GTU Department of Computer Engineering

CSE 222 / 505 – Spring 2022

Homework 1

Ahmet USLUOGLU

1801042602

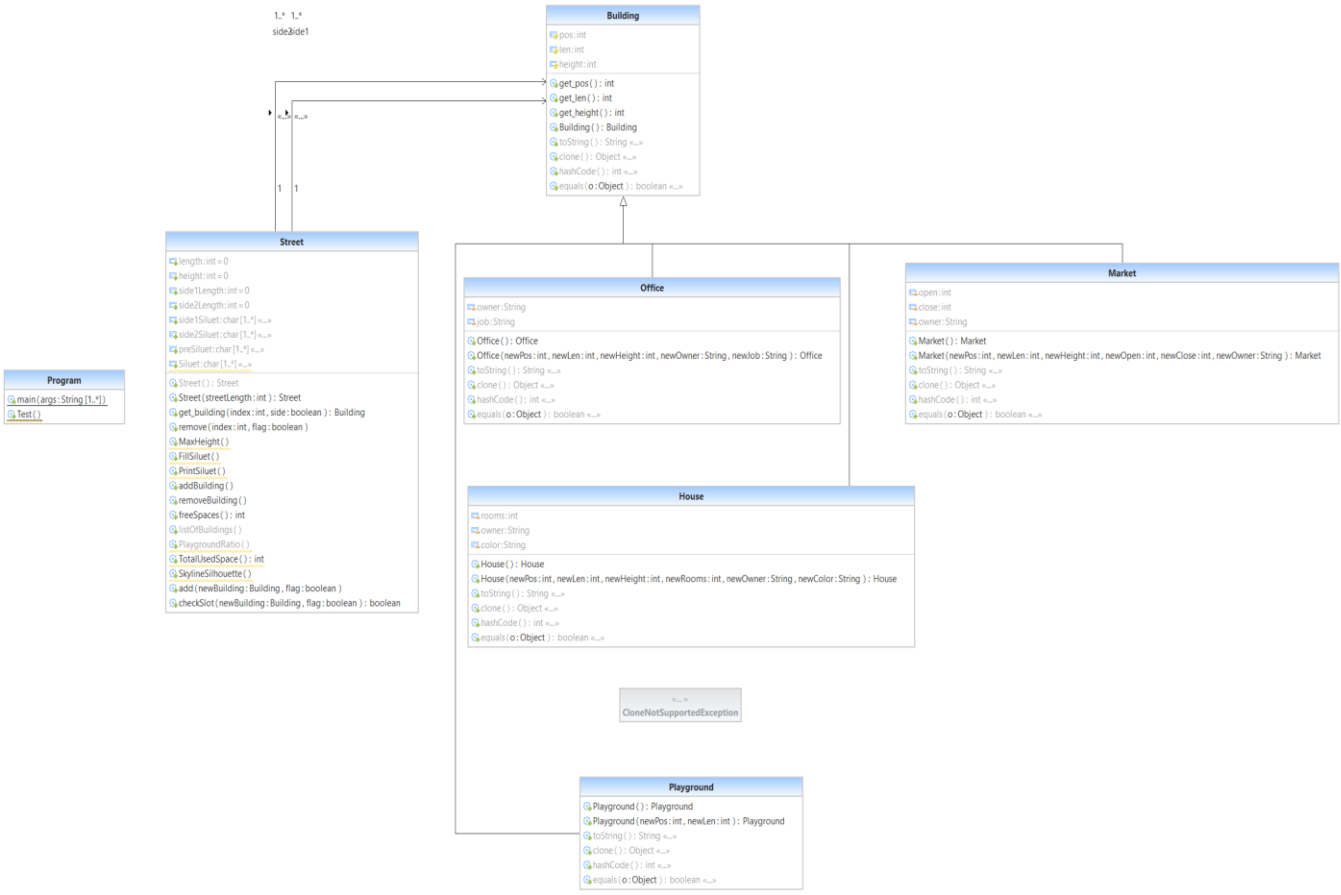
1 – System Requirement

Operating System must have JDK (Java Development Kit) 11 and JRE (Java Runtime Enviroment) 11 or higher.

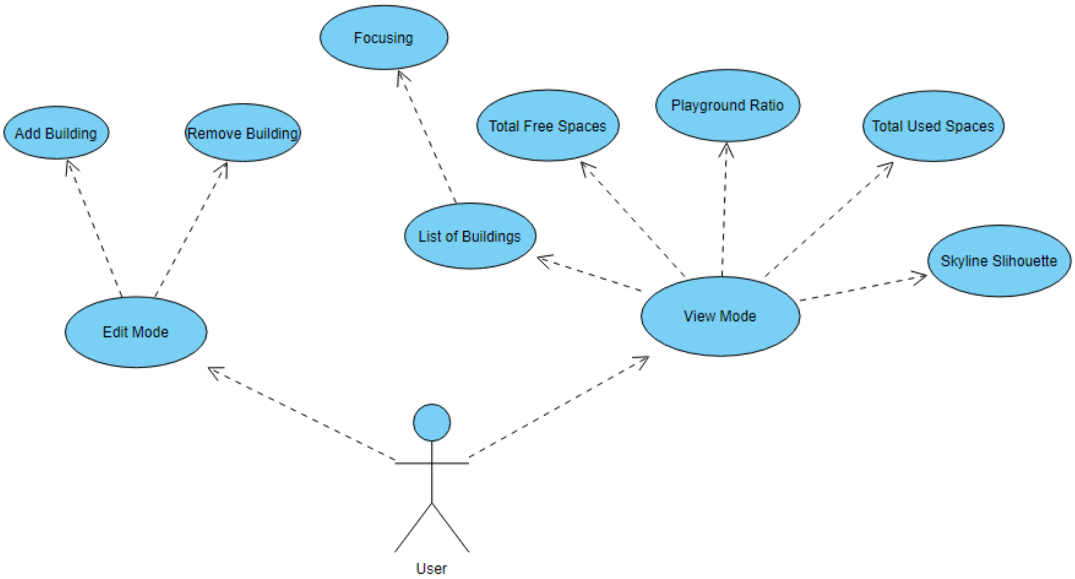
There should be enough space for storing datas.

