Building 3D Models using the Data stored in a Relational Database

B M S Banduthilaka
Department of Civil and Environmental Engineering
Faculty of Engineering
University of Ruhuna
Galle
Sri Lanka
sachindrabanduthilake@gmail.com

C Kariyawasam
Department of Electrical and Information Engineering
Faculty of Engineering
University of Ruhuna
Galle
Sri Lanka

A M N Alagiyawanna
aDepartment of Civil and Environmental Engineering
Faculty of Engineering
University of Ruhuna
Galle
Sri Lanka

Abstract

3D (Three dimensional) maps can be used to express details of a set of objects scattered over an area. Generally, 3D development is done manually which is not convenient for larger scale models such as 3D maps. The purpose of this research was to find solutions for the problems encountered during the manual development of 3D maps. It was found that by automating the modeling process, majority of these difficulties can be mitigated. During practical adaptation of the automation process, several Ruby scripts were written. These scripts included methods to develop 3D models using modeling software. Attributes and parameters of real world items were stored in a Relational Database Management System. These data were retrieved during the runtime of the scripts and the relevant 3D models were developed. To illustrate the automated procedure, an interactive map was developed using Sketchup, Ruby and MySQL.

Key words: 3D models, Databases, Interactive map, Sketchup

1. Introduction

Interactive map is a special kind of a data accessing method which used to display geo spatial data. Google map is a popular 2D (2 Dimensional) map system.

Conversely three dimensional models can be used to demonstrate the physical appearance of an object. It has many advantages compared to two dimensional models. A cluster of 3D (3 dimensional) models can be used to display a group of objects scattered over area. This concept has been used to develop 3D maps over a particular area and demonstrate the architectural importance of the structures over the terrain (Salamanca, Ospina and Vargas, 2013).

When considering the representation of information of objects in a geographical location, 3D models can provide more information, compared with images and other 2D media. 3D interactive maps can be used to display information of historical monuments located in heritage sites. 3D map illustrates the relevant 3D model of that particular area. 3D models of historical monuments can be used effectively to visualize the architectural and historical information of these monuments.

Currently many organizations are working on developing 3D interactive maps for multitude of purposes. Majority of them encounter difficulties in developing a large number of 3D objects in an interactive map simultaneously. Manual development of large 3D interactive map is a difficult and cumbersome task. Depending on the situation a trivial mistake during the developing procedure might require recreating the complete map.

The purpose of this research is to address these issues in manual development of 3D models in an interactive map.

2. Objectives

Sharing details through 3D models can be considered as an attentive idea with several difficulties. A 3D map consists of several 3D models, scattered over a large area. These 3D models can be used to represent houses, important historical and architectural monuments and even the landscape (Salamanca, Ospina and Vargas, 2013).

Developing a large number of individual 3D models will require time and effort. Since this is a map, the geo graphical location of these objects has to be precise relative to other 3D objects in the map. The correlation among these objects has to be taken account to illustrate an accurate picture of the particular area. Therefore the orientation of these individual models needs to be considered. Some of these models may share identical objects like columns, stairs, windows and doors. During manual process these identical items has to be developed individually at each location which consumes a lot of time and effort. During the manual process of a 3D map, the possibility of occurring errors are high and addressing errors is a cumbersome task since the number of models are high.

These practical difficulties in manual development of 3D maps demands a more convenient and easy method for constructing 3D individual models. As a result, it was identified that hese difficulties can be eliminated by automating the development process. The objective of this research was to study the feasibilities of automating of developing 3D interactive map.

Most of the above issues can be addressed during automation as listed below.

- Identical objects of the models can be easily rebuilt.
- Precise locations of the individual 3D models and their orientation can be fed to the 3D model during the run time.
- Reduced time and effort.
- Erroneous entities can be easily identified and resolved.

3. Methodology

The methodology was broken into six steps.

1. Identify the best available 3D modeling software.
2. Select the pilot area.
3. Identify the potential candidates for automation.
4. Develop the database.
5. Develop the 3D map.
6. Develop the interactive component.

Initialization

Development of the 3D interactive map

The details of the execution of the above steps are discussed below.

