CityGenerator v 0.1

by Mikołaj Woźniak

Documentation

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND!

Remember it's only a demo and a lot of things need to be done and tested!

This software was created in after hours of my daily work in two weeks.

Do not use this software for: aviation & aeronautics, medicine, military etc.

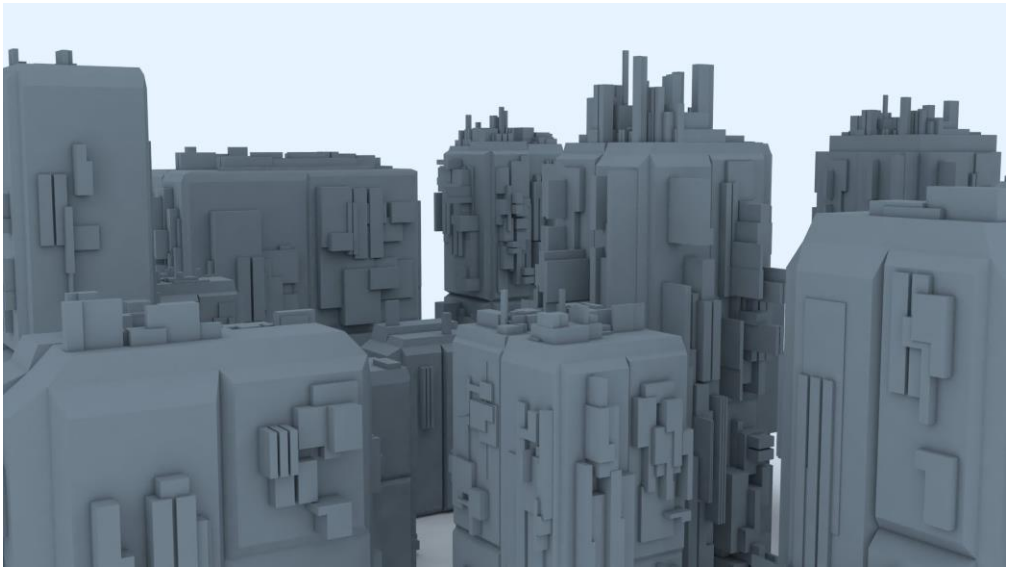
Created and tested with:
Unity 3d 5.3.6, Windows 10, Visual Studio 2015 & .net 4.5,
DevExpress 16.1, HelixToolkit (2014.2.1.1)

Day 0

Inspiration

I like medieval or fantasy games like Dark Souls and The Witcher but I'm loved in Sci-fi worlds and cities. My inspirations to this demo are sci-fi cites from cyberpunk universes like Ghost in the shell, Infinity The Game, Android Netrunner, Eve Online and Blade Runner.

Android Netrunner City



Greeble tool from 3ds max

Dynamic tower (architecture concept)

Mirrors Edge Minimalistic City

Unity 3D & WPF close integration

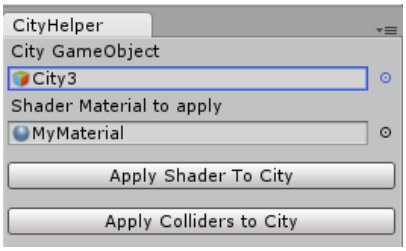
Close integration like using Unity 3D Scene in WPF or WPF ViewPort in unity 3D is almost impossible. Of course all in IT is possible but decompiling Unity 3D or WPF is time consuming. Unity scene or packages are binary format so it also require some kind of reverese engienering.

There is a way to create reusable code with external DLLs. I used it one time with OpenCV and EMGU in Unity 3d for face detection with Kinect but I have a lot of problems with it. It's time consuming method and in my opinion is not worth it. Unity 3D doesn't use .NET but Mono so you need to recompile whole library. Also the Mono version that Unity 3D is using does not support even the all of .NET 2.0 features (!).

Of course you can implement some logic in .NET 2.0 and use it with 4.5 WPF tool with multi-targeting options but you need to compile same code in Mono for Unity. But there is another problem: recreating visual preview of scene in Unity 3D and in WPF ViewPort as far as I know there isn't any framework to do this.

Possible approaches with WPF and Unity 3d integration

If you are creating tool to generate meshes with materials in WPF and export scene to *.obj or import from *.obj or others 3D common formats it’s not a problem but things can go tricky if you want to use something more complicated or uncommon for example objects like sharders, lights, animations, colliders or other features that WPF don’t provide. One way to integrate tool with Unity 3d and Unity 3d with your tool is your own protocol or format for example xml and reconstruction in both sides based on information in file. This solution have one disadvantage you need to write code for each 3d tool for example Unreal Engine, 3dsmax, blender, TopoGun etc as plugins.