User must create a Street with a fixed length at the start of the program. User must enter correct input values while adding a new building.

2 – Class Diagrams


1..* 1..*
side2side1
«...» «...»
1
1
Building
pos:int
len:int
height:int
get_pos ( ) : int
get_len ( ) : int
get_height ( ) : int
Building ( ) : Building
toString ( ) : String «...»
clone ( ) : Object «...»
hashCode ( ) : int «...»
equals (o : Object) : boolean «...»
Street
length:int = 0
height:int = 0
side1Length:int = 0
side2Length:int = 0
side1Siluet:char [1..*] «...»
side2Siluet:char [1..*] «...»
preSiluet:char [1..*] «...»
Siluet:char [1..*] «...»
Street ( ) : Street
Street (streetLength : int) : Street
get_building (index : int , side : boolean) : Building
remove (index : int , flag : boolean)
MaxHeight ( )
FillSiluet ( )
PrintSiluet ( )
addBuilding ( )
removeBuilding ( )
freeSpaces ( ) : int
listOfBuildings ( )
PlaygroundRatio ( )
TotalUsedSpace ( ) : int
SkylineSilhouette ( )
add (newBuilding : Building , flag : boolean)
checkSlot (newBuilding : Building , flag : boolean) : boolean
Program
main (args : String [1..*])
Test ( )
Office
owner:String
job:String
Office ( ) : Office
Office (newPos : int , newLen : int , newHeight : int , newOwner : String , newJob : String) : Office
toString ( ) : String «...»
clone ( ) : Object «...»
hashCode ( ) : int «...»
equals (o : Object) : boolean «...»
Market
open:int
close:int
owner:String
Market ( ) : Market
Market (newPos : int , newLen : int , newHeight : int , newOpen : int , newClose : int , newOwner : String) : Market
toString ( ) : String «...»
clone ( ) : Object «...»
hashCode ( ) : int «...»
equals (o : Object) : boolean «...»
House
rooms:int
owner:String
color:String
House ( ) : House
House (newPos : int , newLen : int , newHeight : int , newRooms : int , newOwner : String , newColor : String) : House
toString ( ) : String «...»
clone ( ) : Object «...»
hashCode ( ) : int «...»
equals (o : Object) : boolean «...»
«... »
CloneNotSupportedException
Playground
Playground ( ) : Playground
Playground (newPos : int , newLen : int) : Playground
toString ( ) : String «...»
clone ( ) : Object «...»
hashCode ( ) : int «...»
equals (o : Object) : boolean «...»

2.1 – Use Case Diagrams



3. Problem Solution Approach

Problem: Designing a small Street planning software with the feature of Slihouette of Skyline. Street consists of 2 sides. Each side has free spaces to construct a building. Street’s Length is fixed and space is limited. Construct buildings that does not conflicts with other buildings at the same side.

Solution: The buildings that can be constructed (House, Office, Market, Playground) on the Street inherited from a Building class which has length, postion and height properities.

I have then created a Street class to oversee all related functions that will be needed to construct buildings on the street.

I have stored the buildings in 2 separate arrays respective to each side of the Street.

Each side of the Street has 1 Building array to store buildings information and 1 char array to store slihouette of the side.