4. Identifying the best 3D Modeling Software

Currently several 3D modeling software products are available in the market. Capabilities and limitations of these products were studied to select the most suitable 3D modeling tool to automate the 3D development process. Among them Google Sketchup was identified as the best option in developing 3D models considering the cost and the functionalities it offers.

Google Sketchup is a 3D modeling tool used in civil, architectural, mechanical and game applications Peter, Paul & Drury, 2008). This program includes a tool panel which allows users to develop their models in a pre defined layout. Various styles and effects can be added to the model using the tool box. One of the most innovative features in Sketchup is the “Pushpull” functionality which can be used to add a third dimension to a two dimensional objects.

4.1 Ruby API (Application Programming Interface)

The specialty of Skethcup is that the 3D development process can be done either manually or programmatically. The 3D development process can be automated via a Ruby API (Application Program Interface). Ruby is a dynamic, reflective, general-purpose object oriented programming language similar to Java. It has several innovative features which are more advanced compared to Java (Peter, Paul and Drury 2008). These Ruby features which are called extensions enable to embed language capabilities such as instantiation, inheritance etc through Ruby API.

Ruby API consists of special Ruby functions which are used to invoke Sketchup tools. These invoked functions can be used to develop 3D models. Sketchup can implement the Ruby language OOP (Object Oriented Programming) capabilities using this method. Also a MySQL database can be accessed through Sketchup using the embedded Ruby API. It means 3D models can be modified merely by changing the relevant parameters in the database.

5. Selection of the Pilot area

To demonstrate the concept, Galle Fort, one of the best preserved examples of 17th century coastal fortifications in Sri Lanka was selected as the pilot area. Galle Fort is located in Galle (6.025833°N 80.2175°E), in Southern Sri Lanka facing the Indian Ocean. Galle Fort is adjacent to Galle harbor. Despite being credited as a Dutch structure, it was actually the Portuguese who started its construction. After the Dutch rule, the Fort was also maintained and operated by British (Nelson, 1984). It is concluded that Galle Fort has been operated and maintained by three nations which had diverse architectural values. Galle Fort contains buildings with historical importance, bastions, gun points, dungeons etc. These features make Galle Fort an ideal pilot area to demonstrate the concept.

6. Development of the 3D Map

6.1 Storing of parameters and attributes.

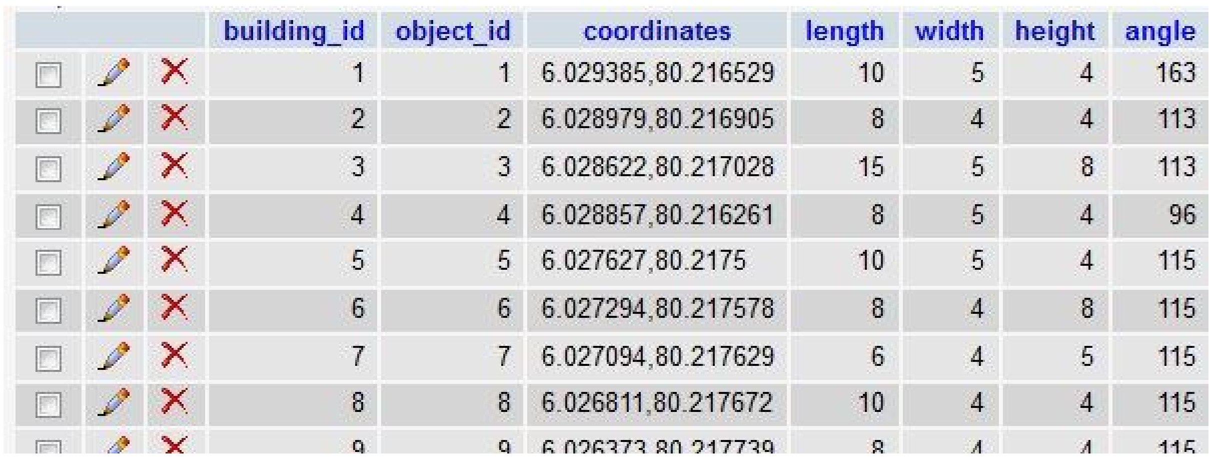
As discussed earlier, the 3D map can be considered as a cluster of 3D models. An individual 3D model represents a structure in particular area. To develop the 3D model of a structure, the physical parameters of the structure, its geographical location and orientation were needed. As an example to develop 3D model of a building, the height, length and width of the structure, the coordinate of the building location and orientation were needed. This type of data was stored in a MySQL database.

