Computer Animation Production 1
Maya Python Scripting Assignment Report

i7208422
Sim Chun You

Contents

User Manual

Detailed examples of how the program can be used are in the artefacts\Videos section.
Create a project and import the textures into the source images folder.
To run the code, copy the code into the script editor and run it.
To enable the AI system, run in the expression editor: **python("update("+frame+")");** and make sure the renderer is set to: **Legacy Default Viewport**

Introduction

Inspired by city building games such as Sim City, I tried to create a python program that allows the user to customise his own building or use existing ones and create his own city by placing objects along grids.

Algorithm

Object Placement

For the placement algorithm, I need to find the coordinates and rotation of a selected face to place an object.
First, I calculate both the normals and the coordinates of the face centres of the selected face by using the face points (Figure 01). Next, I did 2 different versions of the dot product rule to get the rotation values. (Figure02, Figure03). The first method involves looking at the normal of the face in 3D space and rotating accordingly. The second method involves representing the normal of the selected face as 2 2D vector. These methods give different results as shown in (Figure04).

Figure01

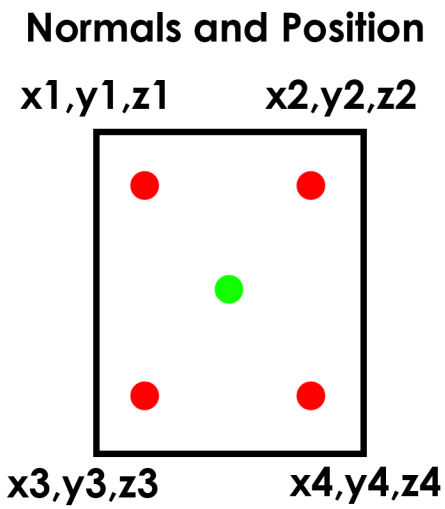


Figure02

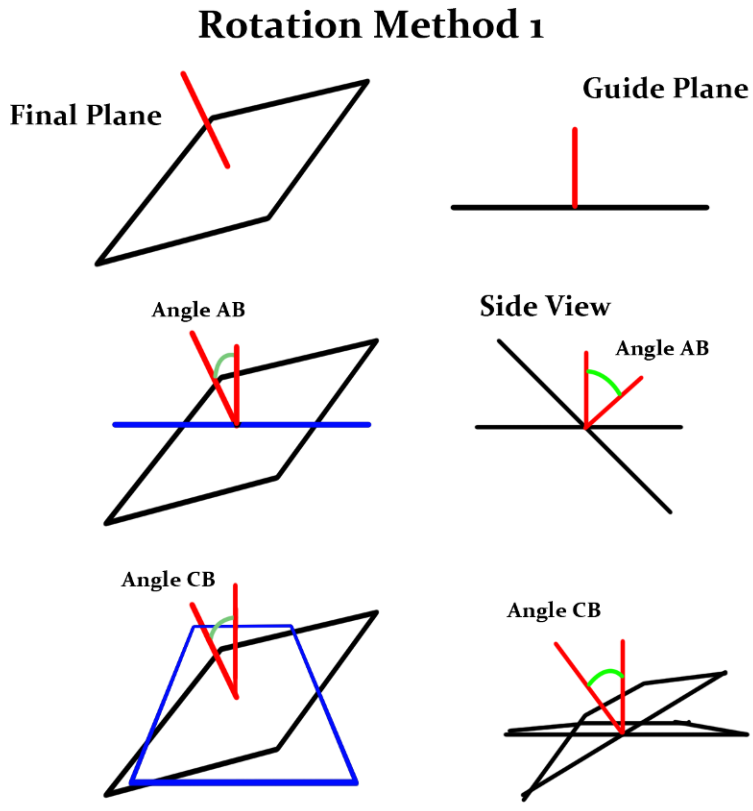


Figure03

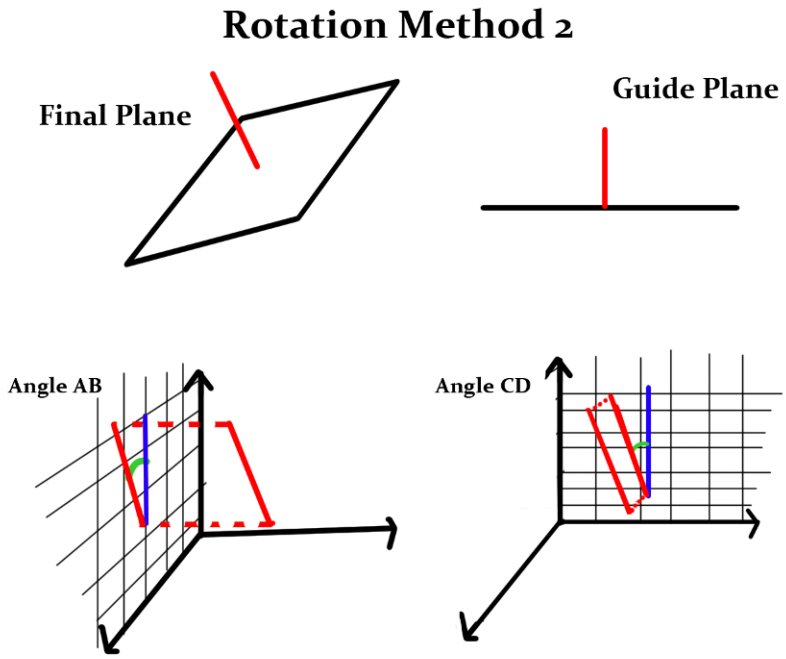
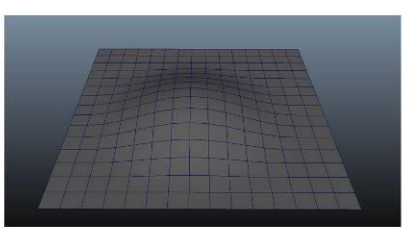