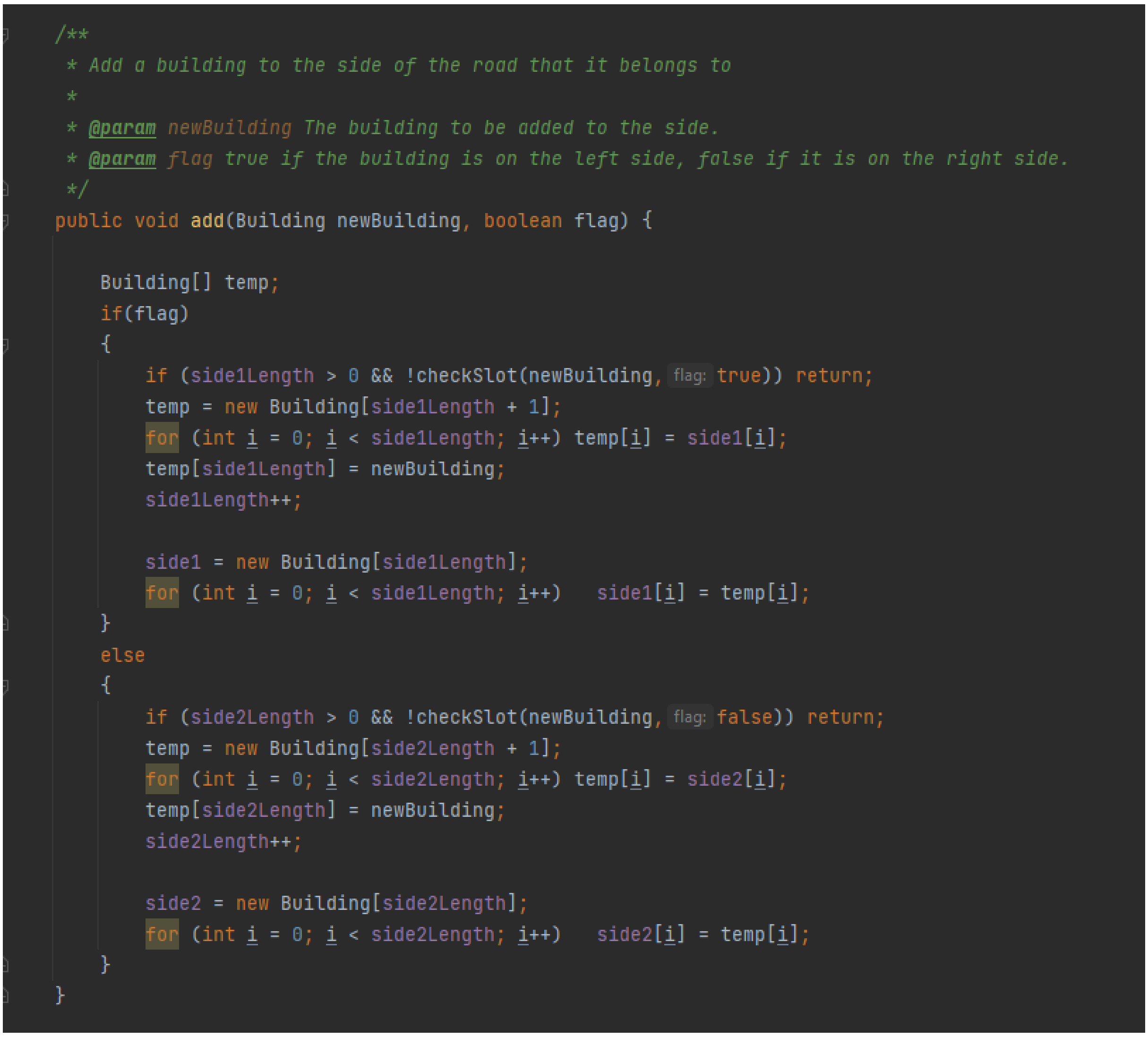
After creating 2 different char array with slihouettes i have combined them to 1 as preSlihouette array by only adding the parts where both of the arrays are empty.

The Empty points in the preSlihouette array is not finished. I have reversed the array and only printed the walls of the buildings at the outer most side. Thus Skyline Slihouette is completed.

Other functions are simple calculation functions that sums up total lengths, empty spaces, used spaces etc.