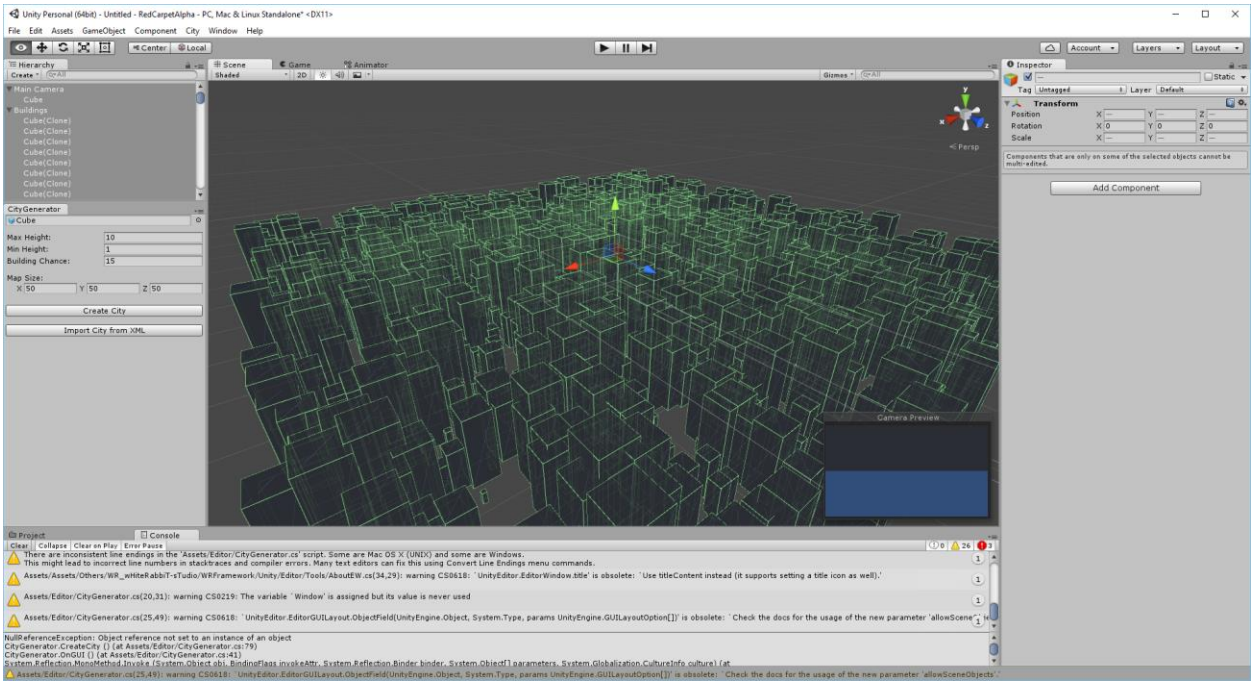


I decided to export data in *.obj and *.mtl format for meshes and materials, it’s simple and common format for 3d models. It’s also possible to create / reconstruct other object like shaders, colliders etc in MenuEdit Helper plugin for Unity 3D.

It’s also possible to migrate lights and animations (Animatable class) from WPF to Unity 3D but you need to have come contract for each object to recreate it in Unity 3D from XAML/XML. It might be also possible to migrate even shaders with SharpDX integration. This things are not implemented yet due to time consuming. I had enough problems with *.obj export mainly with localization and textures.

