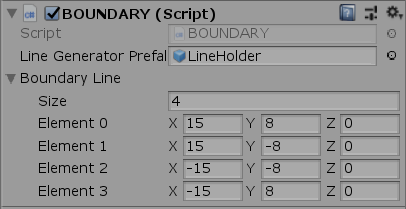
It is the guard position, and this project is focusing the visibility effect of related to this view point.

# Implementation

## 3.1 Generate obstacle and boundary

### 3.1.1 Boundary

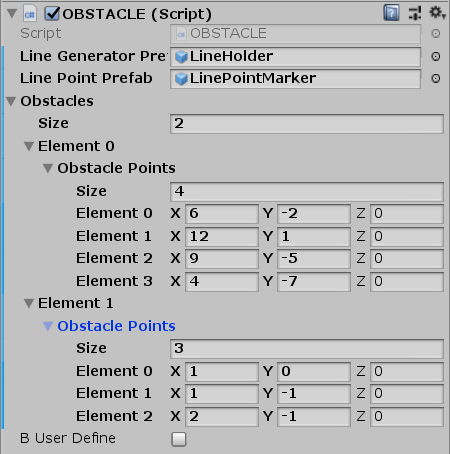
The Boundary Manager receive a list of Vector2D points denoted the vertex of the boundary,



it generate the edge collider 2D by loop through all the vertex and choose the adjacent pairs. We will have a list of edge collider before we generate the visibility area.

### 3.1.2 Obstacle

The Obstacle Manager receive a list of obstacles. Each obstacle is a list of Vector2D points.



Based on the list of obstacles, I generated a list of Polygon Collider 2D by loop through all the obstacles. This is basically same as generating the edge Collider 2D.

### 3.1.3 Triangulation

After we generate all the Polygon Collider 2D for each obstacles. We need to detect whether it is a concave obstacle or convex obstacle. If it is a concave obstacle, then we need to do triangulation. This mainly because when