6.2 Data extraction.

The objective of storing data in a database was to facilitate data extraction during runtime. When a building was developed using the automated process, physical parameters such as coordinates of the building, height and length will be extracted from the database automatically (Refer Appendix A). This does not require manual intervention. This will also simplify editing 3D models. Instead of editing 3D model manually, it can be done at the database level.

6.3 Ruby extensions to automate the development process.

The development of the 3D map is done usng Ruby extensions. Ruby extensions were written using native Ruby language and Sketchup Ruby API functions. These extensions can be saved in the plugins directory of the Sketchup folder. These plugins can be executed by calling them from the Sketchup GUI (Graphical User Interface) (Sketchup Ruby API Documentation, 2008). During the research, several Ruby extensions were developed with several Ruby classes and a main Ruby program.



building_id	object_id	coordinates	length	width	height	angle
1	1	6.029385,80.216529	10	5	4	163
2	2	6.028979,80.216905	8	4	4	113
3	3	6.028622,80.217028	15	5	8	113
4	4	6.028857,80.216261	8	5	4	96
5	5	6.027627,80.2175	10	5	4	115
6	6	6.027294,80.217578	8	4	8	115
7	7	6.027094,80.217629	6	4	5	115
8	8	6.026811,80.217672	10	4	4	115
9	9	6.026373,80.217739	8	4	4	115

Figure 1 Data stored in the Building table

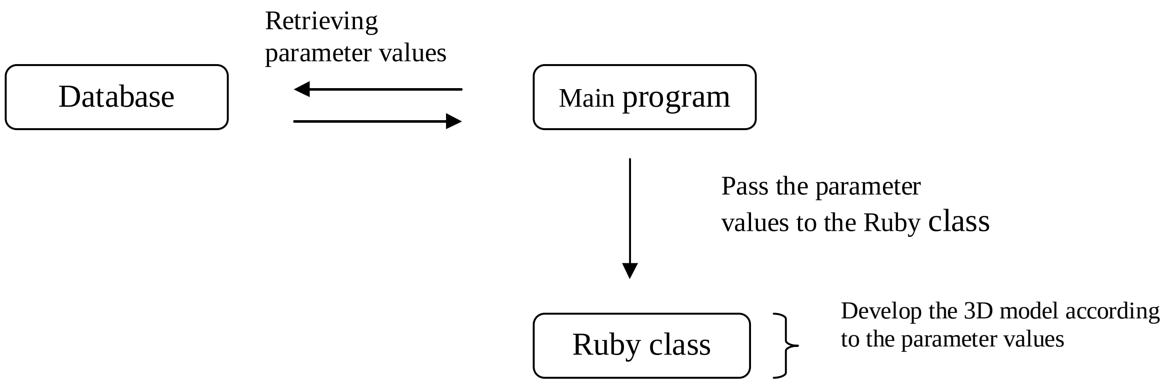

Retrieving parameter values
Database
Main program
Pass the parameter values to the Ruby class
Ruby class
Develop the 3D model according to the parameter values

Figure 2 Flow chart of Ruby extensions

6.4 Application of the automation procedure to pilot area.

The Ruby extensions consist of a main Ruby extension and several Ruby classes. The main Ruby program initiates the development of the 3D map. It creates a new Sketchup active model for the 3D map. It will connect to the MySQL database and retrieve the necessary construction details to the Ruby 3D development class (Refer Appendix A).

Three Ruby classes were developed to instantiate 3D models like buildings, gun points and bastions. These classes require input parameters. These input parameters will be passed to the Ruby classes at the object creation. The Ruby class consists of the instructions to build the 3D model. As an example, Building class has four input parameters namely model, corner coordinates, orientation, height and the building id (Refer Appendix B). These parameters will be passed to the building class at the time of the creation of the building object. The building class has the instructions to develop the 3D model of a building. It will use the values of these parameters and build the 3D model of the relevant building.