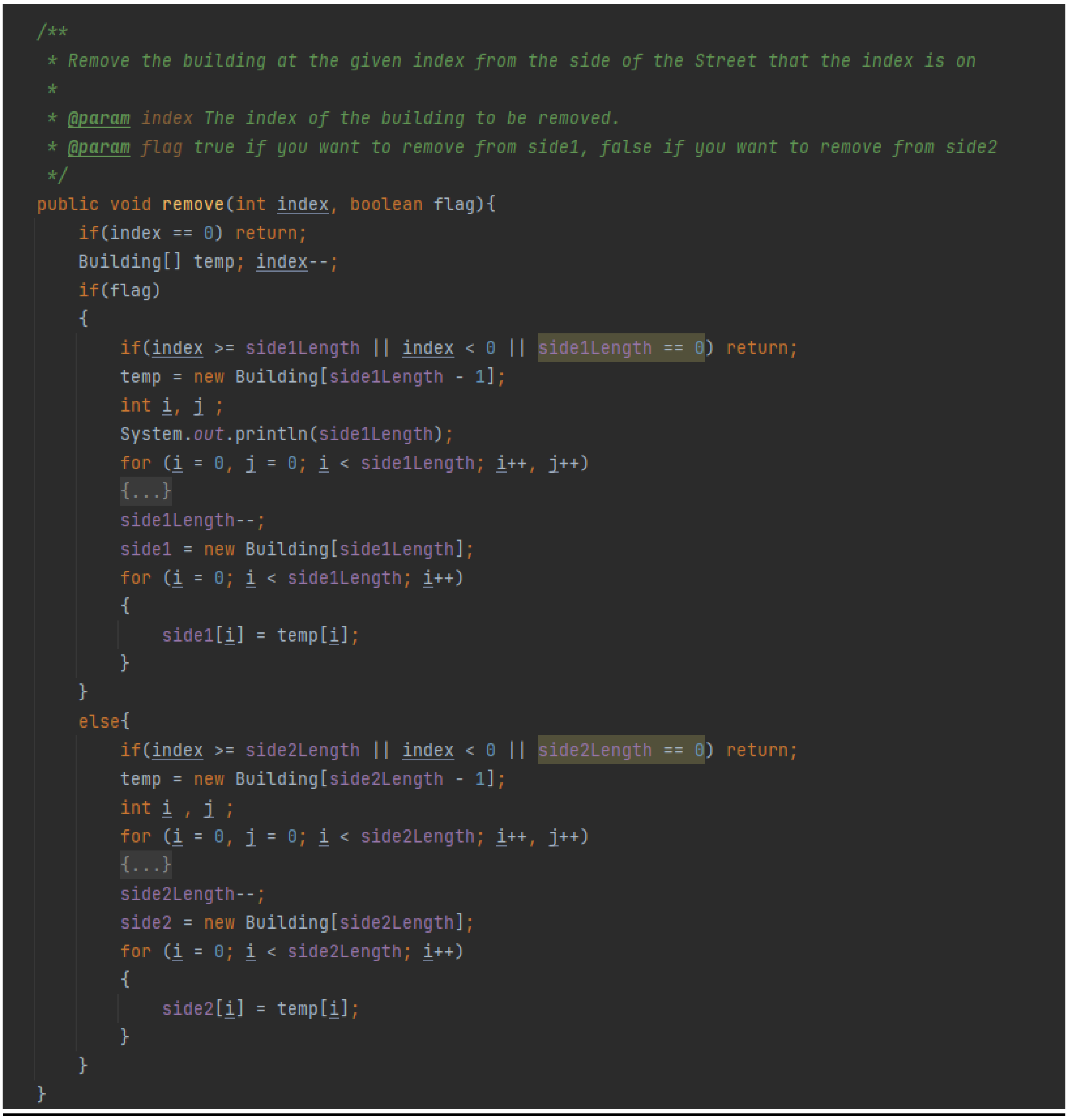
Exceptions are handled in case of invalid values for inputs are given.

Example of Add Building Function



```java
/**
 * Add a building to the side of the road that it belongs to
 *
 * @param newBuilding The building to be added to the side.
 * @param flag true if the building is on the left side, false if it is on the right side.
 */
public void add(Building newBuilding, boolean flag) {

    Building[] temp;
    if(flag)
    {
        if (side1Length > 0 && !checkSlot(newBuilding, flag: true)) return;
        temp = new Building[side1Length + 1];
        for (int i = 0; i < side1Length; i++) temp[i] = side1[i];
        temp[side1Length] = newBuilding;
        side1Length++;

        side1 = new Building[side1Length];
        for (int i = 0; i < side1Length; i++)   side1[i] = temp[i];
    }
    else
    {
        if (side2Length > 0 && !checkSlot(newBuilding, flag: false)) return;
        temp = new Building[side2Length + 1];
        for (int i = 0; i < side2Length; i++) temp[i] = side2[i];
        temp[side2Length] = newBuilding;
        side2Length++;

        side2 = new Building[side2Length];
        for (int i = 0; i < side2Length; i++)   side2[i] = temp[i];
    }
}
```

Example of Remove Building Function



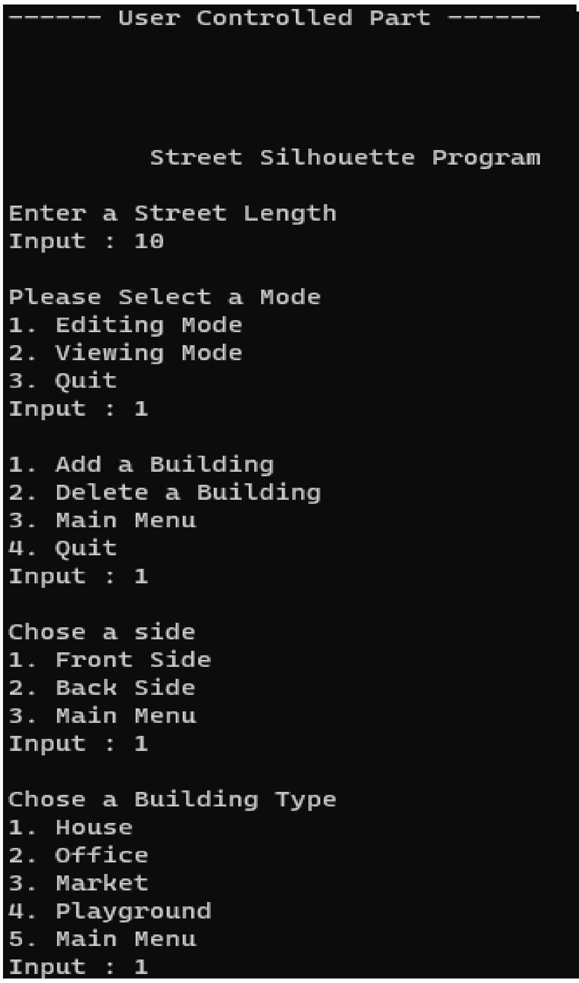
```java
/**
 * Remove the building at the given index from the side of the Street that the index is on
 *
 * @param index The index of the building to be removed.
 * @param flag true if you want to remove from side1, false if you want to remove from side2
 */
public void remove(int index, boolean flag){
    if(index == 0) return;
    Building[] temp; index--;
    if(flag)
    {
        if(index >= side1Length || index < 0 || side1Length == 0) return;
        temp = new Building[side1Length - 1];
        int i, j ;
        System.out.println(side1Length);
        for (i = 0, j = 0; i < side1Length; i++, j++)
        {...}
        side1Length--;
        side1 = new Building[side1Length];
        for (i = 0; i < side1Length; i++)
        {
            side1[i] = temp[i];
        }
    }
    else{
        if(index >= side2Length || index < 0 || side2Length == 0) return;
        temp = new Building[side2Length - 1];
        int i , j ;
        for (i = 0, j = 0; i < side2Length; i++, j++)
        {...}
        side2Length--;
        side2 = new Building[side2Length];
        for (i = 0; i < side2Length; i++)
        {
            side2[i] = temp[i];
        }
    }
}
```

