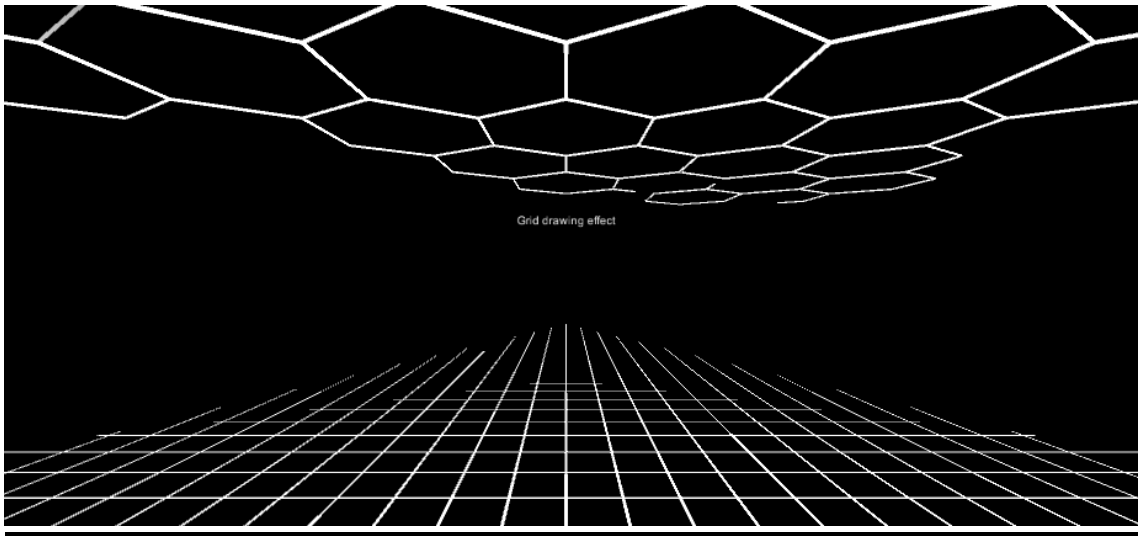
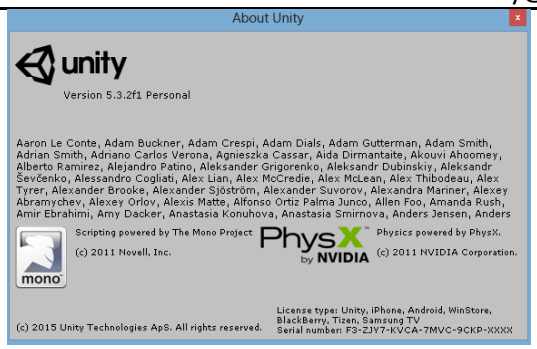
Dynamic Grid Drawing



1. GENERAL INFORMATION

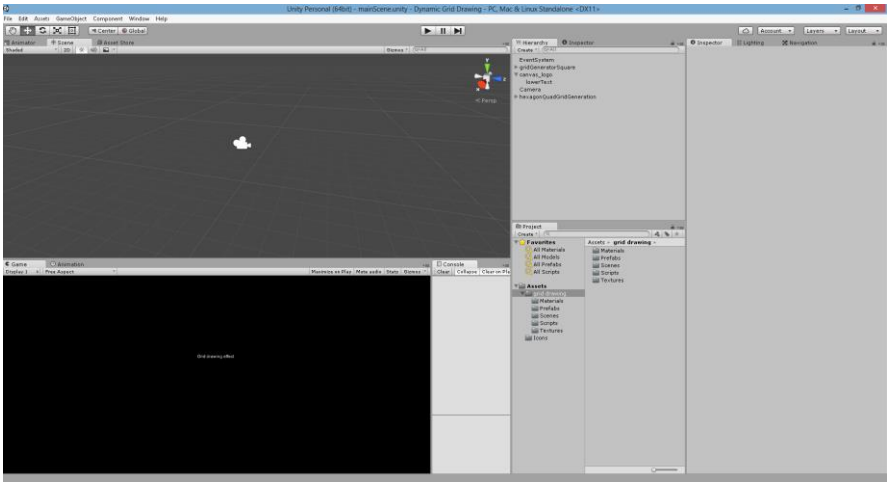
DATE OF DOCUMENT	26/03/2016
NAME OF THE PROJECT	Dynamic Grid Drawing
AUTHOR	Michael Soler
UNITY VERSION	5.3.2.F1 PERSONAL
CONTACT	michael.soler.beatty@gmail.com



Index

2. IMPORTING INFORMATION

This package works independently. Your project should look like this once imported:



Select “mainScene” from the scenes folder.