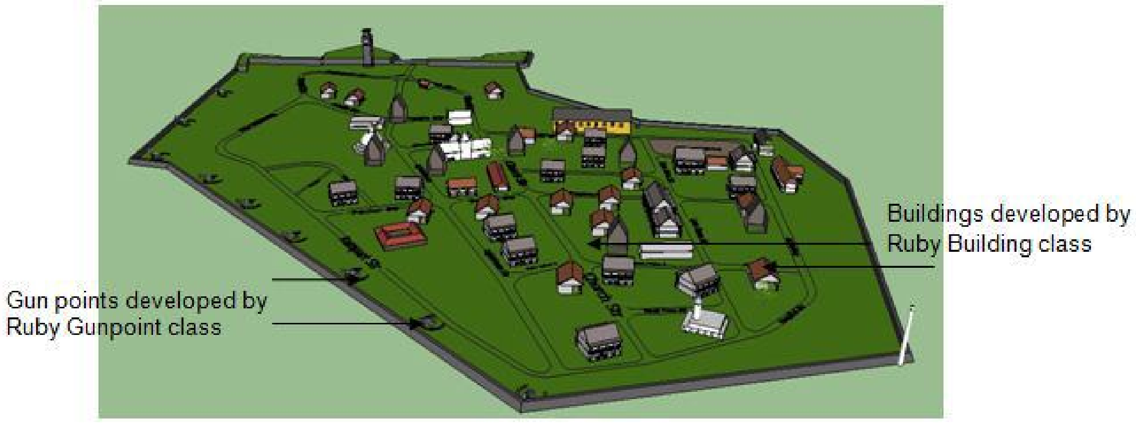

Buildings developed by Ruby Building class
Gun points developed by Ruby Gunpoint class

Figure 3 Three Dimensional interactive map

6.5 Developing identical components.

Generally structures have identical items such as stairs, columns, windows and doors. These are potential candidates for loop functions. Initially a single item was developed and later the loop functionality was used replicate the items at desired geographical locations. As an example after creating a single column, a "for" loop is applied to repeat it during particular number of times to get a line if columns (Collingbourne, 2008) (Refer Appendix C).

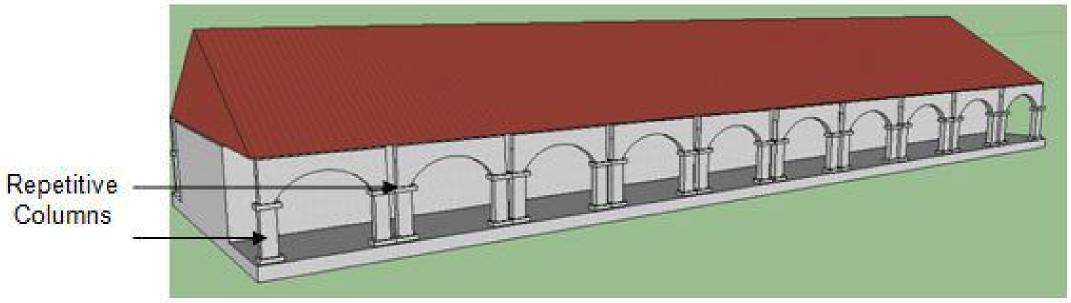
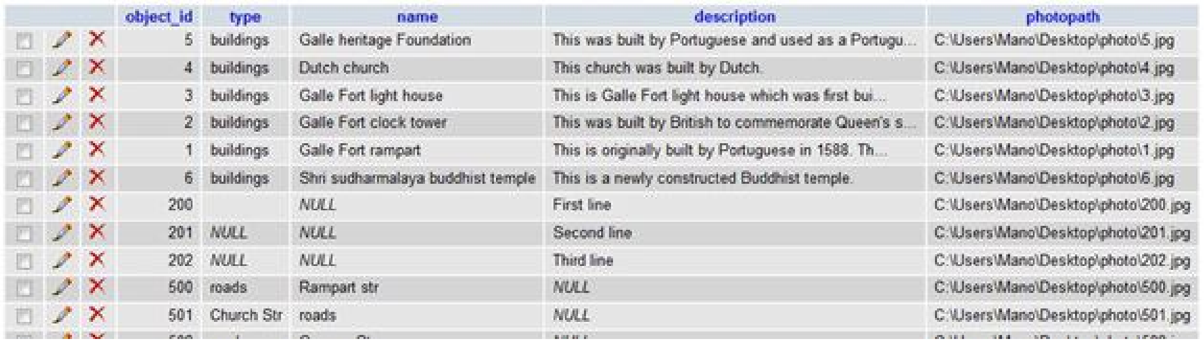