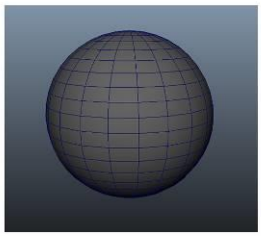


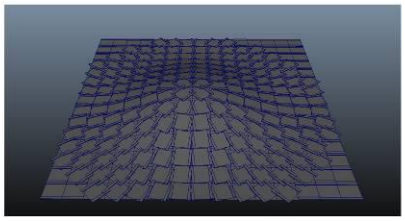
Figure04



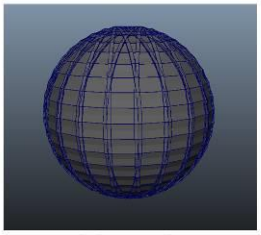
Method 1



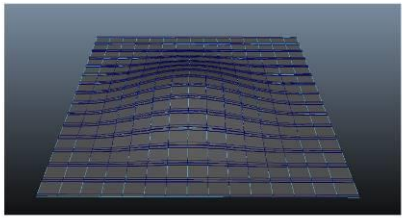
Method 1

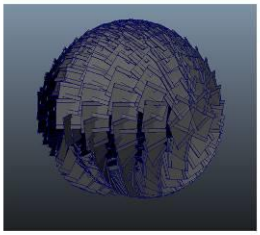


Method 2



Method 2





Traffic Simulation

For the traffic simulation to work, I tried using the concept of finite state machines. This is done by having an object undergo various states and performing various actions depending on the conditions that it will meet.

I used Ramesh Balachandran's approach to creating traffic simulation. I create several lists with each element of the list representing an object as well as other information needed for calculations (Figure05, Figure06). With the expression **python("update("+frame+")");** being created in the expression editor, every time I move the time slider, the function 'update' will rerun, allowing all calculations to be recalculated and all lists to be updated. For each time the update function is run, the code check whether the car is in any junction, if it is near any traffic lights, cars etc and if true, it will react accordingly while updating its state in the list(Figure07, Figure08). A more detailed representation of the update function is shown at the bottom of the report. For a quick summary of the calculations done in the update function, each time the function is run, it will check and update the state of the traffic lights, check and update the rotation and position of the cars and check if a car is near a traffic light, another car or at a junction.