4 – Test Cases

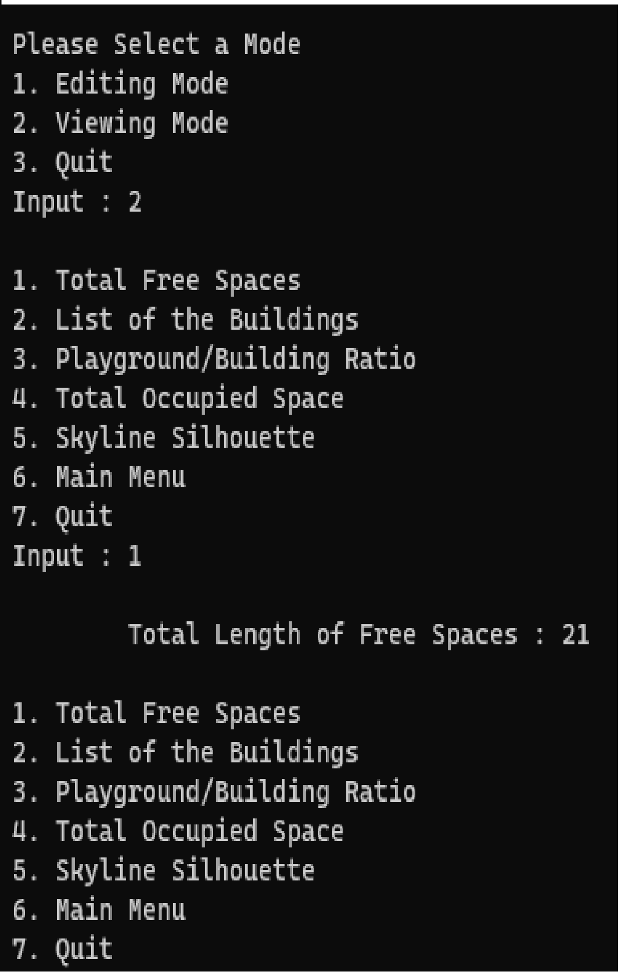
1- Create a Street
2- Create and Add Buildings
3- Display Slihouette
4- Display Details
5- Remove Buildings
6- Display Total Free Spaces
7- Display Total Used Spaces
8- Display Playground to Buildings Ratio
9- Print List of the Bulidings

5 – Running Program and Results

Firstly User Interface Menus

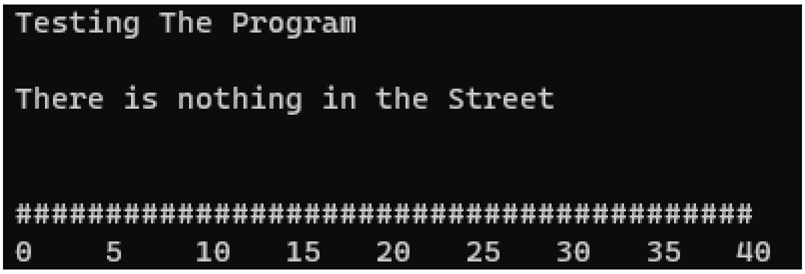


```
------ User Controlled Part ------



         Street Silhouette Program

Enter a Street Length
Input : 10

Please Select a Mode
1. Editing Mode
2. Viewing Mode
3. Quit
Input : 1

1. Add a Building
2. Delete a Building
3. Main Menu
4. Quit
Input : 1

Chose a side
1. Front Side
2. Back Side
3. Main Menu
Input : 1

Chose a Building Type
1. House
2. Office
3. Market
4. Playground
5. Main Menu
Input : 1
```