Figure 4 A 3D model with repetitive columns

6.6 Developing interactive component of the 3D map

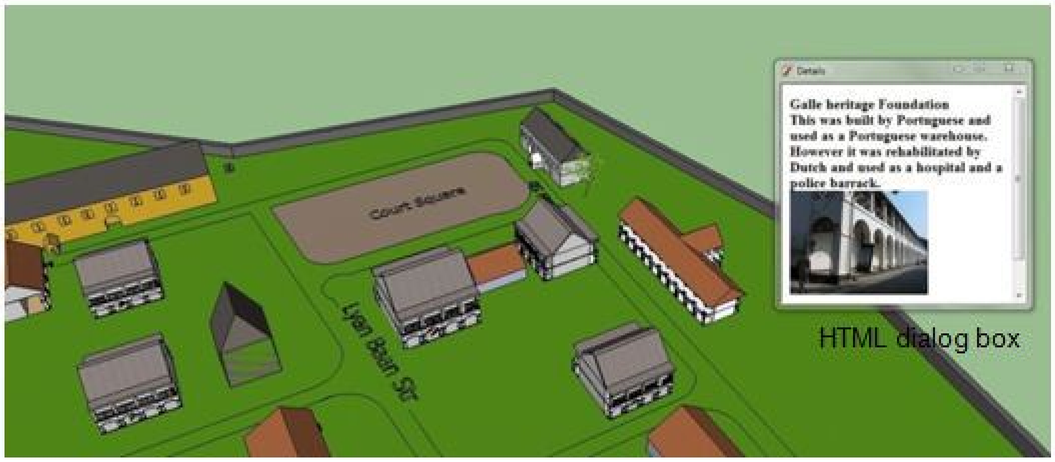
The MySQL database was created to store data pertaining to objects. The data contained in the database can be categorized into two groups according to their usage. Generally buildings, gunpoints and roads tables contain the details which will be needed during the model development process. This data will be extracted and used during runtime which means at the time of development of the 3D map. These details may bear no significant importance for general users including tourists.

Conversely historical and important information about the objects in the map are important to general users. In this project this kinds of general information about the structures were stored in a separate table in the database so that users can access these data. While interacting with the map, users have to trigger the on click action on the interested 3D model to retrieve information. Then the information related to that structure will be displayed on the screen in HTML dialog box (Refer Appendix D).



			object_id	type	name	description	photopath
☐	✎	✕	5	buildings	Galle heritage Foundation	This was built by Portuguese and used as a Portugu...	C:\Users\Mano\Desktop\photo\5.jpg
☐	✎	✕	4	buildings	Dutch church	This church was built by Dutch.	C:\Users\Mano\Desktop\photo\4.jpg
☐	✎	✕	3	buildings	Galle Fort light house	This is Galle Fort light house which was first bui...	C:\Users\Mano\Desktop\photo\3.jpg
☐	✎	✕	2	buildings	Galle Fort clock tower	This was built by British to commemorate Queen's s...	C:\Users\Mano\Desktop\photo\2.jpg
☐	✎	✕	1	buildings	Galle Fort rampart	This is originally built by Portuguese in 1588. Th...	C:\Users\Mano\Desktop\photo\1.jpg
☐	✎	✕	6	buildings	Shri sudharmalaya buddhist temple	This is a newly constructed Buddhist temple.	C:\Users\Mano\Desktop\photo\6.jpg
☐	✎	✕	200		NULL	First line	C:\Users\Mano\Desktop\photo\200.jpg
☐	✎	✕	201	NULL	NULL	Second line	C:\Users\Mano\Desktop\photo\201.jpg
☐	✎	✕	202	NULL	NULL	Third line	C:\Users\Mano\Desktop\photo\202.jpg
☐	✎	✕	500	roads	Rampart str	NULL	C:\Users\Mano\Desktop\photo\500.jpg
☐	✎	✕	501	Church Str	roads	NULL	C:\Users\Mano\Desktop\photo\501.jpg

Figure 5 Data records in Object table



Galle heritage Foundation
This was built by Portuguese and used as a Portuguese warehouse. However it was rehabilitated by Dutch and used as a hospital and a police barrack.