Figure 05

AI list reference (part 1)

juncList

Contains all the junctions.
#0 = junction name
#1 = junction type

carList

Contains all the cars
#0 = car name
#1 = car state
#2 = car in which junction index.(-1 if not in junction)
#3 = car action count
#4 = car previous state
#5 = car state before stop
#6 = car previous junction count

trafficList

Contains all the traffic except for the linked traffic lights
#0 = traffic light name
#1 = current state
#2 = wait count

linked_trafficList

Contains all the traffic lights list that have links
#00 = All the traffic lights that are linked
#01 = All the traffic lights corresponding states
#02 = All the traffic lights corresponding wait count

Figure 06

AI list reference (part 2)

trafficRoadList

Contains all the road with a traffic light build on it
#0 = Name of Road
#1 = Name of Traffic Light
#2 = State

RoadList

Contains all the roads

T_J_RoadList/X_J_RoadList/straightRoadList/ CurveRoadList

Contains all the respective roads
#0 = Name of Road
#1 = State (except for straightRoadList)

Figure07

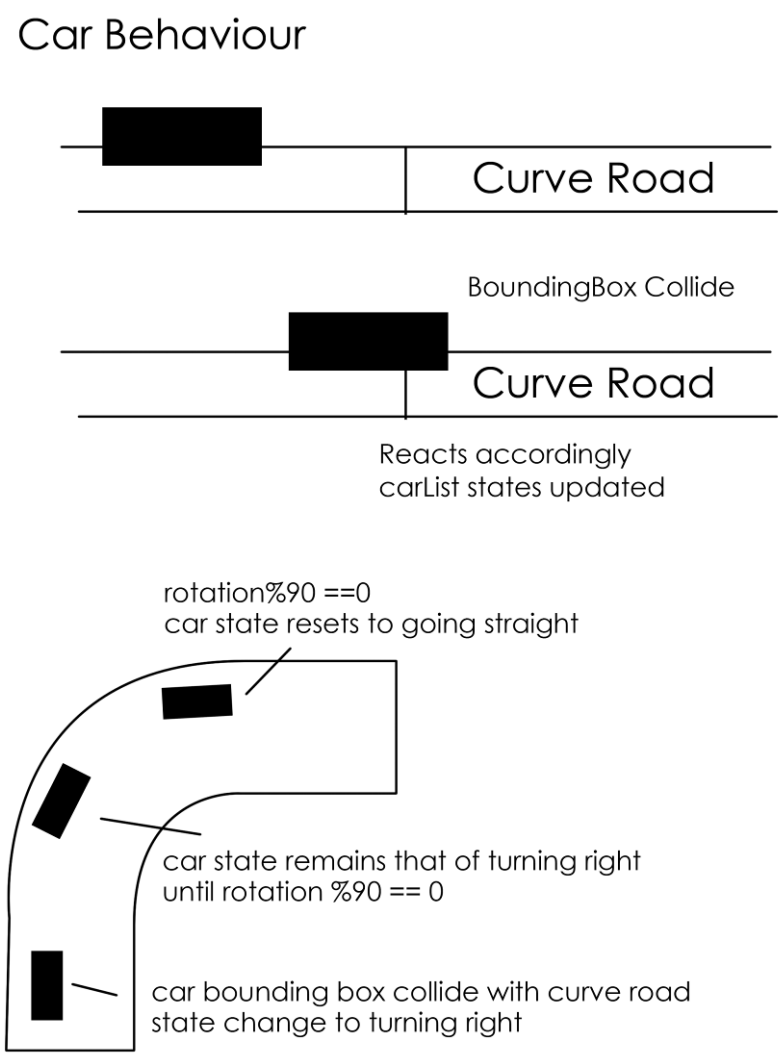
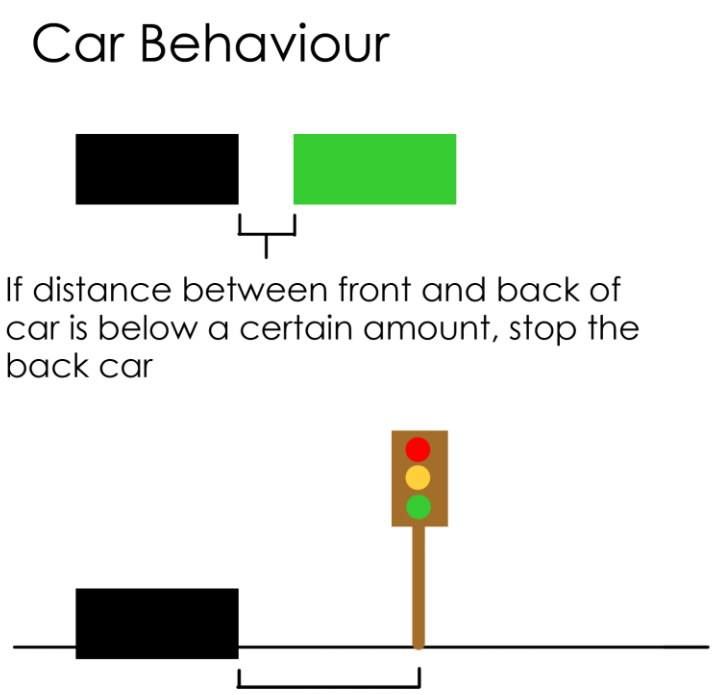
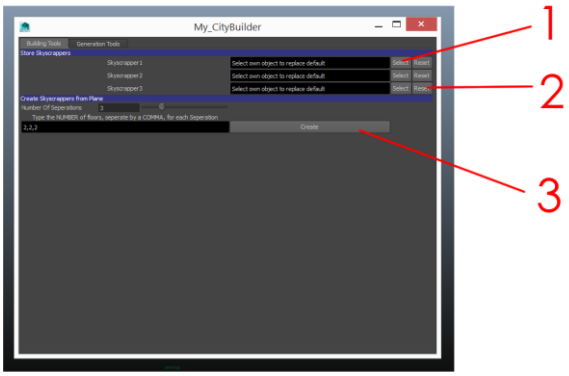