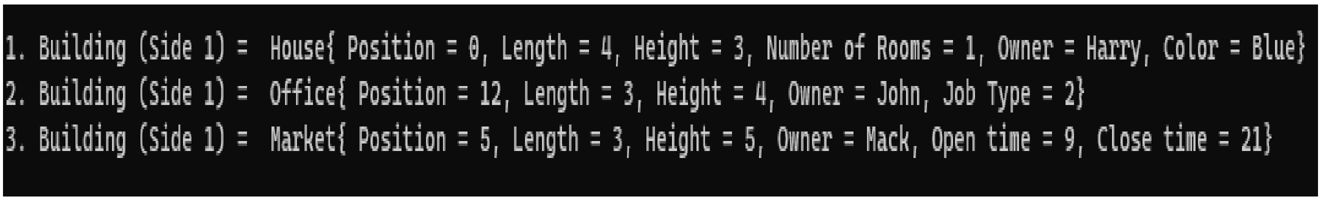
```
Please Select a Mode
1. Editing Mode
2. Viewing Mode
3. Quit
Input : 2

1. Total Free Spaces
2. List of the Buildings
3. Playground/Building Ratio
4. Total Occupied Space
5. Skyline Silhouette
6. Main Menu
7. Quit
Input : 1

        Total Length of Free Spaces : 21

1. Total Free Spaces
2. List of the Buildings
3. Playground/Building Ratio
4. Total Occupied Space
5. Skyline Silhouette
6. Main Menu
7. Quit
```

1 – Create a Street



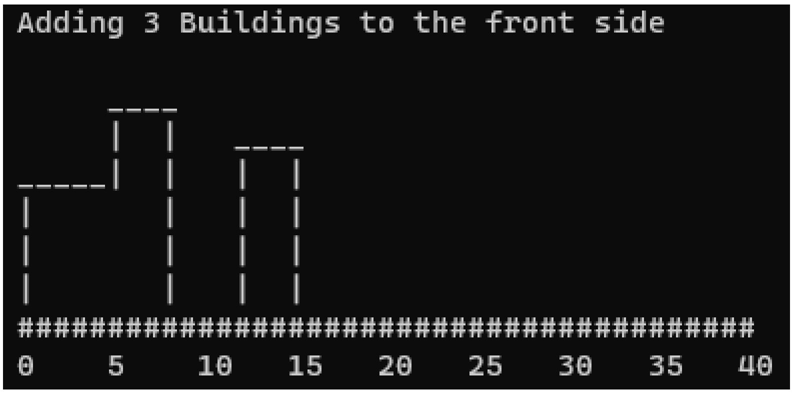
```
Testing The Program

There is nothing in the Street


#########################################
0    5    10   15   20   25   30   35   40
```

2 – Create And Add Buldings



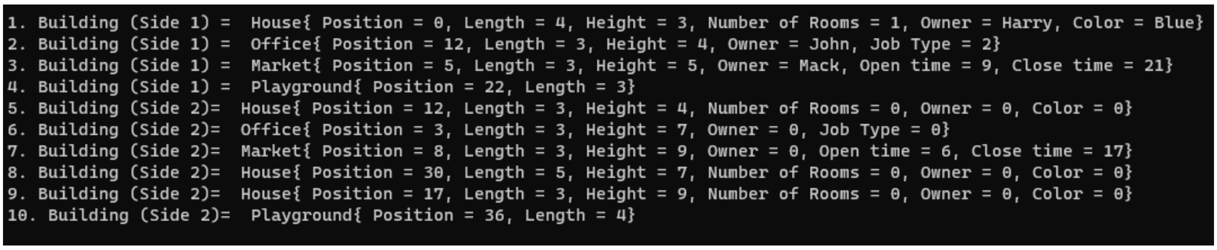
```
1. Building (Side 1) =  House{ Position = 0, Length = 4, Height = 3, Number of Rooms = 1, Owner = Harry, Color = Blue}
2. Building (Side 1) =  Office{ Position = 12, Length = 3, Height = 4, Owner = John, Job Type = 2}
3. Building (Side 1) =  Market{ Position = 5, Length = 3, Height = 5, Owner = Mack, Open time = 9, Close time = 21}
```