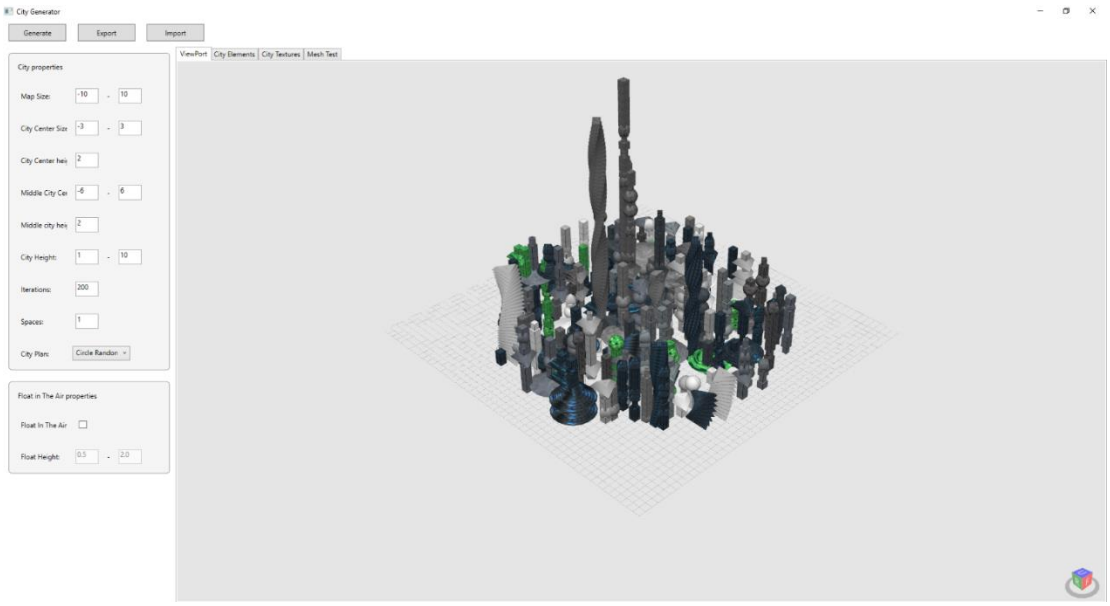
Day 1

Proof of concept as Unity 3D plugin

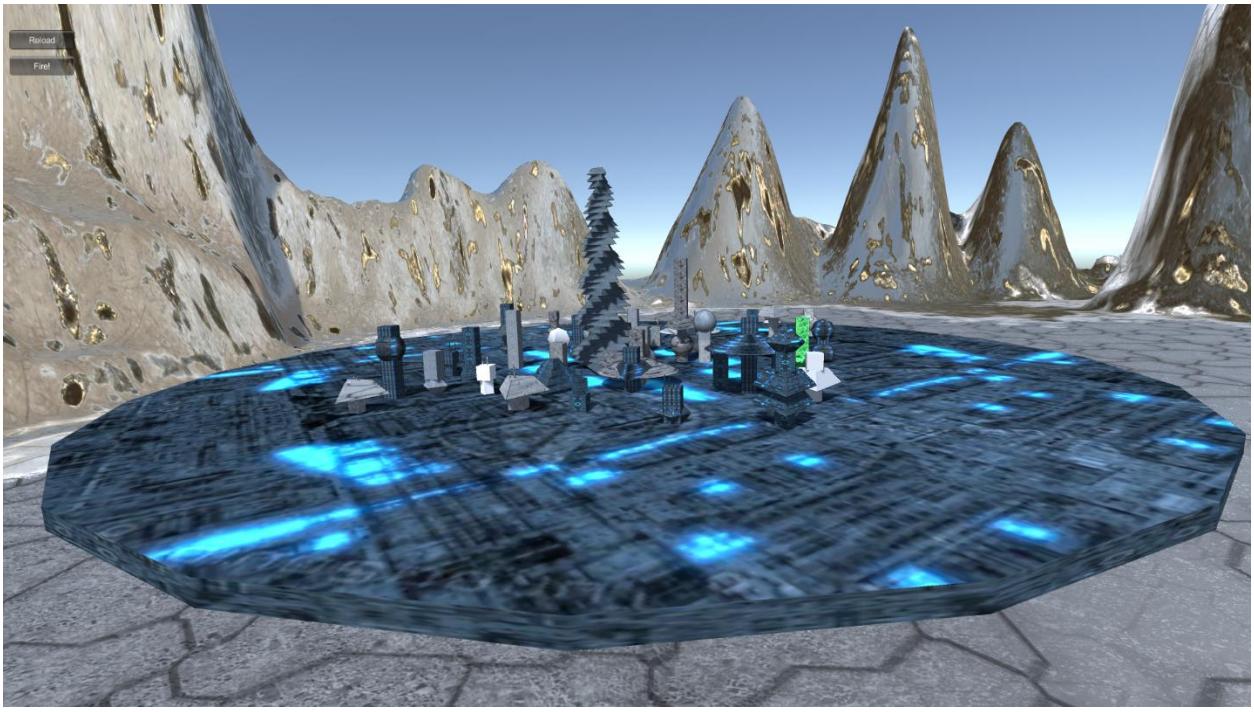


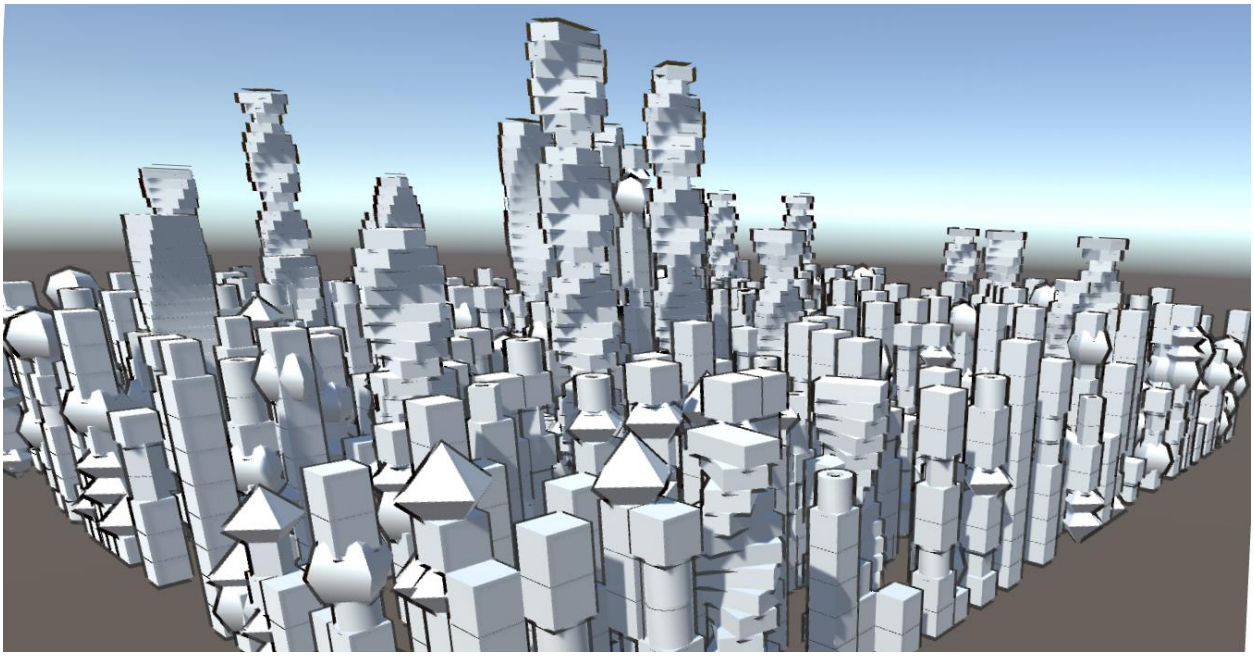
Day 5

WPF tool

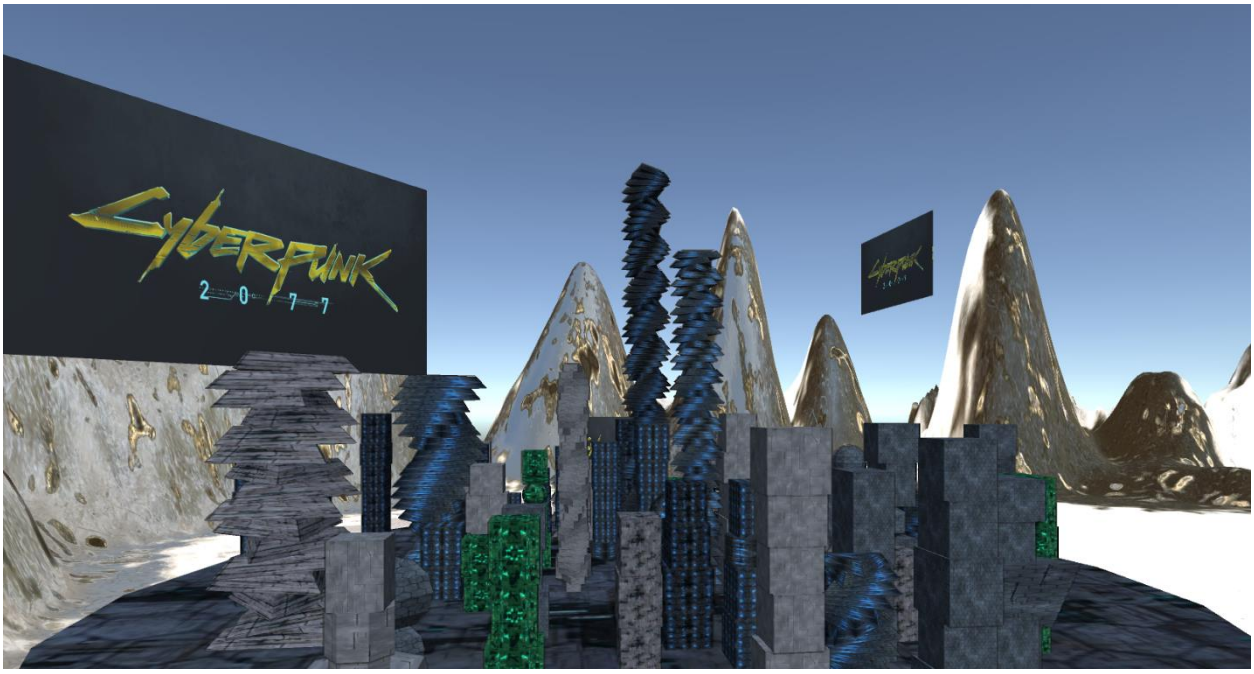


Day 8

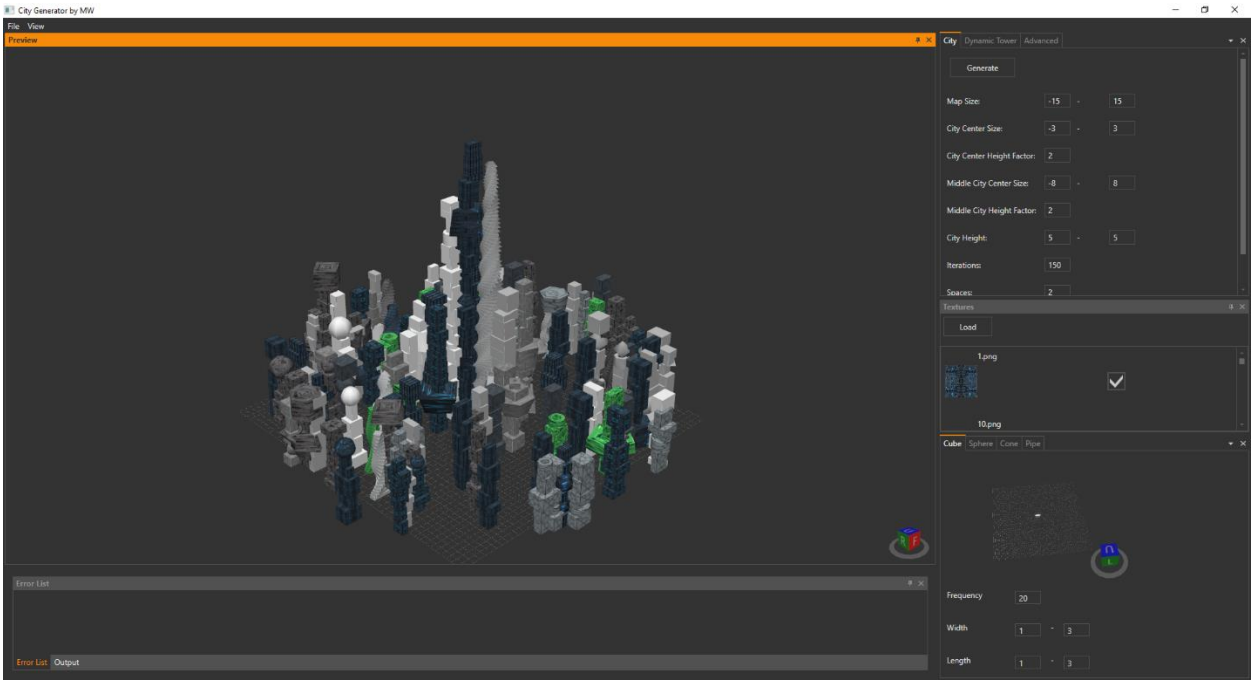




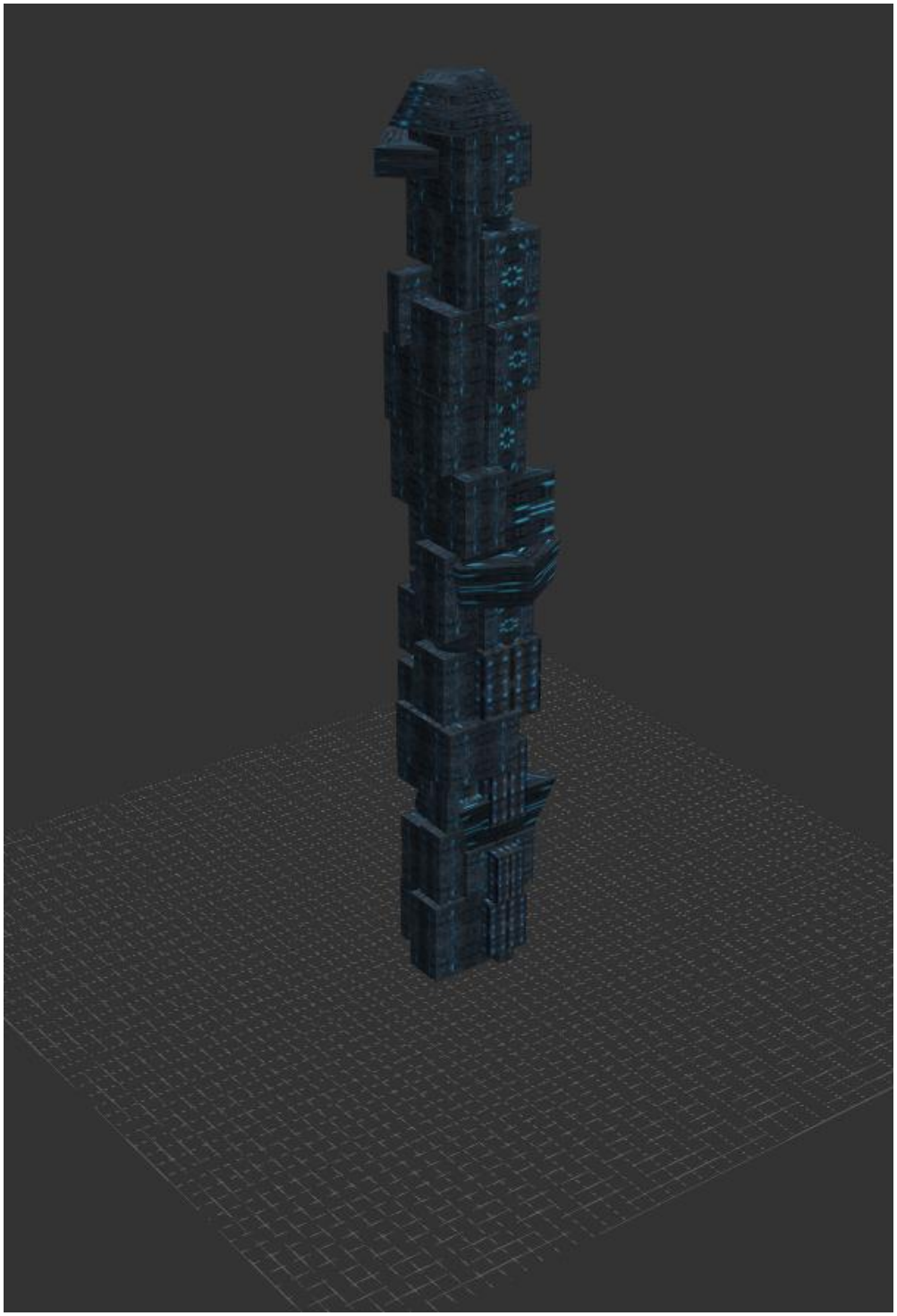
Day 11


CYBERPUNK
2077

Day 12


City Generator by MW
File View
Preview
City
Dynamic Tower
Advanced
Generate
Map Size:
City Center Size:
City Center Height Factor:
Middle City Center Size:
Middle City Height Factor:
City Height:
Iterations:
Spaces:
Textures
Load
1.png
10.png
Cube
Sphere
Cone
Pipe
Frequency
Width
Length
Error List
Error List
Output

Lacking motivation and inspiration? Creating sci-fi cities? Just use my tool in single building mode with low iterations and small map size and random city plan.



Using this tool in future real life scenario

In future workflow will be as follow:

- 3d modeler with texture graphic create city segments,
- Tool load segments of city from *.obj or other format and procedurally generated whole cities or just buildings based on segments with changeable logic.

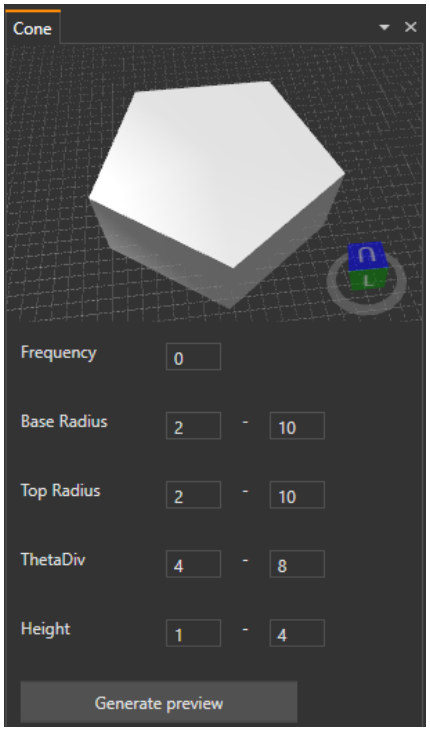
Parameters

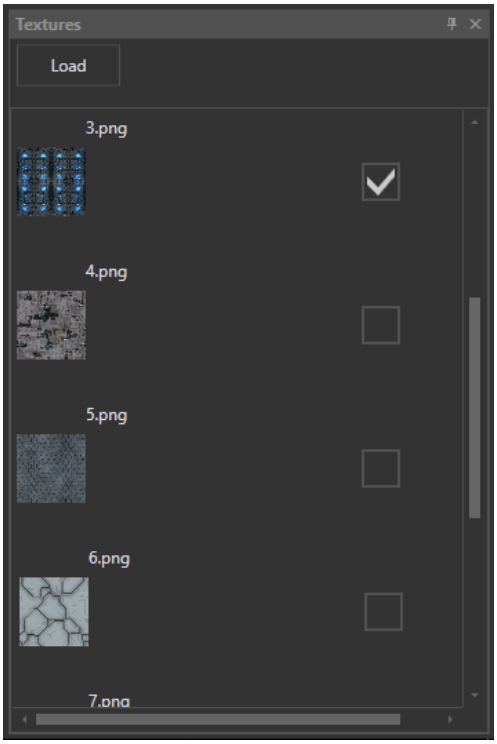
Each module have min / max parameters for example width, height etc. if you want get preview of each of it you need to wait 5 seconds or click “Generate preview” to reload it.