3. PROJECT DESCRIPTION

This package creates dynamic hexagonal and square grids from trail rendered. This means that a seed is moving in the actual grid, while the trail creates the effect of a complete grid object. Trail time and generation speed can be modified by script. This package contains:

- Hexagonal dynamic grid script.

- Square dynamic grid script.

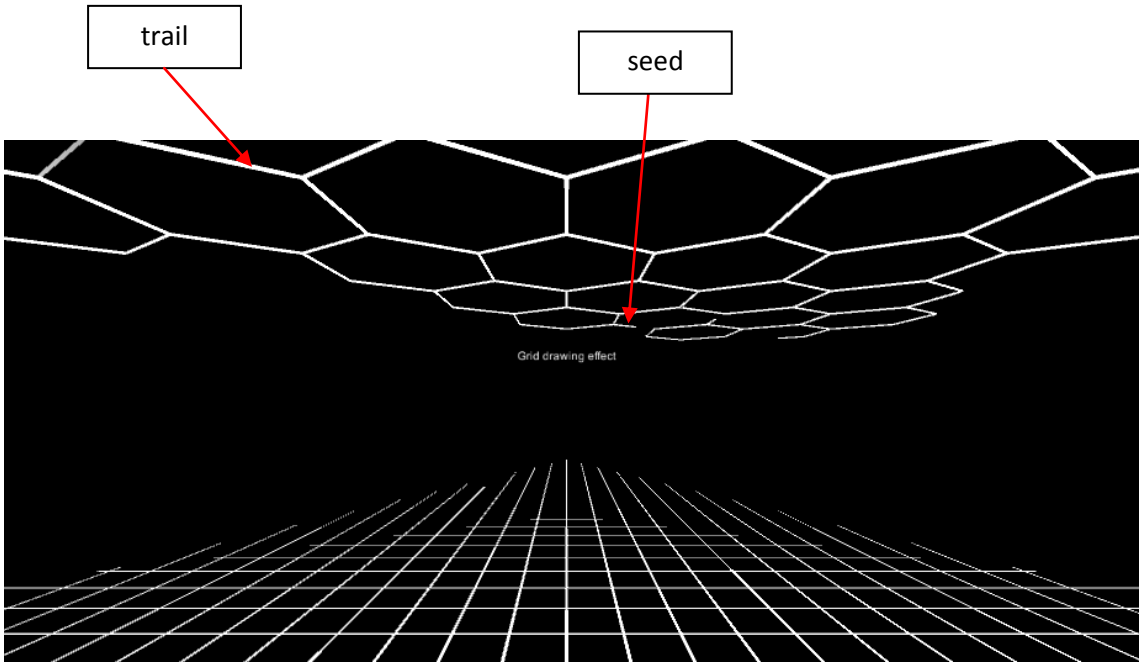
- Square resizing grid script.

- Square and hexagonal static grid scripts.

This package could be also used to generate AI movement in a plane following hexagon or square grids.

The program works as follows:

A sphere object (which no renderer) moves between the vertices of the grid while its trail is shown. This gives the effect of a grid creation.



4. LAYERS, TAGS AND COLLIDERS

1.LAYERS

All objects are placed on the default layer.

2.TAGS:

All gameobjects are untagged.

3.COLLIDERS

No colliders are used

5. SCRIPTING INFORMATION

We explain each script with some detail in the following table:

- **AutoResize.cs:**

  It is used to create the drawing effect.

IMPORTANT VARIABLES
public float speed → is the growing speed of the line. **public float maxSize** → is the maximum size of the line after the iterations. **public float initialSize=0.03f** → is the initial width of the line
public void fixedUpdate() → The size of the line is modified after a certain amount of time.

- **GridSquare.cs:**

  It is used to create new grid lines in the case of the square grid.

IMPORTANT VARIABLES
public GameObject line → is the line object that will be copied. **public float indx=1;** → is the grid index. **public float squareSize;** → it is the size of one square **public float maxNumberOfLines;** → max number of lines that will be created (total grid width). **public float elapsed , timeBetweenLines, increment;** → There variables are used to measure the time between line creation.
public void fixedUpdate() → four lines are created if the elapsed time respect criteria.

- **MoveInHexagon.cs:**

  It manages the movement of an object in hexagons.

IMPORTANT VARIABLES
public float distance; → the distance that the particle moves before changing its direction. **public float divisor;** → it is the inverse of the speed. An exact division must be used to prevent grid gaps or discrepancies.
Void giveNewPoint()→ this function seeks for the next grid point.

- **MoveInSquare.cs:**

  It manages the movement of an object in squares. It has a similar structure as the previous function, but grid points are calculated differently.

IMPORTANT VARIABLES
public float distance; → the distance that the particle moves before changing its direction. **public float divisor;** → it is the inverse of the speed. An exact division must be used to prevent grid gaps or discrepancies.
Void giveNewPoint()→ this function seeks for the next grid point.