Figure 6 Data retrieval process

The procedure for triggering an on click action is to develop a special Sketchup tool which is a special Ruby Sketchup interface which needed to create a SketchUp tool. This tool will be able to pick the point where the user clicked, locate the 3D model which correlate with the point and display information. The methods of the class should contain the instructions to handle the required event.

7. Conclusion

It was concluded that a 3D model can be developed using the information stored in a relational database. The parameters and the attribute data of objects are stored in tables of RDBMS. These data were retrieved using a Ruby extension to develop the relevant 3D models. In this approach, the user need not have extensive knowledge using software tools. All what user needs to know are primitive database operations such as add, select, modify and delete.

The main advantage of this system is that any modification to the model is done at the database level. This is a much more efficient and convenient method than modifying the 3D model using the generic tools of the 3D modeling software.

In this research, the proposed procedure was illustrated using Sketchup, MySQL and Ruby. However several problems were encountered due to incompatibility of the versions of three packages. As such there is a requirement to build a compatibility matrix of this software.

References

[1] Collingbourne, H. (2008) Little book of Ruby. 2nd Ed. Hartland: Dark Neon Ltd. Retrieved from http://www.sapphiresteel.com/The-Little-Book-Of-Ruby .

[2] Nelson, W. A. (1984) The Dutch Forts of Sri Lanka, The southern group. Scotland: Canongate of Edinburgh.

[3] Peter G. E, Paul A. T & Drury B.C, (2008) Energy Design Plugin: An Energyplus Plugin for Sketchup, National Renewable Energy Laboratory, United States Department of Energy, Washington, DC

[4] Salamanca L.A.R, Ospina F.R.M & Vargas G.N.A (2013), 3D model city to enable Bogota as a tourist city, Esri International User Conference Paper Sessions,

[5] Sketchup Ruby API Documentation (2008) Model class [Online] Retrieved from: http://www.sketchup.com/intl/en/developer/docs/ourdoc/model.php#latlong_to_point.

Appendix A

Main Program

Code	Purpose
require 'sketchup.rb'	Call Sketchup library
require 'mysql-2.7.3-mswin32/ext/mysql.so'	Call MySQL library
UI.menu("PlugIns").add_item("originalhouse_main")	Initiate the plugin
{	

model = Sketchup.active_model
shadowinfo = model.shadow_info
	Develop Sketchup active model

shadowinfo["City"] = "Galle" shadowinfo["Country"] = "Sri lanka" shadowinfo["Latitude"] = 6.029753 shadowinfo["Longitude"] = 80.213746	Set geographic location for the object
con = Mysql.new('localhost', 'root', '', 'gallefort')	Connect to the database
rs = con.query('select * from buildings')	Query the records in the buildings table
con.close	Close the database connection
rs.each_hash do |r|	“rs” is a hash object. The Ruby each_hash method iterates through the hash object and tears it into record blocks

building_id = Integer(r['building_id'])
object_id = Integer(r['object_id'])
coordinates = r['coordinates']
z = Float(r['z'])
height = Float(r['height'])
	Five attributes values are assigned to Ruby variables

array = Array.new
array = coordinates.split("/")
pts = Array.new
 	
array.length.times do |i|
ary = array[i].split(",")
f_lng = Float(ary[0])
f_lat = Float(ary[1])
pts[i] = model.latlong_to_point [f_lng,f_lat]
end
	Convert the polar coordinate values to a real point in the Sketchup layout system

points = Array.new
points[0]= Geom::Point3d.new (pts[0].x,pts[0].y,z)
points[2]= Geom::Point3d.new (pts[1].x,pts[1].y,z)
buildinghouse.new(model,points,height,object_id).creategroup
end
	The values are passed to the crated Ruby class