## 3.2 Generate Ray Cast

### 3.2.1 Stable view point

### 3.2.2 Movable view point

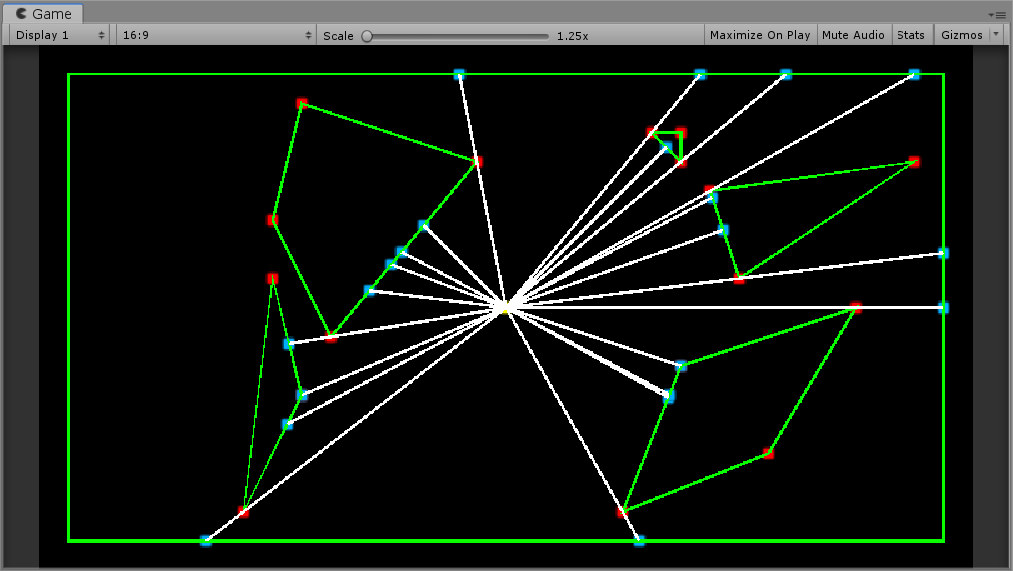
## 3.3 Generate Mesh

### 3.3.1 Manage the order

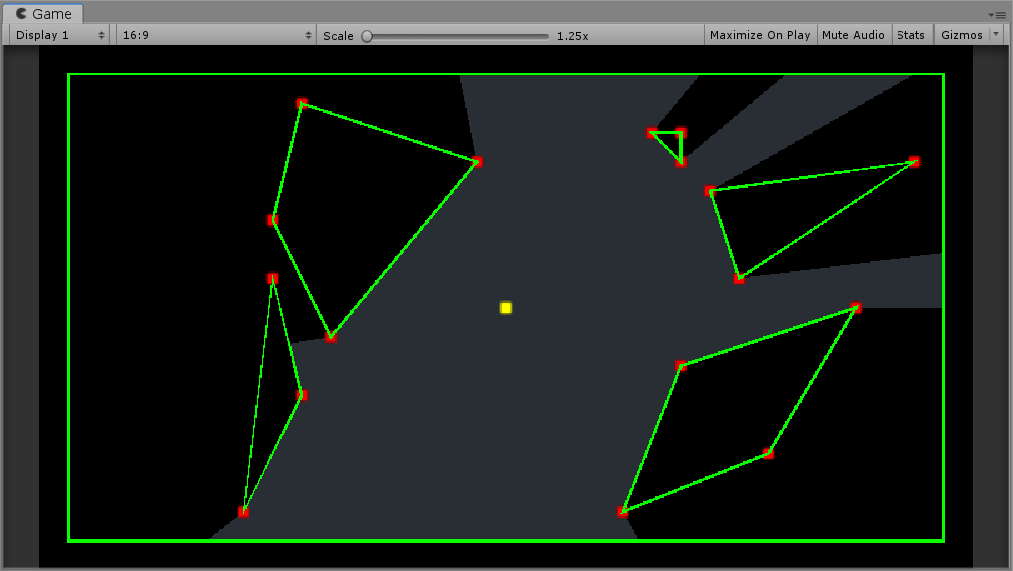
# Test Case

## 4.1 Convex Polygon Obstacle

a)



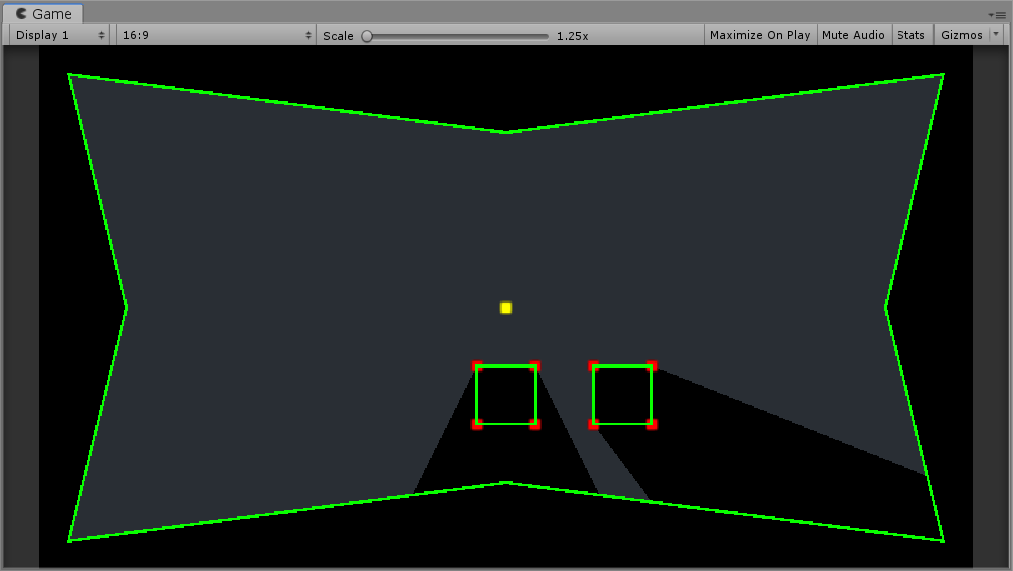
The polygon surrounded by the red points are obstacle and the view point is denoted by yellow point. The blue points are the critical point.



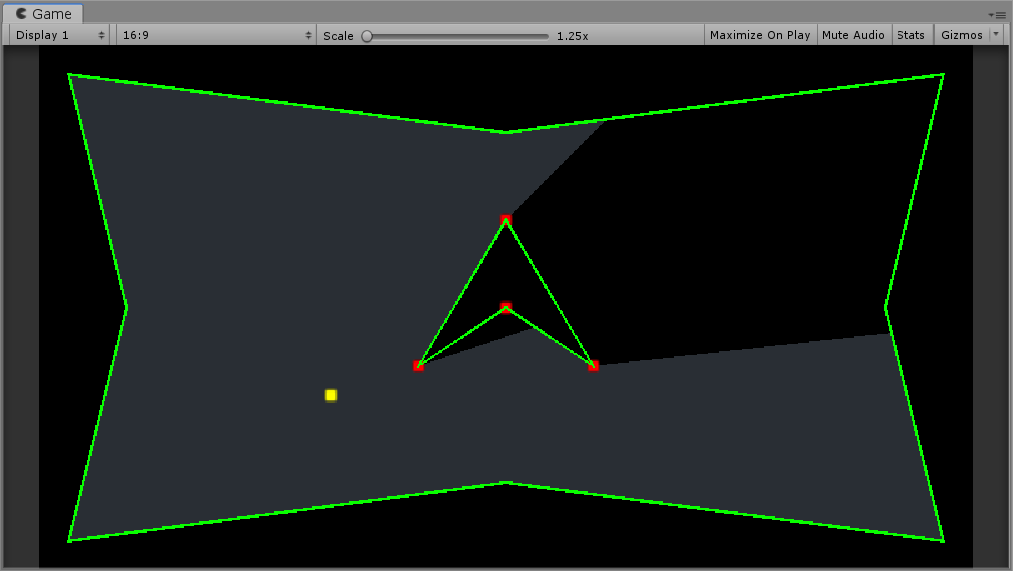
The gray area is the visibility effect of that view point without sight range or sight angle restrictions.