3 – Display Slihouette



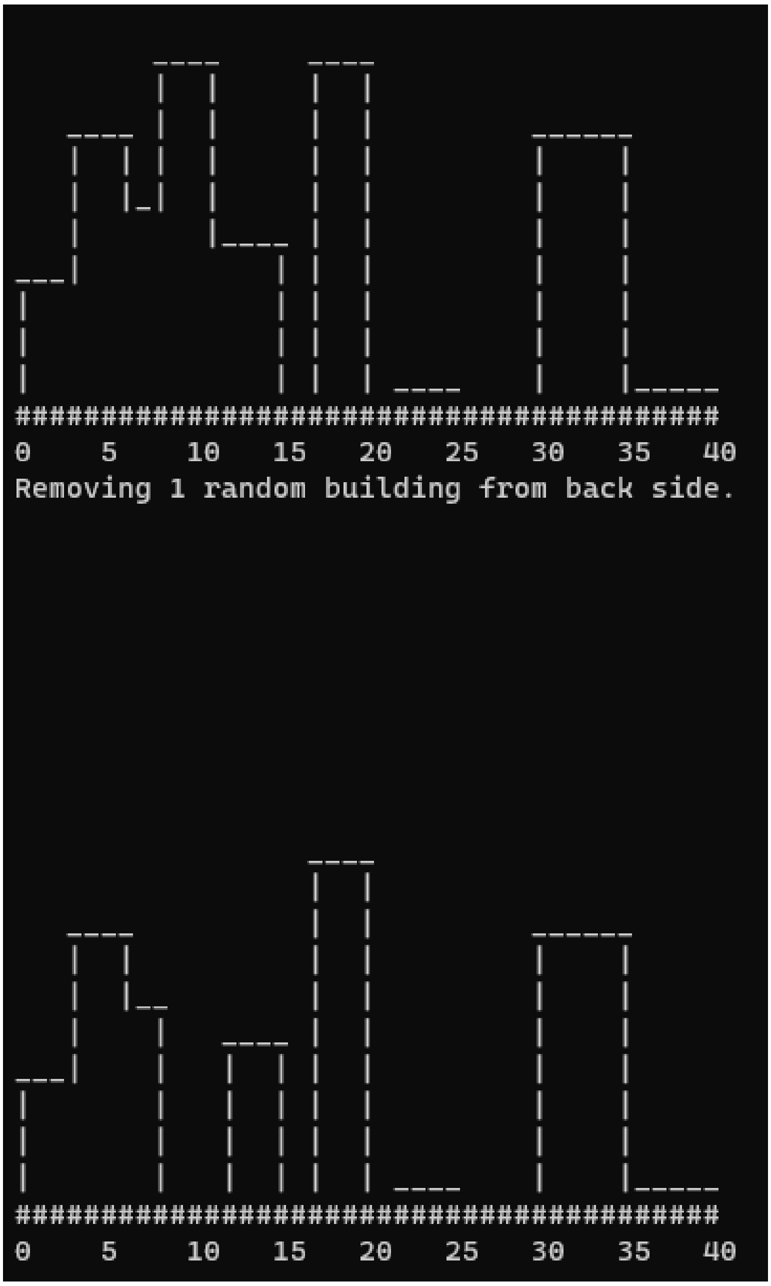
```
Adding 3 Buildings to the front side


     ____
     |  |
     |  |   ____
_____|  |   |  |
|       |   |  |
|       |   |  |
|       |   |  |
#########################################
0    5    10   15   20   25   30   35   40
```

4- Display Details



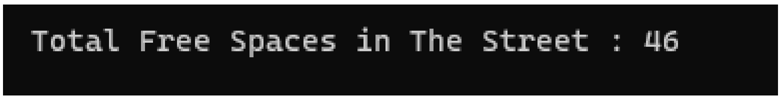
```
1. Building (Side 1) =  House{ Position = 0, Length = 4, Height = 3, Number of Rooms = 1, Owner = Harry, Color = Blue}
2. Building (Side 1) =  Office{ Position = 12, Length = 3, Height = 4, Owner = John, Job Type = 2}
3. Building (Side 1) =  Market{ Position = 5, Length = 3, Height = 5, Owner = Mack, Open time = 9, Close time = 21}
4. Building (Side 1) =  Playground{ Position = 22, Length = 3}
5. Building (Side 2)=  House{ Position = 12, Length = 3, Height = 4, Number of Rooms = 0, Owner = 0, Color = 0}
6. Building (Side 2)=  Office{ Position = 3, Length = 3, Height = 7, Owner = 0, Job Type = 0}
7. Building (Side 2)=  Market{ Position = 8, Length = 3, Height = 9, Owner = 0, Open time = 6, Close time = 17}
8. Building (Side 2)=  House{ Position = 30, Length = 5, Height = 7, Number of Rooms = 0, Owner = 0, Color = 0}
9. Building (Side 2)=  House{ Position = 17, Length = 3, Height = 9, Number of Rooms = 0, Owner = 0, Color = 0}
10. Building (Side 2)=  Playground{ Position = 36, Length = 4}
```

5 - Remove Buildings



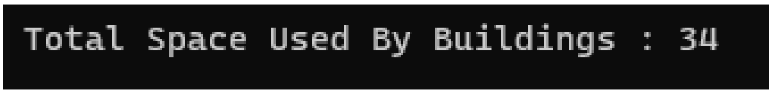
```
        ____     ____
        |  |     |  |
   ____ |  |     |  |         ______
   |  | |  |     |  |         |    |
   |  |_|  |     |  |         |    |
   |       |____ |  |         |    |
___|           | |  |         |    |
|              | |  |         |    |
|              | |  |         |    |
|              | |  | ____    |    |_____
#########################################
0    5    10   15   20   25   30   35   40
Removing 1 random building from back side.




                 ____
                 |  |
   ____          |  |         ______
   |  |          |  |         |    |
   |  |__        |  |         |    |
   |    |   ____ |  |         |    |
___|    |   |  | |  |         |    |
|       |   |  | |  |         |    |
|       |   |  | |  |         |    |
|       |   |  | |  | ____    |    |_____
#########################################
0    5    10   15   20   25   30   35   40
```

6 – Display Total Free Spaces



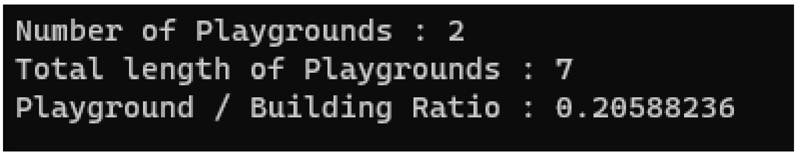
```
 Total Free Spaces in The Street : 46
```