Car Behaviour
Curve Road
BoundingBox Collide
Curve Road
Reacts accordingly
carList states updated
rotation%90 ==0
car state resets to going straight
car state remains that of turning right
until rotation %90 == 0
car bounding box collide with curve road
state change to turning right

Figure08


Car Behaviour
If distance between front and back of car is below a certain amount, stop the back car
If distance between traffic light and car is at a certain amount and the traffic light is facing the car and is red, stop the car.

Flow of control

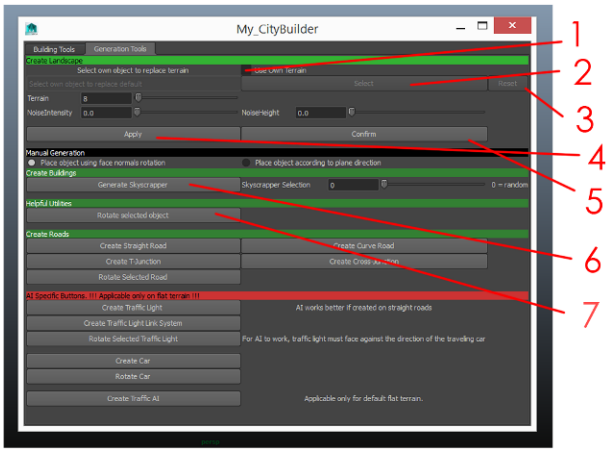
The flow of control from the user interface to the various functions are as shown below.



1 UserObjectSelected()

2 ResetUserObjectSelected()

3 actionCreateSkyscrapperFromPlane()



1 actionUsingOwnTerrain()

2 UserObjectSelected()

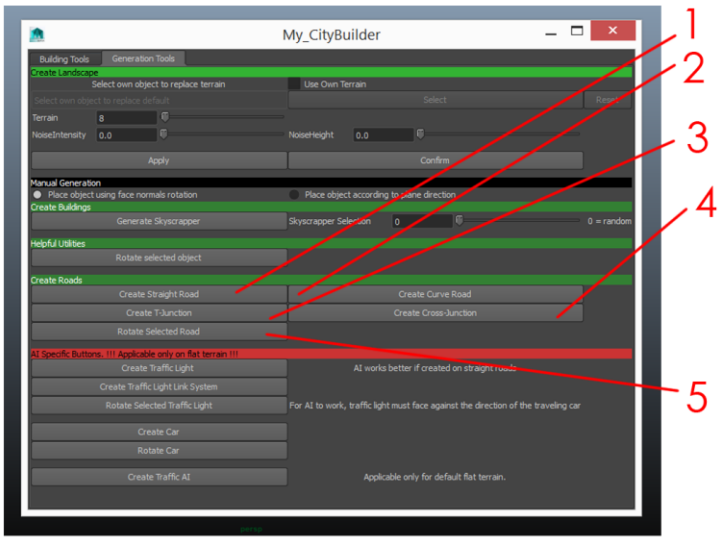
3 ResetUserObjectSelected()