## 4.2 Concave Polygon

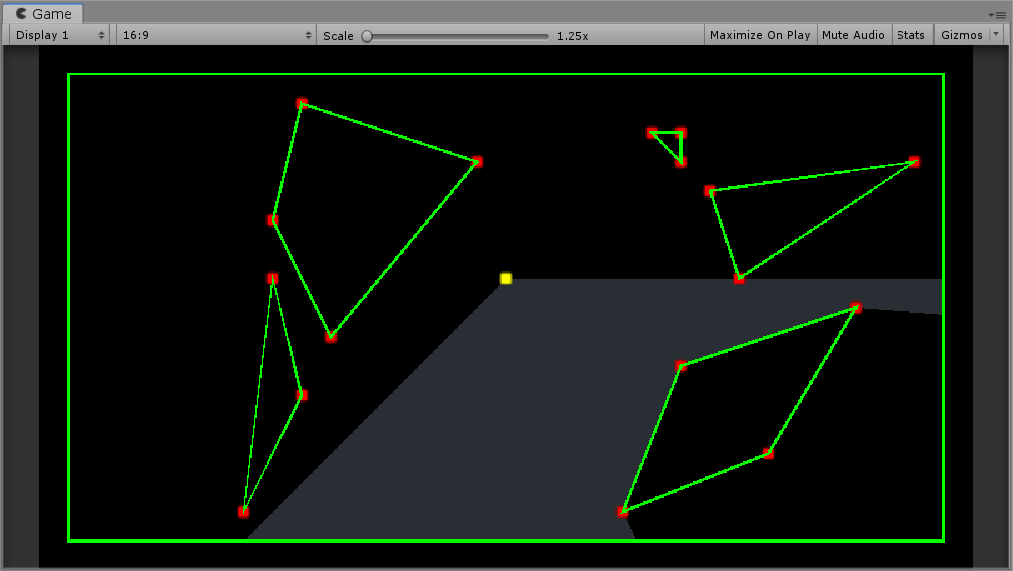
### 4.2.1 Concave Boundary



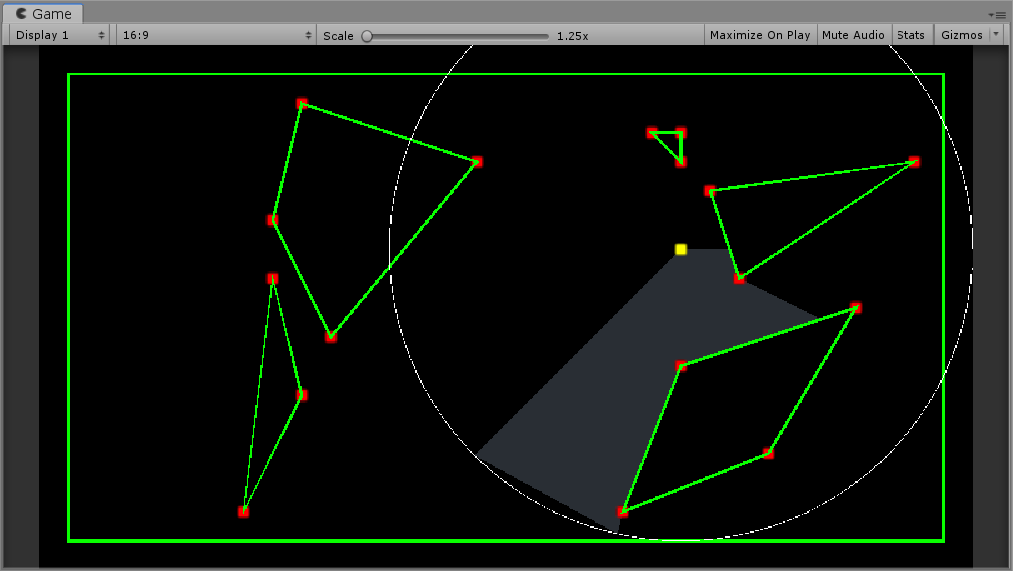
### 4.2.2 Concave Obstacle



## 4.3 Partially View



## 4.4 Visibility With Range Limitation



# Conclusion(impact)

# Reference

* 1. define visiblity star shape polygon citation
  2. Art gallary book Ason’ algorithm