}	

Appendix B

Building class

Code	**Purpose**
class Building	Define the Ruby Building class
def initialize(model,array,h,id)	Initiate the initialize method of Ruby
@model = model @array = array @h = h @id = id @pts = Array.new num = 0 for i in @array @pts[num] = i num = num + 1 end	The initialize method initializes the parameters of the building object and these values will be assigned to the object attributes within the object
end	End of the initialize method
def buildinghouse	Initiate the buildinghouse method
entities = model.entities group = entities.add_group group_entities = group.entities	Create a group
/Building development code /	Building development code
entity = group_entities[0] entity.set_attribute("id_table", "id", id) id_str = @id.to_s componentdefinition= group.to_component.definition.name=id_str	Assign particular building's id value to the 3D model in the interactive map. This is will be later used to retrieve information about the building through the map.
end	

Appendix C

For loop

Code	Purpose
w = 3700	Assign values to the number of columns (num) and gap between columns(w)
num = 24	
for i in 0..num	The location of the next column is decided within the For loop. The circle_column method is called to develop the column.
y = i*w	
pt = Geom::Point3d.new 0,y,4000	
ents = Sketchup.active_model.entities	
circle_column(pt,ents)	
end	
def circle_column(pt,ent)	Initiate the circle_column method
@pt = pt	
@ent = ent	
x = @pt.x	
y = @pt.y	
z = @pt.z+100	
square1 = @ent.add_face [x-260,y-260,z+0], [x+260,y-260,z+0], [x+260,y+260,z+0], [x-260,y+260,z+0]	
square1.pushpull 100	
round1 = @ent.add_circle [x+0,y+0,z+100.005], [0,0,1], 260	
round1_face = @ent.add_face(round1)	
round1_face.pushpull 100	Develop the column
circle = ent.add_circle [x+0,y+0,z+200.01], [0,0,1], 220	
circle_face = ent.add_face(circle)	
circle_face.pushpull 3500	
round4 = ent.add_circle [x+0,y+0,z+3700], [0,0,1], 260	
round4_face = ent.add_face(round4)	
round4_face.pushpull 100	
square2=@ent.add_face[x-260,y-260,z+3800],[x+260,y-260,z+3800],[x+260,y+260,z+3800],[x260,y+260,z+3800]	
square2.pushpull 100	
end	End of the method

Appendix D

Tool class

Code	purpose
class MyTool	Define the tool class "MyTool"
def onLButtonUp(flags, x, y, view)	MyTool has instructions in only one method, the "onLButtonUp" method. It means when user release the left mouse button is released, the method will run
ph = view.pick_helper	Pickhelper class was used to pick entities that reside under the current cursor location.
ph.do_pick x,y	Return the picked entities.
componentinstance = ph.best_picked	The "best_picked" returns the best entity picked (the 3D model which user clicked on).

name = componentinstance.definition.name
int = Integer(name)
	Identify the id of the 3D model resides under the user click.

con = Mysql.new('localhost', 'root', '', 'gallefort')
sql = "select name, description, photopath from objects where object_id=#{int}"
rs = con.query(sql)
con.close
	Connect to the database
	Query the result set relevant to the given id value "int".

rs.each_hash do |r| @name = (r['name']).to_s @description = (r['description']).to_s @photopath = (r['photopath']).to_s end	The result is a hash object which contains attribute values for the relevant record. The each_hash method will iterate through the hash object and spitted into field values

dialog = UI::WebDialog.new("Details", true, "", 410, 875, 1030, 0, true)
dialog.add_action_callback("pass_data")
{|dialog,htmlpage|
java1 =set_id(#{@name.inspect},#{@description.inspect})"
java2 = "set_photo(#{@photopath.inspect})"
dialog.execute_script(java1)
dialog.execute_script(java2)
}
dialog.set_file 'C:\Program Files\Google\Google SketchUp 8\Plugins\set_id\set_data_oct25.html'
dialog.show()
end
end
	The objects table has two attributes as description, photopath.
	These values then passed to a HTML page to display.

command = UI::Command.new("MyTool")
{
Sketchup.active_model.select_tool MyTool.new
}
tool_menu = UI.menu "Tools"
tool_menu.add_separator
tool_menu.add_item command
	Add MyTool to the Sketchup tool set.