7 – Display Total Used Spaces



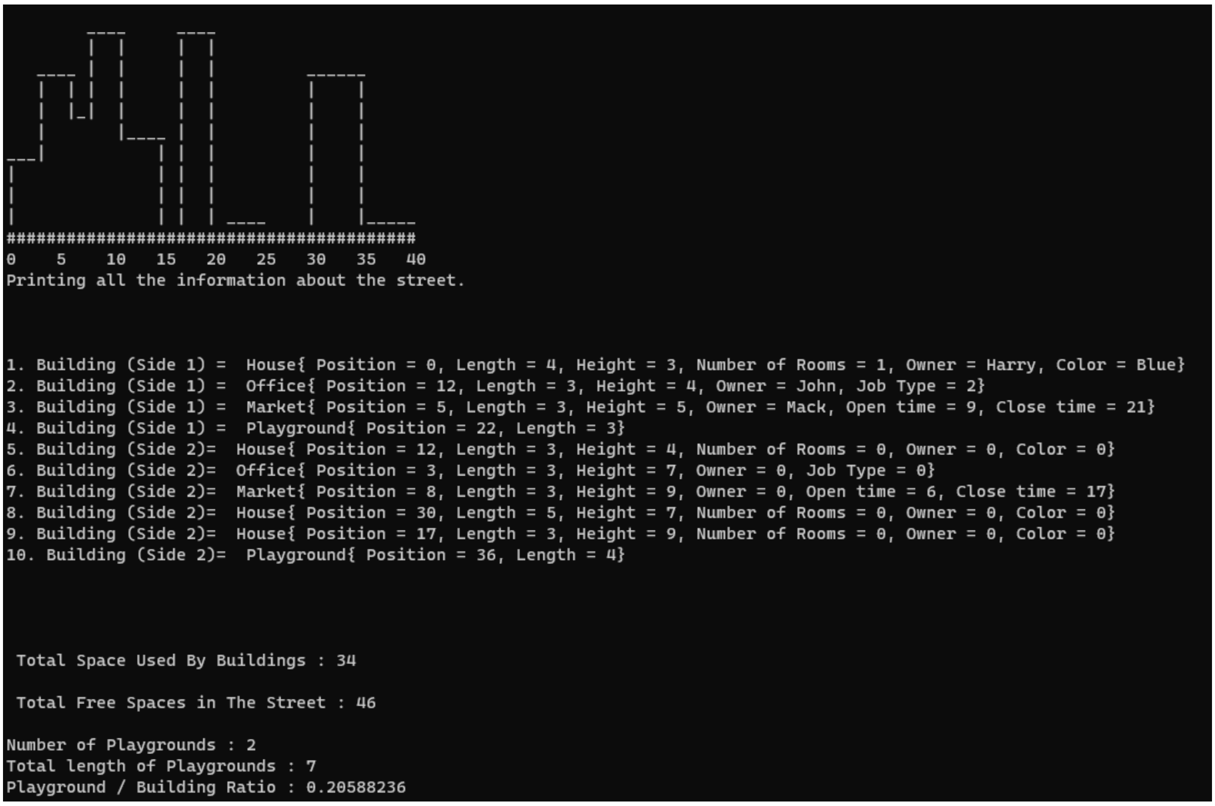
```
 Total Space Used By Buildings : 34
```

8 – Display Playground Ratio



```
Number of Playgrounds : 2
Total length of Playgrounds : 7
Playground / Building Ratio : 0.20588236
```

9 – Display List of Buildings



```
        ____     ____
        |  |     |  |
   ____ |  |     |  |         ______
   |  | |  |     |  |         |    |
   |  |_|  |     |  |         |    |
   |       |____ |  |         |    |
___|           | |  |         |    |
|              | |  |         |    |
|              | |  |         |    |
|              | |  | ____    |    |_____
#########################################
0    5    10   15   20   25   30   35   40
Printing all the information about the street.



1. Building (Side 1) =  House{ Position = 0, Length = 4, Height = 3, Number of Rooms = 1, Owner = Harry, Color = Blue}
2. Building (Side 1) =  Office{ Position = 12, Length = 3, Height = 4, Owner = John, Job Type = 2}
3. Building (Side 1) =  Market{ Position = 5, Length = 3, Height = 5, Owner = Mack, Open time = 9, Close time = 21}
4. Building (Side 1) =  Playground{ Position = 22, Length = 3}
5. Building (Side 2)=  House{ Position = 12, Length = 3, Height = 4, Number of Rooms = 0, Owner = 0, Color = 0}
6. Building (Side 2)=  Office{ Position = 3, Length = 3, Height = 7, Owner = 0, Job Type = 0}
7. Building (Side 2)=  Market{ Position = 8, Length = 3, Height = 9, Owner = 0, Open time = 6, Close time = 17}
8. Building (Side 2)=  House{ Position = 30, Length = 5, Height = 7, Number of Rooms = 0, Owner = 0, Color = 0}
9. Building (Side 2)=  House{ Position = 17, Length = 3, Height = 9, Number of Rooms = 0, Owner = 0, Color = 0}
10. Building (Side 2)=  Playground{ Position = 36, Length = 4}




 Total Space Used By Buildings : 34

 Total Free Spaces in The Street : 46

Number of Playgrounds : 2
Total length of Playgrounds : 7
Playground / Building Ratio : 0.20588236
```