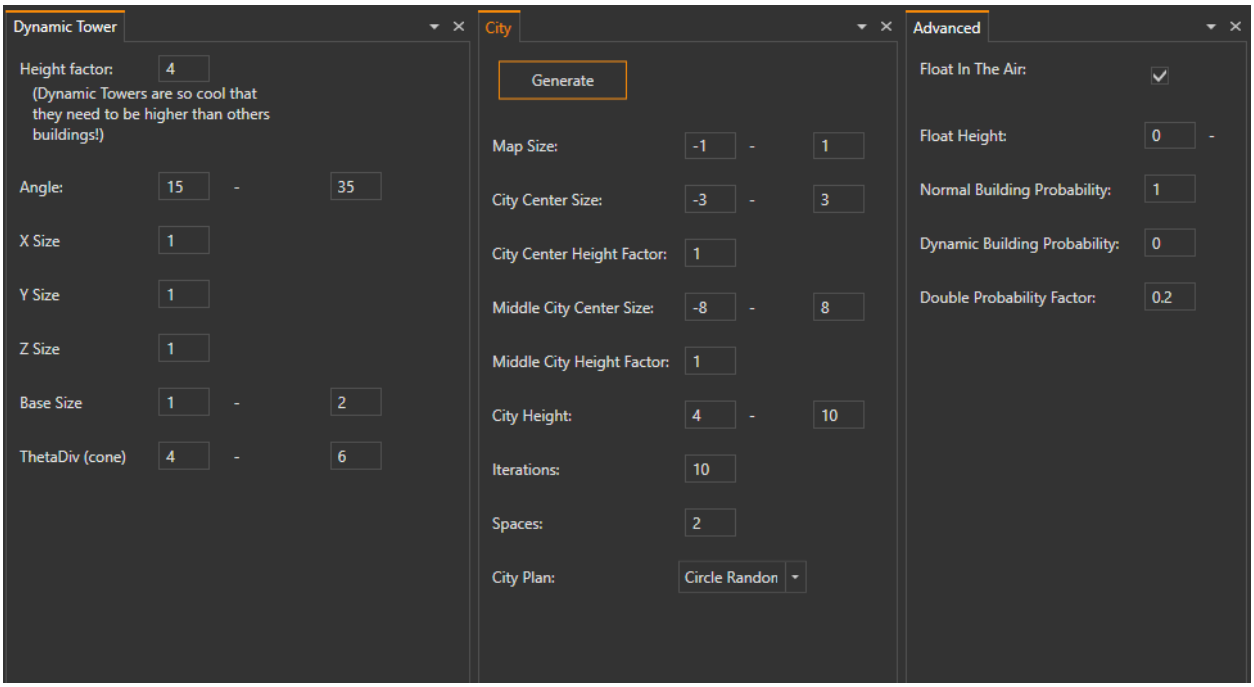
Remember that is only a demo and some parameters are not validate for example you can't create sphere with lack of triangles.

If you want to disable module use Frequency 0 property.

You can also enable/disable textures and load them from external folder as *.jpg, *.jpeg, *.png files.







Currently there are only **2 types of buildings, normal ones** created from random modules and **dynamic towers**. Each dynamic tower might by made of cones or cubes with 50/50 probability.

You can specify **map size** of city center, middle city and rest of city with parameters for city size. Also you can use height factor to scale your city in these areas.

Iterations are just number of buildings to create in one generation.

Spaces and city plans are not fully supported by now and need to be change and tested but you can try them out as they are today.

Float in the air (advanced option) is a feature to creating elements in distances for example for city with super ant-gravity boson higgs technology or just to create fancy single buildings.

You can also change **probability of each building type.**

Double factor (advanced option) is a scalar that can change probability for modules functionality. If you want more ordered buildings for example cubes with 0.5, 1, 1.5, 2.0 lengths you should set it on 0.5.

Dynamic towers have x,y,z sizes factors and min/max angle random per element per building. In future it might have different angles like tower in inspiration section

Unity 3d import Bug (occasionally / irreplaceable)

I have bad luck with 3d models exports and imports or it is just nature of 3d formats that they does not work properly (so why call them format anyway?). It's not the first time in my career that something is wrong. Firstly I had problems with scaling rotation and localization but I noticed that sometimes big cities are not possible to import with in Unity 3d due to this bug:

https://issuetracker.unity3d.com/issues/importfbx-errors-after-importing-obj-file

I have the same problem.

ImportFBX Errors:

Couldn't import file C:/Users/mikiw_000/Dropbox/Tool/CityProject/Assets/test3/export 2.obj

Also I'm wondering why the error message is "ImportFBX Errors: " while importing obj.

Every City works fine in Blender so I decided to use it to reimport and export for big Cities.

I also wanted to import *.obj file in Unity 3D runtime but this plugin https://www.assetstore.unity3d.com/en/#!/content/15862 works only with simple meshes.