4 actionGenerateTerrain() → createTerrain() → createTerrainNoise()

5 actionConfirm()

6 actionGenerateSkyscrapper() → calculateSelectedFaces() → generateSkyscrapper() → placeStructure()

7 actionRotateObject()



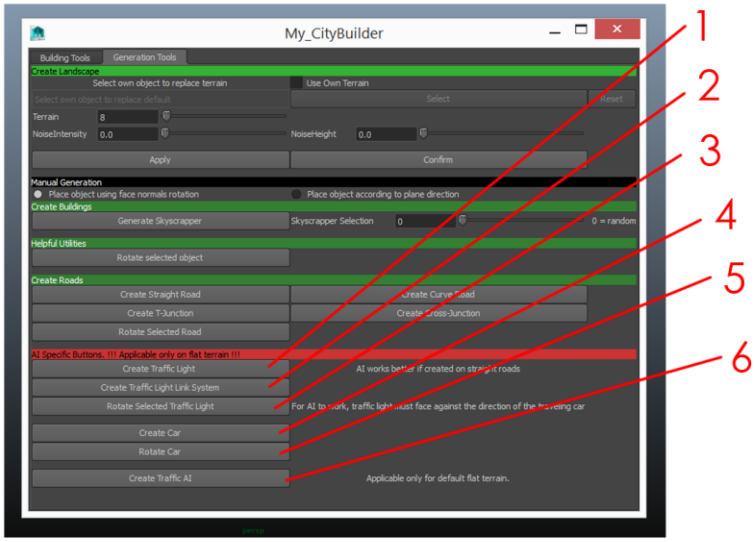
1 actionGenerateStraightRoad() → calculateSelectedFaces()
→ generateStraightRoadList() → placeStructure()

2 actionGenerateRtCurveRoad() → calculateSelectedFaces()
→ generateRtCurveRoadList() → placeStructure()

3 actionGenerateTJunctionRoad() → calculateSelectedFaces()
→ generateTJunctionRoadList() → placeStructure()

4 actionGenerateCrossJunctionRoad() → calculateSelectedFaces()
→ generateCrossJunctionRoadList() → placeStructure()

5 actionRotateRoad()



1 actionGenerateTrafficLight() → calculateSelectedFaces() → generateTrafficLightList() → placeStructure()

2 actionCreateTrafficLinks()

3 actionRotateTrafficLight()

4 actionGenerateCar() → calculateSelectedFaces() → generateCarTempList() → placeStructure()

5 actionRotateCar()

6 actionAIPrep()

Traffic Simulation flow (update function)

These set of procedures will be run everytime the
time slider frame is changed

Check and change the state of the traffic light

Look at how many turns a traffic light has been red/orange/green.
Change after a certain limit of turns

Do some checking and updating of lists.

AngleTweak() - Help to reposition and clamp angles rotation
inWorld() - Check if cars are still on the plane
PositionTweak() - Make sure that if the car is traveling on a road, it keeps to the correct lane after finishing a turn.

Check if car is near a traffic light

onTrafficRoad() - Check if cars are on a traffic road.
If true, check the distance between the car and traffic lights.
If it is below a certain amount, check the state of the traffic lights.
If it is 'R' red, or 'O' orange:
    change car state to stop.
    state_carStop()
else:
    change car state to go
    state_goStraight()

Check if car is too near another car

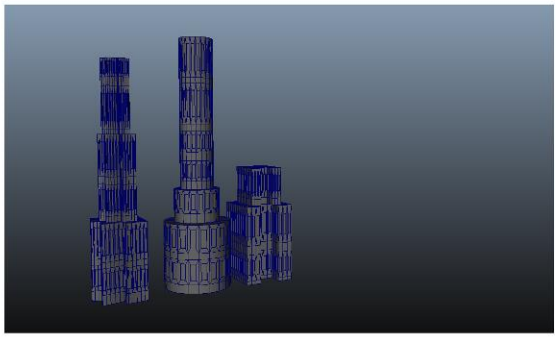
distance() - Check if cars are too close to each other or not
If below a certain limit:
    Check if the car isn't stopped before:
        If it wasn't stopped before, stop it. state_carStop()
If above the limit:
    Check if the car is stopped before:
        If it was stopped before, change it back to the state it was originally.