Currently I'm analyzing decompiled code of Unity 3D to evaluate this problem.

Future

Version 2.0? 2.0 versions are funny things - there never appear.

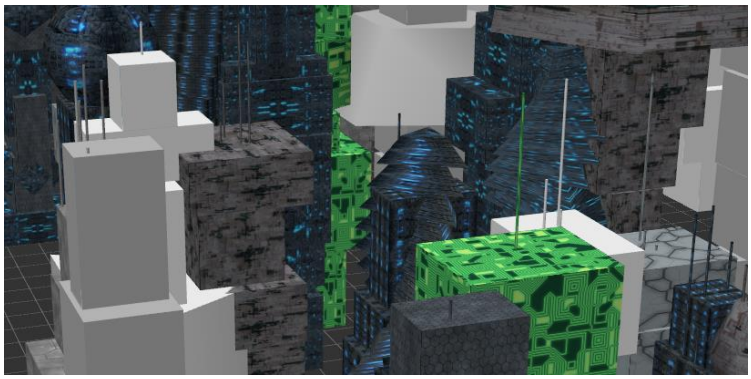
That is why I decided to name this section future

You can add infinity numbers of features and combinations of this tool but I decided to stop adding feature after one week. Like in oil peak I noticed that my code is a mess and I need to close and integrate it in working solution.

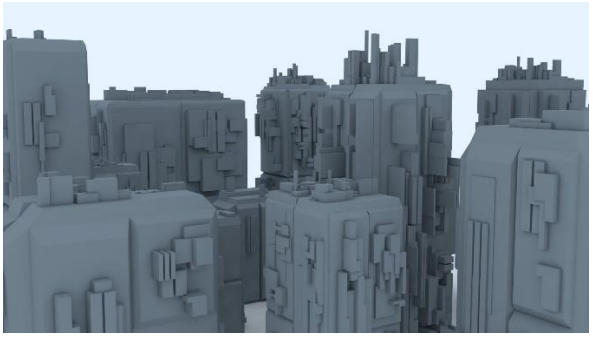
- Loading meshes as city segments from *.obj or other format and generating cities based on segments with texture.
- Import / Export issues textures etc
- Colliding meshes issues
- Collisions of elements because there is no collision detection out of the box in wpf / helixtoolkit / sharpdx so you need to write your own or check collisions or use Unity 3d collisions or create simple grid collision system based on localizations.
- Validation Framework with aspect programing for example in PostSharp. I Didn't create validation so be sure that max values are greater than min values, don't use double in int fields, you can't create sphere with 2 faces etc. etc. (sorry ☹)
- Integration With SharpDX for better ViewPort, tessellation options, shaders, animations, DirectX 11/12 and other stuff
- Possibility to move elements with mouse and drag & drop of new elements, scaling etc.
- Lights
- Generate City at terrain Mesh
- Streets both land and in the air because of anti-gravity higgs boson technology
- Machine Learning & Classifier for complicated and simple cities from previews and marked datasets
- Rule engine that allow to add rules like:
    - 2 of this elements in one building
    - Top of building only with this elements
    - 3 of this elements in row
    - Mathematical equations for city generator and buildings generator for example dynamic tower in trigonometric equation
    - Others
- Plugins in dll's files loaded with MEF
- Animations for dynamic tower with different speed per element.
- New buildings types with more configurations
- XAML export/import features in WPF/Unity 3d for elements like light, animations etc.
- Different variations of building modules (copy feature / add feature). It should work like list of possible elements not 4 modules like today.

- Generate city in runtime in Unity 3D. Unity 3D button click > fire outside city generator service > service generate *.obj and *.mtl > Unity 3D import model 3d with textures in runtime not in design mode.
- Improve tool UX after users feedback
- More improved texture mapping module (texture per module?)
- Import / export of tool configuration
- Import cities (its commented now but it's possible to do it)
- Log4net for exception handling and information like export/import errors or warning and integration with console module in GUI
- Code / Refactoring:
    - Lot of refactoring is still need to be made. I didn't move all code to mvvm pattern and also some OOP improvements are needed.
    - Dynamic generated controls based on properties based on reflection or mef modules
    - More Loose coupled code using ioc container and mvvm bootstrap from caliburn micro
- More options, more parameters and more details! And much much more...

For example antennas (this module is unfinish so I commented it in code)



Greeble like mesh modifier module

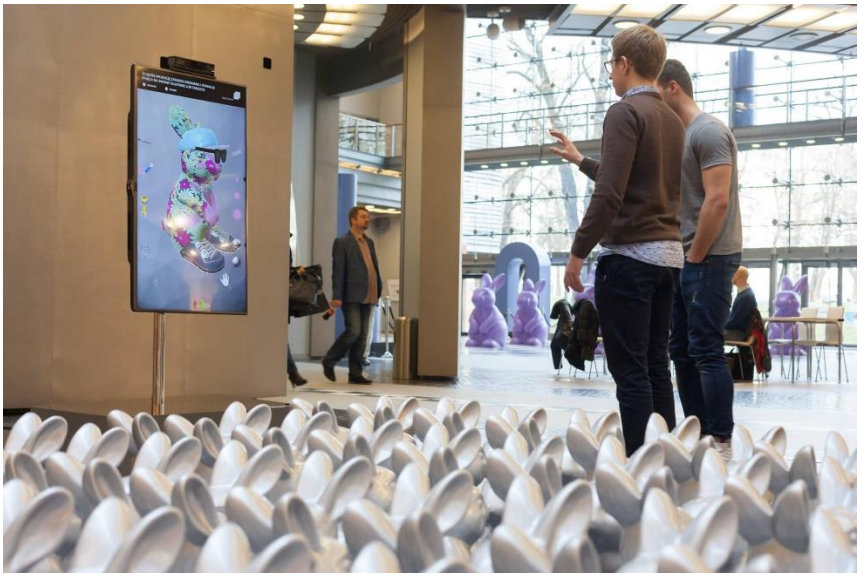


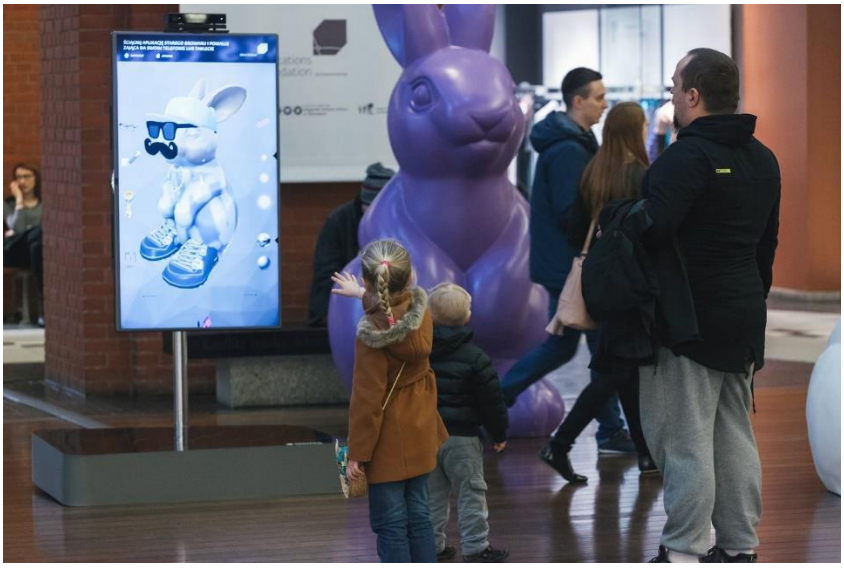
Some of my projects

Easter Bunny for Stary Browar Mall Center

Challenges:

- Kinect v1 and v2 integration
- Low level optimization of texture painting
- Smooth hand tracking
- Unmanagement Kinect code





Virtual Merchandising (Gear VR)

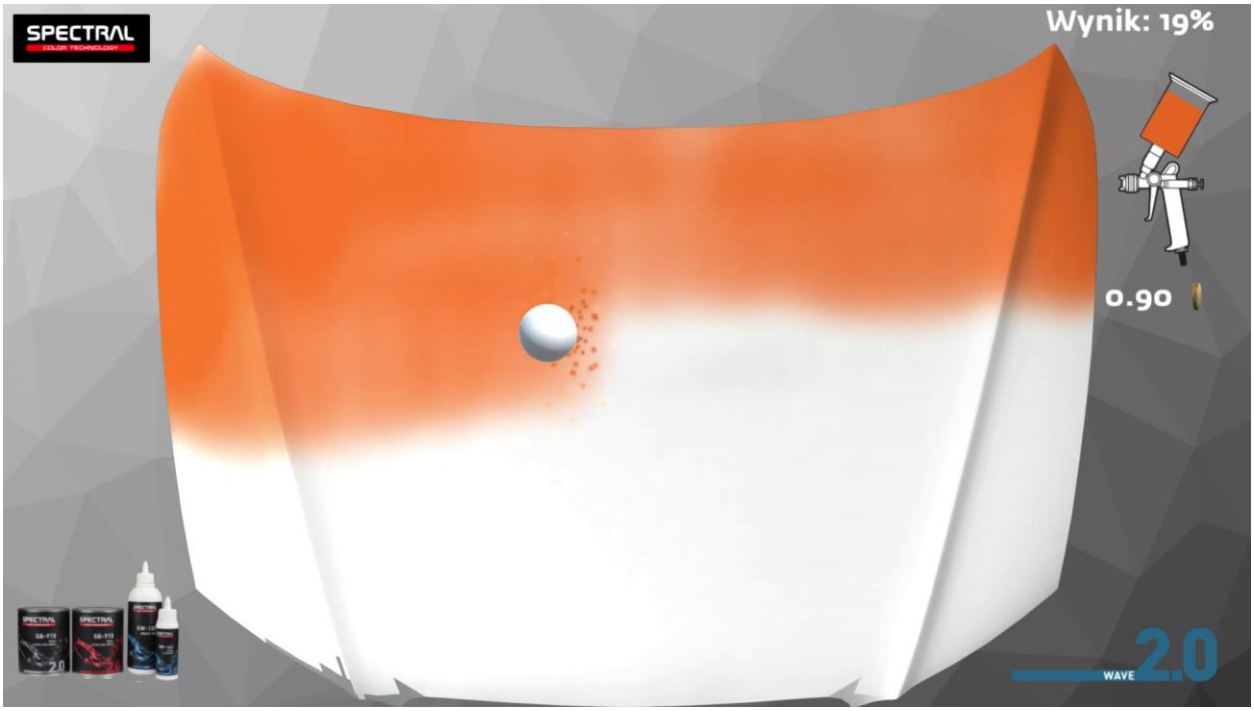
Challenges:

- Gear VR optimization for Samsung smartphones
- Perfect sphere tessellations for scene without headache (it's very important with these technologies)
- Camera Animations
- Measure time of looking per object

Virtual Car Spraying

Challenges:

- Simulation of car spraying mechanics and counting effectivity based of movements
- Leap motion integration in v1.0
- HTC Vive integration in v2.0



And many more

Little red carpet Game.