Check if car is at a junction

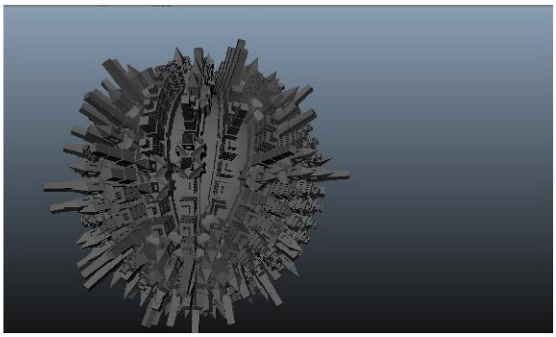
inJunction() - Check if car is in a junction or not.
If car is in a junction and previously not:
    Change the car state depending on the junction type.
If car is still in the process of turning:
    Continue turning until its angle % 90 == 0
If car is continuing straight at a junction (example cross junctions or t junctions):
    Update action count until a certain limit to not allow the car to change its decision to turn left or right halfway through crossing the junction.
If car has finished turning or not in the state of turning
    Make the car go straight.
    state_goStraight()

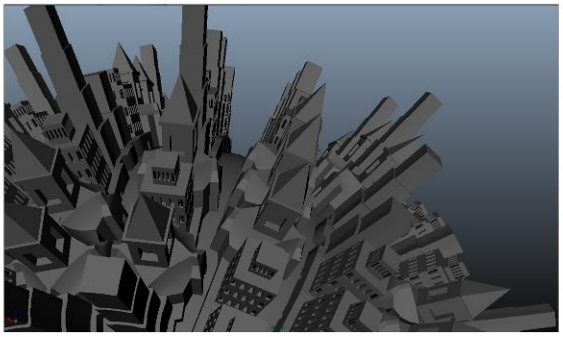
Results

Buildings from plane

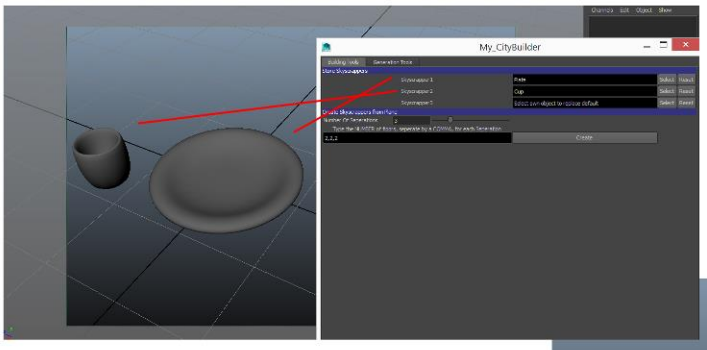


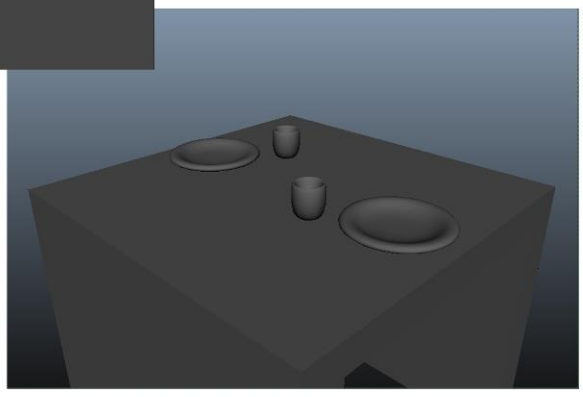
Generate buildings on any quad surface



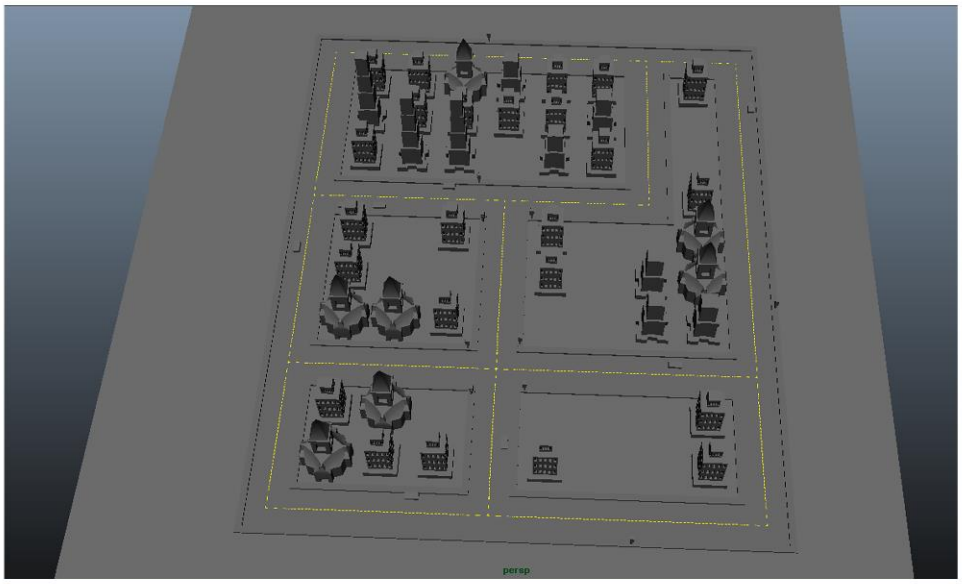


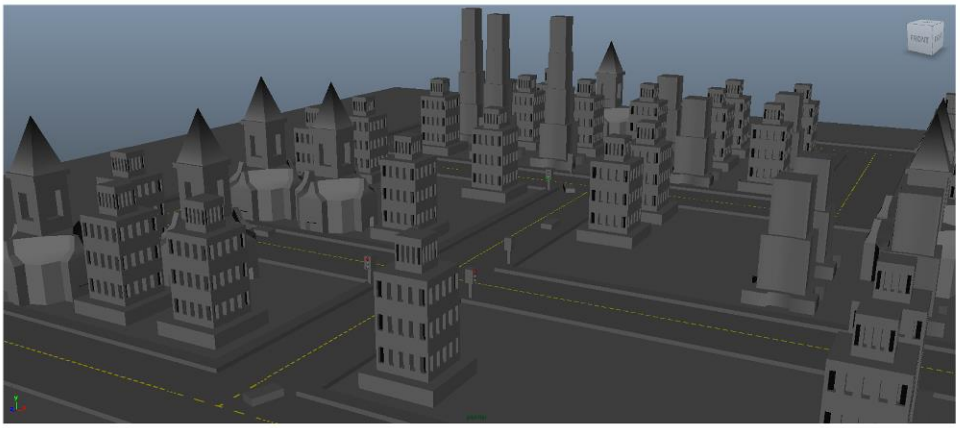
Using the program to place tables and cups on a table

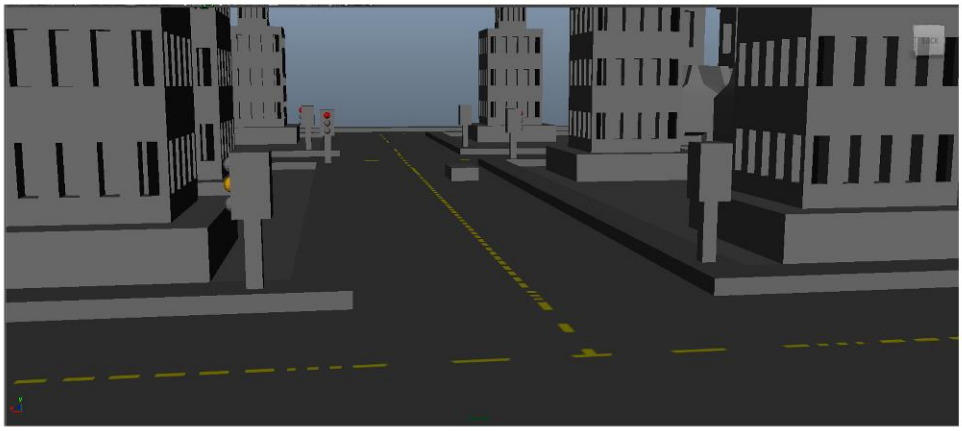




Generate a full customisable city with traffic simulation







Bibliography

Balachandran,R.,No Date.*AI_Script_Final.py*[online]. Available from: https://drive.google.com/folderview?id=0By0_WhpQpCZlWmNmaHYtMi1SbTQ&usp=sharing [Accessed 5 April 2015]